Signal Integrity Toolbox™

User's Guide

MATLAB®

R2022a

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™ User's Guide

Trademarks

Patents

Revision History

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

Contents

1 Serial Link Examples

2 Configure Serial Link

3 Pre-Layout Analysis of Serial Link

4 Post-Layout Verification of Serial Link

5 Serial Link Featured Examples

6 Configure Parallel Link

7 Pre-Layout Analysis of Parallel Link

8 Post-Layout Verification of Parallel Link

9 Parallel Link Featured Examples

10 Jitter and Noise

11 Industry Standard Examples

12 Signal Integrity Topics

1

Serial Link Examples

PCIe-3 Compliance Kit

This example shows how to test the compliance of simulation models and topologies to the PCI Express generation 3 (PCIe-3) specification.

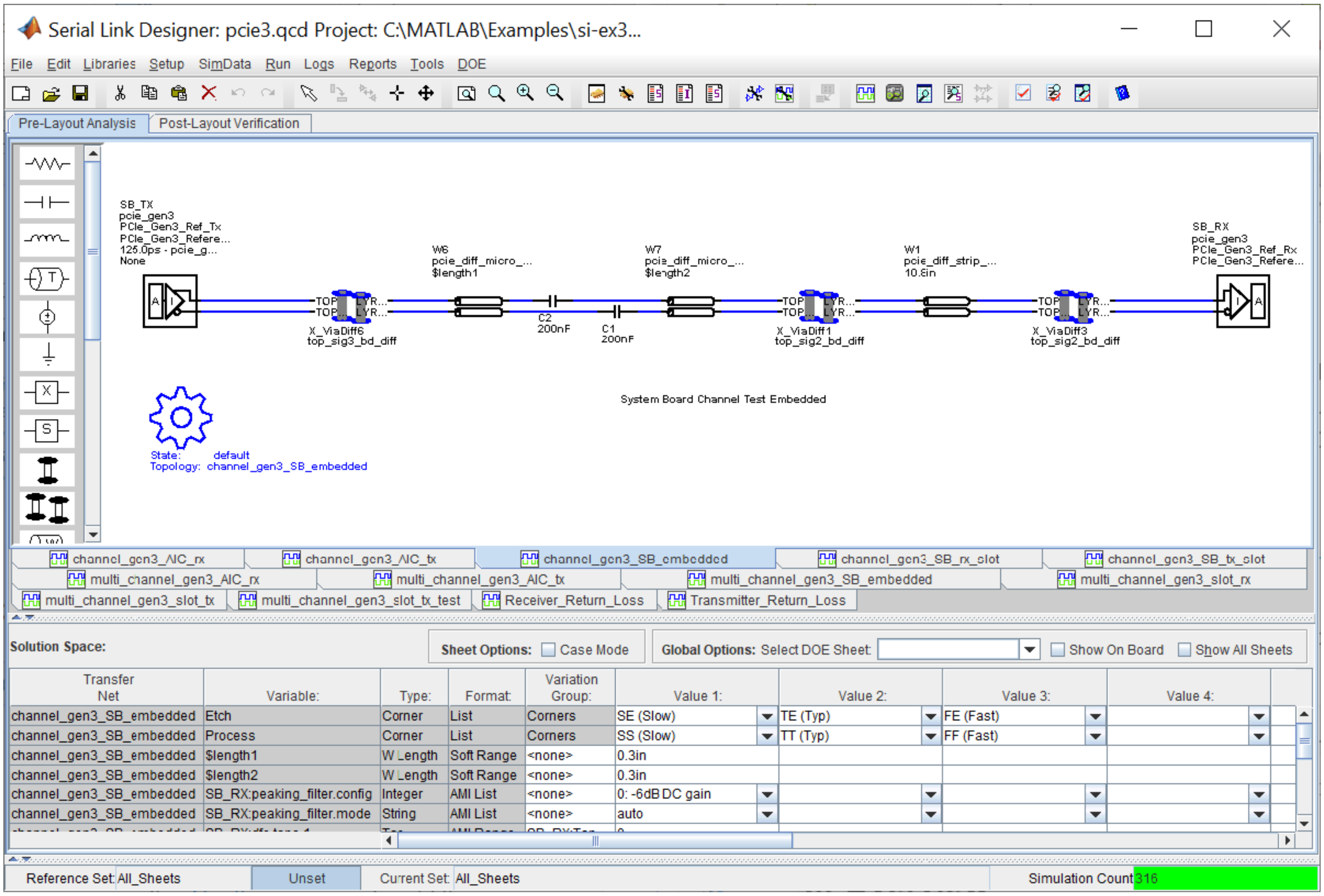
The PCIe-3 signal integrity kit includes all the transfer nets, topologies, generic buffer models and compliance rules for a PCIe-3 high-speed SerDes interface. This includes PCIe-3 technology IBIS-AMI models for the SerDes transmitter and receiver, PCIe-3 compliance masks and transfer nets preconfigured for TX and RX characterization that are customizable for a specific PCIe-3 add-in card (AIC), system board (SB), and PCIe-3 embedded channel.

This kit enables you to insert a channel design and characterize and validate its performance using the specification masks to determine if the channel has a high confidence of success. If the channel does not meet the compliance masks or BER estimates, further investigation or redesign, along with simulation, will need to be performed to determine possible changes to meet compliance.

Open PCIe-3 Kit

Open the PCIe-3 kit in the **Serial Link Designer** app using the `openSignalIntegrityKit` function.

```
openSignalIntegrityKit("PCIe_Gen3_NVMe");
```



Serial Link Designer: pcie3.qcd Project: C:\MATLAB\Examples\si-ex3...

Solution Space:

Sheet Options: □ Case Mode

Global Options: Select DOE Sheet: □ Show On Board □ Show All Sheets

Transfer Net	Variable:	Type:	Format:	Variation Group:	Value 1:	Value 2:	Value 3:	Value 4:
channel_gen3_SB_embedded	Etch	Corner	List	Corners	SE (Slow)	TE (Typ)	FE (Fast)	
channel_gen3_SB_embedded	Process	Corner	List	Corners	SS (Slow)	TT (Typ)	FF (Fast)	
channel_gen3_SB_embedded	$length1	W Length	Soft Range	<none>	0.3in			
channel_gen3_SB_embedded	$length2	W Length	Soft Range	<none>	0.3in			
channel_gen3_SB_embedded	SB_RX:peaking_filter.config	Integer	AMI List	<none>	0: -6dB DC gain			
channel_gen3_SB_embedded	SB_RX:peaking_filter.mode	String	AMI List	<none>	auto			

Reference Set: All_Sheets Unset Current Set: All_Sheets Simulation Count 316

Kit Overview

- Project Name: PCIe_Gen3_NVMe
- Interface Name: pcie3
- Target Operating Frequency: 8.0 Gb/s, 4.0 GHz (Nyquist) (125ps)

The PCIe-3 kit defines four schematic sets:

- **All_Sheets**: All schematic sheets
- **AIC**: Schematic sheets for add-in card design
- **SB_Slot**: Schematic sheets for system board with slot design
- **SB_Emb**: Schematic sheets for system board embedded design

For more information about the PCIe-3 channel compliance schematics, transfer net properties and compliance rules, refer to the document PCIe_gen3.pdf that is attached to this example as a supporting file.

Configure Serial Link

Simulation Parameters Used in Serial Link Design

You can set parameters that control how a simulation is run in **Serial Link Designer** using the Simulation Parameters dialog from the **Setup** > **Simulation Parameters** menu item. This dialog contains a table with parameters, their values, and the part of the analysis flow they affect. You can sort the columns by clicking on the table headers.

Simulation Parameter Definitions

Parameter	Description
Samples Per Bit	Number of time steps in a bit time. Defines the time step used in the **Serial Link Designer** ".tran" statement.
Max Channel Delay	Maximum length of the channel impulse response supplied by the user. This value is also used in FFT block size calculation that defines the message length used for statistical analysis. For more information, see "Determining FFT Block Size" on page 2-5.
Target BER	Array of bit error rates (BER) to measure eye height and width. The array is sorted from the smallest to the largest. If fewer than four values are entered the results will include four values. The additional values are created by multiplying the last value by `1e3`.
Minimum Ignore Bits	Start time for Time Domain waveform analysis. Allows time for all of the AMI models to reach steady state. This is used if models do not define Ignore Time set in the AMI model, or the defined Ignore Time is less than this value. The larger of this value or a value from a model is used as the Ignore Bits for the analysis.
Max Input Frequency	Maximum frequency valid for the network model, determined by the maximum frequency for which S parameters are available. This frequency can be limited or extended by the user. For more information, see "Max Input Frequency" on page 2-4.
Max Output Frequency	Maximum frequency output for transfer function and S parameters describing the end to end passive electrical interconnect. For more information, see "Max Output Frequency" on page 2-4.
S-Param Frequency Step	Frequency step size for S-parameters output from the **Serial Link Designer** app. The value of this parameter controls the behavior of the Serial Link app: • Auto : The frequency step is $\frac{1}{6 \times D}$, where D is the longest through path delay in the network. • Non-zero value: Use this value as the S-Parameter Frequency Step.
Record Start	Time to start saving waveforms in a Time Domain simulation.
Record Bits	Number of bits of the waveform to save.
Time Domain Stop	The stop time of the Time Domain simulation.
Block Size	The number of samples in a single waveform segment in a Time Domain simulation. This sets the granularity of the parameter outputs returned by AMI models. Also used in determining FFT block size.

Parameter	Description
Output Clock Ticks	If yes, then Time Domain simulation will output the recovered clock ticks to a file.
STATify	Control how statistical techniques are applied to Time Domain simulations and Getwave-only models. The values are: • Stat_with_Getwave: Uses a PRBS and derived pulse response from time domain analysis as the basis for statistical analysis. Allows statistical analysis to be done for models that are Getwave-only. • TD_Extrapolation: Extrapolates the bathtub curve to account for the effects of ISI at lower probabilities than can be derived from the Time Domain simulation alone. When this parameter is selected, Time Domain simulation does the following: • Run a PRBS pattern at the end of the Time Domain simulation • Generate a pulse response for the equalized channel from the PRBS data. • Generate a statistical eye from the pulse response • Use the statistical eye to extrapolate the bathtub curves For the extrapolation to be accurate the clock recovery loop and DFE (if any) must be settled at the end of the time domain analysis. • Both: Perform both TD_Extrapolation and Stat_with_Getwave. • None: Do not perform TD_Extrapolation or Stat_with_Getwave.
Results Storage Control	Determine which results to store.
Time Domain Crosstalk Mode	How to account for crosstalk in Time Domain simulation. • Semi-Analytic — Crosstalk is accounted from statistical analysis. • Explicit — Crosstalk is active during Time Domain simulation.
SPICE Rise Time	Transition time from 0 to 100 percent of the stimulus input to the Driver.
SPICE Sample Interval	The time step used in the SPICE ".tran" statement. The value is an integer greater than 1 or units of time in seconds. If the value is an integer then it is the number of time steps in a bit time.
SPICE Buffer Models	• LTI — The Tx and Rx buffer models used in the SPICE simulations are LTI models. • IBIS/SPICE — The Tx and Rx models are SPICE transistor models or IBIS behavioral models depending on the serial link app setting.
SPICE Ignore Bits	The time before the start of the SPICE step in the step response simulation. It is either in UI or in units of seconds.
SPICE Step Stop	Stop time of SPICE step response simulation.
SPICE Time Domain Stop	Stop time of SPICE Time Domain simulation.
Include IBIS Package	Include (Yes) or do not include (No) IBIS Package model as defined in the IBIS file.

Parameter	Description
Conductor Roughness	Surface roughness of conductors in microns (RMS). Used for lossy transmission line models created outside Serial Link app.
NC/TD Simulation Mode	Simulator used for network characterization (NC) and time domain (TD) phases of channel analysis. Mode used is based on modes supported by models. • Prefer Native/Native — Use the Serial Link app engine for network characterization if the models support it. Use the Serial Link app engine for time domain analysis. • SPICE/Native — Use SPICE for network characterization. Use the Serial Link app engine for time domain analysis. • SPICE/SPICE — Use SPICE for both network characterization and for time domain analysis.
Tx Spectral Table	Specifies the spectral table to use for the transmitter. The list box shows the spectral tables that have been imported into the project libraries.
Rx Spectral Table	Specifies the spectral table to use for the receiver. The list box shows the spectral tables that have been imported into the project libraries.
Spectral Analysis Resolution BW	Resolution bandwidth of clock spectral analysis.
Clock Analysis	Clock phase noise spectral density analysis and output.

Max Input Frequency

Maximum frequency valid for all network models, determined by the maximum frequency for which S parameters are available. This frequency can be limited or extended by the user. The value of this parameter controls the behavior of the Serial Link app:

- *Auto or Zero* — This option is an automatic mode. Serial Link app engine calculates the maximum frequency from a combination of the sample interval and the S-Parameter blocks in the netlist. The app then chooses the highest frequency for which all the circuit elements can be defined. If there are no S-Parameter blocks, then the highest frequency is defined by the equation:

  $$\frac{1}{2 \times SampleInterval}$$

- *Non-Zero Value* — Use this value or the maximum frequency that defines all the S-Parameters whichever is smaller as the maximum frequency of the network model. The frequency value effects the TDR rise time via the following equation:

  $$\frac{1}{\pi \times MaxInputFrequency}$$

  The default is Auto.

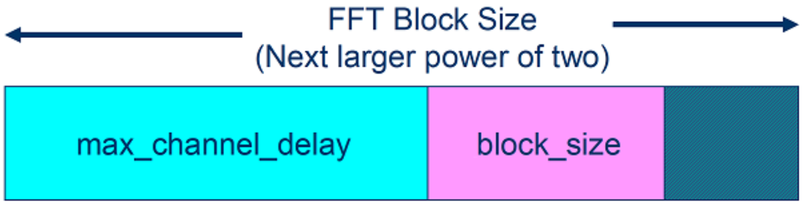
Max Output Frequency

Maximum frequency output for transfer function and S parameters describing the end to end passive electrical interconnect. The value of this parameter controls the behavior of the Serial Link app as follows:

- Auto or Zero — This option is an automatic mode. Serial Link app sets the output frequency to 1.5 × *DR*, where *DR* is the highest data rate of any TX in the analysis. The default is Auto.
- Non-Zero Value — Use this value or the maximum frequency.

Determining FFT Block Size

Two of the parameters, Max Channel Delay and Block Size, determine the FFT block size used in network characterization and statistical analysis. The actual FFT block size is rounded up to the nearest power of two.

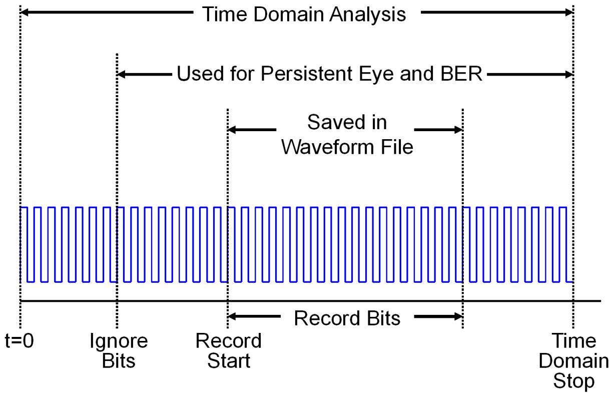


Time Domain Start and Stop Parameters

The time domain parameters that control the start and stop of the simulation are:

- Time Domain Stop
- Record Start
- Record Bits
- Minimum Ignore Bits

This figure demonstrates the relationship of several Time Domain simulation parameters.



See Also

More About

- “Specify Corner Conditions in Serial Link Design” on page 2-7
- “Stimulus Patterns in Serial Link Design” on page 2-9

- “Model Jitter and Noise While Designing Serial Link” on page 10-9

Specify Corner Conditions in Serial Link Design

Corner conditions are used to define process corners. In process corners, the parameters are within the specified range for that parameter but outside the range of normal operations. You can specify corner conditions using the Corners Conditions dialog from the **Setup** > **Corner Condition** menu item.

IC Environment Corners

The IC Environment Corners area contains the temperature parameter for each corner along with a voltage factor scale that can be used. The temperature setting is used as the .TEMP parameter in the SPICE simulations.

Note The temperature parameter does not affect IBIS buffer models as they are created at designated temperatures and can not be scaled.

The voltage factors are used to scale all voltage sources in the netlist. The typical corner value is scaled by the scaling factor to create the values for the slow and fast corners. For voltage sources, the value entered in the schematic or specified for a voltage net in post-layout is scaled by the scaling factor.

I/O buffer voltages can use the three values specified in the IBIS [Voltage Range] parameter for the three corners or use the typical value from the [Voltage Range] and scale it.

Etch Corners

Etch Corners specify the scaling factors for the Z_0 and T_{pd} parameters of transmission line models. These scaling factors account for manufacturing variation in the PCB. Both ideal and lossy transmission line models are scaled.

Lossy transmission line models are scaled by computing the values of Z_0 and T_{pd} from the typical corner L and C values. The computed Z_0 and T_{pd} are then scaled by the scaling factors to create the Z_0 and T_{pd} values for the slow and fast corners. The slow and fast corner L and C are computed from the slow and fast Z_0 and T_{pd}.

Impact of Corner Settings

The elements that are affected by corner settings are:

- **I/O buffer voltages**: If scaling is enabled for I/O buffer voltages, the typical value of the IBIS [Voltage Range] parameter is multiplied by the scaling factor for the IC corner selected.
- **I/O buffer data**: The data that is used for each process corner is summarized in Process Corner Model Data Usage.
- **Voltage sources on schematics**: The voltage parameter of the element is multiplied by the scaling factor for the IC corner selected.
- **Voltage nets in post-layout**: The voltage set on the net on import of the board is multiplied by the scaling factor for the IC corner selected.
- **Ideal transmission lines (SPICE T elements)**: The Z0 and Tpd parameters are multiplied by the Z0 and Tpd factors for the selected corner.

- **Lossy transmission lines (SPICE W elements)**: The models without explicit slow and fast corner models in the library are scaled using the Z0 and Tpd factors in Corner Conditions. Models that have `_te` (typical), `_fe` (fast) or `_se` (slow) appended to the model name are used for the appropriate etch corner if they exist.
- **SPICE subcircuits**: file and subcircuit names can contain {etch} and {corner}. If present, the current corner is substituted.

Process Corner Model Data Usage

IC Process Corner	Model or Setting	Data Used
FF	IBIS buffer in HSPICE	typ=fast HSPICE option
	IBIS buffer in IsSPICE4	IBIS maximum IV and VT data
	HSPICE buffer	HSPICE FF wrapper
	Temperature	FF Temperature from Corner Conditions
TT	IBIS buffer in HSPICE	typ=typ HSPICE option
	IBIS buffer in IsSPICE4	IBIS typical IV and VT data
	HSPICE buffer	HSPICE TT wrapper
	Temperature	TT Temperature from Corner Conditions
SS	IBIS buffer in HSPICE	typ=slow HSPICE option
	IBIS buffer in IsSPICE4	IBIS minimum IV and VT data
	HSPICE buffer	HSPICE SS wrapper
	Temperature	SS Temperature from Corner Conditions

See Also

More About

- “Simulation Parameters Used in Serial Link Design” on page 2-2
- “Stimulus Patterns in Serial Link Design” on page 2-9
- “Model Jitter and Noise While Designing Serial Link” on page 10-9

Stimulus Patterns in Serial Link Design

You can specify stimulus patterns for the time domain analysis in the **Serial Link Designer** app. If the specified pattern for a designator has fewer bits than the simulation length, the pattern is repeated from the first bit of the pattern. If the pattern is longer than the simulation length the simulation will end at the time specified by **Time Domain Stop** parameter.

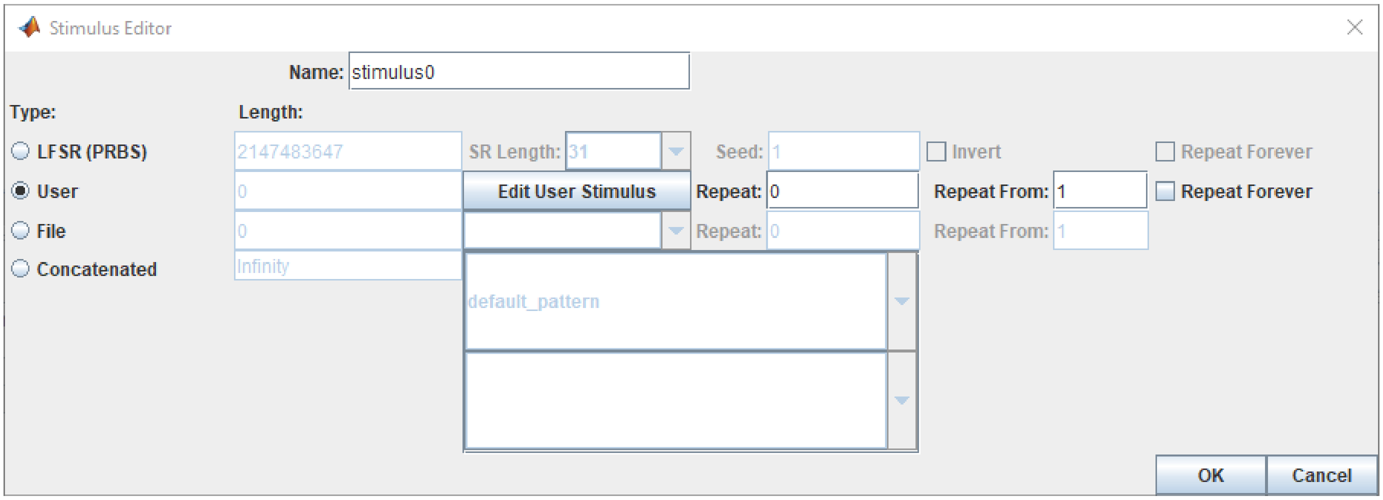
To create and manage stimulus patterns, launch the Stimuli dialog box from **Setup** > **Stimulus** from the app toolbar. The Stimuli dialog box has a table of stimulus patterns with columns for the name, length in bits and description of each stimulus pattern. You can edit, delete, copy, or add new stimulus patterns.

Types of Stimulus Patterns

Stimulus Pattern	Description
LFSR (Linear Feedback Shift Register)	PRBS generated from a shift register with feedback. You need to specify how many bits to generate from the shift register (length), the shift register length (SR length), and the initial value of the shift register (seed)
User	User defined series of ones and zeros.
File	Stimulus defined in a file.
Concatenated	Created from one of more stimulus patterns.

User Stimulus Editor

The User Stimulus Editor is used to create a stimulus.



Stimulus Editor

Name: stimulus0

Type: ○ LFSR (PRBS) ● User ○ File ○ Concatenated

Type	Length:						
○ LFSR (PRBS)	2147483647	SR Length: 31	Seed: 1	□ Invert		□ Repeat Forever
● User	0	Edit User Stimulus	Repeat: 0	Repeat From: 1		□ Repeat Forever
○ File	0		Repeat: 0	Repeat From: 1		
○ Concatenated	Infinity	default_pattern				

Ones and zeroes can be directly typed in the main window. A specified number of ones, zeroes or zero-one sequences can be entered from the fields on the left.

When the sequence has been entered click OK to return to the main Stimulus Editor.

The fields on the Stimuli dialog for User patterns are:

- *Length* — The length of the pattern created in the User Stimulus Editor.
- *Repeat* — The number of times to repeat the pattern. For a pattern of length *n*, a repeat of zero will result in a pattern of length *n*, a repeat of one will result in a pattern of length 2**n* and so on.
- *Repeat From* — The bit position to repeat from. Bit 1 is the first bit in the pattern.

Using Stimulus Patterns

To specify a stimulus pattern (other than the default) on an individual designator basis, edit the designator element properties by double clicking any of the pre-layout designators. This can also be done by clicking on the **Properties** button in the Transfer Net Properties dialog box in the pre- or post-layout. In the resulting Designator Element Properties dialog box, select the desired stimulus pattern from the Stimulus drop-down menu.

See Also

More About

- “Simulation Parameters Used in Serial Link Design” on page 2-2
- “Specify Corner Conditions in Serial Link Design” on page 2-7
- “Model Jitter and Noise While Designing Serial Link” on page 10-9

Pre-Layout Analysis of Serial Link

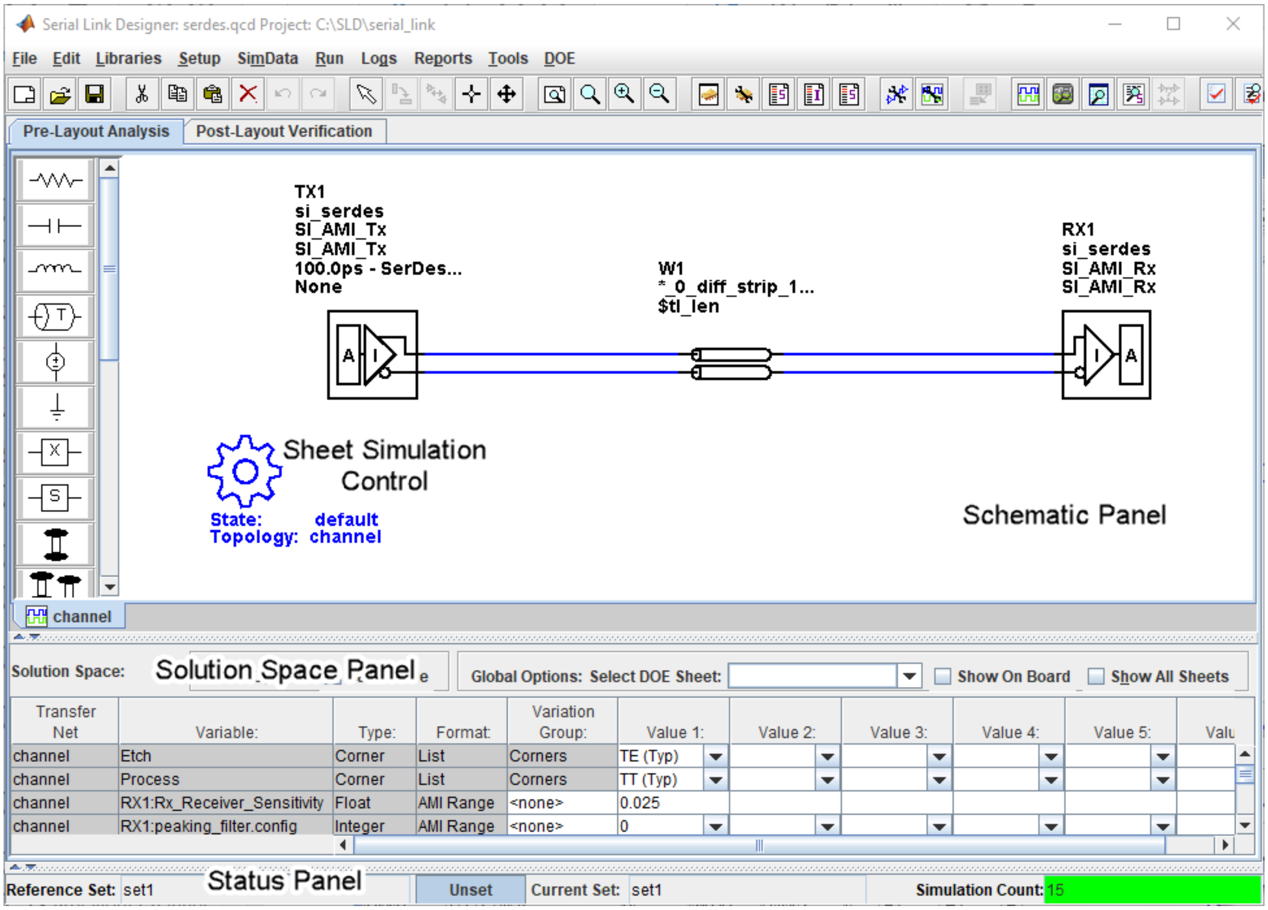
Pre-Layout Analysis of Serial Link

Pre-layout analysis provides you with an integrated signal integrity, timing and crosstalk analysis environment to determine system-level noise and timing margins. The pre-layout analysis environment is used to generate design guidelines for your board layouts, package layouts, connectors and cabling. From the Pre-layout tab, you may perform simple or complex solution space analysis by varying elements, such as: topology, termination, voltage, temperature, process (silicon and etch), models, UIs, corner conditions, populations, and coupling.

A schematic represents an uncoupled net or a coupled net. Uncoupled nets can be thought of as net classes. The **Serial Link Designer** app stores this information as a Transfer Net, which is used as the underlying data structure for all of the analysis. The Transfer Net data can be re-used in post-layout and other projects.

The Pre-Layout Analysis tab consists of three major panels:

- *Schematic Panel* —This is where you graphically create and edit the circuit schematic. You can also define the data from the sheet simulation control settings.
- *Solution Space Panel* — This is where you enter your solution space values for performing parameter sweeps.
- *Status Panel* — This panel displays the simulation counts and schematic set information.



Solution Space:

Global Options: Select DOE Sheet: ☐ Show On Board ☐ Show All Sheets

Transfer Net	Variable:	Type:	Format:	Variation Group:	Value 1:	Value 2:	Value 3:	Value 4:	Value 5:	Valu
channel	Etch	Corner	List	Corners	TE (Typ)					
channel	Process	Corner	List	Corners	TT (Typ)					
channel	RX1:Rx_Receiver_Sensitivity	Float	AMI Range	<none>	0.025					
channel	RX1:peaking_filter.config	Integer	AMI Range	<none>	0					

Reference Set: set1 | Unset | Current Set: set1 | Simulation Count: 15

Double clicking a symbol on a schematic sheet launches an Element Properties dialog box for that symbol type. Each symbol type has a unique set of properties that are set from the Element Properties dialog box. If the properties are parameters that can be swept, that is also controlled from the Element Properties dialog box.

Schematic Elements

Designator — The I/O buffer is represented by a designator in the schematic. A schematic must have at least one designator that can be a driver. The buffers can be single-ended or differential. Buffer symbols has a default I/O buffer model after being placed on the schematic. You can change the buffer model for a designator in three different ways: from the Edit Designator Properties dialog, from the Select IBIS File & Model dialog, and from the default model menu items. IBIS files must be imported into the libraries before they can be used. HSPICE models must be wrapped and put in the libraries before they can be used.

Transmission Line — There are two types of transmission lines: ideal transmission lines and lossy transmission lines. Ideal transmission line models have two parameters: Impedance (Z_0) and delay (T_{pd}). Lossy transmission lines have a frequency dependent RLGC model that is created by a 2-D field solver. Lossy transmission lines can be single-ended or differential..

Via — You can create via models based on a stackup and via physical parameters. Via models can be single-ended or differential. The first time a via symbol is placed on a sheet the default stackup is created. A dialog launches to allow the number of signal layers in the default stackup to be specified.

S-Parameters — You must import the S-Parameter files into the **Serial Link Designer** app before you can use them in schematic sheet. After a symbol has been placed on a schematic, the port map can be edited by right clicking on the symbol and selecting Edit Port Map from the menu.

Passive Subcircuits — You must manually import the SPICE subcircuit models for passive elements in the **Serial Link Designer** app libraries before you can place them on the schematic.

Probe — Voltage probe can be single-ended or differential. When a probe symbol is placed on a schematic it automatically creates a waveform node in the waveform file at the probed location. The waveform at the node can be viewed in the **SI Viewer** app.

Solution Space

The Solution Space panel is used to create parameter sweeps. There are variables that are always part of the solution space. Other variables in the table are created when parameters are set to be swept. The values can be typed into fields, lists or range/steps depending on the variable type.

The solution space panel can be in one of two modes:

- *Permutation mode* — Each row is treated as an independent variable unless they are in the same variation group. The number of simulations represented by the solution space is all of the combinations of all of the variable values.
- *Case mode* — Each column represents a simulation case. The number of simulations represented by the solution space is the number of columns.

Sheet Simulation Control

You can specify the specify the simulation state, unit interval (UI), topology, transfer net type, AC noise type, and the number of aggressors for SSO/coupled mode analysis of each schematic sheet using the sheet simulation control symbol.

See Also

Serial Link Designer

More About

- “Customize Serial Link Project for Pre-Layout Analysis” on page 3-5
- “Results of Pre-Layout Analysis in Serial Link” on page 3-7

Customize Serial Link Project for Pre-Layout Analysis

You can modify the schematic elements to customize your designs in the **Serial Link Designer**. app.

Using I/O buffers

An I/O buffer is represented by a designator. You change the buffer model for a designator in three ways:

- *Edit Designator Part/Pins dialog box*

  Right clicking on the designator and selecting **Edit Designator Part/Pins** opens the Edit Designator Part/Pins dialog box. The **Designator** parameter allows the designator name to be changed. The **Part Name** parameter lists the parts in all libraries. When a specific part is selected in the dropdown menu, the IBIS file name referenced by that part is shown in the **IBIS File** parameter. The IBIS component name for the selected part is shown in the **IBIS Component** parameter. The table on the left shows all of the pins in the IBIS component. To associate a pin or pins with the designator select the pin or pins on the left and click one of the arrow buttons between the two tables. The pins in the table on the left can be filtered using the **Wildcard Filter** parameter. To add a column that shows the name of the transfer net that uses the pins, select **Generate Used Pin Information**.
- *Select IBIS File & Model dialog box*

  Right clicking on the designator and selecting **Select IBIS Model and File** opens the Select IBIS File & Model dialog box. You can select an IBIS file from the table provided, or import your own. You can also select one or more pins from the table of pins in the selected IBIS files.
- *Default model*

  To assign a default model to a designator, right click on the designator and select **Use Default Driver**, **Use Default Receiver** or **Use Default I/O**.

Using Transmission Lines

The app uses two types of transmission lines:

- *Ideal transmission lines*

  Ideal transmission line models have two parameters: Impedance (Z0) and delay (Tpd). These parameters are set from the Element Properties dialog box for ideal transmission lines. Double click on an ideal transmission line symbol on the schematic to launch the Element Properties dialog box. There are columns for Impedance and Delay/Distance and checkboxes to sweep the parameters. Checking a sweep checkbox creates a variable in the solution space for the parameter.

  The model on the schematic is the model for the typical etch corner. If other etch corners are simulated the Z0 and Tpd parameters are scaled according to the corner conditions specified in the Corner Conditions dialog box. See “Specify Corner Conditions in Serial Link Design” on page 2-7 for more information.
- *Lossy transmission lines*
- The lossy transmission line have a frequency dependent RLGC model that is created by a 2-D field solver.

The app has a field solver with a transmission line editor for entering a cross-section. The transmission line editor can be used to create models in the libraries or to edit the model for a symbol.

To associate a model in the library with the transmission line, right click on the symbol and choose **Select T-Line Model**. You can edit the default model by right clicking on the symbol and choosing **Edit T-Line Model**.

Using Vias

You can create via models based on a stackup and via physical parameters. Via models can be single-ended or differential. The first time a via symbol is placed on a sheet the default stackup is created. A dialog launches to allow the number of signal layers in the default stackup to be specified. For more information, see “Via and Stackup Management in Serial Link Project” on page 4-9.

Using S-Parameters

S-Parameter files must be imported into the app before being used on a sheet. As connections are added the body of the S-Parameter symbol will be red if there are any unbalanced connections or unused ports. The app assumes unused ports are terminated by default. S-Parameters can be checked for consistency and correctness using the S-Parameter check feature. For more information, see “Edit Imported S-Parameter Data” on page 5-2.

See Also

Serial Link Designer

Related Examples

- “Edit Imported S-Parameter Data” on page 5-2
- “Analyze Backplane with Line Cards” on page 5-9

More About

- “Pre-Layout Analysis of Serial Link”
- “Results of Pre-Layout Analysis in Serial Link” on page 3-7

Results of Pre-Layout Analysis in Serial Link

The **Serial Link Designer** app produces one or more reports and logs for each simulation and process you run.

The tabs within a report are organized to aid in the process of progressive discovery. The first tab is the log tab, providing a progress summary of the analysis and its errors and warnings. The other tabs contain summaries of the data and successively more detailed information, letting you track down a particular result to a specific simulation file and transition number or time.

Validation Reports

Validation reports indicate the syntax errors in the data. When relevant, the reports provide the corresponding part name, IBIS file and component names, and timing file and model names.

Report	Description
Validation Summary	Number and location of warnings and errors.
Coverage Warnings	Parts or pins in parts that are not referenced in the transfer netlist or timing model.
Transfer Net Summary	Details on each transfer net such as whether the type of the net is data, clock, or strobe, whether the net is differential or single-ended, and the number of nodes. This report also lists information on the clock, noise, and probe points.
Part Summary	Details on each part.
Model Overview	Details on every signal integrity, HSPICE, and IBIS parameter or extension associated with each model in the design. This includes model name, corner and mode information, waveform DRC and timing extensions among other parameters.
Part Pin Summary	Summary of part Transfer Nets and timing pin definitions.
Differential Pin Summary	Lists of the differential pins and components associated with each part.
Timing Delay Summary	Summary of all output delays and setup and hold statements in each timing model.
Model Details	Lists of most of the waveform DRC rules and timing levels used by the product. The report includes the actual parameter used (following the precedence rules) and the value assigned to that parameter.
Transfer Net Errors	Inconsistencies between Transfer Nets, IBIS components and timing models. The part, IBIS and timing files listed are not necessarily where the error occurred, but simply a listing of all files involved in the error checking.

Netlist Generation Report

The Netlist Generation Report contains multiple tabs of data that summarize the netlists that were generated for analysis.

Report	Description
Simulation Decks	Information related to generating simulation decks such as simulation modes, filter sensitivities, clock recovery information, and more.
Parameter Details	Detailed information of all the signal integrity parameters such as jitter, noise, frequency, clock recovery, BER, and more.

Channel Analysis Report

The Channel Analysis Report provides a summary of the simulations that has been run. Tabs for network characterization, statistical analysis, time domain analysis, and the worst case PDA (peak distortion analysis) bit pattern are provided.

Report	Description
Channel Analysis Summary	Status of channel analysis, error and warning messages.
Network	Network characterization results which includes unequilized system responses such as impulse response, step response, pulse response, S-parameters, transfer functions, and more.
Statistical	Statistical analysis results such as statistical eye, BER, bathtub, contour, crosstalk, and more.
Time Domain	Time domain analysis results such as statistical eye, BER, bathtub, contour, deterministic jitter probability function, crosstalk, and more.
PDA	The data pattern which produces the minimum eye opening at the center of the eye.

See Also

Serial Link Designer | Signal Integrity Viewer

More About

- “Pre-Layout Analysis of Serial Link” on page 3-2
- “Customize Serial Link Project for Pre-Layout Analysis” on page 3-5

Post-Layout Verification of Serial Link

Post-Layout Verification of Serial Link

In this section...

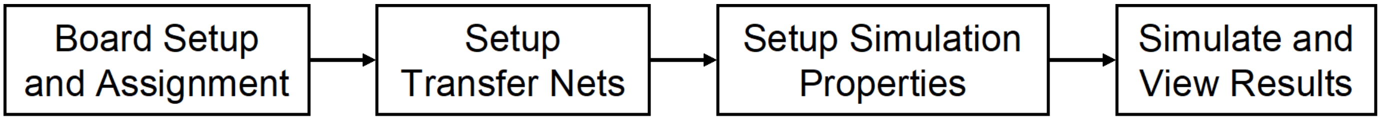
Post-layout verification provides you with an integrated signal integrity and timing environment to verify system-level SI and timing margins for your fully or partially routed PCB design databases.

The post-layout process supports single-board and multi-board analysis, along with connectivity through packages, connectors, and cabling. The post-layout verification environment provides you the ability to extract and analyze PCB databases from any combination of the following CAD (Computer Aided Design) formats:

- MathWorks® Neutral
- IPC-2581
- OBD++
- Cadence® Allegro
- Mentor PADS Layout
- Mentor Board Station
- Mentor Expedition PCB
- Cadence APB
- Intercept Pantheon
- Altium® Designer
- Altium P-CAD
- IBIS EBD

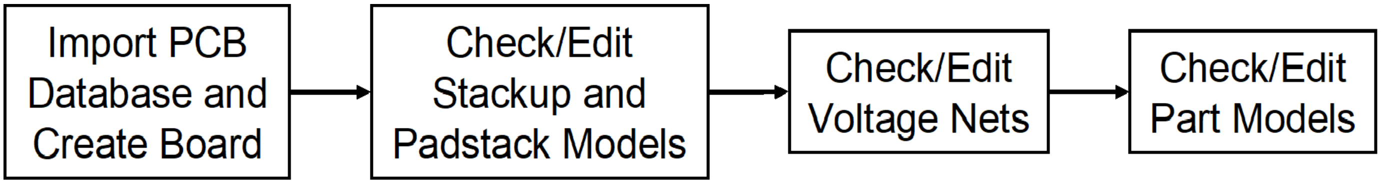
Post-layout analysis takes place in the interface of a serial link design project. If the interface you are working in has pre-layout transfer nets defined, post-layout uses them from the reference schematic set. If there are no transfer nets in the reference schematic set of the interface, the **Serial Link Designer** app creates sheets with system transfer nets (STNETs).

The post-layout verification workflow is the same for each PCB database type. First import the PCB databases, setup the boards, connect the instances if there are multiple boards in the system, run assignment, setup and analyze the nets, set up simulation properties, then simulate and view the results.


Board Setup and Assignment
Setup Transfer Nets
Setup Simulation Properties
Simulate and View Results

Board

The first step in the post-layout verification process is board set-up and assignment. A PCB database you import to the **Serial Link Designer** app is called a board. At the board level, check and edit all stackups, voltage nets, and models. To create variations of a PCB database using different stackups, voltages, or models, create multiple boards with unique names.


Import PCB Database and Create Board
Check/Edit Stackup and Padstack Models
Check/Edit Voltage Nets
Check/Edit Part Models

To perform the setup and assignment functions, access the Post-Layout Setup & Assignment dialog box from the **Setup** > **Setup & Assignment** menu in the app toolstrip.

For each board in the system, specify the type of the PCB database and the files in the database, view or edit the stackup, view or set voltage nets, and manage models by clicking the **Import & Setup Board** button in the Post-Layout Setup & Assignment dialog box. The Import & Setup Board dialog box has four tabs:

- **Import**

  Use the **Import Board** tab to import a PCB database and create a board. Select the PCB database type from the **PCB Database Type** selector list. By default, the **Serial Link Designer** app creates an instance for each board and copies the PCB database files into the current project. If you do not copy the PCB database into the project, you cannot re-import the database files.
- **Stackup**

  The **Stackup** tab shows the stackup from the PCB database and allows control of padstack models. The Stackup Editor on the left side of the tab shows the stackup as it is read in from the PCB database. If necessary, you can override the auto-generated trace models using the editor. The right side of the tab has controls for the auto-generated padstack backdrill options, differential extraction, and DRC control. For more information, see “Stackup and Extraction Control in Serial Link Project” on page 4-6.
- **Voltages**

  The **Voltages** tab shows the CAD nets in the PCB database for the board and allows you to specify the voltage for specific voltage nets. Non-voltage nets have an NA value in the voltage column.

  Note The **Voltages** tab does not control the voltages in the IBIS or SPICE models for TX/RX designators. This tab is mainly used to correctly define the on-board terminations that require connection to a specific voltage.
- **Parts**

  Use the **Parts** tab to match models to parts in the PCB database.

Instance

An instance is an internal copy of a board that you can connect to other instances and analyze. Every board that is used in the design has at least one instance. If you use the same board more than once, you must define a separate instance for each use. For example, a system consisting of a motherboard with two DIMM slots that has the same type of DIMM plugged into each slot will have one instance of the motherboard and two instances of the DIMM.

Connection

A Connection is a pin-to-pin path from the pins of a reference designator on one instance to the pins of a reference designator on a second instance. In a multi-board system, connections between instances are specified in the Connections pane of the Post-Layout Setup & Assignment dialog box. To add a connection, click the **Add Connection** button.

Assignment

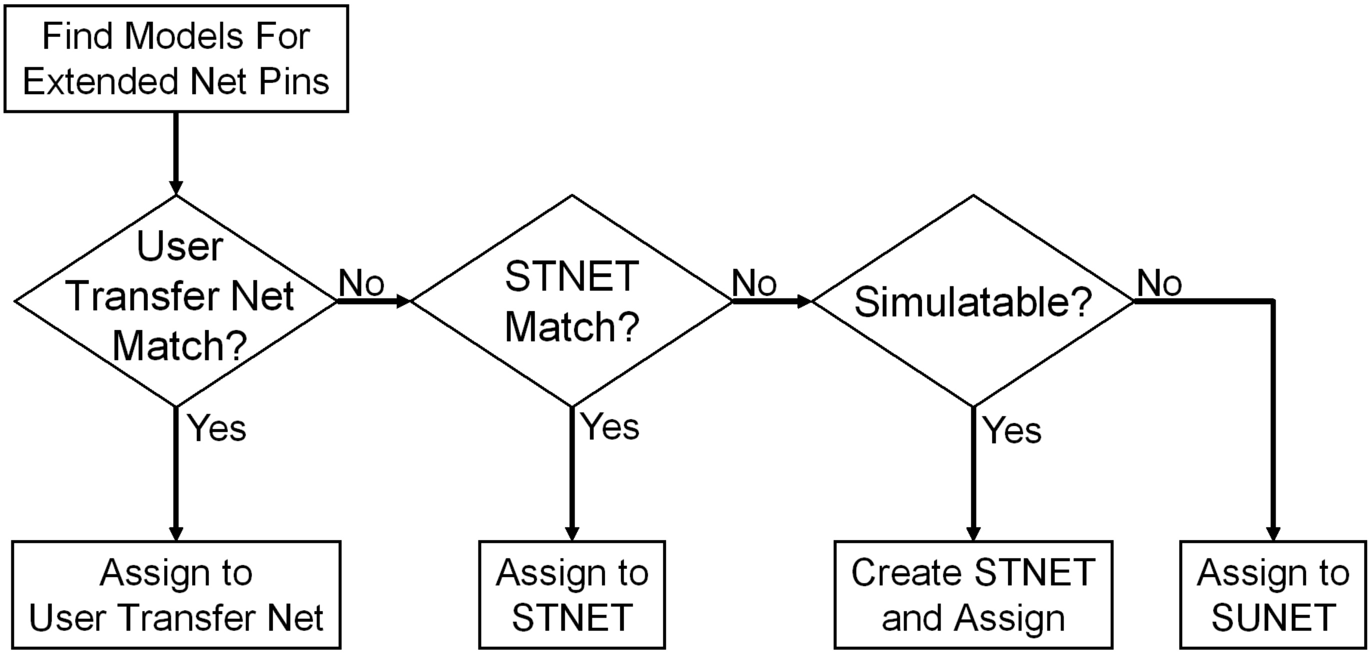
The Assignment process is an automated process for associating nets in the PCB database with transfer nets. This simplifies the setup of the essential net properties in the typical scenarios that you will face:

- **Interface without Transfer Nets**

  If you did not do a pre-layout analysis in an interface, you can create and edit transfer nets from the post-layout interface. When you set the properties of a transfer net, you set the properties of all nets assigned to that transfer net. For example, when you change the properties of a transfer net, the app automatically assigns those properties to all nets in a data bus.
- **Interface with Transfer Nets from Pre-Layout Analysis**

  If you completed pre-layout analysis in an interface, the app automatically assigns the nets you created in post-layout analysis to the transfer nets you created in the pre-layout analysis.
- **Design Kits**

  A design kit is an interface with models and preconfigured transfer nets. The app automatically assigns the nets you created in post-layout analysis to these transfer nets.



In all cases, the transfer nets and the assignment process ensure that all nets in an interface are set up and ready to simulate in a fraction of the time needed to set up each net in the interface individually.

Population

Populations allow you to setup multiple configurations of a system for simulation in one project. The app handles populations through the naming of instances.

For example, if a one-slot motherboard can accept one of three DIMMs (dual in-line memory modules), it can be set up by creating three instances of the motherboard and one instance of each DIMM. In this case, three populations can be defined: the motherboard with RCA installed, the motherboard with RCB installed and the motherboard with RCC installed

Simulation

Before you run a simulation, you must select the nets for the post-layout verification. Select the nets and add them to the list of nets to simulate. You also need to set up the stimulus patterns from simulation properties.

Topology

Extended nets that can be simulated (assigned to an STNET or user transfer nets) can have topologies created from the extracted PCB data. View these topologies from the **Pre-Layout Analysis** tab. The topologies are useful for understanding how an actual network is routed and to resolve waveform quality or timing issues identified by using post-layout verification. Once the extracted post-layout networks are in the pre-layout analysis environment, you can perform quick “what-if” analyses to identify an appropriate solution.

See Also

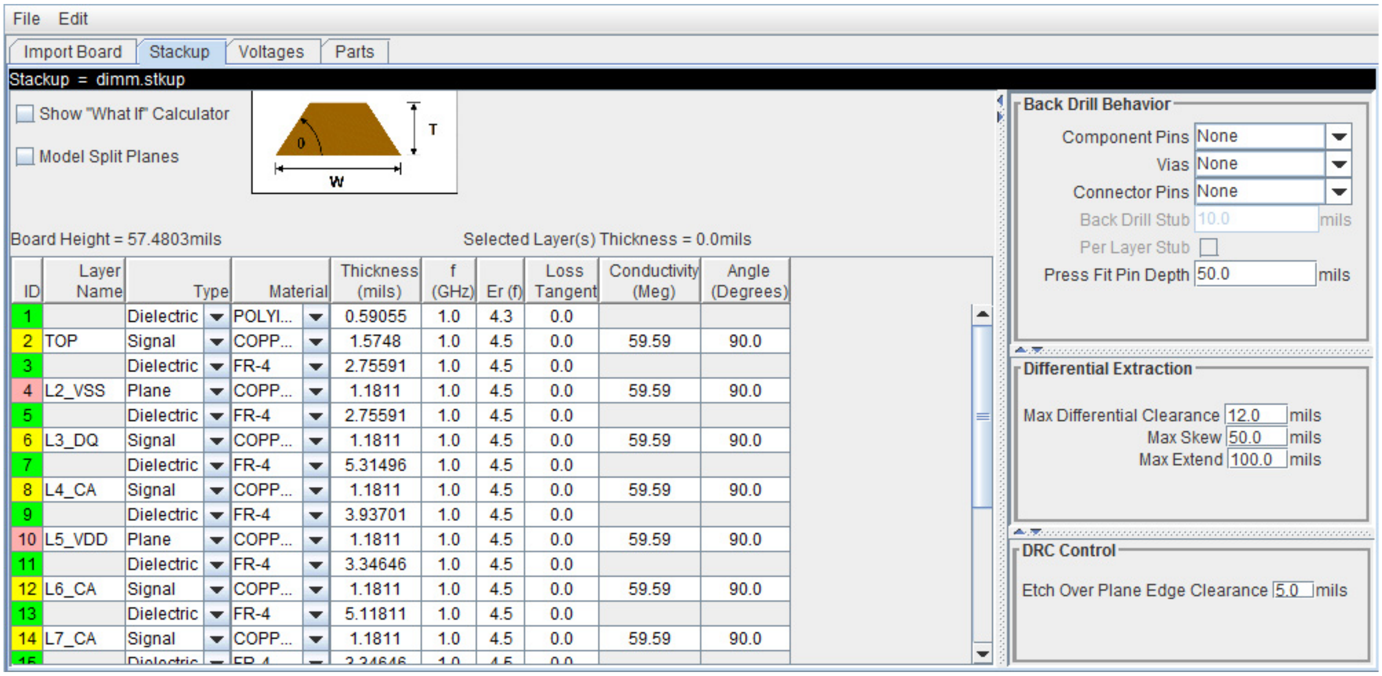
More About

- “Stackup and Extraction Control in Serial Link Project” on page 4-6
- “Via and Stackup Management in Serial Link Project” on page 4-9

Stackup and Extraction Control in Serial Link Project

The **Stackup** tab of the **Import & Setup Board** dialog shows the stackup from the PCB Database and allows for extraction control for padstack, differential traces, and DRC.

The tab is divided into two areas: **Stackup Editor** and **Extraction Control**. The **Stackup Editor** on the left side of the tab shows the stackup that was read from the PCB Database and allows the override of the auto-generated stackup thicknesses, material properties, and trapezoidal angle as well as the ability to do "What If" exploration and select whether to model discontinuities associated with etches crossing split planes. The right side of the tab controls the padstack backdrill options, differential extraction, and DRC control.



Stackup = dimm.stkup

□ Show "What If" Calculator

□ Model Split Planes

Board Height = 57.4803mils

Selected Layer(s) Thickness = 0.0mils

ID	Layer Name	Type	Material	Thickness (mils)	f (GHz)	Er (f)	Loss Tangent	Conductivity (Meg)	Angle (Degrees)
1		Dielectric	POLYI...	0.59055	1.0	4.3	0.0		
2	TOP	Signal	COPP...	1.5748	1.0	4.5	0.0	59.59	90.0
3		Dielectric	FR-4	2.75591	1.0	4.5	0.0		
4	L2_VSS	Plane	COPP...	1.1811	1.0	4.5	0.0	59.59	90.0
5		Dielectric	FR-4	2.75591	1.0	4.5	0.0		
6	L3_DQ	Signal	COPP...	1.1811	1.0	4.5	0.0	59.59	90.0
7		Dielectric	FR-4	5.31496	1.0	4.5	0.0		
8	L4_CA	Signal	COPP...	1.1811	1.0	4.5	0.0	59.59	90.0
9		Dielectric	FR-4	3.93701	1.0	4.5	0.0		
10	L5_VDD	Plane	COPP...	1.1811	1.0	4.5	0.0	59.59	90.0
11		Dielectric	FR-4	3.34646	1.0	4.5	0.0		
12	L6_CA	Signal	COPP...	1.1811	1.0	4.5	0.0	59.59	90.0
13		Dielectric	FR-4	5.11811	1.0	4.5	0.0		
14	L7_CA	Signal	COPP...	1.1811	1.0	4.5	0.0	59.59	90.0
15		Dielectric	FR-4	3.34646	1.0	4.5	0.0		

Back Drill Behavior

Component Pins: None

Vias: None

Connector Pins: None

Back Drill Stub: 10.0 mils

Per Layer Stub □

Press Fit Pin Depth: 50.0 mils

Differential Extraction

Max Differential Clearance: 12.0 mils

Max Skew: 50.0 mils

Max Extend: 100.0 mils

DRC Control

Etch Over Plane Edge Clearance: 5.0 mils

Stackup Editor

The **Stackup Editor** displays one row for each signal, plane, and dielectric layer in the stackup. Parameter values can be changed if desired by typing new values into the table cells. The stackup data plus the trace width data are used by the field solver to create lossy transmission line models for post-layout nets.

Each layer must be defined as either `Dielectric`, `Mixed`, `Plane`, or `Signal` in the stackup column called **Type**. Signal layers can be either type `Mixed` or `Signal`. The `Mixed` designation is provided primarily for boards and packages where sections of the signal layer may contain small planes for impedance control. In most cases the `Signal` designation would be sufficient, but it is important to carefully review the board layout and identify cases where `Mixed` may be required.

Checking **Model Split Planes** enables modeling of discontinuities associated with etches crossing over splits in planes. The change in trace cross-section results in an impedance change in the model.

Checking **Table-Driven Loss Model** allows a table-driven loss model to be used. When checked, the list contains the names of imported loss models and the item **Assign Per Layer Loss Model**. When a loss model is selected, it is used for all layers in the stackup. If **Assign Per Layer Loss Model** is selected, from the list the **Table-Driven Loss Model** column appears in the stackup with a list to choose the loss model for each signal layer.

You may use the **Stackup Editor** as a calculator to compute trace impedance based upon the width and separation. To use the calculator:

1. Check the **Show "What If" Calculator** check box to display the calculator columns
2. Enter one or more values in the appropriate cells followed by the tab key
3. Click **Calculate**

This uses the stackup data with the **Desired Width** and **Desired Separation** values to calculate the single-ended and differential impedance for that layer.

Extraction Control

The **Extraction Control** section of the tab controls the backdrill behavior, differential extraction, and DRC control.

Backdrilling uses Must Not Cut Layers. Must Not Cut Layers are layers that define a valid backdrill depth. In the stackup there are columns for Must Not Cut Layers from the top and bottom. The backdrill goes from the top or bottom up to but not through the last Must Not Cut Layer that is encountered before a trace connection to a via or pin. If no Must Not Cut layer is encountered before the trace connection to the via or pin, then the via or pin is modeled as not backdrilled.

Backdrill Behavior Choice	Description
`None`	No backdrilling. The complete via or pin is extracted, and a model generated based on the PCB data for start and end layers.
`Top`	The via or pin is modeled as if it were drilled from the top of the board. The via or pin ends at the lowest layer with **Backdrill Top Must Not Cut Layer** checked in the stackup that is above the highest layer with a trace connected to the via or pin. A stub equal to the **Back Drill Stub** parameter is left. If there is no layer with **Backdrill Top Must Not Cut Layer** checked that is above the highest trace connection to the via or pin, the via or pin is not backdrilled.
`Bottom`	The via or pin is modeled as if it were drilled from the bottom of the board. The via or pin ends at the highest layer with **Backdrill Bottom Must Not Cut Layer** checked in the stackup that is below the lowest layer with a trace connected to the via or pin. A stub equal to the **Back Drill Stub** parameter is left. If there is no layer with **Backdrill Bottom Must Not Cut Layer** checked that is below the lowest trace connection to the via or pin, the via or pin is not backdrilled.

Backdrill Behavior Choice	Description
`Both`	Both top and bottom are modeled as described above.
`Longest Stub`	Drills from the side that remove the longest stub based on the Must Not Cut layers defined in the stackup.

In the **Differential Extraction section** of the **Padstack Editor**, you can define the parameters that control the extraction of the differential nets.

Parameter	Description
Max Differential Clearance	The maximum edge-to-edge clearance two traces can have and still be extracted as a differential transmission line model. If the clearance is larger than this parameter, the traces are extracted as two single-ended transmission line models.
Max Skew	The maximum length difference between the two traces in a single differential trace w-line model. It is recommended that this be set no larger than 1/10 of the wavelength of the maximum frequency of interest.
Max Extend	The maximum total length of single-ended trace that can be combined with a differential trace in a w-line model.

The DRC control defines the minimum distance from a trace to a plane edge when the trace crossing DRC is run using the **Etch Over Plane Edge Clearance** parameter.

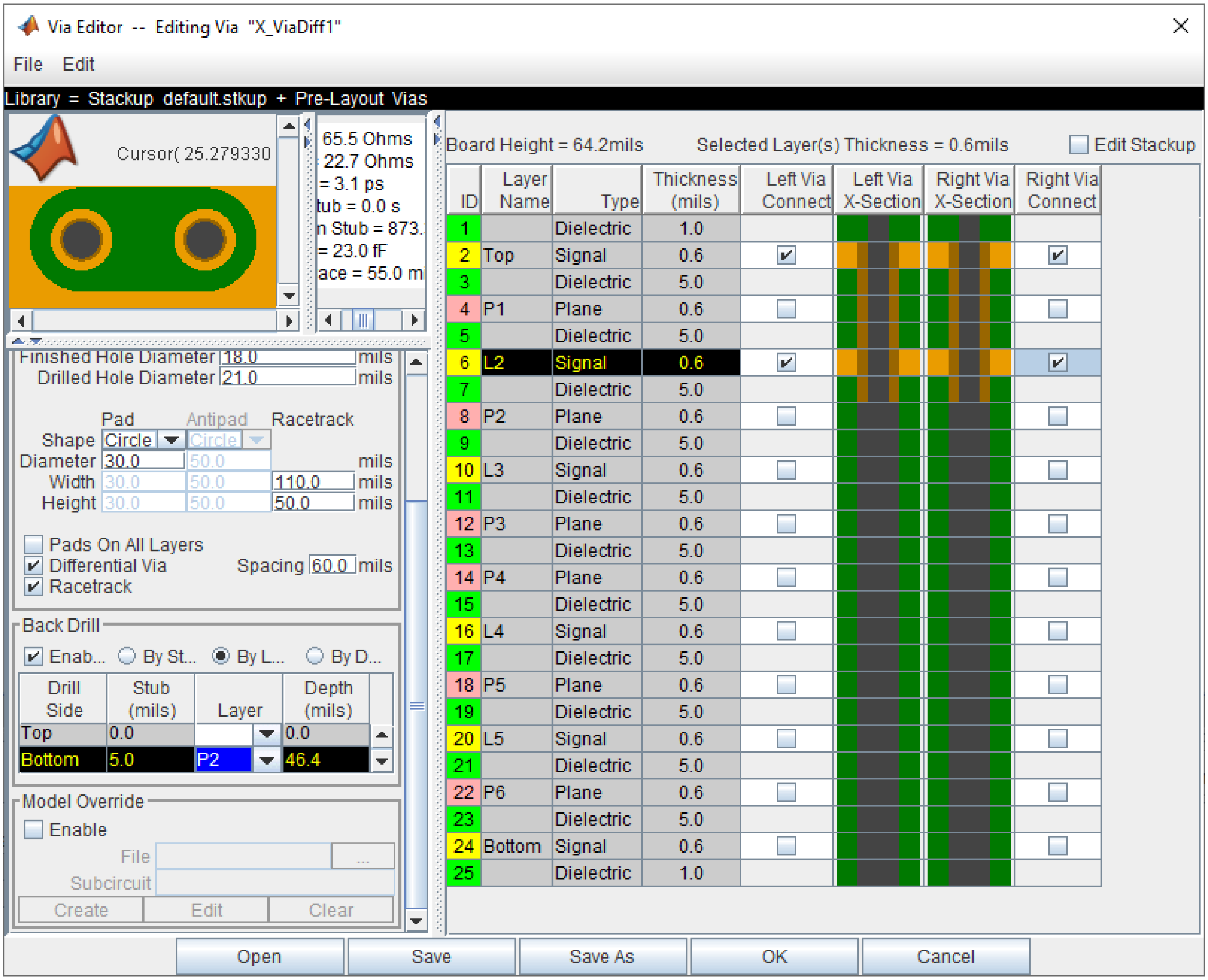
See Also

More About

- “Post-Layout Verification of Serial Link” on page 4-2
- “Via and Stackup Management in Serial Link Project” on page 4-9

Via and Stackup Management in Serial Link Project

The vias are associated with a stackup in the library where they are stored. There can be multiple stackup and via libraries in a project. The first time you edit a via in pre-layout you are prompted for the number of layers to use for the default pre-layout stackup. In post-layout the stackup and vias are from the PCB database by default. Use the **Via Editor** dialog box by right clicking on the vias in the Pre-Layout Analysis tab to manage them. The elements in the via editor can be divided into three groups: common via elements, pre-layout specific via elements, and post-layout specific via elements.



Via Editor -- Editing Via "X_ViaDiff1"

Library = Stackup default.stkup + Pre-Layout Vias

Cursor(25.279330

65.5 Ohms
22.7 Ohms
= 3.1 ps
tub = 0.0 s
n Stub = 873.
= 23.0 fF
ace = 55.0 m

Finished Hole Diameter 18.0 mils
Drilled Hole Diameter 21.0 mils

	Pad	Antipad	Racetrack	
Shape	Circle	Circle		
Diameter	30.0	50.0		mils
Width	30.0	50.0	110.0	mils
Height	30.0	50.0	50.0	mils

☐ Pads On All Layers
☑ Differential Via Spacing 60.0 mils
☑ Racetrack

Back Drill

☑ Enab... ○ By St... ● By L... ○ By D...

Drill Side	Stub (mils)	Layer	Depth (mils)
Top	0.0		0.0
Bottom	5.0	P2	46.4

Model Override

☐ Enable

File
Subcircuit

Board Height = 64.2mils Selected Layer(s) Thickness = 0.6mils ☐ Edit Stackup

ID	Layer Name	Type	Thickness (mils)	Left Via Connect	Left Via X-Section	Right Via X-Section	Right Via Connect
1		Dielectric	1.0				
2	Top	Signal	0.6	☑			☑
3		Dielectric	5.0				
4	P1	Plane	0.6	☐			☐
5		Dielectric	5.0				
6	L2	Signal	0.6	☑			☑
7		Dielectric	5.0				
8	P2	Plane	0.6	☐			☐
9		Dielectric	5.0				
10	L3	Signal	0.6	☐			☐
11		Dielectric	5.0				
12	P3	Plane	0.6	☐			☐
13		Dielectric	5.0				
14	P4	Plane	0.6	☐			☐
15		Dielectric	5.0				
16	L4	Signal	0.6	☐			☐
17		Dielectric	5.0				
18	P5	Plane	0.6	☐			☐
19		Dielectric	5.0				
20	L5	Signal	0.6	☐			☐
21		Dielectric	5.0				
22	P6	Plane	0.6	☐			☐
23		Dielectric	5.0				
24	Bottom	Signal	0.6	☐			☐
25		Dielectric	1.0				

Via Elements

Via Element	Description
Top view and electrical characteristics	The top view shows the via as it would appear when viewed from the top of the board. The electrical characteristics show the impedance, delay backdrill, and other characteristics. The reported delay is for the barrel of the via.
Via geometry	You can edit the geometry of the via by defining the start and end layers, hole diameters, and shape and dimensions of the pad and antipad. You can also select if a via model is single-ended or differential.
Via backdrill	You can select the depth of via backdrill by stub, layer, or depth.
Override via model	You can override a via model by using your custom subcircuit saved in one of the SPICE libraries.
Connect via layers	The **Left Via Connect** column is used to select the layer connections that will appear on the left side of the via symbol. The **Right Via Connect** column is used to select the layer connections that will appear on the right side of the via symbol. A layer is connected when the checkbox for that layer is checked. The **Via X-Section** columns show a representation of the via cross section.
Modify stackup	To modify the stackup, check the Edit Stackup checkbox.

There are several important definitions for vias and pins:

- A via under a BGA is a via, not a pin.
- A through hole connector padstack is a pin, not a via.
- A connector means a multi-board mated connector (connects two board Instances).

Editing Via for Pre-Layout Simulations

To edit vias for pre-layout simulation, open the Via Editor dialog box by selecting **Tools > Via Editor** or by right-clicking on a via schematic symbol and selecting **Edit Differential Via Model** or **Edit Single Ended Via Model**. You need to enter the number of conducting layers for the default stackup the first time you open the **Via Editor** dialog box.

The Via Editor works in a selected library. Vias can be edited, added or deleted from a library. In pre-layout, the Via Editor creates a default library that contains a default via model and a default stackup. The Library operations can be selected from the File menu.

Editing Via for Post-Layout Simulations

The Padstack/Trace Manager is used to view and manage overrides to padstacks and traces in post-layout as well as manage backdrilling of pins and vias by net, RefDes or Part. You can edit the geometry of a single via, or multiple vias at one time. You can also assign overrides by individual instance of a padstack or by occurrence of each padstack's library name in the database.

- **Back Drill Setup Tab**

  The **Back Drill Setup** tab allows backdrill information to be viewed and changed by net, by padstack, by RefDes, or by Part by selecting from the **View Mode** list. In each case the backdrill can be turned on or off. The view modes are:

  - *Back Drill by Net* — One row per Extended Net per Board.
  - *Back Drill by Padstack* — One row per Padstack.
  - *Back Drill by RefDes* — One row per Reference Designator.
  - *Back Drill by Part* — One row per Part Number.

  The **Back Drill Setup** tab is only enabled if backdrilling is enabled on one or more boards on the **Stackup** tab of the **Setup Board** dialog.
- **Via/Pin Editor Tab**

  Padstack models are created automatically from the PCB data for vias, surface mount pads and through-hole pins using the internal padstack solver.

Padstack Definitions

Padstack Elements	Definitions
Padstack	The geometry information from the PCB database. Contains the start and end layer of the padstack, barrel dimensions, etc. A Padstack does not contain the layers connected or XY coordinates.
Padstack Configuration	A Padstack plus layer connections. A Padstack Configuration does not contain XY coordinates.
Padstack Configuration Instance	A Padstack Configuration at a specific XY coordinate on a board. A specific via has geometry, connectivity and a location on a PCB. A specific pin has geometry, connectivity, a location, a reference designator and a pin number

A Padstack can be used for multiple Padstack Configurations. A Padstack Configuration can be used for multiple Padstack Configuration Instances.

Padstack Editor View Modes

The views are selected from the list on the Via/Pin tab. The view modes correspond to the definitions above. In each view mode there is one row for each item of the selected type:

- *Padstack (Geometry)* — One row per Padstack. This rolls up all Padstack Configurations and Padstack Configuration Instances that use a Padstack.
- *Padstack Configuration (Connectivity)* — One row per Padstack Configuration. This rolls up all Padstack Configuration Instances that use a Padstack Configuration.
- *Padstack Configuration Instance* — One row per Padstack Configuration Instances.

Padstack Editor Edit Modes

The padstack editor has two modes:

- *Padstack* — All changes made in the editable columns apply to the padstack. This means all Padstack Configuration Instances that use the same Padstack as the row being edited will change. For example, if the View Mode is Padstack Configuration Instance and the Edit Mode is Padstack, a change to one row is applied to all rows that have the same Padstack.
- *Instance* — All changes apply to the Padstack Configuration Instance only. For example, if the View Mode is Padstack Configuration Instance and the Edit Mode is Instance, a change to one row is only applied to that row.

Common Operations

Editing geometry of a single via	To edit the geometry of a single via (one via at one XY coordinate), use the Padstack Configuration Instance View Mode and the Instance Edit Mode. Any changes to the geometry is applied to the specific via edited when in this mode.
Editing using the Via Editor	Right-click on a row and choose Visual Via Editor from the menu.
Changing the Padstack	To change the Padstack that a Padstack Configuration Instance is based on, use the Padstack Configuration Instance View Mode and in the Base Padstack column choose a different Padstack. The list of padstacks are the padstacks that share the same start and end layer with the original padstack for this Padstack Configuration Instance.
Editing a Padstack	To edit a Padstack, use the View Mode Padstack. The behavior depends on Edit Mode: • *Padstack* — Changes are made to the Padstack being edited and is applied to all Padstack Configurations and Padstack Configuration Instances that use the Padstack. • *Instance* — A new Padstack is created as a copy of the Padstack being edited, and the changes you make is applied to this new Padstack

Overriding a via model	Via models are typically done with connectivity to specific layers. Therefore, the Padstack Configuration View Mode or the Padstack Configuration Instance View Mode are used to override a via model. In both modes the Model Override column is part of the table. To override a model right-click and select one of: • *Browse* — Browse to an existing model in the libraries. This could be the .smod file for an S-Parameter via model that was imported. • *Create* — Create a subcircuit with the default via model. This subcircuit can be modified. If the View Mode is Padstack Configuration the model is applied to every Padstack Configuration Instance that uses that Padstack Configuration. If the View Mode is Padstack Configuration Instance the model is applied to the single Padstack Configuration Instance that you edited. **Note** The edit mode must be Padstack.

Trace Overrides Tab

The Trace Overrides tab of the Padstack/Trace Manager is used for trace model overrides. The lossy transmission line models for traces created by the field solver from the stackup and trace width can be overridden with user-provided models. The Trace Overrides tab shows the trace widths on each layer of each board.

The models used for overrides are assumed to be RLGC models with one model per file, and the base name of the file must be the same as the model name.

For single-ended traces there is one row for each trace width found on each layer. Select one or more rows and click the Select Model button to browse to a transmission line model in the library.

For differential traces, there is one row for each trace width on each layer, and columns for differential separation and coupling layer. The coupling layer is a list containing the same layer and any adjacent signal or mixed layers. Select an adjacent layer for broadside coupled differential traces. When a separation is added a new row is created for that layer and trace width. This allows models for multiple separations to be specified for each width on each layer.

The tolerance for overrides is 0.1 mm in width. In other words, if an override is specified for a trace of width 4.0 mm on a layer, the override is applied to all traces with widths from 3.9 mm to 4.1 mm on that layer.

Example One-Conductor Model

For file name `sl_55ohm.mod`:

```
model sl_55ohm W ModelType=RLGC N=1
+ Lo = +3.60600E-07
+ Co = +1.20300E-10
+ Ro = +6.07368E+00
+ Rs = +1.48880E-03
+ Gd = +1.89000E-11
```

Example Differential Model

For file name `sl_55ohm_diff.mod`:

```
.model sl_55ohm_diff W ModelType=RLGC N=2
+ Lo = +3.58800E-07 +4.84700E-08 +3.58800E-07
+ Co = +1.23200E-10 -1.66400E-11 +1.23200E-10
+ Ro = +6.07368E+00 +0.00000E+00 +6.07368E+00
+ Rs = +1.50556E-03 +1.12767E-04 +1.50556E-03
+ Gd = +1.93500E-11 -2.61400E-12 +1.93500E-11
```

See Also

More About

- “Post-Layout Verification of Serial Link” on page 4-2
- “Stackup and Extraction Control in Serial Link Project” on page 4-6

Serial Link Featured Examples

Edit Imported S-Parameter Data

This example shows how you can check and modify the S-Parameter data using the **Serial Link Designer** app. An example S-Parameter file with bad data is imported and analyzed using the tools available in the S-Parameter Checklist dialog box in the **Serial Link Designer** app.

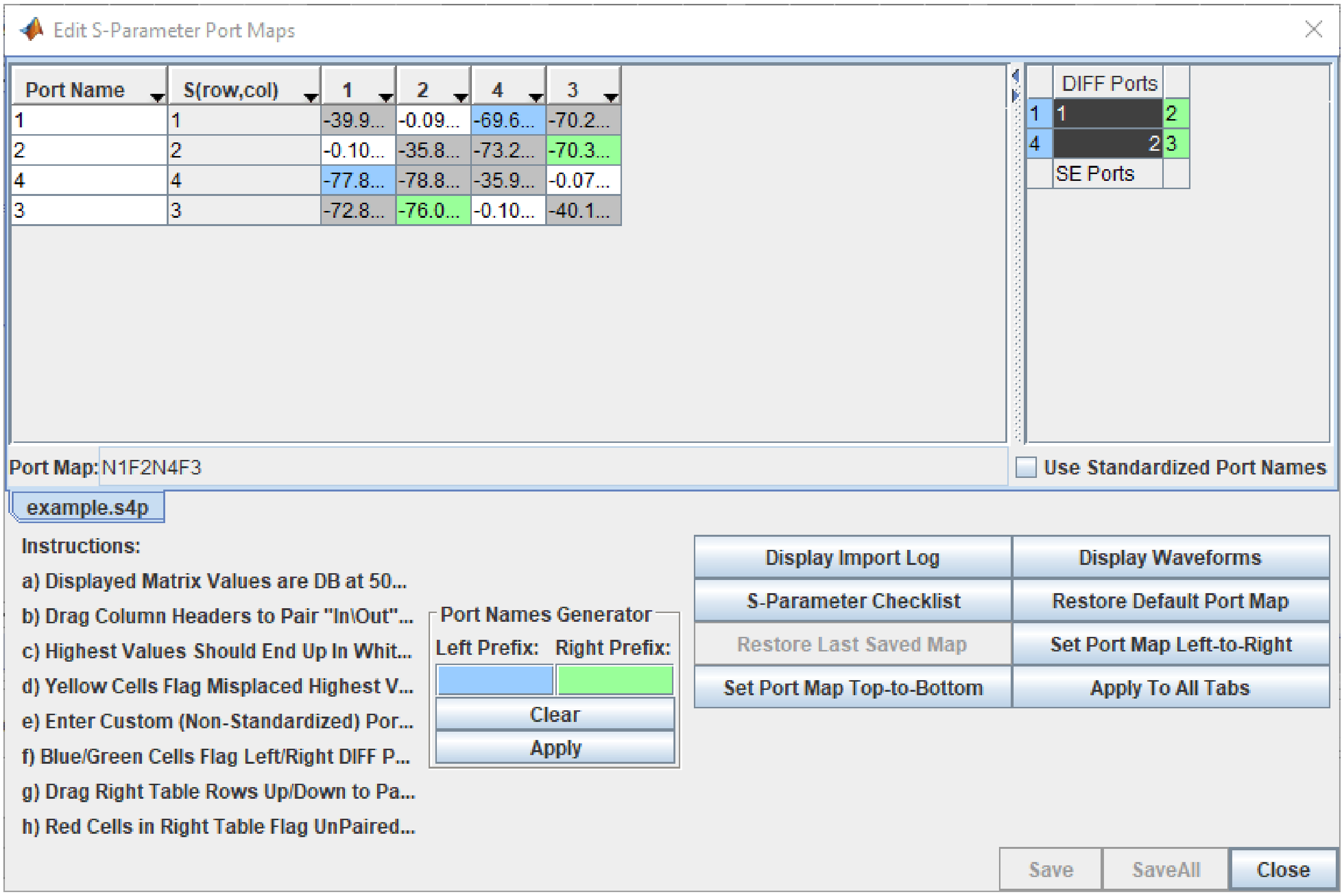
Create New Project

Open the **Serial Link Designer** app.

```
serialLinkDesigner
```

Create a new project by selecting **File** > **New Project**. In the newly opened dialog box, name the project as `edit_sparameter`, the interface as `serdes`, and the schematic sheet as `channel`. The **Pre-Layout Analysis** tab shows the blank schematic sheet.

A custom S-Parameter data file (`example.s4p`) is attached as supporting file to this example. This file represents PCB traces on a board. Download the Touchstone® (.s4p) files. To import the S-Parameter data, select **Libraries** > **Import S-Parameter**. Browse to the location where you saved the downloaded file and import the file. This launches the Edit S-Parameter Port Maps dialog box with two warnings. Ignore the warning that pops up for now, the example explores them in detail.



Port Name	S(row,col)	1	2	4	3
1	1	-39.9...	-0.09...	-69.6...	-70.2...
2	2	-0.10...	-35.8...	-73.2...	-70.3...
4	4	-77.8...	-78.8...	-35.9...	-0.07...
3	3	-72.8...	-76.0...	-0.10...	-40.1...

	DIFF Ports	
1	1	2
4	2	3
	SE Ports	

Port Map: N1F2N4F3

☐ Use Standardized Port Names

Instructions:

a) Displayed Matrix Values are DB at 50...

b) Drag Column Headers to Pair "In\Out"...

c) Highest Values Should End Up In Whit...

d) Yellow Cells Flag Misplaced Highest V...

e) Enter Custom (Non-Standardized) Por...

f) Blue/Green Cells Flag Left/Right DIFF P...

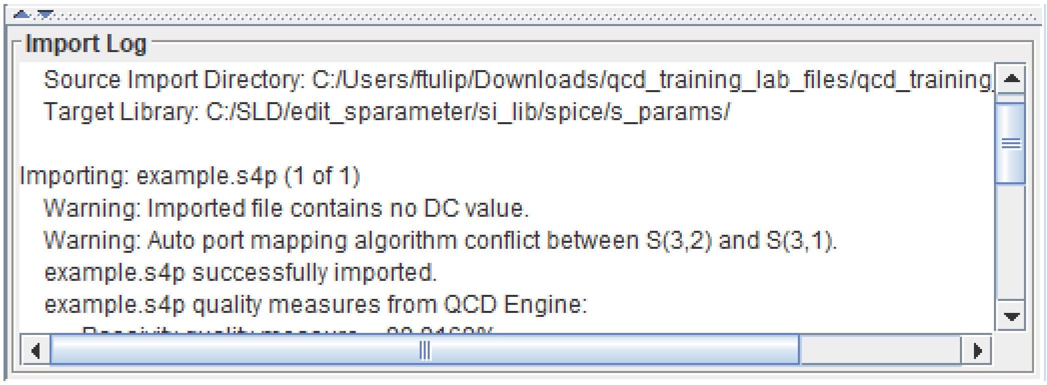
g) Drag Right Table Rows Up/Down to Pa...

h) Red Cells in Right Table Flag UnPaired...

Use S-Parameter Checklist to Analyze S-Parameter Data

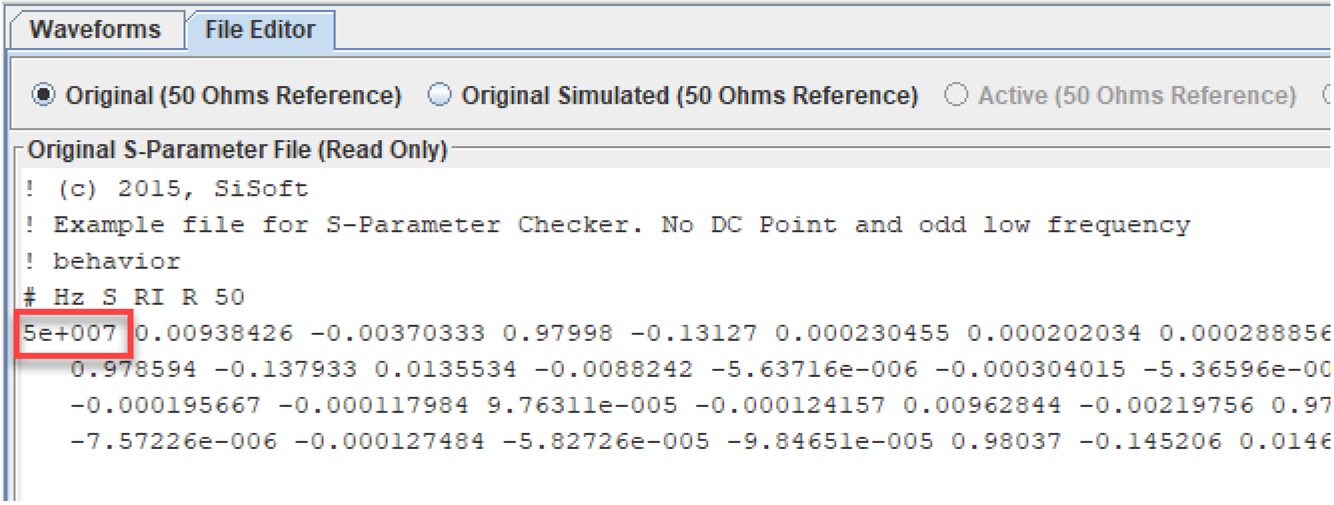
Click on the **S-Parameter Checklist** button on the Edit S-Parameter Port Maps dialog box to launch the S-Parameter Checklist dialog box. The S-Parameter Checklist dialog box has two panels with synchronized views. On the right are the frequency/time domain plots and a text editor. The left panel contains data and explanations of the plots.

First, look at the import log information in the left panel. The warnings indicate that there is no DC value in the S-Parameter file and that the port mapping algorithm found inconsistent data when trying to determine which ports go on the left of the symbol and which ports go on the right of the symbol.



```
   Source Import Directory: C:/Users/ftulip/Downloads/qcd_training_lab_files/qcd_training
   Target Library: C:/SLD/edit_sparameter/si_lib/spice/s_params/

Importing: example.s4p (1 of 1)
   Warning: Imported file contains no DC value.
   Warning: Auto port mapping algorithm conflict between S(3,2) and S(3,1).
   example.s4p successfully imported.
   example.s4p quality measures from QCD Engine:
```

If you look at the **File Editor** tab in the right panel you can see that the lowest frequency point is 50MHz in the S-Parameter file. This is why the import log reported the no DC value warning.



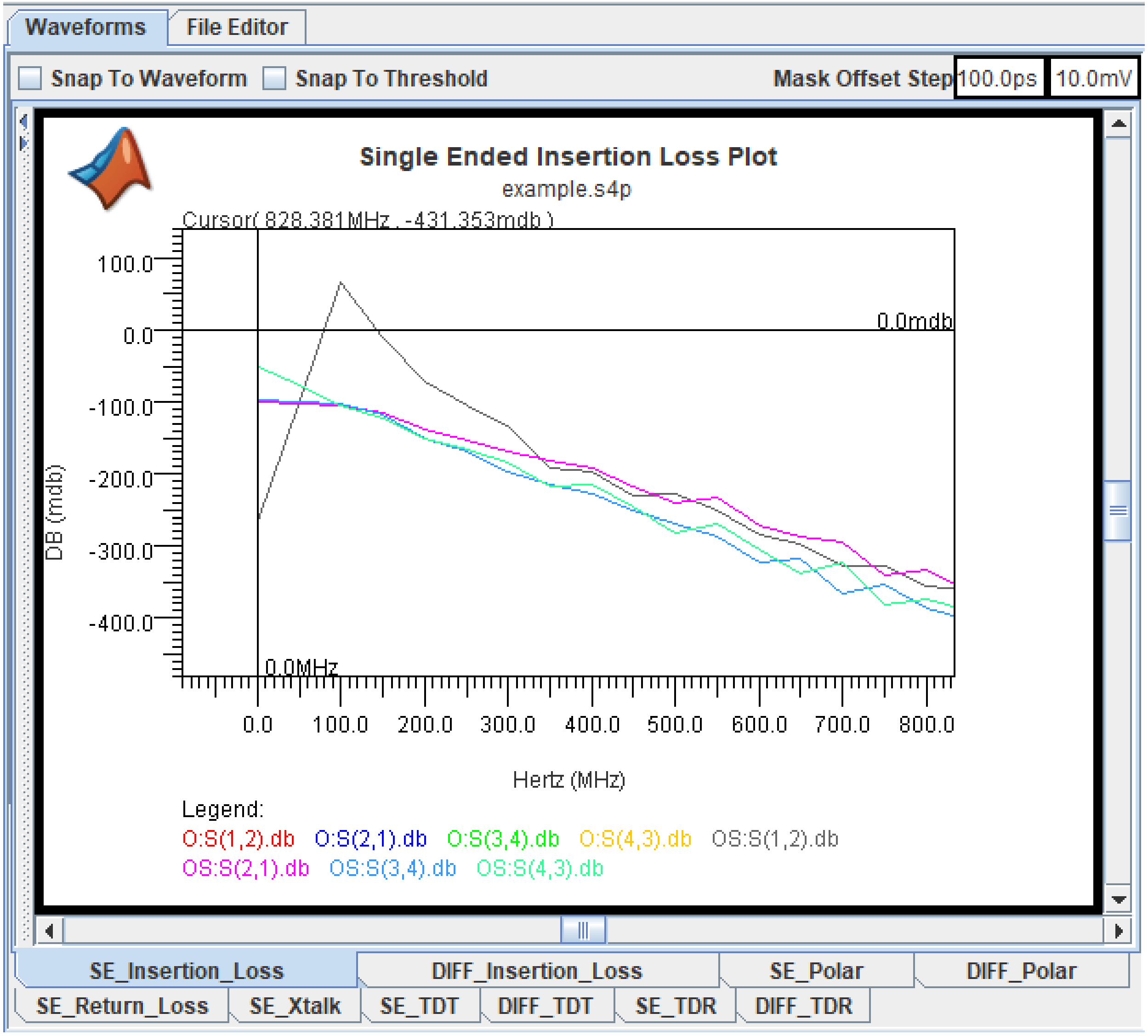
Waveforms
File Editor
◉ Original (50 Ohms Reference)
○ Original Simulated (50 Ohms Reference)
○ Active (50 Ohms Reference)
Original S-Parameter File (Read Only)

```
! (c) 2015, SiSoft
! Example file for S-Parameter Checker. No DC Point and odd low frequency
! behavior
# Hz S RI R 50
5e+007 0.00938426 -0.00370333 0.97998 -0.13127 0.000230455 0.000202034 0.00028885€
     0.978594 -0.137933 0.0135534 -0.0088242 -5.63716e-006 -0.000304015 -5.36596e-00
     -0.000195667 -0.000117984 9.76311e-005 -0.000124157 0.00962844 -0.00219756 0.97
     -7.57226e-006 -0.000127484 -5.82726e-005 -9.84651e-005 0.98037 -0.145206 0.014€
```

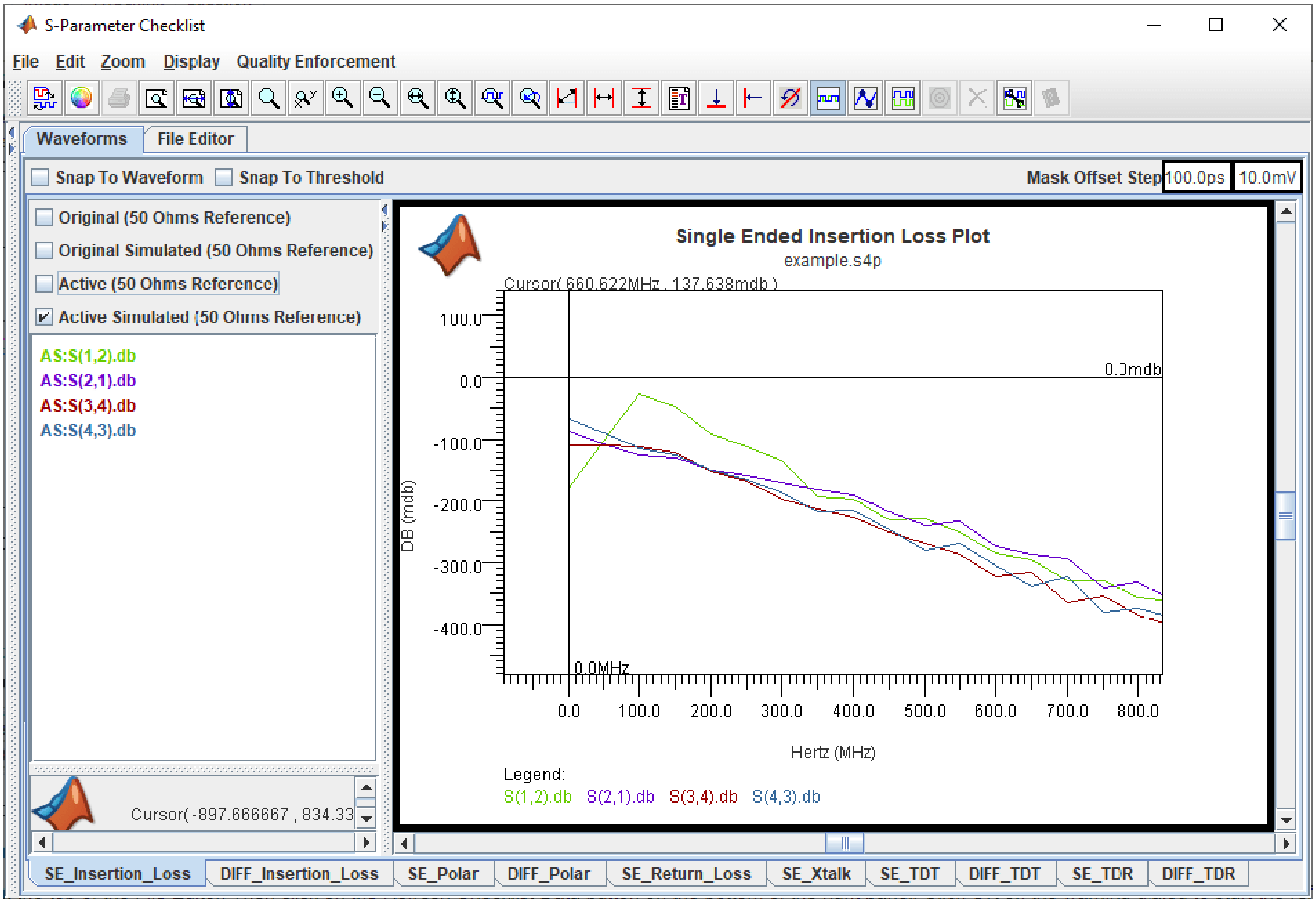
The warning about the port mapping algorithm is from the comparison of the 50MHz data in the file that is used to determine the pinout of the schematic symbol. This is due to the odd nature of the data in this S-Parameter file at low frequencies.

View Insertion Loss at Low Frequency

Click on the **SE_Insertion_Loss** tab at the left panel to take a look at the through path (insertion loss) data. Zoom in at lower frequencies. You can see there is some non-passive behavior and non-reciprocal behavior.



Enforce passivity to investigate further. Select Quality **Enforcement** > **Make Passive** from the toolstrip. To refresh the data, go back to the **Visual_Inspection** tab. Select **Enable Editing** at the top of the File Editor. Then click on the **Refresh Checklist Data** button on the bottom of the right panel. Refresh the data and look at the **SE_Insertion_Loss** tab again. To see only the refreshed data, deselect the **Active (50 Ohms Reference)** option above the node list.



☐ Snap To Waveform ☐ Snap To Threshold

Mask Offset Step 100.0ps 10.0mV

☐ Original (50 Ohms Reference)

☐ Original Simulated (50 Ohms Reference)

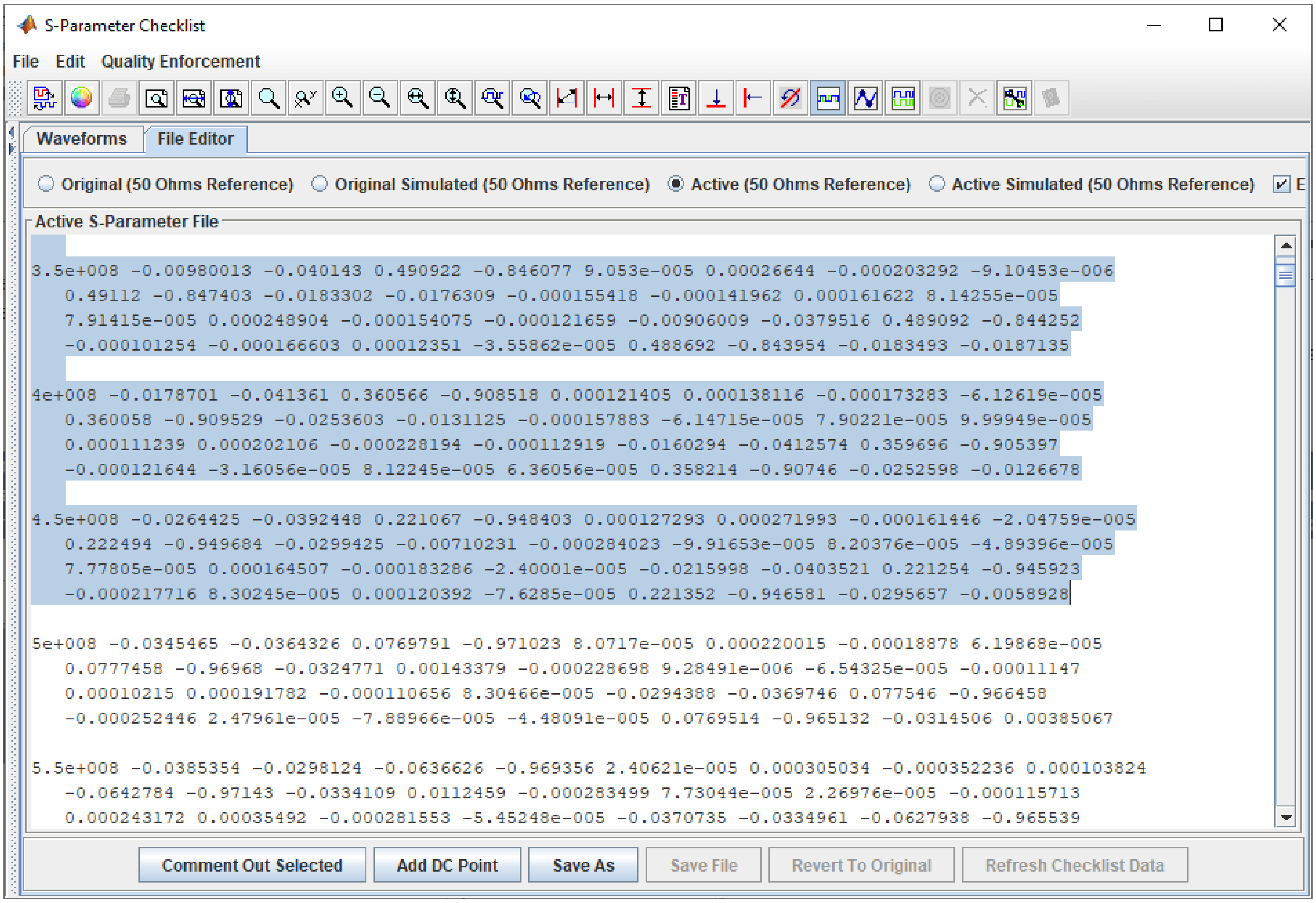
☐ Active (50 Ohms Reference)

☑ Active Simulated (50 Ohms Reference)

The insertion loss data is now all passive, but there are still some strange issues at the lowest frequencies. The loss is greater at DC than at low frequencies. This does not make sense for PCB traces.

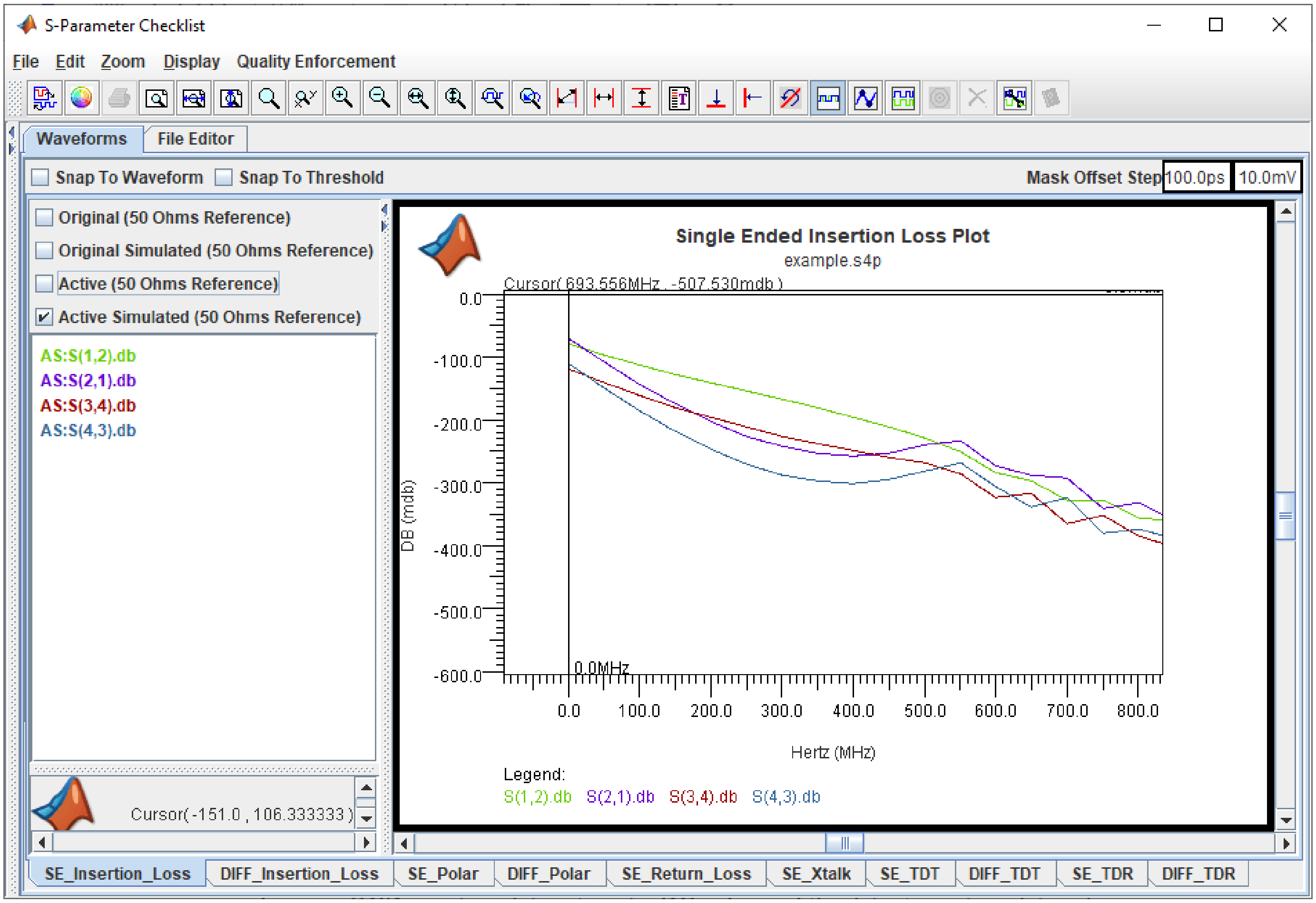
Delete Bad Low Frequency Data

To see if it the insertion loss behavior is caused by bad low frequency data you can delete the low frequency points from the file. Go back to the **Visual_Inspection** tab. In the right panel select the S-Parameter data for frequencies less than 500MHz and delete the data points. Save the changes by clicking the **Save File** button.


S-Parameter Checklist
File Edit Quality Enforcement
Waveforms
File Editor
○ Original (50 Ohms Reference)
○ Original Simulated (50 Ohms Reference)
● Active (50 Ohms Reference)
○ Active Simulated (50 Ohms Reference)
☑ E
Active S-Parameter File
Comment Out Selected
Add DC Point
Save As
Save File
Revert To Original
Refresh Checklist Data

```

3.5e+008 -0.00980013 -0.040143 0.490922 -0.846077 9.053e-005 0.00026644 -0.000203292 -9.10453e-006
   0.49112 -0.847403 -0.0183302 -0.0176309 -0.000155418 -0.000141962 0.000161622 8.14255e-005
   7.91415e-005 0.000248904 -0.000154075 -0.000121659 -0.00906009 -0.0379516 0.489092 -0.844252
   -0.000101254 -0.000166603 0.00012351 -3.55862e-005 0.488692 -0.843954 -0.0183493 -0.0187135

4e+008 -0.0178701 -0.041361 0.360566 -0.908518 0.000121405 0.000138116 -0.000173283 -6.12619e-005
   0.360058 -0.909529 -0.0253603 -0.0131125 -0.000157883 -6.14715e-005 7.90221e-005 9.99949e-005
   0.000111239 0.000202106 -0.000228194 -0.000112919 -0.0160294 -0.0412574 0.359696 -0.905397
   -0.000121644 -3.16056e-005 8.12245e-005 6.36056e-005 0.358214 -0.90746 -0.0252598 -0.0126678

4.5e+008 -0.0264425 -0.0392448 0.221067 -0.948403 0.000127293 0.000271993 -0.000161446 -2.04759e-005
   0.222494 -0.949684 -0.0299425 -0.00710231 -0.000284023 -9.91653e-005 8.20376e-005 -4.89396e-005
   7.77805e-005 0.000164507 -0.000183286 -2.40001e-005 -0.0215998 -0.0403521 0.221254 -0.945923
   -0.000217716 8.30245e-005 0.000120392 -7.6285e-005 0.221352 -0.946581 -0.0295657 -0.0058928

5e+008 -0.0345465 -0.0364326 0.0769791 -0.971023 8.0717e-005 0.000220015 -0.00018878 6.19868e-005
   0.0777458 -0.96968 -0.0324771 0.00143379 -0.000228698 9.28491e-006 -6.54325e-005 -0.00011147
   0.00010215 0.000191782 -0.000110656 8.30466e-005 -0.0294388 -0.0369746 0.077546 -0.966458
   -0.000252446 2.47961e-005 -7.88966e-005 -4.48091e-005 0.0769514 -0.965132 -0.0314506 0.00385067

5.5e+008 -0.0385354 -0.0298124 -0.0636626 -0.969356 2.40621e-005 0.000305034 -0.000352236 0.000103824
   -0.0642784 -0.97143 -0.0334109 0.0112459 -0.000283499 7.73044e-005 2.26976e-005 -0.000115713
   0.000243172 0.00035492 -0.000281553 -5.45248e-005 -0.0370735 -0.0334961 -0.0627938 -0.965539
```

Refresh the checklist data and go back to the **SE_Inserion_Loss** tab in the left panel.The loss is now increasing with frequency. The only issue is that the S(1,2) and S(4,3) data do not have the same DC point. This is because the **Serial Link Designer** app is extrapolating the data to get a DC value and the data it is extrapolating from are not the same for the two paths.



□ Snap To Waveform □ Snap To Threshold

Mask Offset Step 100.0ps 10.0mV

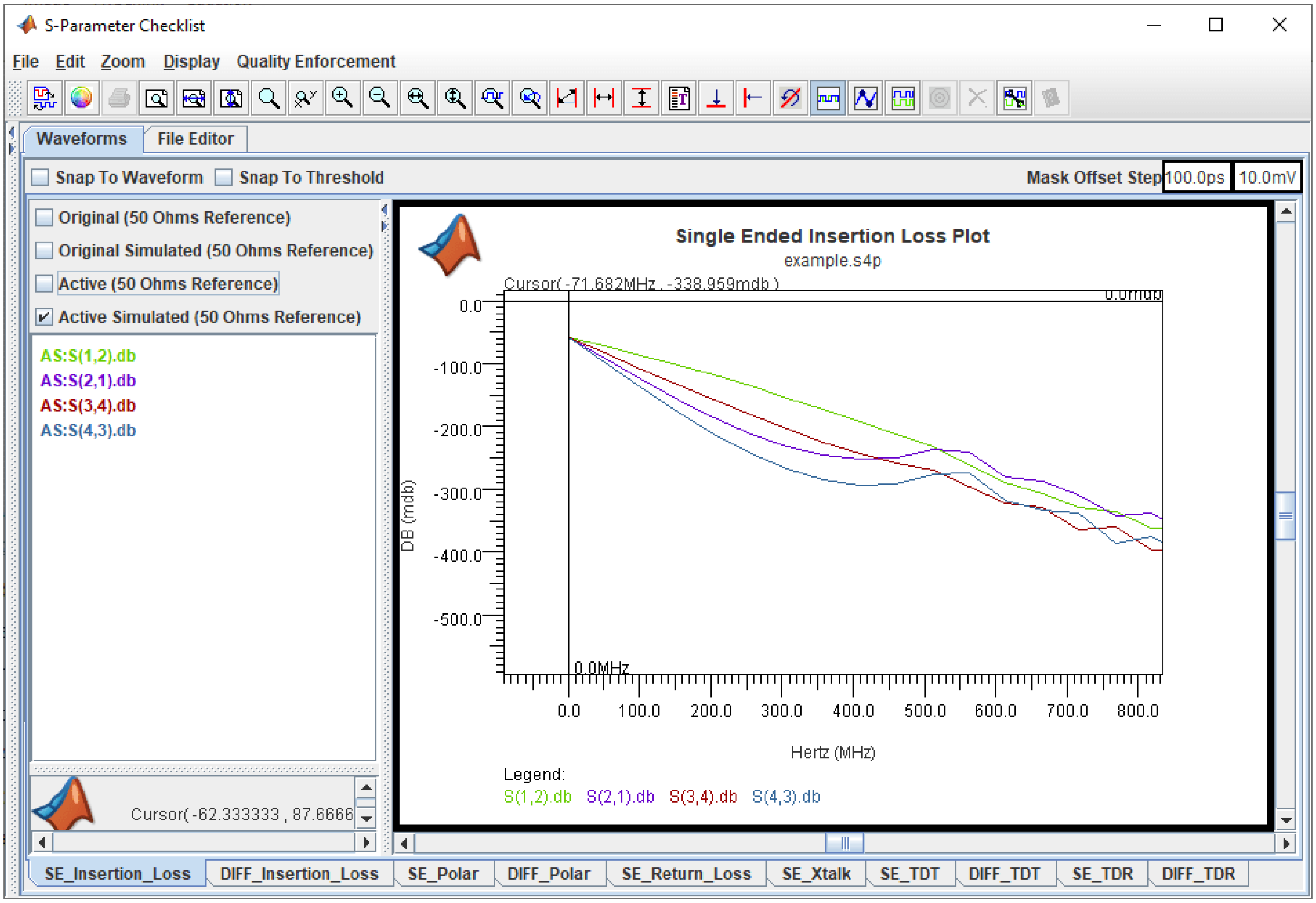
□ Original (50 Ohms Reference)
□ Original Simulated (50 Ohms Reference)
□ Active (50 Ohms Reference)
☑ Active Simulated (50 Ohms Reference)

AS:S(1,2).db
AS:S(2,1).db
AS:S(3,4).db
AS:S(4,3).db

Add DC Point

On the assumption that the lengths and thus the DC values should be the same for both paths you can add a DC point manually. Go back to the **Visual_Inspecion** tab in the left panel. Right click on the text editor in the right panel and select **Add DC Point**. On the Add DC Point dialog box that opens select **Trace Geometry Based Resistance Calculation.** Change the **Trace Length** to `2600` mils. The length is an estimate based on the maximum through path delay value that is in the S-Parameter file Metrics section of the left panel and an assumed delay of 180ps/in for stripline.

Click **Insert DC Point**, then save the file and refresh the checklist data. Go back to the **SE_Insertion_Loss** tab in the left tab. Now the insertion loss is increasing with frequency and all the data has the same DC point.



□ Snap To Waveform □ Snap To Threshold

Mask Offset Step 100.0ps 10.0mV

□ Original (50 Ohms Reference)

□ Original Simulated (50 Ohms Reference)

□ Active (50 Ohms Reference)

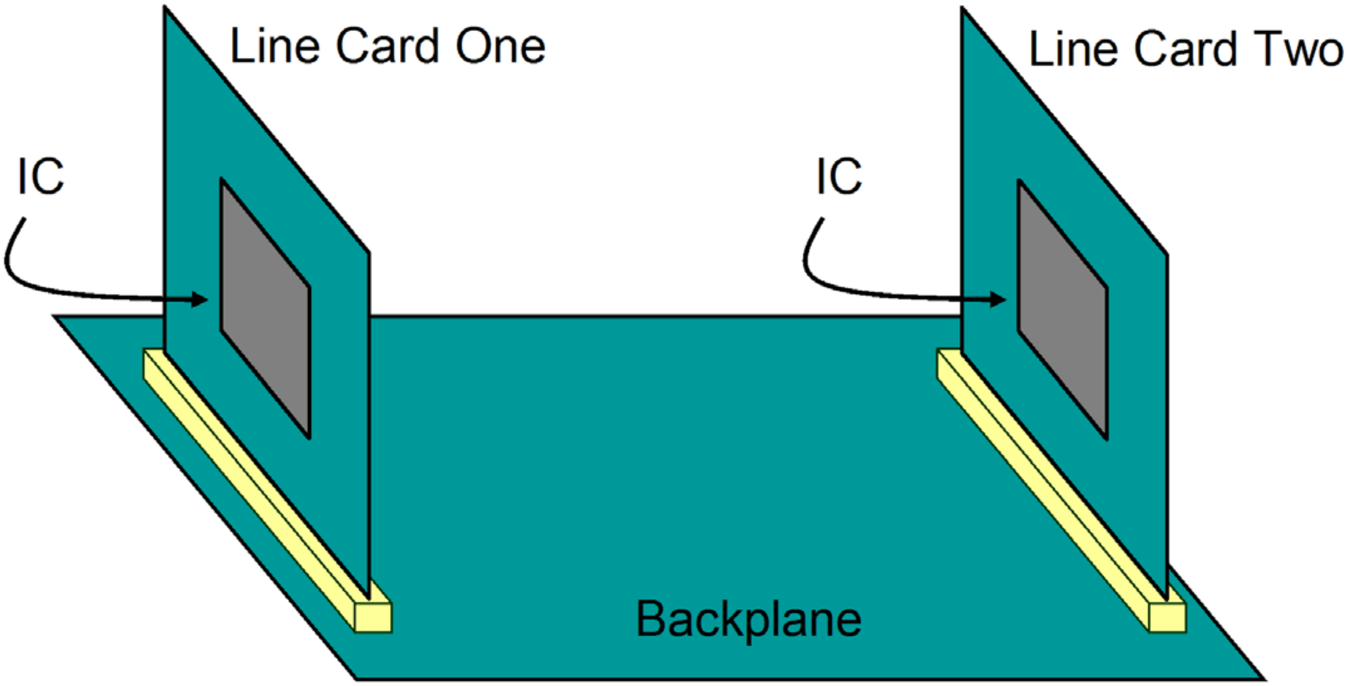
☑ Active Simulated (50 Ohms Reference)

AS:S(1,2).db
AS:S(2,1).db
AS:S(3,4).db
AS:S(4,3).db

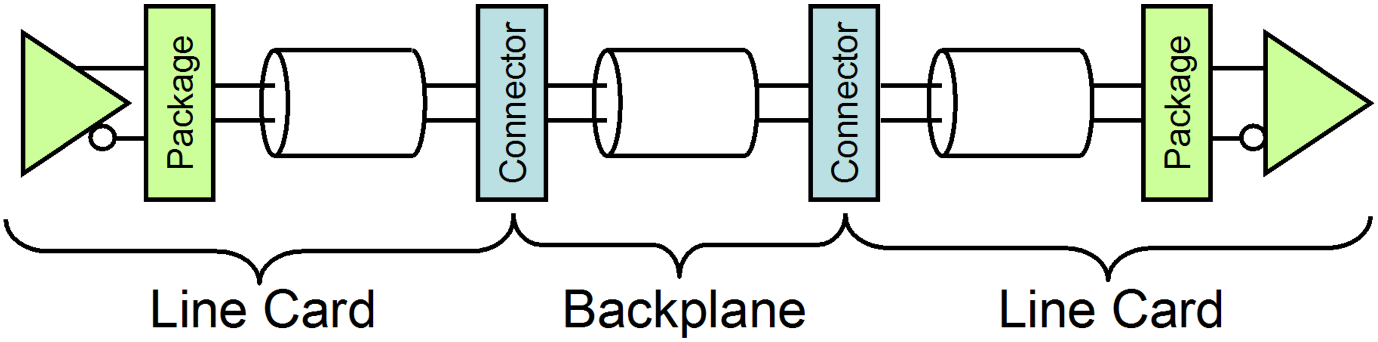
Analyze Backplane with Line Cards

This example shows how you can analyze a serial link consisting of a backplane and two line cards with the **Serial Link Designer** app. You can model the SerDes drivers/receivers, capture a topology for analysis, run network characterization, and evaluate the impact of different solution space variables on your design's performance.

The serial link to be modeled is a backplane with two line cards.



The channel topology is represented by:



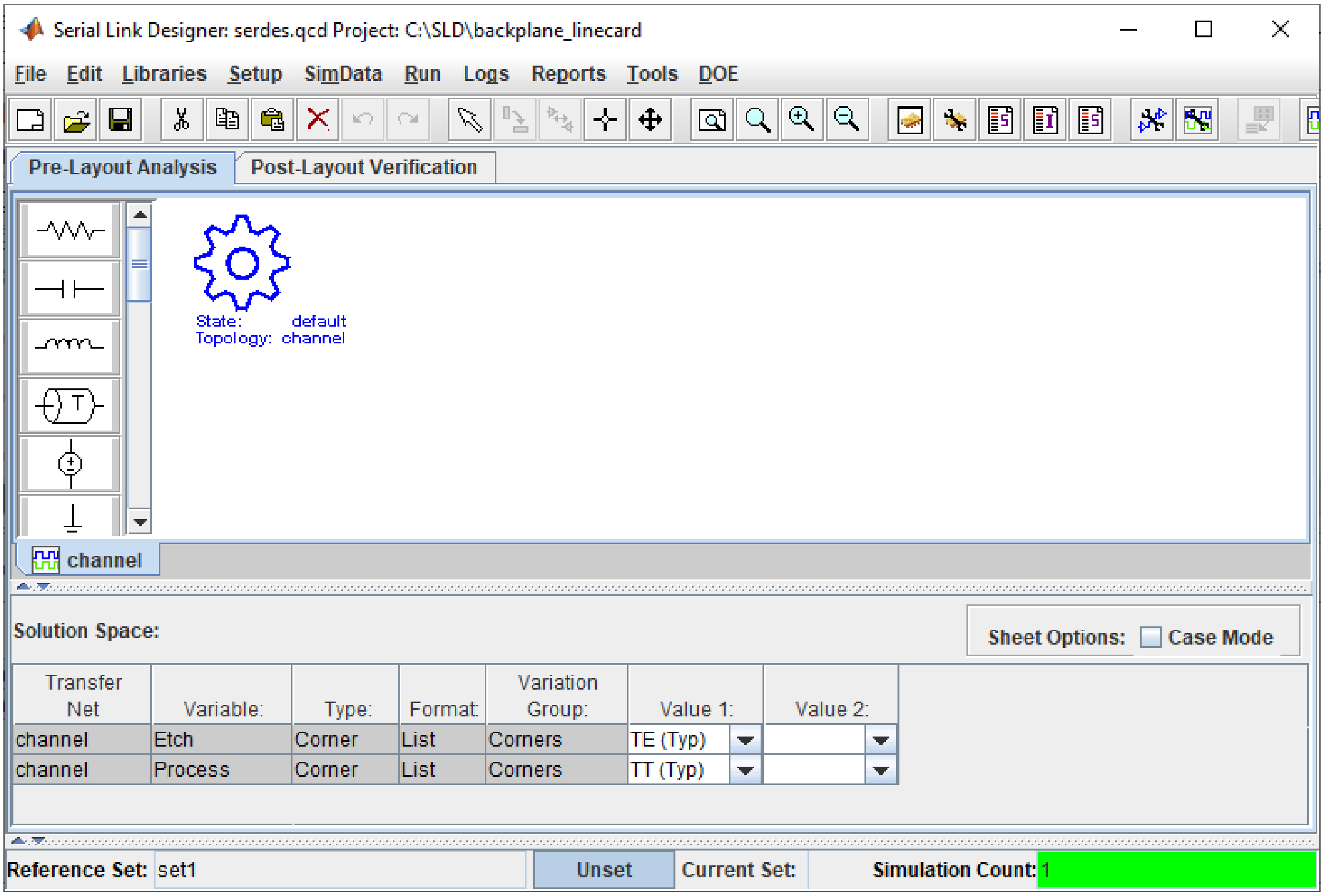
The packages and connectors are modeled with S-Parameters. The traces are modeled with w-lines.

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 `backplane_linecard`, the interface as `serdes`, and the schematic sheet as `channel`. The **Pre-Layout Analysis** tab shows the blank schematic sheet.


Serial Link Designer: serdes.qcd Project: C:\SLD\backplane_linecard
File Edit Libraries Setup SimData Run Logs Reports Tools DOE
Pre-Layout Analysis
Post-Layout Verification
State: default
Topology: channel
channel
Unset

Solution Space:

Sheet Options: □ Case Mode

Transfer Net	Variable:	Type:	Format:	Variation Group:	Value 1:	Value 2:
channel	Etch	Corner	List	Corners	TE (Typ)	
channel	Process	Corner	List	Corners	TT (Typ)	

Reference Set: set1

Current Set:

Simulation Count: 1

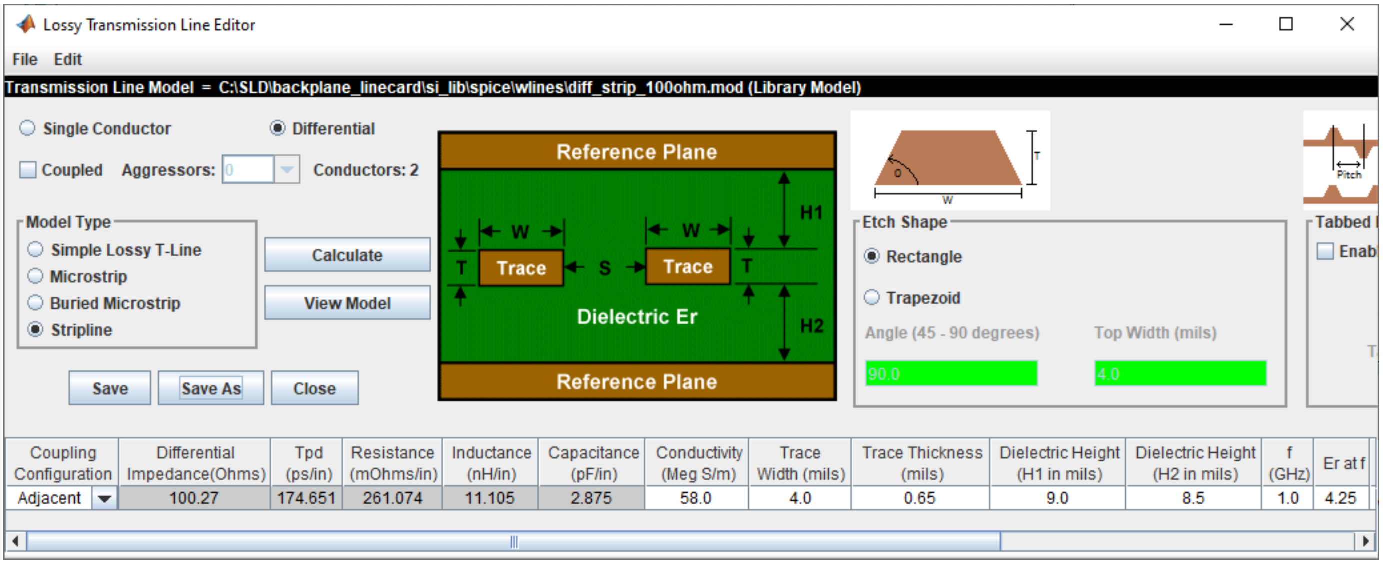
Setup Libraries

You can create the library elements for the transmission lines, packages, connectors, and designators.

Create a differential lossy transmission line model based on a stripline cross-section. Select **Tools** > **Lossy Transmission Line Editor**. In the newly opened Lossy Transmission Line Editor dialog box, select **Differential** and select Model Type as **Stripline**.

The traces are 4 mils wide and 0.65 mils thick. They are 9.0 mils above and 8.5 mils below planes with a dielectric constant of Er 4.25. The trace separation is 5 mils. So change the parameters **Dielectric Height (H1 in mils)** to 9, **Dielectric Height (H2 in mils)** to `8.5`, and **Differential Separation (mils)** to 5.

Click the **Calculate** button to run the 2-D field solver. The Impedance at the bottom left changes from derived to the calculated value.



Lossy Transmission Line Editor

File Edit

Transmission Line Model = C:\SLD\backplane_linecard\si_lib\spice\wlines\diff_strip_100ohm.mod (Library Model)

○ Single Conductor ● Differential

□ Coupled Aggressors: 0 Conductors: 2

Model Type
○ Simple Lossy T-Line
○ Microstrip
○ Buried Microstrip
● Stripline

Calculate

View Model

Save | Save As | Close

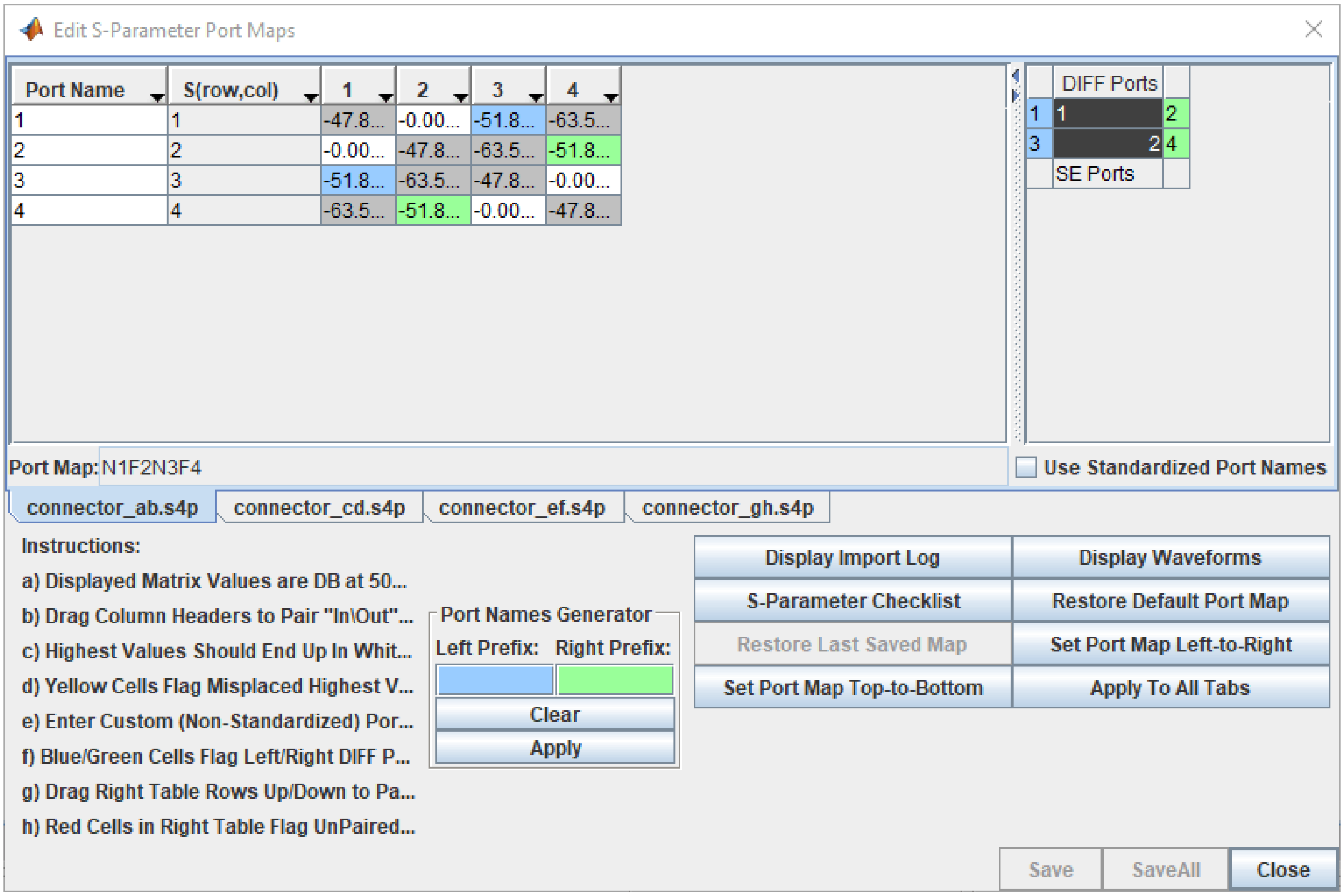
Etch Shape
● Rectangle
○ Trapezoid
Angle (45 - 90 degrees): 90.0
Top Width (mils): 4.0

Tabbed I
□ Enab

Coupling Configuration	Differential Impedance(Ohms)	Tpd (ps/in)	Resistance (mOhms/in)	Inductance (nH/in)	Capacitance (pF/in)	Conductivity (Meg S/m)	Trace Width (mils)	Trace Thickness (mils)	Dielectric Height (H1 in mils)	Dielectric Height (H2 in mils)	f (GHz)	Er at f
Adjacent	100.27	174.651	261.074	11.105	2.875	58.0	4.0	0.65	9.0	8.5	1.0	4.25

Click the **Save As** button to save the model in the project's library. Use the default name diff_strip_100ohm. Make sure the directory is <Project Library>/spice/wlines. Close the Lossy Transmission Line Editor.

Four custom S-Parameter data files (connector_ab.s4p, connector_cd.s4p, connector_ef.s4p, and connector_gh.s4p) are attached as supporting files to this example. Download all four Touchstone® (.s4p) files. To import the connector S-Parameter data, select **Libraries** > **Import S-Parameter**. Browse to the location where you saved the downloaded Touchstone files and select all four. Verify that the **Merge Wrappers** checkbox is selected on the Import S-Parameter File(s) dialog box. Merging the connector wrappers makes it possible to sweep them. Import the files. This launches the Edit S-Parameter Port Maps dialog box. The dialog box contains a separate tab for each connector file.



Port Name	S(row,col)	1	2	3	4
1	1	-47.8...	-0.00...	-51.8...	-63.5...
2	2	-0.00...	-47.8...	-63.5...	-51.8...
3	3	-51.8...	-63.5...	-47.8...	-0.00...
4	4	-63.5...	-51.8...	-0.00...	-47.8...

	DIFF Ports	
1	1	2
3	2	4
	SE Ports	

Port Map: N1F2N3F4

Instructions:

a) Displayed Matrix Values are DB at 50...

b) Drag Column Headers to Pair "In\Out"...

c) Highest Values Should End Up In Whit...

d) Yellow Cells Flag Misplaced Highest V...

e) Enter Custom (Non-Standardized) Por...

f) Blue/Green Cells Flag Left/Right DIFF P...

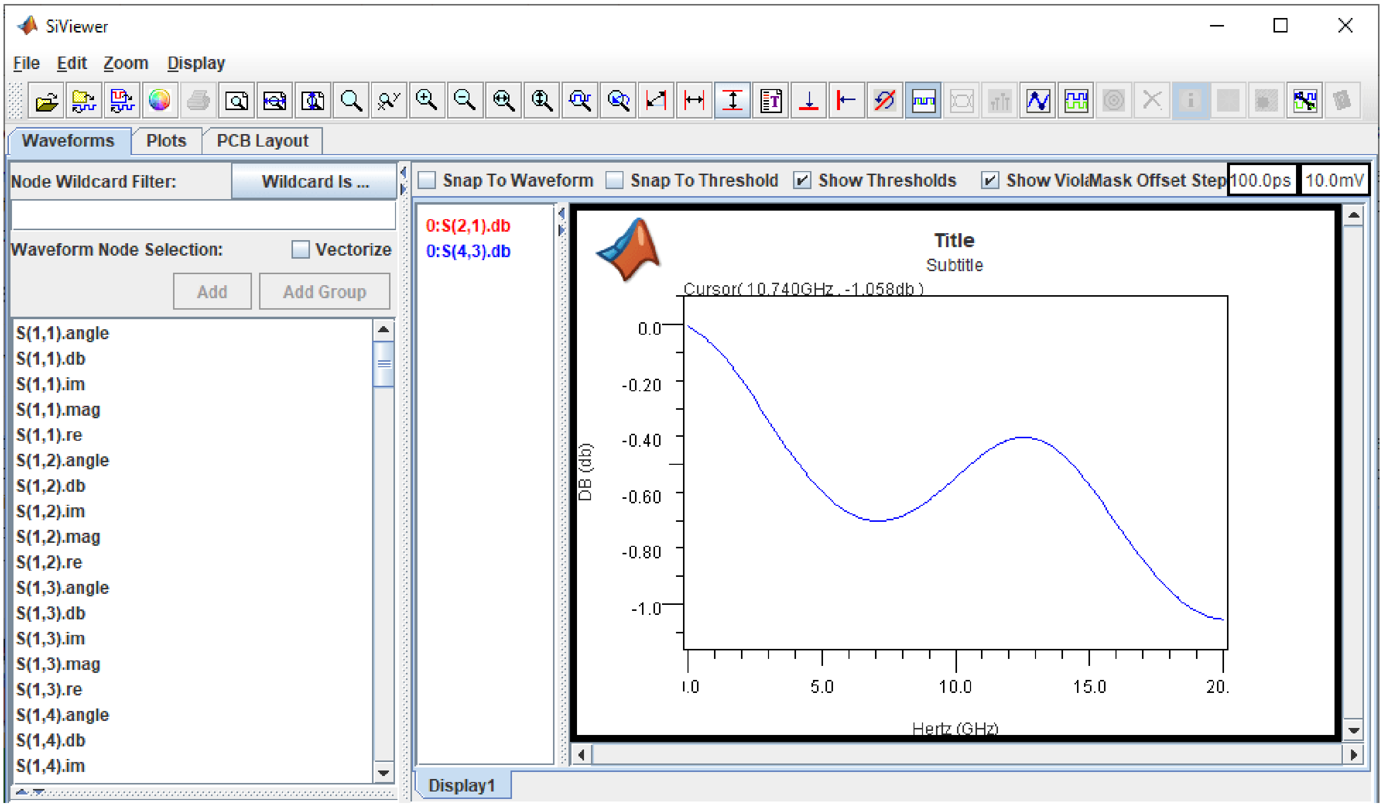
g) Drag Right Table Rows Up/Down to Pa...

h) Red Cells in Right Table Flag UnPaired...

The table on the left shows the loss at 50 MHz between each pair of ports. The cells in white show the smallest loss. Generally, the smallest loss occurs at the ports that are the through path. The blue cells indicate the left-hand differential port. The green cells indicate the right-hand differential port.

The table on the right shows orientation of the S-parameter block as it will be on the schematic sheet and identifies the differential ports.

To view the through path dB vs. frequency responses of the single-ended paths, click the **Display Waveforms** button. This launches the **Signal Integrity Viewer** app.



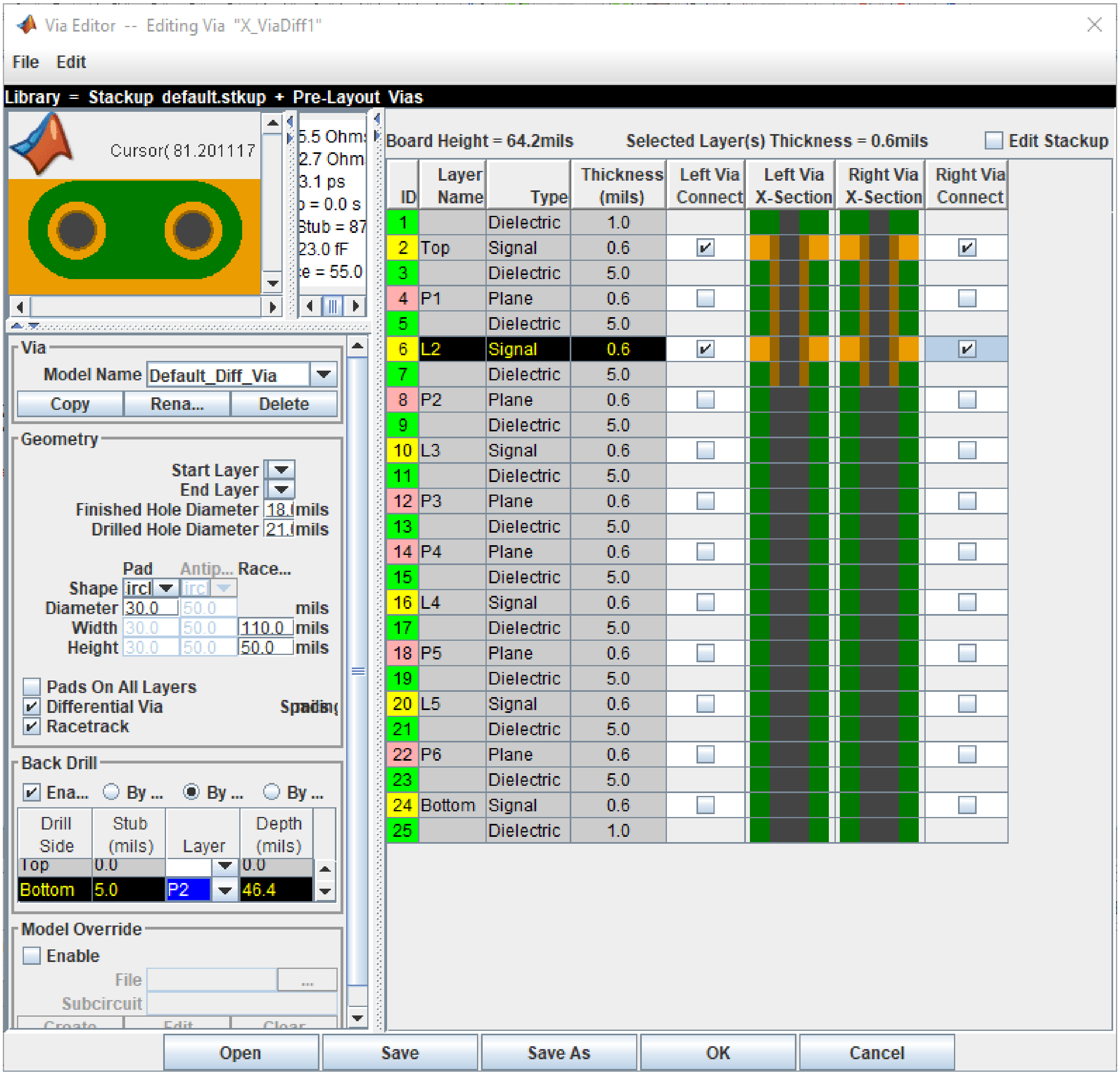
You can add new display to view all the data in real/imaginary, magnitude/angle and dB. Close the **Signal Integrity Viewer** app, Edit S-Parameter Port Maps dialog box, and Import S-Parameter File(s) dialog box.

Create Channel Schematic

Add the backplane transmission line by selecting the differential lossy transmission line element to the blank canvas on the Pre-Layout Analysis tab. Right-click on the symbol and choose **Select T-Line Model**. Switch to the `<Project Library>/spice/wlines` library if it is not selected. Select the `diff_strip_100ohm` model.

Add two differential via models between the backplane traces and the connector.

To start, add a new differential via element with 12 layers of connecting layers to create the default stackup to the left of the transmission line. Right-click on the via symbol and choose **Edit Differential Via Model** to launch the Via Editor dialog box. The default via connects the top layer to the bottom layer. Uncheck the **Left Via Connect** and **Right Via Connect** checkboxes for the layer `Bottom` and check the checkboxes for layer `L2`. This changes the via to a via that is connecting the layer Top to the layer L2. It is still a through-hole via with a stub from layer L2 to the bottom of the board. To model a backdrilled via check **Enable** in the Backdrill panel, check **By Layer**, then select layer P2 in the list for **Bottom**. The layers view will change to show that the barrel of the via is gone from the bottom through layer P2.



Via Editor -- Editing Via "X_ViaDiff1"

File Edit

Library = Stackup default.stkup + Pre-Layout Vias

Cursor(81.201117

5.5 Ohm
2.7 Ohm
3.1 ps
p = 0.0 s
Stub = 87
23.0 fF
e = 55.0

Via

Model Name: Default_Diff_Via

Copy | Rena... | Delete

Geometry

Start Layer
End Layer
Finished Hole Diameter 18.(mils
Drilled Hole Diameter 21.(mils

	Pad	Antip...	Race...	
Shape	ircl	irc		
Diameter	30.0	50.0		mils
Width	30.0	50.0	110.0	mils
Height	30.0	50.0	50.0	mils

□ Pads On All Layers
☑ Differential Via
☑ Racetrack
Spacing

Back Drill

☑ Ena... ○ By ... ● By ... ○ By ...

Drill Side	Stub (mils)	Layer	Depth (mils)
Top	0.0		0.0
Bottom	5.0	P2	46.4

Model Override

□ Enable
File
Subcircuit

Create | Edit | Clear

Board Height = 64.2mils   Selected Layer(s) Thickness = 0.6mils   □ Edit Stackup

ID	Layer Name	Type	Thickness (mils)	Left Via Connect	Left Via X-Section	Right Via X-Section	Right Via Connect
1		Dielectric	1.0				
2	Top	Signal	0.6	☑			☑
3		Dielectric	5.0				
4	P1	Plane	0.6	□			□
5		Dielectric	5.0				
6	L2	Signal	0.6	☑			☑
7		Dielectric	5.0				
8	P2	Plane	0.6	□			□
9		Dielectric	5.0				
10	L3	Signal	0.6	□			□
11		Dielectric	5.0				
12	P3	Plane	0.6	□			□
13		Dielectric	5.0				
14	P4	Plane	0.6	□			□
15		Dielectric	5.0				
16	L4	Signal	0.6	□			□
17		Dielectric	5.0				
18	P5	Plane	0.6	□			□
19		Dielectric	5.0				
20	L5	Signal	0.6	□			□
21		Dielectric	5.0				
22	P6	Plane	0.6	□			□
23		Dielectric	5.0				
24	Bottom	Signal	0.6	□			□
25		Dielectric	1.0				

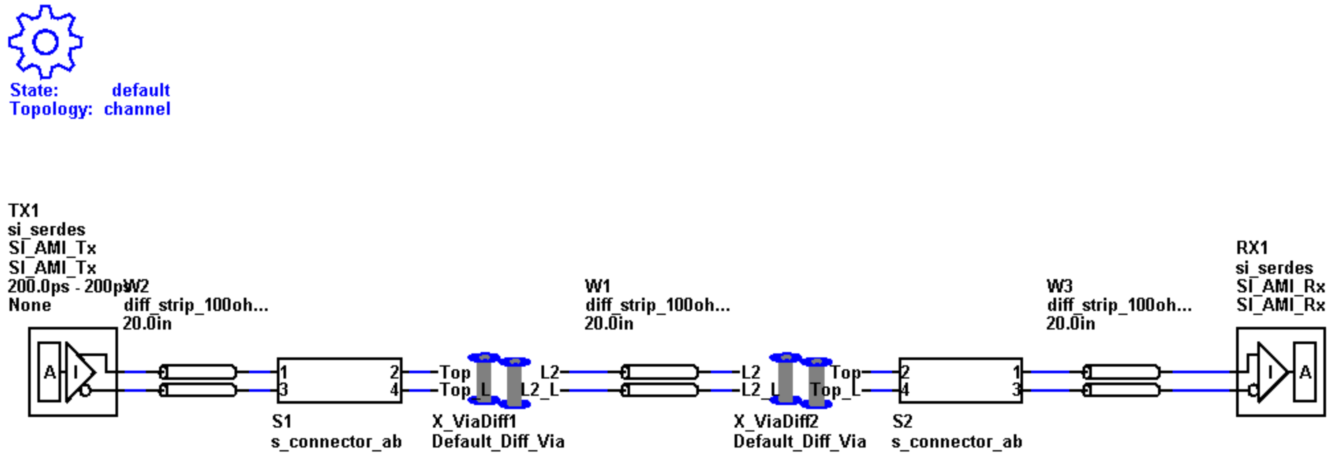
Open | Save | Save As | OK | Cancel

Save and close the Via Editor dialog box. Copy, paste, and mirror another via to the right of the transmission line.

To add the connectors, add a new S-Parameter element. Choose `connector_s4p.smod` and `s_connector_ab` from the `<Project Library>/spice/s_params` directory in the Select S-Parameter Model dialog box. Add two connectors (mirrored) on the left and right of the vias.

Copy the backplane transmission line symbol and paste one copy on the far left and one copy on the far right to represent the traces on the two line cards. Add two differential buffer elements (mirrored) and place one in the far left to designate the transmitter and one in the far right to designate the receiver.

Connect the elements together to complete the schematic.



Double-click on one of the W-line symbols to launch the Lossy Transmission Line Element Properties dialog box. Enable the **Sweep Length** checkbox for each w-line. Change the name of the backplane symbol to `$bp_len` and the line card symbols to `$lc_len`. By changing the two line card w-lines to the same name you can use the same solution space variable for both w-line symbols. Close the Lossy Transmission Line Element Properties dialog box.

In the Solution Space panel, change the **Value 1** value for **Variable** `$bp_len` to 16in and **Variable** `$lc_len` to 3in.

Double-click on one of the connector symbols to launch the Spice Subcircuit Element Properties dialog box. There are two rows, one for each connector symbol. Enable the **Sweep Model** checkbox in each row and change the variable names to `$connector`.

Double click on the TX symbol to launch the Designator Element Properties dialog box. Set the UI (Unit Interval) for TX1 to `Serdes_10G` by selecting it from the dropdown menu of the **UI** parameter. The UI is set to 100 ps. Save the changes to the schematic.

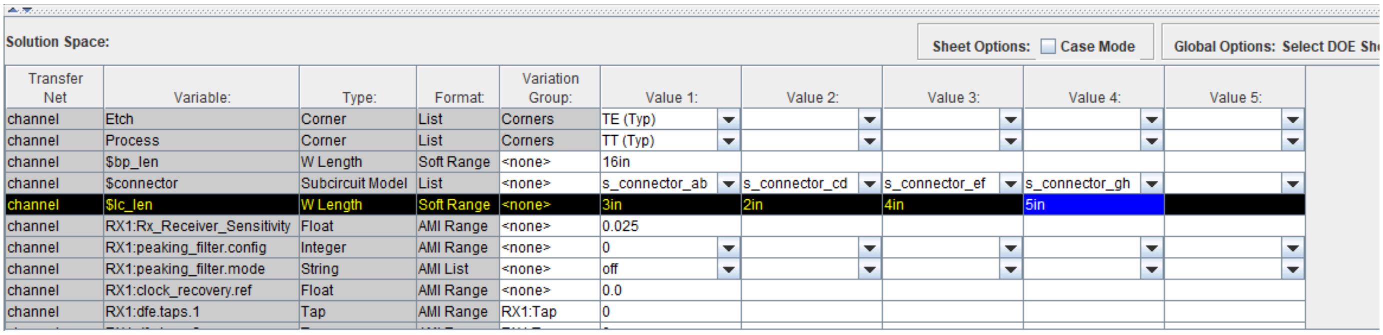
Validate the schematic set by selecting **Run** > **Validate Current Schematic Set**. The validation should run without warning or errors.

Network Characterization

To see the effects of sweeping the package model, connector model, and line card trace lengths on the physical channel characteristics, run network characterization. Network characterization derives the LTI signature of the analog network. The analog network includes the analog TX and RX characteristics as well as the channel elements themselves. The **Serial Link Designer** app frequency domain network solver derives the transfer function of the analog network. From the transfer function, the app derives the impulse and step response. The app also derives the pulse response using the UI set during schematic creation. It also computes the insertion loss, return loss, ripple, impulse width and other metrics.

To sweep the connector model, select the `$connector` **Variable**, right click and select **Set All Values**. The solution space becomes populated with the four models you imported.

To sweep the line card length, select the `$lc_len` **Variable** and add the values 2in, 4in, and 5in. Save the changes.



Solution Space:

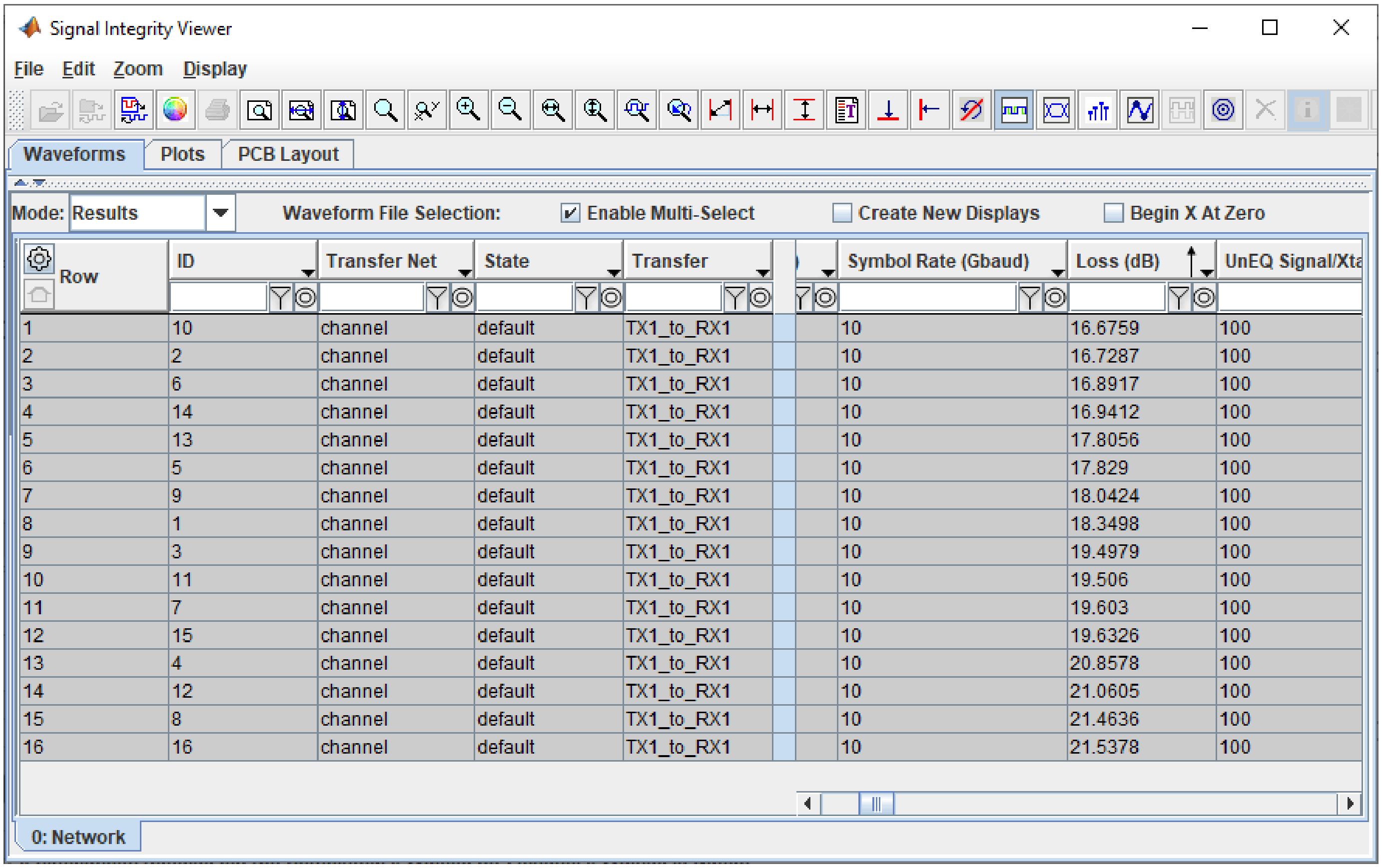
Sheet Options: □ Case Mode

Global Options: Select DOE Sh

Transfer Net	Variable:	Type:	Format:	Variation Group:	Value 1:	Value 2:	Value 3:	Value 4:	Value 5:
channel	Etch	Corner	List	Corners	TE (Typ)				
channel	Process	Corner	List	Corners	TT (Typ)				
channel	$bp_len	W Length	Soft Range	<none>	16in				
channel	$connector	Subcircuit Model	List	<none>	s_connector_ab	s_connector_cd	s_connector_ef	s_connector_gh	
channel	$lc_len	W Length	Soft Range	<none>	3in	2in	4in	5in	
channel	RX1:Rx_Receiver_Sensitivity	Float	AMI Range	<none>	0.025				
channel	RX1:peaking_filter.config	Integer	AMI Range	<none>	0				
channel	RX1:peaking_filter.mode	String	AMI List	<none>	off				
channel	RX1:clock_recovery.ref	Float	AMI Range	<none>	0.0				
channel	RX1:dfe.taps.1	Tap	AMI Range	RX1:Tap	0				

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 **Signal Integrity 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 difference between the lowest (16.67dB) and highest (21.54dB) loss is around 5 dB.

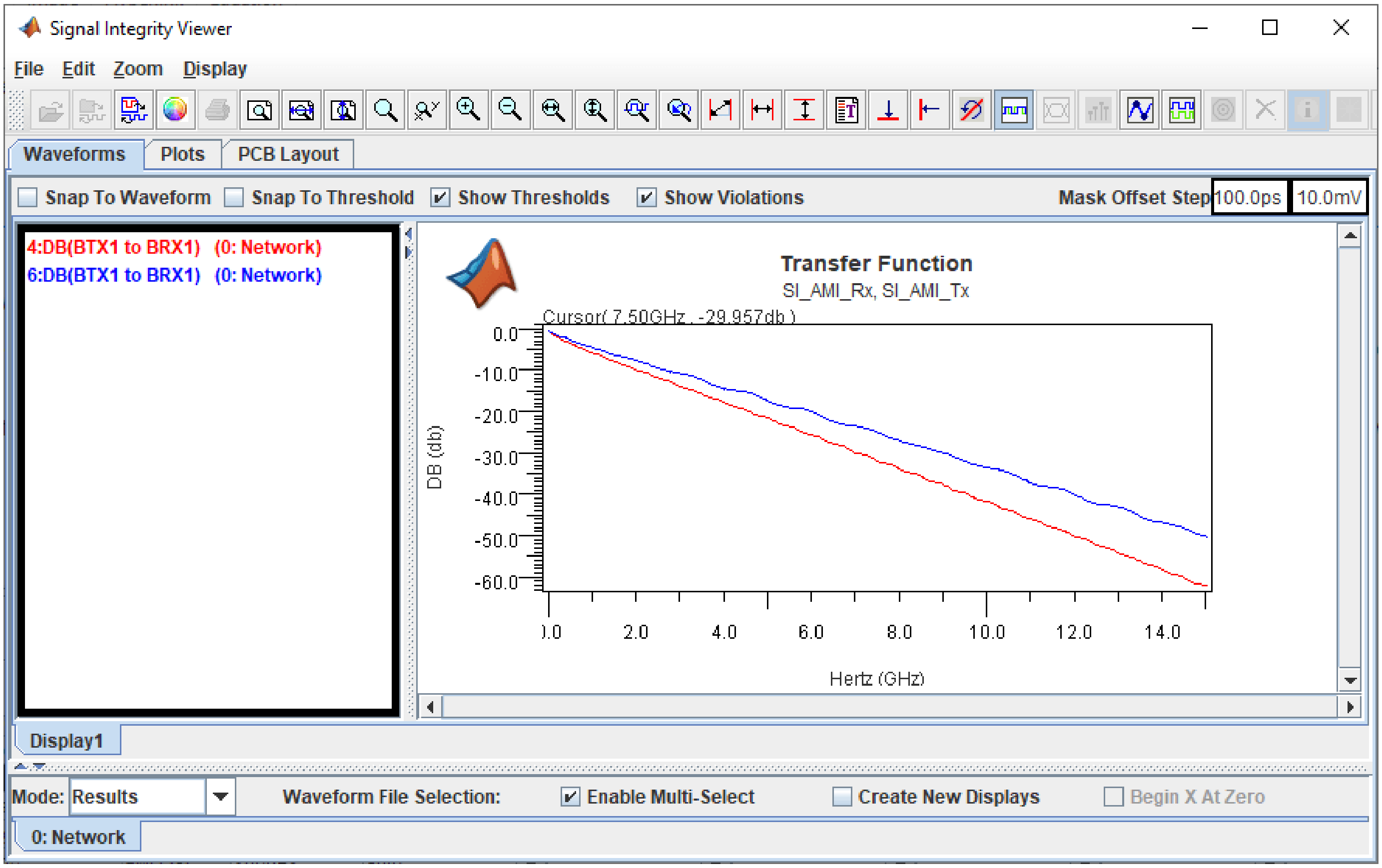


Mode: Results

Waveform File Selection: ☑ Enable Multi-Select ☐ Create New Displays ☐ Begin X At Zero

Row	ID	Transfer Net	State	Transfer	)	Symbol Rate (Gbaud)	Loss (dB)	UnEQ Signal/Xta
1	10	channel	default	TX1_to_RX1		10	16.6759	100
2	2	channel	default	TX1_to_RX1		10	16.7287	100
3	6	channel	default	TX1_to_RX1		10	16.8917	100
4	14	channel	default	TX1_to_RX1		10	16.9412	100
5	13	channel	default	TX1_to_RX1		10	17.8056	100
6	5	channel	default	TX1_to_RX1		10	17.829	100
7	9	channel	default	TX1_to_RX1		10	18.0424	100
8	1	channel	default	TX1_to_RX1		10	18.3498	100
9	3	channel	default	TX1_to_RX1		10	19.4979	100
10	11	channel	default	TX1_to_RX1		10	19.506	100
11	7	channel	default	TX1_to_RX1		10	19.603	100
12	15	channel	default	TX1_to_RX1		10	19.6326	100
13	4	channel	default	TX1_to_RX1		10	20.8578	100
14	12	channel	default	TX1_to_RX1		10	21.0605	100
15	8	channel	default	TX1_to_RX1		10	21.4636	100
16	16	channel	default	TX1_to_RX1		10	21.5378	100

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



□ Snap To Waveform □ Snap To Threshold ☑ Show Thresholds ☑ Show Violations Mask Offset Step 100.0ps 10.0mV

4:DB(BTX1 to BRX1) (0: Network)
6:DB(BTX1 to BRX1) (0: Network)

Mode: Results ☑ Enable Multi-Select □ Create New Displays □ Begin X At Zero

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

Statistical Channel Analysis

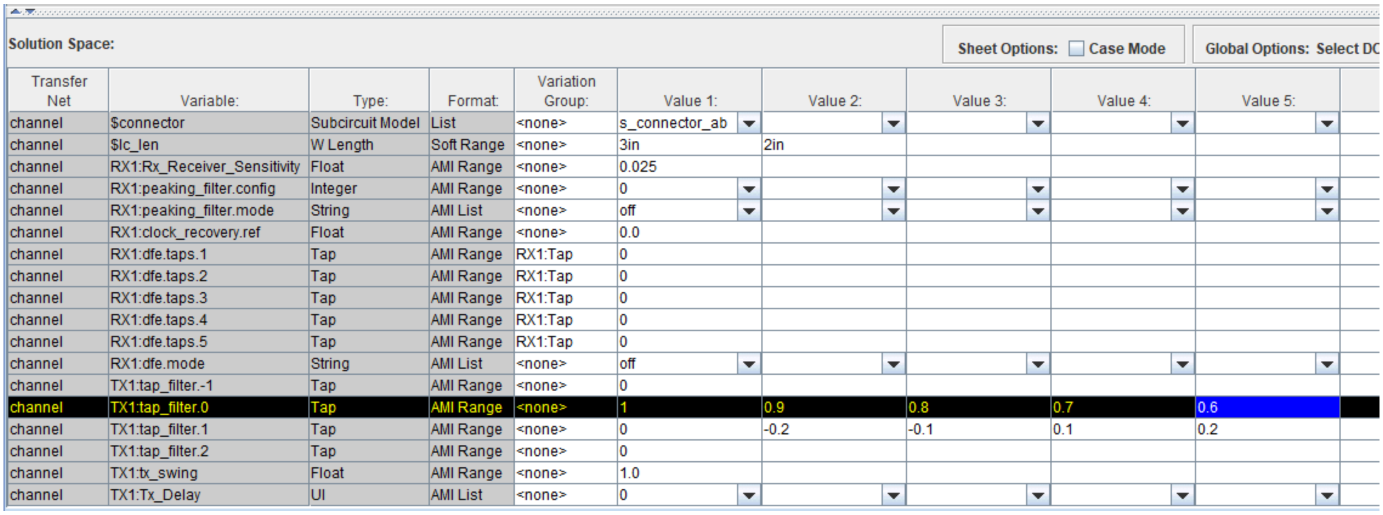
Statistical analysis can analyze the channel with LTI TX and RX equalization. This example shows how you can sweep the TX equalization and RX CTLE for statistical analysis.

To remove the solution space entries for the connector model, select the `$connector` **Variable**, right click and select **Set to Default**. This will leave an entry in Value 1 only for the connector. Delete the entries for 4in and 5in for $lc_len by removing the columns.

Select the symbol for TX1 on the schematic to highlight the solution space table rows for the TX AMI parameters. The transmitter has three taps in the Variation Group TX1:tap. Delete the variation group from the taps so that they can be swept independently.

Select the `TX1:tap_filter.0` **Variable** and add the values `0.9`, `0.8`, and `0.7`.

Select the `TX1:tap_filter.1` **Variable** and add the values `-0.2`, `-0.1`, `0.1,` and `0.2`. Save the changes.



Solution Space:

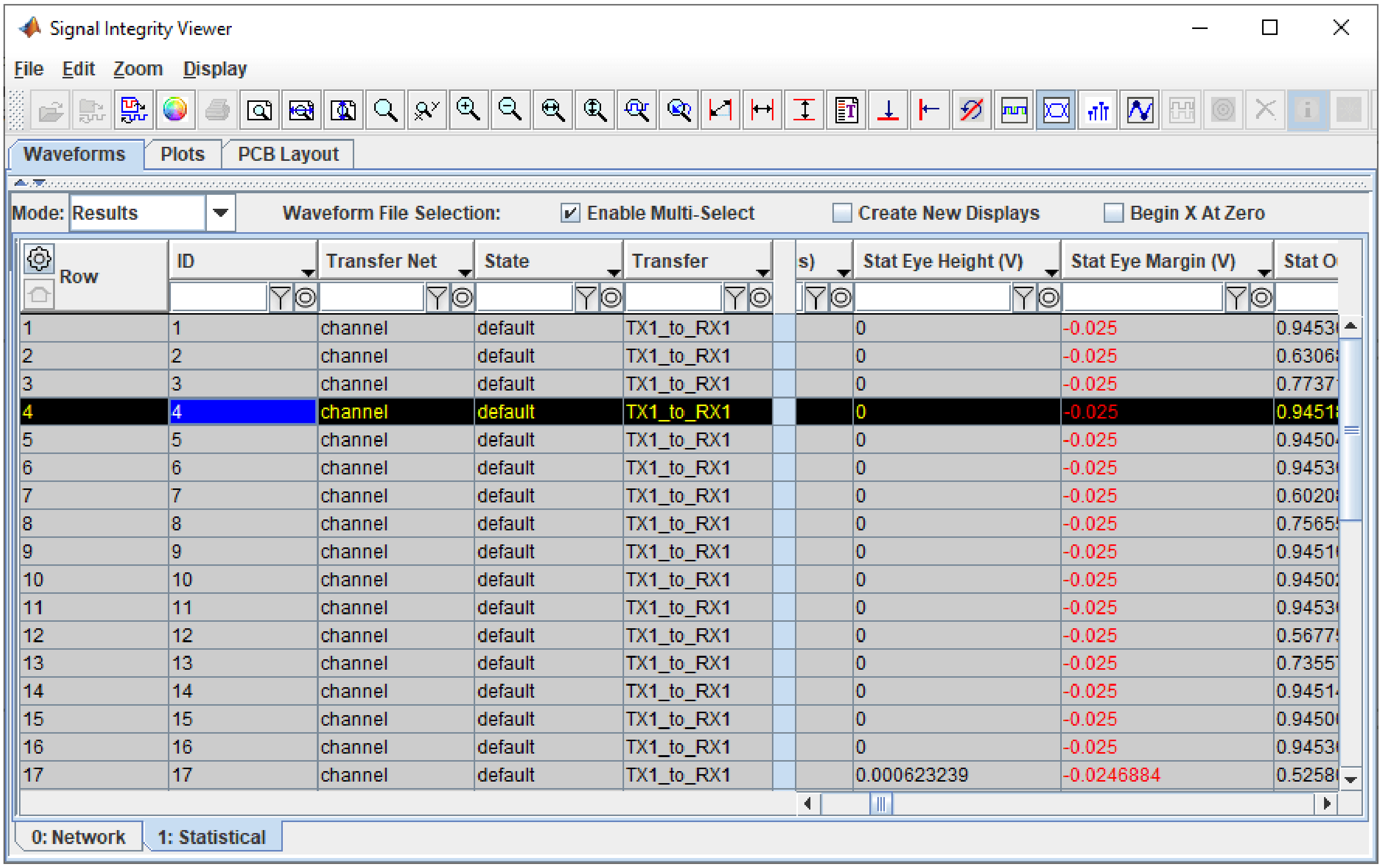
Sheet Options: □ Case Mode

Global Options: Select DC

Transfer Net	Variable:	Type:	Format:	Variation Group:	Value 1:	Value 2:	Value 3:	Value 4:	Value 5:
channel	$connector	Subcircuit Model	List	<none>	s_connector_ab				
channel	$lc_len	W Length	Soft Range	<none>	3in	2in			
channel	RX1:Rx_Receiver_Sensitivity	Float	AMI Range	<none>	0.025				
channel	RX1:peaking_filter.config	Integer	AMI Range	<none>	0				
channel	RX1:peaking_filter.mode	String	AMI List	<none>	off				
channel	RX1:clock_recovery.ref	Float	AMI Range	<none>	0.0				
channel	RX1:dfe.taps.1	Tap	AMI Range	RX1:Tap	0				
channel	RX1:dfe.taps.2	Tap	AMI Range	RX1:Tap	0				
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				
channel	TX1:tap_filter.-1	Tap	AMI Range	<none>	0				
channel	TX1:tap_filter.0	Tap	AMI Range	<none>	1	0.9	0.8	0.7	0.6
channel	TX1:tap_filter.1	Tap	AMI Range	<none>	0	-0.2	-0.1	0.1	0.2
channel	TX1:tap_filter.2	Tap	AMI Range	<none>	0				
channel	TX1:tx_swing	Float	AMI Range	<none>	1.0				
channel	TX1:Tx_Delay	UI	AMI List	<none>	0				

Run the simulation. In the Prelayout Channel Analysis dialog box, select **Validate**, **Generate Netlists**, **Include Statistical Analysis**, **Perform Channel Analysis**, and **Autoload Results**. The **Signal Integrity Viewer** app launches when the simulation is complete.

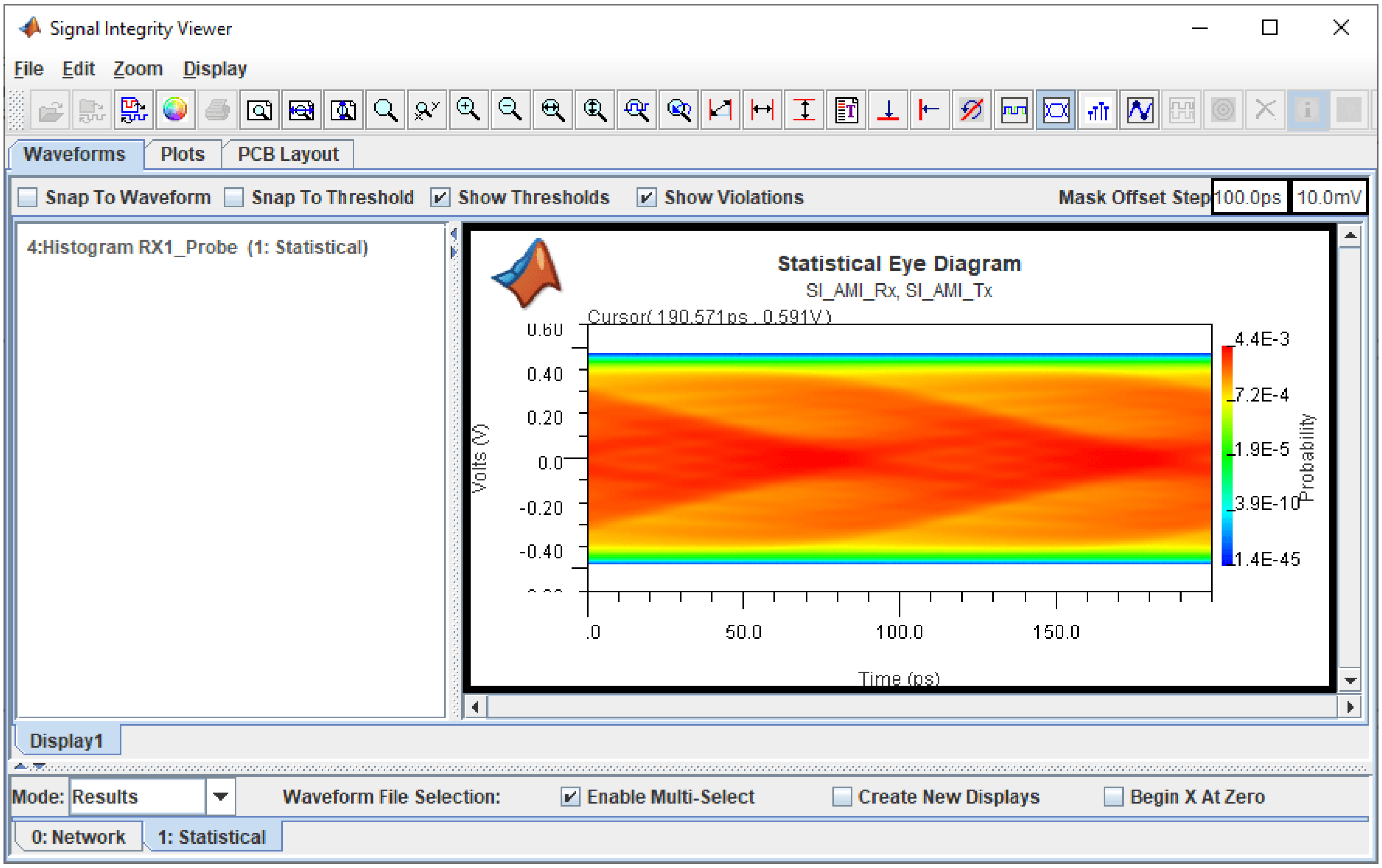
On the Statistical tab of the Signal Integrity Viewer window, click on the column header for **Stat Eye Margin (V)**. The margin is negative on all of the simulation. In fact the eye is completely closed on all sims, so TX equalization is not enough to get this channel working.



Mode: Results

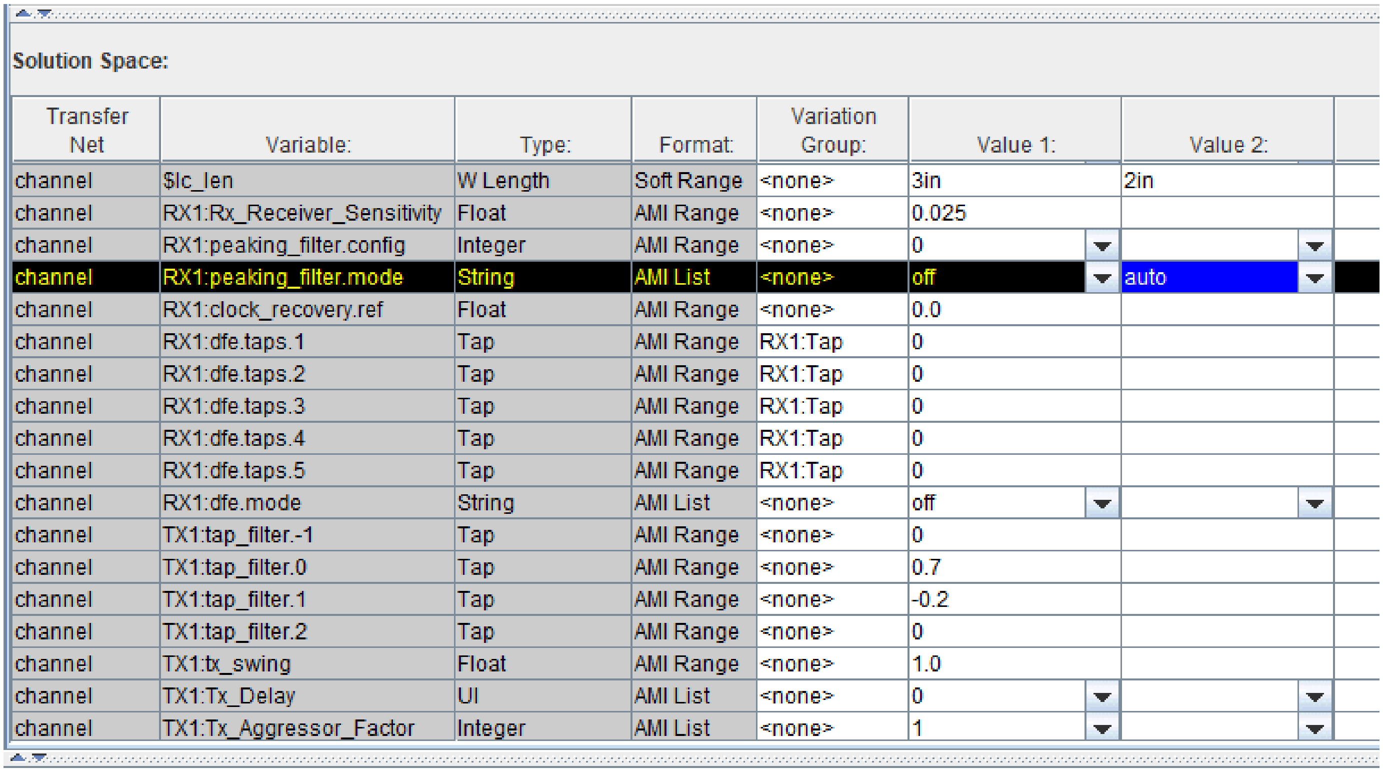
Waveform File Selection: ☑ Enable Multi-Select □ Create New Displays □ Begin X At Zero

Row	ID	Transfer Net	State	Transfer	s)	Stat Eye Height (V)	Stat Eye Margin (V)	Stat O
1	1	channel	default	TX1_to_RX1		0	-0.025	0.9453
2	2	channel	default	TX1_to_RX1		0	-0.025	0.6306
3	3	channel	default	TX1_to_RX1		0	-0.025	0.7737
4	4	channel	default	TX1_to_RX1		0	-0.025	0.9451
5	5	channel	default	TX1_to_RX1		0	-0.025	0.9450
6	6	channel	default	TX1_to_RX1		0	-0.025	0.9453
7	7	channel	default	TX1_to_RX1		0	-0.025	0.6020
8	8	channel	default	TX1_to_RX1		0	-0.025	0.7565
9	9	channel	default	TX1_to_RX1		0	-0.025	0.9451
10	10	channel	default	TX1_to_RX1		0	-0.025	0.9450
11	11	channel	default	TX1_to_RX1		0	-0.025	0.9453
12	12	channel	default	TX1_to_RX1		0	-0.025	0.5677
13	13	channel	default	TX1_to_RX1		0	-0.025	0.7355
14	14	channel	default	TX1_to_RX1		0	-0.025	0.9451
15	15	channel	default	TX1_to_RX1		0	-0.025	0.9450
16	16	channel	default	TX1_to_RX1		0	-0.025	0.9453
17	17	channel	default	TX1_to_RX1		0.000623239	-0.0246884	0.5258



Click on the **Stat BER** header to get the smallest BER (4.64e-10 in this example) at the top. To see the tap settings for the top row right-click on the row in the table and select **Show Solution Space**. In the dialog that appears you can see the tap settings: `TX1.tap_filter.0 = 0.7` and `TX1.tap_filter.1 = -0.2`.

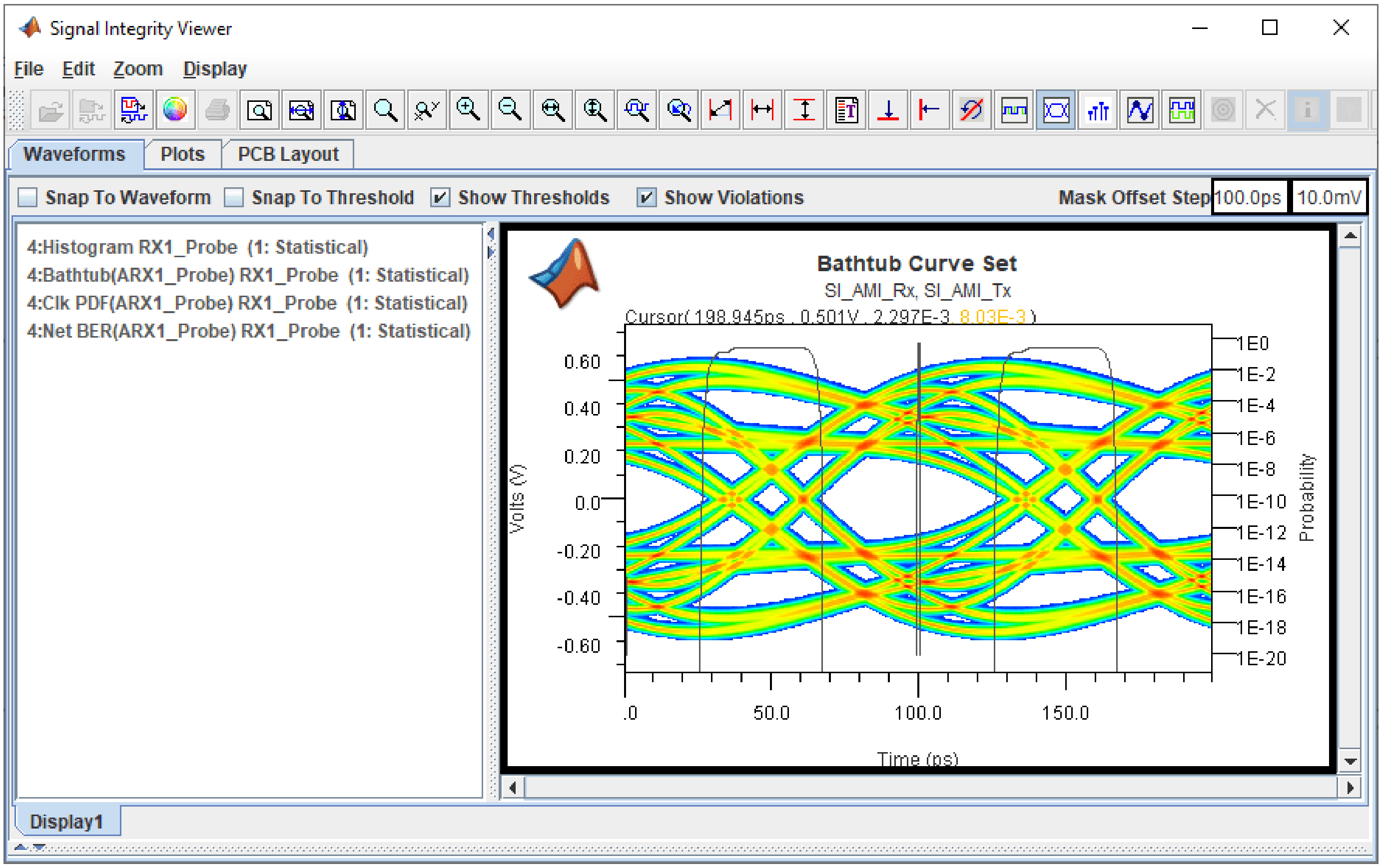
Go back to the **Serial Link Designer** app Solution Space panel. Change the TX equalizer taps to the values that gave the best BER from above (`0.0, 0.7, -0.2, 0.0`). Change **Value 2** for `RX1:peaking_filter.mode` to `Auto`.



Solution Space:

Transfer Net	Variable:	Type:	Format:	Variation Group:	Value 1:	Value 2:
channel	$lc_len	W Length	Soft Range	<none>	3in	2in
channel	RX1:Rx_Receiver_Sensitivity	Float	AMI Range	<none>	0.025	
channel	RX1:peaking_filter.config	Integer	AMI Range	<none>	0	
channel	RX1:peaking_filter.mode	String	AMI List	<none>	off	auto
channel	RX1:clock_recovery.ref	Float	AMI Range	<none>	0.0	
channel	RX1:dfe.taps.1	Tap	AMI Range	RX1:Tap	0	
channel	RX1:dfe.taps.2	Tap	AMI Range	RX1:Tap	0	
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	
channel	TX1:tap_filter.-1	Tap	AMI Range	<none>	0	
channel	TX1:tap_filter.0	Tap	AMI Range	<none>	0.7	
channel	TX1:tap_filter.1	Tap	AMI Range	<none>	-0.2	
channel	TX1:tap_filter.2	Tap	AMI Range	<none>	0	
channel	TX1:tx_swing	Float	AMI Range	<none>	1.0	
channel	TX1:Tx_Delay	UI	AMI List	<none>	0	
channel	TX1:Tx_Aggressor_Factor	Integer	AMI List	<none>	1	

Save the changes and rerun the simulation. Two of the four simulations now show positive statistical eye margin. Select one of the rows with a positive margin, right click and select **Show BER**. You can see the statistical eye, the bathtub curve and the clock PDF.



☐ Snap To Waveform ☐ Snap To Threshold ☑ Show Thresholds ☑ Show Violations Mask Offset Step 100.0ps 10.0mV

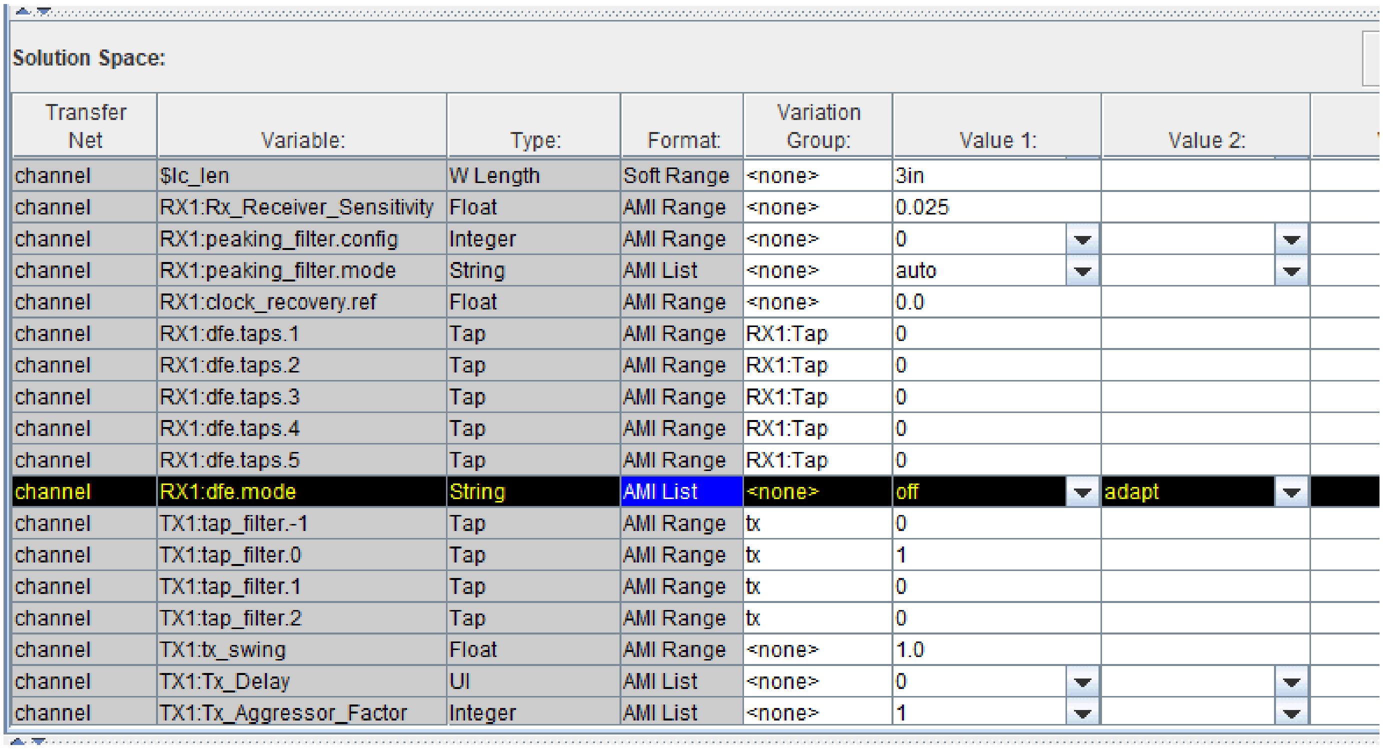
4:Histogram RX1_Probe (1: Statistical)
4:Bathtub(ARX1_Probe) RX1_Probe (1: Statistical)
4:Clk PDF(ARX1_Probe) RX1_Probe (1: Statistical)
4:Net BER(ARX1_Probe) RX1_Probe (1: Statistical)

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

Time Domain Analysis

The DFE adaptation behavior is non-LTI, so running time domain analysis will let you see how the DFE converges over time.

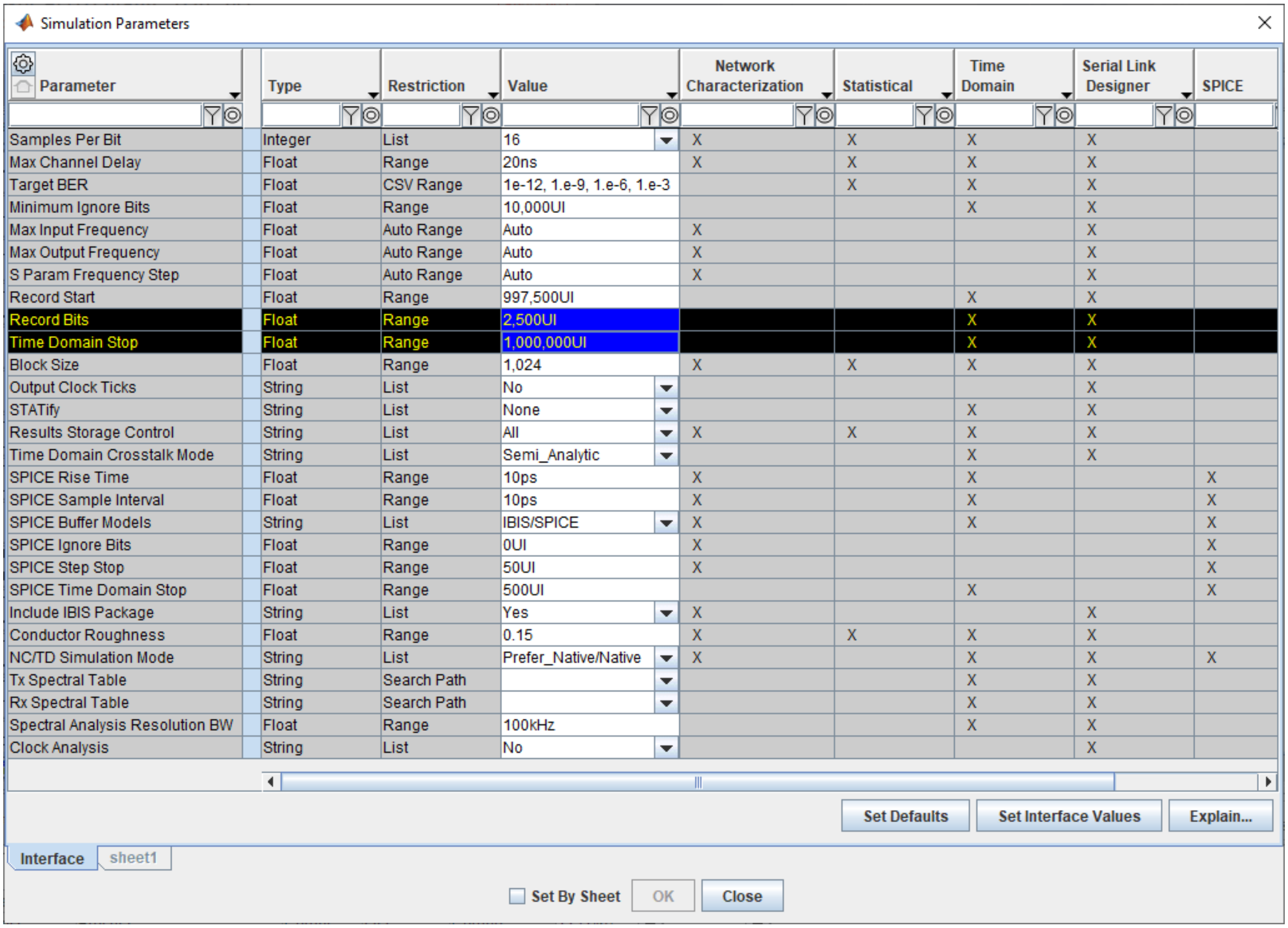
To set up the time domain analysis, in the Solution Space panel of the **Serial Link Designer** app, delete the **Value 2** (2in) of the `$lc_len` **Variable**. Set the values of **Variable** `RX1:peaking_filter.mode` **Value 1** to `auto` and **Value 2** to blank. Change the **Variation Group** of TX1 tap filters to `tx` and set the values of (`0, 1, 0, 0`). Set the **Value 2** to of `RX1:dfe.mode` **Variable** to `adapt`.



Solution Space:

Transfer Net	Variable:	Type:	Format:	Variation Group:	Value 1:	Value 2:
channel	$lc_len	W Length	Soft Range	<none>	3in	
channel	RX1:Rx_Receiver_Sensitivity	Float	AMI Range	<none>	0.025	
channel	RX1:peaking_filter.config	Integer	AMI Range	<none>	0	
channel	RX1:peaking_filter.mode	String	AMI List	<none>	auto	
channel	RX1:clock_recovery.ref	Float	AMI Range	<none>	0.0	
channel	RX1:dfe.taps.1	Tap	AMI Range	RX1:Tap	0	
channel	RX1:dfe.taps.2	Tap	AMI Range	RX1:Tap	0	
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	adapt
channel	TX1:tap_filter.-1	Tap	AMI Range	tx	0	
channel	TX1:tap_filter.0	Tap	AMI Range	tx	1	
channel	TX1:tap_filter.1	Tap	AMI Range	tx	0	
channel	TX1:tap_filter.2	Tap	AMI Range	tx	0	
channel	TX1:tx_swing	Float	AMI Range	<none>	1.0	
channel	TX1:Tx_Delay	UI	AMI List	<none>	0	
channel	TX1:Tx_Aggressor_Factor	Integer	AMI List	<none>	1	

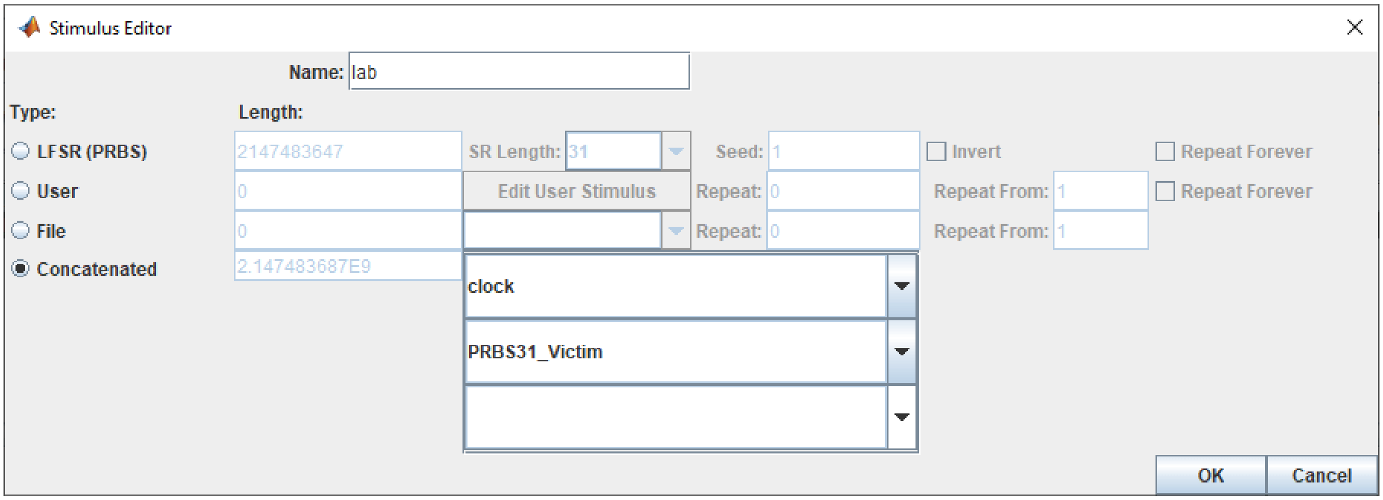
Select **Setup** > **Simulation Parameters** and check that the **Time Domain Stop** is set to `1,000,000 UI` and the **Record Bits** is set to `2,500 UI`. Right-click on the RX symbol on the schematic and select **Edit AMI File(s)**. In the AMI file that opens the **Ignore_Bits** parameter is set to `500,000 UI`. The largest value of either the AMI file parameter **Ignore_Bits** or the **Simulation Parameters** setting for **Ignore_Bits** is used during simulation. In this case, the value of 500,000 UI from the AMI file will be used instead of the value of 10,000 UI in Simulation Parameters. This group of settings configures the simulation to run for one million UI. The last 500,000 UI is used for the persistent eye and the BER and the last 2500 UI of the waveform is saved.



Simulation Parameters

Parameter	Type	Restriction	Value	Network Characterization	Statistical	Time Domain	Serial Link Designer	SPICE
Samples Per Bit	Integer	List	16	X	X	X	X	
Max Channel Delay	Float	Range	20ns	X	X	X	X	
Target BER	Float	CSV Range	1e-12, 1.e-9, 1.e-6, 1.e-3		X	X	X	
Minimum Ignore Bits	Float	Range	10,000UI			X	X	
Max Input Frequency	Float	Auto Range	Auto	X			X	
Max Output Frequency	Float	Auto Range	Auto	X			X	
S Param Frequency Step	Float	Auto Range	Auto	X			X	
Record Start	Float	Range	997,500UI			X	X	
Record Bits	Float	Range	2,500UI			X	X	
Time Domain Stop	Float	Range	1,000,000UI			X	X	
Block Size	Float	Range	1,024	X	X	X	X	
Output Clock Ticks	String	List	No				X	
STATify	String	List	None			X	X	
Results Storage Control	String	List	All	X	X	X	X	
Time Domain Crosstalk Mode	String	List	Semi_Analytic			X	X	
SPICE Rise Time	Float	Range	10ps	X		X		X
SPICE Sample Interval	Float	Range	10ps	X		X		X
SPICE Buffer Models	String	List	IBIS/SPICE	X		X		X
SPICE Ignore Bits	Float	Range	0UI	X				X
SPICE Step Stop	Float	Range	50UI	X				X
SPICE Time Domain Stop	Float	Range	500UI			X		X
Include IBIS Package	String	List	Yes	X			X	
Conductor Roughness	Float	Range	0.15	X	X	X	X	
NC/TD Simulation Mode	String	List	Prefer_Native/Native	X		X	X	X
Tx Spectral Table	String	Search Path				X	X	
Rx Spectral Table	String	Search Path				X	X	
Spectral Analysis Resolution BW	Float	Range	100kHz			X	X	
Clock Analysis	String	List	No				X	

Double click on the TX symbol on the schematic to launch the Designator Element Properties dialog box. Click on the **Stimulus** button to open the Stimuli dialog box. Click on **New** button to create a new stimulus. Set the **Name** to `lab` and **Type** to `Concatenated`. Make it a concatenated stimulus that is `clock` followed by `PRBS31_Victim`. Save the changes.



Stimulus Editor

Name: lab

Type: Length:

○ LFSR (PRBS) 2147483647 SR Length: 31 Seed: 1 ☐ Invert ☐ Repeat Forever

○ User 0 Edit User Stimulus Repeat: 0 Repeat From: 1 ☐ Repeat Forever

○ File 0 Repeat: 0 Repeat From: 1

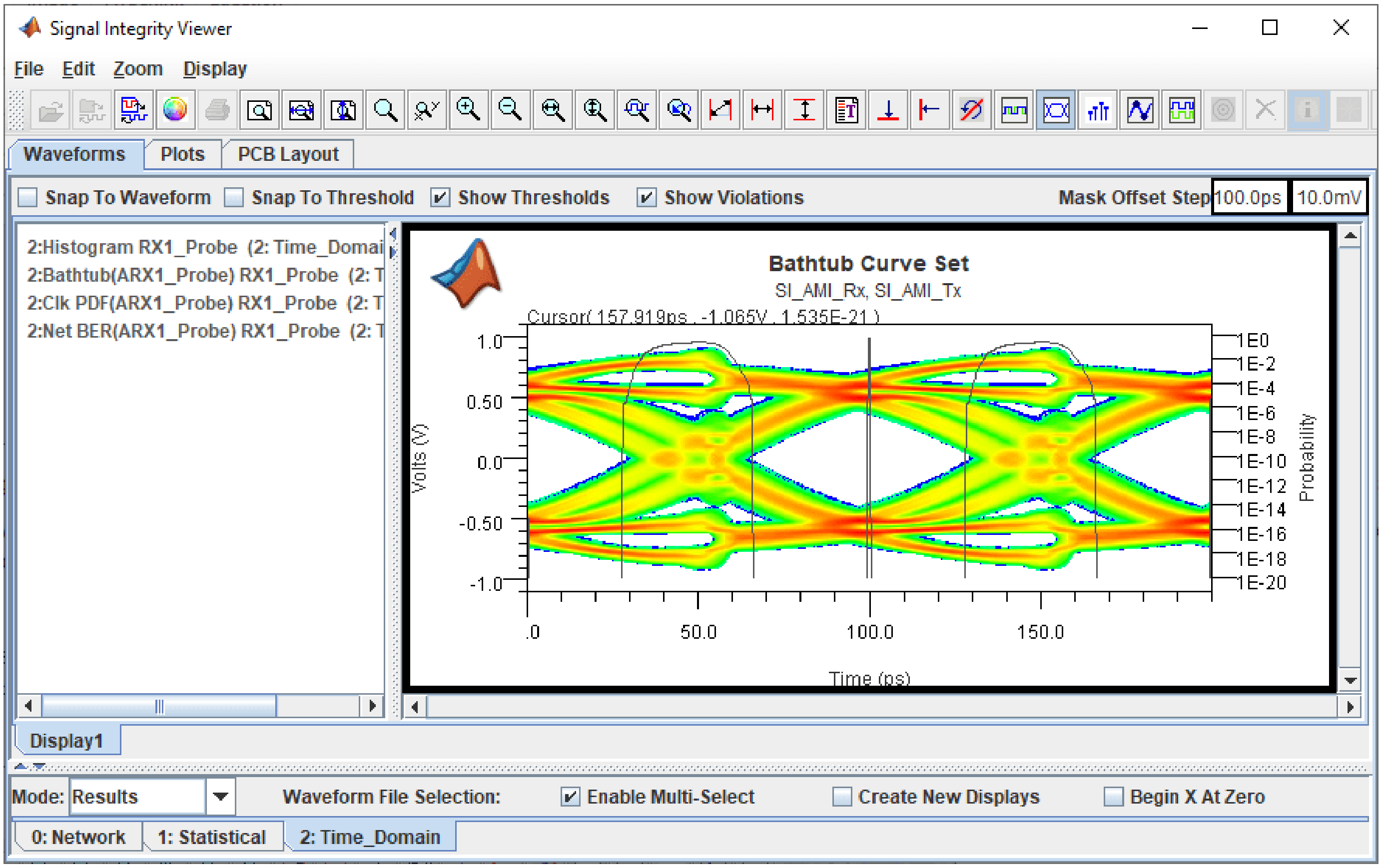
● Concatenated 2.147483687E9

clock

PRBS31_Victim

On the Designator Element Properties dialog box, select the **Stimulus** as `lab`. Save the changes. Run the simulation and select Include Time Domain Analysis in the Prelayout Channel Analysis dialog box.

The **Signal Integrity Viewer** app launches when the simulation is complete. Select the Time_Domain tab and right click on the result rows and select **Show Solution Space** to see which row is showing the result of the DFE adapt mode. Select the row corresponding to the DFE adapt mode, right click and select **Show BER**.



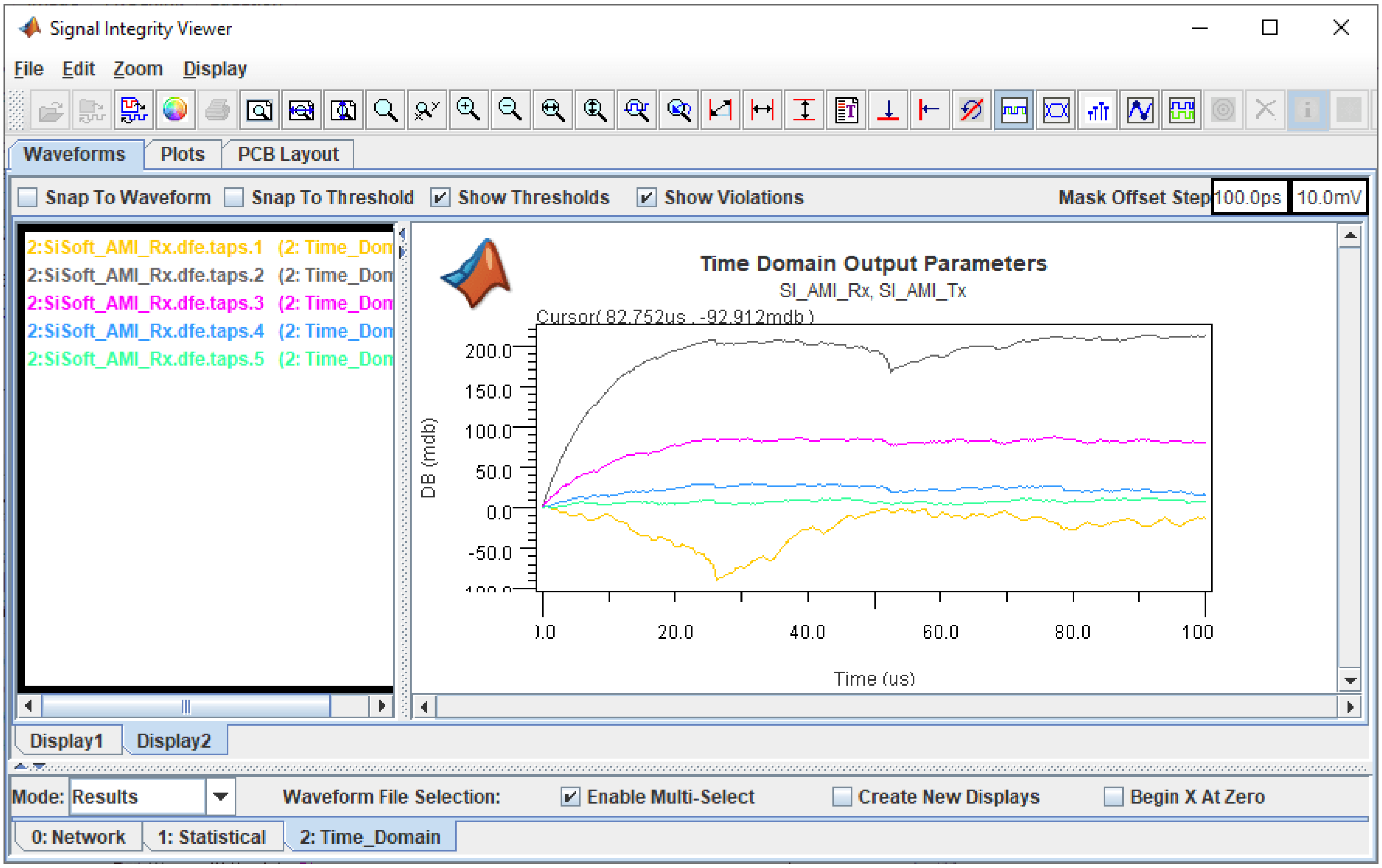
□ Snap To Waveform □ Snap To Threshold ☑ Show Thresholds ☑ Show Violations Mask Offset Step 100.0ps 10.0mV

2:Histogram RX1_Probe (2: Time_Domai
2:Bathtub(ARX1_Probe) RX1_Probe (2: T
2:Clk PDF(ARX1_Probe) RX1_Probe (2: T
2:Net BER(ARX1_Probe) RX1_Probe (2: T

Mode: Results Waveform File Selection: ☑ Enable Multi-Select □ Create New Displays □ Begin X At Zero

0: Network 1: Statistical 2: Time_Domain

Right click on the Display panel and add a new display. On the Time_Domain tab right-click on the results row for the DFE Adapt simulation and select **Show IBIS-AMI Output Parameters > RX1_SiSoft_AMI_Rx**. Delete the nodes that are not DFE taps and zoom to view the tap coefficients over time as they adapt.



□ Snap To Waveform □ Snap To Threshold ☑ Show Thresholds ☑ Show Violations Mask Offset Step 100.0ps 10.0mV

2:SiSoft_AMI_Rx.dfe.taps.1 (2: Time_Don
2:SiSoft_AMI_Rx.dfe.taps.2 (2: Time_Don
2:SiSoft_AMI_Rx.dfe.taps.3 (2: Time_Don
2:SiSoft_AMI_Rx.dfe.taps.4 (2: Time_Don
2:SiSoft_AMI_Rx.dfe.taps.5 (2: Time_Don

Mode: Results Waveform File Selection: ☑ Enable Multi-Select □ Create New Displays □ Begin X At Zero

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

Creating Compliance Masks in Serial Link Designer

Compliance masks can be created and applied to simulation results in Serial Link Designer. A mask to be used in Serial Link Designer is defined in a file called a “rules file”. Rules files are text files that define the mask limits for a parameter, or parameters in the channel simulation results.

This example shows how you can create compliance masks and apply them to the simulation results in the **Serial Link Designer** app. The masks used in the **Serial Link Designer** app are defined in the *rules file*. Rules files are text files that define the mask limits for a parameter, or parameters in the channel simulation results. These are user created files that can be applied as Eye Masks or NetworkLoss Characteristics. The file extension of a rules file must be “.rules” for Serial Link Designer to recognize it as a mask file. It also must be located in the “<Project_Directory>\si_lib \rules\” directory of the project.

Creating Masks for Compliance Tests

To create and apply rules files, follow the four step process:

1. Create a rules file that describes the compliance test limits.
2. Modify Transfer Net properties to apply the rule to a Transfer Net.
3. Run the simulation. The margin to the applied rules will be displayed in the Channel Analysis report and can also be seen in the SiViewer when simulation results are loaded.
4. Plot the compliance test results in the viewer.

Rules File Creation

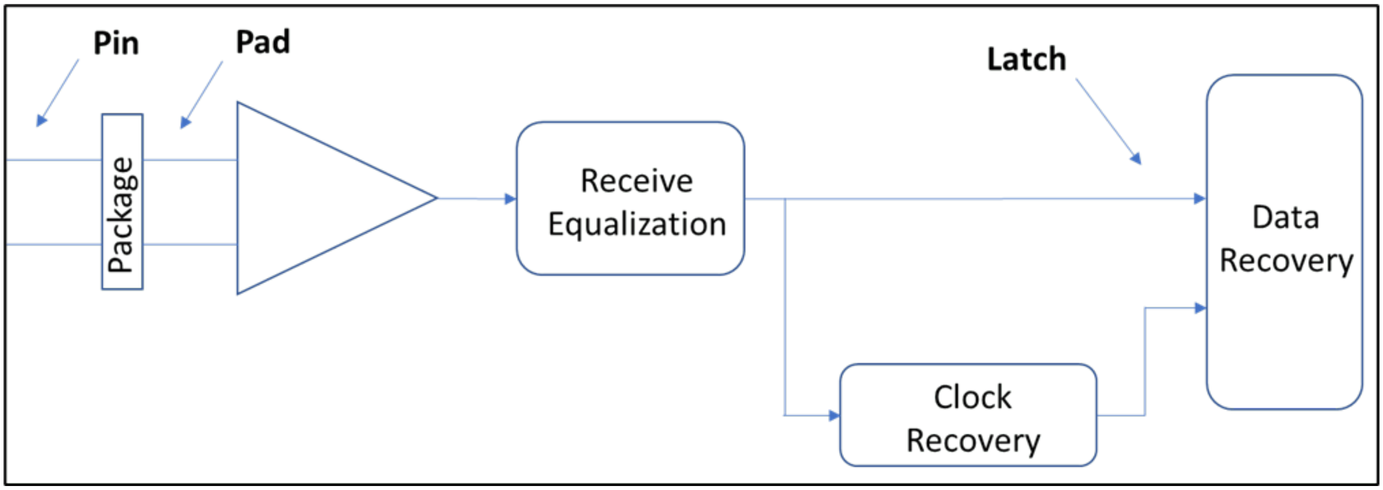
There are four keywords in the rules file that define the characteristics of the test:

1. First is the keyword `“[Rule]”`; this defines the name of this rule that can be applied to the simulation. This can be any ASCII string up to the first terminator character (space, TAB, or comma).
2. “`Method`” defines the test to be performed, for example: checking Insertion Loss.
3. "`Apply_To`" tells the simulator where to apply the mask (Pad, Pin, Latch).
4. The “`Mask_Data`” keyword introduces an arbitrary length section of data records that defines the measurement limits.

Rule File Creation for Eye Mask

A common compliance test is a measurement of margin to an eye mask. This generally involves an inner eye mask (measuring the open area of the eye) and sometimes an outer eye mask (measuring maximum voltage excursion). The various probe points that need to be defined in the rules file are shown. There is the package 'pin', the die 'pad' and the 'latch'. Note that the probe point ‘latch’, also called the ‘data decision point’, is the primary and default measurement point for all built-in Serial Link Designer Eye Mask tests. In the **Signal Integrity Viewer** app, when a result is selected from the drop-down menu items such as ‘**Show BER**’, ‘**Show Statistical Eye**’, the result in view is always the data at the Latch. If data has been kept at the ‘Pin’ or ‘Pad’ probe points, it is available in the drop-down menu under ‘Show Probed BER’, and so forth. If both pin and pad have been probed, due to multiple tests, then both results show up under "Probed". You must take care in this case to identify which result is at the pin and which is at the pad.

Example for probe points:



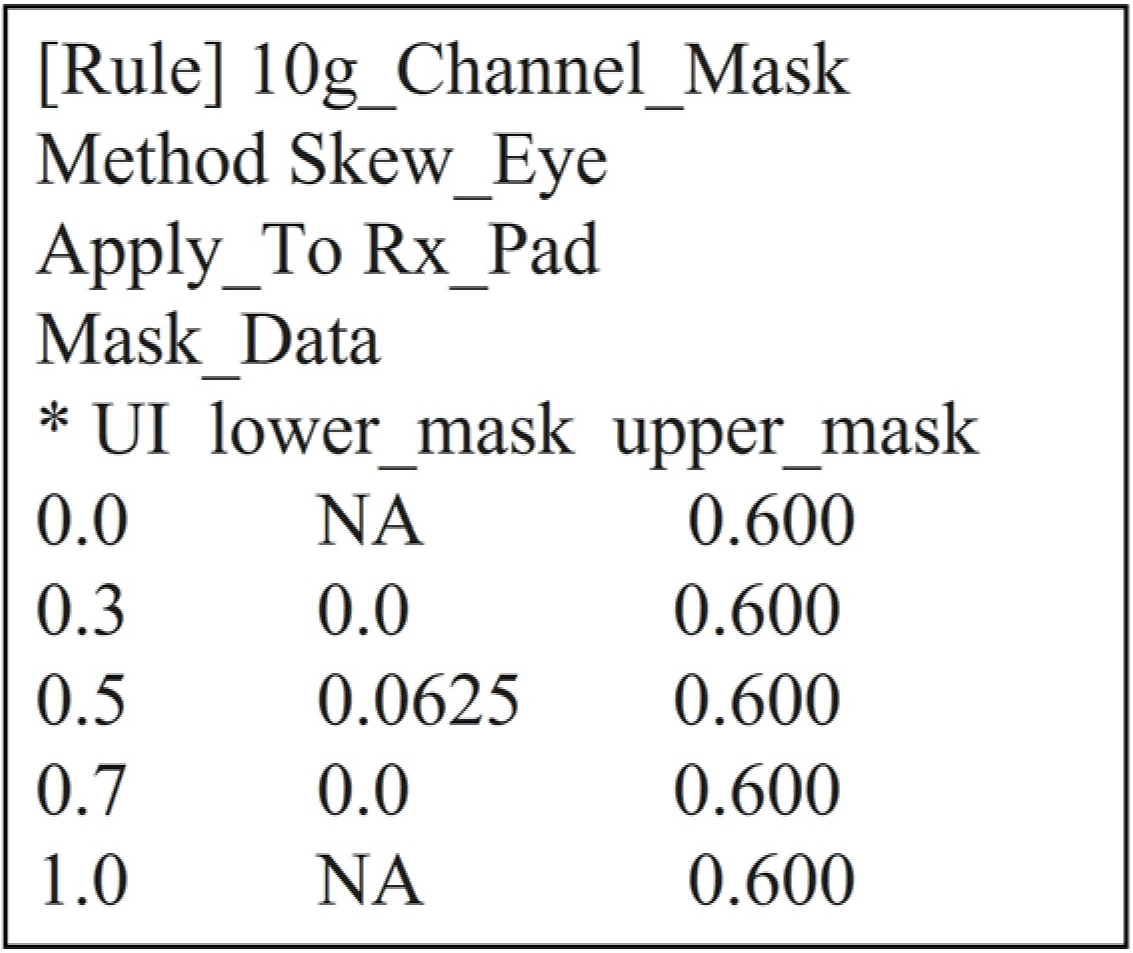
For testing eye masks, the 'Mask_Data' records are of the form:

```
UI_fraction inner_mask outer_mask
```

The UI_fraction is a number from 0 to 1, indicating the point in the Unit Interval (UI) where the values are to be applied. The inner and outer mask values are the voltage limits at that point. In the rules file shown below, the mask data has five data points. These occur at the beginning and end of the unit interval (UI=0.00, UI=1.00), as well as at the 30%, 50% and 70% point in the unit interval.

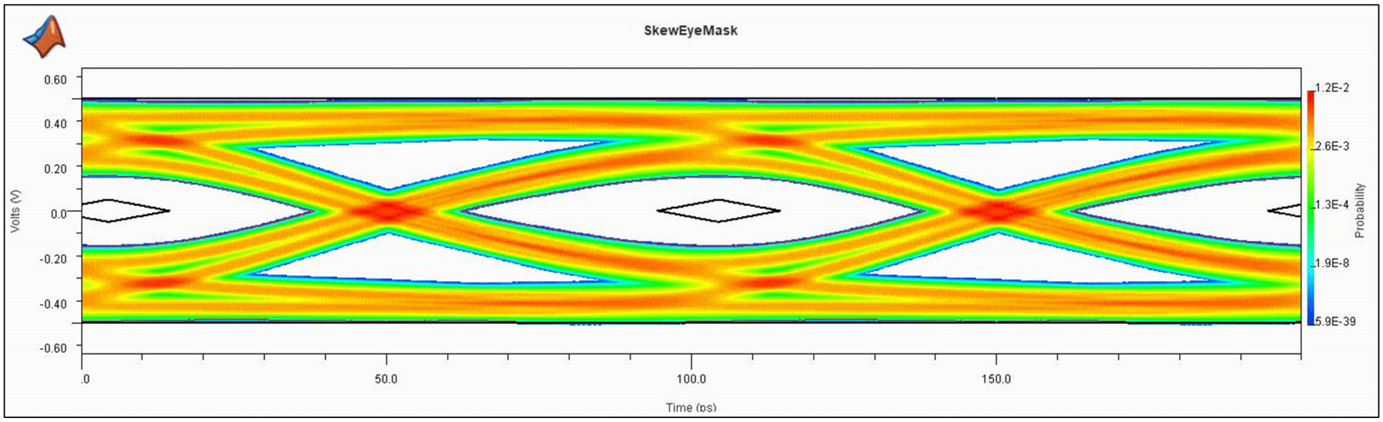
The numbers specified in the Mask_Data are positive numbers only, but the equivalent negative numbers are implied, forming a complete mask that is symmetrical about the 0V axis. Eye mask rules are applied to Statistical and Time Domain results, not Network. So, the rule is only present in the menu when viewing Statistical or Time Domain tabs in the viewer, not when viewing the Network tab.

Example eye mask rule looks like:



```
[Rule] 10g_Channel_Mask
Method Skew_Eye
Apply_To Rx_Pad
Mask_Data
* UI  lower_mask  upper_mask
0.0       NA          0.600
0.3       0.0         0.600
0.5       0.0625      0.600
0.7       0.0         0.600
1.0       NA          0.600
```

The example rule is applied at the ‘`Rx_Pad'`, so the relevant eye diagram is the Statistical Eye, at the pad. The statistical eye diagram with an eye mask applied looks like.



Enabling Probe Points

When running a simulation without any rules files applied, Serial Link Designer only reports data at the latch. The presence of any rules file test in the project automatically triggers **Serial Link**

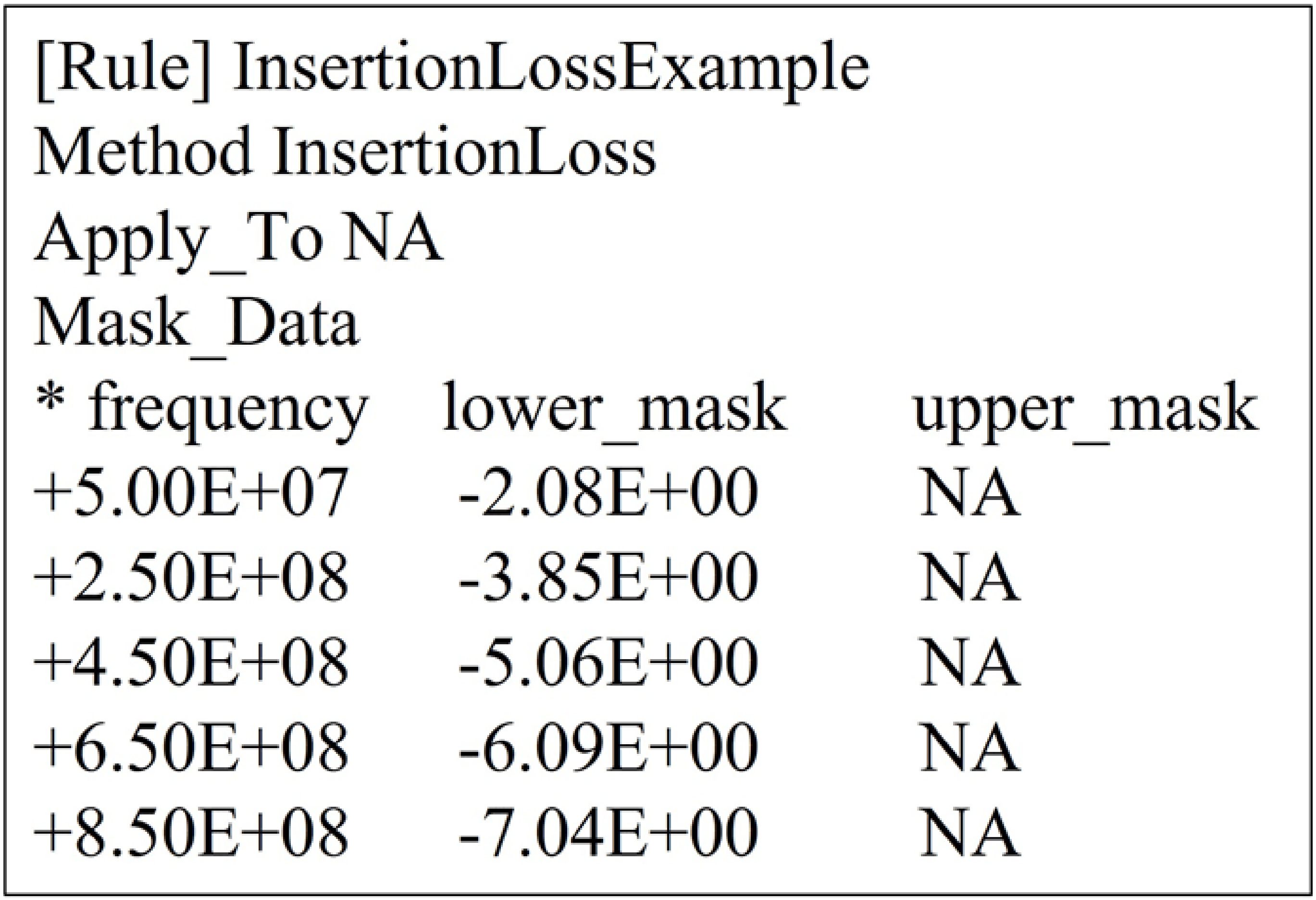
Designer to retain data at the probe point defined by the 'Apply_To' keyword in the rule. It looks for (Pin, Pad or latch). So for the receiver, the probe point are **Rx_pin, Rx_pad or Rx_latch**.

Rule File Creation for Network Insertion Loss

You can define network characterization masks for minimum, maximum and min/max limits.

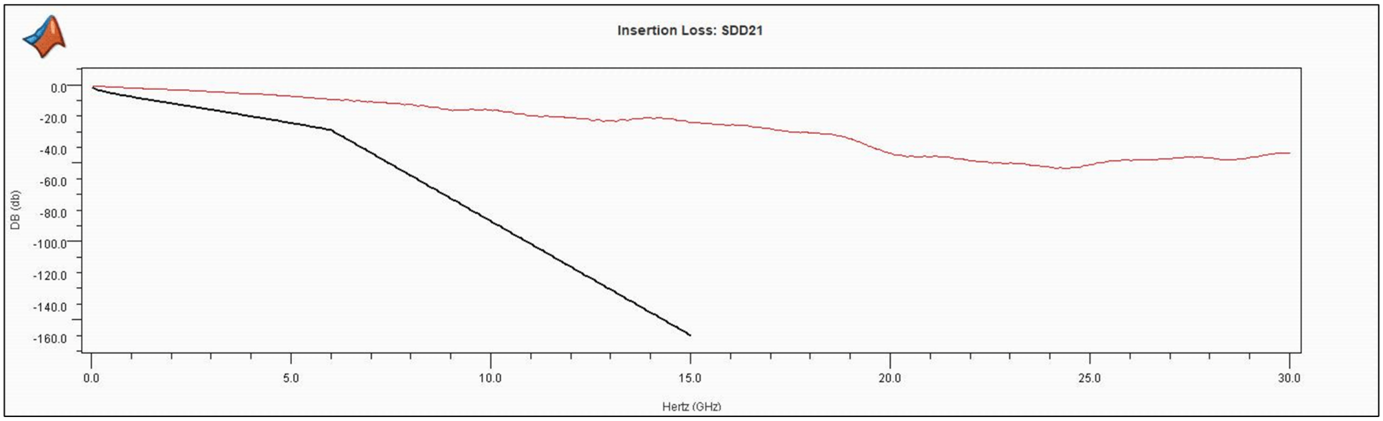
Testing for Minimum Limit

In this case, the simulated result is tested against a lower bound only. The test for insertion loss (SDD21). For all s-parameter rules, the form of the 'Mask_Data' records contain the frequency, lower_limit and upper_limit. The frequency is in hertz (Hz) and limits are in decibels (dB). When the upper limit is not applicable for a given test, the last column is left blank, or alternatively filled with "**NA**". The lines beginning with an asterisk are comment lines.



```
[Rule] InsertionLossExample
Method InsertionLoss
Apply_To NA
Mask_Data
* frequency   lower_mask     upper_mask
+5.00E+07      -2.08E+00      NA
+2.50E+08      -3.85E+00      NA
+4.50E+08      -5.06E+00      NA
+6.50E+08      -6.09E+00      NA
+8.50E+08      -7.04E+00      NA
```

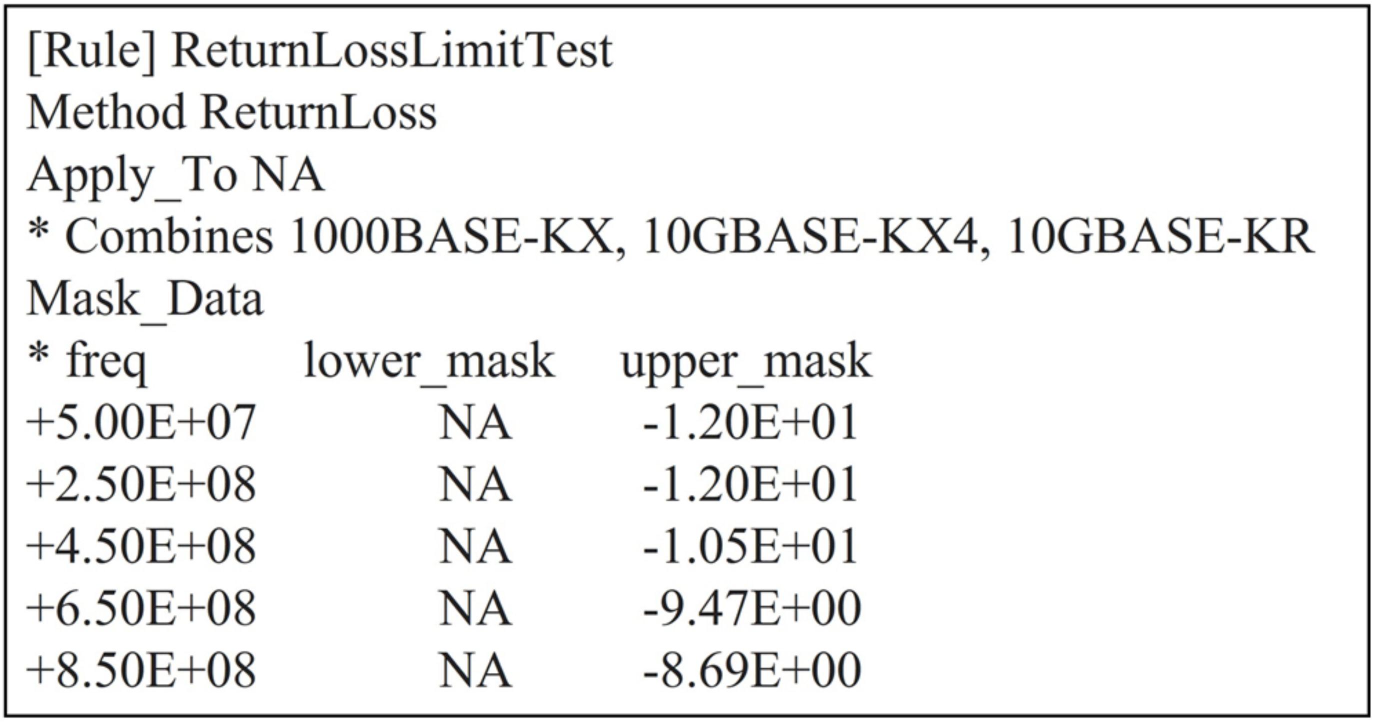
Compare the plot of the insertion loss of a simulated path (red) with the applied insertion loss mask (black).



As the test has only lower limit data, there is only a single black line indicating the limit. In this case the simulated data is always well above the line; if there were a case where it dropped below the line, the results table in Serial Link Designer would identify a rules violation. As a note, this example shows only a single rule defined in this file, a rules file may contain multiple rule definitions. This will be covered later in this example.

Testing for Maximum Limit

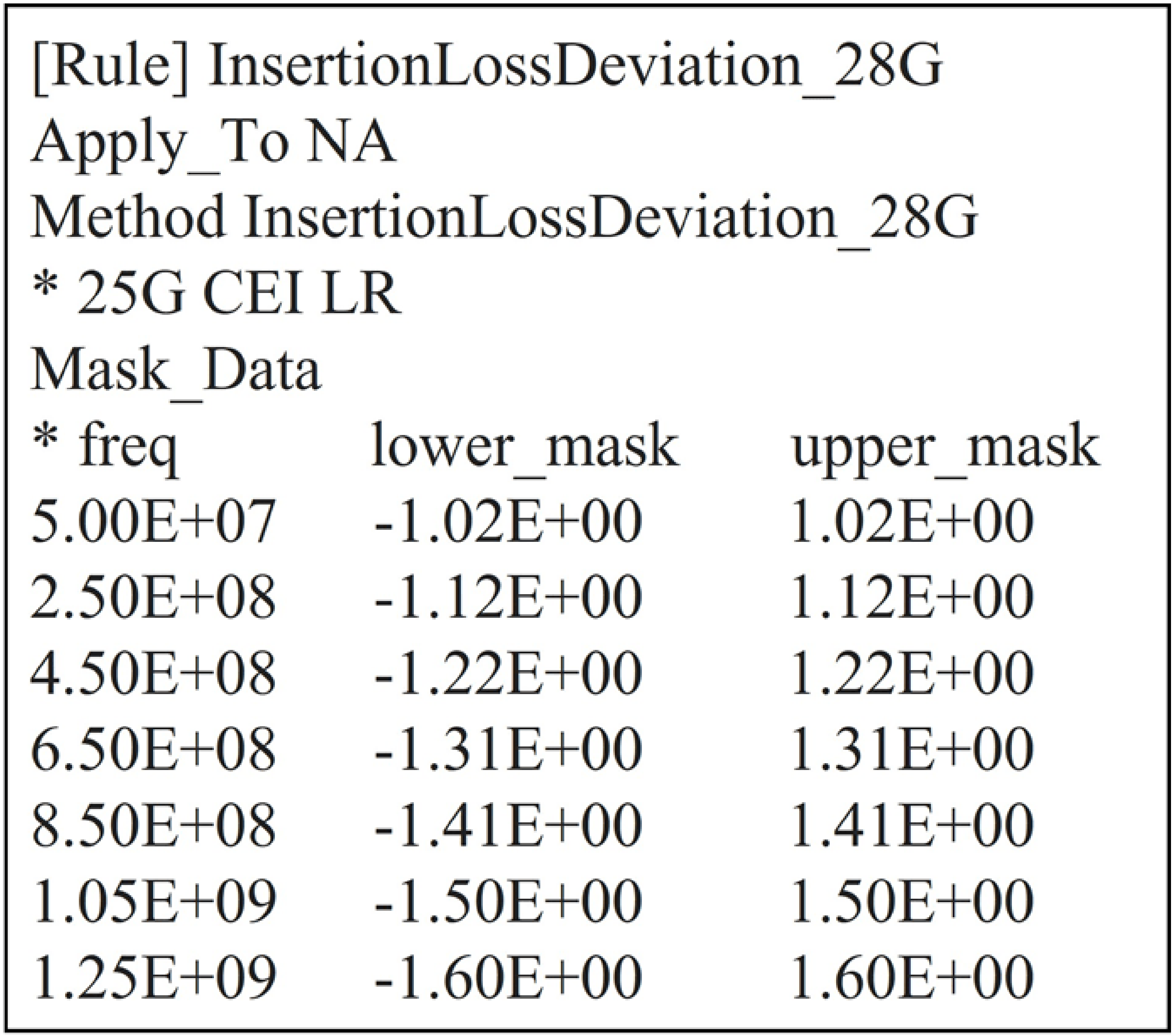
The procedure for testing maximum limits is similar to testing for minimum limit. The only difference is in the'`Mask_Data`' records in the rules file.



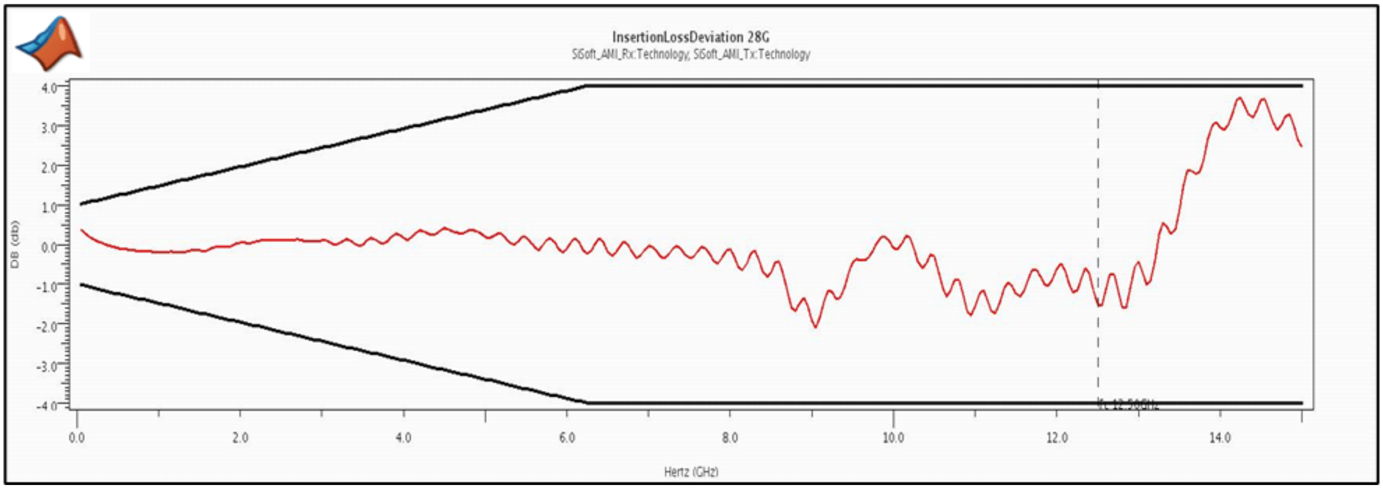
```
[Rule] ReturnLossLimitTest
Method ReturnLoss
Apply_To NA
* Combines 1000BASE-KX, 10GBASE-KX4, 10GBASE-KR
Mask_Data
* freq          lower_mask    upper_mask
+5.00E+07             NA        -1.20E+01
+2.50E+08             NA        -1.20E+01
+4.50E+08             NA        -1.05E+01
+6.50E+08             NA        -9.47E+00
+8.50E+08             NA        -8.69E+00
```

Testing for Minimum and Maximum Limit

An example of testing both minimum and maximum limits is Insertion Loss Deviation_28G (ILD). The rule and resulting plot is shown.

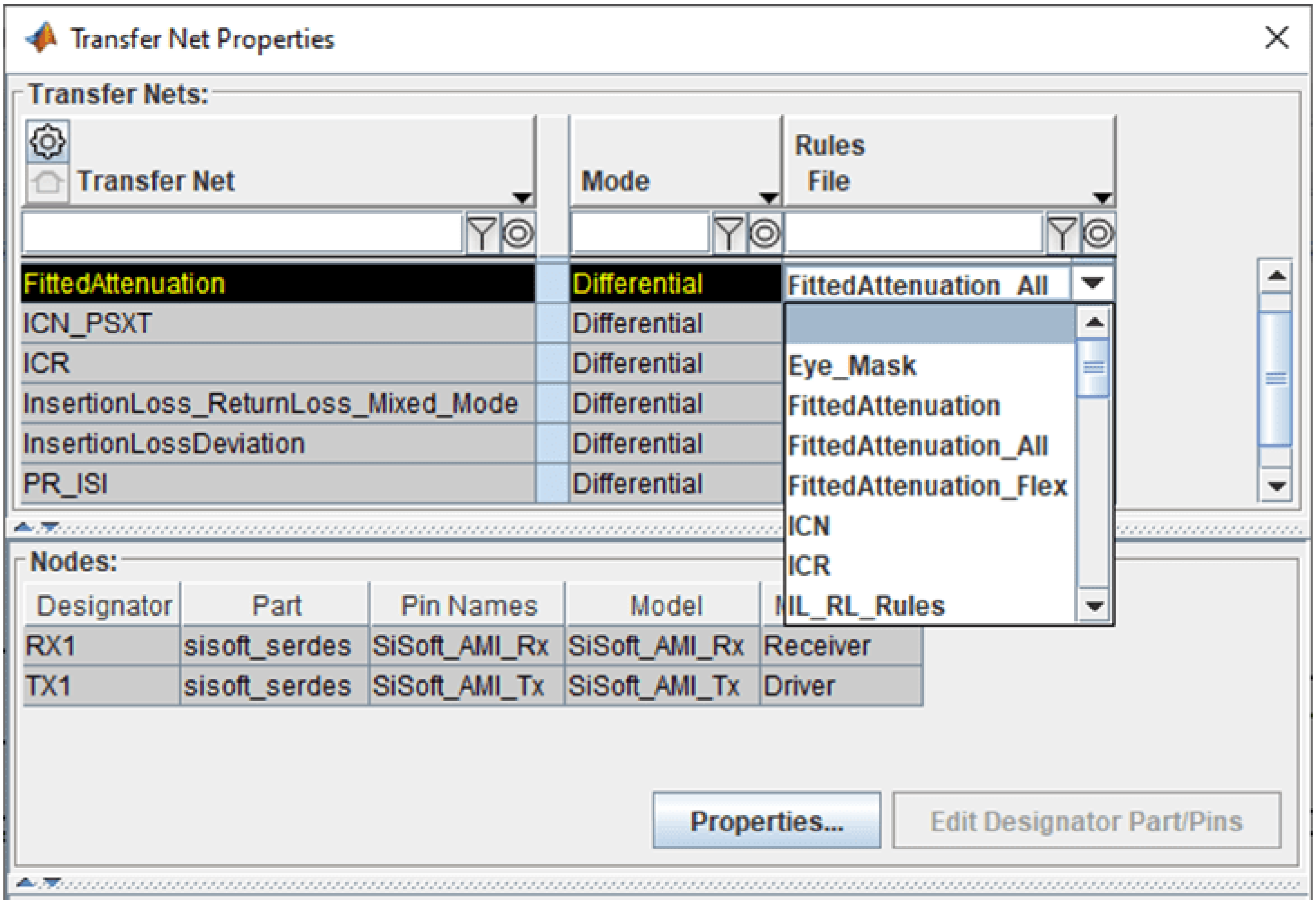


```
[Rule] InsertionLossDeviation_28G
Apply_To NA
Method InsertionLossDeviation_28G
* 25G CEI LR
Mask_Data
* freq        lower_mask     upper_mask
5.00E+07      -1.02E+00      1.02E+00
2.50E+08      -1.12E+00      1.12E+00
4.50E+08      -1.22E+00      1.22E+00
6.50E+08      -1.31E+00      1.31E+00
8.50E+08      -1.41E+00      1.41E+00
1.05E+09      -1.50E+00      1.50E+00
1.25E+09      -1.60E+00      1.60E+00
```



Adding Rules File to Transfer Net

To add a rules file to a transfer net select the “Transfer Net Properties” tab or (setup->tnet properties).



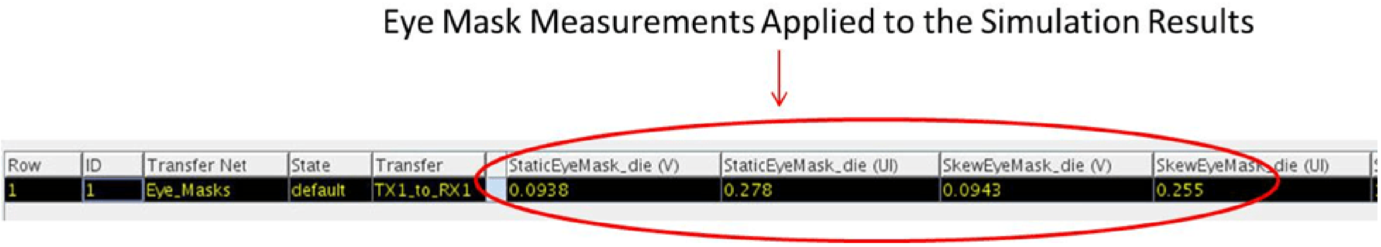
Transfer Net		Mode	Rules File
FittedAttenuation		Differential	FittedAttenuation_All
ICN_PSXT		Differential	
ICR		Differential	
InsertionLoss_ReturnLoss_Mixed_Mode		Differential	
InsertionLossDeviation		Differential	
PR_ISI		Differential	

Designator	Part	Pin Names	Model	
RX1	sisoft_serdes	SiSoft_AMI_Rx	SiSoft_AMI_Rx	Receiver
TX1	sisoft_serdes	SiSoft_AMI_Tx	SiSoft_AMI_Tx	Driver

Once you define the rules in a file and apply to the Transfer Net, you can simulate any extended nets associated with that Transfer Net. The mask defined in the rules file is compared with the simulation results to calculate the margin with respect to the mask.

Applying a Mask

After you run the simulations and the **Signal Integrity Viewer** app loads the ded, a column consisting of the reported margin to the mask is visible for each of the eye mask measurements (height (V) and width (UI or pS). The number reported indicates the margin to the defined mask for the transfer. The margin reported is the smallest distance between the actual eye contour and the mask, minus the limit set in the rule. For an eye mask, there is only a single value listed in the results table; it is the *smaller* of the inner and outer margins (if both are defined in the rules file). If either or both violates the margin, the smaller of the two is shown in red, as a negative number.



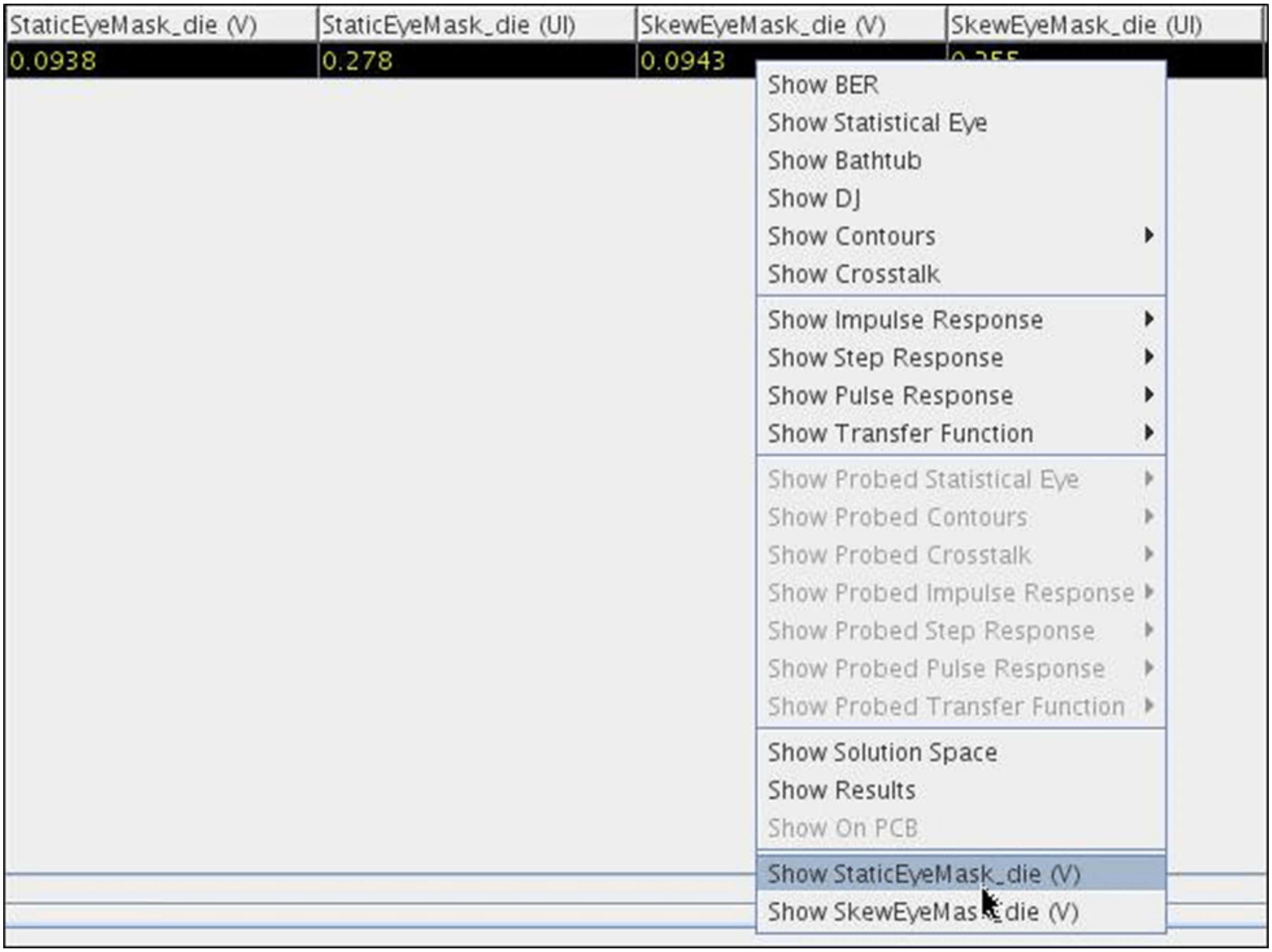
Eye Mask Measurements Applied to the Simulation Results

Row	ID	Transfer Net	State	Transfer		StaticEyeMask_die (V)	StaticEyeMask_die (UI)	SkewEyeMask_die (V)	SkewEyeMask_die (UI)
1	1	Eye_Masks	default	TX1_to_RX1		0.0938	0.278	0.0943	0.255

To display a mask to a simulation result:

1. In the **Signal Integrity Viewer** app, load the simulation results if they have not been automatically loaded after the simulation.
2. Select the simulation result row, or rows, that you wish to plot (multiple selections are allowed).
3. Right click on the row and select the rules file name at the bottom of the drop down menu.
4. Eye mask is applied and shown in the viewer in comparison to the eye contours defined for minimum BER set in the project or for the transfer (To set target BER, select: **Setup> Simulation Parameters** in the **Serial Link Designer** app).
5. You can also apply the mask to an eye diagram by plotting the statistical or persistent eye and then selecting the mask. This can also be done by selecting "**Show BER**".
6. Insertion Loss masks are plotted the same way, but are applied in the network characterization results in the **Signal Integrity Viewer** app.

You can apply the StaticEyeMask rule of transfer net.



StaticEyeMask_die (V)	StaticEyeMask_die (UI)	SkewEyeMask_die (V)	SkewEyeMask_die (UI)
0.0938	0.278	0.0943	0.255

The inner and outer contours of the eye diagram are shown along with the eye mask. The margin areas based on the calculated margins are shown for reference. If only an inner or outer mask is defined in the rules file then that is the only mask applied.

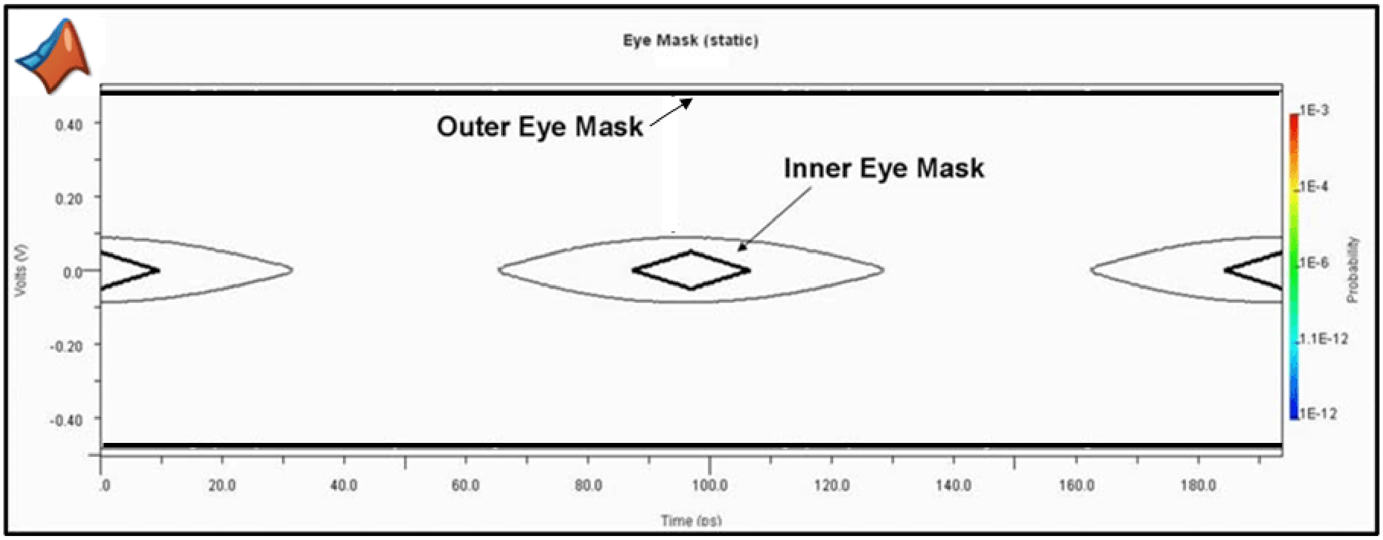
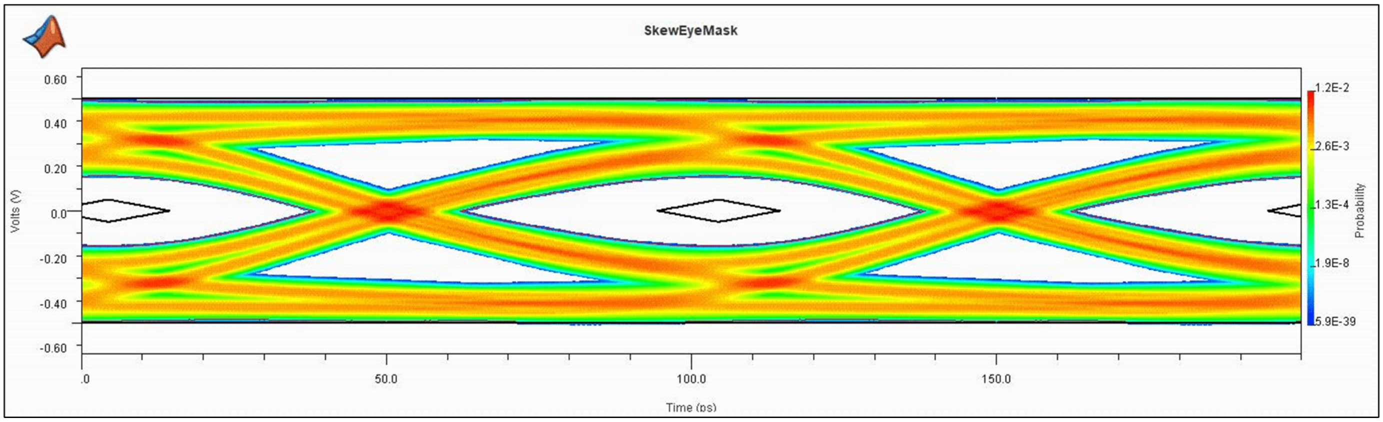


Figure 13: Plotted Eye Mask and Eye Contour of the Plotted Simulation Result.

You can also apply a mask to the eye diagram by plotting the eye and then selecting the mask.



Included Files

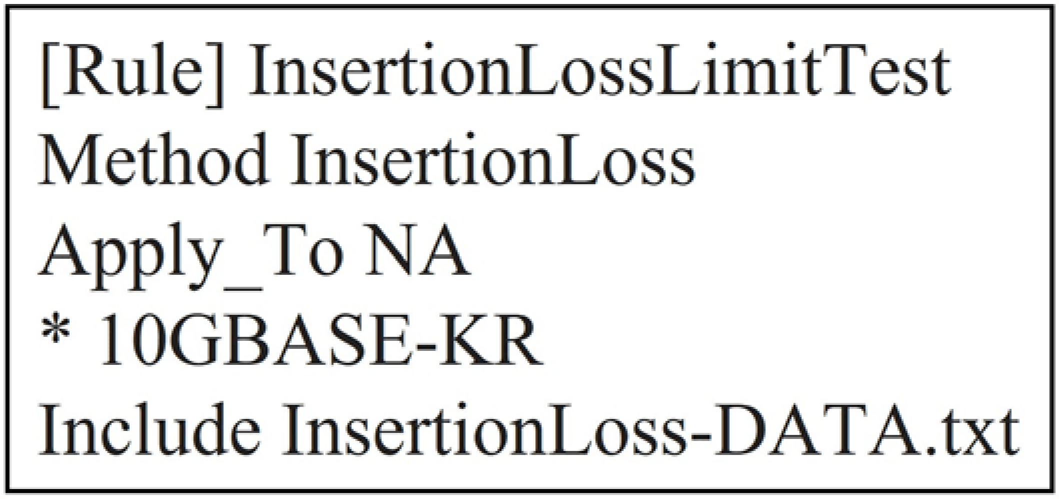
At any point within a rules file, you can place an ‘`include`’ statement in the form '**Include rules_file_basename'**, or '**Include any_file_fullname'.** The contents of the named file is then incorporated at the point of the `Include` statement. In the case of a normal rules file, **Serial Link Designer** automatically appends the ‘.rules’ suffix, so it does not need to be present in the rules file base name given. If any ‘.’ character appears in the given filename, it is then taken to be a general file, with a full name given (e.g. ‘mask_data_records.txt’) and the ‘.rules’ suffix is not appended.

You can have one ‘top-level’ rules file, which includes other rules files. Here is the content of one such file, named Mask_Set.rules:

```
Include Channel_Mask_Pin

Include Channel_Mask_Pad

Include Channel_Mask_Latch
Include InsertionLoss1
```

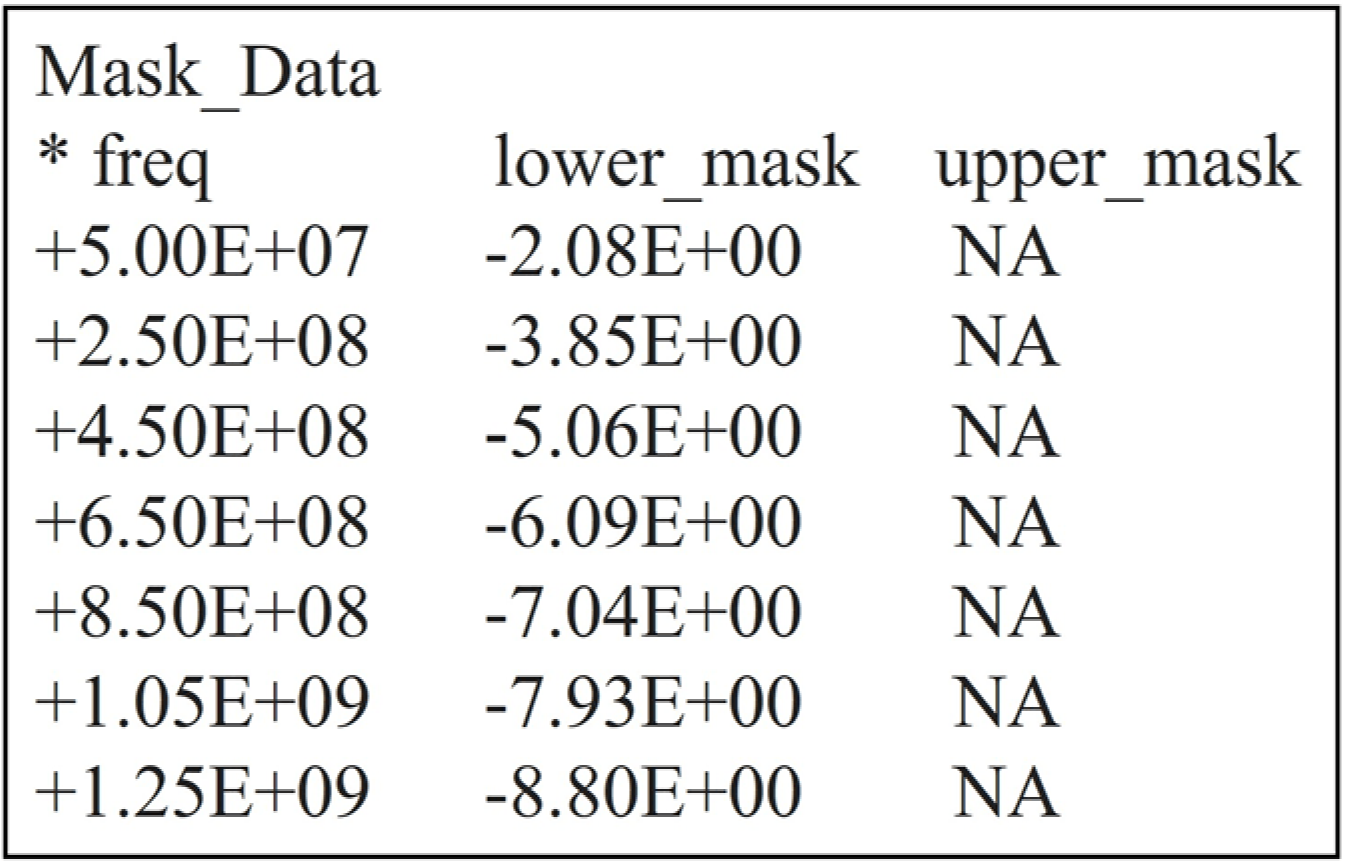
If a Transfer Net property is set to apply ‘`Mask_Set’`, it is applied to all the individual rules enumerated in the top-level file. This is a good way to apply a complete set of compliance rules for a protocol specification. You can also use the include statement to incorporate data from a separate file. The data may be in its own file for organizational reasons, or to make sharing of common data easier, or any other reason.

In the above case, the file ‘`InsertionLoss1.rules`’ is included by the top-level file ‘`Mask_set.rules’`. An example of the `InsertionLossLimitTest.rules` is shown:



```
[Rule] InsertionLossLimitTest
Method InsertionLoss
Apply_To NA
* 10GBASE-KR
Include InsertionLoss-DATA.txt
```

Here the rule is defined and applied, but the data defining the curve exists in a separate file ‘InsertionLoss-DATA.txt’ :.



```
Mask_Data
* freq          lower_mask   upper_mask
+5.00E+07      -2.08E+00      NA
+2.50E+08      -3.85E+00      NA
+4.50E+08      -5.06E+00      NA
+6.50E+08      -6.09E+00      NA
+8.50E+08      -7.04E+00      NA
+1.05E+09      -7.93E+00      NA
+1.25E+09      -8.80E+00      NA
```

Now you can create the mask data (which may be fairly extensive) separately, shared among multiple rules files, and updated easily.

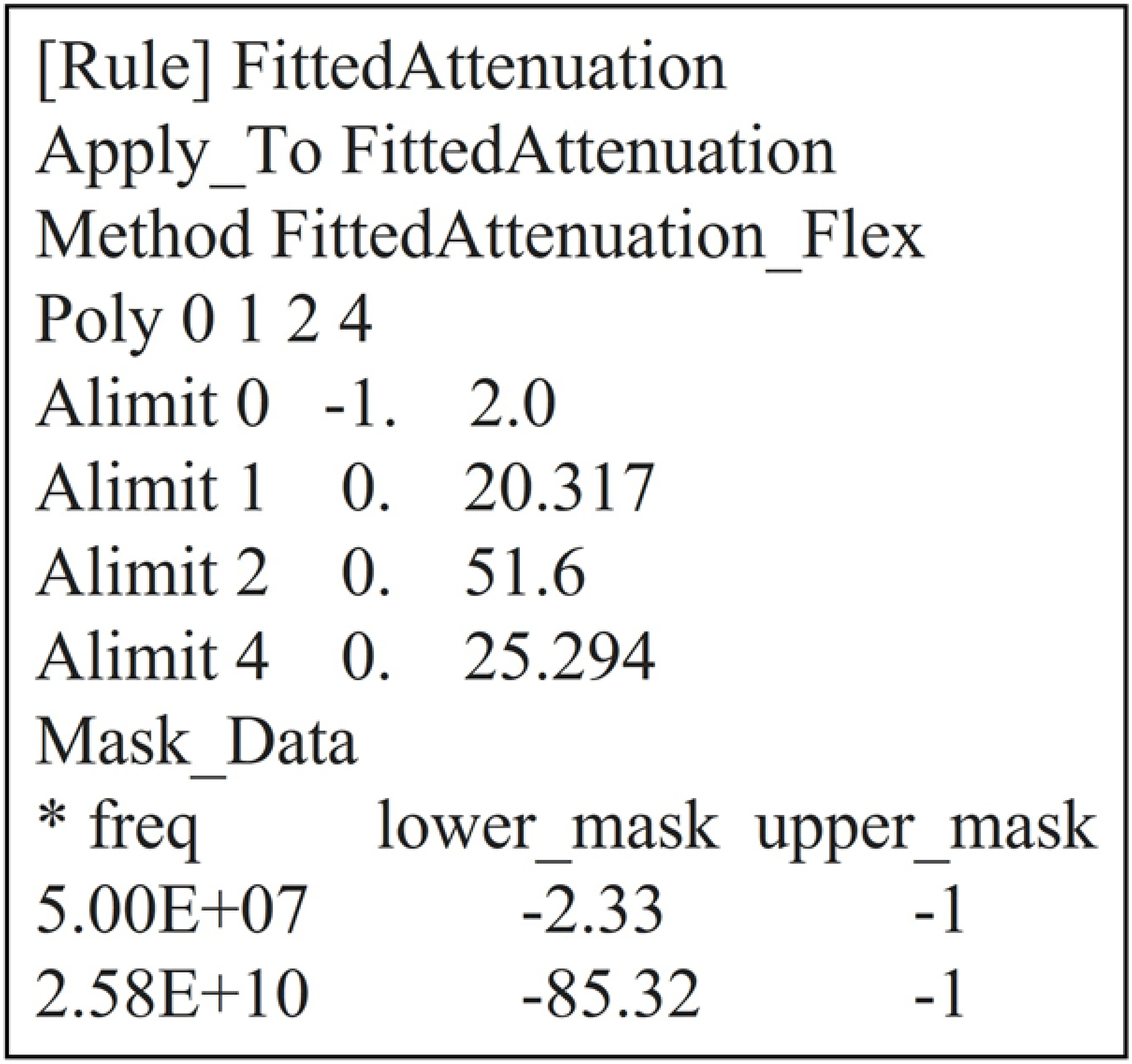
Note: Rules files can be nested; that is, ‘Included’ files may include other files.

Fitted Attenuation and Insertion Loss Deviation Rules

Fitted Attenuation and Insertion Loss Deviation rules use the methods **'Fitted Attenuation_Flex'** and **'Insertion Loss Deviation_Flex'**. Although the methods **Fitted Attenuation** and **Insertion Loss Deviation** are still supported, **Fitted Attenuation_Flex** and **Insertion Loss Deviation_Flex** allow you to specify the equation coefficients from the specification. They fit terms a0, a1, a2 and a4 in the following polynomial from CEI-28G-VSR:

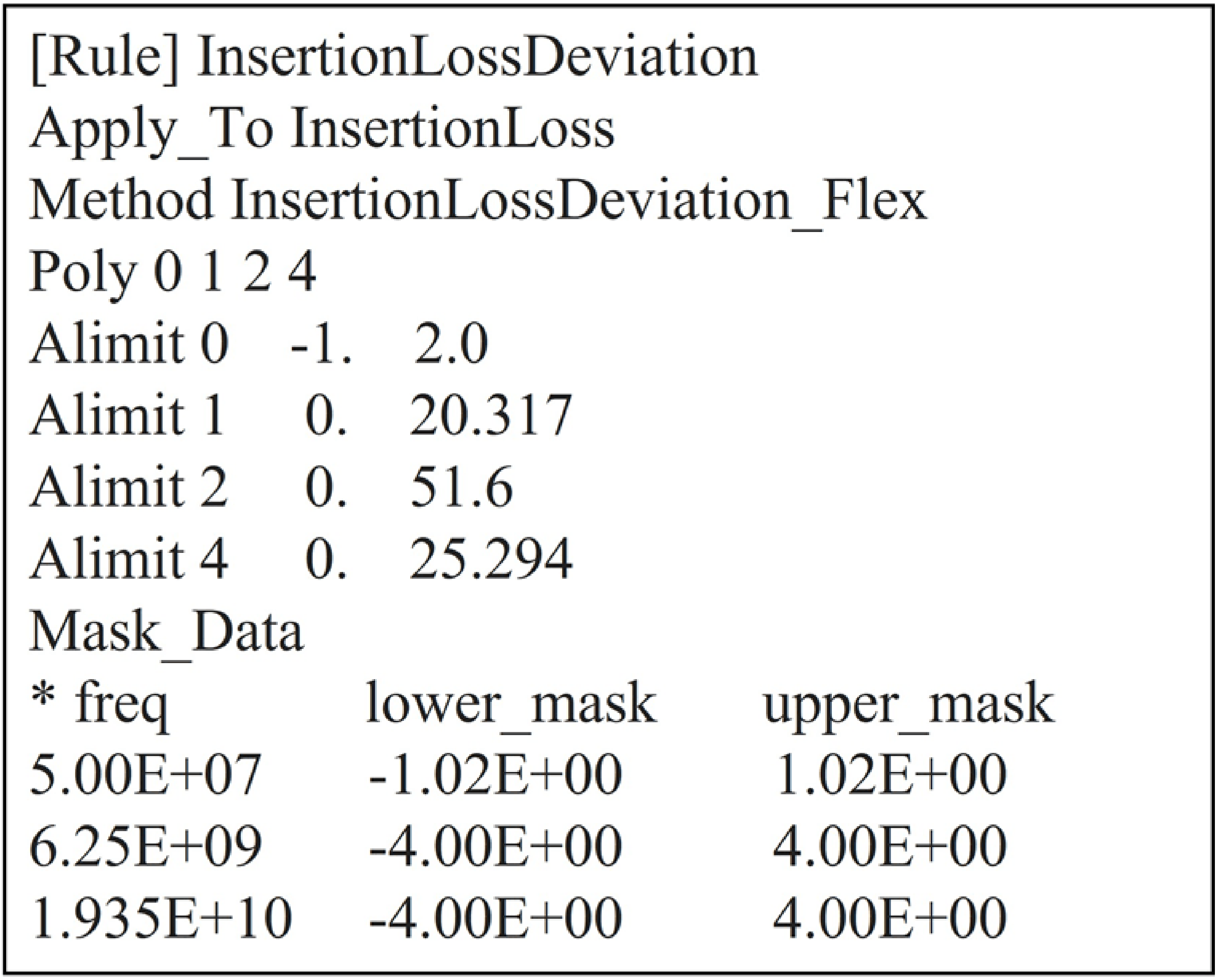
$$IL_{fitted}(f) = a0 + a1\sqrt{\frac{f}{fb}} + a2\frac{f}{fb} + a4\left(\frac{f}{fb}\right)^{2} \ (dB)$$

In the Rules file, the following keywords put constrains on Ai; 'Poly i j k' . Where ‘i’ ‘j’ ‘k’ are the polynomial coefficients to include in the fit. An example of OIF_CEI_4.0 fitted attenuation is shown:



```
[Rule] FittedAttenuation
Apply_To FittedAttenuation
Method FittedAttenuation_Flex
Poly 0 1 2 4
Alimit 0   -1.    2.0
Alimit 1    0.    20.317
Alimit 2    0.    51.6
Alimit 4    0.    25.294
Mask_Data
* freq          lower_mask  upper_mask
5.00E+07            -2.33          -1
2.58E+10            -85.32         -1
```

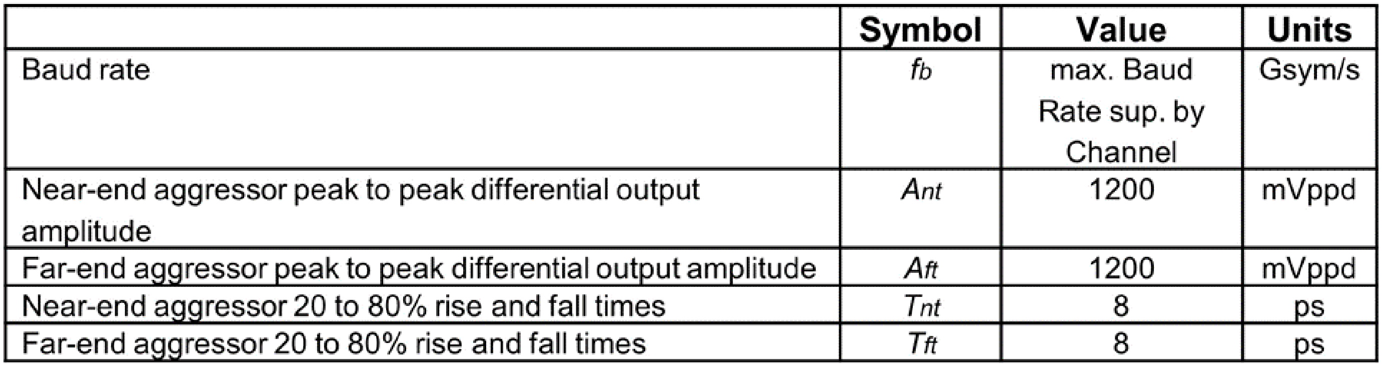
This is also the default. You can add additional polynomial coefficients as long as they are designated properly and follow the ‘Alimit i min max’ format where ‘i’ is the polynomial coefficient number which is constrained to be >= min and <= max. An example rule for insertion loss deviation which also follows the polynomial method is shown:.



```
[Rule] InsertionLossDeviation
Apply_To InsertionLoss
Method InsertionLossDeviation_Flex
Poly 0 1 2 4
Alimit 0    -1.    2.0
Alimit 1     0.    20.317
Alimit 2     0.    51.6
Alimit 4     0.    25.294
Mask_Data
* freq         lower_mask      upper_mask
5.00E+07      -1.02E+00        1.02E+00
6.25E+09      -4.00E+00        4.00E+00
1.935E+10     -4.00E+00        4.00E+00
```

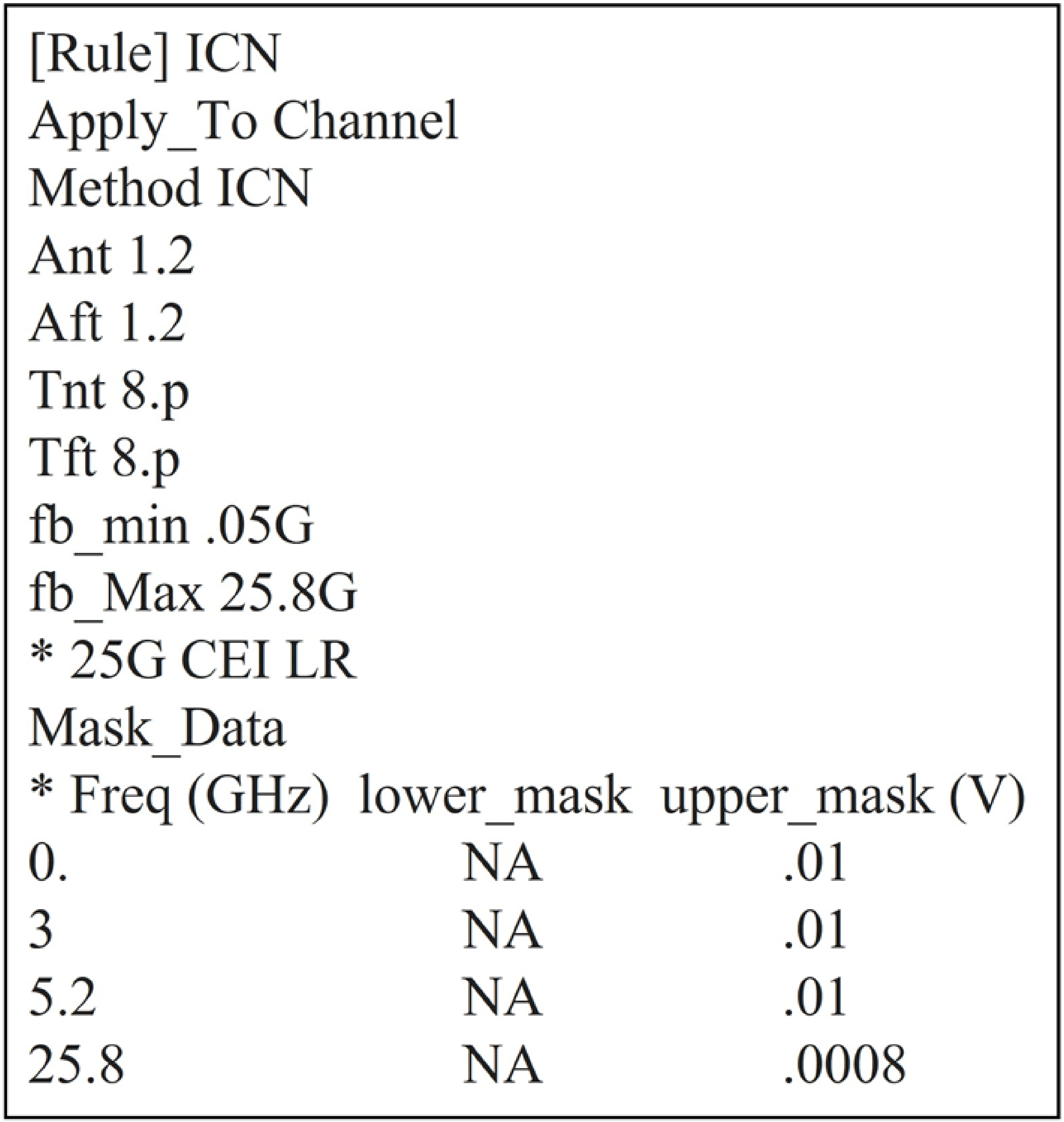
ICN (Integrated Crosstalk Noise)

The Method ICN is implemented with the rules file for OIF_CEI_4.0. The definitions and default values for amplitude and rise times based on OIF CEI-3.1 specification is given:

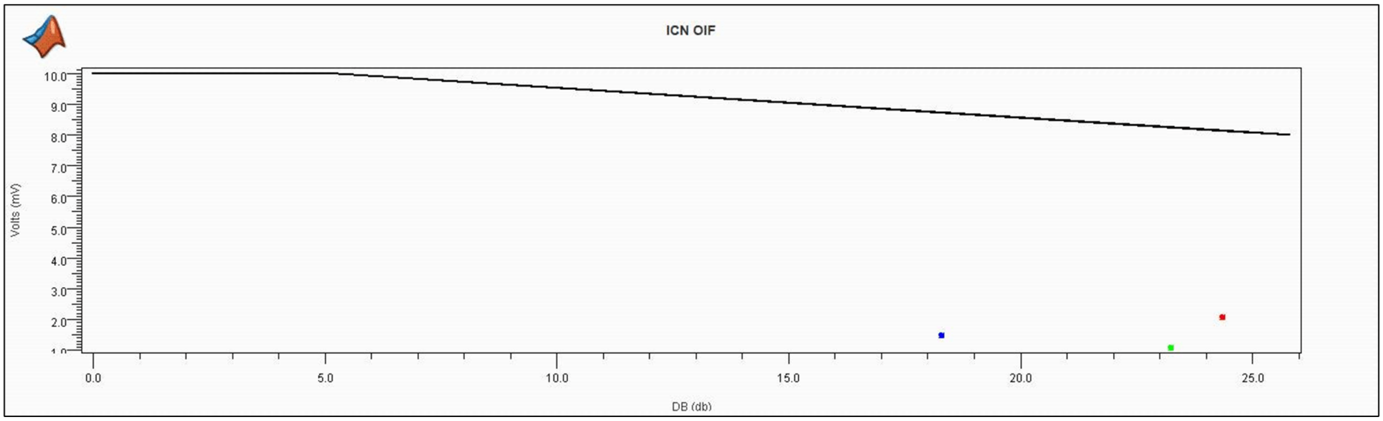


	Symbol	Value	Units
Baud rate	f_b	max. Baud Rate sup. by Channel	Gsym/s
Near-end aggressor peak to peak differential output amplitude	A_{nt}	1200	mVppd
Far-end aggressor peak to peak differential output amplitude	A_{ft}	1200	mVppd
Near-end aggressor 20 to 80% rise and fall times	T_{nt}	8	ps
Far-end aggressor 20 to 80% rise and fall times	T_{ft}	8	ps

The parameters **Fb_min** and **Fb_max**are the minimum and maximum frequency range of the ICN calculation from the standard. The mask is the rule as a function of the Insertion Loss of the channel at Fc. An example ICN rule and the plotted simulation results for ICN are shown below:.



```
[Rule] ICN
Apply_To Channel
Method ICN
Ant 1.2
Aft 1.2
Tnt 8.p
Tft 8.p
fb_min .05G
fb_Max 25.8G
* 25G CEI LR
Mask_Data
* Freq (GHz)  lower_mask  upper_mask (V)
0.                  NA            .01
3                   NA            .01
5.2                 NA            .01
25.8                NA            .0008
```

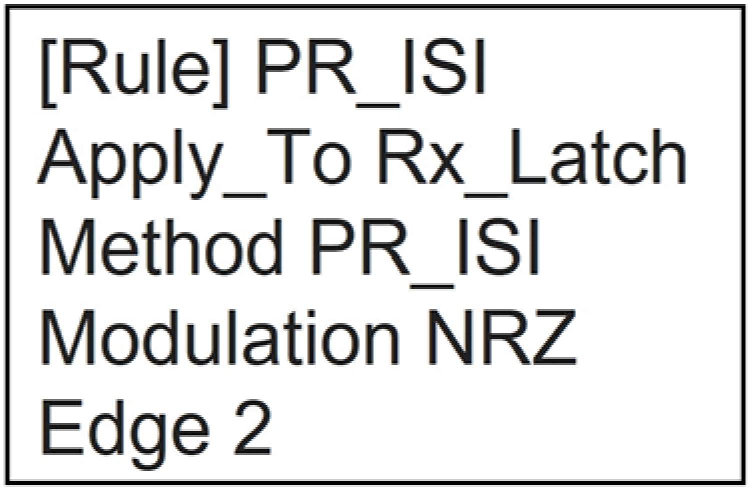


Note that ICN is only one point (volts/dB) and this is showing ICN on multiple channels.

PR_ISI (Pulse Response ISI)

Serial Link Designer can analyze the equalized pulse response of the channel. It first finds the center of the eye using an algorithm known as “*Hula Hoop*”. The Hula Hoop algorithm is simply taking a hula hoop (circle) with a diameter of 1UI and dropping it down horizontally on the pulse response until it stops. The center of the hula hoop is the main cursor time. The voltage at this point is **'PR Cursor (V)'**. The time of this point is **'PR Delay (ns)'**. The ISI is the sum of all of the absolute values of the pulse response at steps of 1 UI from the main cursor (except at the main cursor time). The **ISI Right** is the sum of all these absolute values after the *Edge*UI* to the right of the main cursor. The **ISI Left** is the sum of all these absolute values from the beginning to and including Edge*UI to the right of the main cursor.

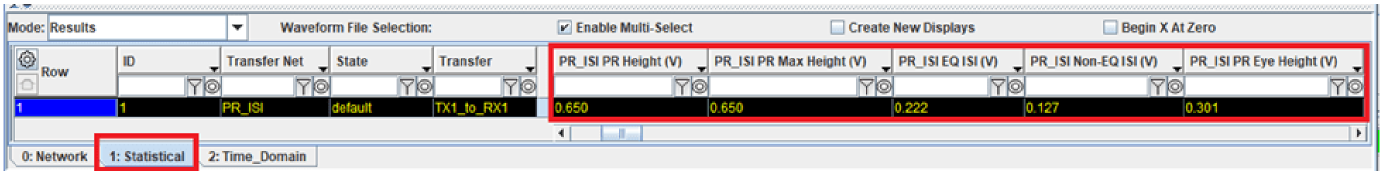
Consider an Rx with a 4 tap DFE. Then **Edge**is 4. The **ISI Left** is ISI that can be corrected by a combination of Tx FFE, Rx CTLE, and DFE. An excellent equalization makes “**ISI Left”** very small. “**ISI Right**” cannot be equalized. An example rule file for PR_ISI is shown:



```
[Rule] PR_ISI
Apply_To Rx_Latch
Method PR_ISI
Modulation NRZ
Edge 2
```

Along with ‘Edge’, ‘Modulation’ must be specified in the rule as either PAM4 or NRZ.

The posted results are provided in the Statistical tab of the channel analysis report, or statistical tab of the SiViewer after loading results. The reporting is shown below.



Mode: Results

Waveform File Selection:

☑ Enable Multi-Select □ Create New Displays □ Begin X At Zero

Row	ID	Transfer Net	State	Transfer	PR_ISI PR Height (V)	PR_ISI PR Max Height (V)	PR_ISI EQ ISI (V)	PR_ISI Non-EQ ISI (V)	PR_ISI PR Eye Height (V)
1	1	PR_ISI	default	TX1_to_RX1	0.650	0.650	0.222	0.127	0.301

The highlighted columns in the results tabs in the figure above are:

PR Height (V): Height of Pulse Response in center of Halo. This ise the height of the statistical eye at the densest point of the eye.

PR Max Height (V) : Max height of the pulse response.

PR Eye Height (V) : The inner height of the statistical eye.

EQ ISI (V): Sum of absolute values of pulse response at center -1UI, -2UI, ..., +1UI, +2UI, ...+NUI. This is “equalizable ISI”.

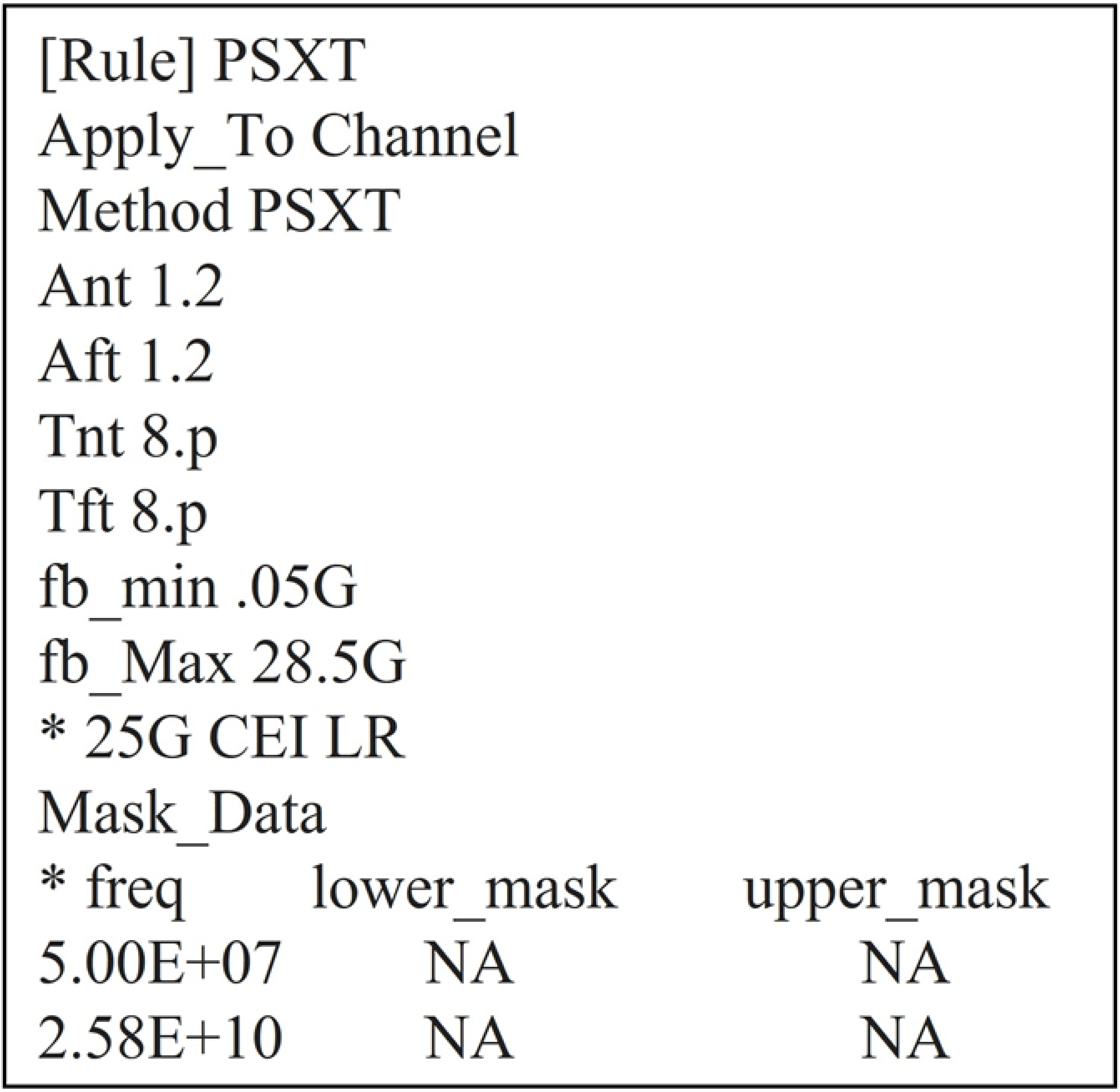
Non-EQ ISI (V): Sum of absolute values of Pulse response at center +(N+1)UI, +(N+2)UI, ...+NUI. This is “non-equalizable ISI”.

PSXT (Power Sum Crosstalk)

The PSXT rule is based on Equation 69B-17 of the IEEE 802.3-2012 specification. This method is also applicable to the OIF_CEI_4.0 specification for CEI-25G LR implementation. According to the 802.3 specification, “the differential crosstalk is calculated as the power sum of the individual NEXT and FEXT aggressors (PSXT)". PSXT is computed as:.

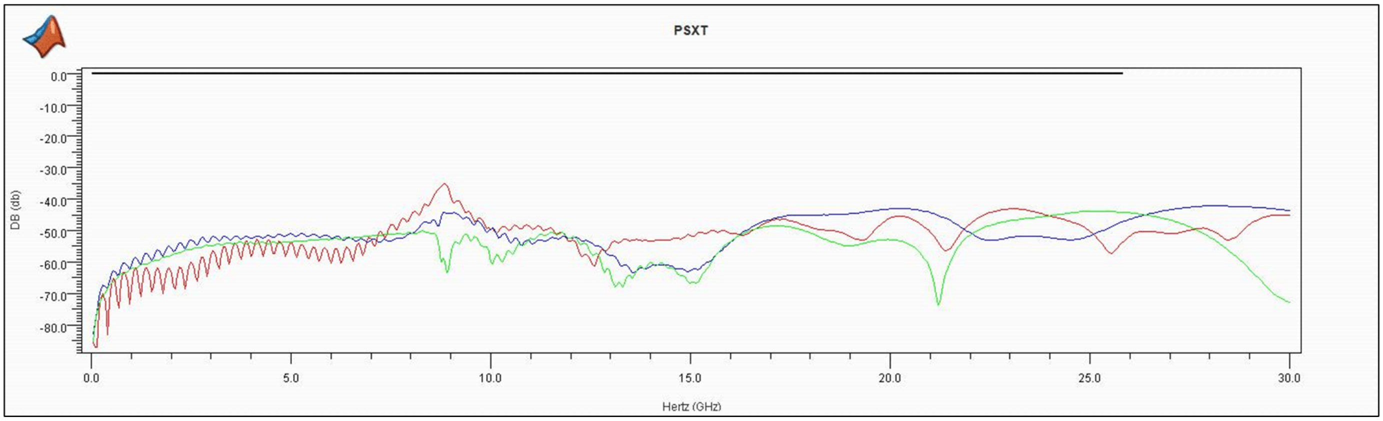
$$PSXT(f) = -10\log\left(10^{-\frac{PSNEXT(f)}{10}} + 10^{-\frac{PSFEXT(f)}{10}}\right)$$

The rules file example looks like:



```
[Rule] PSXT
Apply_To Channel
Method PSXT
Ant 1.2
Aft 1.2
Tnt 8.p
Tft 8.p
fb_min .05G
fb_Max 28.5G
* 25G CEI LR
Mask_Data
* freq       lower_mask       upper_mask
5.00E+07        NA                NA
2.58E+10        NA                NA
```

PSXT plotted results looks like:



IEEE Return Loss

The rule for return loss is reported as a negative number. This is consistent with how return loss is historically characterized in many specifications. But this characterization is inconsistent with the IEEE definition. Ideally, when expressed in decibels, loss quantities are positive numbers. The IEEE definition of return loss is the difference in dB between the incident power sent towards the Device Under Test (DUT) and the power reflected, resulting in a positive sign:

$$RL(dB) = 10\log_{10}\left(\frac{Pi}{Pr}\right)$$

However, taking the ratio of reflected to incident power results in a negative sign for return loss.

$$RL(dB) = 10\log_{10}\left(\frac{P_r}{P_i}\right)$$

, where

$\left(\frac{P_r}{P_i}\right)$ becomes a negative of $\left(\frac{P_i}{P_r}\right)$ from above.

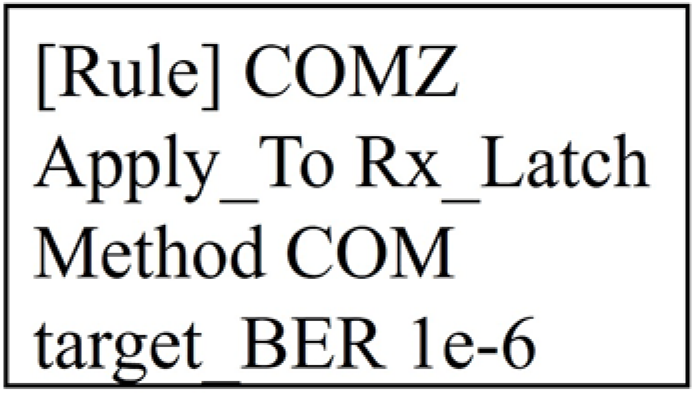
For some standards compliance work, this metric should be reported as a positive number. The rule “IEEE_Return_Loss” reports the magnitude of loss as a positive value.

COM (Channel Operating Margin)

COM or channel operating margin is a figure of merit which is essentially a signal to noise ratio for evaluating the viability of high-speed channels. The IEEE 802.3bj standards committee first developed the COM requirement to support 100GBASE-KR backplane applications running 4-channels at 25Gbps. CEI-56G-LR standard which includes PAM4 signaling adopted the COM metric in 2016. It is expected that 112Gbps applications will use some form of COM in channel compliance.

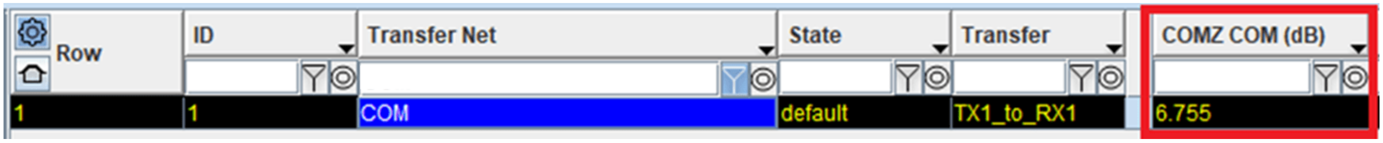
The IEEE 802.3bj standard committee created a MATLAB script that calculates the COM metric for users who have MATLAB software and access to channel s-parameter models. In Serial Link Designer this is further expanded where the channel models from the user's pre-layout design are automatically fed into the COM script to interactively analyze the viability of the user's channel design.

If you do not have access to the MATLAB script, a COM type rule gives you insight to the signal to noise characteristics of your channel design. Although it is not an exact duplication of COM, it gives you insight to the performance of their channel relative to others. An example of it as it is used in an Serial Link Designer rules file to report this value is shown:



```
[Rule] COMZ
Apply_To Rx_Latch
Method COM
target_BER 1e-6
```

This rule requires no configuration spreadsheet the way IEEE COM does. The simulator gets all of the data from the channel design and user settings. QCD makes the calculation and shows results in the channel analysis report. These results can also be seen in the **Signal Integrity Viewer** and are reported in the 'Statistical' results tab. You can choose the rule name.. You can set the `Apply to` parameter to Rx_pin, Rx_pad, or Rx_latch and set the target BER your specifications.. It is required that the Method be COM. Lastly this rule applies to either NRZ or PAM4 Modulation.



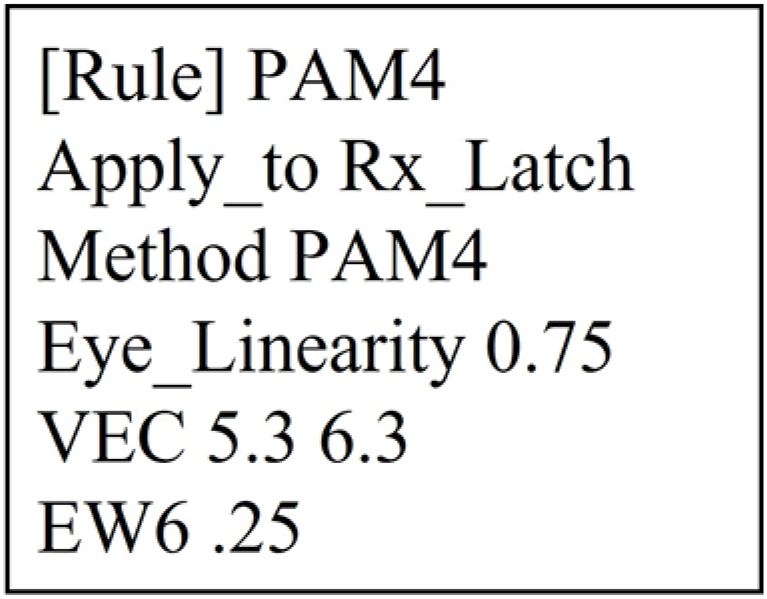
Row	ID	Transfer Net	State	Transfer	COMZ COM (dB)
1	1	COM	default	TX1_to_RX1	6.755

Note: This rule is applied during statistical analysis, Unlike the IEEE COM results which are reported in the network tab, the QCD COM results are located in the statistical tab of the viewer.

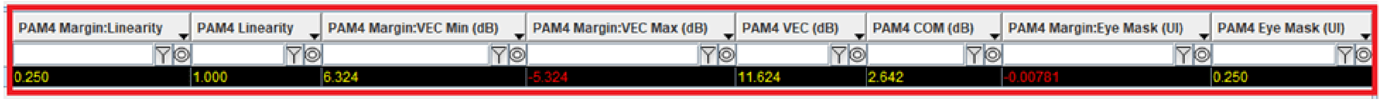
PAM4

Serial Link Designer rules for PAM4 analysis are implemented for the IEEE and CEI 56G specifications to report eye linearity and vertical eye closure (VEC).

Eye linearity is the proportion of the smallest of the three PAM4 eye heights to the largest of the three. Vertical eye closure is a report of the closure of the ideal eye height in dB with respect to the simulated or measured eye height at a specific bit-error ratio. An example rule that includes the eye linearity limit of 0.75, vertical eye closure limits are between 5.3dB and 6.3 dB and the eye width minimum limit at 10-6 of 0.25UI is shown:



```
[Rule] PAM4
Apply_to Rx_Latch
Method PAM4
Eye_Linearity 0.75
VEC 5.3 6.3
EW6 .25
```



PAM4 Margin:Linearity	PAM4 Linearity	PAM4 Margin:VEC Min (dB)	PAM4 Margin:VEC Max (dB)	PAM4 VEC (dB)	PAM4 COM (dB)	PAM4 Margin:Eye Mask (UI)	PAM4 Eye Mask (UI)
0.250	1.000	6.324	-5.324	11.624	2.642	-0.00781	0.250

USB 3.1 IMR and IXT

IMR (Integrated Multi-Reflection) and IXT (Integrated Crosstalk) are compliance metrics called out in the USB 3.1 specification. The equations used for these rules are taken from the USB 3.1 specification. To calculate IMR the ILD (Insertion Loss Deviation) needs to be calculated. ILD is the difference between the raw insertion loss of the channel and the IL fit(fitted insertion loss)and is shown as:.

$$ILD(f) = IL(f) - ILfit(f)$$

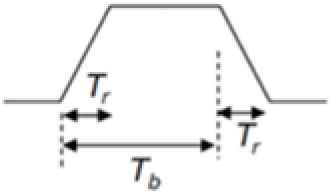
Here IL is the insertion loss of the channel over frequency minus the fitted insertion loss. The insertion loss fit is given by:.

$$ILfit(f) = a + b\sqrt{f} + cf + d\sqrt{f^3}$$

Where f is frequency and a, b, c and d are the fitting coefficients of the polynomial. The integrated multi-reflection, or IMR, is calculated using:

$$IMR = \sqrt{\int_0^{f_{\max}} |ILD(f)|^2 |V_{in}(f)|^2 df / f_{Nq}} *1000 \text{ (in mV)}$$

where fNq is the Nyquist frequency (5 GHz), fmax is chosen as the 2 times the Nyquist frequency, and Vin(f) is the input trapezoidal pulse spectrum shown below:.



$$|V_{in}(\omega)| = \left| \frac{\sin\left(\frac{\omega T_r}{2}\right)}{\frac{\omega T_r}{2}} \cdot \frac{\sin\left(\frac{\omega T_b}{2}\right)}{\frac{\omega T_b}{2}} \right|$$

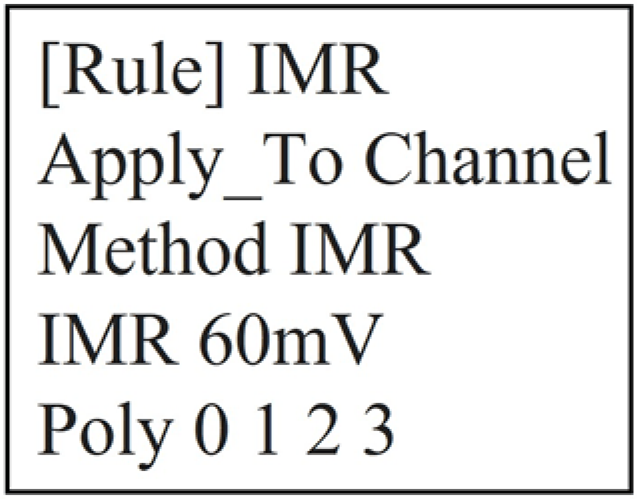
T_b=Unit Interval=100 ps
T_r=Rise time (0-100%)=0.2T_b
$\omega=2\pi f$

Integrated crosstalk, IXT, is defined as:

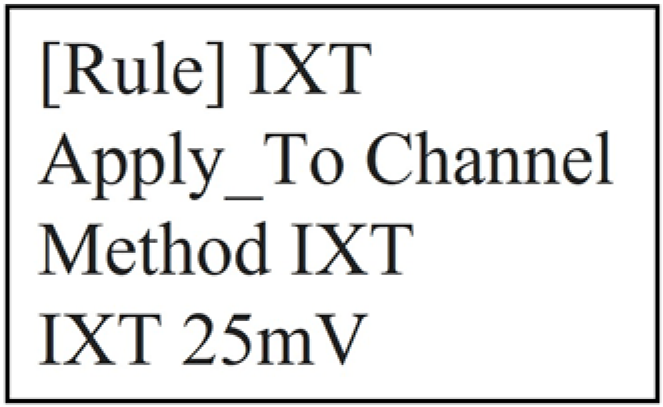
$$IXT = \sqrt{\int_0^{f_{\max}} |NEXT(f)|^2 |V_{in}(f)|^2 df / f_{Nq}} *1000 \text{ (in mV)}$$

Where NEXT(f) is the near-end crosstalk between the SuperSpeed Gen2 signal pairs. The contribution of USB 2 D+/D- pair to SuperSpeed signal pairs is relatively small and is not included in IXT for simplicity.

Example rules for IMR and IXT are shown:

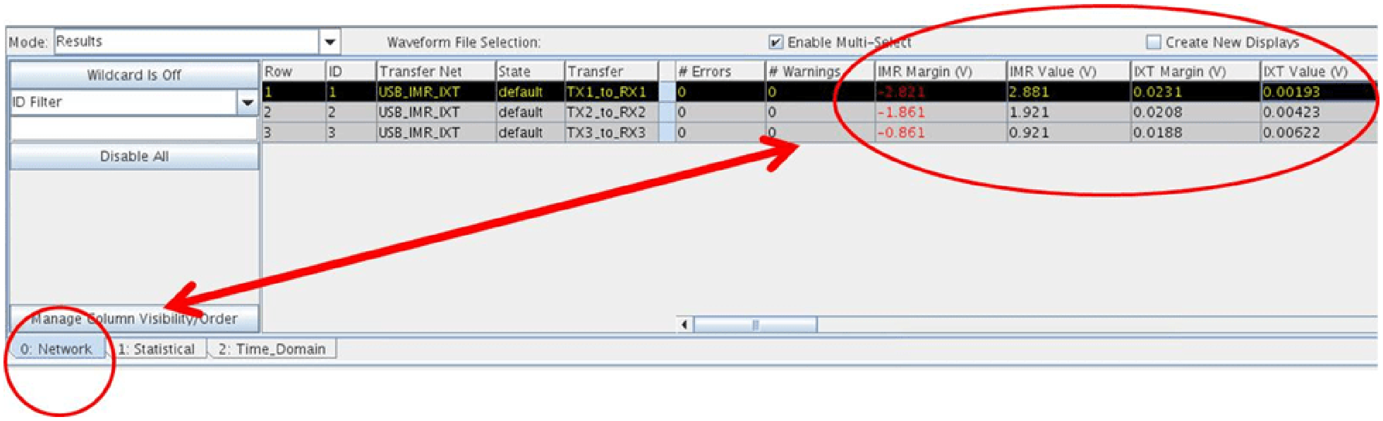


```
[Rule] IMR
Apply_To Channel
Method IMR
IMR 60mV
Poly 0 1 2 3
```



```
[Rule] IXT
Apply_To Channel
Method IXT
IXT 25mV
```

There are no masks to be applied for these two rules, the results displayed are for the calculated value and the margin to the limit specified in the rule. In the cases below IMR is specified with a limit of 60mV and IXT is set at 25mV. The results are displayed in the network tab of the **Signal Integrity Viewer** after simulation.



Mode: Results

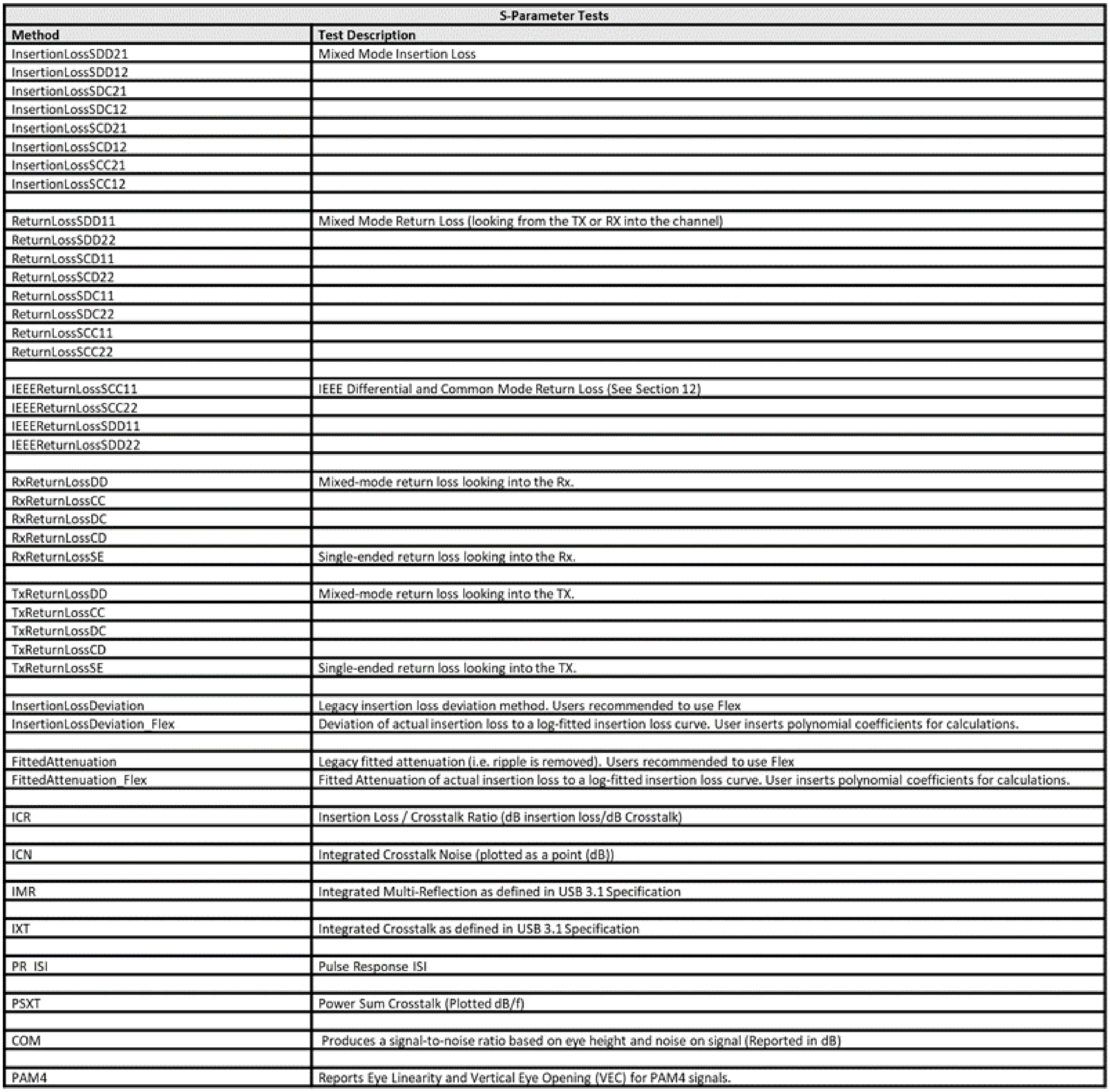
Row	ID	Transfer Net	State	Transfer		# Errors	# Warnings	IMR Margin (V)	IMR Value (V)	IXT Margin (V)	IXT Value (V)
1	1	USB_IMR_IXT	default	TX1_to_RX1		0	0	-2.821	2.881	0.0231	0.00193
2	2	USB_IMR_IXT	default	TX2_to_RX2		0	0	-1.861	1.921	0.0208	0.00423
3	3	USB_IMR_IXT	default	TX3_to_RX3		0	0	-0.861	0.921	0.0188	0.00622

Rules Definitions (Data Fields)

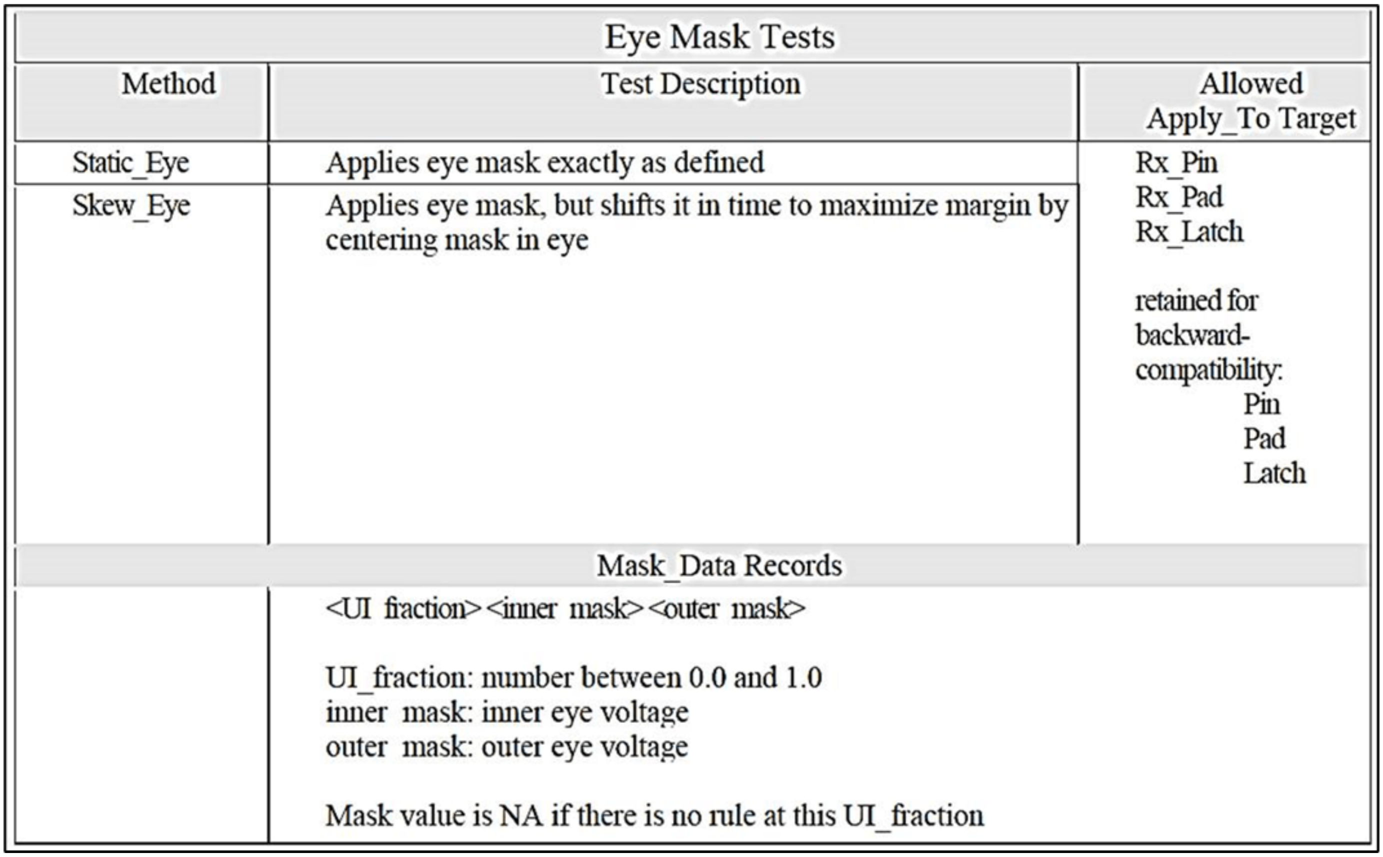
Rule name can be any ASCII string allowable by the host operating system. Name terminates upon the first occurrence of a separator character (space, TAB, or comma). `Methods`, `Apply_To` targets and `Mask_Data` records are described in the following tables. For convenience, rules are divided into two categories, by the type of test that is applied: Eye Mask Tests and S-Parameter Tests.

Note: If there is no “`Apply_To`” listed in the transfer net table, then it is not required. Use “Apply_To NA” in the rules file.

The tables below are references of the available rules.

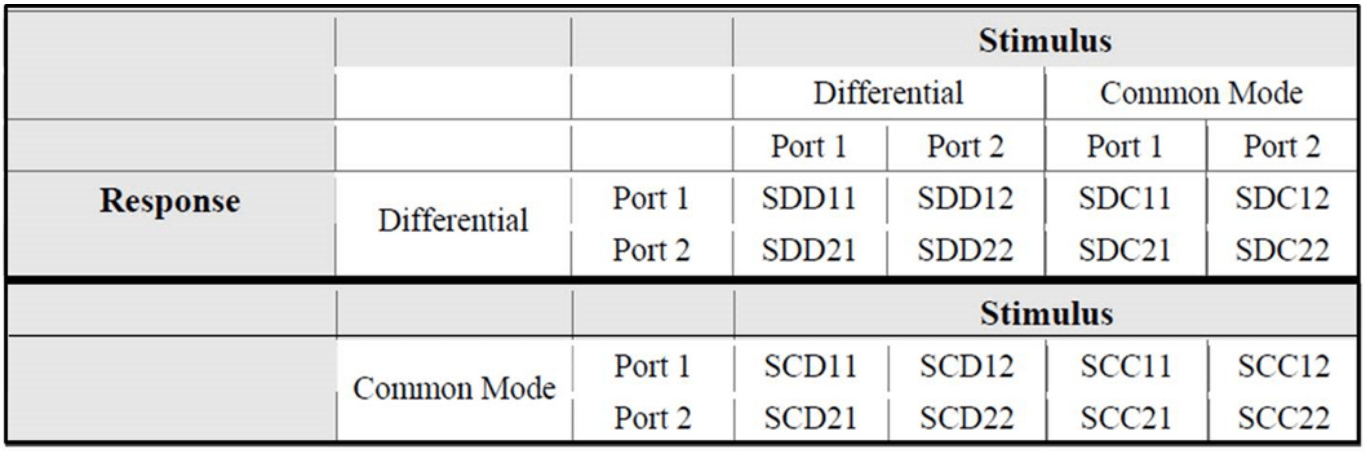


S-Parameter Tests	
Method	**Test Description**
InsertionLossSDD21	Mixed Mode Insertion Loss
InsertionLossSDD12	
InsertionLossSDC21	
InsertionLossSDC12	
InsertionLossSCD21	
InsertionLossSCD12	
InsertionLossSCC21	
InsertionLossSCC12	
ReturnLossSDD11	Mixed Mode Return Loss (looking from the TX or RX into the channel)
ReturnLossSDD22	
ReturnLossSCD11	
ReturnLossSCD22	
ReturnLossSDC11	
ReturnLossSDC22	
ReturnLossSCC11	
ReturnLossSCC22	
IEEEReturnLossSCC11	IEEE Differential and Common Mode Return Loss (See Section 12)
IEEEReturnLossSCC22	
IEEEReturnLossSDD11	
IEEEReturnLossSDD22	
RxReturnLossDD	Mixed-mode return loss looking into the Rx.
RxReturnLossCC	
RxReturnLossDC	
RxReturnLossCD	
RxReturnLossSE	Single-ended return loss looking into the Rx.
TxReturnLossDD	Mixed-mode return loss looking into the TX.
TxReturnLossCC	
TxReturnLossDC	
TxReturnLossCD	
TxReturnLossSE	Single-ended return loss looking into the TX.
InsertionLossDeviation	Legacy insertion loss deviation method. Users recommended to use Flex
InsertionLossDeviation_Flex	Deviation of actual insertion loss to a log-fitted insertion loss curve. User inserts polynomial coefficients for calculations.
FittedAttenuation	Legacy fitted attenuation (i.e. ripple is removed). Users recommended to use Flex
FittedAttenuation_Flex	Fitted Attenuation of actual insertion loss to a log-fitted insertion loss curve. User inserts polynomial coefficients for calculations.
ICR	Insertion Loss / Crosstalk Ratio (dB insertion loss/dB Crosstalk)
ICN	Integrated Crosstalk Noise (plotted as a point (dB))
IMR	Integrated Multi-Reflection as defined in USB 3.1 Specification
IXT	Integrated Crosstalk as defined in USB 3.1 Specification
PR_ISI	Pulse Response ISI
PSXT	Power Sum Crosstalk (Plotted dB/f)
COM	Produces a signal-to-noise ratio based on eye height and noise on signal (Reported in dB)
PAM4	Reports Eye Linearity and Vertical Eye Opening (VEC) for PAM4 signals.



Eye Mask Tests		
Method	Test Description	Allowed Apply_To Target
Static_Eye	Applies eye mask exactly as defined	Rx_Pin Rx_Pad Rx_Latch retained for backward-compatibility: Pin Pad Latch
Skew_Eye	Applies eye mask, but shifts it in time to maximize margin by centering mask in eye	
Mask_Data Records		
	<UI fraction><inner mask><outer mask> UI_fraction: number between 0.0 and 1.0 inner mask: inner eye voltage outer mask: outer eye voltage Mask value is NA if there is no rule at this UI_fraction	

For reference, mixed-mode S-parameters referred to in the rules are defined in the table below.



			Stimulus			
			Differential		Common Mode	
			Port 1	Port 2	Port 1	Port 2
Response	Differential	Port 1	SDD11	SDD12	SDC11	SDC12
		Port 2	SDD21	SDD22	SDC21	SDC22
			Stimulus			
	Common Mode	Port 1	SCD11	SCD12	SCC11	SCC12
		Port 2	SCD21	SCD22	SCC21	SCC22

Kit Overview

A sample project is provided containing examples of various types of compliance masks that can be defined in Serial Link Designer. These masks can be related to a data eye, based on S-parameter characteristics of a network, limits for coupled noise or ISI. The examples are provided in this sample project as a training tool and quick reference of how to create and apply masks in Serial Link Designer.

To access the example kit, type the following at the Matlab command prompt.

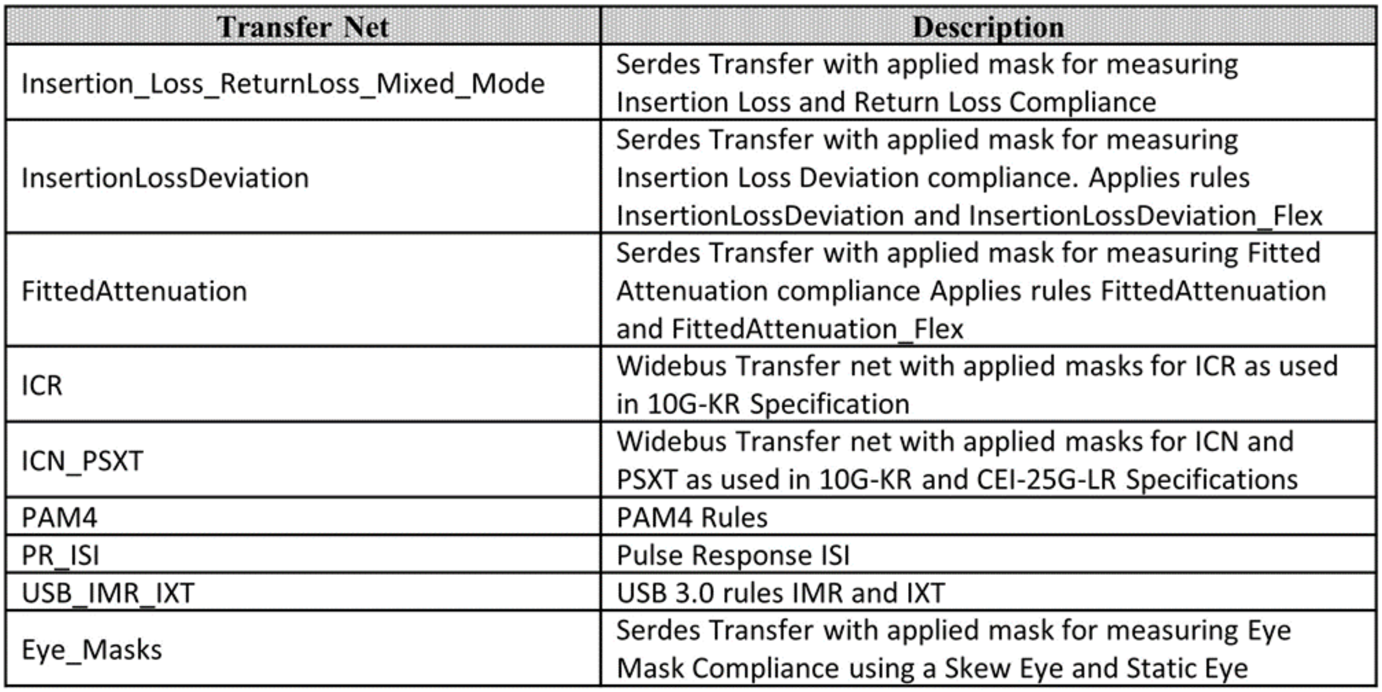
```
openSignalIntegrityKit("SLD_Compliance_Masks");
```

Schematic Sets

One schematic set has been defined in this interface, 'Set1'.

Transfer Nets

Table 5 shows a list of all transfer nets included in this kit. Many of the examples are based on compliance for network characterization. Eye mask examples are also included to demonstrate their usage in Serial Link Designer rules.

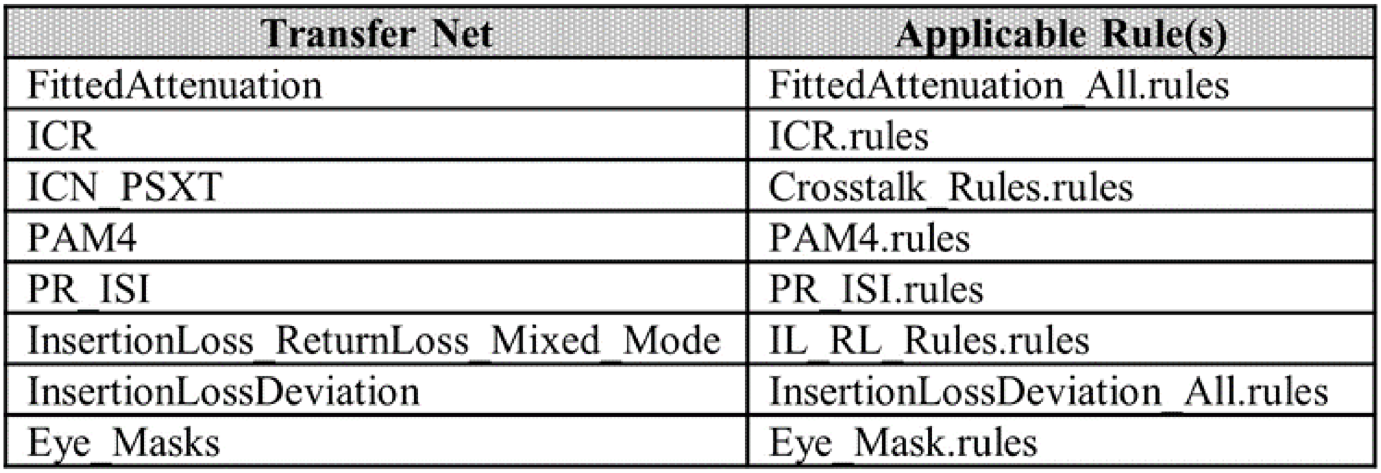


Transfer Net	Description
Insertion_Loss_ReturnLoss_Mixed_Mode	Serdes Transfer with applied mask for measuring Insertion Loss and Return Loss Compliance
InsertionLossDeviation	Serdes Transfer with applied mask for measuring Insertion Loss Deviation compliance. Applies rules InsertionLossDeviation and InsertionLossDeviation_Flex
FittedAttenuation	Serdes Transfer with applied mask for measuring Fitted Attenuation compliance Applies rules FittedAttenuation and FittedAttenuation_Flex
ICR	Widebus Transfer net with applied masks for ICR as used in 10G-KR Specification
ICN_PSXT	Widebus Transfer net with applied masks for ICN and PSXT as used in 10G-KR and CEI-25G-LR Specifications
PAM4	PAM4 Rules
PR_ISI	Pulse Response ISI
USB_IMR_IXT	USB 3.0 rules IMR and IXT
Eye_Masks	Serdes Transfer with applied mask for measuring Eye Mask Compliance using a Skew Eye and Static Eye

Many of the examples are based on compliance for network characterization. Eye mask examples are also included to demonstrate their usage in Serial Link Designer rules.

Kit Transfer Nets and Properties

The properties for each of the transfer nets in the kit along with which rule(s) is applied are listed as:.



Transfer Net	Applicable Rule(s)
FittedAttenuation	FittedAttenuation_All.rules
ICR	ICR.rules
ICN_PSXT	Crosstalk_Rules.rules
PAM4	PAM4.rules
PR_ISI	PR_ISI.rules
InsertionLoss_ReturnLoss_Mixed_Mode	IL_RL_Rules.rules
InsertionLossDeviation	InsertionLossDeviation_All.rules
Eye_Masks	Eye_Mask.rules

Channel Operating Margin (COM) for Serial Link

Channel Operating Margin (COM) is a figure of merit for a passive channel expressed in decibels and is calculated using the ratio of signal amplitude factors to noise amplitude factors. Channel bit rate, insertion loss, return loss, cross-coupling, transmitter and receiver equalization and IC package models are some of the factors applied to determine COM. While it is required for compliance in some applications, COM can also be a valuable part of channel design methodology in general. This example assumes that you have familiarized yourself with the topic page, Channel Operating Margin (COM).

Overview:

The IEEE 802.3bj 100GBASE specification defines the 100GBASE interface to consist of four channels each operating at 25.78125Gbps. These channel designs can involve PCB only, backplane or copper cables. Signaling is accomplished with either NRZ (Non Return to Zero) or PAM4 (Pulse Amplitude Modulation). Encoding the packets with forward error correction (FEC) is optional but can greatly improve a channel BER (Bit Error Ratio). Testing the compliance of the passive electrical channel to the specification requires it to meet or exceed what is known as COM (or Channel Operating Margin) as measured in decibel units. This document provides information on COM and how to use it within a Serial Link Designer project. You can use the implemenation kit for 100GBASE-KR, which is an interface that operates at 25G-Baud per Lane, with 4 Lanes per Link.

About COM:

COM is a figure of merit derived from the scattering parameters of the passive channel. The overall objective is to give the user insight on the quality of the passive channel design. The calculated metric is related to the ratio of the calculated signal amplitude to its calculated noise amplitude. Channel bit rate, insertion loss, return loss, cross-coupling, transmitter and receiver equalization and IC package models are some of the factors applied to determine this figure of merit. Figure 1 shows the 802.3bj channel model with associated test points. The passive channel referenced is between TP0 and TP5. It is important for the user to keep in mind that COM is required for compliance in an 802.3bj application, but can also be a valuable part of a channel design methodology.

COM Example Project:

You can reference the implementation kit for 100GBASE-KR, which is a Serial Link Designer project consisting of a 25Gbps per-lane design, and can be used as an example to show the procedure for running COM within Serial Link Designer.

```
openSignalIntegrityKit("100GBASE_KR");
```

The procedures, some examples and tips are given to demonstrate how one may use COM in the analysis of the simulation results. Figure 3 is the schematic of the channel design that has been created for analysis.

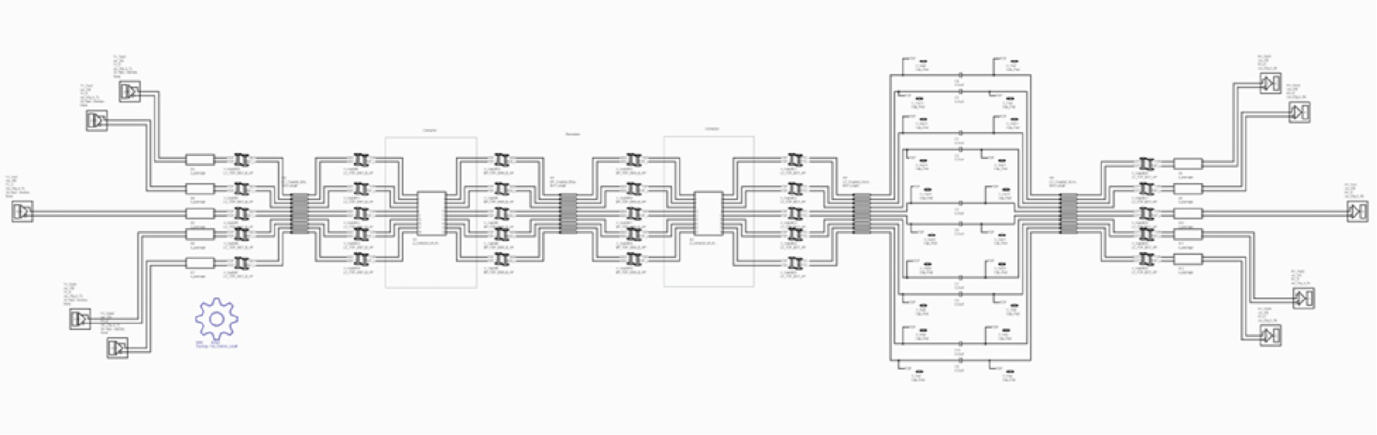


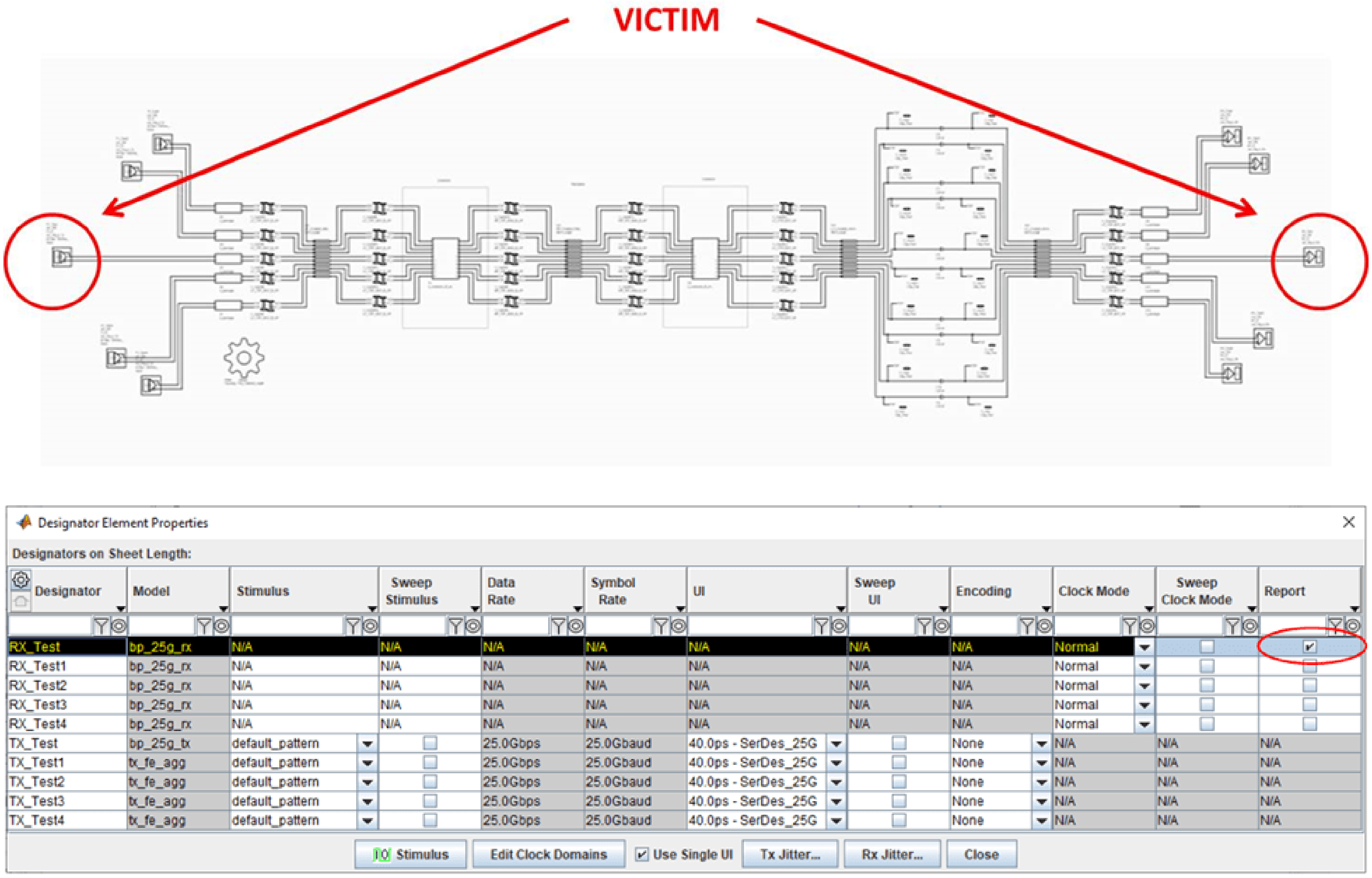
Figure 3. 25Gbps Backplane Schematic Example

The channel design is of a 25Gbps backplane with two line cards and high speed connectors. It is a custom 25Gbps interface designed to meet a target 1E-15 bit-error-ratio. Two identical schematic sheets are included in the project with different solution spaces. One is configured to evaluate the effects of loss on COM and the other is to view crosstalk effects. The “Crosstalk” sheet varies aggressor spacing to demonstrate the effects of crosstalk. The “Length” sheet varies the backplane trace length which affects channel loss and thus will affect the channel COM. TX and RX s-parameter package models are included on the schematic sheets. Thus package characteristics will be included in the channel s-parameter models that will be passed into COM. The spreadsheet has been edited to exclude the any package model from the COM calculation as it will be part of the channel model.

Running COM in MATLAB:

Step 1: Identify Victim Channel

When the channel design is ready for simulation and the spreadsheet and reported results have been identified, the user can start the process of running COM from within the Serial Link Designer App. The user will need to first identify the victim channel on each of the schematic sheets being simulated. Serial Link Designer requires this such that it will create the appropriate s-parameter files for the victim and any aggressors. To identify the victim channel “Designator Element Properties” must be edited on the schematic sheet. The designator element properties window can be accessed by double clicking on any one of the TX or RX designators on the sheet. Figure 8 shows the example project schematic with victim net identified and the element properties window. The report checkbox in the element properties window should be “checked” for the RX designator of the victim channel only and should be “unchecked” for any other RX designators. The FEXT and NEXT aggressor channels will be automatically determined based on their position on the schematic sheet with respect to the victim. In the case of Figure 8 the RX_Test designator is defined as the victim (See Designator Elements Properties window).



Designator Element Properties

Designators on Sheet Length:

Designator	Model	Stimulus	Sweep Stimulus	Data Rate	Symbol Rate	UI	Sweep UI	Encoding	Clock Mode	Sweep Clock Mode	Report
RX_Test	bp_25g_rx	N/A	N/A	N/A	N/A	N/A	N/A	N/A	Normal	□	☑
RX_Test1	bp_25g_rx	N/A	N/A	N/A	N/A	N/A	N/A	N/A	Normal	□	□
RX_Test2	bp_25g_rx	N/A	N/A	N/A	N/A	N/A	N/A	N/A	Normal	□	□
RX_Test3	bp_25g_rx	N/A	N/A	N/A	N/A	N/A	N/A	N/A	Normal	□	□
RX_Test4	bp_25g_rx	N/A	N/A	N/A	N/A	N/A	N/A	N/A	Normal	□	□
TX_Test	bp_25g_tx	default_pattern	□	25.0Gbps	25.0Gbaud	40.0ps - SerDes_25G	□	None	N/A	N/A	N/A
TX_Test1	tx_fe_agg	default_pattern	□	25.0Gbps	25.0Gbaud	40.0ps - SerDes_25G	□	None	N/A	N/A	N/A
TX_Test2	tx_fe_agg	default_pattern	□	25.0Gbps	25.0Gbaud	40.0ps - SerDes_25G	□	None	N/A	N/A	N/A
TX_Test3	tx_fe_agg	default_pattern	□	25.0Gbps	25.0Gbaud	40.0ps - SerDes_25G	□	None	N/A	N/A	N/A
TX_Test4	tx_fe_agg	default_pattern	□	25.0Gbps	25.0Gbaud	40.0ps - SerDes_25G	□	None	N/A	N/A	N/A

Figure 8: Selecting Victim Channel (Designator Element Properties Window)

Step2: Set Simulation Parameters

It is important that the user set the appropriate simulation parameters in Serial Link Designer prior to running simulations. The extracted s-parameter models for use in COM must have the necessary bandwidth and frequency spacing to get an accurate representation. These parameters are based on the characteristic delay and the channel bit rate. The parameters affected are “Max Output Frequency” and “S Param Frequency Step”.

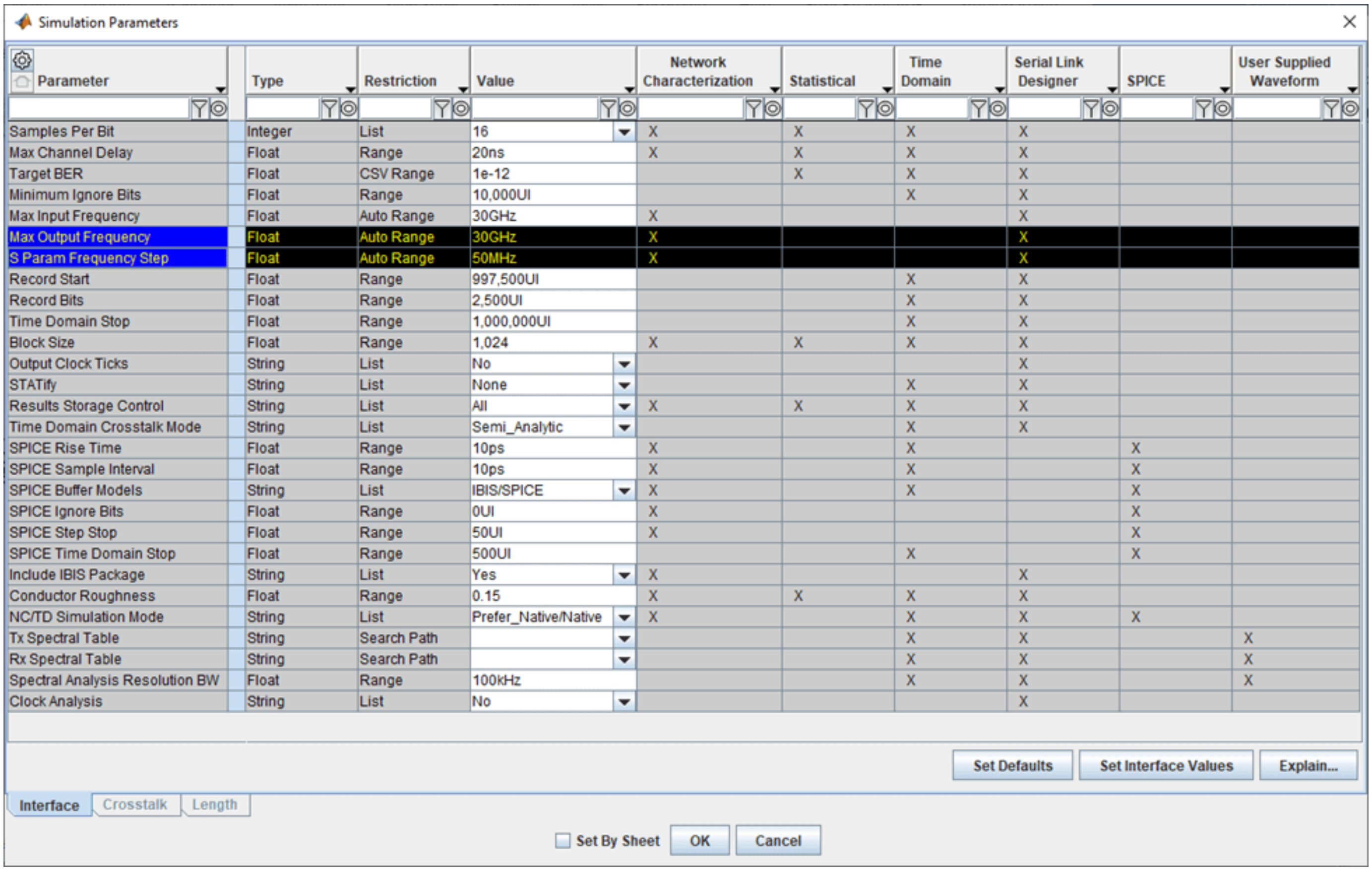
The “Max Output Frequency” parameter should be at least two times the fundamental, or Nyquist, frequency based on the channel bit rate (NRZ signaling). For example, if the channel simulations are based on 25.78125Gbps, the Nyquist or fundamental frequency would be approximately 12.89GHz. The output frequency should then be set to at least 25.78125GHz.

The “S Param Frequency Step” setting is based on the through path delay of the channel. Equation 1 can be used to calculate the step size. The calculation is based on a settling time of three round trip delays for reflective energy.

S Param Frequency Step ≤

Equation 1: Frequency Step Calculation

The simulation parameters are set in the Serial Link Designer “Simulation Parameter” window (Figure 9). To open the Simulation Parameters window, select the pull-down menu SetupàSimulation Parameters.

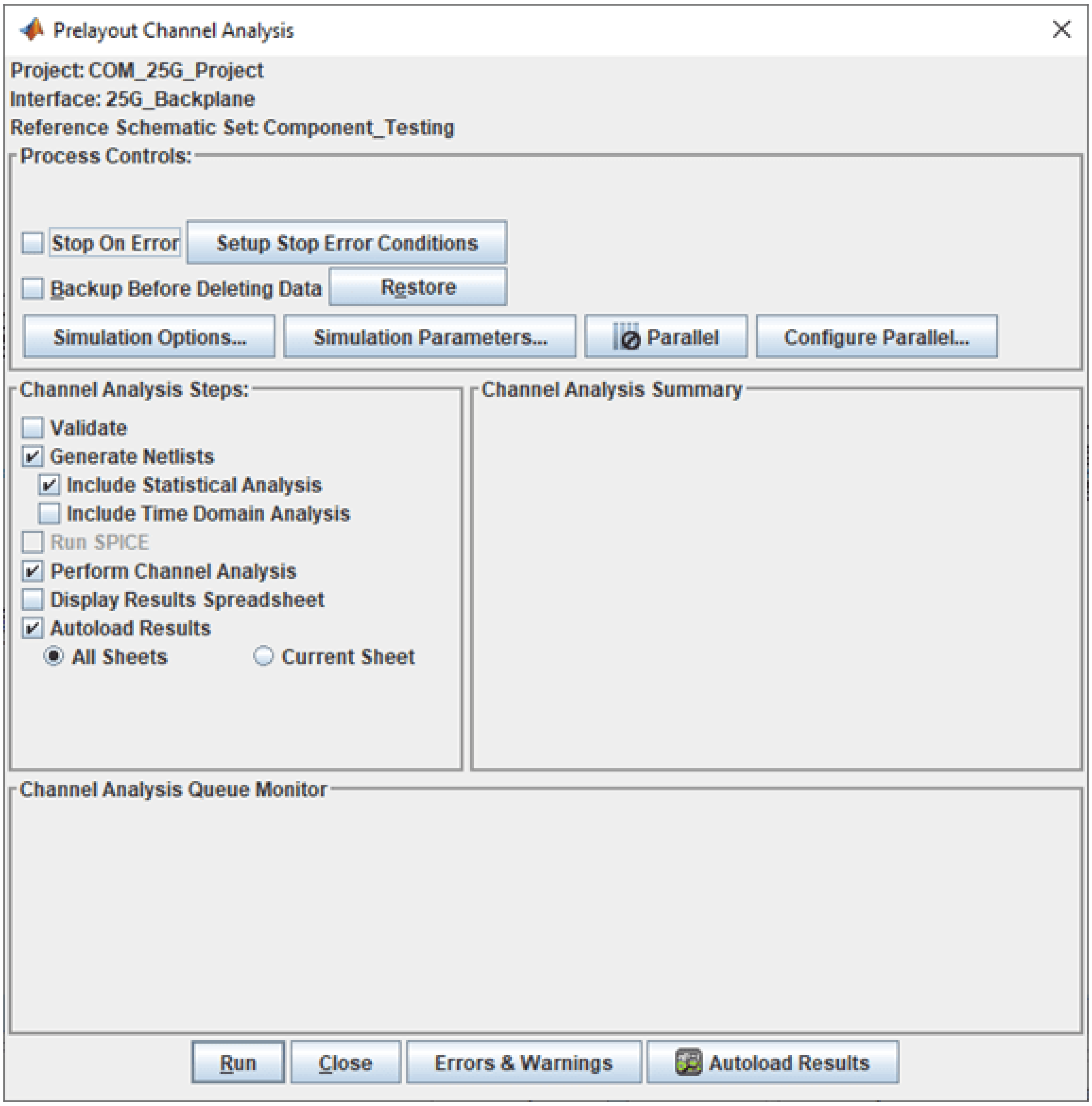


Parameter	Type	Restriction	Value	Network Characterization	Statistical	Time Domain	Serial Link Designer	SPICE	User Supplied Waveform
Samples Per Bit	Integer	List	16	X	X	X	X		
Max Channel Delay	Float	Range	20ns	X	X	X	X		
Target BER	Float	CSV Range	1e-12		X	X	X		
Minimum Ignore Bits	Float	Range	10,000UI			X	X		
Max Input Frequency	Float	Auto Range	30GHz	X			X		
Max Output Frequency	Float	Auto Range	30GHz	X			X		
S Param Frequency Step	Float	Auto Range	50MHz	X			X		
Record Start	Float	Range	997,500UI			X	X		
Record Bits	Float	Range	2,500UI			X	X		
Time Domain Stop	Float	Range	1,000,000UI			X	X		
Block Size	Float	Range	1,024	X	X	X	X		
Output Clock Ticks	String	List	No				X		
STATify	String	List	None			X	X		
Results Storage Control	String	List	All	X	X	X	X		
Time Domain Crosstalk Mode	String	List	Semi_Analytic			X	X		
SPICE Rise Time	Float	Range	10ps	X		X		X	
SPICE Sample Interval	Float	Range	10ps	X		X		X	
SPICE Buffer Models	String	List	IBIS/SPICE	X		X		X	
SPICE Ignore Bits	Float	Range	0UI	X				X	
SPICE Step Stop	Float	Range	50UI	X				X	
SPICE Time Domain Stop	Float	Range	500UI			X		X	
Include IBIS Package	String	List	Yes	X			X		
Conductor Roughness	Float	Range	0.15	X	X	X	X		
NC/TD Simulation Mode	String	List	Prefer_Native/Native	X		X	X	X	
Tx Spectral Table	String	Search Path				X	X		X
Rx Spectral Table	String	Search Path				X	X		X
Spectral Analysis Resolution BW	Float	Range	100kHz			X	X		X
Clock Analysis	String	List	No				X		

Figure 9: Simulation Parameters

Step 3: Simulate the Schematic Sheet

The schematic sheet, or sheets, must be simulated so that Serial Link Designer can generate the necessary s-parameter models for COM. Only network analysis has to be run to create the models, but statistical and time domain can be run if desired (See Figure 10).



Project: COM_25G_Project
Interface: 25G_Backplane
Reference Schematic Set: Component_Testing

□ Stop On Error
□ Backup Before Deleting Data

□ Validate
☑ Generate Netlists
☑ Include Statistical Analysis
□ Include Time Domain Analysis
□ Run SPICE
☑ Perform Channel Analysis
□ Display Results Spreadsheet
☑ Autoload Results
● All Sheets ○ Current Sheet

Figure 10: Simulation Dialog Window

Step 4: Launch COM Interface

After simulations complete COM can be run directly from the GUI. The COM interface can be accessed under “Tools” à“Run COM Interface” (See Figure 11). The Serial Link Designer interface directly invokes MATLAB and the COM application.

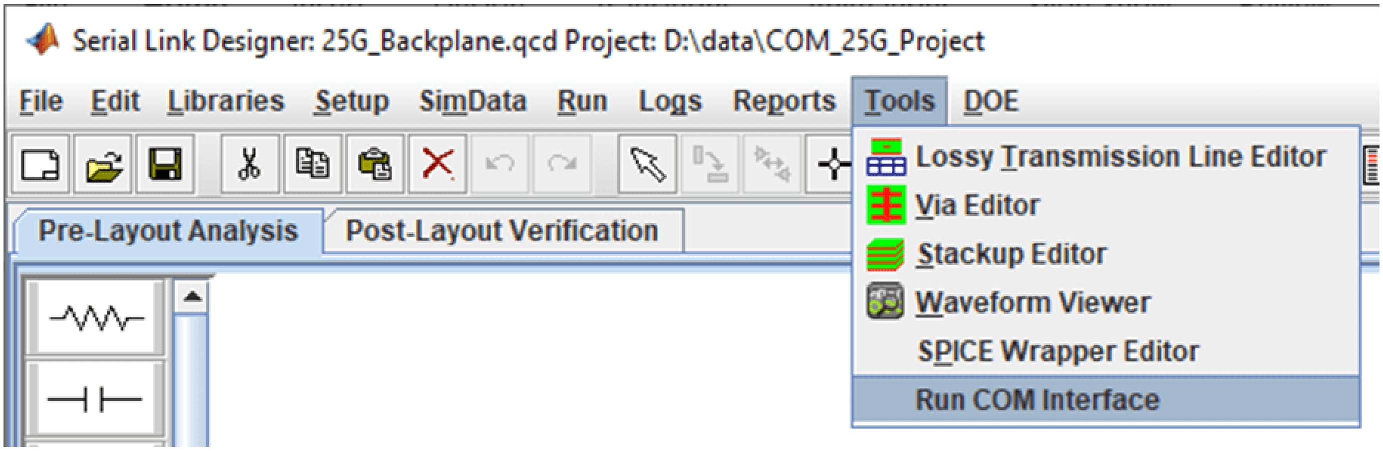
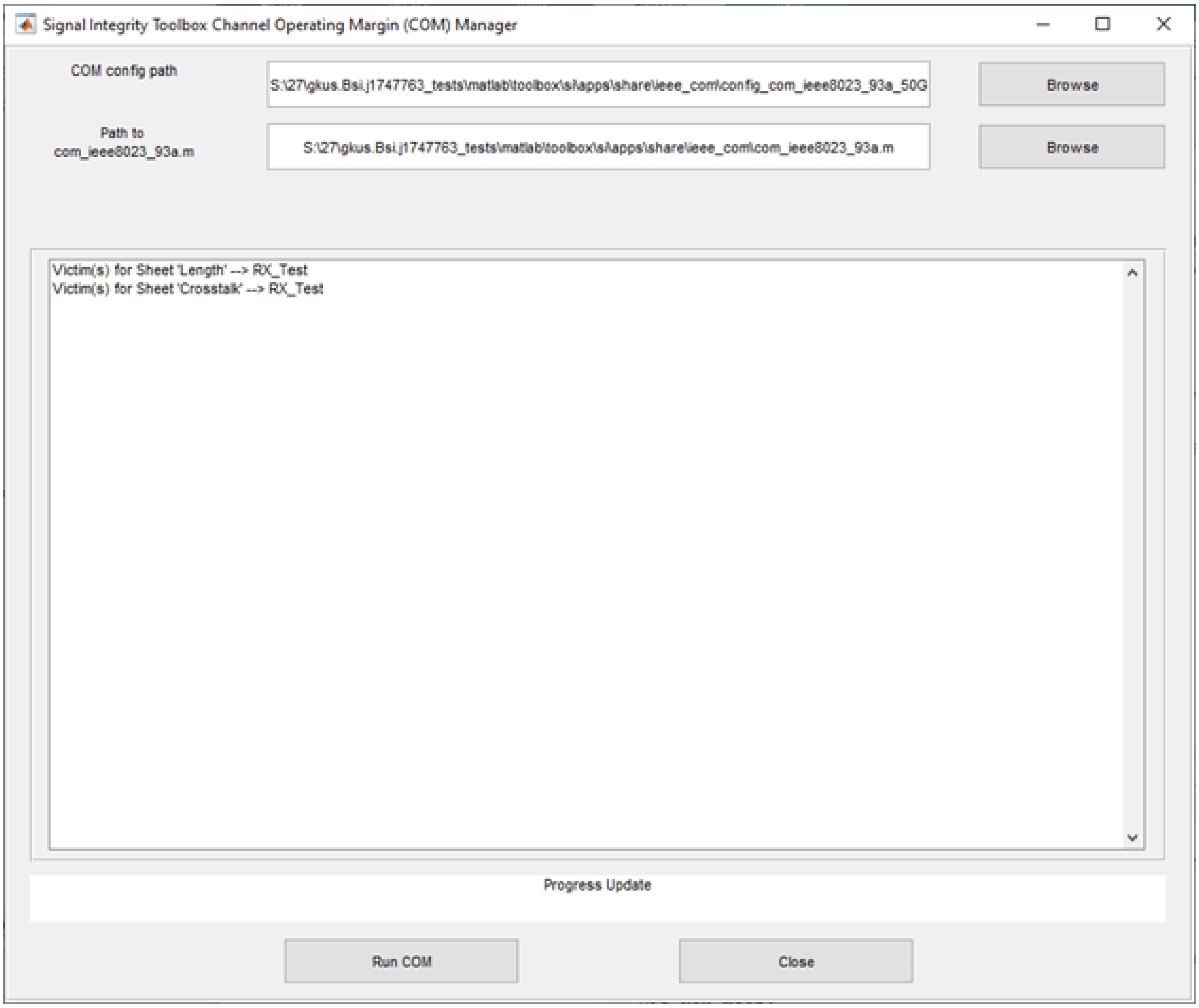


Figure 11: Launch COM Analysis

Step 5: Setup COM

Once the MATLAB application starts, the user will be asked to select the COM configuration spreadsheet and the COM code (Figure 12). The spreadsheet being referenced is the one containing the COM parameters that was configured in Step 1. For this example the file is located in the folder “100GBASE_KR\si_lib\COM\” and is in Excel (.xls) format. The spreadsheet can be kept anywhere on a computer or network as long as it is accessible when browsing the system or network. The second piece of information is the location of the most recent version of the MATLAB COM code “com_ieee8023_93a.m”. This is the MATLAB COM script and is located with the installation of Signal Integrity Toolbox.


Signal Integrity Toolbox Channel Operating Margin (COM) Manager
Browse
Browse
Progress Update
Run COM
Close

COM config path: S:\27\gkus.Bsi.j1747763_tests\matlab\toolbox\si\apps\share\ieee_com\config_com_ieee8023_93a_50G

Path to com_ieee8023_93a.m: S:\27\gkus.Bsi.j1747763_tests\matlab\toolbox\si\apps\share\ieee_com\com_ieee8023_93a.m

```
Victim(s) for Sheet 'Length' --> RX_Test
Victim(s) for Sheet 'Crosstalk' --> RX_Test
```

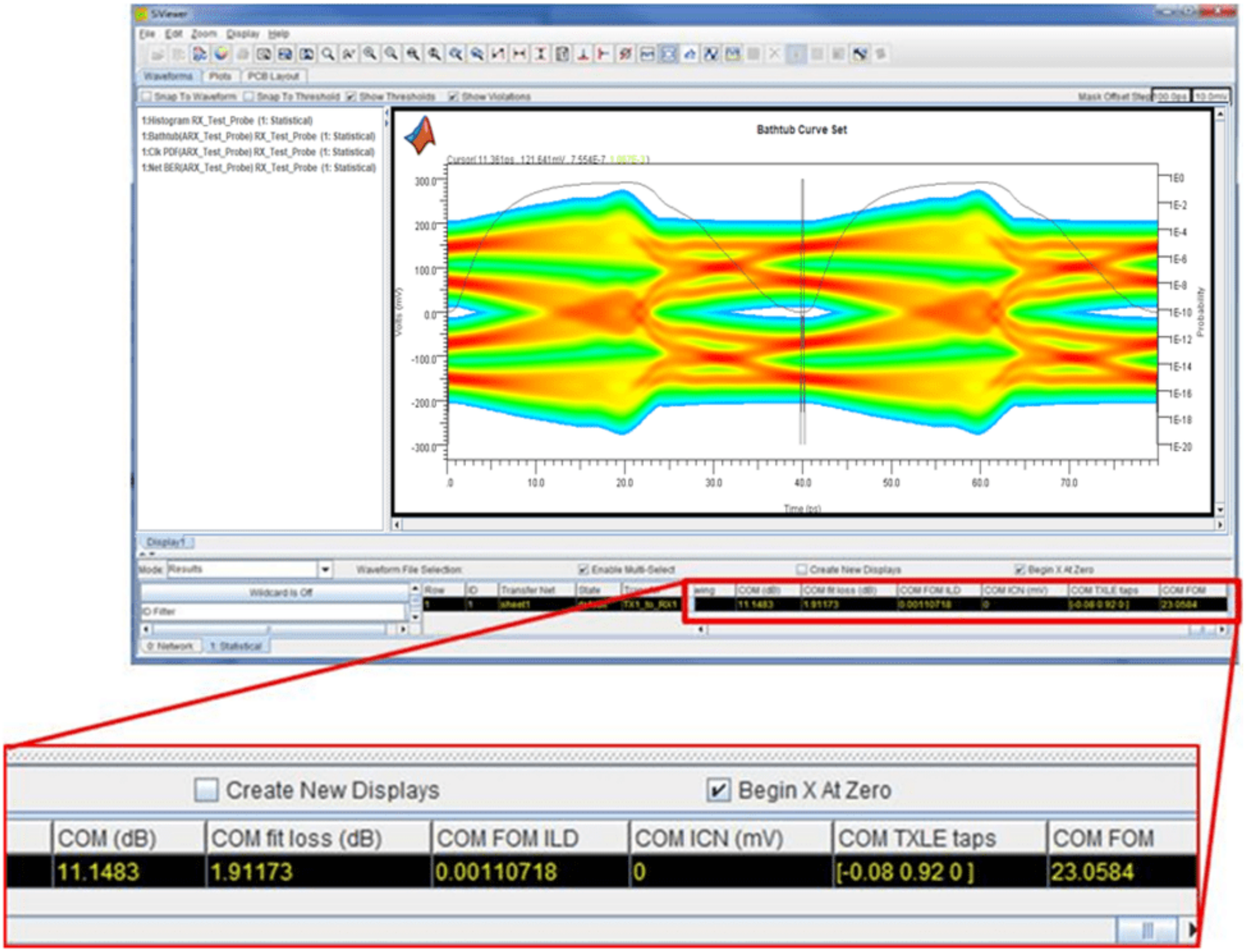
Figure 12: Spreadsheet and Code Selection Window for COM simulation.

Step 6: Run COM Script

Click the “Run COM” button. As the code runs the status is reported in the MATLAB Command Window.

Step 7: View COM Results

Once the COM simulation is complete, the Signal Integrity Viewer will open automatically, and the results of COM and the previous simulation results will be loaded. The network, Statistical and Time Domain tabs in the SiViewer all contain the selected results from the COM simulation (See Figure 14).



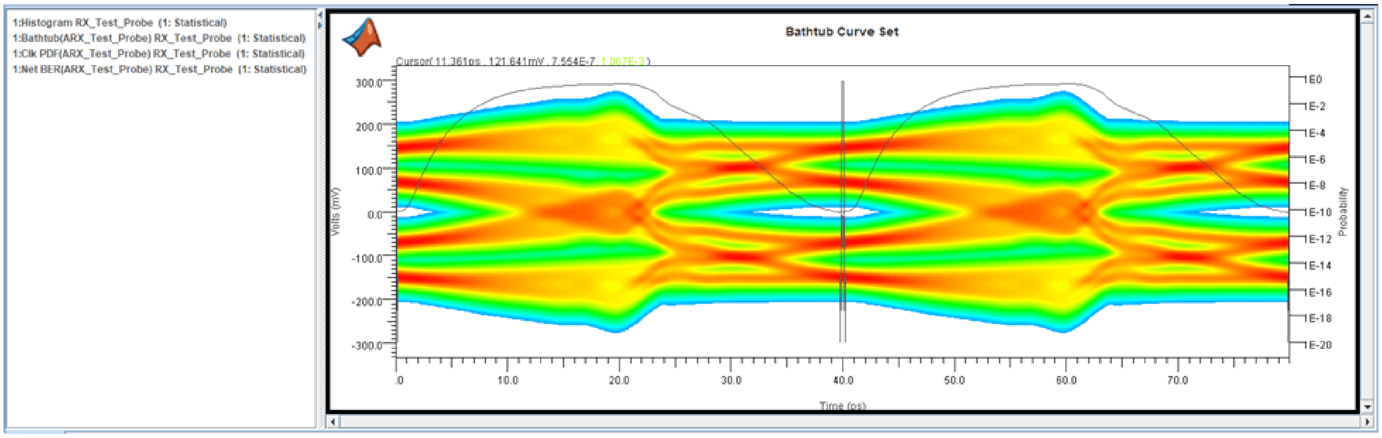
□ Create New Displays ☑ Begin X At Zero

COM (dB)	COM fit loss (dB)	COM FOM ILD	COM ICN (mV)	COM TXLE taps	COM FOM
11.1483	1.91173	0.00110718	0	[-0.08 0.92 0]	23.0584

Figure 14: COM Results Loaded in SiViewer Window

Plotting COM Results:

Signal Integrity Viewer offers many capabilities for viewing simulation results. When analyzing a channel design the user may want to look at eye diagrams, loss plots or noise characteristics. Waveform mode allows the user to view data as a function of frequency or time. Some typical results viewed in waveform mode would be the eye diagram of the signal along with its respective bathtub curves and clock PDF (Figure 15). Another may be an insertion loss versus frequency against a compliance mask as shown in Figure 16.



1:Histogram RX_Test_Probe (1: Statistical)
1:Bathtub(ARX_Test_Probe) RX_Test_Probe (1: Statistical)
1:Clk PDF(ARX_Test_Probe) RX_Test_Probe (1: Statistical)
1:Net BER(ARX_Test_Probe) RX_Test_Probe (1: Statistical)

Figure 15: Statistical Eye Diagram with Bathtub Curves and Clock PDF

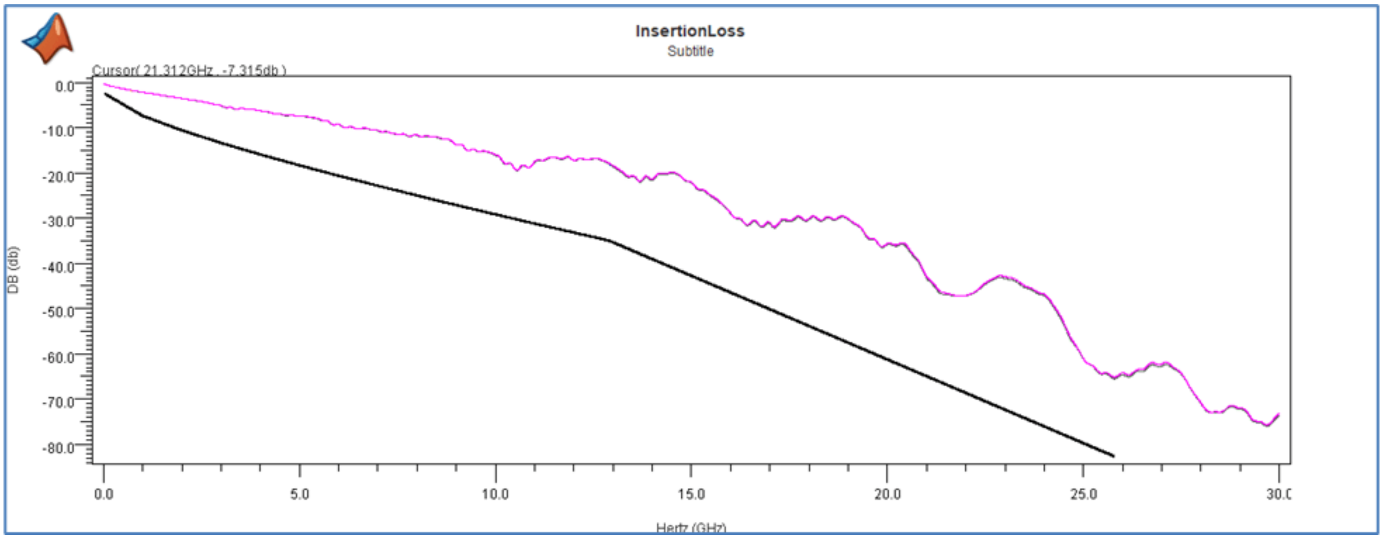


Figure 16: Insertion Loss against a Compliance Mask (Black Line)

The channel COM and other results reported by the COM code are given as single data points. The COM result in particular, as being only a single value, makes it easy to determine the pass/fail behavior. However, if one wants to do investigations into the dependent and independent variables of the simulation results, as they pertain to COM, special plotting capability is needed.

The SiViewer has powerful plotting features when “Plots” mode is selected. This mode allows one to uniquely analyze results and create custom plots of virtually any parameter, variable or result from the simulation. It is an invaluable feature for analyzing large numbers of simulations with many variables. Using “Plots” mode gives the user the power to define multiple variables and plot them against each other on the X or Y axis. Figure 17 shows how to access plots tab in the SiViewer tool. Using this advanced visualization technique one can gain greater insight on the channel or system design especially with very large databases. Identifying trends and finding outliers in the results along with custom plots creation can be the key to a successful design.

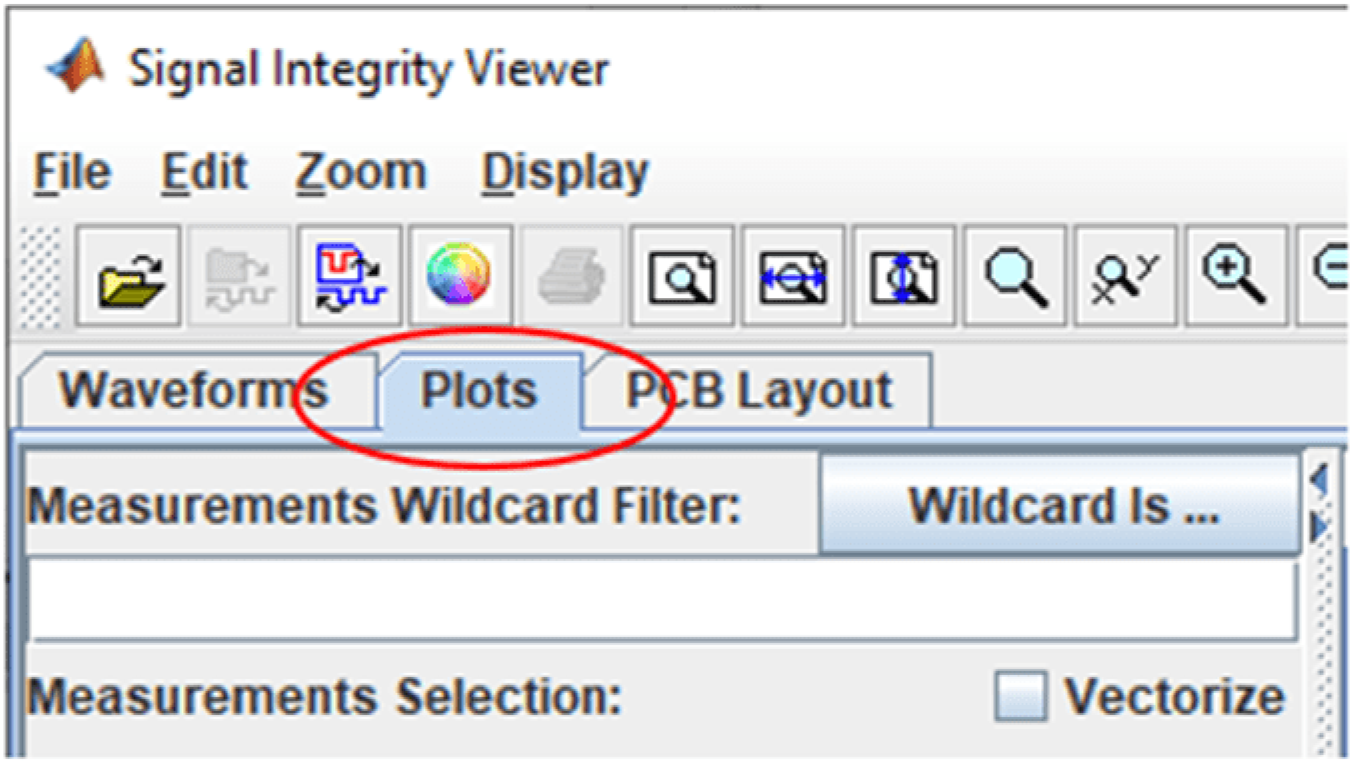


Figure 17: Signal Integrity Viewer Plots Tab

Investigating COM Results:

The example project is set up to demonstrate sweeping backplane trace length and line card aggressor spacing. The first case examined is a sweep of the backplane trace length ("W1" in the solution space) to see how the insertion loss of the channel affects the reported COM value. This is done in the project schematic sheet entitled "Length". The second case is the variation of the aggressor spacing of two of the PCB traces in the channel (W3 and W7) to observe the effect of coupled noise on the victim channel with respect to the reported COM value. This schematic sheet is named "Crosstalk". Once Serial Link Designer and COM simulations had finished, the reported COM and statistical BER were compared. The results show an interesting relationship.

As a side note, part of the COM calculation is the determination of optimal equalization settings for the TX and RX with respect to the channel. COM outputs the tap values and from this representative settings were used TX in the statistical simulations. The RX AMI model used in the simulation has an auto adapt feature for both DFE and CTLE. This feature was used in lieu of extracting fixed tap settings from the COM RX adaptation.

Figure 18 is a plot of COM versus the backplane length. In the plot it can be seen that as backplane length (W1) is increased from 8 inches to 20 inches (X-Axis), the COM value decreases from approximately 3.95dB to approximately 2.8dB. The COM 802.3bj compliance requirement for a 100GBASE-KR4 application is 3dB which is marked by the horizontal line on the plot. The data shows that a backplane length up to approximately 18 inches would meet the compliance requirement for COM assuming all other variables in the channel remained constant.

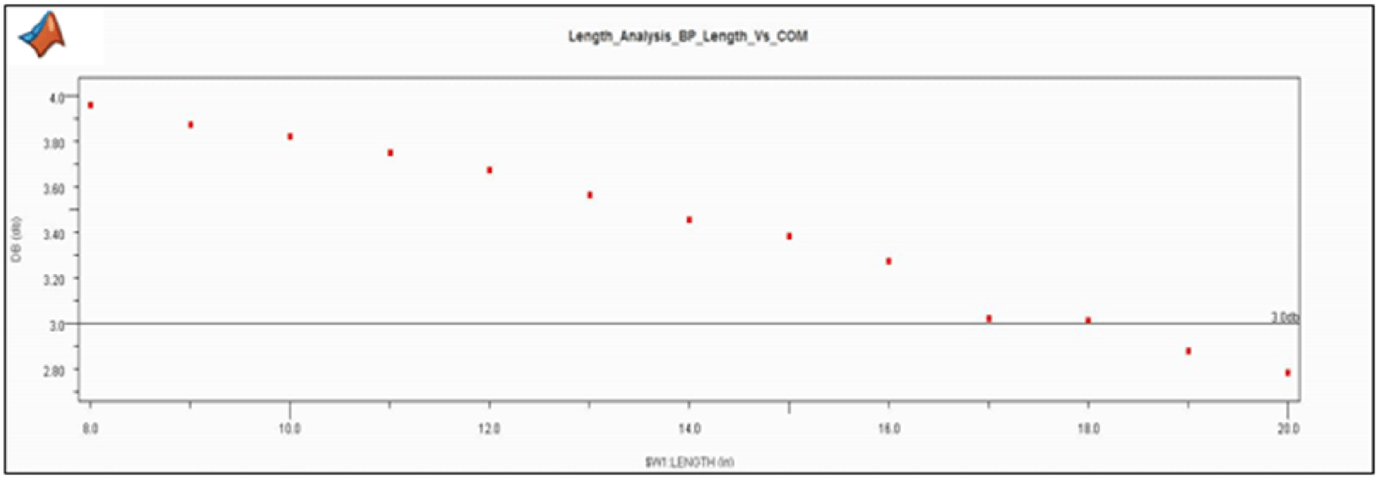


Figure 18: COM versus Channel Insertion Loss

The statistical BER reported by Serial Link Designer for the sweep of backplane length is shown in Figure 19. The BER limit of 1E-12 is marked horizontally on the plot. This plot reveals that backplane lengths up to 15 inches would meet the BER requirement. The plot also reveals some possible resonances in the channel that affect the BER between 8 inches and 12 inches where the BER actually goes down as the length is increased. This behavior is does not exhibit itself in the COM values reported in Figure 18.

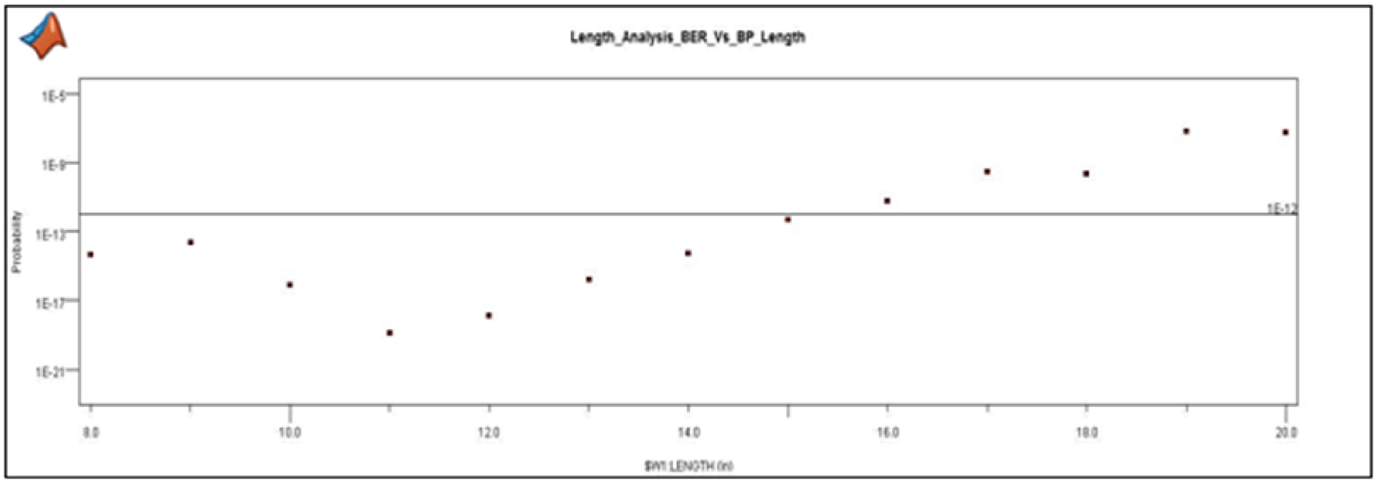


Figure 19: BER versus Backplane Length

Figure 20 is a comparison of BER and COM for each case. An interesting observation is that the BER limit is reached at COM values between 3.4dB and 3.3dB. One possibility for this would be that the TX and RX equalization of the IBIS AMI models in the simulation could be better optimized. Another possibility is accounting for the receiver sensitivity and calculating a BER based on different eye contour requirements.

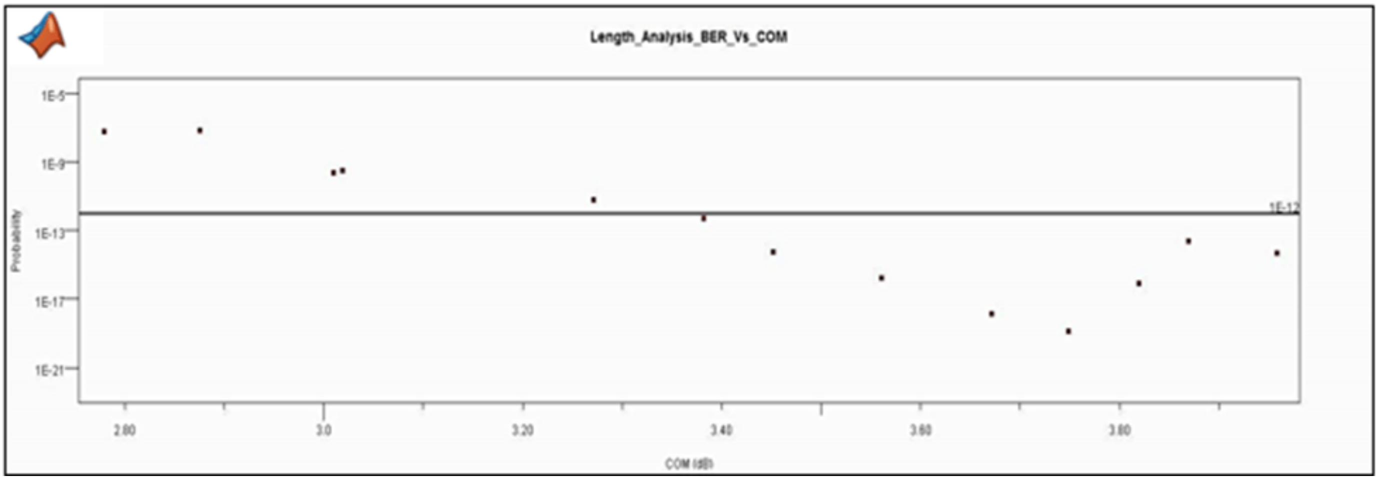


Figure 20: BER versus COM

Figure 21 is a plot of COM versus the aggressor spacing of the trace on either side of the AC coupling capacitors (W3 and W7 on the schematic sheet). The spacing between the victim and each of the aggressor channels was varied from 20mil to 40mil in 5mil increments and is plotted on the X-axis. The horizontal marker represents the COM limit of 3db. The data shows that aggressor spacing of 30mil and greater in the solution space meet the COM limit. Following the trend one could extrapolate that 26-27mil would probably be right at the 3dB line.

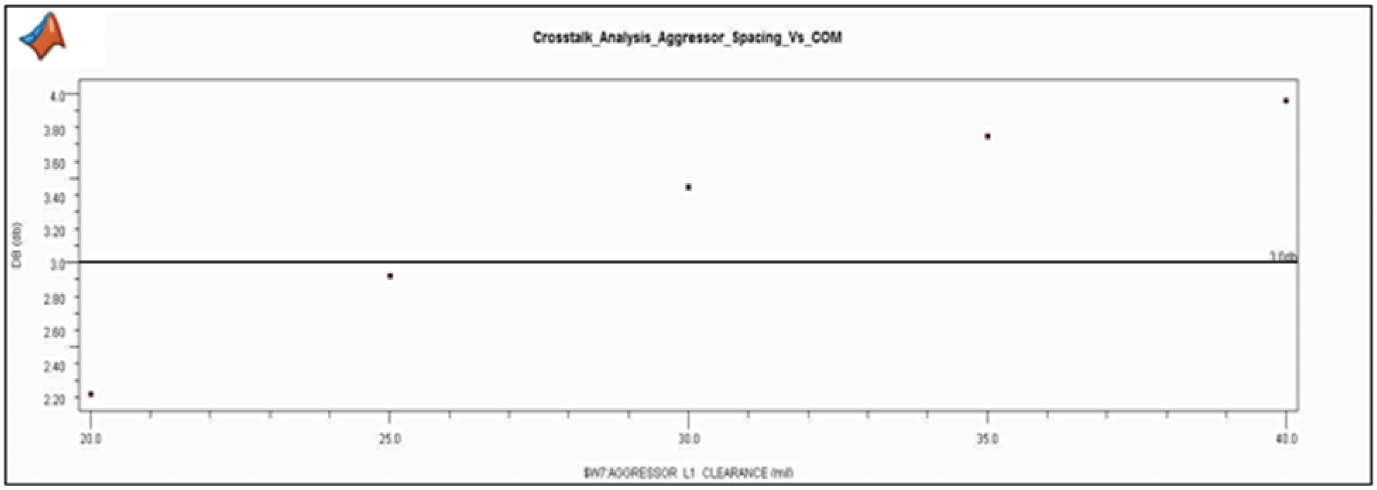


Figure 21: COM versus Aggressor Spacing (Line represents COM limit, above the line passes)

Looking at statistical BER versus the aggressor spacing (Figure 22) one can see a very similar trend to Figure 21 where aggressor spacing above 30mil meet the BER requirement of 1E-12. Although the overall results agree the 30mil data point is a slight outlier from the trend. A straight line extrapolation would reveal that 30mil spacing would not meet the target BER.

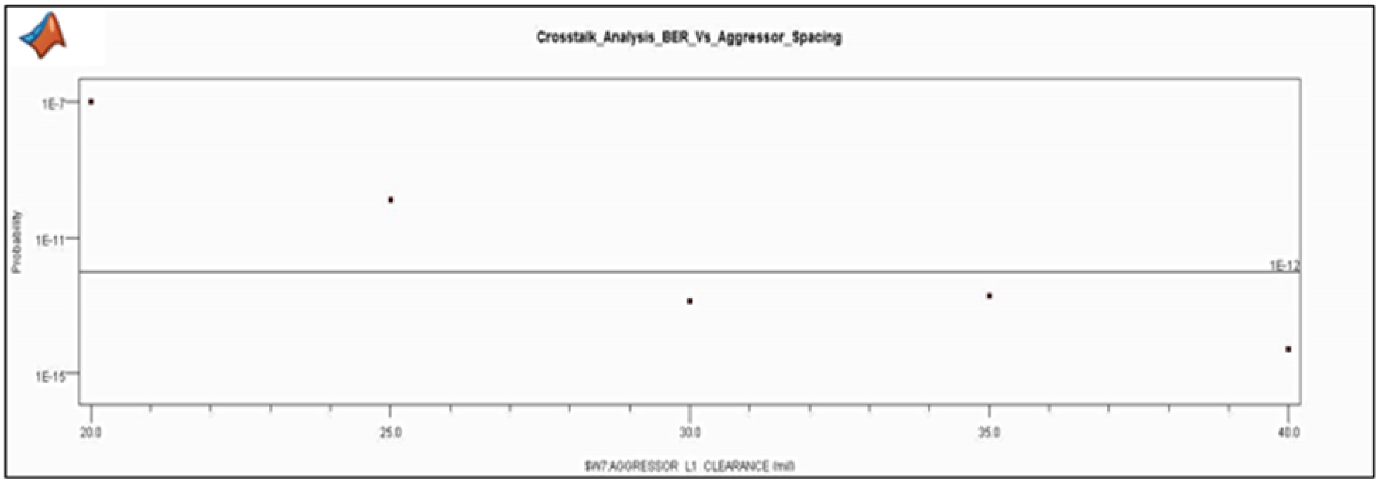


Figure 22: Channel BER versus Aggressor Spacing (Line represents BER limit, below the line passes)

Lastly, Figure 23 shows BER versus COM for the aggressor spacing variation of W3 and W7. The data is plotted from left (40mil spacing) to right (20mil spacing). The 1E-12 BER limit is marked horizontally and any result below the horizontal marker would meet the requirement. The plot shows that COM and statistical BER agree, however COM results below 3.3dB would not appear to meet the target BER. This is consistent with the findings of the backplane length variation.

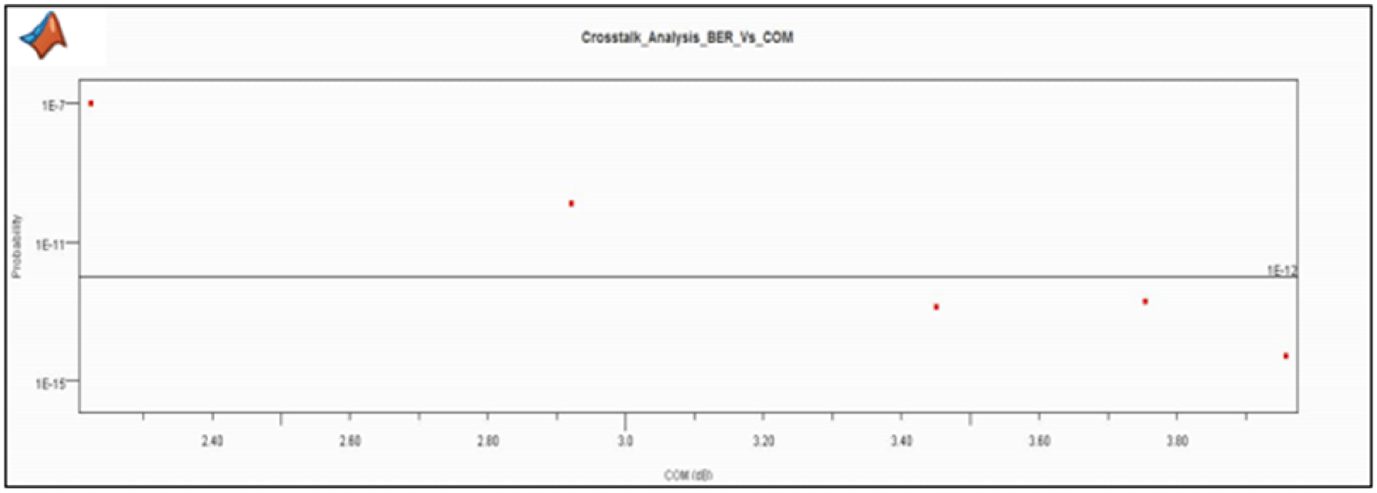


Figure 23: Channel BER vs. COM (Horizontal line: BER 1E-12, Vertical line: COM 3dB)

COM returns many other metrics which can be very useful when evaluating performance or diagnosing a problem with the channel design. Going into further detail of these is beyond the scope of this document. So the reader is encouraged to read through the 802.3bj specification and the documentation that is provided with the COM code. These documents show how COM results are calculated and reported which users can utilize those which may be helpful in the design or debug of their channel.

Summary:

This application note and associated kit provides the information necessary to use the IEEE 802.3bj COM application from within the Serial Link Designer App. This document suggests that although a channel meets COM it must still be simulated to determine if the TX and RX can provide enough equalization to meet the target BER. Meeting COM is essential for 802.3bj compliance and it can also add very useful metrics when incorporated in a channel design methodology.

References:

- IEEE Standard for Ethernet: Amendment 2: Physical Layer Specifications and Management Parameters for 100 Gb/s Operation Over Backplanes and Copper Cables.
- 802.3bj Specification: 802.3bj_2014.pdf
- COM Quick Guide for April 2014: R. Mellitz (Intel Corp.), Adee Ran (Intel Corp.)
- mellitz_3bj_01_0414.pdf
- COM Configuration Documentation: config_com_ieee8023_93a_doc.pdf

See Also

More About

- “Channel Operating Margin (COM)” on page 12-11

Configure Parallel Link

Simulation Parameters Used in Parallel Link Design

You can set parameters that control how a simulation is run in **Parallel Link Designer** using the Simulation Parameters dialog from the **Setup** > **Simulation Parameters** menu item. This dialog contains a table with parameters, their values, and the part of the analysis flow they affect. You can sort the columns by clicking on the table headers.

Non-STAT Mode SPICE Simulation

These parameters affect the SPICE simulation in non-STAT mode.

Parameter	Description
Rise Time	Edge time of the stimulus input to the driver in the SPICE simulation. It can be overridden on a model-by-model basis. The default is 100 ps.
Tran Extension	Add time to the simulation. The default simulation time in the SPICE ".tran" statement is one bit time plus 4 ns past the last transition in the stimulus. This is to ensure that the receiver transition of the last stimulus transition will occur within the simulation time. For very long interconnects of approximately 24 inches or more, you may need to extend simulations further. The default is 0 ns.
Tran Time Step	Initial time step in the SPICE ".tran" statement and the plotting time step for pre-layout and post-layout transfer net simulations. The default is 20 ps.
Max Tran Time Step	Maximum time step to use during analysis. The default is 20 ps.
Stdload Tran Time Step	Initial time step in the SPICE ".tran" statement and the plotting time step for standard load simulations. The default is 20 ps.

STAT Mode SPICE Simulation

These parameters affect the SPICE simulation in STAT Mode. STAT Mode is a simulation mode that uses a statistical engine to perform network characterization, statistical and time domain simulations. For more information, see “Using STAT Mode” on page 7-6.

Parameter	Description
STAT Rise Time	Edge time of the stimulus input to the driver in the SPICE simulation. It can be overridden on a model-by-model basis. The default is 1 ps.
STAT Tran Time Step	Initial time step in the SPICE “.tran” statement and the plotting time step for pre-layout and post-layout transfer net simulations. The default is 2 ps.
STAT Max Tran Time Step	Maximum time step to use during analysis. The default is 1 ps.

Waveform Analysis Parameters

These parameters affect waveform analysis.

Parameter	Description
Skip Data Edge	Number of edges to skip at the beginning of waveforms of Type Data. Skipped edges are not checked for overshoot or quality violations and are not used for etch delay calculation. The default is 1.
Skip Strobe Edge	Number of edges to skip at the beginning of waveforms of Type Strobe. Skipped edges are not checked for overshoot or quality violations and are not used for etch delay calculation. The default is 1.
Skip Clock Edge	Number of edges to skip at the beginning of waveforms of Type Clock. Skipped edges are not checked for overshoot or quality violations and are not used for etch delay calculation. The default is 6.
Skip Time	Amount of time to skip at the beginning of a waveform before starting waveform processing. No overshoot or waveform quality checks are done in the skipped time and any edges in skipped time are ignored for etch delay calculation. The default is 0 ns. In cases where a pulse width is reported (such as Derating Details), data will be reported for the edge before the first edge skipped. For example, if three edges are skipped there will be data for edge number three in some reports.

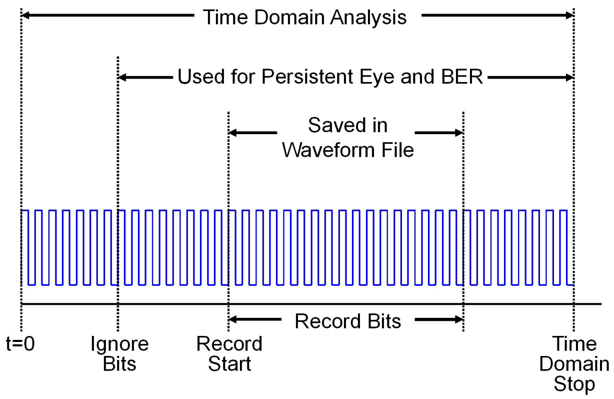
STAT Mode Analysis Parameters

These parameters affect STAT mode analysis.

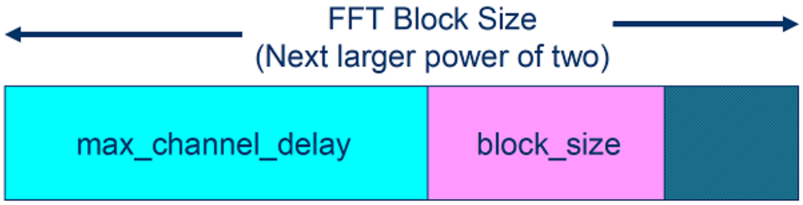
Parameter	Description
Samples Per Bit	The number of time steps in a bit time. Defines the time step used in the STAT Mode ".tran" statement.
Max Channel Delay	User-supplied value for the maximum length of the channel impulse response. Goes into FFT block size calculation, which also defines the message length used for statistical analysis.
Target BER	An array of error rates at which eye height and width are to be measured. The array is sorted smallest to largest on focus change. If fewer than four values are entered the results will include four values, the additional values will be created by multiplying the last value by 1e3.
Record Start	Time at which to start saving waveforms in a STAT Mode time domain simulation.
Record Bits	Number of bits of the waveform to save.
Waveform Analysis Bits	The number of bits from the STAT Mode simulation to use for Waveform Analysis.
Minimum Ignore Bits	STAT Mode time domain waveform analysis will start at this time in the simulation. Allows time for all of the AMI models to reach steady state. This is used if models do not define Ignore Time set in the AMI model, or the defined Ignore Time is less than this value. In other words, the larger of this value or a value from a model is used as the Ignore Bits for the analysis.
Time Domain Stop	The stop time of the STAT Mode time domain simulation.

Parameter	Description
Block Size	The number of samples in a single waveform segment in a time domain simulation. This sets the granularity of the parameter outputs returned by AMI models. Also used in determining FFT block size.
Output Clock Ticks	If yes, then QCD Time Domain Simulation will output the recovered clock ticks to a file.
STATify	Control how statistical techniques are applied to time domain simulations and Getwave-only models. The values are: • TD_Extrapolation: Extrapolates the bathtub curve to account for the effects of ISI at lower probabilities than can be derived from the time domain simulation alone. When this parameter is set to Yes, STAT mode will do the following: • Run a PRBS pattern at the end of the time domain simulation. • Generate a pulse response for the equalized channel from the PRBS data. • Generate a statistical eye from the pulse response. • Use the statistical eye to extrapolate the bathtub curves. For the extrapolation to be accurate the clock recovery loop and DFE (if any) must be settled at the end of the time domain analysis. • Stat_with_Getwave: Uses a PRBS and derived pulse response from time domain analysis as the basis for statistical analysis. Allows statistical analysis to be done for models that are Getwave-only. • Both: Perform both TD_Extrapolation and Stat_with_Getwave. • None: Do not perform TD_Extrapolation or Stat_with_Getwave.
Step Response Type	Step response used by STAT mode. The app supports rising, falling, and dual step responses.
SPICE Ignore Bits	The time before the start of the SPICE step in the STAT Mode step response simulation. It is either in UI or in units of seconds.
Include IBIS Package	Include (Yes) or do not include (No) IBIS Package.

This figure demonstrates the relationship of several STAT Mode time domain simulation parameters.



Two of the STAT mode analysis parameters, Max Channel Delay and Block Size, determine the FFT block size used in network characterization and statistical analysis. The actual FFT block size is rounded up to the nearest power of two.



See Also

More About

- “Specify Corner Conditions in Parallel Link Design” on page 6-6
- “Stimulus Patterns in Parallel Link Design” on page 6-8
- “Model Jitter and Noise While Designing Parallel Link” on page 10-2

Specify Corner Conditions in Parallel Link Design

Corner conditions are used to define process corners. In process corners, the parameters are within the specified range for that parameter but outside the range of normal operations. You can specify corner conditions using the Corners Conditions dialog from the **Setup** > **Corner Condition** menu item.

IC Environment Corners

The IC Environment Corners area contains the temperature parameter for each corner. This will be used as the .TEMP parameter in the SPICE simulations.

Note The temperature parameter does not affect IBIS buffer models.

The voltage factors are used to scale all voltage sources in the netlist. The typical corner value is scaled by the scaling factor to create the values for the slow and fast corners. For voltage sources, the value entered in the schematic or specified for a voltage net in postlayout is scaled by the scaling factor.

I/O buffer voltages can use the three values specified in the IBIS [Voltage Range] parameter for the three corners or use the typical value from the [Voltage Range] and scale it.

Etch Corners

You can use the Etch Corners area to specify scaling factors for the Z0 and Tpd parameters of transmission line models. Scaling factors account for manufacturing variation in the PCB. Both ideal and lossy transmission line models are scaled.

Lossy transmission line models are scaled by computing the values of Z0 and Tpd from the typical corner L and C values. The computed Z0 and Tpd are then scaled by the scaling factors to create the Z0 and Tpd values for the slow and fast corners. The slow and fast corner L and C are computed from the slow and fast Z0 and Tpd.

Impact of Corner Settings

The elements that are affected by corner settings are:

- **I/O buffer voltages**: If scaling is enabled for I/O buffer voltages, the typical value of the IBIS [Voltage Range] parameter is multiplied by the scaling factor for the IC corner selected.
- **I/O buffer data**: The data that is used for each process corner is summarized in Process Corner Model Data Usage.
- **Voltage sources on schematics**: The voltage parameter of the element is multiplied by the scaling factor for the IC corner selected.
- **Voltage nets in post-layout**: The voltage set on the net on import of the board is multiplied by the scaling factor for the IC corner selected.
- **Ideal transmission lines (SPICE T elements)**: The Z0 and Tpd parameters are multiplied by the Z0 and Tpd factors for the selected corner.
- **Lossy transmission lines (SPICE W elements)**: The models without explicit slow and fast corner models in the library are scaled using the Z0 and Tpd factors in Corner Conditions. Models

that have `_te` (typical), `_fe` (fast) or `_se` (slow) appended to the model name are used for the appropriate etch corner if they exist.

- **SPICE subcircuits**: file and subcircuit names can contain {etch} and {corner}. If present, the current corner is substituted.

Process Corner Model Data Usage

IC Process Corner	Model or Setting	Data Used
FF	IBIS buffer in HSPICE	typ=fast HSPICE option
	IBIS buffer in IsSPICE4	IBIS maximum IV and VT data
	HSPICE buffer	HSPICE FF wrapper
	Temperature	FF Temperature from Corner Conditions
TT	IBIS buffer in HSPICE	typ=typ HSPICE option
	IBIS buffer in IsSPICE4	IBIS typical IV and VT data
	HSPICE buffer	HSPICE TT wrapper
	Temperature	TT Temperature from Corner Conditions
SS	IBIS buffer in HSPICE	typ=slow HSPICE option
	IBIS buffer in IsSPICE4	IBIS minimum IV and VT data
	HSPICE buffer	HSPICE SS wrapper
	Temperature	SS Temperature from Corner Conditions

See Also

More About

- “Simulation Parameters Used in Parallel Link Design” on page 6-2
- “Stimulus Patterns in Parallel Link Design” on page 6-8
- “Model Jitter and Noise While Designing Parallel Link” on page 10-2

Stimulus Patterns in Parallel Link Design

You can specify stimulus patterns independently for each transfer net type (Data, Clock and Strobe) or designator using the **Parallel Link Designer** app. To create and manage stimulus patterns, launch the Stimuli dialog box from **Setup** > **Stimulus** from the app toolbar.

The Stimuli dialog box has a table of stimulus patterns with columns for the name, length in bits and description of each stimulus pattern. You can edit, delete, copy, or add new stimulus patterns. There are three default stimulus patterns for each transfer net:

Transfer net type	Stimulus pattern
Data	default_data
	default_data_victim
	default_data_aggressor
Clock	default_clock
	default_clock_victim
	default_clock_aggressor
Strobe	default_strobe
	default_strobe_victim
	default_strobe_aggressor

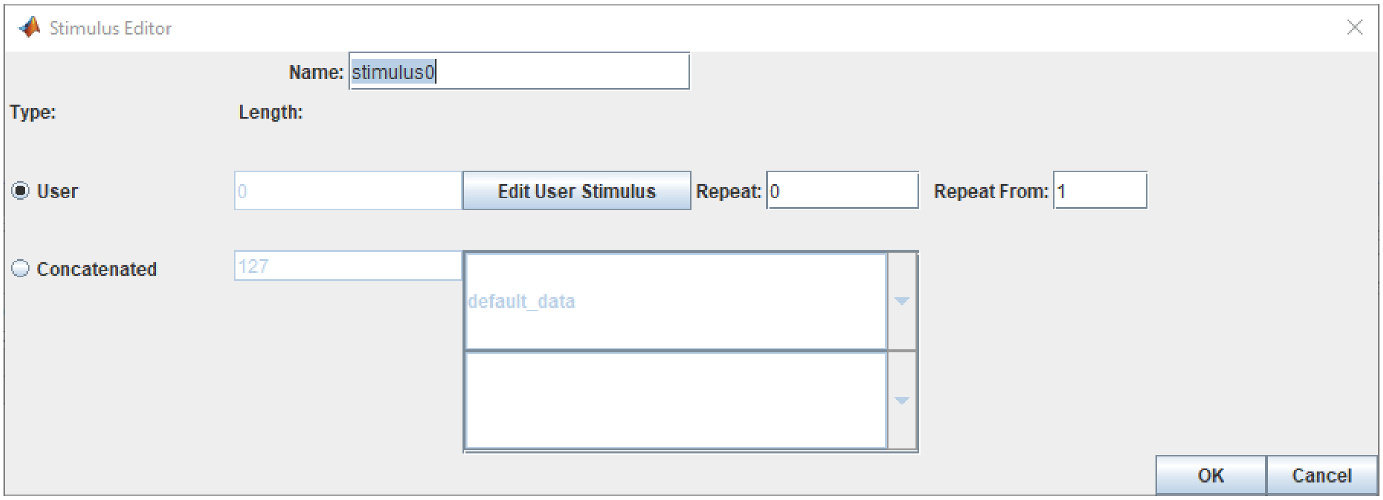
You cannot delete or rename the default stimuli, only edit their patterns. The victim and aggressor patterns are used in pre-layout coupled/SSO (widebus) simulation mode.

The `default_data`, `default_clock` and `default_strobe patterns` are used in pre-layout and post-layout simulations. The Transfer Net Type controls the stimulus used for a simulation. The default stimulus patterns can be edited and can be up to 4000 bits long.

In pre-layout analysis each designator can have its own stimulus. You can create and use new stimulus patterns for individual designators.

Editing Stimulus Patterns

You can create a new stimulus or edit an existing one by using the Stimulus Editor dialog box. To access the Stimulus Editor dialog box, first open the Stimuli dialog box by double clicking on a designator symbol and clicking the **Stimulus** button.


Stimulus Editor

Name: stimulus0

Type:

Length:

● User 0 Repeat: 0 Repeat From: 1

○ Concatenated 127 default_data

A Parallel Link Designer stimulus can be of two types:

- User — user defined series of ones and zeroes.
- Concatenated — one or more User stimulus patterns combined sequentially.

Each stimulus has a name. To create a User stimulus, select **User** and click **Edit User Stimulus** to launch the User Stimulus Editor dialog box.

To create a Concatenated stimulus, select **Concatenated** and then select one or more stimulus names in the table. In the simulation netlist the stimuli is concatenated with the stimulus at the top of the list appearing first in the netlist. The **Length** parameter shows the length of the concatenated stimuli.

User Stimulus Editor

The User Stimulus Editor dialog box is used to create a user defined stimulus. You can directly type in the ones and zeroes in the main window. You can also add a specific number of ones, zeroes or zero-one sequences. After completing entering your desired sequence, click **OK** to return to the Stimulus Editor dialog box. The Stimulus Editor dialog box contains the following parameters about the User stimulus:

- **Length** — the length of the pattern created in the User Stimulus Editor.
- **Repeat** — the number of times to repeat the pattern. For a pattern of length *n*, setting **Repeat** to `0` results in a pattern of length *n*, setting **Repeat** to `1` results in a pattern of length `2`*n* and so on.
- **Repeat From** — the bit position to repeat from. Bit 1 is the first bit in the pattern.

PDA Stimulus

You can use the **Parallel Link Designer** app to determine the worst-case pattern through Peak Distortion Analysis (PDA) and use that stimulus for a designator by selecting **PDA** in the Designator Element Properties dialog box. When PDA stimulus is selected, the **Generate SPICE** process during simulation first creates step response simulations. The step response simulations are processed to find the worst-case pattern. The normal Transfer Net simulations are then generated using the PDA pattern as the stimulus.

The Generate SPICE Log (**Logs** > **SI/Timing SPICE Generation Log**) and Generate SPICE Report (**Reports** > **SI/Timing SPICE Generation Report**) have details and any errors and warnings from the process. The log is also included in the Errors & Warnings dialog box.

Using Stimulus Pattern

To specify a stimulus pattern (other than the default) on an individual designator basis, open the Designator Element Properties dialog box by double clicking on any designator in the Pre-Layout Analysis tab. You can also access the dialog box by clicking on the **Properties** button in the Transfer Net Properties dialog box in the pre- or post-layout. In the Designator Element Properties dialog box, select the desired stimulus pattern from the Stimulus drop-down menu.

See Also

More About

- “Simulation Parameters Used in Parallel Link Design” on page 6-2
- “Specify Corner Conditions in Parallel Link Design” on page 6-6
- “Model Jitter and Noise While Designing Parallel Link” on page 10-2

Pre-Layout Analysis of Parallel Link

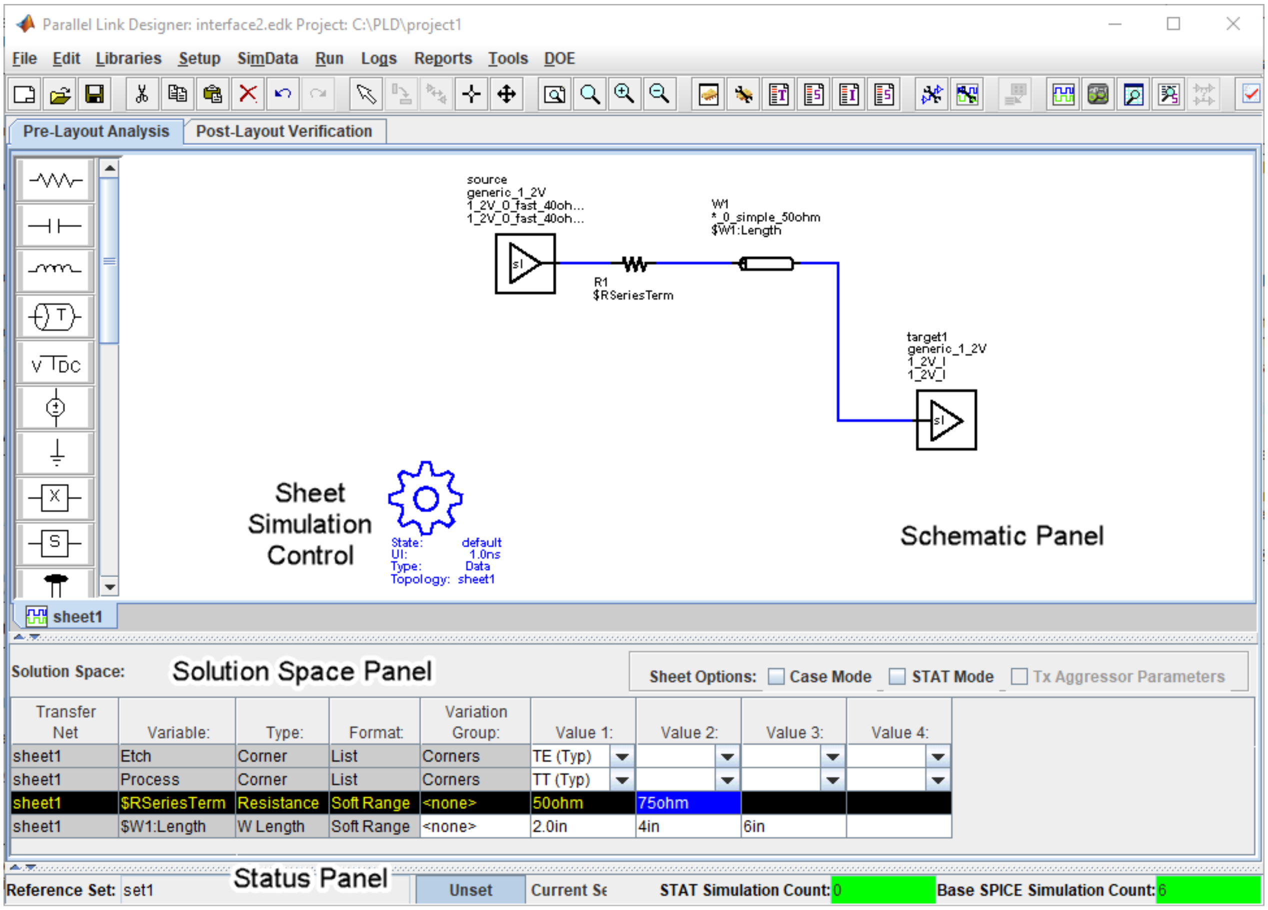
Pre-Layout Analysis of Parallel Link

Pre-layout analysis provides you with an integrated signal integrity, timing and crosstalk analysis environment to determine system-level noise and timing margins. The pre-layout analysis environment is used to generate design guidelines for your board layouts, package layouts, connectors and cabling. From the Pre-layout tab, you may perform simple or complex solution space analysis by varying elements, such as: topology, termination, voltage, temperature, process (silicon and etch), models, UIs, corner conditions, populations, and coupling.

A schematic represents an uncoupled net or a coupled net. Uncoupled nets can be thought of as net classes. The **Parallel Link Designer** app stores this information as a transfer net, which is used as the underlying data structure for all of the analysis. The transfer net data can be re-used in post-layout and other projects.

The Pre-Layout Analysis tab consists of three major panels:

- *Schematic Panel* —This is where you graphically create and edit the circuit schematic. You can also define the data from the sheet simulation control settings.
- *Solution Space Panel* — This is where you enter your solution space values for performing parameter sweeps.
- *Status Panel* — This panel displays the simulation counts and schematic set information.



State: default
UI: 1.0ns
Type: Data
Topology: sheet1

Solution Space:

Sheet Options: □ Case Mode □ STAT Mode □ Tx Aggressor Parameters

Transfer Net	Variable:	Type:	Format:	Variation Group:	Value 1:	Value 2:	Value 3:	Value 4:
sheet1	Etch	Corner	List	Corners	TE (Typ)			
sheet1	Process	Corner	List	Corners	TT (Typ)			
sheet1	$RSeriesTerm	Resistance	Soft Range	<none>	50ohm	75ohm		
sheet1	$W1:Length	W Length	Soft Range	<none>	2.0in	4in	6in	

Reference Set: set1 Unset Current Se STAT Simulation Count: 0 Base SPICE Simulation Count: 6

Double clicking a symbol on a schematic sheet launches an Element Properties dialog box for that symbol type. Each symbol type has a unique set of properties that are set from the Element Properties dialog box. If the properties are parameters that can be swept, that is also controlled from the Element Properties dialog box.

Schematic Elements

Designator — The I/O buffer is represented by a designator in the schematic. A schematic must have at least one designator that can be a driver. The buffers can be single-ended or differential. Buffer symbols has a default I/O buffer model after being placed on the schematic. You can change the buffer model for a designator in three different ways: from the Edit Designator Properties dialog, from the Select IBIS File & Model dialog, and from the default model menu items. IBIS files must be imported into the libraries before they can be used. HSPICE models must be wrapped and put in the libraries before they can be used.

Transmission Line — There are two types of transmission lines: ideal transmission lines and lossy transmission lines. Ideal transmission line models have two parameters: Impedance (Z_0) and delay (T_{pd}). Lossy transmission lines have a frequency dependent RLGC model that is created by a 2-D field solver. Lossy transmission lines can be single-ended or differential..

Via — You can create via models based on a stackup and via physical parameters. Via models can be single-ended or differential. The first time a via symbol is placed on a sheet the default stackup is created. A dialog launches to allow the number of signal layers in the default stackup to be specified.

S-Parameters — You must import the S-Parameter files into the **Parallel Link Designer** app before you can use them in schematic sheet. After a symbol has been placed on a schematic, the port map can be edited by right clicking on the symbol and selecting Edit Port Map from the menu.

Passive Subcircuits — You must manually import the SPICE subcircuit models for passive elements in the **Parallel Link Designer** app libraries before you can place them on the schematic.

Probe — Voltage probe can be single-ended or differential. When a probe symbol is placed on a schematic it automatically creates a waveform node in the waveform file at the probed location. The waveform at the node can be viewed in the **SI Viewer** app.

Solution Space

The Solution Space panel is used to create parameter sweeps. There are variables that are always part of the solution space. Other variables in the table are created when parameters are set to be swept. The values can be typed into fields, lists or range/steps depending on the variable type.

The solution space panel can be in one of two modes:

- *Permutation mode* — Each row is treated as an independent variable unless they are in the same variation group. The number of simulations represented by the solution space is all of the combinations of all of the variable values.
- *Case mode* — Each column represents a simulation case. The number of simulations represented by the solution space is the number of columns.

Sheet Simulation Control

You can specify the specify the simulation state, unit interval (UI), topology, transfer net type, AC noise type, and the number of aggressors for SSO/coupled mode analysis of each schematic sheet using the sheet simulation control symbol.

See Also

More About

- “Customize Parallel Link Project for Pre-Layout Analysis” on page 7-5
- “Results of Pre-Layout Analysis in Parallel Link” on page 7-8

Customize Parallel Link Project for Pre-Layout Analysis

You can modify the schematic elements to customize your designs in the **Parallel Link Designer**. app.

Using I/O buffers

An I/O buffer is represented by a designator. You change the buffer model for a designator in three ways:

- *Edit Designator Part/Pins dialog box*

  Right clicking on the designator and selecting **Edit Designator Part/Pins** opens the Edit Designator Part/Pins dialog box. The **Designator** parameter allows the designator name to be changed. The **Part Name** parameter lists the parts in all libraries. When a specific part is selected in the dropdown menu, the IBIS file name referenced by that part is shown in the **IBIS File** parameter. The IBIS component name for the selected part is shown in the **IBIS Component** parameter. The table on the left shows all of the pins in the IBIS component. To associate a pin or pins with the designator select the pin or pins on the left and click one of the arrow buttons between the two tables. The pins in the table on the left can be filtered using the **Wildcard Filter** parameter. To add a column that shows the name of the transfer net that uses the pins, select **Generate Used Pin Information**.
- *Select IBIS File & Model dialog box*

  Right clicking on the designator and selecting **Select IBIS Model and File** opens the Select IBIS File & Model dialog box. You can select an IBIS file from the table provided, or import your own. You can also select one or more pins from the table of pins in the selected IBIS files.
- *Default model*

  To assign a default model to a designator, right click on the designator and select **Use Default Driver**, **Use Default Receiver** or **Use Default I/O**.

Using Transmission Lines

The app uses two types of transmission lines:

- *Ideal transmission lines*

  Ideal transmission line models have two parameters: Impedance (Z0) and delay (Tpd). These parameters are set from the Element Properties dialog box for ideal transmission lines. Double click on an ideal transmission line symbol on the schematic to launch the Element Properties dialog box. There are columns for Impedance and Delay/Distance and checkboxes to sweep the parameters. Checking a sweep checkbox creates a variable in the solution space for the parameter.

  The model on the schematic is the model for the typical etch corner. If other etch corners are simulated the Z0 and Tpd parameters are scaled according to the corner conditions specified in the Corner Conditions dialog box. See “Specify Corner Conditions in Parallel Link Design” on page 6-6 for more information.
- *Lossy transmission lines*

- The lossy transmission line have a frequency dependent RLGC model that is created by a 2-D field solver.

  The app has a field solver with a transmission line editor for entering a cross-section. The transmission line editor can be used to create models in the libraries or to edit the model for a symbol.

  To associate a model in the library with the transmission line, right click on the symbol and choose **Select T-Line Model**. You can edit the default model by right clicking on the symbol and choosing **Edit T-Line Model**.

Using Vias

You can create via models based on a stackup and via physical parameters. Via models can be single-ended or differential. The first time a via symbol is placed on a sheet the default stackup is created. A dialog launches to allow the number of signal layers in the default stackup to be specified. For more information, see “Via and Stackup Management in Parallel Link Project” on page 8-9.

Using S-Parameters

S-Parameter files must be imported into the app before being used on a sheet. After importing and adding the S-Parameter to your schematic, you can edit the port map by right clicking on the S-Parameter symbol and selecting **Edit Port Map**.

Using STAT Mode

STAT Mode is a simulation mode that uses a statistical engine to perform network characterization, statistical and time domain simulations. The simulation methodology is derived from the IBIS-AMI specification for performing high speed channel simulations with IBIS-AMI models. STAT mode can also be used to simulate any type of buffer models (IBIS or SPICE) to analyze the response and performance of a network through statistical and time domain analysis.

The app performs network characterization using HSPICE to determine the network's response to a step input it then post-processes that information to derive the network transfer function. The transfer function is used by the statistical engine to determine the statistical eye along with a bit error ratio (BER) and other data. Statistical analysis is based on an LTI (Linear Time Invariant) network assumption along with LTI equalization (if supported by the model).

Time Domain Analysis uses the same network characterization results as statistical along with a bit sequence to derive the output waveform, persistent eye, BER estimate and other data. The persistent eye is the amplitude statistics accumulated from a specific time domain waveform. It is accumulated by triggering using an ideal recovered clock in exactly the same way that an eye diagram is accumulated in a modern digital sampling scope. Unlike statistical analysis time domain analysis is a bit-by-bit simulation that can be used to analyze the network with any non-LTI behavior taken into account.

The STAT Mode control is in the Sheet Options area of the solution space panel.

See Also

More About

- “Pre-Layout Analysis of Parallel Link” on page 7-2
- “Results of Pre-Layout Analysis in Parallel Link” on page 7-8

Results of Pre-Layout Analysis in Parallel Link

The **Parallel Link Designer** app produces one or more reports and logs for each simulation and process you run.

The tabs within a report are organized to aid in the process of progressive discovery. The first tab is the log tab, providing a progress summary of the analysis and its errors and warnings. The other tabs contain summaries of the data and successively more detailed information, letting you track down a particular result to a specific simulation file and transition number or time.

Validation Reports

Validation reports indicate the syntax errors in the data. When relevant, the reports provide the corresponding part name, IBIS file and component names, and timing file and model names.

Report	Description
Validation Summary	Number and location of warnings and errors.
Part Errors	Errors in the part properties file.
IBIS Errors	Syntax errors and omissions in the IBIS files. The report includes the signal name, model name, and number of the pin of the component in the IBIS file, and the IBIS model type for the model in the IBIS file.
Timing Errors	Syntax and consistency errors and omissions in the timing file data.
IBIS Timing Errors	Inconsistencies between IBIS components and timing models data. The report includes information about the pin of the component in the IBIS file, including the signal name, model name, timing model name, number, and I/O type. The report also includes the IBIS model type for the model in the IBIS file.
Coverage Warnings	Parts or pins in parts that are not referenced in the transfer netlist or timing model.
Transfer Net Summary	Details on each transfer net such as whether the type of the net is data, clock, or strobe, whether the net is differential or single-ended, and the number of nodes. This report also lists information on the clock, noise, and probe points.
Part Summary	Details on each part.
Model Overview	Lists every signal integrity, HSPICE, and IBIS parameter or extension associated with each model in the design. This includes model name, corner and mode information, waveform DRC and timing extensions among other parameters.
Part Pin Summary	Summary of part transfer nets and timing pin definitions.
Differential Pin Summary	Lists the differential pins and components associated with each part.
Timing Delay Summary	Summary of all output delays and setup and hold statements in each timing model.
Model Details	Lists most of the waveform DRC rules and timing levels used by the product. The report includes the actual parameter used (following the precedence rules) and the value assigned to that parameter.

Report	Description
Transfer Net Errors	Inconsistencies between transfer nets, IBIS components and timing models. The part, IBIS and timing files listed are not necessarily where the error occurred, but simply a listing of all files involved in the error checking.

Waveform and Timing Report

The waveform and timing report summarizes the waveform analysis and timing results for both pre-layout and post-layout simulations.

Report	Description
Waveform Summary	Number of errors and warnings found during waveform analysis.
Waveform Fatal	Lists any fatal waveform error found on any edge during waveform processing. Fatal errors are errors that cause the inability to generate any waveform or timing data at all. This tab will only appear if there are fatal violations of the DRC rules.
Waveform Quality	Lists violations of waveform rules as applied to each edge. The product applies a number of waveform rules to each edge to verify that the transition meets various IC vendor AC specs including edge rate, ringback and monotonic (clock nets). If the transition violates any of these rules, the timing of the transition may be suspect.
Waveform Overshoot	Lists violations of these waveform overshoot rules. Overshoot does not affect the signaling operation of an I/O buffer but can affect the lifetime of an IC. Overshoot can occur in two ways: when the waveform instantaneously exceeds absolute overshoot limits set in the IBIS model, and when the waveform exceeds a lesser overshoot voltage limit for more than a prescribed time.
Eye Rollups	Lists a summary of eye details for each node in each transfer net.
Eye Details	Eye information for each receiver node in each simulation.
Derating Details	Details of slew-rate derating calculations. This tab will be present if one or more models contain slew rate derating tables.
Statistical	Variables and results from the statistical analysis simulation (STAT mode only).
Time Domain	Variables and results from the time domain analysis simulation (STAT mode only).
Xtalk Contours	Crosstalk and eye heights of the widebus sheets that have been simulated.
Waveform Margin by TNET	Summary of the waveform margins for each transfer net.
Waveform Margin by Variation	Summary of the waveform errors (if any) and waveform DRC margins associated with each simulation.
Model Overview	Summary of the data for each IBIS model used in the simulation. The report includes the measurement thresholds and the parameters that are used for each threshold.

Report	Description
Mask By Channel	Available when the analysis is defined using TiVW and ViVW (DDR4 and DDR5).
Mask By Receiver Corner	Available when the analysis is defined using TiVW and ViVW (DDR4 and DDR5).
Mask by Driver Receiver Corner	Available when the analysis is defined using TiVW and ViVW (DDR4 and DDR5).
Mask Training Details	Available in post-layout when the analysis is defined using TiVW and ViVW (DDR4 and DDR5).
Mask Eye Details	Available when the analysis is defined using TiVW and ViVW (DDR4 and DDR5).
Timing	Rolls up the By Variation Details tab by combining all transitions in the same transfer net.
By Transfers	Rolls up the By Variation tab by combining identical transfers (same driver and receiver).
By Variation	Includes setup margin, hold margin, etch delay, AC noise, transfer net, and extended net details.
By Variation Details	Contains the setup and hold margins for both rising and falling edges at each receiver in each simulation.
By Variation Details Summary	Available only in post-layout. This tab contains two rows for each transfer net in the By Variation Details Summary Tab. One has the smallest setup margin for that transfer net, the other has the smallest hold margin for that transfer net.
By Driver	Rolls up the By Variation tab by combining identical drivers.
By Receiver	Rolls up the By Variation tab by combining identical receivers.
Synchronous Details	Contains the setup and hold margin for rising and falling data edges in each simulation.
Source Synchronous Details	Contains the setup and hold margin for rising and falling data edges in each simulation.
Dynamic Clock Skew	Lists the skews between the clock pins used in synchronous timing analysis.
Dynamic Clock Skew Details	Lists the source pins and calculations that are used to create the skews between the clock pins used in synchronous timing analysis.
No Strobe Details	Contains the details of source-synchronous constraints that do not have a strobe.
Coupling Pushout	Contains the coupling effects on timing.
Coupling Noise Tab	Contains the voltage variation on victim nets caused by coupling.
Edge Details	Summarizes each edge in each simulation.
Timing Waveform Margin	Rolls up timing margins, waveform DRC violations and waveform margins for each transfer net.
Model by Designator	Contains information about nets (transfer and extended), designator, parts, IBIS model, and timing model.

Report	Description
Model Details	Contains most of the waveform DRC rules and timing levels used by the product. The actual parameter used (following the precedence rules) and the value assigned to that parameter.

Assignment Report

Assignment reports contains the assignment summary report of transfer nets.

Report	Description
Assign Netlist	Complete netlist with model data for each pin.
Swizzled Nets	List of nets whose connections appear to be incorrect. The tool looks at the logical pin names on all pins connected to a net and looks for inconsistencies that may indicate swapped bits of a bus. For example, if an extended net has a pin with logical name `DATA0` on one device and a pin with logical name `DATA7` on another device the net will be considered swizzled.

SPICE Generation Report

The SPICE generation report contains the SPICE generation log with information about the SPICE decks generated and any errors (HSPICE or IsSPICE). The report generates similar information for each of these processes:

- Pre-layout simulation (single net)
- Pre-layout simulation (all nets)
- Post-layout simulation

To view the SPICE generation report after running a pre-layout or post-layout simulation, select **Reports > SPICE Generation Report**.

Report	Description
Generate Spice Log	Number of simulation decks generated, errors in their generation and consistency checks.
Spice Decks	Data on the Spice decks generated listed by simulation name, including whether models are Spice or IBIS
Spice Deck Errors	List of reasons why a Spice deck was not generated.

See Also

More About

- “Pre-Layout Analysis of Parallel Link” on page 7-2
- “Customize Parallel Link Project for Pre-Layout Analysis” on page 7-5

8

Post-Layout Verification of Parallel Link

Post-Layout Verification of Parallel Link

In this section...

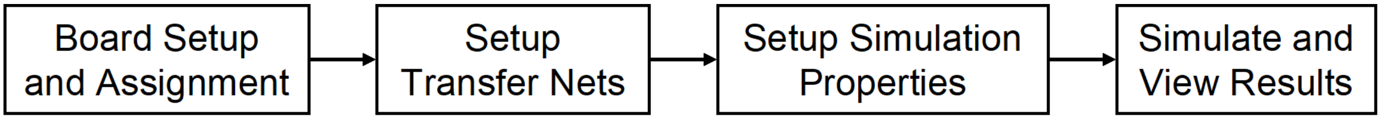
Post-layout verification provides you with an integrated signal integrity and timing environment to verify system-level SI and timing margins for your fully or partially routed PCB design databases.

The post-layout process supports single-board and multiboard analysis, along with connectivity through packages, connectors, and cabling. The post-layout verification environment provides you the ability to extract and analyze PCB databases from any combination of the following CAD (Computer Aided Design) formats:

- Cadence Allegro
- Mentor PADS Layout
- Mentor Board Station
- Mentor Expedition PCB
- Cadence APB
- Intercept Pantheon
- Altium Designer
- Altium P-CAD
- IBIS EBD

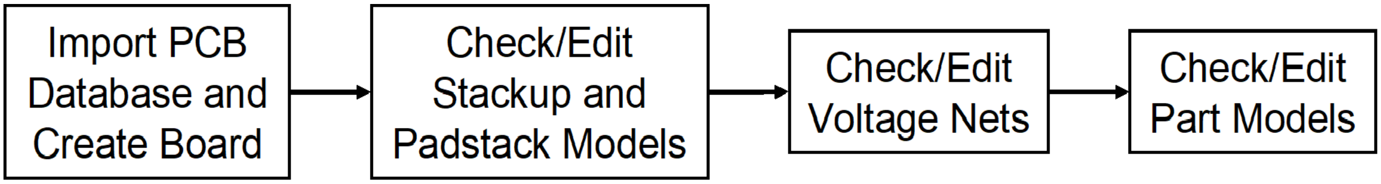
Post-layout analysis takes place in the interface of a serial link design project. If the interface you are working in has pre-layout transfer nets defined, post-layout uses them from the reference schematic set. If there are no transfer nets in the reference schematic set of the interface, the **Parallel Link Designer** app creates sheets with system transfer nets (STNETs).

The post-layout verification workflow is the same for each PCB database type. First import the PCB databases, setup the boards, connect the instances if there are multiple boards in the system, run assignment, setup and analyze the nets, set up simulation properties, then simulate and view the results.


Board Setup and Assignment
Setup Transfer Nets
Setup Simulation Properties
Simulate and View Results

Board

The first step in the post-layout verification process is board set-up and assignment. A PCB database you import to the **Parallel Link Designer** app is called a board. At the board level, check and edit all stackups, voltage nets, and models. To create variations of a PCB database using different stackups, voltages, or models, create multiple boards with unique names.


Import PCB Database and Create Board
Check/Edit Stackup and Padstack Models
Check/Edit Voltage Nets
Check/Edit Part Models

To perform the setup and assignment functions, access the Post-Layout Setup & Assignment dialog box from the **Setup** > **Setup & Assignment** menu in the app toolstrip.

For each board in the system, specify the type of the PCB database and the files in the database, view or edit the stackup, view or set voltage nets, and manage models by clicking the **Import & Setup Board** button in the Post-Layout Setup & Assignment dialog box. The Import & Setup Board dialog box has four tabs:

- **Import**

  Use the **Import Board** tab to import a PCB database and create a board. Select the PCB database type from the **PCB Database Type** selector list. By default, the **Parallel Link Designer** app creates an instance for each board and copies the PCB database files into the current project. If you do not copy the PCB database into the project, you cannot re-import the database files.
- **Stackup**

  The **Stackup** tab shows the stackup from the PCB database and allows control of padstack models. The Stackup Editor on the left side of the tab shows the stackup as it is read in from the PCB database. If necessary, you can override the auto-generated trace models using the editor. The right side of the tab has controls for the auto-generated padstack backdrill options, differential extraction, and DRC control.. For more information, see “Stackup and Extraction Control in Parallel Link Project” on page 8-6.
- **Voltages**

  The **Voltages** tab shows the CAD nets in the PCB database for the board and allows you to specify the voltage for specific voltage nets. Non-voltage nets have an NA value in the voltage column.

  Note The **Voltages** tab does not control the voltages in the IBIS or SPICE models for TX/RX designators. This tab is mainly used to correctly define the on-board terminations that require connection to a specific voltage.
- **Parts**

  Use the **Parts** tab to match models to parts in the PCB database.

Instance

An instance is an internal copy of a board that you can connect to other instances and analyze. Every board that is used in the design has at least one instance. If you use the same board more than once, you must define a separate instance for each use. For example, a system consisting of a motherboard with two DIMM slots that has the same type of DIMM plugged into each slot will have one instance of the motherboard and two instances of the DIMM.

Connection

A Connection is a pin-to-pin path from the pins of a reference designator on one instance to the pins of a reference designator on a second instance. In a multi-board system, connections between instances are specified in the Connections pane of the Post-Layout Setup & Assignment dialog box. To add a connection, click the **Add Connection** button.

Assignment

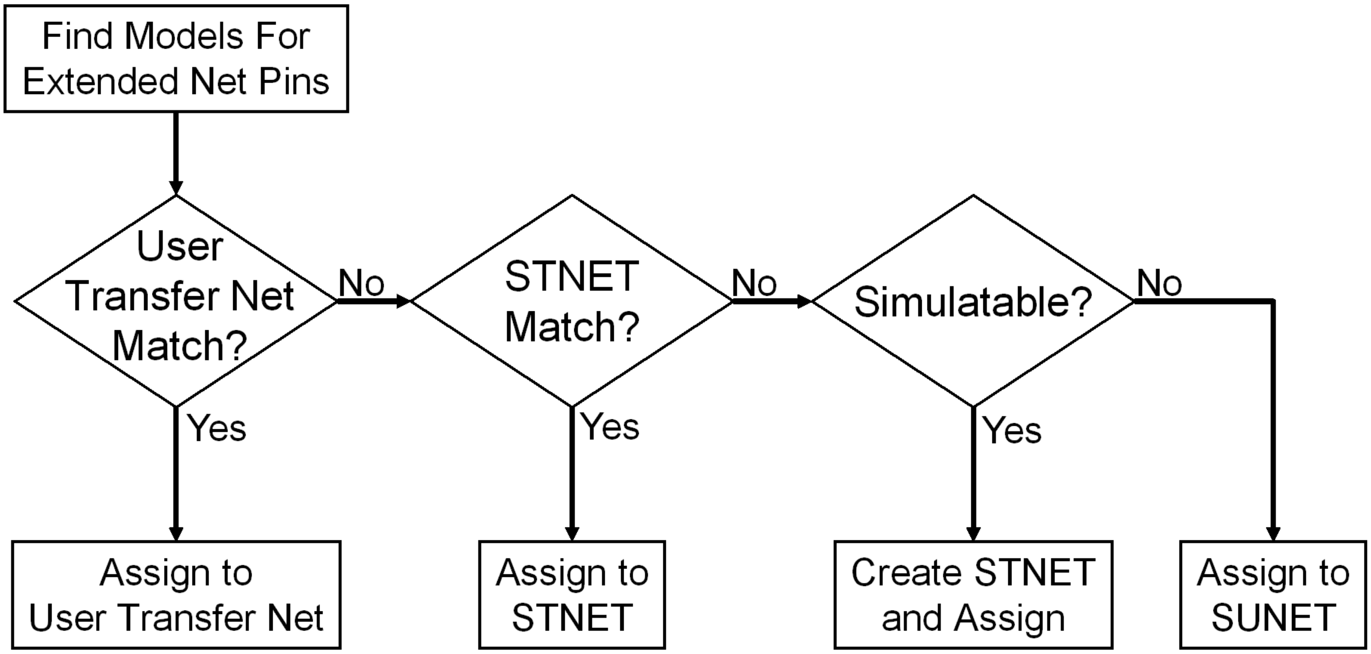
The Assignment process is an automated process for associating nets in the PCB database with transfer nets. This simplifies the setup of the essential net properties in the typical scenarios that you will face:

- **Interface without Transfer Nets**

  If you did not do a pre-layout analysis in an interface, you can create and edit transfer nets from the post-layout interface. When you set the properties of a transfer net, you set the properties of all nets assigned to that transfer net. For example, when you change the properties of a transfer net, the app automatically assigns those properties to all nets in a data bus.
- **Interface with Transfer Nets from Pre-Layout Analysis**

  If you completed pre-layout analysis in an interface, the app automatically assigns the nets you created in post-layout analysis to the transfer nets you created in the pre-layout analysis.
- **Design Kits**

  A design kit is an interface with models and preconfigured transfer nets. The app automatically assigns the nets you created in post-layout analysis to these transfer nets.



In all cases, the transfer nets and the assignment process ensure that all nets in an interface are set up and ready to simulate in a fraction of the time needed to set up each net in the interface individually.

Population

Populations allow you to setup multiple configurations of a system for simulation in one project. The app handles populations through the naming of instances.

For example, if a one-slot motherboard can accept one of three DIMMs (dual in-line memory modules), it can be set up by creating three instances of the motherboard and one instance of each DIMM. In this case, three populations can be defined: the motherboard with RCA installed, the motherboard with RCB installed and the motherboard with RCC installed

Simulation

Before you run a simulation, you must select the nets for the post-layout verification. Select the nets and add them to the list of nets to simulate. You also need to set up the stimulus patterns from simulation properties.

Topology

Extended nets that can be simulated (assigned to an STNET or user transfer nets) can have topologies created from the extracted PCB data. View these topologies from the **Pre-Layout Analysis** tab. The topologies are useful for understanding how an actual network is routed and to resolve waveform quality or timing issues identified by using post-layout verification. Once the extracted post-layout networks are in the pre-layout analysis environment, you can perform quick “what-if” analyses to identify an appropriate solution.

See Also

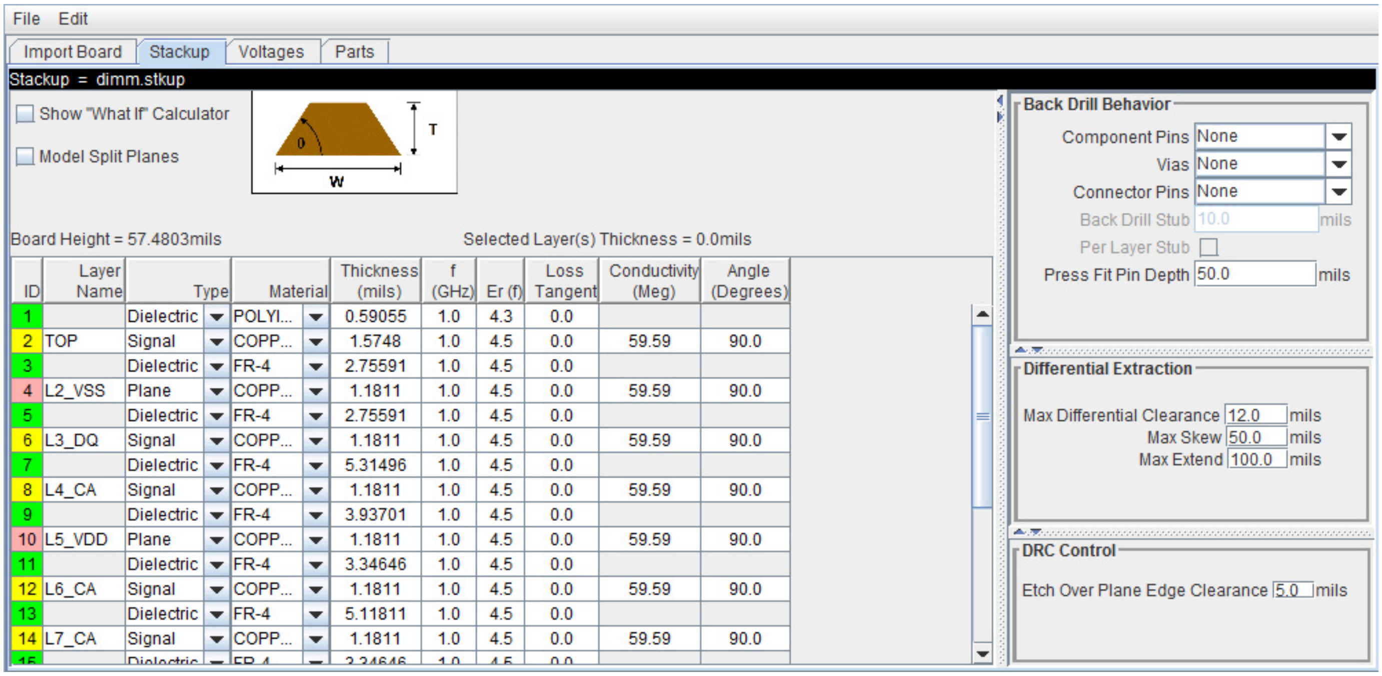
More About

- “Stackup and Extraction Control in Parallel Link Project” on page 8-6
- “Via and Stackup Management in Parallel Link Project” on page 8-9
- “Post-Layout Verification of Parallel Link” on page 8-2

Stackup and Extraction Control in Parallel Link Project

The **Stackup** tab of the **Import & Setup Board** dialog shows the stackup from the PCB Database and allows for extraction control for padstack, differential traces, and DRC.

The tab is divided into two areas: **Stackup Editor** and **Extraction Control**. The **Stackup Editor** on the left side of the tab shows the stackup that was read from the PCB Database and allows the override of the auto-generated stackup thicknesses, material properties, and trapezoidal angle as well as the ability to do “What If” exploration and select whether to model discontinuities associated with etches crossing split planes. The right side of the tab controls the padstack backdrill options, differential extraction, and DRC control.



Stackup = dimm.stkup

□ Show "What If" Calculator

□ Model Split Planes

Board Height = 57.4803mils

Selected Layer(s) Thickness = 0.0mils

ID	Layer Name	Type	Material	Thickness (mils)	f (GHz)	Er (f)	Loss Tangent	Conductivity (Meg)	Angle (Degrees)
1		Dielectric	POLYI...	0.59055	1.0	4.3	0.0		
2	TOP	Signal	COPP...	1.5748	1.0	4.5	0.0	59.59	90.0
3		Dielectric	FR-4	2.75591	1.0	4.5	0.0		
4	L2_VSS	Plane	COPP...	1.1811	1.0	4.5	0.0	59.59	90.0
5		Dielectric	FR-4	2.75591	1.0	4.5	0.0		
6	L3_DQ	Signal	COPP...	1.1811	1.0	4.5	0.0	59.59	90.0
7		Dielectric	FR-4	5.31496	1.0	4.5	0.0		
8	L4_CA	Signal	COPP...	1.1811	1.0	4.5	0.0	59.59	90.0
9		Dielectric	FR-4	3.93701	1.0	4.5	0.0		
10	L5_VDD	Plane	COPP...	1.1811	1.0	4.5	0.0	59.59	90.0
11		Dielectric	FR-4	3.34646	1.0	4.5	0.0		
12	L6_CA	Signal	COPP...	1.1811	1.0	4.5	0.0	59.59	90.0
13		Dielectric	FR-4	5.11811	1.0	4.5	0.0		
14	L7_CA	Signal	COPP...	1.1811	1.0	4.5	0.0	59.59	90.0
15		Dielectric	FR-4	3.34646	1.0	4.5	0.0		

Back Drill Behavior

Component Pins None

Vias None

Connector Pins None

Back Drill Stub 10.0 mils

Per Layer Stub □

Press Fit Pin Depth 50.0 mils

Differential Extraction

Max Differential Clearance 12.0 mils

Max Skew 50.0 mils

Max Extend 100.0 mils

DRC Control

Etch Over Plane Edge Clearance 5.0 mils

Stackup Editor

The **Stackup Editor** displays one row for each signal, plane, and dielectric layer in the stackup. Parameter values can be changed if desired by typing new values into the table cells. The stackup data plus the trace width data are used by the field solver to create lossy transmission line models for post-layout nets.

Each layer must be defined as either `Dielectric`, `Mixed`, `Plane`, or `Signal` in the stackup column called **Type**. Signal layers can be either type `Mixed` or `Signal`. The `Mixed` designation is provided primarily for boards and packages where sections of the signal layer may contain small planes for impedance control. In most cases the `Signal` designation would be sufficient, but it is important to carefully review the board layout and identify cases where `Mixed` may be required.

Checking **Model Split Planes** enables modeling of discontinuities associated with etches crossing over splits in planes. The change in trace cross-section results in an impedance change in the model.

You may use the **Stackup Editor** as a calculator to compute trace impedance based upon the width and separation. To use the calculator:

1. Check the **Show “What If” Calculator** check box to display the calculator columns
2. Enter one or more values in the appropriate cells followed by the tab key
3. Click **Calculate**

This uses the stackup data with the **Desired Width** and **Desired Separation** values to calculate the single-ended and differential impedance for that layer.

Extraction Control

The **Extraction Control** section of the tab controls the backdrill behavior, differential extraction, and DRC control.

Backdrilling uses Must Not Cut Layers. Must Not Cut Layers are layers that define a valid backdrill depth. In the stackup there are columns for Must Not Cut Layers from the top and bottom. The backdrill goes from the top or bottom up to but not through the last Must Not Cut Layer that is encountered before a trace connection to a via or pin. If no Must Not Cut layer is encountered before the trace connection to the via or pin, then the via or pin is modeled as not backdrilled.

Backdrill Behavior Choice	Description
`None`	No backdrilling. The complete via or pin is extracted, and a model generated based on the PCB data for start and end layers.
`Top`	The via or pin is modeled as if it were drilled from the top of the board. The via or pin ends at the lowest layer with **Backdrill Top Must Not Cut Layer** checked in the stackup that is above the highest layer with a trace connected to the via or pin. A stub equal to the **Back Drill Stub** parameter is left. If there is no layer with **Backdrill Top Must Not Cut Layer** checked that is above the highest trace connection to the via or pin, the via or pin is not backdrilled.
`Bottom`	The via or pin is modeled as if it were drilled from the bottom of the board. The via or pin ends at the highest layer with **Backdrill Bottom Must Not Cut Layer** checked in the stackup that is below the lowest layer with a trace connected to the via or pin. A stub equal to the **Back Drill Stub** parameter is left. If there is no layer with **Backdrill Bottom Must Not Cut Layer** checked that is below the lowest trace connection to the via or pin, the via or pin is not backdrilled.
`Both`	Both top and bottom are modeled as described above.
`Longest Stub`	Drills from the side that remove the longest stub based on the Must Not Cut layers defined in the stackup.

In the **Differential Extraction section** of the **Padstack Editor**, you can define the parameters that control the extraction of the differential nets.

Parameter	Description
Max Differential Clearance	The maximum edge-to-edge clearance two traces can have and still be extracted as a differential transmission line model. If the clearance is larger than this parameter, the traces are extracted as two single-ended transmission line models.
Max Skew	The maximum length difference between the two traces in a single differential trace w-line model. It is recommended that this be set no larger than 1/10 of the wavelength of the maximum frequency of interest.
Max Extend	The maximum total length of single-ended trace that can be combined with a differential trace in a w-line model.

The DRC control defines the minimum distance from a trace to a plane edge when the trace crossing DRC is run using the **Etch Over Plane Edge Clearance** parameter.

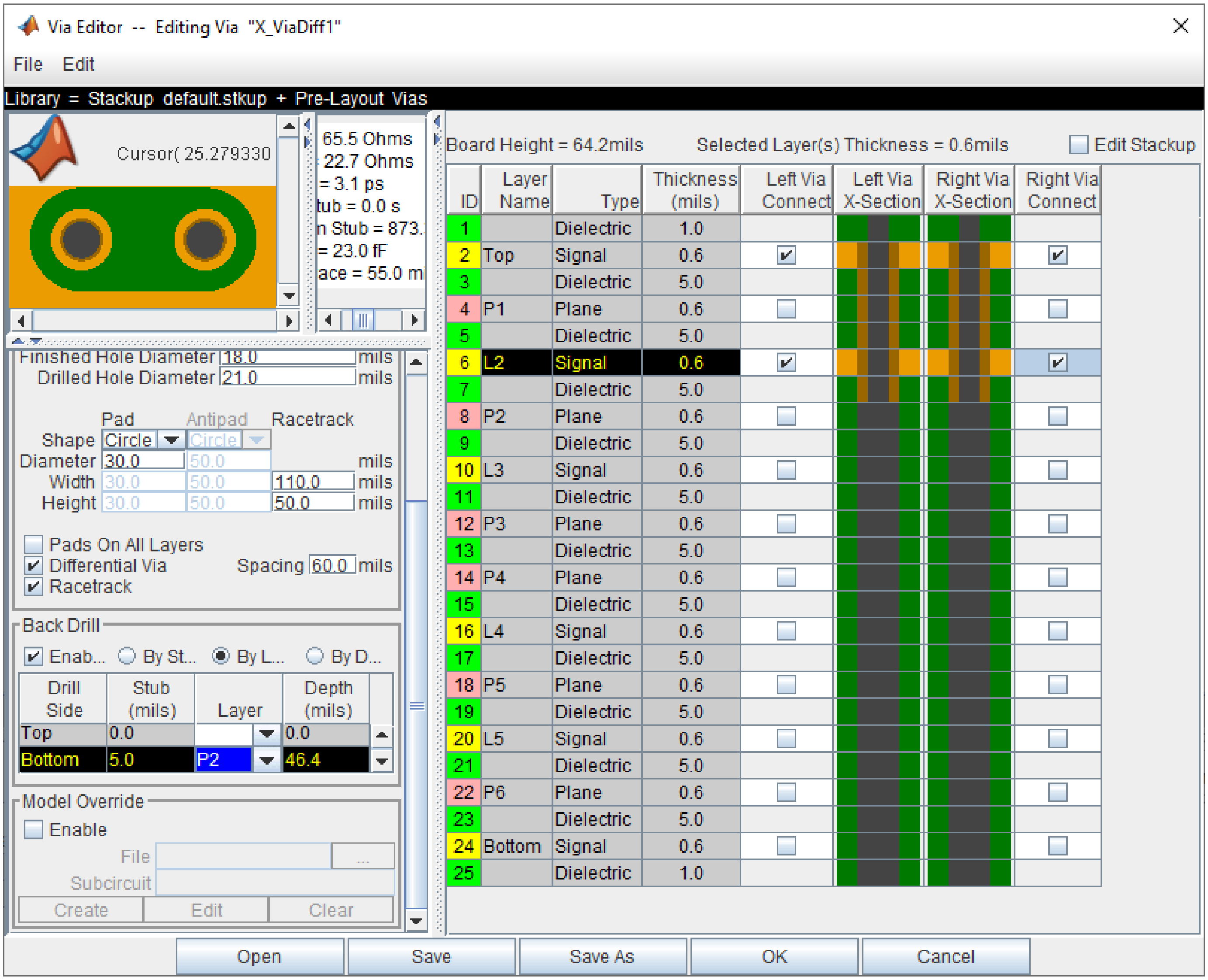
See Also

More About

- “Post-Layout Verification of Parallel Link” on page 8-2
- “Via and Stackup Management in Parallel Link Project” on page 8-9

Via and Stackup Management in Parallel Link Project

The vias are associated with a stackup in the library where they are stored. There can be multiple stackup and via libraries in a project. The first time you edit a via in pre-layout you are prompted for the number of layers to use for the default pre-layout stackup. In post-layout the stackup and vias are from the PCB database by default. Use the **Via Editor** dialog box by right clicking on the vias in the Pre-Layout Analysis tab to manage them. The elements in the via editor can be divided into three groups: common via elements, pre-layout specific via elements, and post-layout specific via elements.



Library = Stackup default.stkup + Pre-Layout Vias

Cursor(25.279330

65.5 Ohms
22.7 Ohms
= 3.1 ps
tub = 0.0 s
n Stub = 873.
= 23.0 fF
ace = 55.0 m

Finished Hole Diameter 18.0 mils
Drilled Hole Diameter 21.0 mils

	Pad	Antipad	Racetrack	
Shape	Circle	Circle		
Diameter	30.0	50.0		mils
Width	30.0	50.0	110.0	mils
Height	30.0	50.0	50.0	mils

□ Pads On All Layers
☑ Differential Via Spacing 60.0 mils
☑ Racetrack

Back Drill

☑ Enab... ○ By St... ● By L... ○ By D...

Drill Side	Stub (mils)	Layer	Depth (mils)
Top	0.0		0.0
Bottom	5.0	P2	46.4

Model Override

□ Enable

File

Subcircuit

Board Height = 64.2mils Selected Layer(s) Thickness = 0.6mils □ Edit Stackup

ID	Layer Name	Type	Thickness (mils)	Left Via Connect	Left Via X-Section	Right Via X-Section	Right Via Connect
1		Dielectric	1.0				
2	Top	Signal	0.6	☑			☑
3		Dielectric	5.0				
4	P1	Plane	0.6	□			□
5		Dielectric	5.0				
6	L2	Signal	0.6	☑			☑
7		Dielectric	5.0				
8	P2	Plane	0.6	□			□
9		Dielectric	5.0				
10	L3	Signal	0.6	□			□
11		Dielectric	5.0				
12	P3	Plane	0.6	□			□
13		Dielectric	5.0				
14	P4	Plane	0.6	□			□
15		Dielectric	5.0				
16	L4	Signal	0.6	□			□
17		Dielectric	5.0				
18	P5	Plane	0.6	□			□
19		Dielectric	5.0				
20	L5	Signal	0.6	□			□
21		Dielectric	5.0				
22	P6	Plane	0.6	□			□
23		Dielectric	5.0				
24	Bottom	Signal	0.6	□			□
25		Dielectric	1.0				

Via Elements

Via Element	Description
Top view and electrical characteristics	The top view shows the via as it would appear when viewed from the top of the board. The electrical characteristics show the impedance, delay backdrill, and other characteristics. The reported delay is for the barrel of the via.
Via geometry	You can edit the geometry of the via by defining the start and end layers, hole diameters, and shape and dimensions of the pad and antipad. You can also select if a via model is single-ended or differential-ended.
Via backdrill	You can select the depth of via backdrill by stub, layer, or depth.
Override via model	You can override a via model by using your custom subcircuit saved in one of the SPICE libraries.
Connect via layers	The **Left Via Connect** column is used to select the layer connections that will appear on the left side of the via symbol. The **Right Via Connect** column is used to select the layer connections that will appear on the right side of the via symbol. A layer is connected when the checkbox for that layer is checked. The **Via X-Section** columns show a representation of the via cross section.
Modify stackup	To modify the stackup, check the Edit Stackup checkbox.

There are several important definitions for vias and pins:

- A via under a BGA is a via, not a pin.
- A through hole connector padstack is a pin not a via.
- A connector means a multi-board connector (connects two Instances).

Editing Via for Pre-Layout Simulations

To edit vias for pre-layout simulation, open the Via Editor dialog box by selecting **Tools > Via Editor** or by right-clicking on a via schematic symbol and selecting **Edit Differential Via Model** or **Edit Single Ended Via Model**. You need to enter the number of conducting layers for the default stackup the first time you open the **Via Editor** dialog box.

The Via Editor works in a selected library. Vias can be edited, added or deleted from a library. In pre-layout, the Via Editor creates a default library that contains a default via model and a default stackup. The Library operations can be selected from the File menu.

Editing Via for Post-Layout Simulations

The Padstack/Trace Manager is used to view and manage overrides to padstacks and traces in Post-Layout as well as manage backdrilling of pins and vias by net, RefDes or Part. You can edit the geometry of a single via, or multiple vias at one time.

- **Back Drill Setup Tab**

  The **Back Drill Setup** tab allows backdrill information to be viewed and changed by net, by padstack, by RefDes, or by Part by selecting from the **View Mode** list. In each case the backdrill can be turned on or off. The view modes are:

  - *Back Drill by Net* — One row per Extended Net per Board.
  - *Back Drill by Padstack* — One row per Padstack.
  - *Back Drill by RefDes* — One row per Reference Designator.
  - *Back Drill by Part* — One row per Part Number.

  The **Back Drill Setup** tab is only enabled if backdrilling is enabled on one or more boards on the **Stackup** tab of the **Setup Board** dialog.

- **Via/Pin Editor Tab**

  Padstack models are created automatically from the PCB data for vias, surface mount pads and through-hole pins using the internal padstack solver.

Padstack Definitions

Padstack Elements	Definitions
Padstack	The geometry information from the PCB database. Contains the start and end layer of the padstack, barrel dimensions, etc. A Padstack does not contain the layers connected or XY coordinates.
Padstack Configuration	A Padstack plus layer connections. A Padstack Configuration does not contain XY coordinates.
Padstack Configuration Instance	A Padstack Configuration at a specific XY coordinate on a board. A specific via has geometry, connectivity and a location on a PCB. A specific pin has geometry, connectivity, a location, a reference designator and a pin number

A Padstack can be used for multiple Padstack Configurations. A Padstack Configuration can be used for multiple Padstack Configuration Instances.

Padstack Editor View Modes

The views are selected from the list on the Via/Pin tab. The view modes correspond to the definitions above. In each view mode there is one row for each item of the selected type:

- *Padstack (Geometry)* — One row per Padstack. This rolls up all Padstack Configurations and Padstack Configuration Instances that use a Padstack.

- *Padstack Configuration (Connectivity)* — One row per Padstack Configuration. This rolls up all Padstack Configuration Instances that use a Padstack Configuration.
- *Padstack Configuration Instance* — One row per Padstack Configuration Instances.

Padstack Editor Edit Modes

The padstack editor has two modes:

- *Padstack* — All changes made in the editable columns apply to the padstack. This means all Padstack Configuration Instances that use the same Padstack as the row being edited will change. For example, if the View Mode is Padstack Configuration Instance and the Edit Mode is Padstack, a change to one row is applied to all rows that have the same Padstack.
- *Instance* — All changes apply to the Padstack Configuration Instance only. For example, if the View Mode is Padstack Configuration Instance and the Edit Mode is Instance, a change to one row is only applied to that row.

Common Operations

Editing geometry of a single via	To edit the geometry of a single via (one via at one XY coordinate), use the Padstack Configuration Instance View Mode and the Instance Edit Mode. Any changes to the geometry is applied to the specific via edited when in this mode.
Editing using the Via Editor	Right-click on a row and choose Visual Via Editor from the menu.
Changing the Padstack	To change the Padstack that a Padstack Configuration Instance is based on, use the Padstack Configuration Instance View Mode and in the Base Padstack column choose a different Padstack. The list of padstacks are the padstacks that share the same start and end layer with the original padstack for this Padstack Configuration Instance.
Editing a Padstack	To edit a Padstack, use the View Mode Padstack. The behavior depends on Edit Mode: • *Padstack* — Changes are made to the Padstack being edited and is applied to all Padstack Configurations and Padstack Configuration Instances that use the Padstack. • *Instance* — A new Padstack is created as a copy of the Padstack being edited, and the changes you make is applied to this new Padstack

Overriding a via model	Via models are typically done with connectivity to specific layers. Therefore, the Padstack Configuration View Mode or the Padstack Configuration Instance View Mode are used to override a via model. In both modes the Model Override column is part of the table. To override a model right-click and select one of: • *Browse* — Browse to an existing model in the libraries. This could be the .smod file for an S-Parameter via model that was imported. • *Create* — Create a subcircuit with the default via model. This subcircuit can be modified. If the View Mode is Padstack Configuration the model is applied to every Padstack Configuration Instance that uses that Padstack Configuration. If the View Mode is Padstack Configuration Instance the model is applied to the single Padstack Configuration Instance that you edited. **Note** The edit mode must be Padstack.

Trace Overrides Tab

The Trace Overrides tab of the Padstack/Trace Manager is used for trace model overrides. The lossy transmission line models for traces created by the field solver from the stackup and trace width can be overridden with user-provided models. The Trace Overrides tab shows the trace widths on each layer of each board.

The models used for overrides are assumed to be RLGC models with one model per file, and the base name of the file must be the same as the model name.

For single-ended traces there is one row for each trace width found on each layer. Select one or more rows and click the Select Model button to browse to a transmission line model in the library.

For differential traces, there is one row for each trace width on each layer, and columns for differential separation and coupling layer. The coupling layer is a list containing the same layer and any adjacent signal or mixed layers. Select an adjacent layer for broadside coupled differential traces. When a separation is added a new row is created for that layer and trace width. This allows models for multiple separations to be specified for each width on each layer.

The tolerance for overrides is 0.1 mm in width. In other words, if an override is specified for a trace of width 4.0 mm on a layer, the override is applied to all traces with widths from 3.9 mm to 4.1 mm on that layer.

Example One-Conductor Model

For file name `sl_55ohm.mod`:

```
model sl_55ohm W ModelType=RLGC N=1
+ Lo = +3.60600E-07
+ Co = +1.20300E-10
+ Ro = +6.07368E+00
+ Rs = +1.48880E-03
+ Gd = +1.89000E-11
```

Example Differential Model

For file name `sl_55ohm_diff.mod`:

```
.model sl_55ohm_diff W ModelType=RLGC N=2
+ Lo = +3.58800E-07 +4.84700E-08 +3.58800E-07
+ Co = +1.23200E-10 -1.66400E-11 +1.23200E-10
+ Ro = +6.07368E+00 +0.00000E+00 +6.07368E+00
+ Rs = +1.50556E-03 +1.12767E-04 +1.50556E-03
+ Gd = +1.93500E-11 -2.61400E-12 +1.93500E-11
```

See Also

More About

- “Post-Layout Verification of Parallel Link” on page 8-2
- “Stackup and Extraction Control in Parallel Link Project” on page 8-6

9

Parallel Link Featured Examples

Configure DDR Controller with Two Memory Designators

This example shows how you can configure a DDR controller with two custom memory designators.

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 `ddr2_controller`, the interface as `ddr2`, and the schematic sheet as `dq`. The **Pre-Layout Analysis** tab shows the blank schematic sheet.

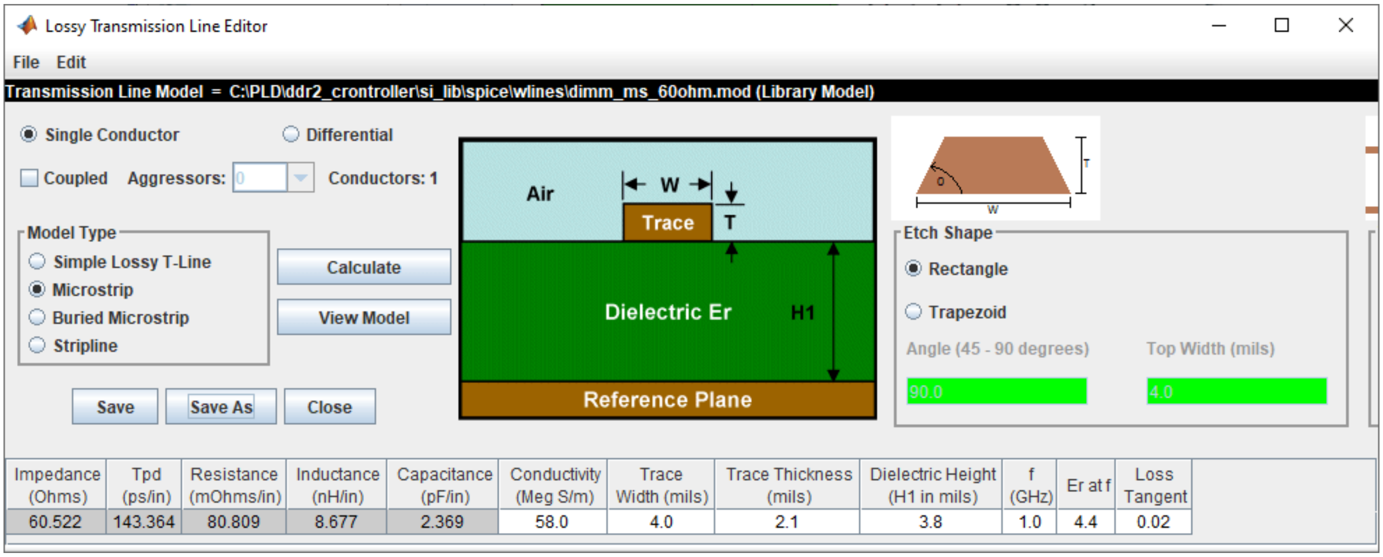
Set Up Libraries

You can create the library elements for the transmission lines, packages, connectors, and designators. In this case, you want to model a DIMM (dual in-line memory module) that has a stackup that gives a 60 ohm impedance for traces on the top and bottom layers (where the dq nets are routed). So, you need to create a 60 ohm transmission line model to be used for the transmission line segments of the DIMM.

Create T Line Model

Create a differential lossy transmission line model based on a stripline cross-section. Select **Tools** > **Lossy Transmission Line Editor**. In the newly opened Lossy Transmission Line Editor dialog box, select **Single Conductor** and select Model Type as **Microstrip**. The **Microstrip** model type routes data lines on the top and bottom of the DIMM. The traces are 4 mils wide and 2.1 mils thick. They are 3.8 mils above a dielectric of Er 4.4. So change the parameters **Trace Thickness (mils)** to `2.1`, **Dielectric Height (H1 in mils)** to `3.8`, and **Er at f** to `4.4`.

Click the **Calculate** button to run the 2-D field solver. The Impedance at the bottom left changes from derived to the calculated value.



Transmission Line Model = C:\PLD\ddr2_crontroller\si_lib\spice\wlines\dimm_ms_60ohm.mod (Library Model)

● Single Conductor ○ Differential

□ Coupled Aggressors: 0 Conductors: 1

Model Type
○ Simple Lossy T-Line
● Microstrip
○ Buried Microstrip
○ Stripline

Etch Shape
● Rectangle
○ Trapezoid
Angle (45 - 90 degrees) 90.0
Top Width (mils) 4.0

Impedance (Ohms)	Tpd (ps/in)	Resistance (mOhms/in)	Inductance (nH/in)	Capacitance (pF/in)	Conductivity (Meg S/m)	Trace Width (mils)	Trace Thickness (mils)	Dielectric Height (H1 in mils)	f (GHz)	Er at f	Loss Tangent
60.522	143.364	80.809	8.677	2.369	58.0	4.0	2.1	3.8	1.0	4.4	0.02

Click the **Save As** button to save the model in the project's library. Use the name `dimm_ms_60ohm`. Make sure the directory is `<Project Library>/spice/wlines`. Close the Lossy Transmission Line Editor.

Add Connector Model

Download the model file `dimm_connector.mod` attached as a supporting file to the example and copy it to the project library `<project_name>/si_lib/spice/connectors`.

Create Clock Domains

The bit times for the nets in the project come from the clock domain file. Edit the clock domain file by selecting **Setup** > **Clock Domain** and add the following lines:

```
ddr2_ck_period = 5.0

ddr2_ck_ui = ddr2_ck_period/2

ddr2_ctrl_ui = ddr2_ck_period

ddr2_addcmd_ui = 2 * ddr2_ck_period

ddr2_dq_ui = ddr2_ck_period/2

ddr2_dqs_ui = ddr2_ck_period/2
```

This sets the base clock period (ddr2_ck_period) to 5 ns. The data rate for all the other netclasses are set based on this rate. Changing the base clock period changes the data rate for the entire interface.

Save and close the clock domain file.

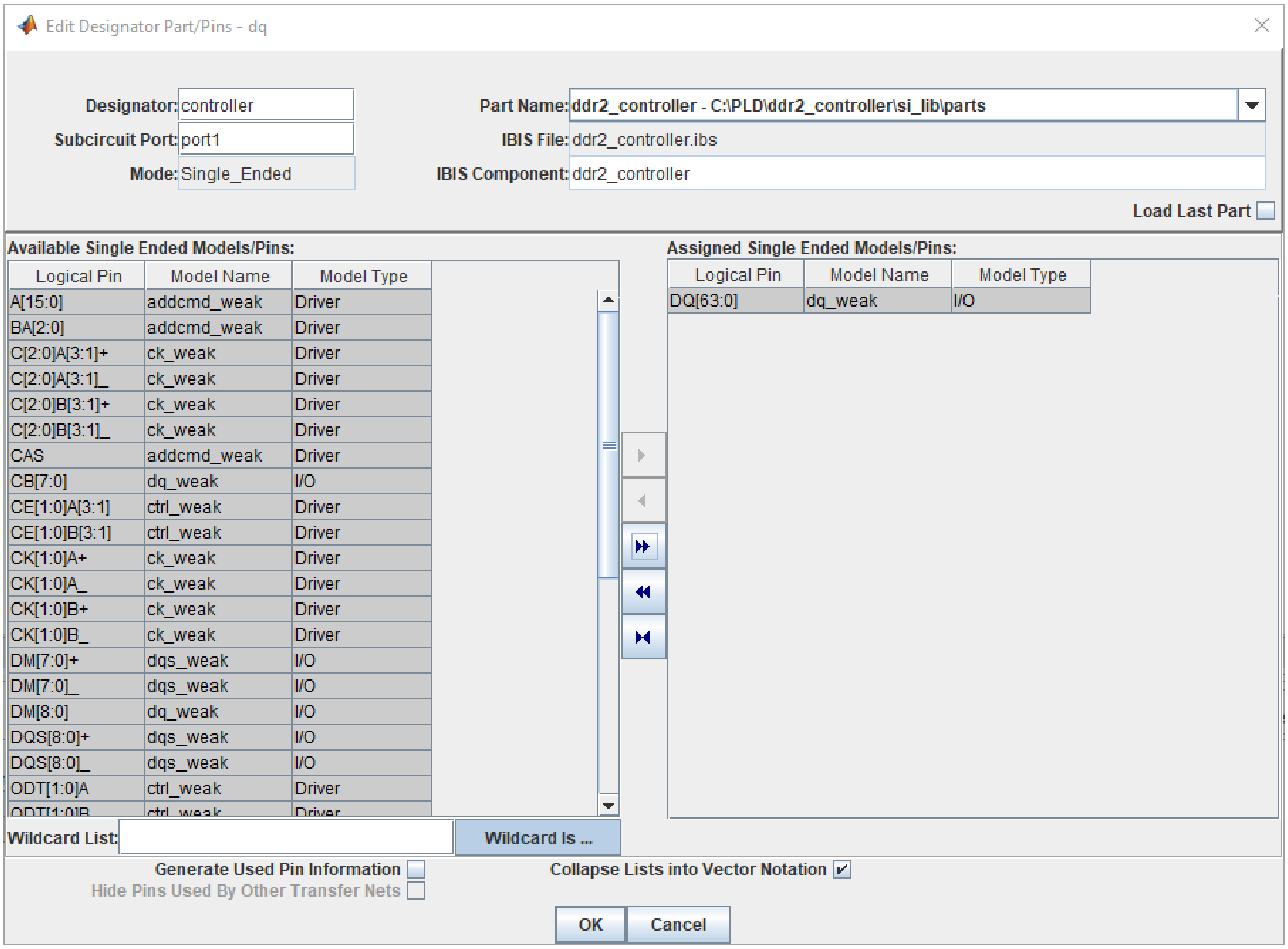
Import IBIS Models

Download and extract the `IBIS_files.zip` attached to this example. Select **Libraries** > **Import IBIS** and browse to the location of the downloaded files to import both ddr2 controller and sdram files.

Note that both IBIS files use the same `dq_dm_sstl_18.inc` file to define the 1.8V SSTL voltage levels, so you may get a Warning dialog that this file already exists. Since it is the same file, you can select either Yes or No.

Create Schematic

Add three single-ended designators, one controller designator on the left and two memory designators on the right. Select the controller designator on the left, right click and select **Edit Designator Parts and Pins**. Set the **Designator** parameter to `controller` and **Part Name** parameter to `ddr2_controller` from the dropdown menu. Since this schematic sheet represents the DQ nets, you need to add all data pins of the controller to the designator. To make it easier to select them, select **Collapse Lists Into Vector Notation**. Select `DQ[63:0]` and click the right arrow button to assign it to the designator.



Edit Designator Part/Pins - dq

Designator: controller
Subcircuit Port: port1
Mode: Single_Ended

Part Name: ddr2_controller - C:\PLD\ddr2_controller\si_lib\parts
IBIS File: ddr2_controller.ibs
IBIS Component: ddr2_controller

Load Last Part □

Available Single Ended Models/Pins:

Logical Pin	Model Name	Model Type
A[15:0]	addcmd_weak	Driver
BA[2:0]	addcmd_weak	Driver
C[2:0]A[3:1]+	ck_weak	Driver
C[2:0]A[3:1]_	ck_weak	Driver
C[2:0]B[3:1]+	ck_weak	Driver
C[2:0]B[3:1]_	ck_weak	Driver
CAS	addcmd_weak	Driver
CB[7:0]	dq_weak	I/O
CE[1:0]A[3:1]	ctrl_weak	Driver
CE[1:0]B[3:1]	ctrl_weak	Driver
CK[1:0]A+	ck_weak	Driver
CK[1:0]A_	ck_weak	Driver
CK[1:0]B+	ck_weak	Driver
CK[1:0]B_	ck_weak	Driver
DM[7:0]+	dqs_weak	I/O
DM[7:0]_	dqs_weak	I/O
DM[8:0]	dq_weak	I/O
DQS[8:0]+	dqs_weak	I/O
DQS[8:0]_	dqs_weak	I/O
ODT[1:0]A	ctrl_weak	Driver
ODT[1:0]B	ctrl_weak	Driver

Assigned Single Ended Models/Pins:

Logical Pin	Model Name	Model Type
DQ[63:0]	dq_weak	I/O

Wildcard List: Wildcard Is ...

Generate Used Pin Information □
Hide Pins Used By Other Transfer Nets □
Collapse Lists into Vector Notation ☑

OK Cancel

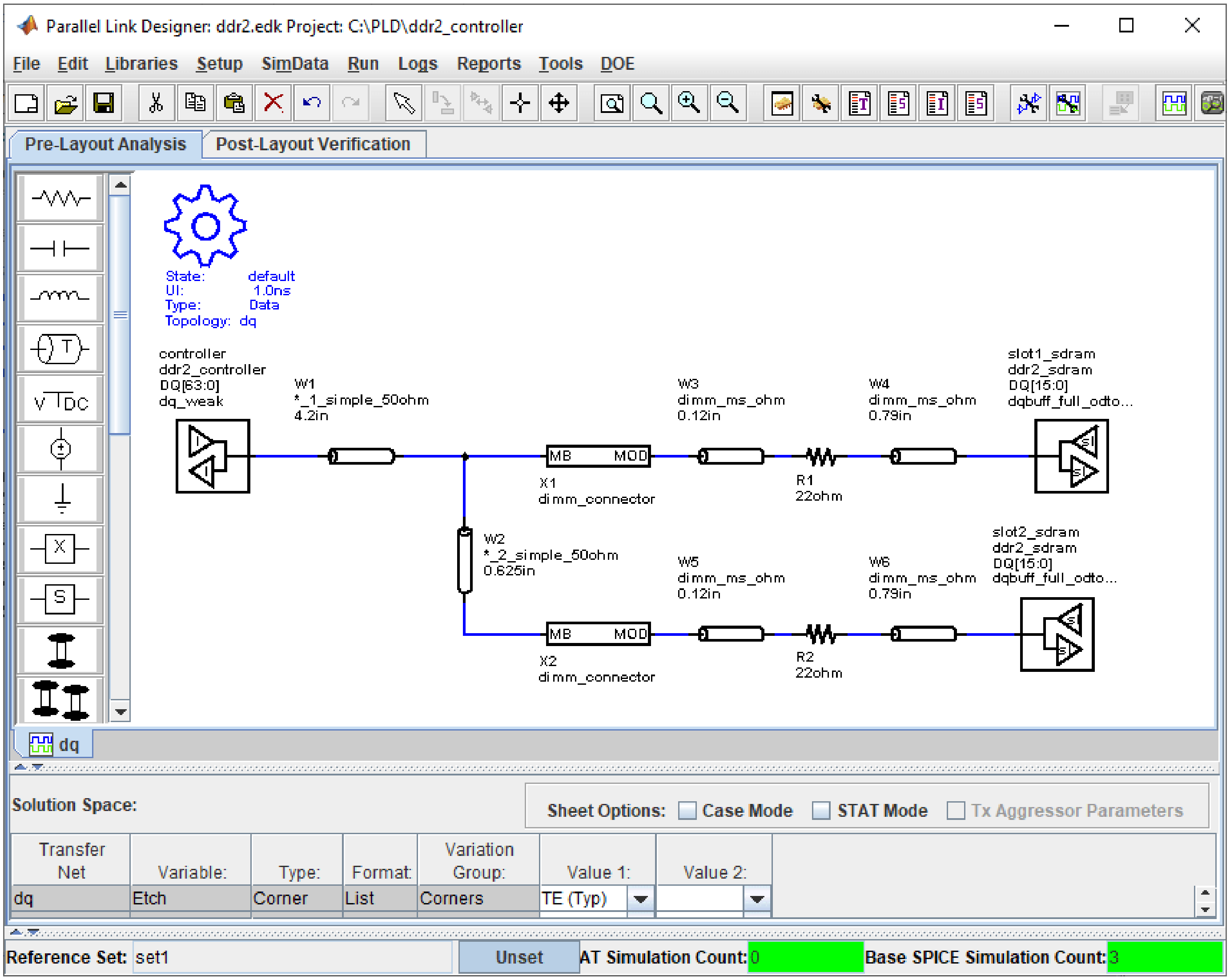
Click **OK** to close the Edit Designator Pat/Pins dialog box.

Edit the two memory designators in the same manner as the controller. Name the top one `slot1_dram` and the bottom one `slot2_dram`. For both designators use the `ddr2_sdram` part and include pins `DQ[15:0]`.

Use two lossy transmission line elements to model the etch from the controller to slot one, and the etch from slot one to slot two. Double click on the transmission line symbols to change their lengths. Change the controller to slot one t-line length to 4.2 in and the slot one to slot two t-line length to 0.625 inches.

To add the DIMM connector models, use the subcircuit element (X). In the newly opened dialog box, set the directory to `<Project Library>\spice\connectors` and select the `dimm_connector.mod` file. Place two connector subcircuits on the schematic, one for each DIMM.

There is a series resistor on each DIMM with a transmission line segment on each side. Add the resistor and a transmission line element. Double click and change the resistor value to 22 ohm. Right click on the transmission line element for one side of the DIMM, select **Select T-Line Mode**l from the menu, go to directory `<Project Library>\spice\wlines` and select `dimm_ms_60ohm` transmission line model that you created. Copy the resistor one more time for slot two, and the transmission line element three more times to have them both sides of the resistors. On the left side, set the transmission line segment lengths to `0.12` inch. On the right side, set the transmission line segment length to `0.79` inch. Connect the elements by double clicking on each connection to add wires to complete the schematic.



Solution Space:

Sheet Options: □ Case Mode □ STAT Mode □ Tx Aggressor Parameters

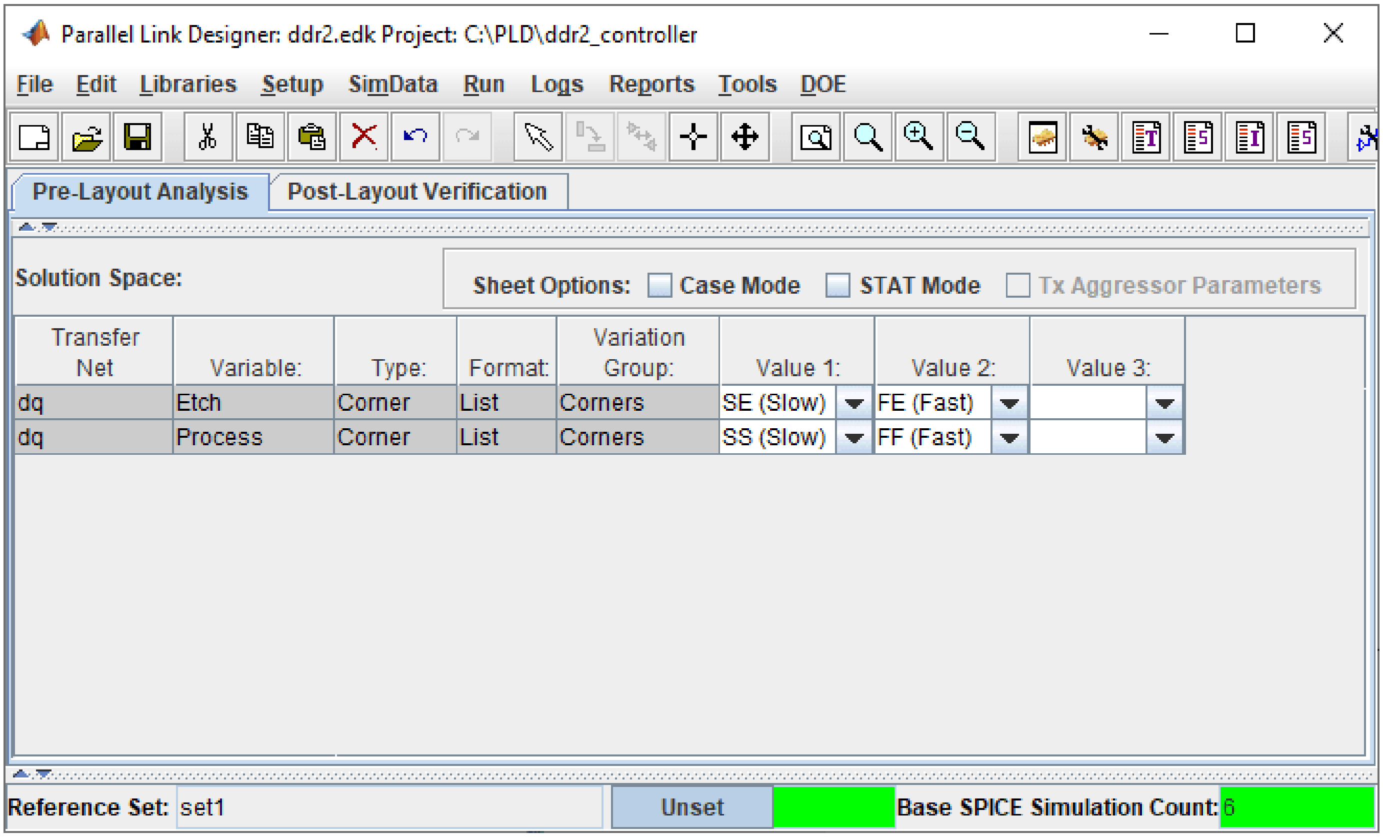
Transfer Net	Variable:	Type:	Format:	Variation Group:	Value 1:	Value 2:
dq	Etch	Corner	List	Corners	TE (Typ)	

Reference Set: set1 | Unset | AT Simulation Count: 0 | Base SPICE Simulation Count: 3

Setup Simulation and Validate Schematic

Double-click on the gear symbol (⚙) to launch the Sheet Simulation Control dialog box. Set the **UI** to `2.5` ns by selecting `2.5ns - ddr2_dq_ui` from the dropdown menu.

In the solution space, select both slow and fast corners for etch and process corners.


Parallel Link Designer: ddr2.edk Project: C:\PLD\ddr2_controller
File Edit Libraries Setup SimData Run Logs Reports Tools DOE
Pre-Layout Analysis
Post-Layout Verification
Unset

Solution Space:

Sheet Options: □ Case Mode □ STAT Mode □ Tx Aggressor Parameters

Transfer Net	Variable:	Type:	Format:	Variation Group:	Value 1:	Value 2:	Value 3:
dq	Etch	Corner	List	Corners	SE (Slow)	FE (Fast)	
dq	Process	Corner	List	Corners	SS (Slow)	FF (Fast)	

Reference Set: set1

Base SPICE Simulation Count: 6

Validate the schematic by selecting **Run** > **Validate Current Schematic Set**. The validation log should report no error and one warning. The warning says that the three transfer net designators have no timing data. This is telling you that there are no timing models for the controller or dram.

See Also

More About

- “Post-layout of DDRx Interface with CPU and DIMMs” on page 9-7

Post-layout of DDRx Interface with CPU and DIMMs

This example demonstrates the use of **Parallel Link Designer** in the Signal Integrity Toolbox™ in MATLAB® to set up a post-layout analysis of a DDRx interface on a Main Board having a CPU connected to two DIMM slots to verify that waveform quality and timing margins are met by the PCB database. Many of the steps illustrated in this example are also applicable to **Serial Link Designer** for post-layout analysis of SERDES links.

Note: Post-layout extraction and anaylsis requires RF PCB Toolbox™ in addition to Signal Integrity Toolbox. The focus of this example is to illustrate how to setup a post-layout project in **Parallel Link Designer** or **Serial Link Designer**.

Note: While there are DDR5 IBIS models in the support project "DDRx_CPU_Dimm_Postlayout," the circuit topologies do not represent a real-world DDR5 system and are meant only to represent an abstract DDRx interface. If your focus is specifically DDR5, please see the DDR5 specific examples.

Overview

This tutorial shows how Parallel Link Designer can be used to analyze a DDRx memory interface in pre-layout and post-layout using an implementation kit as the starting point. This example assumes you are beginning with the implementation kit "DDRx_CPU_Dimm_Postlayout" from the support package site. This kit has Parts pre-configured for ICs such as CPU and SDRAM. It also has pre-layout schematic sheets matching the configuration of the interface to be analyzed on the PCB database. This will allow you to begin Signal Integrity analysis immediately since all models and schematic sheets are part of the kit. If an implementation kit were not available for this interface, you would need to build the simulation environment (Parts, IBIS Models, and pre-layout topologies) from a new project. An implementation kit is simply a way to reuse a project once all of the above tasks have been done. So for example, a copy of your post-layout kit may be used for another hardware design if it uses the same CPU, DIMM and ASIC components.

Post-Layout Verification

Post-layout verification is used to verify that the voltage and timing margins are met on the routed board. In this section you will import and set up the post-layout system, simulate and analyze the nets. Example board databases for a Main Board(file "mb.zip") and DIMM(file "dimm.zip") are provided as attachments to this example in **Parallel Link Designer** "Neutral" format. You can acquire these by clicking the button to download the attachments and place into a folder called "boards." You can also follow the steps in this example as a guide to create a post-layout project with your own PCB databases. The following are the key points to create and configure a Post-layout project.

Import and Setup Boards

- Import Main Board and DIMM
- Configure PCB stackup
- Set voltages for IO and any active-termination nets
- Configure or create Parts (which contain IBIS and IBIS-AMI models)

Create and Connect Instances

- Create Instances of the Main Board and two DIMMs
- Configure connectivity between Instances of the Main Board and DIMMs

- Setup connector models

Setup and Run Assignment

- Select CAD nets to include/exclude for Assignment to Transfer Nets
- Run Setup and Assignment
- Configure any Model Overrides for padstacks (vias) or Traces in project database
- View Transfer Nets of interest in the boards with Signal Integrity Viewer

Configure Transfer Net Properties

- Configure source-destination bus transactions in Transfers dialog
- Set IBIS models for IO drive strength (ODS) and on-die termination (ODT)

Run Post-Layout Simulation

- Configure PostLayout SI/Timing Simulation dialog
- Run project simulation

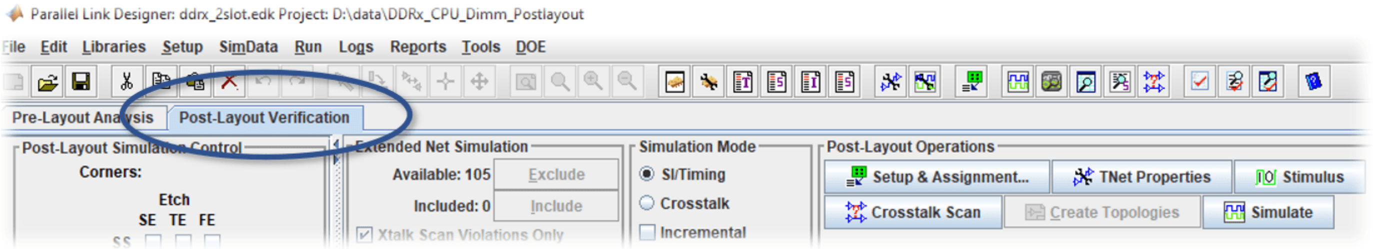
Open DDRx CPU Dimm Postlayout Kit

Open the kit "`DDRx_CPU_Dimm_Postlayout`" in the **Parallel Link Designer** app using the `openSignalIntegrityKit` function:

```
openSignalIntegrityKit("DDRx_CPU_Dimm_Postlayout");
```

Set up Post-Layout Verification

Post-Layout Verification is set up and performed from the Post-Layout Verification tab:

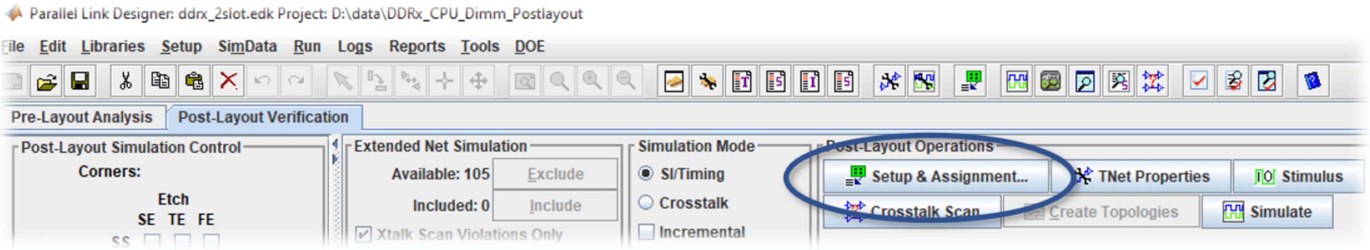


PCB databases are imported into Parallel Link Designer. Then you can configure the PCB stackup information, voltage nets and models assigned and are then placed into a board library which can be used across multiple interfaces. Boards from the library are instantiated and connected to enable Signal Integrigy analysis of a complete end-to-end system. For this tutorial there is a Main Board and a DIMM. You will import and set up these two PCB databases, then create **Instances** for use in simulation analysis, and connect them together by using the **Add Connections** dialog.

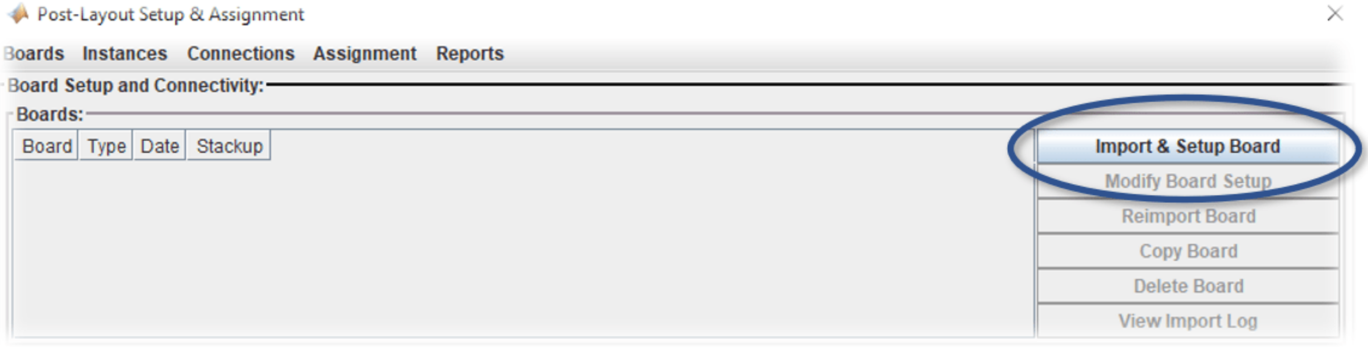
Import and Setup Boards

Import Main Board

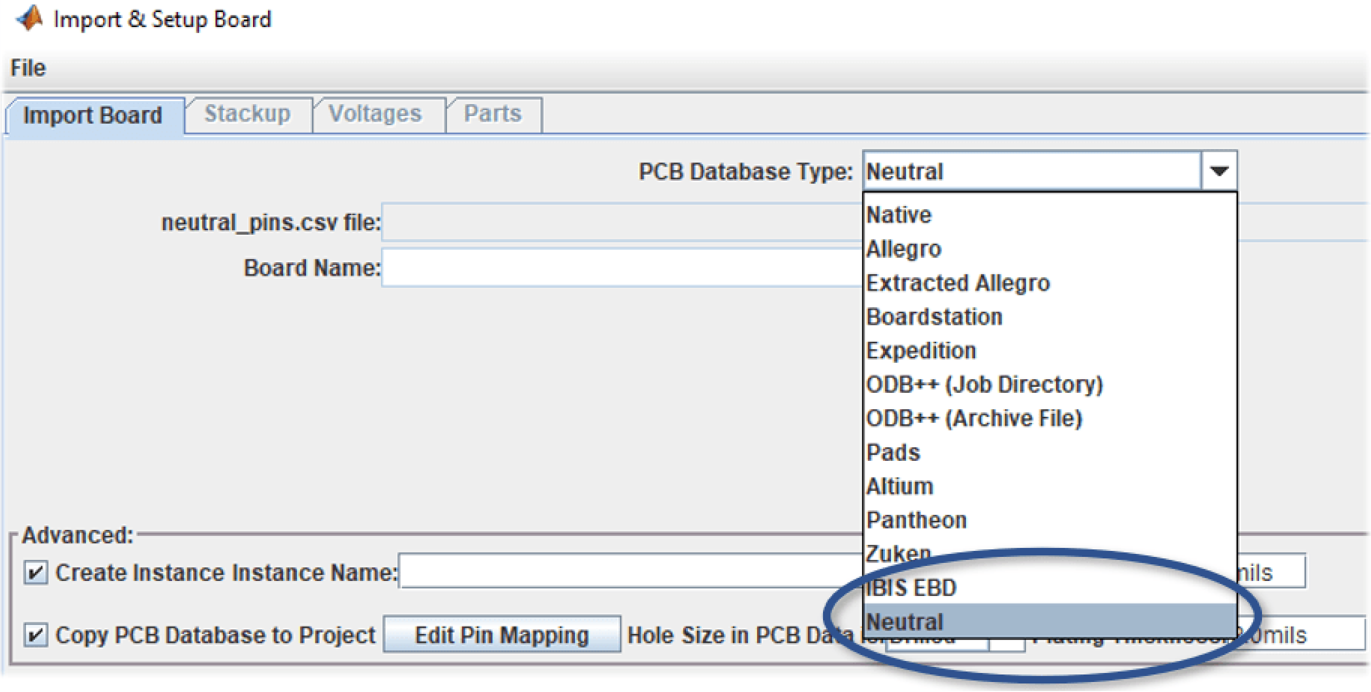
PCB databases are imported from the Setup & Assignment dialog. Click the Setup & Assignment button to launch the dialog.



To import a board click the Import & Setup Board button on the Setup & Assignment dialog.



This will launch the Import & Setup Board dialog with the Import Board tab active. You can select from a number of different formats of PCB database from the **PCB Database Type** dropdown menu. The PCB databases of the Main Board and DIMM attached to this example are in **Parallel Link Designer** "Neutral" format, so select this option from the list.



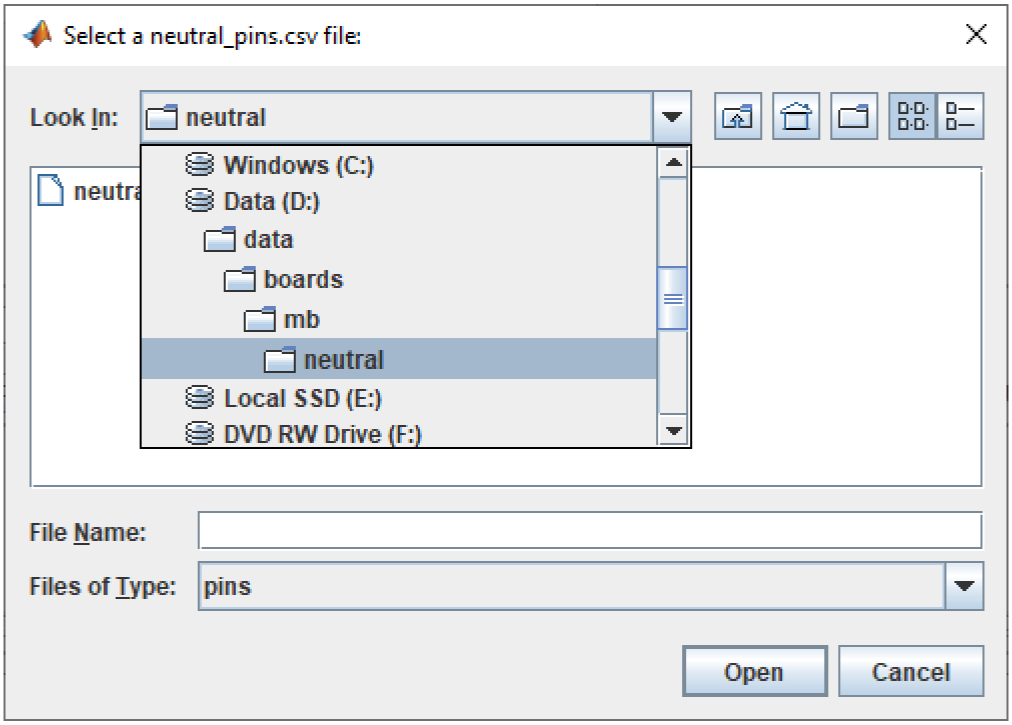
PCB Database Type: Neutral

☑ Create Instance Instance Name:

☑ Copy PCB Database to Project

Attached to this example are two PCB databases, which are provided in **Parallel Link Designer** "Neutral" format as archives "mb.zip" for the Main Board PCB and "dimm.zip" for the DRAM memory DIMM. You download and extract these .zip files into a folder called "boards" for organization purposes.

Click the **Browse** button and navigate to the folder where you downloaded the PCB databases and click **Open** button to select the the boards\mb\neutral directory.

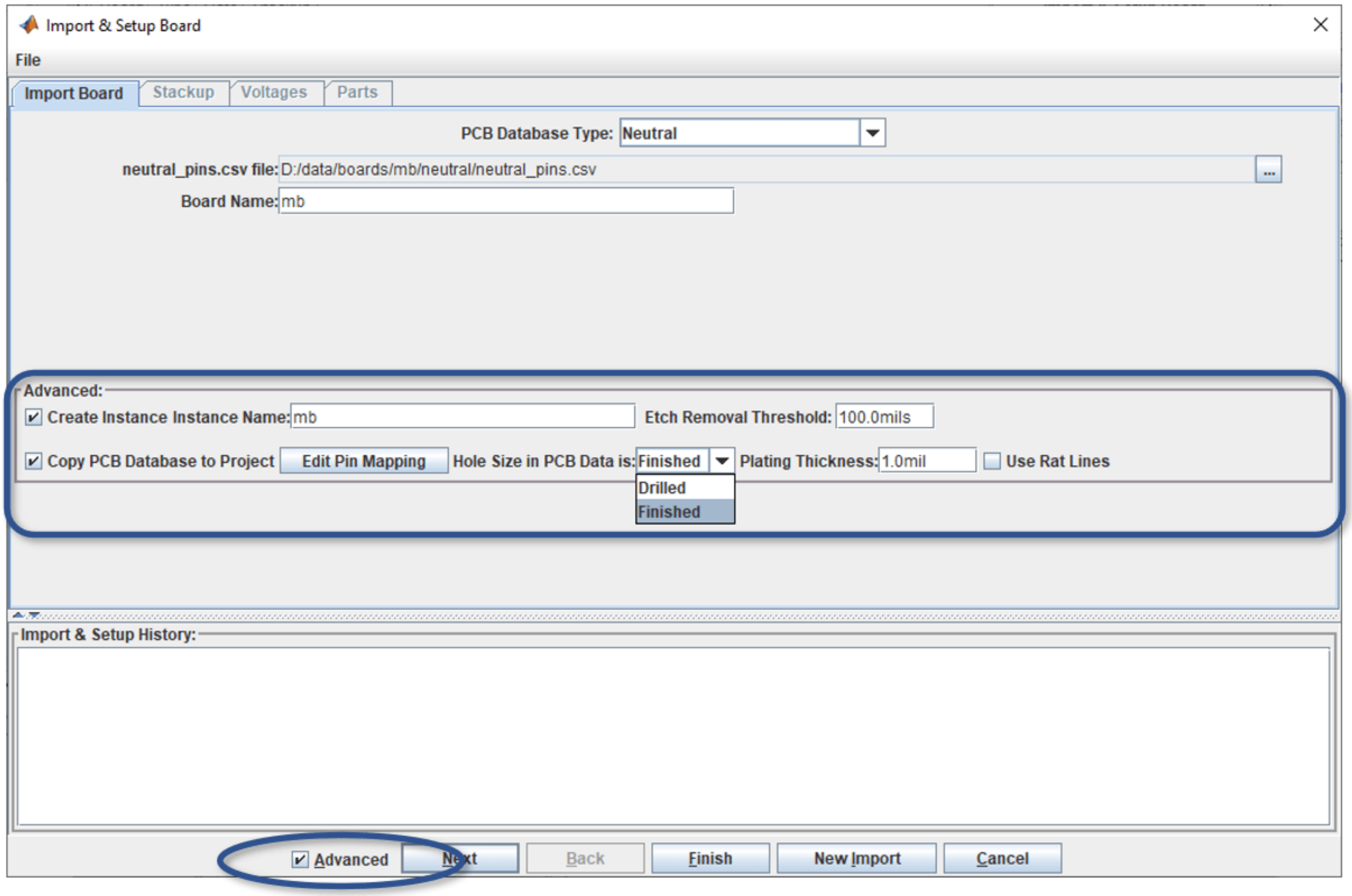


Select a neutral_pins.csv file:

Look In: neutral

File Name:

Files of Type: pins



PCB Database Type: Neutral

neutral_pins.csv file: D:/data/boards/mb/neutral/neutral_pins.csv

Board Name: mb

Advanced:

☑ Create Instance Instance Name: mb

Etch Removal Threshold: 100.0mils

☑ Copy PCB Database to Project

Hole Size in PCB Data is: Finished

Plating Thickness: 1.0mil

□ Use Rat Lines

Import & Setup History:

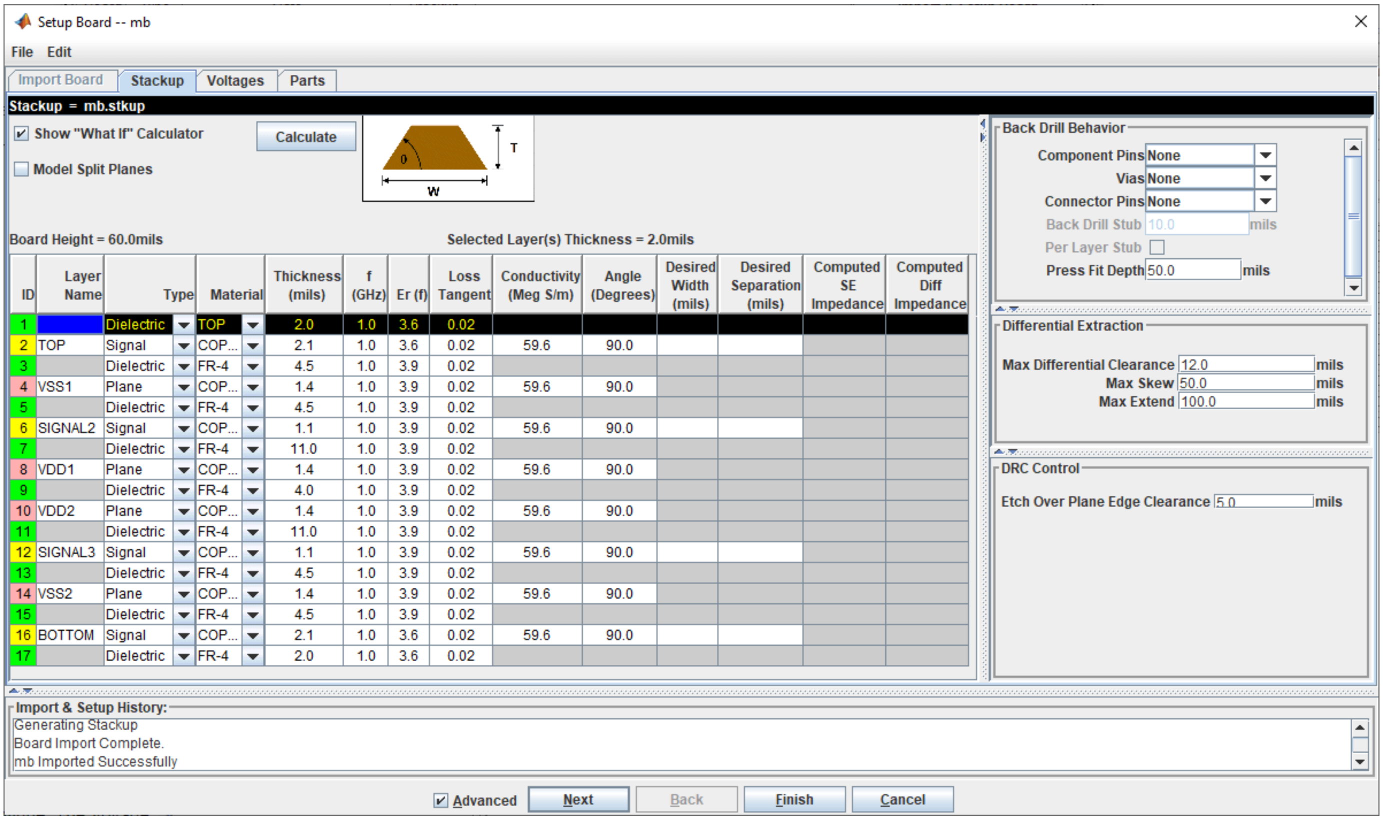
☑ Advanced

Before clicking the **Next** button to import a board, you may wish to select the "Advanced" option. This displays options for:

- Create a new **Instance Name**
- Copy original PCB Database to the Project folder
- Set PCB fabrication option for **Etch Removal Threshold**
- Declare Padstack Hole Size in PCB database as **Drilled** or **Finished**
- Set **Padstack Plating Thickness**
- Option to determine connectivity by using CAD **Rat Lines** if traces are not yet laid out

Click the **Next** button at the bottom of the Import & Setup dialog to read the PCB database and generate the PCB stackup information.

When the PCB database has completed importing, the Stackup tab will appear. This shows the PCB stackup information that Parallel Link Designer read from the PCB database.



Setup Board -- mb

File Edit

Import Board | Stackup | Voltages | Parts

Stackup = mb.stkup

☑ Show "What If" Calculator

□ Model Split Planes

Board Height = 60.0mils

Selected Layer(s) Thickness = 2.0mils

ID	Layer Name	Type	Material	Thickness (mils)	f (GHz)	Er (f)	Loss Tangent	Conductivity (Meg S/m)	Angle (Degrees)	Desired Width (mils)	Desired Separation (mils)	Computed SE Impedance	Computed Diff Impedance
1		Dielectric	TOP	2.0	1.0	3.6	0.02						
2	TOP	Signal	COP...	2.1	1.0	3.6	0.02	59.6	90.0				
3		Dielectric	FR-4	4.5	1.0	3.9	0.02						
4	VSS1	Plane	COP...	1.4	1.0	3.9	0.02	59.6	90.0				
5		Dielectric	FR-4	4.5	1.0	3.9	0.02						
6	SIGNAL2	Signal	COP...	1.1	1.0	3.9	0.02	59.6	90.0				
7		Dielectric	FR-4	11.0	1.0	3.9	0.02						
8	VDD1	Plane	COP...	1.4	1.0	3.9	0.02	59.6	90.0				
9		Dielectric	FR-4	4.0	1.0	3.9	0.02						
10	VDD2	Plane	COP...	1.4	1.0	3.9	0.02	59.6	90.0				
11		Dielectric	FR-4	11.0	1.0	3.9	0.02						
12	SIGNAL3	Signal	COP...	1.1	1.0	3.9	0.02	59.6	90.0				
13		Dielectric	FR-4	4.5	1.0	3.9	0.02						
14	VSS2	Plane	COP...	1.4	1.0	3.9	0.02	59.6	90.0				
15		Dielectric	FR-4	4.5	1.0	3.9	0.02						
16	BOTTOM	Signal	COP...	2.1	1.0	3.6	0.02	59.6	90.0				
17		Dielectric	FR-4	2.0	1.0	3.6	0.02						

Back Drill Behavior

Component Pins None
Vias None
Connector Pins None
Back Drill Stub 10.0 mils
Per Layer Stub □
Press Fit Depth 50.0 mils

Differential Extraction

Max Differential Clearance 12.0 mils
Max Skew 50.0 mils
Max Extend 100.0 mils

DRC Control

Etch Over Plane Edge Clearance 5.0 mils

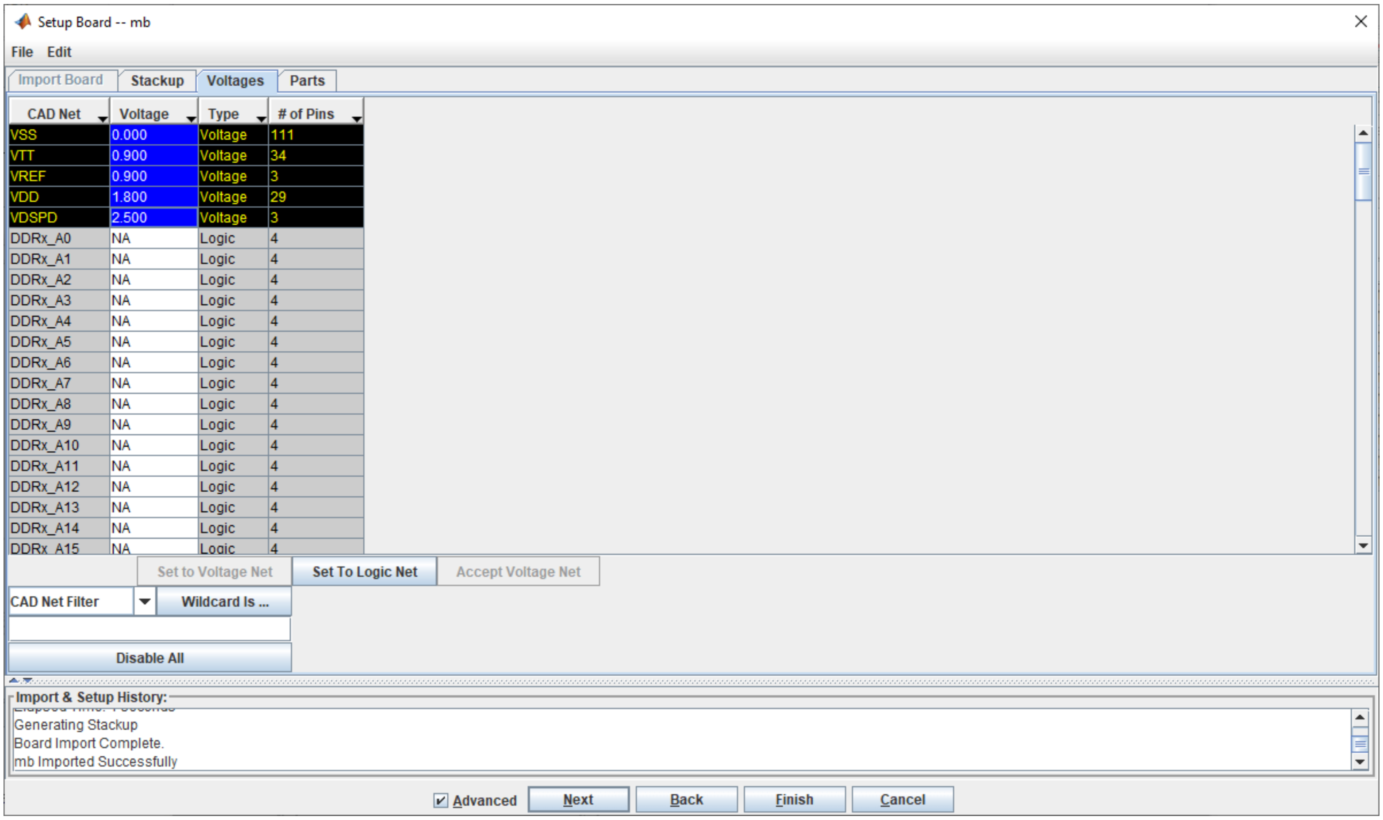
Import & Setup History:

Generating Stackup
Board Import Complete.
mb Imported Successfully

☑ Advanced

You may need to add a layer to the top and bottom of the stackup that represents dielectric or solder mask. Typically its depth ranges between 0.5 to 2.0 mils. You also may need to make corrections to Dielectric Constant (Er) or Loss Tangent (or Dissipation Factor, tan δ) for layers not configured by CAD in the original PCB database.

In this case, no changes to the PCB stackup are needed, so click the **Next** button at the bottom of the Import & Setup Board dialog to continue. The Voltage Nets tab will appear (See Figure).



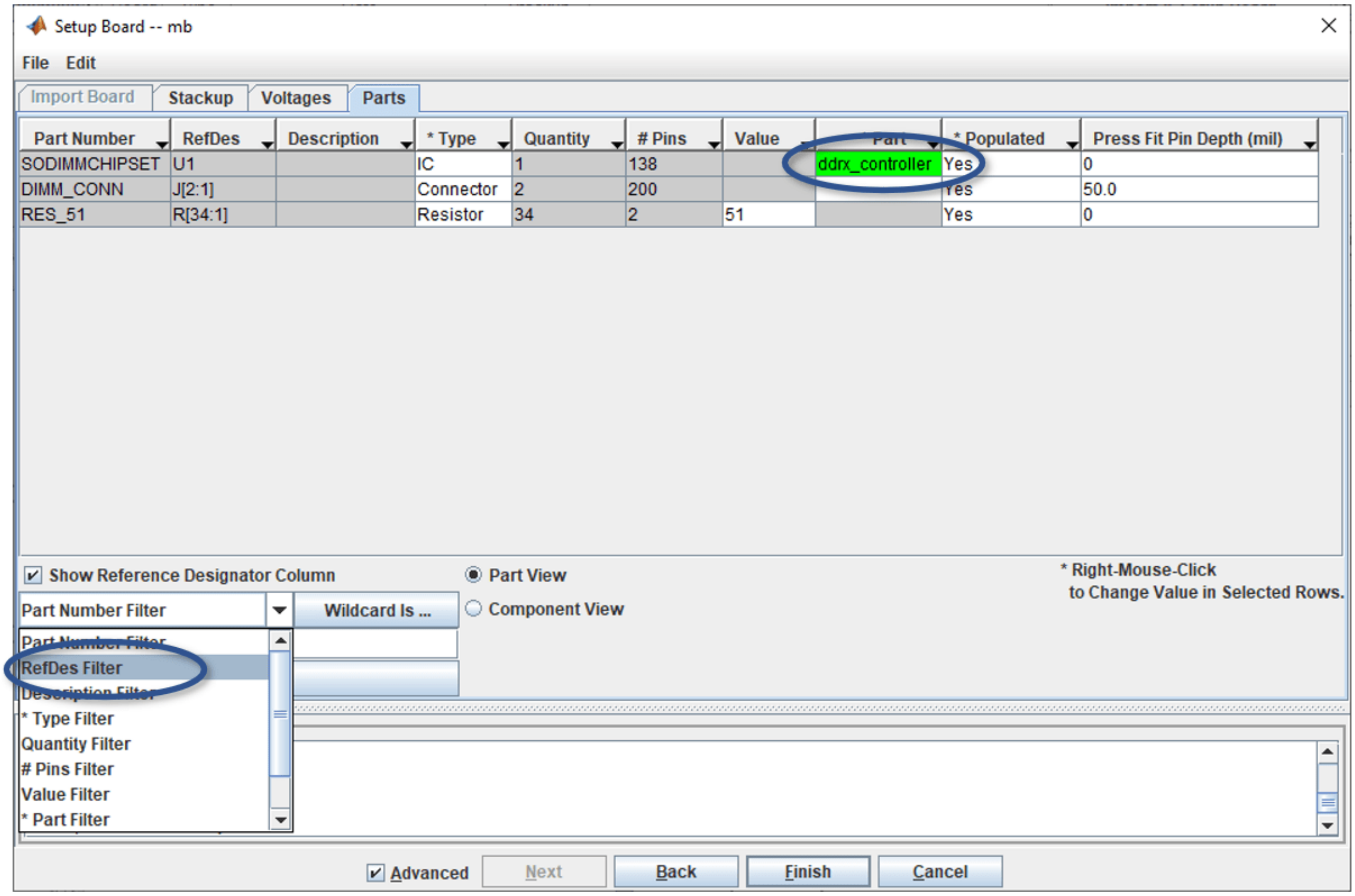
CAD Net	Voltage	Type	# of Pins
VSS	0.000	Voltage	111
VTT	0.900	Voltage	34
VREF	0.900	Voltage	3
VDD	1.800	Voltage	29
VDSPD	2.500	Voltage	3
DDRx_A0	NA	Logic	4
DDRx_A1	NA	Logic	4
DDRx_A2	NA	Logic	4
DDRx_A3	NA	Logic	4
DDRx_A4	NA	Logic	4
DDRx_A5	NA	Logic	4
DDRx_A6	NA	Logic	4
DDRx_A7	NA	Logic	4
DDRx_A8	NA	Logic	4
DDRx_A9	NA	Logic	4
DDRx_A10	NA	Logic	4
DDRx_A11	NA	Logic	4
DDRx_A12	NA	Logic	4
DDRx_A13	NA	Logic	4
DDRx_A14	NA	Logic	4
DDRx_A15	NA	Logic	4

Generating Stackup
Board Import Complete.
mb Imported Successfully

Parallel Link Designer has read the voltage properties and attempts to automatically parse the voltage net names to set their values. It is important to verify the values of your specific database are correct. For DDRx analysis, correct values for VDD, VSS, and VTT are often required for correct results to be achieved. Also, if there are active terminations in the circuits to be analyzed such as a system clock or various logic families, then the voltages must be correct in order for SPICE simulation to provide valid transient waveform data. Review this list for accuracy and verify that all voltage nets have been defined as Type Voltage, and that their voltage value is correct.

All voltages on this PCB have been imported correctly. Click the **Next** button at the bottom of the Import & Setup Board dialog to continue.

The Parts tab will appear. A part must exist on each board or each end of a Transfer Net for setup and assignment to complete successfully.



Part Number	RefDes	Description	* Type	Quantity	# Pins	Value	Part	* Populated	Press Fit Pin Depth (mil)
SODIMMCHIPSET	U1		IC	1	138		ddrx_controller	Yes	0
DIMM_CONN	J[2:1]		Connector	2	200			Yes	50.0
RES_51	R[34:1]		Resistor	34	2	51		Yes	0

☑ Show Reference Designator Column

● Part View

○ Component View

* Right-Mouse-Click
to Change Value in Selected Rows.

Part Number Filter

- Part Number Filter
- RefDes Filter
- Description Filter
- * Type Filter
- Quantity Filter
- # Pins Filter
- Value Filter
- * Part Filter

☑ Advanced

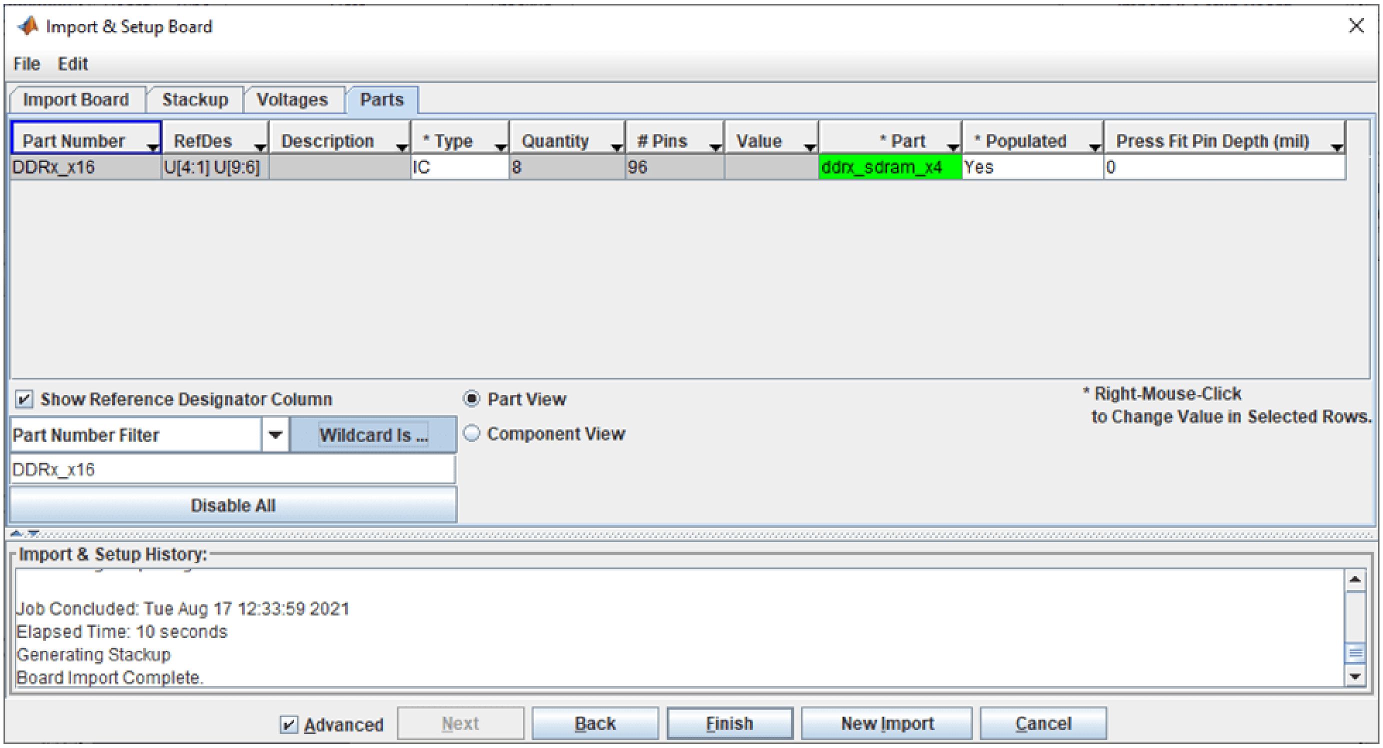
The example Main Board has devices with different part numbers on it: the memory controller, the two DIMM connectors. The **Part**, or library element, is filled in for the controller. **Parallel Link Designer** can automatically match the controller library element to the device through the part number. If you do not see the controller "ddrx_controller" listed, you can right-click and browse to it in the parts list of the project. If starting from scratch or with your own project, you may need to create a new **Part** in the project- please reference the User Guide on this topic as it is beyond the scope of this example.

Note: You can use the drop down to select how to search for parts using a Wildcard string: by Part Number, CAD RefDes, Description, etc.

The Main Board is now imported and set up. Click the **Finish** button on the bottom of the I**mport & Setup Board** dialog to go back to the **Setup & Assignment** dialog. You will see the board "mb" in the table in the **Setup & Assignment** dialog. After you import the DIMM, you will be able to add the ddrx_sdram part to the DIMM in its parts tab using these steps as a guide.

Import DIMM

Now follow the same steps you used to import the Main Board to import the DIMM (there is a single DIMM PCB database but you will create two instances of it in the project). You may see some warnings displayed but they are beyond scope of this example and may be ignored. Be sure to verify the values on the Voltages Tab and to configure the part ddrx_sdram in the Parts Tab for the DIMM board.



Part Number	RefDes	Description	* Type	Quantity	# Pins	Value	* Part	* Populated	Press Fit Pin Depth (mil)
DDRx_x16	U[4:1] U[9:6]		IC	8	96		ddrx_sdram_x4	Yes	0

☑ Show Reference Designator Column

● Part View

○ Component View

* Right-Mouse-Click
to Change Value in Selected Rows.

Part Number Filter: DDRx_x16

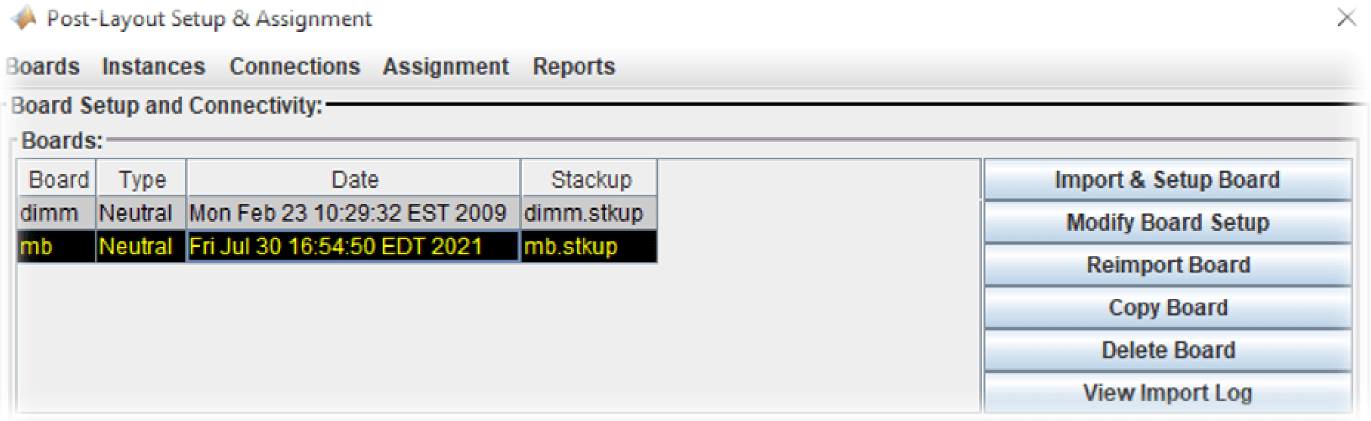
Import & Setup History:

```
Job Concluded: Tue Aug 17 12:33:59 2021
Elapsed Time: 10 seconds
Generating Stackup
Board Import Complete.
```

☑ Advanced

Figure: Set the DIMM Part Number DDRx_x16 to Part ddrx_sdram_x4 from the project library.

You will now have two boards in the table in the Boards section of the Setup & Assignment dialog ready to create instances for connectivity.



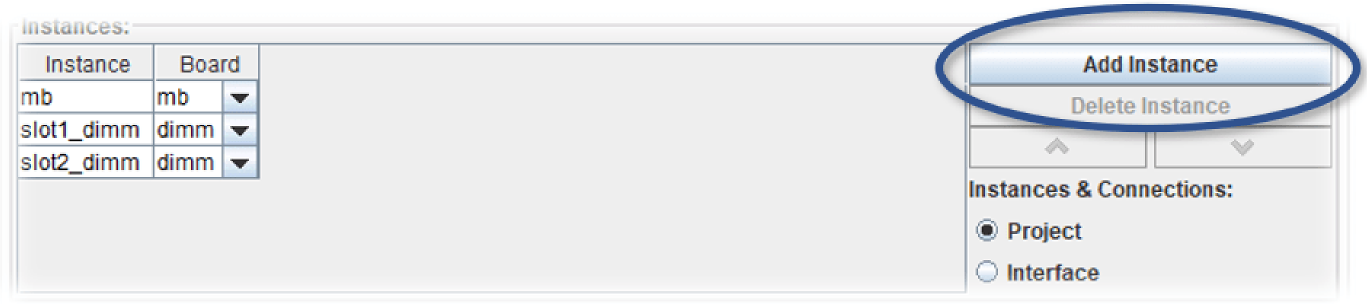
Board	Type	Date	Stackup
dimm	Neutral	Mon Feb 23 10:29:32 EST 2009	dimm.stkup
mb	Neutral	Fri Jul 30 16:54:50 EDT 2021	mb.stkup

Create and Connect Instances

Instances are instantiations of a Board used by the project to establish connectivity and setup CAD net assignment. An instance of each Board has already been created automatically when the PCB database was imported. You will need to create a second instance of the DIMM so that there are two available to connect to the Main Board. You may need to rename the two DIMM instances so that each is clearly identifiable when you setup connectivity to the Main Board instance.

Create New DIMM Instance

You can create a second instance of the DIMM in order to connect two DIMMs to the Main Board. In the Instances area of the **Setup & Assignment** dialog select the Instance of the DIMM and click the **Add Instance** button.



Instance	Board
mb	mb
slot1_dimm	dimm
slot2_dimm	dimm

Instances & Connections:

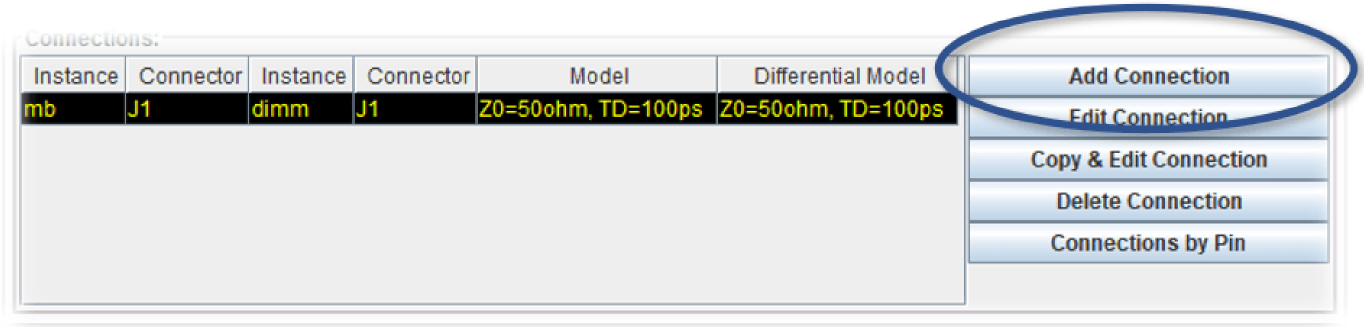
● Project

○ Interface

Change the names of each DIMM instance to slot1_dimm and slot2_dimm (as shown in the Figure).

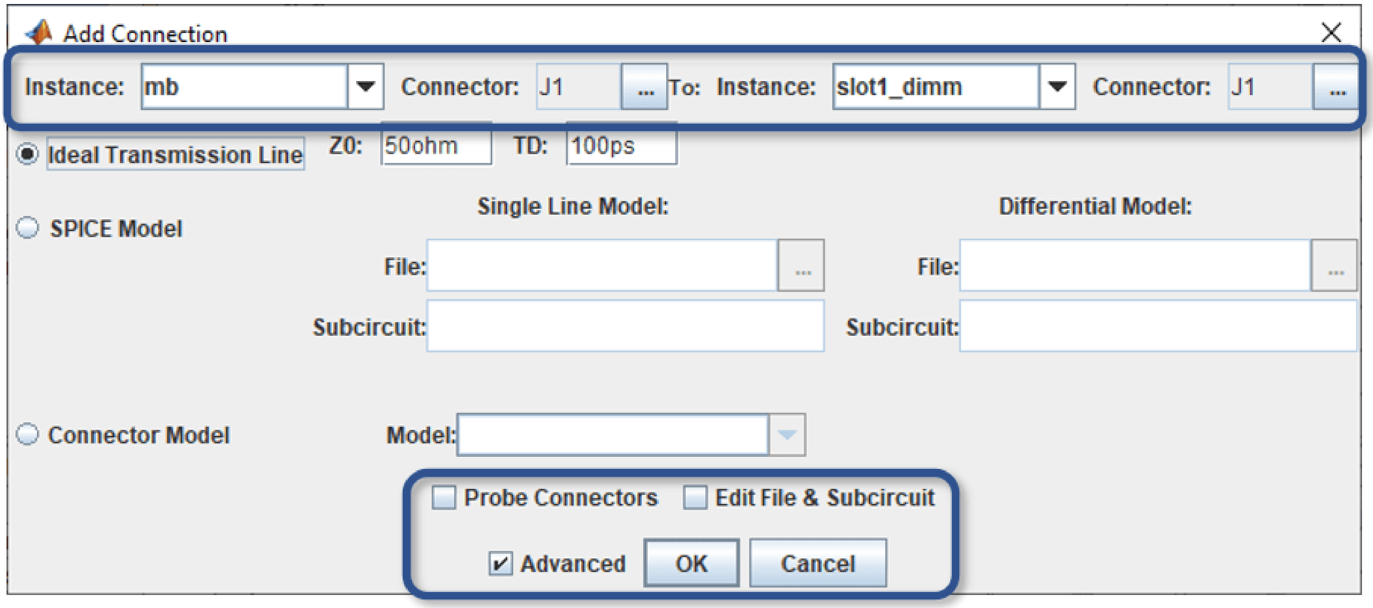
Connect Instances

The DIMM instances can now be connected to the Main Board. In the **Connections** area of the **Setup & Assignment** dialog.



Instance	Connector	Instance	Connector	Model	Differential Model
mb	J1	dimm	J1	Z0=50ohm, TD=100ps	Z0=50ohm, TD=100ps

Click the Add Connection button. This will launch the Add Connection dialog. To create the connection for slot1, select Instance mb in the left hand Instance list.



Instance: mb Connector: J1 To: Instance: slot1_dimm Connector: J1

● Ideal Transmission Line Z0: 50ohm TD: 100ps

○ SPICE Model

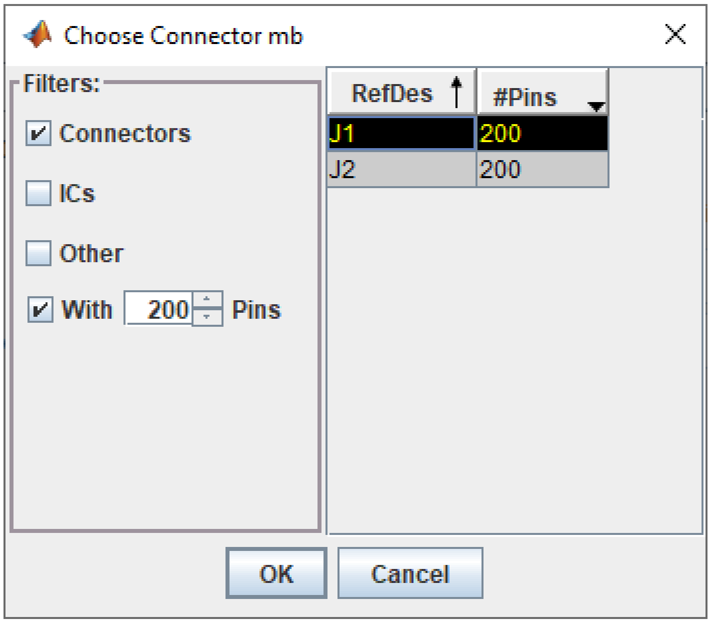
○ Connector Model

□ Probe Connectors □ Edit File & Subcircuit

☑ Advanced

Figure : Add Connection Dialog: enable the Advanced checkbox to view options for Probe Connectors and Edit File & Subcircuit

Click the **Browse** button to select a connector on the Instance mb. The Choose Connector dialog lets you select the connector on the Instance mb that will be connected to the DIMM. Select J1.



☑ Connectors

□ ICs

□ Other

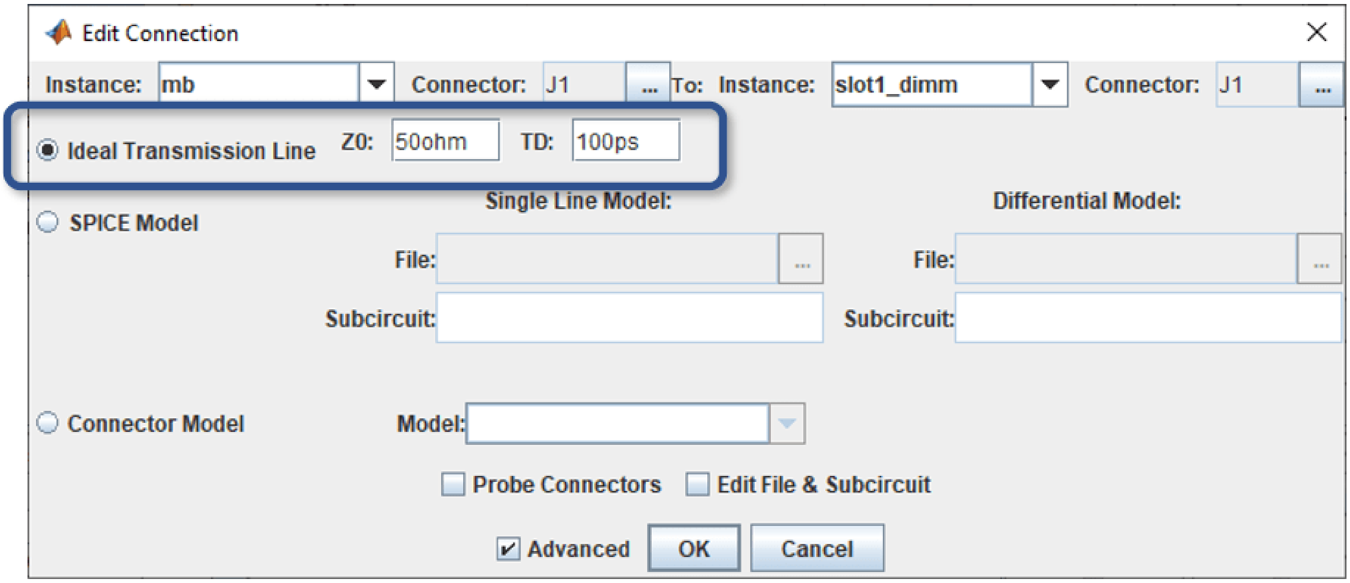
☑ With 200 Pins

RefDes	#Pins
J1	200
J2	200

Click **OK**.

Now select the Instance and connector on the right-hand side. Select the Instance slot1_dimm and the connector J1.

The last thing to specify for the connection is the connector model. You can select Spice Model or S-Parameter file. In this example, the connector is modelled as an ideal transmission line model.



Edit Connection

Instance: mb Connector: J1 To: Instance: slot1_dimm Connector: J1

● Ideal Transmission Line Z0: 50ohm TD: 100ps

○ SPICE Model

Single Line Model: File: Subcircuit:

Differential Model: File: Subcircuit:

○ Connector Model Model:

□ Probe Connectors □ Edit File & Subcircuit

☑ Advanced

Click **OK** to complete the connection.

Now create a connection between the Main Board and the second DIMM:

1. Click Add Connection.

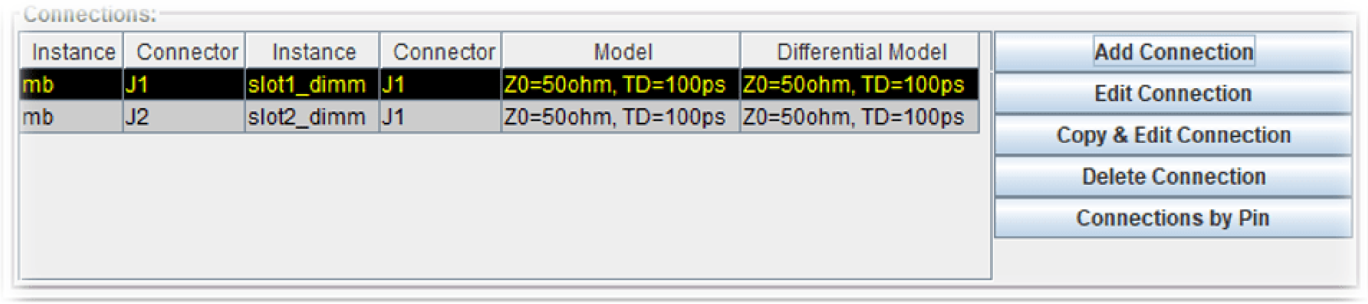
2. Select Instance mb and connector J2 on the left.

3. Select Instance slot2_dimm and connector J1 on the right.

4. Select ideal transmission line connector model.

5. Click OK to complete the second connection.

The **Connections** area of the **Setup & Assignment** dialog will have two connections in the table, as shown in the figure.



Connections:

Instance	Connector	Instance	Connector	Model	Differential Model
mb	J1	slot1_dimm	J1	Z0=50ohm, TD=100ps	Z0=50ohm, TD=100ps
mb	J2	slot2_dimm	J1	Z0=50ohm, TD=100ps	Z0=50ohm, TD=100ps

Setup and Run Assignment

Now that the connectivity is specified Parallel Link Designer can extract the connectivity from the multi-board system. Parallel Link Designer will also match the nets extracted from the boards to the Transfer Nets in the Design Kit so that the Transfer Net properties can be used in simulation. This process is called Assignment. By default, all nets will be extracted from the PCB database(s), however you can select which nets to include in the assignment using the **Assigment Setup** button.

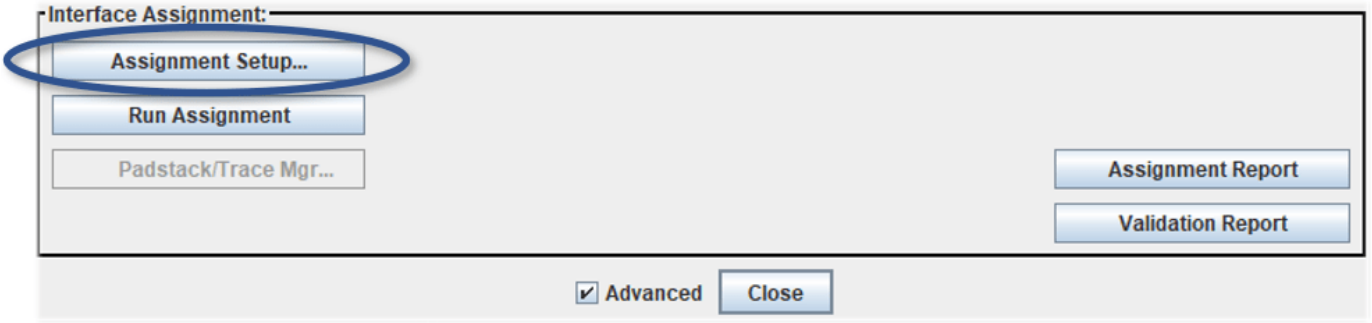
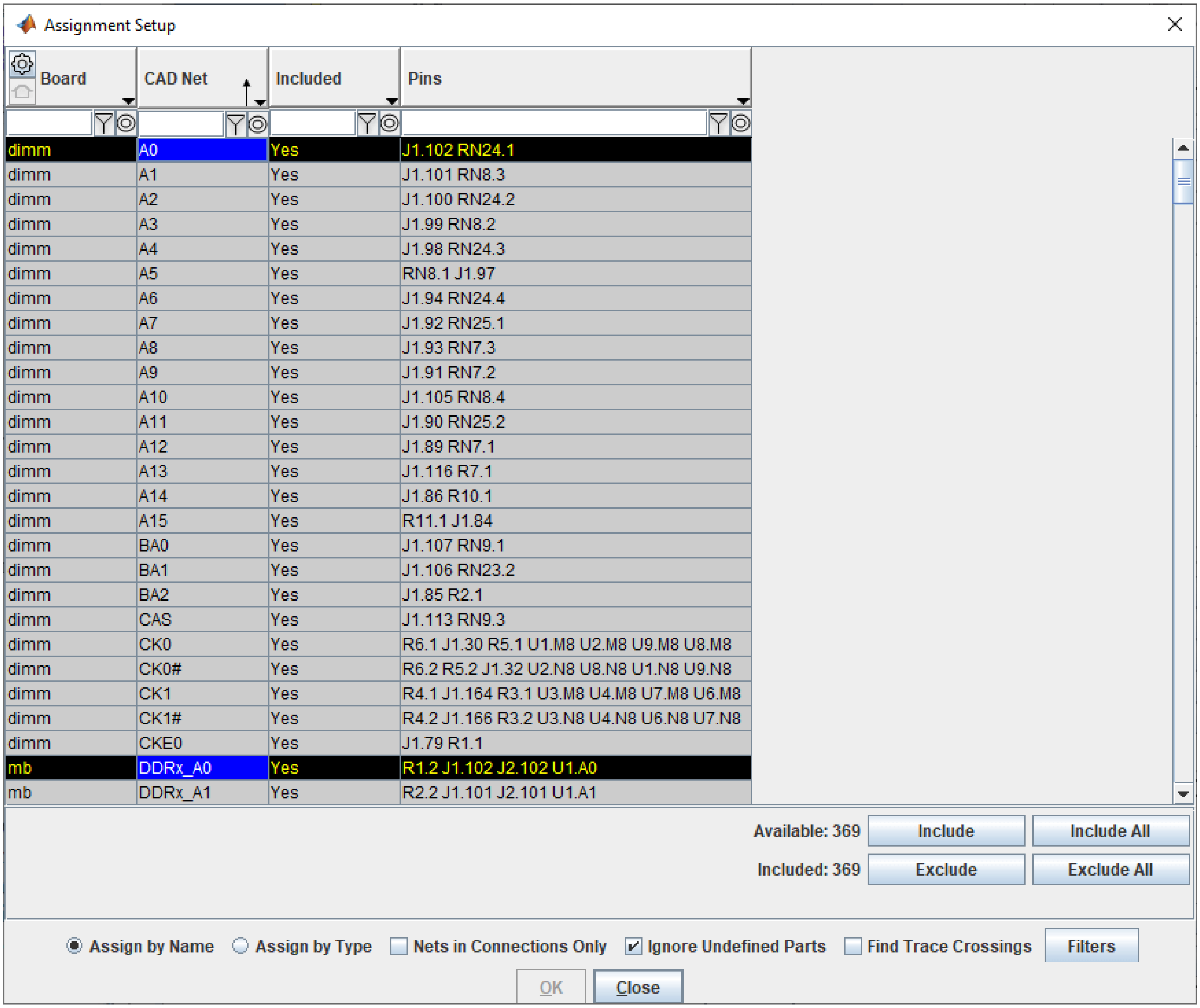


Figure: Assignment Setup Button



Board	CAD Net	Included	Pins
dimm	A0	Yes	J1.102 RN24.1
dimm	A1	Yes	J1.101 RN8.3
dimm	A2	Yes	J1.100 RN24.2
dimm	A3	Yes	J1.99 RN8.2
dimm	A4	Yes	J1.98 RN24.3
dimm	A5	Yes	RN8.1 J1.97
dimm	A6	Yes	J1.94 RN24.4
dimm	A7	Yes	J1.92 RN25.1
dimm	A8	Yes	J1.93 RN7.3
dimm	A9	Yes	J1.91 RN7.2
dimm	A10	Yes	J1.105 RN8.4
dimm	A11	Yes	J1.90 RN25.2
dimm	A12	Yes	J1.89 RN7.1
dimm	A13	Yes	J1.116 R7.1
dimm	A14	Yes	J1.86 R10.1
dimm	A15	Yes	R11.1 J1.84
dimm	BA0	Yes	J1.107 RN9.1
dimm	BA1	Yes	J1.106 RN23.2
dimm	BA2	Yes	J1.85 R2.1
dimm	CAS	Yes	J1.113 RN9.3
dimm	CK0	Yes	R6.1 J1.30 R5.1 U1.M8 U2.M8 U9.M8 U8.M8
dimm	CK0#	Yes	R6.2 R5.2 J1.32 U2.N8 U8.N8 U1.N8 U9.N8
dimm	CK1	Yes	R4.1 J1.164 R3.1 U3.M8 U4.M8 U7.M8 U6.M8
dimm	CK1#	Yes	R4.2 J1.166 R3.2 U3.N8 U4.N8 U6.N8 U7.N8
dimm	CKE0	Yes	J1.79 R1.1
mb	DDRx_A0	Yes	R1.2 J1.102 J2.102 U1.A0
mb	DDRx_A1	Yes	R2.2 J1.101 J2.101 U1.A1

Available: 369

Included: 369

● Assign by Name ○ Assign by Type □ Nets in Connections Only ☑ Ignore Undefined Parts □ Find Trace Crossings

Figure: Select Nets to Include or Exclude for Assignment

In this dialog, you can use the filter and wildcard entries at the top of each column to find nets to include in the simulation. For example, you may wish to select only DDRx-related nets for a larger PCB database if it contains thousands of CAD nets. This would reduce the database size in Parallel Link Designer and optimize other operations for speed, such as board viewer loading the database. To run Assignment click the **Run Assignment** button on the **Setup & Assignment** dialog.

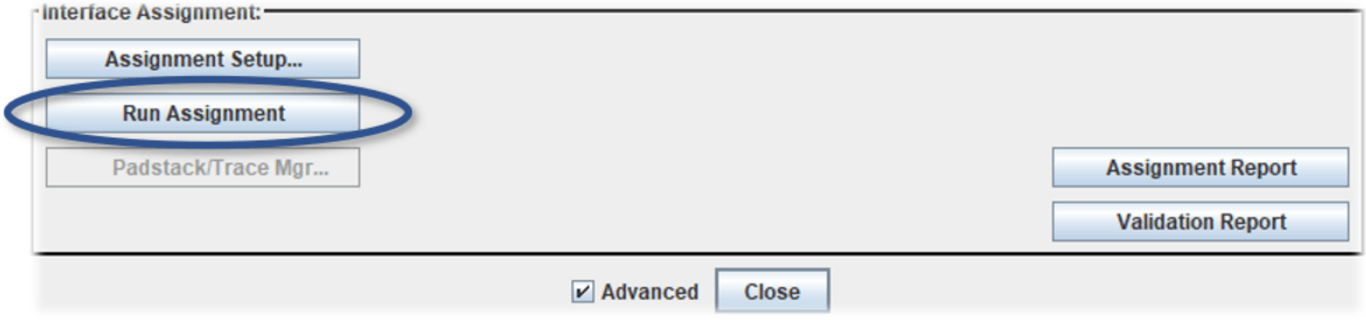
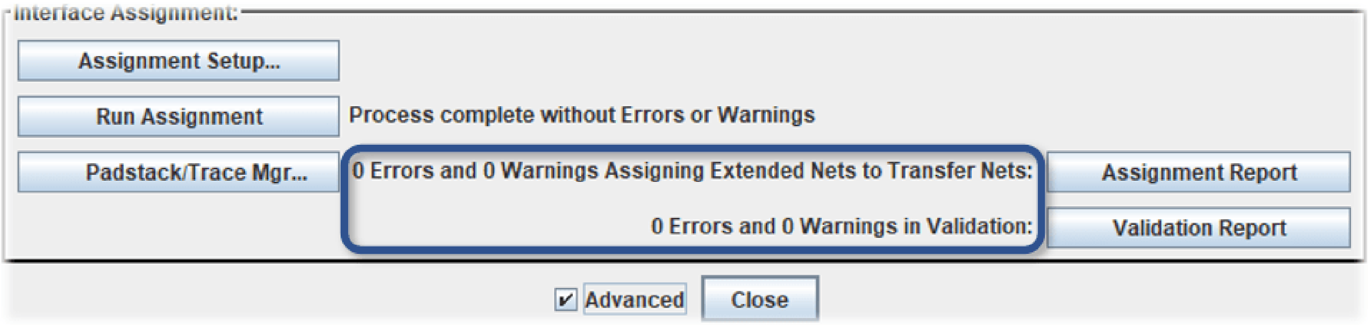


Figure: Run Assignment Button

When the Assignment process is complete the Assignment and Validation Reports are created.



Process complete without Errors or Warnings

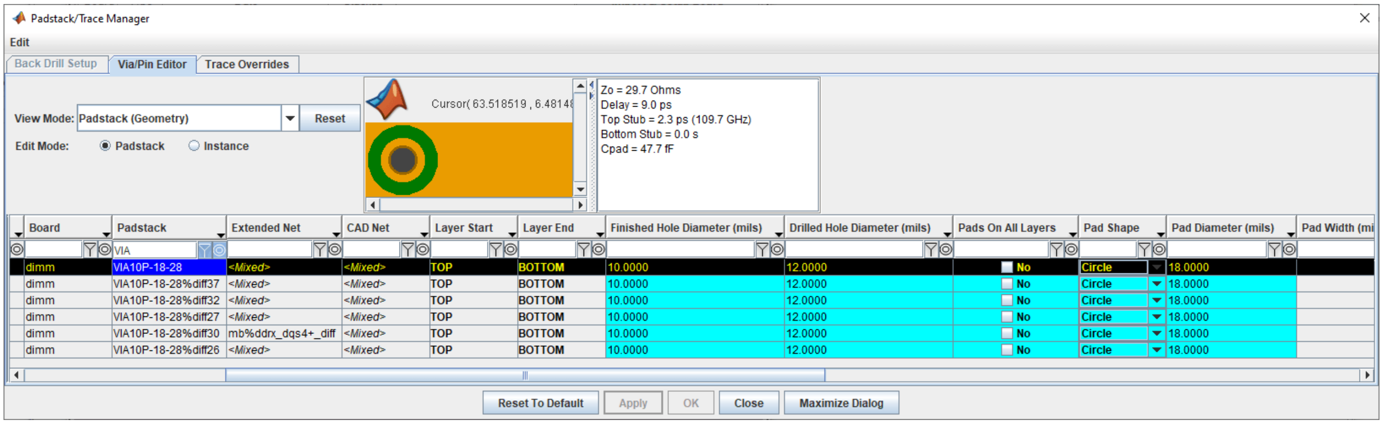
0 Errors and 0 Warnings Assigning Extended Nets to Transfer Nets:

0 Errors and 0 Warnings in Validation:

It is always a good idea to check the status after running Assignment. It can alert you to problems that may be difficult to debug without seeing the error messages. For example, if there are warnings listed they may represent unused parts. Even if you see no errors or warnings it is wise to review the reports so that you clearly understand the project status.

You can use custom models (such as SPICE subcircuit or Touchstone s-parameter files) in your PCB database for traces and padstacks (vias). These are called Model Overrides in the User Guide, which covers this topic in more detail which is out of scope for this example.

Note: The button is enabled for openning the **Padstack/Trace Manager** after **Run Assignment** has completed.



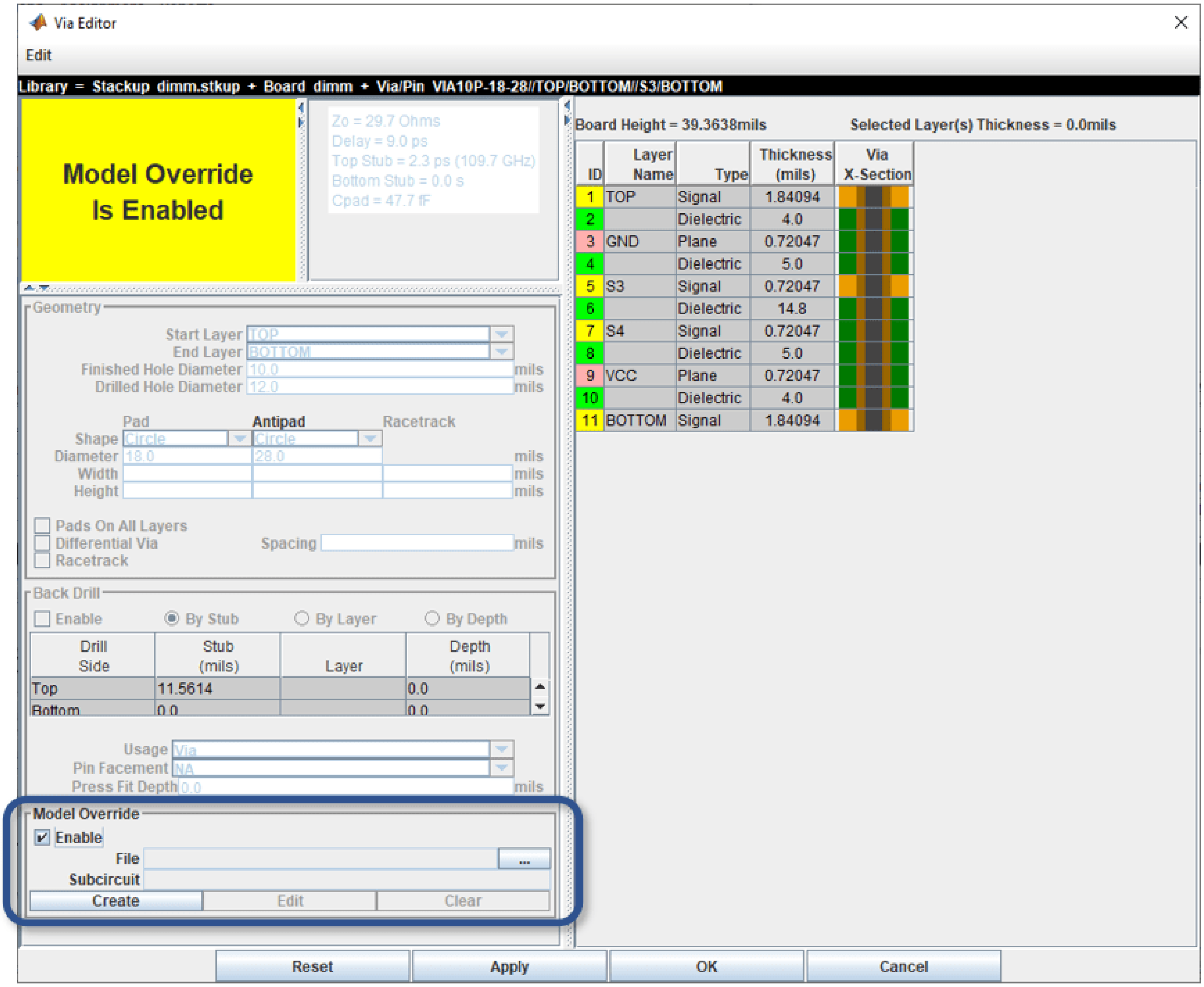
View Mode: Padstack (Geometry)

Edit Mode: ● Padstack ○ Instance

Zo = 29.7 Ohms
Delay = 9.0 ps
Top Stub = 2.3 ps (109.7 GHz)
Bottom Stub = 0.0 s
Cpad = 47.7 fF

Board	Padstack	Extended Net	CAD Net	Layer Start	Layer End	Finished Hole Diameter (mils)	Drilled Hole Diameter (mils)	Pads On All Layers	Pad Shape	Pad Diameter (mils)	Pad Width (mi
	VIA										
dimm	VIA10P-18-28	<Mixed>	<Mixed>	TOP	BOTTOM	10.0000	12.0000	□ No	Circle	18.0000	
dimm	VIA10P-18-28%diff37	<Mixed>	<Mixed>	TOP	BOTTOM	10.0000	12.0000	□ No	Circle	18.0000	
dimm	VIA10P-18-28%diff32	<Mixed>	<Mixed>	TOP	BOTTOM	10.0000	12.0000	□ No	Circle	18.0000	
dimm	VIA10P-18-28%diff27	<Mixed>	<Mixed>	TOP	BOTTOM	10.0000	12.0000	□ No	Circle	18.0000	
dimm	VIA10P-18-28%diff30	mb%ddrx_dqs4+_diff	<Mixed>	TOP	BOTTOM	10.0000	12.0000	□ No	Circle	18.0000	
dimm	VIA10P-18-28%diff26	<Mixed>	<Mixed>	TOP	BOTTOM	10.0000	12.0000	□ No	Circle	18.0000	

This allows you to see all the Padstacks(Vias) in the design. You can right-click to open the **Via Model Editor**, and check the option to use set a **Model Override** for a padstack (via) or trace instance or by occurrence (e.g. to override all padstacks having the same library name).



Via Editor

Edit

Library = Stackup dimm.stkup + Board dimm + Via/Pin VIA10P-18-28//TOP/BOTTOM//S3/BOTTOM

Model Override Is Enabled

Zo = 29.7 Ohms
Delay = 9.0 ps
Top Stub = 2.3 ps (109.7 GHz)
Bottom Stub = 0.0 s
Cpad = 47.7 fF

Geometry

Start Layer: TOP
End Layer: BOTTOM
Finished Hole Diameter: 10.0 mils
Drilled Hole Diameter: 12.0 mils

	Pad	Antipad	Racetrack	
Shape	Circle	Circle		
Diameter	18.0	28.0		mils
Width				mils
Height				mils

□ Pads On All Layers
□ Differential Via — Spacing: mils
□ Racetrack

Back Drill

□ Enable ● By Stub ○ By Layer ○ By Depth

Drill Side	Stub (mils)	Layer	Depth (mils)
Top	11.5614		0.0
Bottom	0.0		0.0

Usage: Via
Pin Facement: NA
Press Fit Depth: 0.0 mils

Model Override

☑ Enable
File
Subcircuit

Board Height = 39.3638mils Selected Layer(s) Thickness = 0.0mils

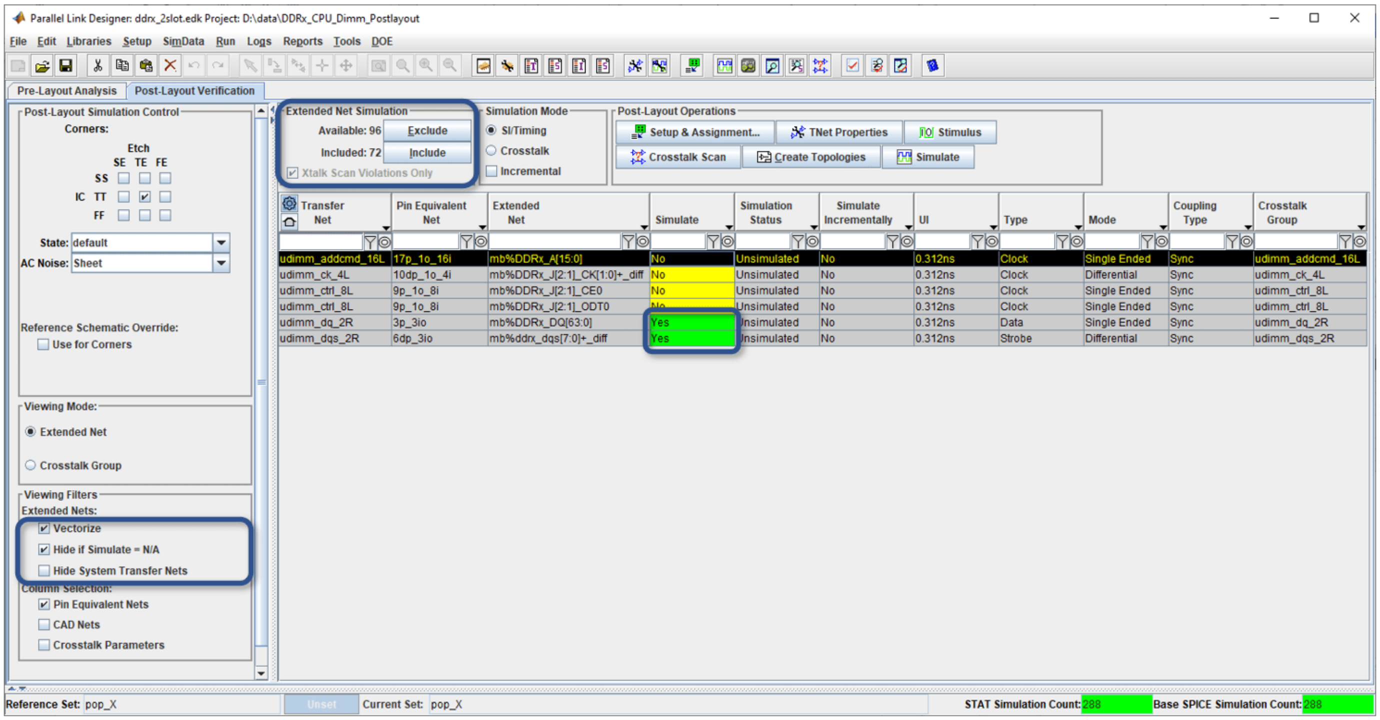
ID	Layer Name	Type	Thickness (mils)	Via X-Section
1	TOP	Signal	1.84094	🟧
2		Dielectric	4.0	🟩
3	GND	Plane	0.72047	🟩
4		Dielectric	5.0	🟩
5	S3	Signal	0.72047	🟧
6		Dielectric	14.8	🟩
7	S4	Signal	0.72047	🟩
8		Dielectric	5.0	🟩
9	VCC	Plane	0.72047	🟩
10		Dielectric	4.0	🟩
11	BOTTOM	Signal	1.84094	🟧

Click the **Ok** button to close the **Via Editor** dialog.

Click the **Close** button to close the **Padstack/Trace Manager** dialog.

Click the **Close** button to close the **Setup & Assignment** dialog.

On the **Post-Layout Verification** tab there is a table with all of the nets listed.



Parallel Link Designer: ddrx_2slot.edk Project: D:\data\DDRx_CPU_Dimm_Postlayout

Post-Layout Verification

Post-Layout Simulation Control
Corners:

		SE	TE	FE
	SS	☐	☐	☐
IC	TT	☐	☑	☐
	FF	☐	☐	☐

State: default
AC Noise: Sheet

Reference Schematic Override: ☐ Use for Corners

Viewing Mode: ● Extended Net ○ Crosstalk Group

Viewing Filters
Extended Nets:
☑ Vectorize
☑ Hide if Simulate = N/A
☐ Hide System Transfer Nets

Column Selection:
☑ Pin Equivalent Nets
☐ CAD Nets
☐ Crosstalk Parameters

Extended Net Simulation
Available: 96 Exclude
Included: 72 Include
☑ Xtalk Scan Violations Only

Simulation Mode: ● SI/Timing ○ Crosstalk ☐ Incremental

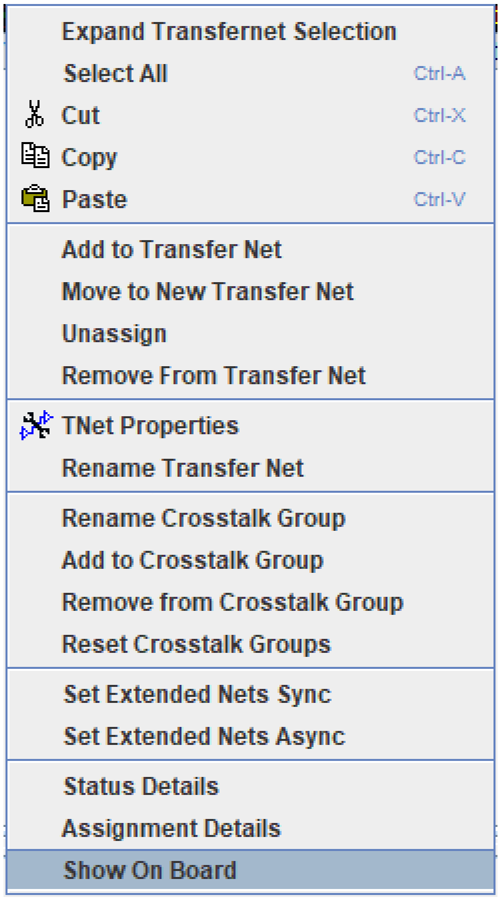
Transfer Net	Pin Equivalent Net	Extended Net	Simulate	Simulation Status	Simulate Incrementally	UI	Type	Mode	Coupling Type	Crosstalk Group
udimm_addcmd_16L	17p_1o_16i	mb%DDRx_A[15:0]	No	Unsimulated	No	0.312ns	Clock	Single Ended	Sync	udimm_addcmd_16L
udimm_ck_4L	10dp_1o_4i	mb%DDRx_J[2:1]_CK[1:0]+_diff	No	Unsimulated	No	0.312ns	Clock	Differential	Sync	udimm_ck_4L
udimm_ctrl_8L	9p_1o_8i	mb%DDRx_J[2:1]_CE0	No	Unsimulated	No	0.312ns	Clock	Single Ended	Sync	udimm_ctrl_8L
udimm_ctrl_8L	9p_1o_8i	mb%DDRx_J[2:1]_ODT0	No	Unsimulated	No	0.312ns	Clock	Single Ended	Sync	udimm_ctrl_8L
udimm_dq_2R	3p_3io	mb%DDRx_DQ[63:0]	Yes	Unsimulated	No	0.312ns	Data	Single Ended	Sync	udimm_dq_2R
udimm_dqs_2R	6dp_3io	mb%ddrx_dqs[7:0]+_diff	Yes	Unsimulated	No	0.312ns	Strobe	Differential	Sync	udimm_dqs_2R

Reference Set: pop_X Current Set: pop_X STAT Simulation Count: 288 Base SPICE Simulation Count: 288

Figure: Post-Layout Tab After Assignment. You can use the Viewing Filters to Vectorize nets by group and to Hide if Simulate = N/A

You can select which Nets or Vectorized Nets to simulate by clicking on a row and presing the **Include** or **Exclude** buttons. You can also check the options in the Viewing Filters pane for Vectorize and Hide if Simulate = N/A. This means that the nets are vectorized so that, for example, all of the dq nets are in one row, and all of the dqs nets are in another row. The option for Hide if Simulate = N/A suppresses any transfer nets that are not valid or otherwise unusable in the current set of assigned nets.

To view any set of nets on the board, highlight the rows you wish to see and select "Show on Board" from the right-click menu.



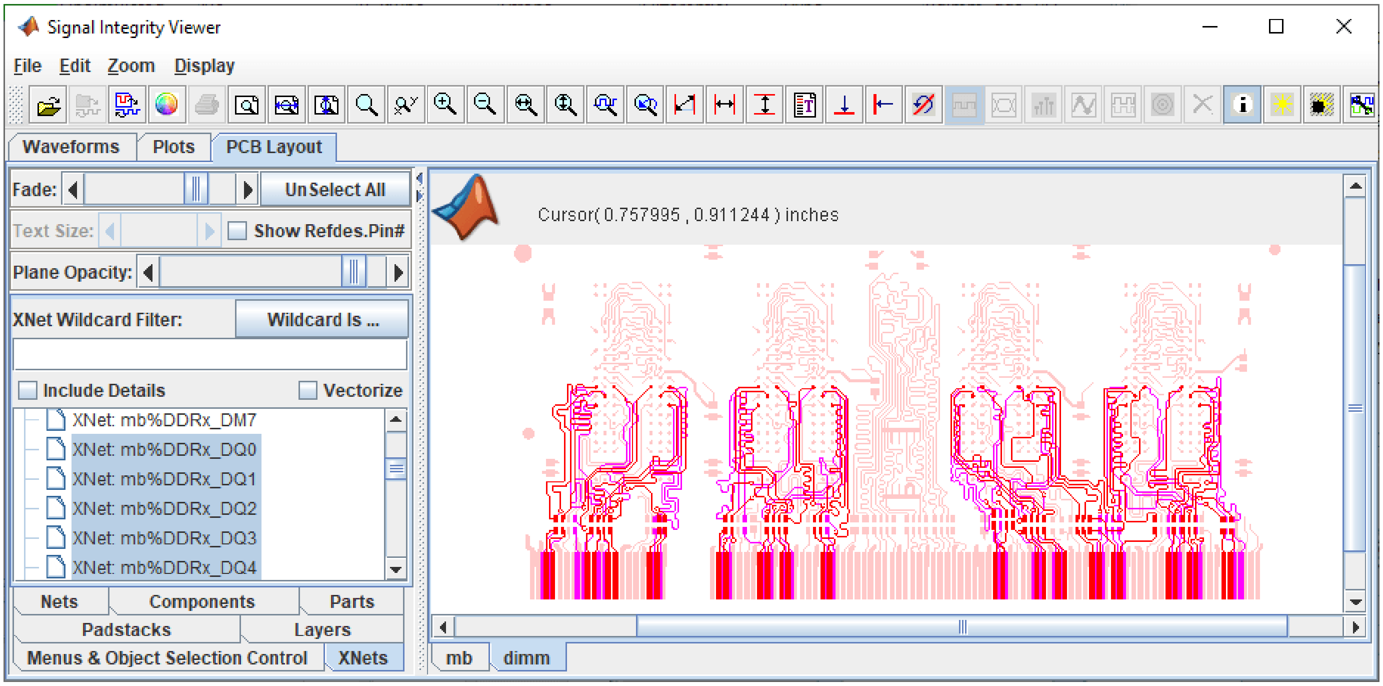
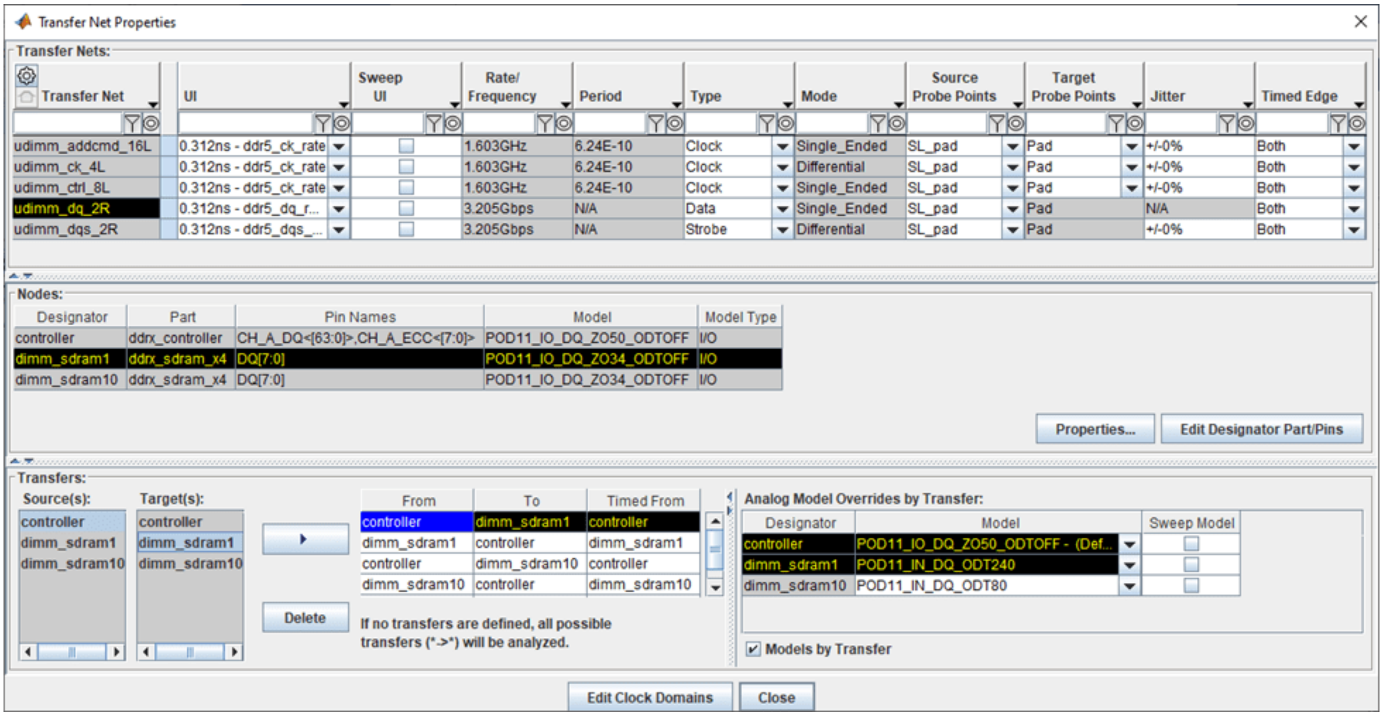


Figure: Signal Integrity Viewer opens with all selected Transfer Nets highlighted.

Configure Transfer Net Properties

In this project, the nets from the boards have been automatically matched with the Transfer Nets that have already been set up in pre-layout. This allows the post-layout simulations to make use of the bit time, bus transaction definition and model overrides (for on-die termination) that were part of the Transfer Nets in the EB1 original project. You can configure the **Transfer Net Properties** to define Transfers between a DDRx controller and DIMM or DRAM. You can also set clock or symbol UI, drive strength (ODS) and on-die termination (ODT).



Transfer Net Properties

Transfer Nets:

Transfer Net	UI	Sweep UI	Rate/ Frequency	Period	Type	Mode	Source Probe Points	Target Probe Points	Jitter	Timed Edge
udimm_addcmd_16L	0.312ns - ddr5_ck_rate	□	1.603GHz	6.24E-10	Clock	Single_Ended	SL_pad	Pad	+/-0%	Both
udimm_ck_4L	0.312ns - ddr5_ck_rate	□	1.603GHz	6.24E-10	Clock	Differential	SL_pad	Pad	+/-0%	Both
udimm_ctrl_8L	0.312ns - ddr5_ck_rate	□	1.603GHz	6.24E-10	Clock	Single_Ended	SL_pad	Pad	+/-0%	Both
udimm_dq_2R	0.312ns - ddr5_dq_r...	□	3.205Gbps	N/A	Data	Single_Ended	SL_pad	Pad	N/A	Both
udimm_dqs_2R	0.312ns - ddr5_dqs_...	□	3.205Gbps	N/A	Strobe	Differential	SL_pad	Pad	+/-0%	Both

Nodes:

Designator	Part	Pin Names	Model	Model Type
controller	ddrx_controller	CH_A_DQ<[63:0]>,CH_A_ECC<[7:0]>	POD11_IO_DQ_ZO50_ODTOFF	I/O
dimm_sdram1	ddrx_sdram_x4	DQ[7:0]	POD11_IO_DQ_ZO34_ODTOFF	I/O
dimm_sdram10	ddrx_sdram_x4	DQ[7:0]	POD11_IO_DQ_ZO34_ODTOFF	I/O

Transfers:

Source(s): controller, dimm_sdram1, dimm_sdram10

Target(s): controller, dimm_sdram1, dimm_sdram10

From	To	Timed From
controller	dimm_sdram1	controller
dimm_sdram1	controller	dimm_sdram1
controller	dimm_sdram10	controller
dimm_sdram10	controller	dimm_sdram10

If no transfers are defined, all possible transfers (*->*) will be analyzed.

Analog Model Overrides by Transfer:

Designator	Model	Sweep Model
controller	POD11_IO_DQ_ZO50_ODTOFF - (Def...	□
dimm_sdram1	POD11_IN_DQ_ODT240	□
dimm_sdram10	POD11_IN_DQ_ODT80	□

☑ Models by Transfer

Figure: Transfer Net Properties Tab after Assignment. You can configure Transfers and Analog Model Overrides by Transfer (e.g. configure IBIS model ODS and ODT)

Note: Configuring the **Transfers** pane is explained with detail in the User Guide, and is beyond the scope of this example.

The nets are now ready to simulate. By matching the nets on the board with the Transfer Nets Parallel Link Designer has reduced the post-layout task to importing and setting up the boards in the system. Parallel Link Designer will automatically extract the actual routed topologies, simulate and analyze each net according to its respective Transfer Net properties, normalize and measure interconnect flight times and compute timing margins for the appropriate transactions.

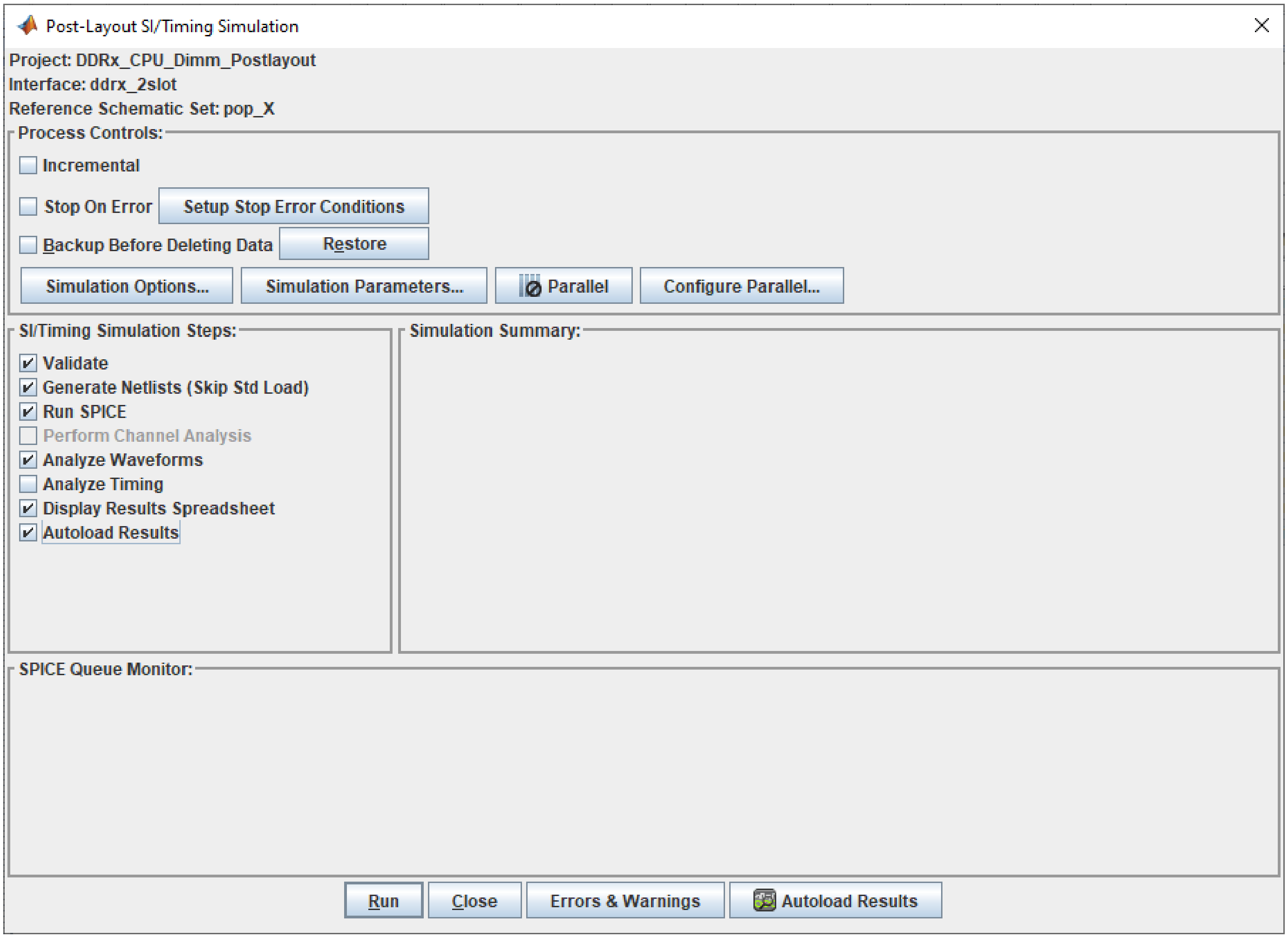
Before selecting the nets to simulate, hide the nets that are from the serial interface or are used for multi-rank DIMMs by checking the Hide if Simulate=N/A checkbox (see Figure).

Run Post-Layout Simulations

By default all nets are excluded from simulation in post-layout. To select all of the nets and include them for simulation select all of the table rows and click the Include button in the Extended Net Simulation area.

The Base Spice Simulation Count field in the lower right shows the number of simulations. This will change depending on the number of Transfer Nets, Model Sweeps, and Corner Conditions selected for your simulation.

Click the **Simulate** toolbar button: or select **Run | Simulate** from the menus to launch the **PostLayout SI/Timing Simulation** dialog. If you are asked to save changes click Yes. Select all checkboxes applicable to the type of DDR in your design (e.g. **Analyze Timing** does not apply to DDR5 but does apply to earlier versions such as DDR3 and remains available as an option for diagnostic purposes).



Post-Layout SI/Timing Simulation

Project: DDRx_CPU_Dimm_Postlayout
Interface: ddrx_2slot
Reference Schematic Set: pop_X

Process Controls:

□ Incremental
□ Stop On Error
□ Backup Before Deleting Data

SI/Timing Simulation Steps:

☑ Validate
☑ Generate Netlists (Skip Std Load)
☑ Run SPICE
□ Perform Channel Analysis
☑ Analyze Waveforms
□ Analyze Timing
☑ Display Results Spreadsheet
☑ Autoload Results

Simulation Summary:

SPICE Queue Monitor:

Click the **Run** button to start the simulations.

Note: The simulations may take several hours to finish.

When the simulations have completed the spreadsheet report will launch. As in pre-layout there are waveform tabs and timing tabs. To interpret the results, see “Results of Pre-Layout Analysis in Parallel Link” on page 7-8.

See Also

More About

- “Results of Pre-Layout Analysis in Parallel Link” on page 7-8
- “DDR5 Implementation Kit” on page 11-71
- “GDDR5 x32 Implementation Kit” on page 11-73
- “Low-Power DDR5 Architectural Kit” on page 11-80

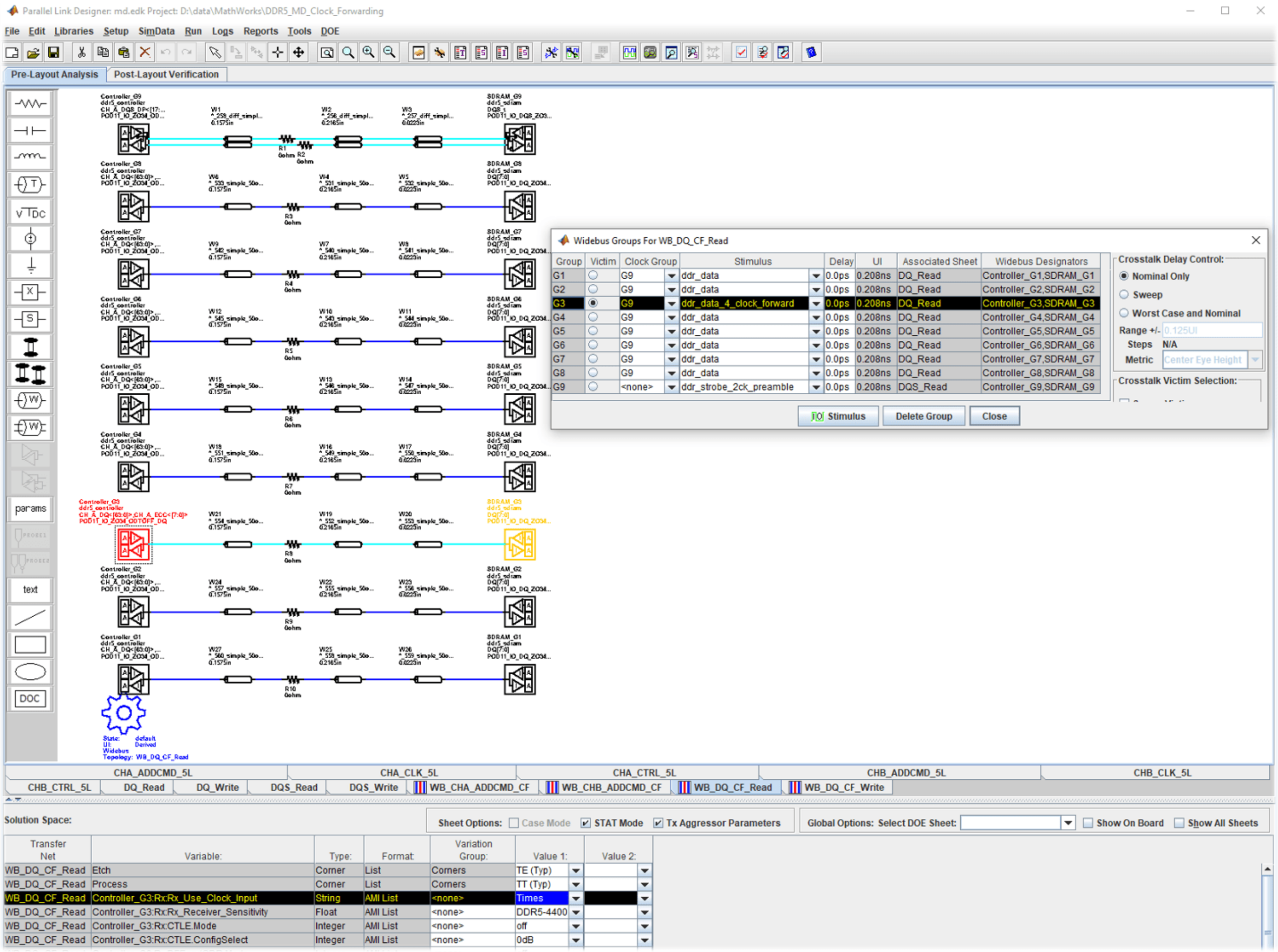
DDR5 IBIS-AMI with Clock Forwarding

This example shows how to use Signal Integrity Toolbox™ for MATLAB to analyze a DDR5 interface with the IBIS-AMI feature Clock-Forwarding enabled for analysis of system margins. IBIS BIRD 204, "*DQ_DQS GetWave Flow for Clock Forwarding Modeling,*" adds the ability to pass in an external Clock signal (or Strobe, as appropriate) to an Address (or Data) IBIS-AMI receiver `GetWave` model using the `clock_times` pointer defined by the IBIS specification. A new AMI Reserver Parameter, `Rx_Use_Clock_Input`, is used to enable this functionality.

Open DDR5 Memory-Down Clock Forwarding Kit

Open the DDR5 Memory-Down Clock Forwarding kit in the **Parallel Link Designer** app using the `openSignalIntegrityKit` function.of first code block

```
openSignalIntegrityKit("DDR5_MD_Clock_Forwarding")
```



Widebus Groups For WB_DQ_CF_Read

Group	Victim	Clock Group	Stimulus	Delay	UI	Associated Sheet	Widebus Designators
G1	○	G9	ddr_data	0.0ps	0.208ns	DQ_Read	Controller_G1,SDRAM_G1
G2	○	G9	ddr_data	0.0ps	0.208ns	DQ_Read	Controller_G2,SDRAM_G2
G3	●	G9	ddr_data_4_clock_forward	0.0ps	0.208ns	DQ_Read	Controller_G3,SDRAM_G3
G4	○	G9	ddr_data	0.0ps	0.208ns	DQ_Read	Controller_G4,SDRAM_G4
G5	○	G9	ddr_data	0.0ps	0.208ns	DQ_Read	Controller_G5,SDRAM_G5
G6	○	G9	ddr_data	0.0ps	0.208ns	DQ_Read	Controller_G6,SDRAM_G6
G7	○	G9	ddr_data	0.0ps	0.208ns	DQ_Read	Controller_G7,SDRAM_G7
G8	○	G9	ddr_data	0.0ps	0.208ns	DQ_Read	Controller_G8,SDRAM_G8
G9	○	<none>	ddr_strobe_2ck_preamble	0.0ps	0.208ns	DQS_Read	Controller_G9,SDRAM_G9

Crosstalk Delay Control:
● Nominal Only
○ Sweep
○ Worst Case and Nominal
Range +/- 0.125UI
Steps N/A
Metric Center Eye Height
Crosstalk Victim Selection:

Stimulus | Delete Group | Close

Solution Space:

Sheet Options: □ Case Mode ☑ STAT Mode ☑ Tx Aggressor Parameters

Global Options: Select DOE Sheet: □ Show On Board □ Show All Sheets

Transfer Net	Variable:	Type:	Format:	Variation Group:	Value 1:	Value 2:
WB_DQ_CF_Read	Etch	Corner	List	Corners	TE (Typ)	
WB_DQ_CF_Read	Process	Corner	List	Corners	TT (Typ)	
WB_DQ_CF_Read	Controller_G3:Rx:Rx_Use_Clock_Input	String	AMI List	<none>	Times	
WB_DQ_CF_Read	Controller_G3:Rx:Rx_Receiver_Sensitivity	Float	AMI List	<none>	DDR5-4400	
WB_DQ_CF_Read	Controller_G3:Rx:CTLE.Mode	Integer	AMI List	<none>	off	
WB_DQ_CF_Read	Controller_G3:Rx:CTLE.ConfigSelect	Integer	AMI List	<none>	0dB	

Kit Overview

- Project name: DDR5_MD_Clock_Forwarding

- Interface name: md
- Target data rate: DDR5-3200 to DDR5-4800
- Widebus sheet for each of two channels of Address/Command (5 DRAMs per lane with ODT selectable)
- Widebus sheet for DQ Read transactions (8 DQ plus 1 DQS in Widebus Group with ODT selectable)
- Widebus sheet for DQ Write transactions (8 DQ plus 1 DQS in Widebus Group with ODT selectable)
- Note: Widebus sheets are required for Clock-Forwarding analysis.

Typical DDR5 coupled channel simulation setup using clock-forwarding

The clock times or waveform generated by DQS is passed to DQ[7:0] using the DQ DLL clock_times pointer. The DQ DLL then operates on these clock times as desired (for example triggering DFE taps, modelling the DQS delay tree or centering the DQ on the DQS waveform) and then passes out the same or modified clock_times.

DDRx Timing and Waveform Mask Analysis

This example demonstrates the use of **Parallel Link Designer** in the Signal Integrity Toolbox™ to perform Timing Analysis using "Setup and Hold" margins or by Waveform "Eye-Diagram" Compliance Masks for the DDR (Double Data Rate) family of memory interface protocols. An example kit where both Timing and Waveform Mask analysis would be DDR4 (Double Data Rate) Memory Down. Specifically for DDR4 systems, Timing Analysis is performed on Clock-Address/Command interface, and Waveform Compliance Masks are applied to Strobe-Data(DQS-DQ) interface. Other DDR types may use only Timing Analysis, or only Waveform Mask Analysis or a blend depending on each individual standard.

Overview

Timing and Waveform Mask Analysis are used in Signal Integrity analysis of a parallel-link system. This tutorial shows how **Parallel Link Designer** can be used to analyze a DDR memory interface using a DDR4 implementation kit as the starting point. This example assumes you are referencing the "DDR4_MD" kit, which will allow you to begin Signal Integrity analysis immediately since all models and schematic sheets are part of the kit. You will also see how to import a new IBIS (.ibs) file for an example SDRAM and see how it references Timing (.tmg) and Include (.inc) files for margin analysis by **Parallel Link Designer**. If an implementation kit were not available for this interface, you would need to build the simulation environment (Parts, IBIS Models, and pre-layout topologies) from a new project. After showing you how to import a new IBIS SDRAM part, this example will then illustrate some of the steps involved in setting up Timing and Include files to enable Timing and Waveform Mask Analysis.

Getting Started

You will see how to import a new IBIS file to an existing DDR4 kit. Signal Integrity Toolbox kits are an easy way to jumpstart your analysis by providing an interface that is completely configured and ready to simulate. These kits can then be customized by adding models and modifying parameter values for your specific implementation. For example, the following documentation will show you how to import a new IBIS file for an SDRAM, then modify a Timing (.tmg) file and Include (.inc) file to enable Timing and Waveform Mask analysis in the **Parallel Link Designer** app. This example will begin with a kit already configured with details such as clock domain, transfer net configuration, and sheets classified as "Data, Clock, Strobe" as appropriate for a DDR4 interface.

Begin with the DDR4 Memory-Down Kit

Open the DDR4_MD kit in the **Parallel Link Designer** app using the `openSignalIntegrityKit` function.

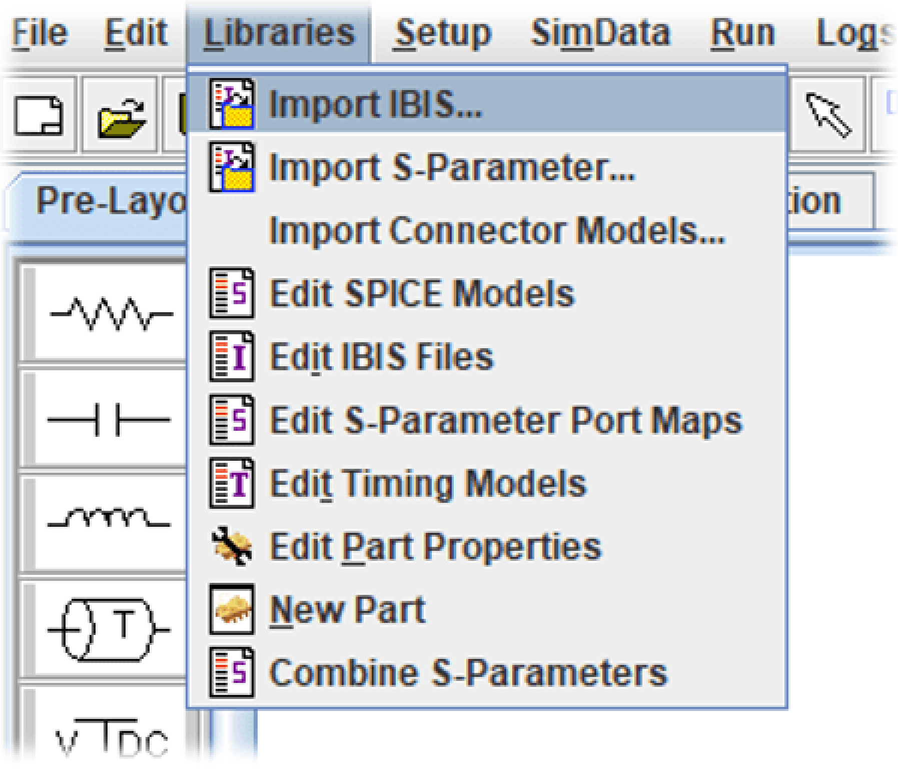
```
openSignalIntegrityKit("DDR4_MD");
```

Then you can download the attached file "ex_sdram_x6.zip" and unarchive to a temporary folder. The contents of this archive are as follows:

- IBIS file representing an imagined "x6 SDRAM" for the purposes of this example
- Include files to configure logic thresholds and eye masks for Waveform Analysis
- A Timing file to place in your project folder "DDR4_MD\si_lib\timing"

Import IBIS File and Reconfigure Sheets

You can import the IBIS file "ex_sdram_x6.ibs" by clicking on the "Libraries->Import IBIS..." menu from the toolbar. You can see in your OS file exlporer that the IBIS (.ibs) file is placed in the folder "DDR4_MD\si_lib\ibis" along with Include (.inc) files called within the IBIS file.



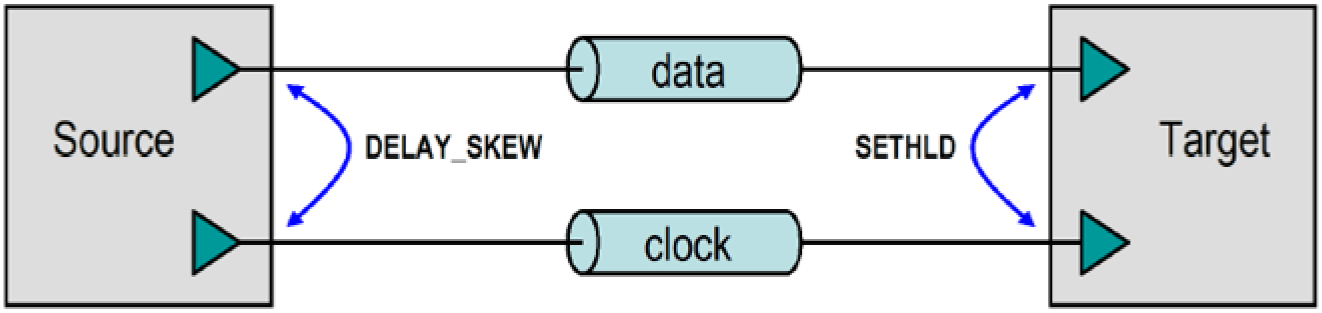
Next, place the file "ex_sdram_x6.tmg" in your project folder "DDR4_MD\si_lib\timing" where this is the Timing file used by **Parallel Link Designer** to simulate with the IBIS file "ex_sdram_x6.ibs." The contents of the Timing file as well as the Include files will be explained in the following sections.

Timing Analysis Configuration

This section describes the Transfer Net and timing model requirements for various types of timing.

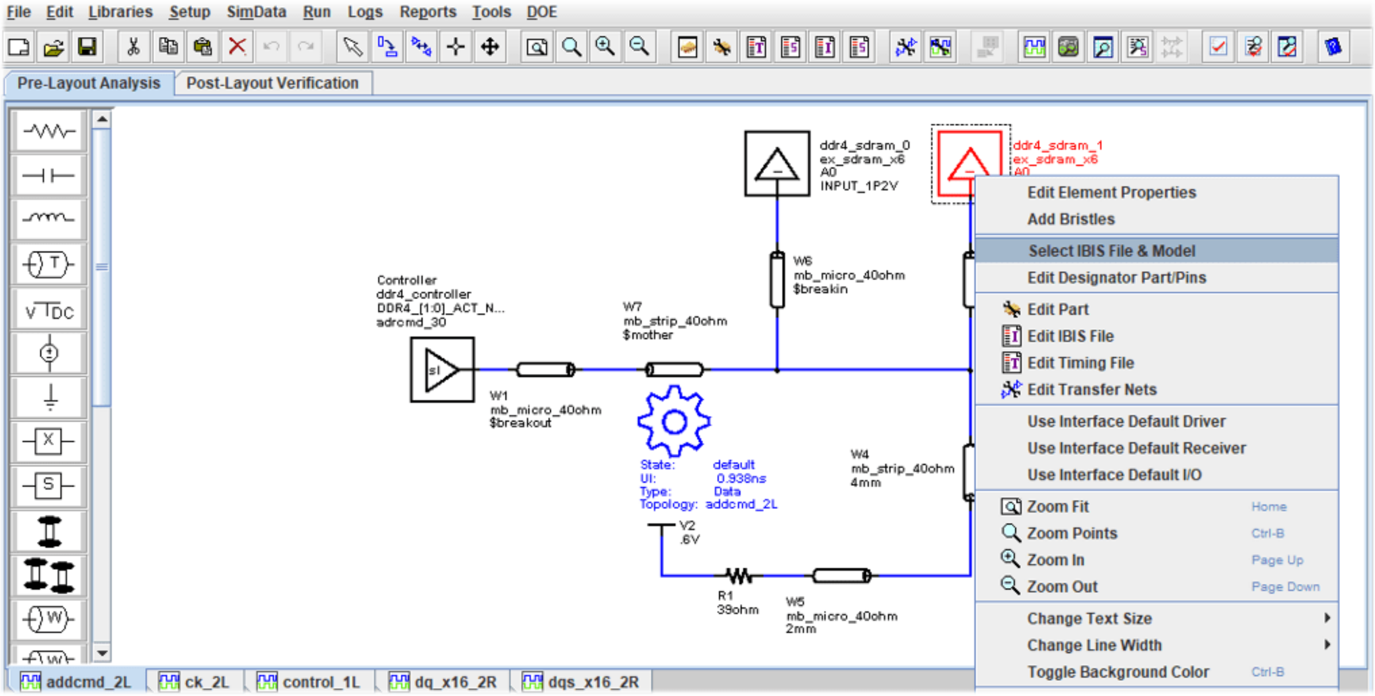
Source-Synchronous Timing

You can see in the figure below an illustration of Source-Synchronous data transfer timing elements that could represent, for this kit example, a DDR memory interface. The figure shows a data net with its associated clock net and is used to describe requirements to configure a Source-Synchronous interface for timing analysis.

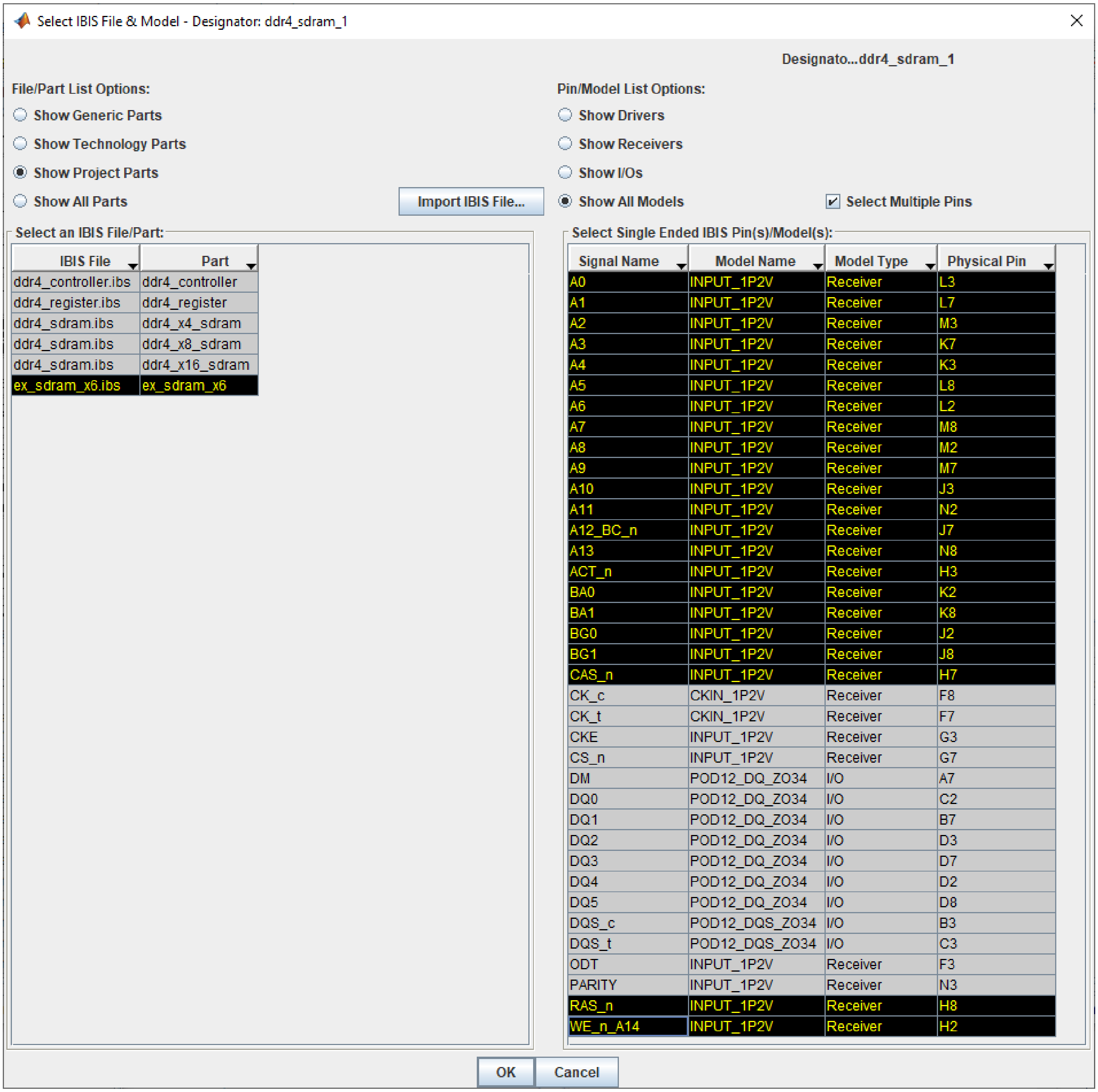


Source-Synchronous analysis has the following configuration requirements:

You must create or configure the appropriate Transfer Net schematic sheets. In the simplest case, there must be two schematic sheets, one for the data net and one for the clock net. The data schematic sheet should be of type Data and the clock schematic sheet should be of type Clock or Strobe. Because you have a kit already configured with these sheets, you can do this in the kit by clicking on the SDRAM parts on each sheet and changing the IBIS File to "ex_sdram_x6.ibs"



Then you can select the Address/Command pins for the "ex_sdram_x6.ibs" model:



Select IBIS File & Model - Designator: ddr4_sdram_1

Designato...ddr4_sdram_1

File/Part List Options:

○ Show Generic Parts
○ Show Technology Parts
● Show Project Parts
○ Show All Parts

Pin/Model List Options:

○ Show Drivers
○ Show Receivers
○ Show I/Os
● Show All Models

☑ Select Multiple Pins

Select an IBIS File/Part:

IBIS File	Part
ddr4_controller.ibs	ddr4_controller
ddr4_register.ibs	ddr4_register
ddr4_sdram.ibs	ddr4_x4_sdram
ddr4_sdram.ibs	ddr4_x8_sdram
ddr4_sdram.ibs	ddr4_x16_sdram
ex_sdram_x6.ibs	ex_sdram_x6

Select Single Ended IBIS Pin(s)/Model(s):

Signal Name	Model Name	Model Type	Physical Pin
A0	INPUT_1P2V	Receiver	L3
A1	INPUT_1P2V	Receiver	L7
A2	INPUT_1P2V	Receiver	M3
A3	INPUT_1P2V	Receiver	K7
A4	INPUT_1P2V	Receiver	K3
A5	INPUT_1P2V	Receiver	L8
A6	INPUT_1P2V	Receiver	L2
A7	INPUT_1P2V	Receiver	M8
A8	INPUT_1P2V	Receiver	M2
A9	INPUT_1P2V	Receiver	M7
A10	INPUT_1P2V	Receiver	J3
A11	INPUT_1P2V	Receiver	N2
A12_BC_n	INPUT_1P2V	Receiver	J7
A13	INPUT_1P2V	Receiver	N8
ACT_n	INPUT_1P2V	Receiver	H3
BA0	INPUT_1P2V	Receiver	K2
BA1	INPUT_1P2V	Receiver	K8
BG0	INPUT_1P2V	Receiver	J2
BG1	INPUT_1P2V	Receiver	J8
CAS_n	INPUT_1P2V	Receiver	H7
CK_c	CKIN_1P2V	Receiver	F8
CK_t	CKIN_1P2V	Receiver	F7
CKE	INPUT_1P2V	Receiver	G3
CS_n	INPUT_1P2V	Receiver	G7
DM	POD12_DQ_ZO34	I/O	A7
DQ0	POD12_DQ_ZO34	I/O	C2
DQ1	POD12_DQ_ZO34	I/O	B7
DQ2	POD12_DQ_ZO34	I/O	D3
DQ3	POD12_DQ_ZO34	I/O	D7
DQ4	POD12_DQ_ZO34	I/O	D2
DQ5	POD12_DQ_ZO34	I/O	D8
DQS_c	POD12_DQS_ZO34	I/O	B3
DQS_t	POD12_DQS_ZO34	I/O	C3
ODT	INPUT_1P2V	Receiver	F3
PARITY	INPUT_1P2V	Receiver	N3
RAS_n	INPUT_1P2V	Receiver	H8
WE_n_A14	INPUT_1P2V	Receiver	H2

The Project Parts must have Timing Models. Each timing model must have a SETHLD (Setup and Hold) timing constraint defined between the data and clock PINDEF groups. You must also add a DELAY_SKEW statement in the timing model for the driving chip between the data and clock PINDEF groups. However, for this example a Timing file has been provided, and will be discussed in more detail below.

- **Note:** You should already have placed the file "ex_sdram_x6.tmg" in your project folder ("DDR4_MD\si_lib\timing").

Configure Include Files for Waveform Analysis:

This example will next show you where the thresholds for Waveform Analysis are configured in the Include file, and where the Compliance Mask is defined as well.

Note: Within IBIS, Timing, and Include files, you may notice the following keyword with an argument following it:

```
|MathWorks <argument> <parameter>...
```

This would normally be considered an IBIS comment line (due to the pipe character, "|") but is in fact is a keyword to flag the parser within **Parallel Link Designer** to configure a parameter specified by the argument provided. For example, logic thresholds for the DDR4 standard would be defined for an SDRAM IBIS [Model] for Address/Command as follows in the Include file "ex_add_ctrl_sstl_12.inc." These are explained in more detail in the User Guide section 12.6: "*Waveform Analysis Parameters.*"

Parameters in the Include file for Waveform Analysis:

```
| TYP MIN MAX

|| Based on VDDQ: 1.2 1.14 1.26

|MathWorks Overshoot_High 1.5 1.44 1.56

|MathWorks AC_Overshoot_High 1.2 1.14 1.26

|MathWorks Vin_AC_High 0.700 0.670 0.730

|MathWorks Vin_Meas_R_High 0.700 0.670 0.730

|MathWorks Vin_Meas_F_High 0.675 0.645 0.705

|MathWorks Vin_DC_High 0.675 0.645 0.705

|MathWorks Vin_DC_Low 0.525 0.495 0.555

|MathWorks Vin_Meas_R_Low 0.525 0.495 0.555

|MathWorks Vin_Meas_F_Low 0.500 0.470 0.530

|MathWorks Vin_AC_Low 0.500 0.470 0.530

|MathWorks AC_Overshoot_Low 0.0 0.0 0.0

|MathWorks Overshoot_Low -0.3 -0.3 -0.3

|MathWorks Slew_Time_Min 25ps

|MathWorks Slew_Time_Max 175ps

|

||Standard load measurement level

|MathWorks Vmeas 0.600 0.570 0.630
```

```
|

||Standard load termination value

|MathWorks Vref 1.2 1.14 1.26
```

There are a few other parameters you will want to enable Eye Mask Compliance for DDR4 or DDR5 or as appropriate for your interface. For example, in the DATA class DQ Include file you would set the following:

```
||| DDR4/5 specific timing parameters:

|

|| Enable eye mask processing on receivers:

|| YES = DDR Eye Mask analysis done.

|| No (Default) = No DDR4/5 analysis done.

|MathWorks MaskAnalysis YES

|

|| Which Vref to use for analysis on receivers:

|| YES = Use IBIS levels relative to Floating Vref

|| No (Default) = Use fixed Vref as defined by Vin_vref

|MathWorks FloatingVref YES

|

|| Granularity for Mask Analysis

|| YES = Do Mask Analysis Per Pin

|| No (Default) = Do Mask Analysis Per Bundle

|MathWorks MaskPerPin Yes

|

|| Which Thresholds to use for Timing Analysis

|| YES = Use Vref +/- (ViVW_Total/2)

|| No (Default) = Use IBIS Model thresholds

|MathWorks TimingViVW Yes
```

Configure Include Files for Waveform Mask Analysis:

Parameters in the Include file to configure Eye Mask analysis:

Here you can see a plot of the mask defined in the DATA class DQ Include file you would set the following at the section for a clock period of 0.938:

Note: The parameters ViVW_Total, TiVW_Total, and TiVW_Output_Skew are defined in the DDR4 or DDR5 standard (JESD79-4 and JESD79-5, respectfully), please refer to those as appropriate for their definitions and more details about them.

```
|MathWorks elseif ( $clock_domain(ddr4_ck_period)==0.938)
```

Where the following statements configure the parameters AC_Overshoot_High_Area and Low_Area for this data rate:

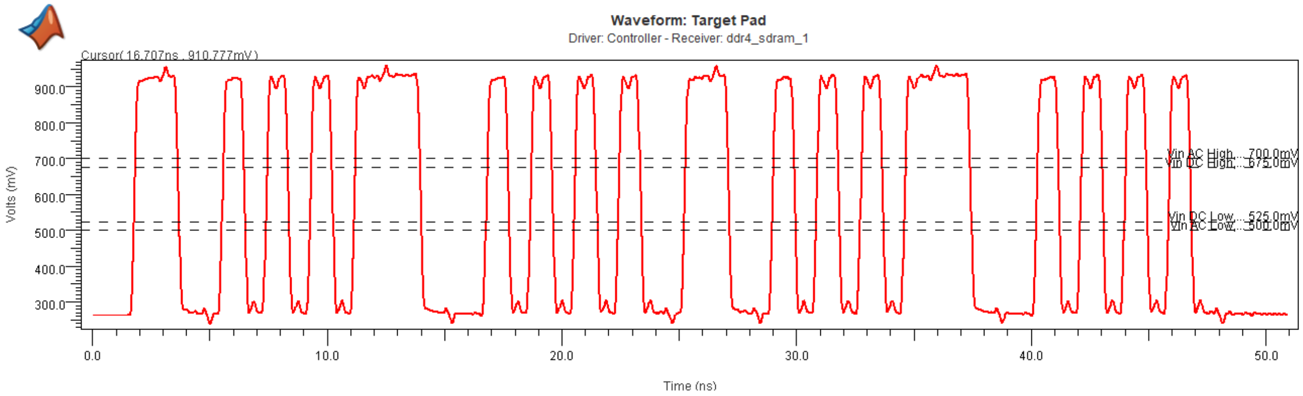
```
|| PLD Waveform Quality and Timing Levels for DDR4-2133

|MathWorks AC_Overshoot_High_Area 0.0901 0.0901 0.0901

|MathWorks AC_Overshoot_Low_Area 0.0788 0.0788 0.0788
```

And where the following configure the Mask for Eye-Diagram compliance and margin calculations for this data rate:

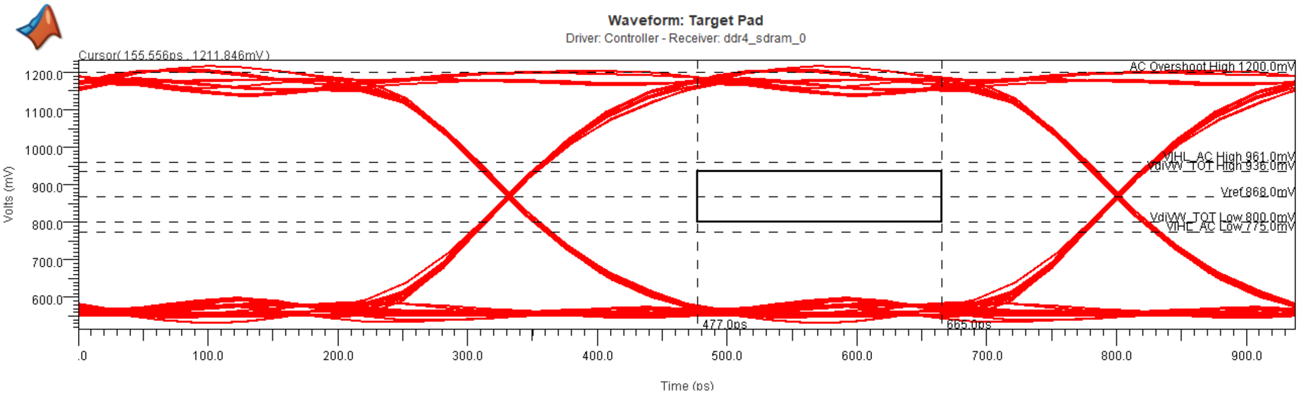
```
|MathWorks TiVW_Total 0.094ns 0.094ns 0.094ns

|MathWorks TiPW 0.272ns 0.272ns 0.272ns

|| Set SDRAM output skew to 0.4UI (DQ_SKEW_MAX - DQ_SKEW_MIN)

|MathWorks TiVW_Output_Skew 0.188ns

|MathWorks SRIN_diVW_Min 1.0

|MathWorks SRIN_diVW_Max 9.0
```

Run Simulation and View Results

After you run the simulation for Clock, Address/Command, Strobe and Data signals, you can evaluate margins for this DDR4 interface using the imagined "x6" SDRAM model deck. You can do this in Signal Integrity Viewer, which has many plot functions available to assist you in analyzing Waveforms and Eye Masks. You can see the result by plotting a Waveform for an Address-Command transaction where the Receiver is "ddr4_sdram_0," then find the appropriate row in the results table of the Signal Integrity Viewer, right-click to open the **Show Waveform** dialog and select **At Target Pad**.



You can see the result by plotting an Eye Diagram for a Data Write transaction where the Receiver is "ddr4_sdram_0," you can find the appropriate row in the results table of the Signal Integrity Viewer, right-click to open the **Show Waveform** dialog and select **At Target Pad**.



Configure Timing Files for Timing Analysis

PARAMS: Section of Timing File

In the Timing file, you can see there are multiple sections. The first section is the `PARAMS:` section, where you can see timing margins are defined for DDR4-2133 (clock period 0.938, DQ bit time 0.469).

```
PARAMS:

<...>

# DDR4-2133

elseif (DQ_BIT_TIME==0.469)

ADDCMD_SETUP = 0.080

ADDCMD_HOLD = 0.105

CTRL_SETUP = 0.080

CTRL_HOLD = 0.105

tQH = 0.76 * DQ_BIT_TIME

tDQSQ = 0.16 * DQ_BIT_TIME

tDQSS_min = -0.27 * CK_PERIOD

tDQSS_max = 0.27 * CK_PERIOD

tDSS = 0.18 * CK_PERIOD # met through write leveling

tDSH = 0.18 * CK_PERIOD # met through write leveling

<...>
```

The next portion of the `PARAMS:` section is used to setup DQ to DQS Input Timing. DQ setup and hold are obsolete parameters in DDR4/5 parlance. The Data signals (DQ) have to meet a mask whose conformation is reported in the Waveform and Timing Report. The Strobe signals (DQS) are assumed to be capable of capturing a DQ bundle (which can be a x4 "Nibble," x8 "Byte," the imagined "x6" SDRAM lane in this example, or any number) that meets the mask.

Note: To assist you with diagnosing any potential Timing Margin issues, this file is configured to find DQ Setup and Hold values. They are derived from the mask parameters. This helps evaluate Timing Analysis, and for verification that DQS training can center DQS on the DQ bundle, accounting for some level of jitter on the DQS line. Per preliminary SDRAM datasheets, this value is `TdiVW_total/2`.

```
<PARAMS: continued>

DQ_SETUP = (0.2 * DQ_BIT_TIME)/2

DQ_HOLD = (0.2 * DQ_BIT_TIME)/2

## DQ to DQS Output timing:

#

DQ_SKEW_MIN = tQH - DQ_BIT_TIME

DQ_SKEW_MAX = tDQSQ

## DQS to CK input timing:

# Assumes that at the controller, timing is defined for

# Rising CK to both Rising and Falling edges of DQS

DQS_R_SETUP = -tDQSS_max

DQS_R_HOLD = tDQSS_min + DQ_BIT_TIME

DQS_F_SETUP = tDSS

DQS_F_HOLD = tDSH - DQ_BIT_TIME

END_PARAMS
```

PINDEF: Section of Timing File

The next section of a Timing file `PINDEF:` declares the pin types as Inputs, Outputs, Bi-Directional, and Uninteresting. Signals listed in the Uninteresting section will have no Timing analysis performed, but any miscellaneous signals can be included for completeness and to eliminate coverage warnings in Validation. The pin names can be grouped by bus, or common timing function.

```
PINDEF: 37 PINS

# Syntax:

# <Timing Group Name> = <IBIS Model pin names>

INPUTS
```

```
## All input signals are listed in this section.
## Example:
# CLK = CLOCK_IN
# DATA_I = DATA_IN<31:0>
# CTRL_I = ADDRESS<7:0>, WRITE_EN, RAS, CAS
ADDCMD = A[11:0], A12_BC_n, A13, WE_n_A14, CAS_n, RAS_n, ACT_n, PARITY, BA[1:0], BG[1:0]
CK = CK_t, CK_c
CTRL = CKE, CS_n, ODT
#
OUTPUTS
## All output signals are listed in this section.
## Example:
# DATA_O = DATA_OUT<31:0>
# REFCLK_1 = CLOCK_OUT1, CLOCK_OUT3
# REFCLK_2 = CLOCK_OUT2, CLOCK_OUT4

BIDIR
## All bi-directional signals are listed in this section.
## Example:
# DATA_IO = D_A_<3:0>, D_B_<7:4>, D_C_<11:8>, D_D_<15:12>
DQ = DQ[5:0]
DQS = DQS_t, DQS_c
#
UNINTERESTING
## All other signals are listed in this section.
## Example:
# MISC = AUD_BITCLK_A_H,AUD_SYNC_A_H,EXT_<11:0>_H,USB_PRTPWR_<2:0>_L,\
# USB_VD_<5:0>_N,USB_VD_<5:0>_P
```

```
DM = DM

#

END_PINDEF
```

Timing Configuration Section of File

There are different sections of the file used to configure Timing based on Input, Output, and other means depending upon the system interface being analyzed in **Parallel Link Designer**:

- Input Timing relationships (Setup and Hold)
- Output Timing: Synchronous Output Delays
- Output Timing: Dynamic Clock Skew Output Delays
- Output Timing: Source-Synchronous Output Delays
- Input Timing: DQ Read Timing

Input timing relationships: (Setup & Hold)

All input pin timing constraints are defined using the `'SETHLD'` keyword. Note that setup and hold constraints can be defined using 1, 2, or 4 values giving the user flexibility to define timing constraints at the required granularity. In the examples below, setup and hold constraints are defined for all pins in the "Data Timing Group" relative to the defined "Edge" (rising or falling) of all pins in the "Clock Timing Group."

Syntax:

```
<Keyword> <Data Timing Group> *TO <Edge> <Clock Timing Group> <Setup/Hold
Time>

<Keyword> <Data Timing Group> *TO <Edge> <Clock Timing Group> <Setup Time>
<Hold Time>

<Keyword> <Data Timing Group> *TO <Edge> <Clock Timing Group> <R Setup Time>
<F Setup Time> <R Hold Time> <F Hold Time>
```

Examples:

```
SETHLD CTRL_I *TO R CLK CTRL_IN

SETHLD DATA_I *TO F CLK DATA_SETUP DATA_HOLD

SETHLD DATA_IO *TO R CLK DATA_R_SETUP DATA_F_SETUP DATA_R_HOLD DATA_F_HOLD
```

Output Timing: Synchronous Output Delays

The `DELAY`, `DELAY_CORRELATED`, and `DELAY_ANTICORRELATED` keywords are used to define synchronous output timing relationships between the data and clock timing groups defined in the `PINDEF` section above. To define output timing relationships on both rising and falling edges, multiple delay statements are required. Note that Clock-to-out (Tco) delay values can be defined using 2 or 4 values, giving the user flexibility to define output timing relationships at the required granularity. In the examples below, output timing relationships are defined for all pins in the "Data Timing Group" relative to the defined "Edge" (rising or falling) of all pins in the "Clock Timing Group".

Syntax:

```
<Keyword> <Edge> <Clock Timing Group> *TO <Data Timing Group> <Tco Min> <Tco
Max>
```

```
<Keyword> <Edge> <Clock Timing Group> *TO <Data Timing Group> <Tco Rmin> <Tco
Rmax> <Tco Fmin> <Tco Fmax>
```

Output Timing: Correlative Analysis for Synchronous Output:

When the `'DELAY'` keyword is used, synchronous timing is done using both the 'Tco Min' and 'Tco Max' delay values for each process corner (FF and SS). This is a worst-case analysis. When the `'DELAY_C'` or `'DELAY_CORRELATED'` keyword is used, synchronous timing is done using the 'Tco Min' value for the FF process corner, and the 'Tco Max' delay value for the SS process corner timing calculation. This is a correlative analysis.

When the `'DELAY_A'` or `'DELAY_ANTI_CORRELATED'` keyword is used, synchronous timing is done using the 'Tco Max' value for the FF process corner, and the 'Tco Min' delay value for the SS process corner timing calculation. This is another form of correlated analysis.

Examples:

```
DELAY R CLK *TO DATA_IO 1.0 5.2
```

```
DELAY_C R CLK *TO DATA_IO 1.0 5.0 1.2 5.2
```

```
DELAY_CORRELATED R CLK *TO DATA_IO 1.0 5.0 1.2 5.2
```

Note: There are no Synchronous Output Delays in this example model because this is a DDR model, so see section "Source-Synchronous Output Delays" below.

Output timing: Dynamic Clock Skew Output Delays

The `'CLOCK_SKEW'` keyword is used to define the out-to-out skew between two output clock timing groups. To define output timing relationships on both rising and falling edges, multiple delay statements are required. This skew along with associated interconnect delays for the source and target clocks are used to determine the setup skew and hold skew used in synchronous dynamic skew clock timing analysis.

Note: For loopback clocks, the two timing groups can be the same.

Syntax:

```
<Keyword> <Edge> <Clock Timing Group> *TO <Clock Timing Group> <Min> <Max>
```

Example:

```
CLOCK_SKEW R REFCLK_1 *TO REFCLK_2 SKEW_MIN SKEW_MAX
```

Note: There are no Dynamic Clock Skew Output Delays in this example model because this is a DDR model, so see section "Source Synchronous Output Delays" below.

Output Timing: Source-Synchronous Output Delays

The `'DELAY_SKEW'` or `'DELAY_S'` keywords are used to define the source-synchronous timing relationships between the timing groups defined in the `PINDEF` section above. Typically, these are defined between output pins of a source synchronous bus. However, these can be defined using and input clock timing group in loop-back clocking scenarios. To analyze timing on both edges, multiple delay statements are required.

Note: The source synchronous delay values can be defined using 2, or 4 values.

Syntax:

```
<Keyword> <Edge> <Clock Timing Group> *TO <Data Timing Group> <DS_Min>
<DS_Max>
```

```
<Keyword> <Edge> <Clock Timing Group> *TO <Data Timing Group> <Rmin> <Rmax>
<Fmin> <Fmax>
```

Examples:

```
DELAY_SKEW R STROBE *TO DATA_O DS_MIN DS_MAX
```

```
DELAY_S R STROBE *TO DATA_O DS_RMIN DS_RMAX DS_FMIN DS_FMAX
```

To Configure DQ Read Timing:

```
DELAY_SKEW R DQS *TO DQ DQ_SKEW_MIN DQ_SKEW_MAX
```

```
DELAY_SKEW F DQS *TO DQ DQ_SKEW_MIN DQ_SKEW_MAX
```

Timing Report Summary

The Waveform and Timing report has multiple worksheets with timing data, and each worksheet shows timing data in a different context. There are many worksheets described in the following section "Timing Report Detailed Worksheets," but you can begin with the following summary to gain some understanding of the information available:

<u>Increasing levels of detail with each sheet:</u>

- The worksheets begin with Waveform Summary on the left, and proceed with greater detail into Timing and then Mask reporting as you click on each tab moving left-to-right through the set of worksheets.
- "Timing" worksheet: worst case result per Transfer Net (could be worst result within hundreds or thousands of simulations)
- As you open sheets to the right of this, more details are provided, for example "By Variation Details" shows each simulation case

<u>Different bus transactions are classified in worksheets:</u>

- By Driver
- By Receiver
- By Corner
- And more

<u>Terminology used in worksheets:</u>

"No AC Specs:" A net that does not have a Setup or Hold constraint for timing analysis

- A clock net would normally be reported this way
- Other signals such as DQ would expect to see a result, so this would aid in debugging your configuration

"No strobe:" A constraint that could not be calculated

- The data or clock/strobe net was not simulated
- A skew statement is missing from a timing model
- A clock or strobe net is not set to Type Clock or Strobe in the tool

Timing Report Detailed Worksheets

When simulations are run it can be automatically launched by checking the **Display Results Spreadsheet** checkbox on the simulate dialog. After simulations have been run it can be displayed from the **Reports | Waveform & Timing Report** menu item or the **View Waveform & Timing Report** toolbar button. The following is a description for each worksheet:

- Timing Report Log tab contains syntax errors in the data and a summary of the Edge Details tab summarizes each edge in each simulation.
- Timing tab rolls up the By Edge tab by combining rising and falling edges.
- By Transfers tab rolls up the By Variation tab by combining identical transfers (same driver and receiver).
- By Variation tab rolls up the By Edge Variation tab by combining rising and falling edges.
- By Variation Details tab
- By Variation Details Summary tab
- Timing Waveform Margin Details tab Summarizes both waveform and timing information across each simulation in a single tab.
- By Driver tab rolls up the By Variation tab by combining identical drivers.
- By Receiver tab rolls up the By Variation tab by combining identical receivers.
- Synchronous Details tab contains the timing data for nets using a regular (centralized) clocking configuration by receiver. It includes Setup and Hold margins and etch delays referenced to driver test load. It also lists receiver Setup and Hold time specs, driver clock to out delays, UI (bit time) and clock skews used to calculate the margins.
- Source Synchronous Details tab contains the timing data for nets using a source synchronous clock, by receiver. It includes Setup and Hold margins as well as data and clock etch delays referenced to driver test load. It also lists receiver Setup and Hold time specs, UI (bit time) and driver skews used to calculate the margins.
- Clock Buffer Details Tab -- Shows the rollup of the DELAY NX and feedback
- Etch Delays for clock buffers.
- Clock Path Details Tab -- Gives details of the Etch Delay calculation for clock paths that traverse buffers or PLLs. This is the calculation shown in the User Guide section 13.3.2: "*Calculating Clock Delay Through Buffers and PLLs*." In the report, the columns labelled **From Buffer R/F/min/max Delay** are the rollup of the Etch Delay of the PLL output, the Etch Delay of the PLL feedback and the PLL DELAY NX. The rollup is shown for each process corner run, and then for **All**. The **All** numbers use the smallest of all corners for the min and the largest for max, so it is a cross corner worst case rollup. This corresponds to the min() and max() functions (in the equations shown in User Guide section 13.3.2: "*Calculating Clock Delay Through Buffers and PLLs*").

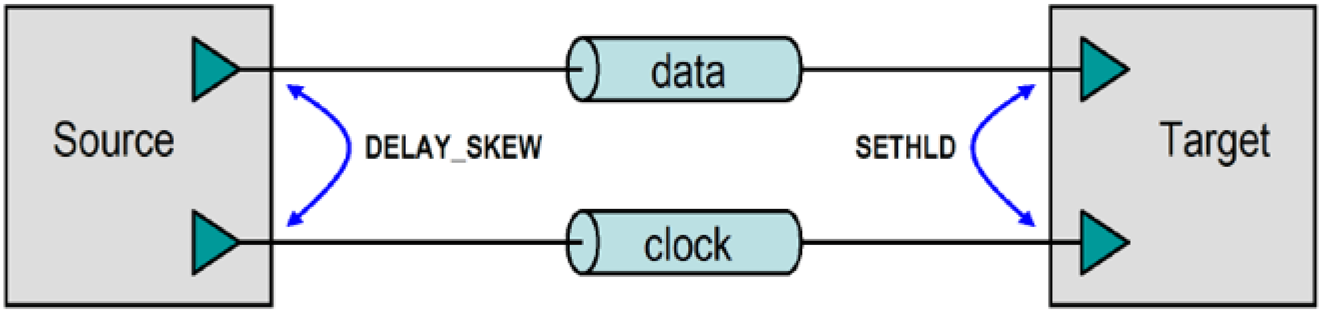
- Training Details Tab -- contains the data showing the Setup and Hold margins without training (Untrained Setup and Hold Margin), the Setup and Hold margins if the tap could be set to any value (Optimal Setup and Hold Margin) and the Setup and Hold margin when their taps are set to there best setting (Trained Setup and Hold Margin).
- The tap used for the best setting and the delay from that tap are reported as well as the Transfer Net, Pin Group (from the timing model) and the type of training statement (from the timing model).
- Edge Details tab
- Timing Waveform Margin tab
- Coupling Pushout tab (SSO mode only) reports the coupling effects on timing.
- Coupling Noise tab (SSO model only) reports the voltage variation on victim nets caused by coupling.

Support for Other Clock Architectures

Parallel Link Designer supports many other clock architectures. They are briefly described below to show you how to write the correct syntax for each clock architecture within a Timing File.

Source-Synchronous Timing

This section will re-summarize the requirements to configure a Source-Synchronous interface for timing analysis.



Source-synchronous analysis has the following configuration requirements:

1. You must create the appropriate Transfer Net schematic sheets. In the simplest case, there must be two schematic sheets, one for the data net and one for the clock net. The data schematic sheet should be of type Data and the clock schematic sheet should be of type Clock or Strobe.

2. The Project Parts must have Timing Models. Each timing model must have a `SETHLD` (Setup and Hold) timing constraint defined between the data and clock `PINDEF` groups. For example, a part with a Setup requirement of 3ns and a Hold requirement of 1ns for the `PINDEF` group data with respect to the `PINDEF` group `CLK` would have the following `SETHLD` statements, and in the event the application is a DDR interface, there must be a statement for both Rising and Falling edges:

```
SETHLD data *TO R clock 3.0 1.0
```

```
SETHLD data *TO F clock 3.0 1.0
```

3. You must add a `DELAY_SKEW` statement in the timing model for the driving chip between the data and clock `PINDEF` groups. A device that has a +/- 200ps skew between clock and data `PINDEF` groups would have the following `DELAY_SKEW` statement:

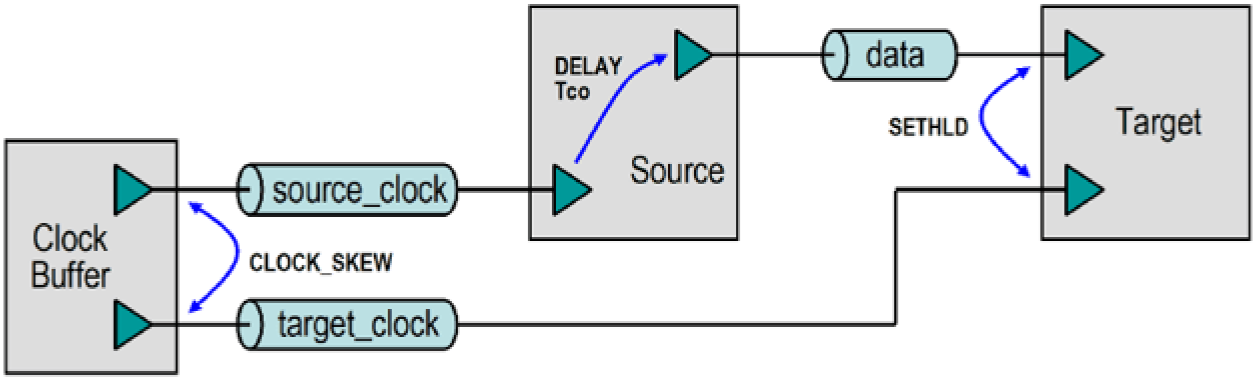
```
DELAY_SKEW R clock *TO data -0.2 0.2
```

```
DELAY_SKEW F clock *TO data -0.2 0.2
```

Note: For DDR, two DELAY_SKEW statements are also required, one for the rising clock edge and one for the falling clock edge.

Synchronous (Common-Clock) Timing Configuration

You can see in the following figure an illustration of Synchronous (Common-Clock) data transfer timing elements. This will be used to describe requirements to configure a synchronous interface for timing analysis.



Create Data Transfer Net Schematic Sheet

This schematic sheet is of type DATA. The Project Parts used for the designators must have timing models.

Define Delay Statements

The timing model for output or I/O pins must have DELAY statements defined between the data and clock PINDEF groups. For Double-Data-Rate (DDR) transfers, two DELAY statements are required, one for the rising clock edge and one for the falling clock edge, but this section will cover single-data-rate transfers.

For example, a part with a clock-to-out from 1ns (min) to 3ns (max) on the PINDEF group DATA with respect to the PINDEF group CLK would have the following DELAY statement:

```
DELAY R CLK *TO DATA 1.0 3.0
```

Define Setup and Hold Timing Constraints

The timing model for input or I/O pins must have a SETHLD (Setup and Hold) timing constraint defined between the data and clock PINDEF groups. For example, a part with a Setup requirement of 3ns and a Hold requirement of 1ns for the PINDEF group DATA with respect to the PINDEF group CLK would have the following SETHLD statement:

```
SETHLD DATA *TO R CLK 3.0 1.0
```

In order to enable the dynamic calculations of clock skew on a synchronous transfer, you will need to perform the following steps:

1. You must create the appropriate Transfer Net schematic sheets for the source and target clock (e.g., `source_clock` and `target_clock` shown in the figure above). These sheets must be of type Clock or Strobe. The clocks/strobes must both originate from the same Project Part clock buffer device and connect to the clock pins of the clocked source and target devices.

2. There must be a `CLOCK_SKEW` statement in the timing model for the clock buffer between the output clock `PINDEF` groups to define the drive uncertainty for these output pins.

Note: All output clock pins may be in the same `PINDEF` group. The `CLOCK_SKEW` statement can reference the `PINDEF` to itself. A device with clock output `PINDEF` groups `CLK1` and `CLK2` that have a skew of +/- 200ps would have the `CLOCK_SKEW` statement as shown below.

```
CLOCK_SKEW R CLK1 *TO CLK2 -0.2 0.2
```

Clock Buffers and PLLs

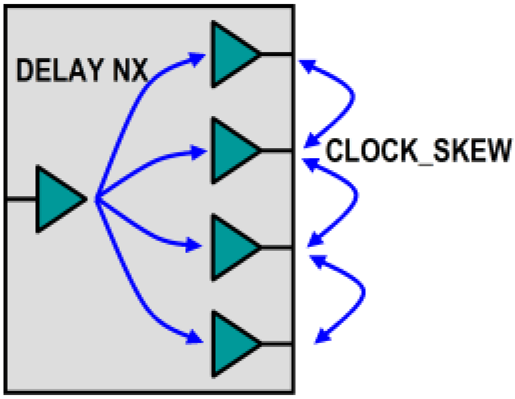
This section will briefly define some other Timing Model statements that support clock buffer and PLL-based systems.

Clock Buffer:

You can see in the illustration below the timing characteristics of a clock buffer include intrinsic delay and skew between output pins.

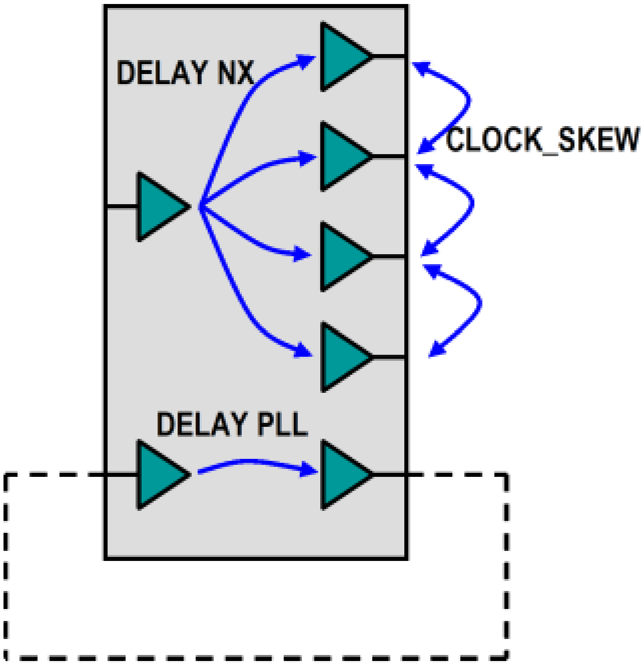
The commands to support this are:

- DELAY NX (delay non-inverting)
- DELAY IX (delay inverting)
- CLOCK_SKEW (skew between outputs)



Then you can see below that for PLLs, there is an additional command:

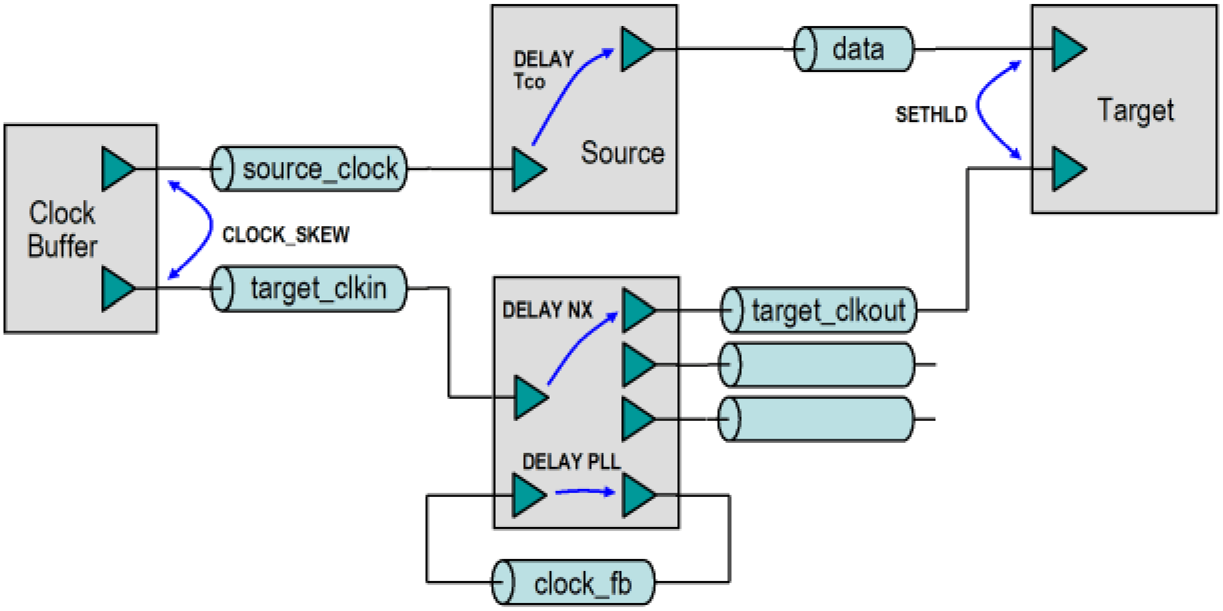
- DELAY PLL (specifies the feedback input and output pins)



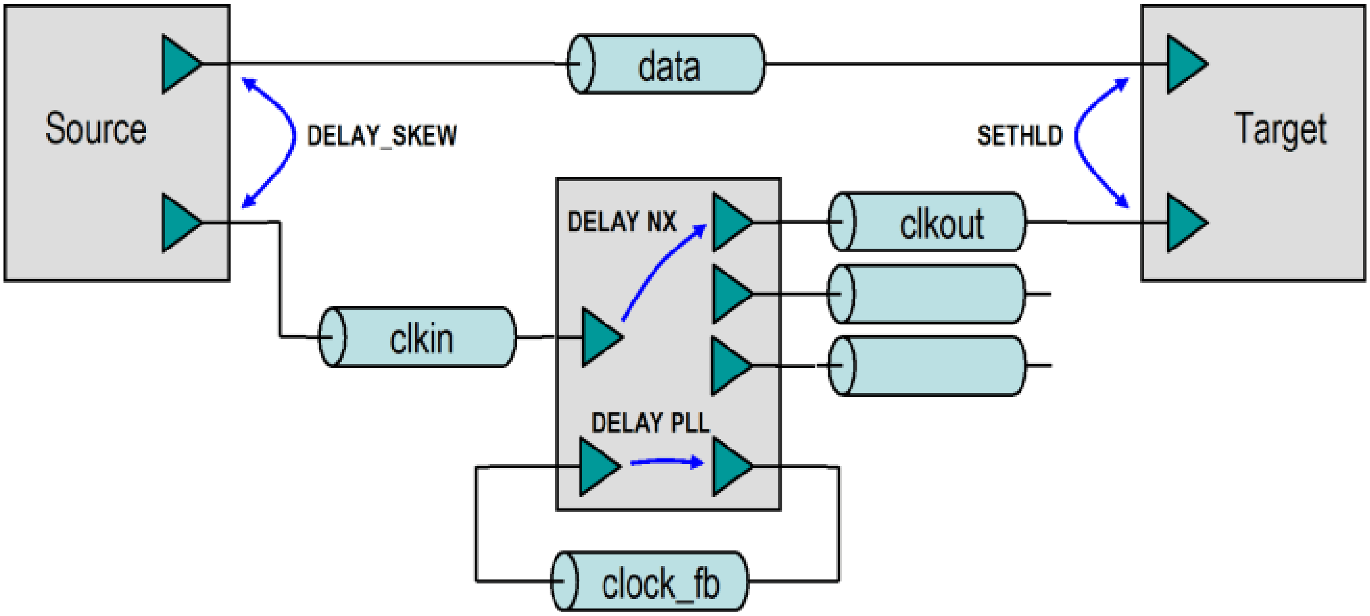
When tracing a clock buffer or PLL, **Parallel Link Designer** uses the "DELAY NX" or "DELAY IX" statements in the timing model to determine the connectivity between the Transfer Net that is connected to the input of the buffer or the PLL and Transfer Net that is connected to the output of the buffer or PLL. **Parallel Link Designer** must have this information in order to trace the complete clock path.

Example Systems

Parallel Link Designer supports timing through clock buffers for both Synchronous and Source-Synchronous timing. As you can see below for a Synchronous system, transfers between a source and target device would have Setup and Hold skew dynamically calculated by tracing back the clock tree from both devices through any clock distribution components to a common source clock component.



For a Source-Synchronous system, transfers between a source and target device would have the clock path delay dynamically calculated by tracing back the clock from the target device through any clock distribution components to the source device.



Timing Configuration Through Clock Buffers

In order to enable timing through a clock buffer, you will need to perform the following:

1. You must first perform the base Synchronous or Source-Synchronous timing configuration as described in User Guide sections 13.5.1 and 13.5.2 for the data signals.

2. You must then create separate Transfer Net schematic sheets of type Clock or Strobe for the clocks into and out of the clock buffer (e.g., `target_clkin` and `target_clkout` shown in the figure above). **Note:** If the outputs of the clock buffer go to different target parts, a Transfer Net will need to be created for each unique target part.

3. Add a `DELAY NX` or `DELAY IX` statement in the timing model for the clock buffer between the input pin and the output pin `PINDEF` groups. A non-inverting clock buffer that has a minimum delay of 500ps and maximum delay of 2.5ns between the input and output pins would have the following `DELAY NX` statement:

```
DELAY NX CKIN *TO CKOUT 0.500 2.500
```

Timing Configuration Through a PLL

Parallel Link Designer supports timing through PLL's, including the feedback path, for both Synchronous and Source-Synchronous timing. This can be accomplished either with or without simulating the feedback path. In order to enable timing through a PLL, you will need to perform the following:

1. You must perform the base Synchronous or Source-Synchronous timing configuration as described in User Guide sections 13.5.1 and 13.5.2 for the data signals.

2. You must create separate Transfer Net schematic sheets of type Clock or Strobe for the clocks into and out of the clock buffer (e.g., "clkout' and "clkin" shown in the figure above showing a PLL).

Note: If the outputs of the clock buffer go to different target parts, a transfer net will need to be created for each unique target part.

3. Create a schematic sheet for the PLL feedback path of type Clock. The source part and the target part will be the PLL.

4. Add a `DELAY PLL` statement in the timing model for the PLL between the feedback output pin and feedback input pin `PINDEF` groups. A feedback path that has a minimum delay of 250ps and maximum delay of 350ps would have the following `DELAY PLL` statement:

```
DELAY PLL FBIN *TO FBOUT 0.250 0.350
```

Note: The above `DELAY PLL` statement can also be used while simulating the feedback path. Any delay values derived from simulation will take precedence over the values in the `DELAY PLL` statement.

Configuration for Trained Timing

Trained Timing is used by devices that align a clock or strobe with a data valid window by delaying the clock or strobe using a DLL (Delay Lock Loop). In the **Parallel Link Designer** timing model a `TRAINED` statement is required to specify the DLL parameters (e.g., step size or granularity) for the `PINDEF` pin group for a clock or strobe.

For example, if a DQ and DQS pin group has the following `DELAY_SKEW` statements in a controller timing model:

```
DELAY_SKEW R DQS1 *TO DQ1 DQ_SKEW_MIN DQ_SKEW_MAX
```

```
DELAY_SKEW F DQS1 *TO DQ1 DQ_SKEW_MIN DQ_SKEW_MAX
```

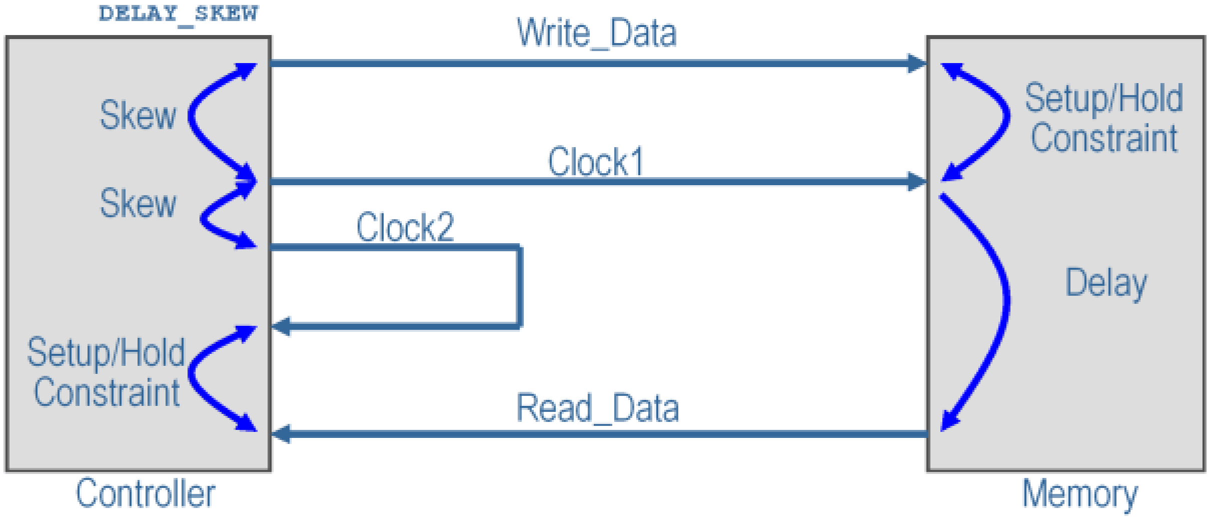
The TRAINED statements for the pin group DQS1 would be:

```
TRAINED_DELAY_SKEW DQS1 MIN_TAP_INC MAX_TAP_INC TAP_GRAN
```

The Waveform and Timing report will include the Training Details tab that will show the margin without training, the optimal sample time and the DLL tap that give margins closest to the optimal margins. For details on the `TRAINED...` statements see the User Guide section 9.8.2.2.8: "*TRAINED_DELAY_SKEW, TRAINED_DELAY_CORELLATED, TRAINED_SETHLD*."

Timing Configuration for External Loop Clock

This is a system where a controller generates a read and write clock, and the read clock is looped back to the controller (see below).



In this system writes (Controller to Memory) are Source-Synchronous and reads (Memory to Controller) are synchronous. The controller will need a `DELAY_SKEW` statement for the `Clock1` to `Write_Data` relationship as shown below:

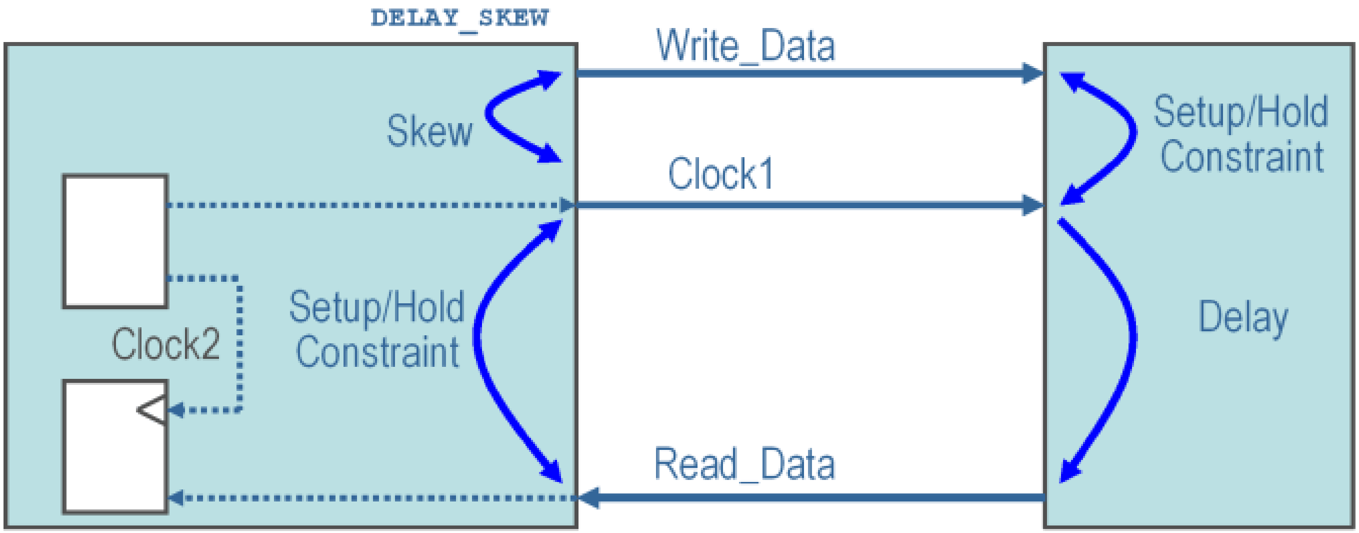
```
DELAY_SKEW R Clock1_out *TO Write_Data SKEW_MIN SKEW_MAX

CLOCK_SKEW Clock1_out *TO Clock2_out SKEW_MIN SKEW_MAX

SETHLD Read_Data *TO R Clock2_in SETUP HOLD
```

The memory timing model would have a `DELAY` statement for the `Clock1` to `Read_Data` timing relationship, as shown below:

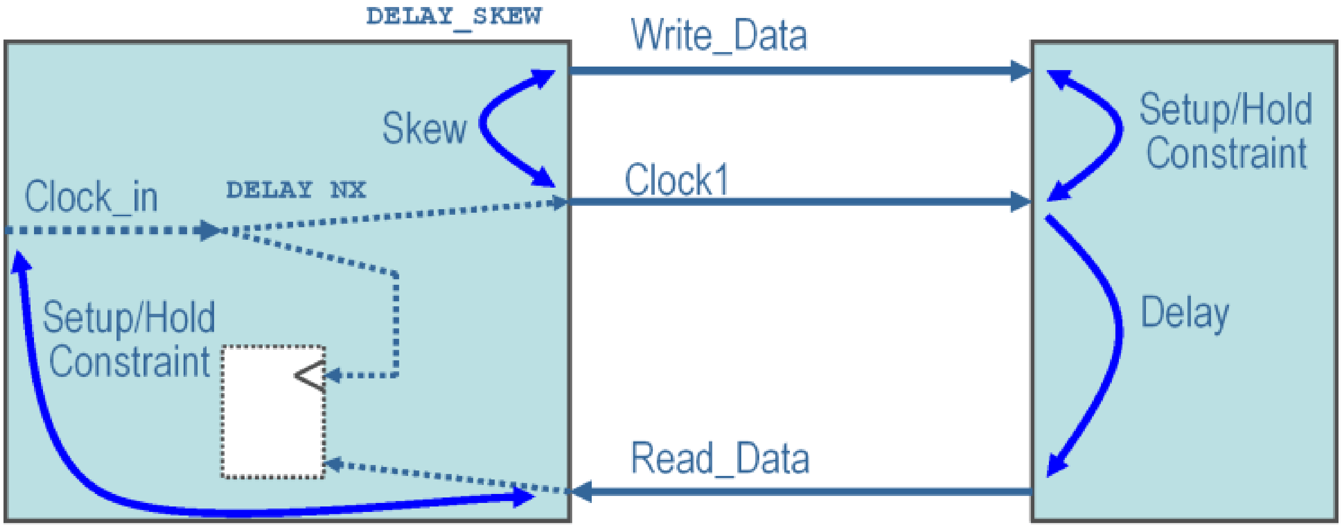
```
DELAY R Clock1_in *TO Read_Data DELAY_MIN DELAY_MAX

SETHLD Write_Data *TO R Clock2_in SETUP HOLD
```

Timing Configuration for Internal Loop Clock

If the clock labeled Clock2 in the diagram above for "External Loop Clock" were inside the controller, the system would look like this:



The read and write timing is the same as in the external case (see User Guide section 13.4: "*Pre-layout vs. Post-layout timing*"), except the clock for the read data is internal to the controller. The data sheet timing references the Setup and Hold of the read data to `Clock1`, the output clock from the controller. **Parallel Link Designer** still requires an input clock for the Setup and Hold constraint for read data at the controller, so a dummy input clock must be created. This dummy input clock has a `DELAY NX` statement with zero delay to `Clock1`, so it has the same timing as `Clock1`. The configuration is shown below:



The dummy input clock pin can be an existing unused clock pin in the IBIS file, or a pin can be added to the IBIS component and timing model `PINDEF`. No Transfer Net is required for the dummy input clock. It is only used to create a common reference point for the synchronous timing calculation for data reads.

The controller timing model would have the following timing statements:

```
DELAY_SKEW R Clock1_out *TO Write_Data SKEW_MIN SKEW_MAX
```

```
DELAY NX Clock1_in *TO Clock1_out 0
```

```
SETHLD Read_Data *TO R Clock_in SETUP HOLD
```

The memory timing model would have a `DELAY` statement for the `Clock1` to `Read_Data` timing relationship, as shown below:

```
DELAY R Clock1_in *TO Read_Data DELAY_MIN DELAY_MAX
```

```
SETHLD Write_Data *TO R Clock2_in SETUP HOLD
```

Summary

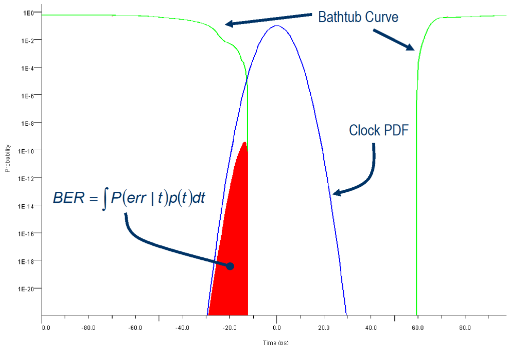
This tutorial has shown you how **Parallel Link Designer** can be used to analyze a Source-Synchronous-Clocked DDRx memory interface for Timing and Waveform Mask Analysis to determine compliance margins, as well as how some other clock architectures can be supported to analyze parallel-link systems.

Jitter and Noise

Model Jitter and Noise While Designing Parallel Link

You can model three major sources of jitter using the STAT mode in the **Parallel Link Designer**: TX clock jitter, RX clock jitter, and RX clock recovery jitter. You can also add RX noise.

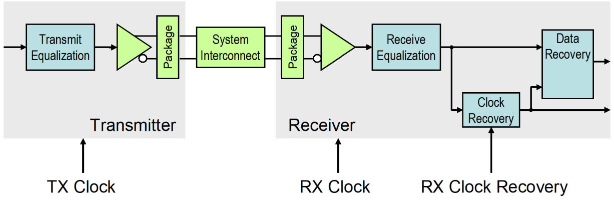
Jitter and noise affect the bit error rate (BER) of a serial channel. Some sources of jitter affect the data bathtub curve and some affect the clock PDF (probability distribution function). The data bathtub and clock PDF are used in the BER calculation, so changing either changes BER.



TX jitter and RX noise always change the data eye and data bathtub. RX jitter and RX clock recovery jitter are handled differently depending on how you set the **Clock Mode** parameter.

- *Normal* — RX clock recovery jitter affects the clock PDF.
- *Clocked* — RX clock recovery jitter affects the data eye and bathtub.
- *Convolved* — RX clock recovery jitter affects the data eye and bathtub.

To access jitter and noise parameters, first select a transfer net sheet by selecting **Setup** > **TNet Properties**. Then open the Designator Element Properties panel by selecting **Properties**. Finally, open the parameters by selecting **Tx Jitter** or **Rx Jitter**. These dialog boxes are also accessible by double clicking any of the designators in the Pre-Layout Analysis tab.



TX Clock Jitter

The app models the TX clock jitter using five parameters. The parameters modify the Tx stimulus (Time Domain analysis) or are added in postprocessing (Statistical Analysis). View and modify these parameters in the TX Jitter dialog box, accessible through the Designator Element Properties dialog box.

TX jitter always changes the data eye and data bathtub.

Jitter Parameter	Description
Tx Rj	Random Gaussian-distributed jitter (RJ) injected at the transmitter. The level is defined as the standard deviation of the RJ, in unit intervals (UI) or seconds. This is sometimes used to model the effects of power supply noise generated by logic switching events in the core of the device. **Tx Rj** affects the stimulus as follows: $\mathit{Time}(n) = n \times \mathit{UI} + \texttt{Tx\_Rj} \times \texttt{randn}$ *Time*(*n*) is the time of edge *n*. `randn` is a function that returns random numbers from the standard normal distribution with mean 0 and variance 1.
Tx Dj	Deterministic jitter (DJ) injected upstream of the link. The level is defined as peak DJ, in UI or seconds. **Tx Dj** accounts for all deterministic and uncorrelated bounded jitter that is not accounted for by **Tx DCD** and **Tx Sj**. DJ is applicable only on the transmit side. The effects of intersymbol interference in the transmission channel are accounted for directly in the analysis or simulation. **Tx Dj** affects the stimulus as follows: $\mathit{Time}(n) = n \times \mathit{UI} + \texttt{Tx\_Dj} \times \texttt{rand}$ `rand` is a function that returns random numbers from a uniform distribution over the interval (-1, 1).
Tx Sj	Sinusoidal jitter (SJ), or sinusoidally varying delay injected at the transmitter. SJ is one half peak-to-peak deviation, in UI or seconds. This is sometimes used to model the effects of power supply noise generated by clock currents in the core of the device. **Tx Sj** affects the stimulus as follows: $\mathit{Time}(n) = n \times \mathit{UI} + \texttt{Tx\_Sj} \times \sin(n \times \mathit{UI} \times 2\pi \times \texttt{Tx\_Sj\_Frequency})$ If **Tx Sj Frequency** is not defined, then **Tx Sj** is ignored.
Tx Sj Frequency	**Tx Sj Frequency** is used explicitly in time domain simulation. Otherwise, **Tx Sj Frequency** is assumed to be much higher than the bandwidth of the clock recovery loop. **Tx Sj Frequency** is specified in Hz.
Tx DCD	Transmission duty cycle distortion (DCD) is defined as the difference in symbol duration between one symbol and the next. The value is the length of the logic `1` side of the clock cycle, as a percentage of the total cycle length, minus 50% in UI or seconds. The calculation assumes that the transmitter is driven by a half-rate clock, with symbols generated on the rising and falling edges of the clock, and further assumes that the duty cycle of that half-rate clock may not be exactly 50%. **Tx DCD** affects the stimulus as follows: $\mathit{Time}(n) = n \times \mathit{UI} + \texttt{Tx\_DCD} \times (-1)^{n}$

Note If you set the TX jitter parameter in the AMI file, you cannot edit the field in the TX Jitter dialog box. To change the jitter, edit the AMI file.

When set to the format `DjRj`, the IBIS-AMI parameter **Tx_Jitter** is translated to TX jitter parameters. The jitter parameter includes the value of *DjMax* and *DjMin*. The parameters are used to generate `Tx_Dj` and `Tx_Rj`:

$$\text{Tx\_Dj} = \frac{DjMax - DjMin}{2}$$

$$\text{Tx\_Rj} = \sigma$$

There is also a shift in the stimulus in time domain analysis:

$$\text{shift} = \frac{DjMax + DjMin}{2}$$

RX Clock Jitter

The RX clock jitter parameters modify the statistics of the recovered clock. These parameters are used to account for jitter that is not included in either the `clock_times` returned by Rx AMI_GetWave or the **Rx_Clock_Recovery** parameters. These parameters are used by the simulator when postprocessing the results from the model and are not passed to the model. These parameters can be viewed and modified in the RX Jitter dialog box.

In the definition of these jitter parameters, *time* is the ideal clock time in statistical analysis and in time domain analysis when Getwave does not exist. *time* is the `clock_time` from Getwave when it exists for time domain analysis.

Jitter Parameter	Description
Rx Rj	Random Gaussian-distributed jitter (RJ) injected at the receiver. The level is defined as the standard deviation of the RJ, in unit intervals (UI) or seconds. This is sometimes used to model the effects of power supply noise generated by logic switching events in the core of the device. **Rx Rj** affects the clock times as follows: $clock\_times(n) = time + \texttt{Rx\_Rj} \times \texttt{randn}$ `randn` is a function that returns random numbers from the standard normal distribution with mean 0 and variance 1.
Rx Dj	Deterministic jitter (DJ), or the worst case half peak-to-peak variation of the recovered clock, not including the random jitter specified by **Rx Rj**, **Rx Sj**, or **Rx DCD**. **Rx Dj** includes all deterministic and uncorrelated bounded jitter that is not accounted for by Rx clock_times, **Rx Rj**, or Rx_Clock_Recovery parameters. **Rx Dj** affects the clock times as follows: $actual\_time = time + \texttt{Rx\_Dj} \times \texttt{rand}$ `rand` is a function that returns random numbers from a uniform distribution over the interval (-1, 1).

Jitter Parameter	Description
Rx Sj	Sinusoidal jitter (SJ), or sinusoidally varying delay injected at the receiver. SJ is one half peak-to-peak deviation, in UI or seconds. This is sometimes used to model the effects of power supply noise generated by clock currents in the core of the device. **Rx Sj** affects the clock times as follows: $actual\_time = time + \texttt{Rx\_Sj} \times \sin(\pi/2 \times \texttt{rand})$ `rand` is a function that returns random numbers from a uniform distribution over the interval (-1, 1).
`Rx_DCD`	Duty cycle distortion (DCD) Difference in symbol duration between one symbol and the next. Assume that the receiver is driven by a half-rate clock, with symbols generated on the rising and falling edges of the clock, and further assume that the duty cycle of that half-rate clock may not be exactly 50%. The value is the length of the logic `1` side of the clock cycle, as a percentage of the total cycle length, minus 50% in unit intervals (UI) or seconds. `Rx_DCD` affects the clock times as follows: $actual\_time = time + \texttt{Rx\_DCD} \times (-1)^n$

Note If you set an RX Jitter parameter in the AMI file, you cannot edit the field in the RX Jitter dialog box. To change the jitter, edit the AMI file.

When set to the format `DjRj`, the IBIS-AMI parameter **Rx_Clock_PDF** is translated to Rx clock recovery jitter parameters. The jitter parameter includes the value of *DjMax* and *DjMin*. The parameters are used to generate `Rx_Clock_Recovery_Dj` and `Rx_Clock_Recovery_Rj`:

$$\text{Rx\_Clock\_Recovery\_Dj} = \frac{DjMax - DjMin}{2}$$

$$\text{Rx\_Clock\_Recovery\_Rj} = \sigma$$

There is also a shift in the stimulus in time domain analysis:

$$\text{Rx\_Clock\_Recovery\_Mean} = \frac{DjMax + DjMin}{2}$$

RX Clock Recovery Jitter

Parallel Link Designer models RX clock recovery jitter using these parameters. This data is used when postprocessing the results from the model. Statistical analysis always uses these parameters. Time domain analysis uses these parameters when the model does not return clock_times or when Rx AMI_GetWave does not exist. These parameters add to any jitter from the RX jitter parameters. Add these parameters to the AMI file using a text editor.

When defining these jitter parameters, *ideal_time* is defined as the halfway between the median of the eye crossing 0.0 on both sides of the eye.

Jitter Parameter	Description
Rx Clock Recovery Mean	Mean phase of recovered clock with respect to the center of the eye diagram (one half symbol from the median data transition time) in unit intervals (UI) or seconds. **Rx Clock Recovery Mean** affects the clock times as follows: $\mathit{actual\_time} = \mathit{ideal\_time} + \texttt{Rx\_Clock\_Recovery\_Mean}$
Rx Clock Recovery Rj	Random Gaussian-distributed jitter (RJ), injected at the clock recovery circuit. The level is defined as the standard deviation of the RJ, in UI or seconds. This is sometimes used to model the effects of power supply noise generated by logic switching events in the core of the device. **Rx Clock Recovery Rj** affects the clock times as follows: $\mathit{actual\_time} = \mathit{ideal\_time} + \texttt{Rx\_Clock\_Recovery\_Rj} \times \texttt{rand}$ `randn` is a function that returns random numbers from the standard normal distribution with mean 0 and variance 1.
Rx Clock Recovery Sj	Sinusoidal jitter (SJ), or sinusoidally varying delay injected at the clock recovery circuit. SJ is one half peak-to-peak deviation, in UI or seconds and a modulation frequency. This is sometimes used to model the effects of power supply noise generated by clock currents in the core of the device. **Rx Clock Recovery Sj** affects the clock times as follows: $\mathit{actual\_time} = \mathit{ideal\_time} + \texttt{Rx\_Clock\_Recovery\_Sj} \times \sin(\pi/2 \times \texttt{rand})$ `rand` is a function that returns random numbers from a uniform distribution over the interval (-1, 1).
Rx Clock Recovery DCD	Duty cycle distortion (DCD) is defined as half the peak-to-peak variation, in UI or seconds, of a clock duty cycle distortion exhibited by the recovered clock. **Rx Clock Recovery DCD** affects the clock times as follows: $\mathit{actual\_time} = \mathit{ideal\_time} + \texttt{Rx\_Clock\_Recovery\_DCD} \times (-1)^{n}$

RX Noise

RX noise parameters modify the statistics associated with the data input to the sampling latch of the receiver. This data is used by **Parallel Link Designer** when postprocessing the results from the model; the budget values specified by the parameters are not passed directly to the model itself.

Noise Parameter	Description
Rx Noise	Standard deviation, in volts into a 100 ohm differential load, of a set of independent samples of a Gaussian noise process measured at the sampling latch of a receiver. **Rx Noise** is Gaussian distributed amplitude noise at the receiver decision point. It is assumed that the samples of this noise process are independent of each other in what is often called an additive white Gaussian noise (AWGN) process. Typically, this noise is generated by shot noise in the receive amplifier. It is seldom if ever accurate to model crosstalk or power supply noise as a Gaussian distributed process. In order to supply an accurate value for this parameter, it might be necessary to account for the gain of the receive amplifier and any analog equalization inserted before the receiver decision point. **Rx Noise** affects the clock times as follows: $wave(t) = wave(t) + \texttt{Rx\_Noise} \times \texttt{randn}$ `randn` is a function that returns random numbers from the standard normal distribution with mean 0 and variance 1. **Note** **Rx GaussianNoise** replaces the **Rx Noise** parameter in the IBIS specification (version 7.0). However, you can use either.
Rx Uniform Noise	Worst-case half peak-to-peak variation, in volts, of a bounded uniform random process. This is added to the signal measured at the sampling latch of a receiver.
Rx Noise Pad	Spectral density of the AWGN at the input to a receiver buffer in volts/sqrt(Hz). Ignored for a driver. Add this parameter to the AMI file using a text editor.

Note If you set an RX Noise parameter in the AMI file, you cannot edit the field in the RX Jitter dialog box. To change the jitter, edit the AMI file.

Set Jitter and Noise in AMI File

You can set jitter and noise parameters in multiple ways depending on the models and the type of simulation. The table shows sample AMI file entries.

Parameter Type	Sample Entry and Description
Value	`(Tx_Rj (Usage Info) (Type UI) (Value 0.01) (Description "TX Random Jitter in UI"))` The Transfer Net Properties dialog box shows the values, but editing is disabled to indicate that the value is controlled by the AMI file.

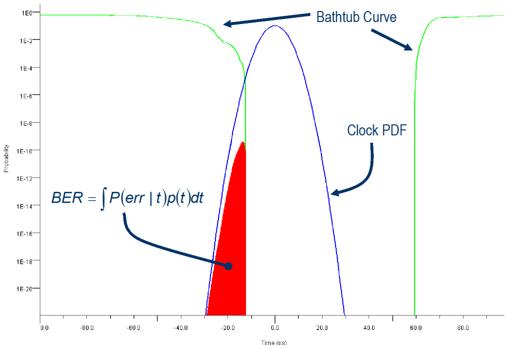
Parameter Type	Sample Entry and Description
Corner	`(Tx_Rj (Usage Info)(Corner 0.005 0.006 0.004)(Type UI) (Description "TX Random Jitter in UI"))` The value used in the analysis is based on the IC process corner selected in the GUI. The Transfer Net Properties dialog box shows <AMI Corner> in the cell for parameters defined in the AMI file as Corner.
Range	`(Tx_Rj (Usage Info)(Format Range 0.0 0.0 0.5)(Type UI) (Default 0) (Description "Tx Random Jitter in UI"))` The parameter will appear in the solution space table and can be swept. The Transfer Net Properties dialog box shows <Sweep> in the cell for parameters defined in the AMI file as Range.

See Also

Model Jitter and Noise While Designing Serial Link

You can model three major sources of jitter using the **Serial Link Designer**: TX clock jitter, RX clock jitter, and RX clock recovery jitter. You can also add RX noise.

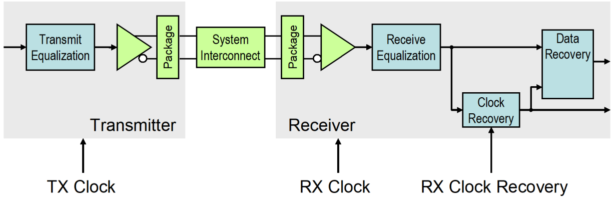
Jitter and noise affect the bit error rate (BER) of a serial channel. Some sources of jitter affect the data bathtub curve and some affect the clock PDF (probability distribution function). The data bathtub and clock PDF are used in the BER calculation, so changing either changes BER.



TX jitter and RX noise always change the data eye and data bathtub. RX jitter and RX clock recovery jitter are handled differently depending on how you set the **Clock Mode** parameter.

- *Normal* — RX clock recovery jitter affects the clock PDF.
- *Clocked* — RX clock recovery jitter affects the data eye and bathtub.
- *Convolved* — RX clock recovery jitter affects the data eye and bathtub.

To access jitter and noise parameters, first select a transfer net sheet by selecting **Setup** > **TNet Properties**. Then open the Designator Element Properties panel by selecting **Properties**. Finally, open the parameters by selecting **Tx Jitter** or **Rx Jitter**. These dialogs are also accessible by double-clicking on any of the designators in the pre-layout analysis tab.



TX Clock Jitter

The app models the TX clock jitter using five parameters. The parameters modify the Tx stimulus (Time Domain analysis) or are added in postprocessing (Statistical Analysis). View and modify these parameters in the TX Jitter dialog box, accessible through the Designator Element Properties dialog box.

TX jitter always changes the data eye and data bathtub.

Jitter Parameter	Description
Tx Rj	Random Gaussian-distributed jitter (RJ) injected at the transmitter. The level is defined as the standard deviation of the RJ, in unit intervals (UI) or seconds. This is sometimes used to model the effects of power supply noise generated by logic switching events in the core of the device. **Tx Rj** affects the stimulus as follows: $Time(n) = n \times UI + \texttt{Tx\_Rj} \times \texttt{randn}$ $Time(n)$ is the time of edge n. `randn` is a function that returns random numbers from the standard normal distribution with mean 0 and variance 1.
Tx Dj	Deterministic jitter (DJ) injected upstream of the link. The level is defined as peak DJ, in UI or seconds. **Tx Dj** accounts for all deterministic and uncorrelated bounded jitter that is not accounted for by **Tx DCD** and **Tx Sj**. DJ is applicable only on the transmit side. The effects of intersymbol interference in the transmission channel are accounted for directly in the analysis or simulation. **Tx Dj** affects the stimulus as follows: $Time(n) = n \times UI + \texttt{Tx\_Dj} \times \texttt{rand}$ `rand` is a function that returns random numbers from a uniform distribution over the interval (-1, 1).
Tx Sj	Sinusoidal jitter (SJ) or sinusoidally varying delay injected at the transmitter. SJ is one half peak-to-peak deviation, in UI or seconds. This is sometimes used to model the effects of power supply noise generated by clock currents in the core of the device. **Tx Sj** affects the stimulus as follows: $Time(n) = n \times UI + \texttt{Tx\_Sj} \times \sin(n \times UI \times 2\pi \times \texttt{Tx\_Sj\_Frequency})$ If **Tx Sj Frequency** is not defined, then **Tx Sj** is ignored.
Tx Sj Frequency	**Tx Sj Frequency** is used explicitly in time domain simulation. Otherwise, **Tx Sj Frequency** is assumed to be much higher than the bandwidth of the clock recovery loop. **Tx Sj Frequency** is specified in Hz.
Tx DCD	Transmission duty cycle distortion (DCD) is defined as the difference in symbol duration between one symbol and the next. The value is the length of the logic `1` side of the clock cycle, as a percentage of the total cycle length, minus 50% in UI or seconds. The calculation assumes that the transmitter is driven by a half-rate clock, with symbols generated on the rising and falling edges of the clock, and further assumes that the duty cycle of that half-rate clock may not be exactly 50%. **Tx DCD** affects the stimulus as follows: $Time(n) = n \times UI + \texttt{Tx\_DCD} \times (-1)^n$

Note If you set a TX jitter parameter in the AMI file, you cannot edit the field in the TX Jitter dialog box. To change the jitter, edit the AMI file.

When set to the format `DjRj`, the IBIS-AMI parameter **Tx_Jitter** is translated to TX jitter parameters. The jitter parameter includes the value of *DjMax* and *DjMin*. The parameters are used to generate `Tx_Dj` and `Tx_Rj`:

$$\text{Tx\_Dj} = \frac{DjMax - DjMin}{2}$$

$$\text{Tx\_Rj} = \sigma$$

There is also a shift in the stimulus in time domain analysis:

$$\text{shift} = \frac{DjMax + DjMin}{2}$$

RX Clock Jitter

The RX clock jitter parameters modify the statistics of the recovered clock. These parameters are used to account for jitter that is not included in either the `clock_times` returned by Rx AMI_GetWave or the **Rx_Clock_Recovery** parameters. These parameters are used by the simulator when postprocessing the results from the model and are not passed to the model. These parameters can be viewed and modified in the RX Jitter dialog box.

In the definition of these jitter parameters, *time* is the ideal clock time in statistical analysis and in time domain analysis when Getwave does not exist. *time* is the `clock_time` from Getwave when it exists for time domain analysis.

Jitter Parameter	Description
Rx Rj	Random Gaussian-distributed jitter (RJ) injected at the receiver. The level is defined as the standard deviation of the RJ, in unit intervals (UI) or seconds. This is sometimes used to model the effects of power supply noise generated by logic switching events in the core of the device. **Rx Rj** affects the clock times as follows: $clock\_times(n) = time + \texttt{Rx\_Rj} \times \texttt{randn}$ `randn` is a function that returns random numbers from the standard normal distribution with mean 0 and variance 1.
Rx Dj	Deterministic jitter (DJ), or the worst case half peak-to-peak variation of the recovered clock, not including the random jitter specified by **Rx Rj**, **Rx Sj**, or **Rx DCD**. **Rx Dj** includes all deterministic and uncorrelated bounded jitter that is not accounted for by Rx clock_times, **Rx Rj**, or Rx_Clock_Recovery parameters. **Rx Dj** affects the clock times as follows: $actual\_time = time + \texttt{Rx\_Dj} \times \texttt{rand}$ `rand` is a function that returns random numbers from a uniform distribution over the interval (-1, 1).

Jitter Parameter	Description
Rx Sj	Sinusoidal jitter (SJ), or sinusoidally varying delay injected at the receiver. SJ is one half peak-to-peak deviation, in UI or seconds. This is sometimes used to model the effects of power supply noise generated by clock currents in the core of the device. **Rx Sj** affects the clock times as follows: $actual\_time = time + \texttt{Rx\_Sj} \times \sin(\pi/2 \times \texttt{rand})$ `rand` is a function that returns random numbers from a uniform distribution over the interval (-1, 1).
`Rx_DCD`	Duty cycle distortion (DCD) Difference in symbol duration between one symbol and the next. Assume that the receiver is driven by a half-rate clock, with symbols generated on the rising and falling edges of the clock, and further assume that the duty cycle of that half-rate clock may not be exactly 50%. The value is the length of the logic `1` side of the clock cycle, as a percentage of the total cycle length, minus 50% in unit intervals (UI) or seconds. `Rx_DCD` affects the clock times as follows: $actual\_time = time + \texttt{Rx\_DCD} \times (-1)^n$

Note If you set an RX Jitter parameter in the AMI file, you cannot the field in the RX Jitter dialog box. To change the jitter, edit the AMI file.

When set to the format `DjRj`, the IBIS-AMI parameter **Rx_Clock_PDF** is translated to Rx clock recovery jitter parameters. The jitter parameter includes the value of *DjMax* and *DjMin*. The parameters are used to generate `Rx_Clock_Recovery_Dj` and `Rx_Clock_Recovery_Rj`:

$$\text{Rx\_Clock\_Recovery\_Dj} = \frac{DjMax - DjMin}{2}$$

$$\text{Rx\_Clock\_Recovery\_Rj} = \sigma$$

There is also a shift in the stimulus in time domain analysis:

$$\text{Rx\_Clock\_Recovery\_Mean} = \frac{DjMax + DjMin}{2}$$

RX Clock Recovery Jitter

Serial Link Designer models RX clock recovery jitter using these parameters. This data is used when postprocessing the results from the model. Statistical analysis always uses these parameters. Time domain analysis uses these parameters when the model does not return clock_times, or when Rx AMI_GetWave does not exist. These parameters add to any jitter from the RX jitter parameters. Add these parameters to the AMI file using a text editor.

While defining these jitter parameters, *ideal_time* is defined as the halfway between the median of the eye crossing 0.0 on both sides of the eye.

Jitter Parameter	Description
Rx Clock Recovery Mean	Mean phase of recovered clock with respect to the center of the eye diagram (one half symbol from the median data transition time) in unit intervals (UI) or seconds. **Rx Clock Recovery Mean** affects the clock times as follows: $actual\_time = ideal\_time + \texttt{Rx\_Clock\_Recovery\_Mean}$
Rx Clock Recovery Rj	Random Gaussian-distributed jitter (RJ), injected at the clock recovery circuit. The level is defined as the standard deviation of the RJ, in UI or seconds. This is sometimes used to model the effects of power supply noise generated by logic switching events in the core of the device. **Rx Clock Recovery Rj** affects the clock times as follows: $actual\_time = ideal\_time + \texttt{Rx\_Clock\_Recovery\_Rj} \times \texttt{rand}$ `randn` is a function that returns random numbers from the standard normal distribution with mean 0 and variance 1.
Rx Clock Recovery Sj	Sinusoidal jitter (SJ), or sinusoidally varying delay injected at the clock recovery circuit. SJ is one half peak-to-peak deviation, in UI or seconds and a modulation frequency. This is sometimes used to model the effects of power supply noise generated by clock currents in the core of the device. **Rx Clock Recovery Sj** affects the clock times as follows: $actual\_time = ideal\_time + \texttt{Rx\_Clock\_Recovery\_Sj} \times \sin(\pi/2 \times \texttt{rand})$ `rand` is a function that returns random numbers from a uniform distribution over the interval (-1, 1).
Rx Clock Recovery DCD	Duty cycle distortion (DCD) is defined as half the peak-to-peak variation, in UI or seconds, of a clock duty cycle distortion exhibited by the recovered clock. **Rx Clock Recovery DCD** affects the clock times as follows: $actual\_time = ideal\_time + \texttt{Rx\_Clock\_Recovery\_DCD} \times (-1)^{n}$

RX Noise

RX noise parameters modify the statistics associated with the data input to the sampling latch of the receiver. This data is used by **Serial Link Designer** when postprocessing the results from the model; the budget values specified by the parameters are not passed directly to the model itself.

Noise Parameter	Description
Rx Noise	Standard deviation, in volts into a 100 ohm differential load, of a set of independent samples of a Gaussian noise process measured at the sampling latch of a receiver. **Rx Noise** is Gaussian distributed amplitude noise at the receiver decision point. It is assumed that the samples of this noise process are independent of each other in what is often called an additive white Gaussian noise (AWGN) process. Typically, this noise would be generated by shot noise in the receive amplifier. It is seldom if ever accurate to model crosstalk or power supply noise as a Gaussian distributed process. In order to supply an accurate value for this parameter, it might be necessary to account for the gain of the receive amplifier and any analog equalization inserted before the receiver decision point. **Rx Noise** affects the clock times as follows: $wave(t) = wave(t) + \texttt{Rx\_Noise} \times \texttt{randn}$ `randn` is a function that returns random numbers from the standard normal distribution with mean 0 and variance 1. **Note** **Rx GaussianNoise** replaces the **Rx Noise** parameter in the IBIS specification (version 7.0). However, you can use either.
Rx Uniform Noise	Worst-case half peak-to-peak variation, in volts, of a bounded uniform random process. This is added to the signal measured at the sampling latch of a receiver.
Rx Noise Pad	Spectral density of the AWGN at the input to a receiver buffer in volts/sqrt(Hz). Ignored for a driver. This parameter must be added to the AMI file using a text editor.

Note If you set an RX Noise parameter in the AMI file, you cannot edit the field in the RX Jitter dialog box. To change the jitter, edit the AMI file.

Set Jitter and Noise in AMI File

You can set jitter and noise parameters in multiple ways depending on the models and the type of simulation. The table shows sample AMI file entries.

Parameter Type	Sample Entry and Description
Value	`(Tx_Rj (Usage Info) (Type UI) (Value 0.01) (Description "TX Random Jitter in UI"))` The Transfer Net Properties dialog box shows this value, but editing is disabled to indicate that the value is controlled by the AMI file.

Parameter Type	Sample Entry and Description
Corner	`(Tx_Rj (Usage Info)(Corner 0.005 0.006 0.004)(Type UI) (Description "TX Random Jitter in UI"))` The value used in the analysis is based on the IC process corner selected in the GUI. The Transfer Net Properties dialog box shows the <AMI Corner> in the cell for parameters defined in the AMI file as Corner.
Range	`(Tx_Rj (Usage Info)(Format Range 0.0 0.0 0.5)(Type UI) (Default 0) (Description "Tx Random Jitter in UI"))` The parameter will appear in the solution space table and can be swept. The Transfer Net Properties dialog box shows the <Sweep> in the cell for parameters defined in the AMI file as Range.

See Also

Industry Standard Examples

10GBASE-KR4 Compliance Kit

Characterize and validate the performance of a 10GBASE-KR4 channel design.

10GBASE-KR is a 10 Gb/s data rate baseband specification, with a backplane medium, using a 64B/66B coding scheme, in a four-lane configuration. The IEEE 802.3-2008 Annex 69B provides guidelines for a backplane design where meeting or exceeding the loss and crosstalk masks provides a high confidence of a successful channel design. However, if a channel does not meet the masks, it does not mean that the backplane will not operate at a specified bit error rate. It means that these channels need more analysis through simulation. It is possible that transmitter and receiver equalization can overcome loss or crosstalk deficiencies that do not meet the appropriate mask given.

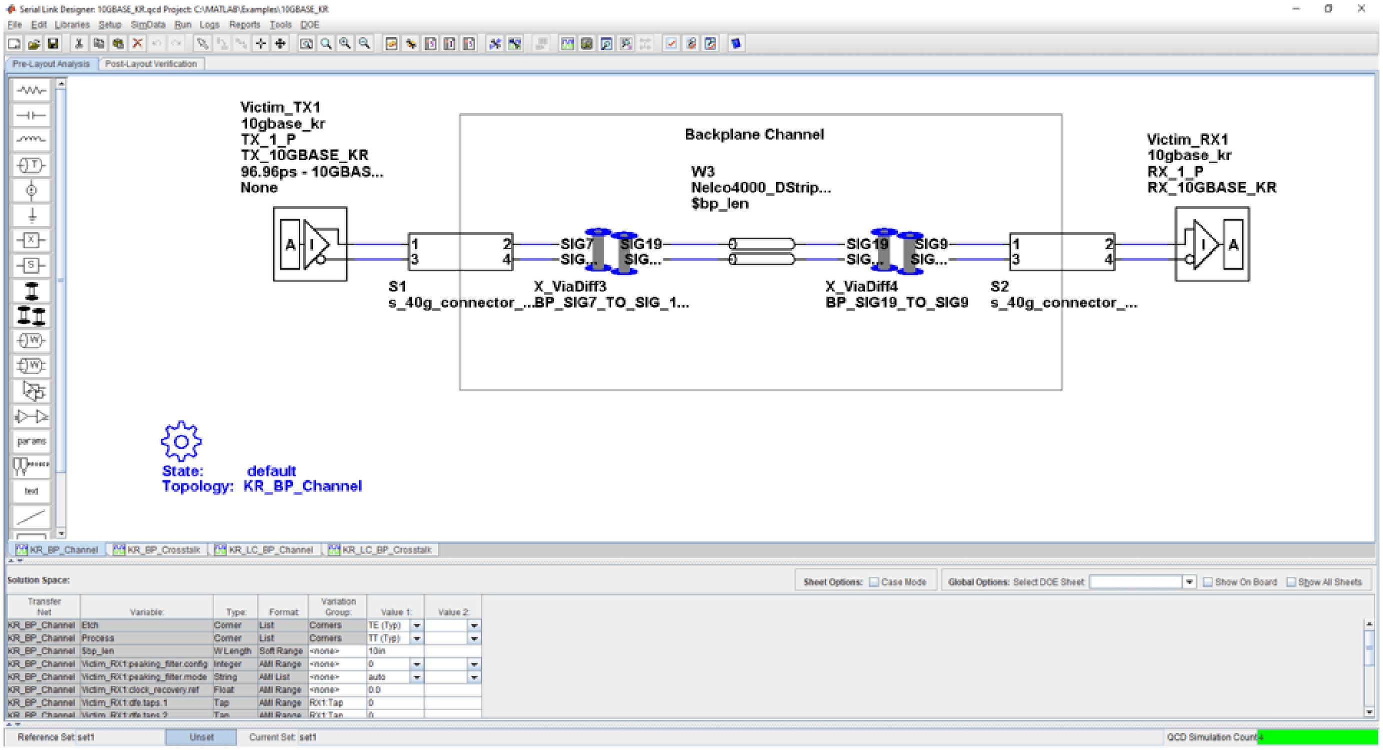
This kit is designed for analysis of a backplane design with two mated connectors as given in the Annex 69B section of the 10GBASE-KR specification. The kit also includes sheets containing the backplane and connectors with two plug-in line cards attached. IBIS-AMI TX and RX models are provided with representative package models. Widebus sheets in this kit are included for crosstalk simulations only.

This kit enables you to insert a channel design and test it against the supplied masks to determine if the channel has a high confidence of success. Otherwise further investigation and simulation will need to be performed to determine if the channel meets the target bit-error rate.

Open 10GBASE-KR4 Kit

Open the 10GBASE-KR4 kit in the **Serial Link Designer** app using the `openSignalIntegrityKit` function.

```
openSignalIntegrityKit("10GBASE_KR");
```



Serial Link Designer: 10GBASE_KR.qcd Project: C:\MATLAB\Examples\10GBASE_KR

State: default
Topology: KR_BP_Channel

Solution Space:

Transfer Net	Variable:	Type:	Format:	Variation Group:	Value 1:	Value 2:
KR_BP_Channel	Etch	Corner	List	Corners	TE (Typ)	
KR_BP_Channel	Process	Corner	List	Corners	TT (Typ)	
KR_BP_Channel	$bp_len	W Length	Soft Range	<none>	10in	
KR_BP_Channel	Victim_RX1:peaking_filter.config	Integer	AMI Range	<none>	0	
KR_BP_Channel	Victim_RX1:peaking_filter.mode	String	AMI List	<none>	auto	
KR_BP_Channel	Victim_RX1:clock_recovery.ref	Float	AMI Range	<none>	0.0	
KR_BP_Channel	Victim_RX1:dfe.taps.1	Tap	AMI Range	RX1:Tap	0	
KR_BP_Channel	Victim_RX1:dfe.taps.2	Tap	AMI Range	RX1:Tap	0	

Reference Set: set1

Current Set: set1

Kit Overview

- Project Name: 10GBASE_KR
- Interface Name: 10GBASE_KR
- Target Operating Frequency: 10.3125 Gb/s (UI = 96.967 ps)

This kit defines one schematic set.

- **10GBASE-KR Sheets** – backplane only (single channel and widebus crosstalk sheets) and one backplane with 2 line cards connected (single channel and widebus crosstalk sheets)

For more information about the 10GBASE-KR4 channel compliance schematics, transfer net properties, and compliance rules, refer to the document 10GBASE_KR4.pdf that is attached to this example as a supporting file.

References

[1] IEEE Std 802.3-2008 (Annex 68B). Module 10GBASE-KR parameters.

See Also

Serial Link Designer

100GBASE-KR4 Compliance Kit

Characterize and validate the performance of a 100GBASE-KR4 channel design.

100GBASE-KR is a 100 Gb/s data rate baseband specification, with a backplane medium, using a 64B/66B coding scheme, in a four-lane configuration. The IEEE 802.3bj specification Annex 93A defines methods for compliance of electrical channels operating at 25.78 Gb/s. Compliance of a channel to this specification is determined by its channel operating margin (COM). The COM calculation that is defined in the specification is based on many factors, including insertion loss, return loss, and cross-coupling. The transmitter and receiver equalization and package models are also taken into account.

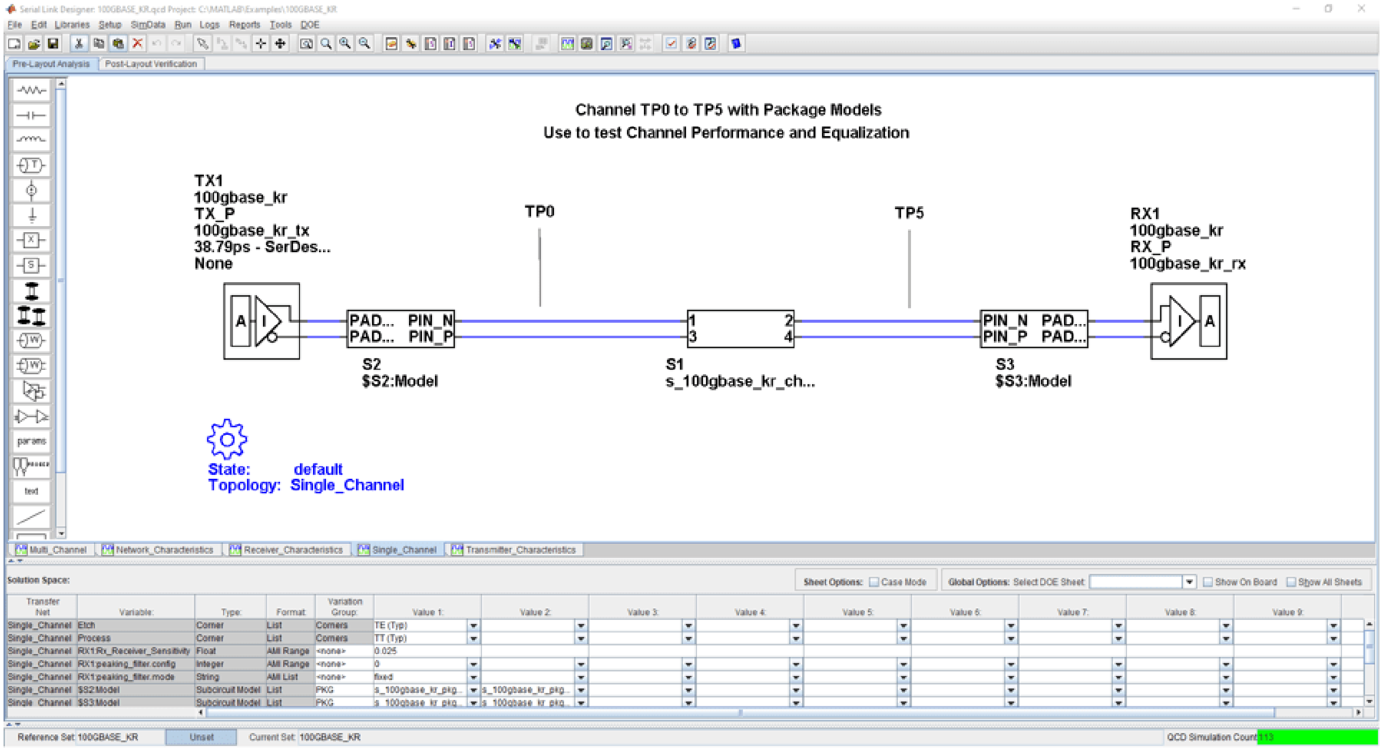
This kit is designed for analysis of a 25.78 Gb/s channel based on 802.3bj compliance. The kit provides a test environment for designing and analyzing channels and their performance prior to COM testing. You can also run COM on channel designs. The kit implements most of the characteristics and parameters for transmitters, receivers and channels outlined in Section 93 of 802.3bj-2014.pdf.

The interface contains five schematic sheets. One sheet is used to test channel characteristics such as insertion loss and return loss. Also included are individual sheets for testing transmitter and receiver characteristics. Lastly, two sheets are used for statistical and time domain analysis on single and coupled channels. TX and RX IBIS-AMI models are provided that implement the equalization requirements of COM. Package models were created from transmission lines and package-to-board capacitance to represent lengths of 12 mm and 30 mm.

Open 100GBASE-KR4 Kit

Open the 100GBASE-KR4 kit in the **Serial Link Designer** app using the `openSignalIntegrityKit` function.

```
openSignalIntegrityKit("100GBASE_KR");
```



Solution Space:

Sheet Options: □ Case Mode

Global Options: Select DOE Sheet: □ Show On Board □ Show All Sheets

Transfer Net	Variable:	Type:	Format:	Variation Group:	Value 1:	Value 2:	Value 3:	Value 4:	Value 5:	Value 6:	Value 7:	Value 8:	Value 9:
Single_Channel	Etch	Corner	List	Corners	TE (Typ)								
Single_Channel	Process	Corner	List	Corners	TT (Typ)								
Single_Channel	RX1:Rx_Receiver_Sensitivity	Float	AMI Range	<none>	0.025								
Single_Channel	RX1:peaking_filter.config	Integer	AMI Range	<none>	0								
Single_Channel	RX1:peaking_filter.mode	String	AMI List	<none>	fixed								
Single_Channel	$S2:Model	Subcircuit Model	List	PKG	s_100gbase_kr_pkg...	s_100gbase_kr_pkg...							
Single_Channel	$S3:Model	Subcircuit Model	List	PKG	s_100gbase_kr_pkg...	s_100gbase_kr_pkg...							

Reference Set: 100GBASE_KR Unset Current Set: 100GBASE_KR QCD Simulation Count 113

Kit Overview

- Project Name: 100GBASE_KR
- Interface Name: 100GBASE_KR
- Target Operating Frequency: 25.78 Gb/s (UI = 38.79 ps)

This kit defines one schematic set.

- **100GBASE_KR** – Network_Characteristics, Single_Channel and Multi_Channel

For more information about the 100GBASE-KR4 channel compliance schematics, transfer net properties, and compliance rules, refer to the document 100GBASE_KR4.pdf that is attached to this example as a supporting file.

References

[1] IEEE Standard for Ethernet, Amendment 2: Physical Layer specifications and Management Parameters for 100Gb/s Operation Over Backplanes and Copper cables. 802.3bj-2014.pdf.

See Also

Serial Link Designer

CAUI-4 Chip-to-Chip Compliance Kit

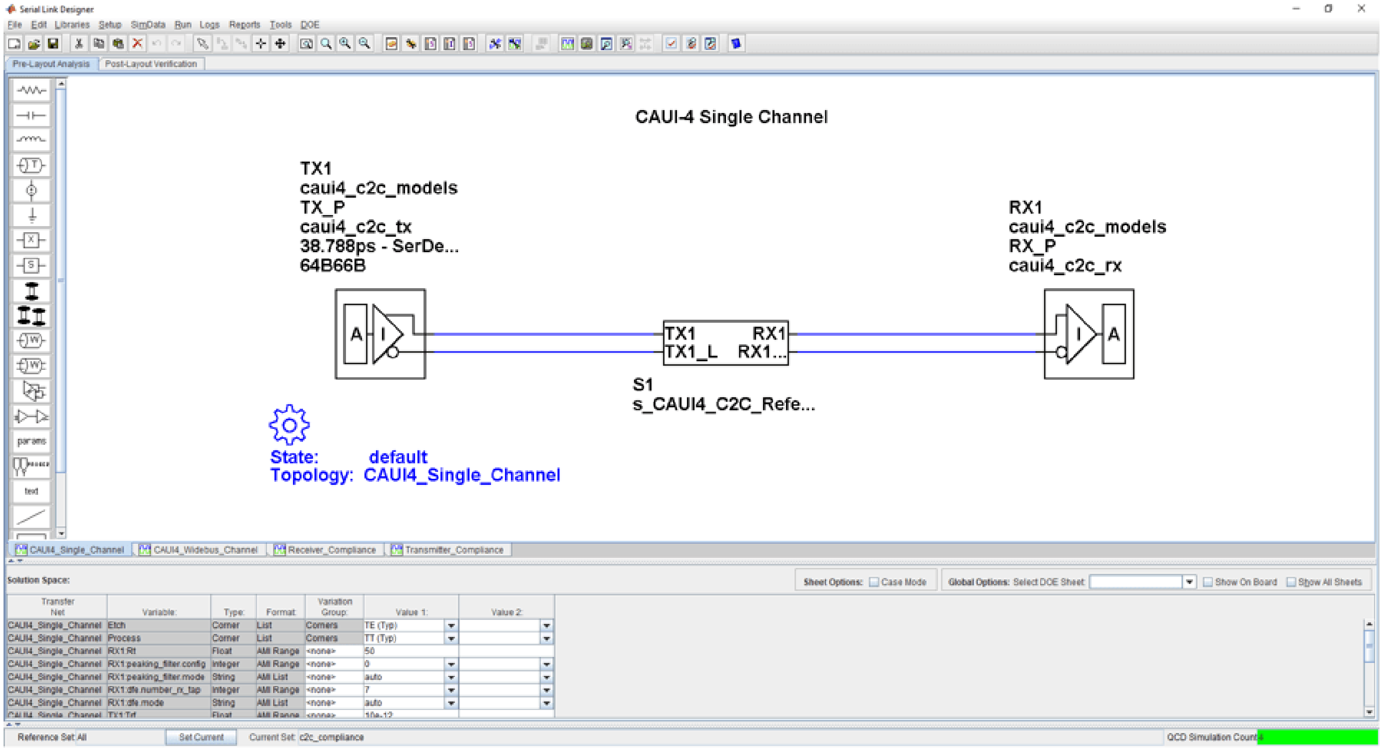
Test the compliance of simulation models and topologies to the CAUI-4 chip-to-chip (C2C) specification.

This kit is designed for a chip-to-chip interface between system devices with up to 25 inches of PCB etch and one connector. The kit includes IBIS-AMI TX and RX models for reference and compliance testing. This kit enables you to insert a channel design and test for compliance as specified in the CAUI-4 C2C specification (802.3bm-2015, Annex 83D).

Open CAUI-4 C2C Kit

Open the CAUI-4 C2C kit in the **Serial Link Designer** app using the `openSignalIntegrityKit` function.

```
openSignalIntegrityKit("CAUI_4_C2C_83D");
```



Solution Space:

Sheet Options: □ Case Mode

Global Options: Select DOE Sheet: □ Show On Board □ Show All Sheets

Transfer Net	Variable:	Type:	Format:	Variation Group:	Value 1:	Value 2:
CAUI4_Single_Channel	Etch	Corner	List	Corners	TE (Typ)	
CAUI4_Single_Channel	Process	Corner	List	Corners	TT (Typ)	
CAUI4_Single_Channel	RX1:Rt	Float	AMI Range	<none>	50	
CAUI4_Single_Channel	RX1:peaking_filter.config	Integer	AMI Range	<none>	0	
CAUI4_Single_Channel	RX1:peaking_filter.mode	String	AMI List	<none>	auto	
CAUI4_Single_Channel	RX1:dfe.number_rx_tap	Integer	AMI Range	<none>	7	
CAUI4_Single_Channel	RX1:dfe.mode	String	AMI List	<none>	auto	
CAUI4_Single_Channel	TX1:Trf	Float	AMI Range	<none>	10e-12	

Reference Set: All

Current Set: c2c_compliance

Kit Overview

- Project Name: CAUI4_C2C_83D
- Interface Name: CAUI4_C2C_83D
- Target Operating Frequency: 25.781 Gb/s (UI = 38.788 ps)

For more information about the CAUI-4 C2C channel compliance schematics, transfer net properties and compliance rules, refer to the document CAUI4_C2C.pdf that is attached to this example as a supporting file.

References

[1] IEEE 802.3bj-2014 Section 93 and Annex 93A (COM). 802.3bj-2014.pdf.

[2] IEEE 802.3bm-2015 Annex 83D. 802.3bm-2015.pdf.

See Also

Serial Link Designer

CAUI-4 Chip-to-Module Compliance Kit

Test the compliance of simulation models and topologies to the CAUI-4 chip-to-module (C2M) specification.

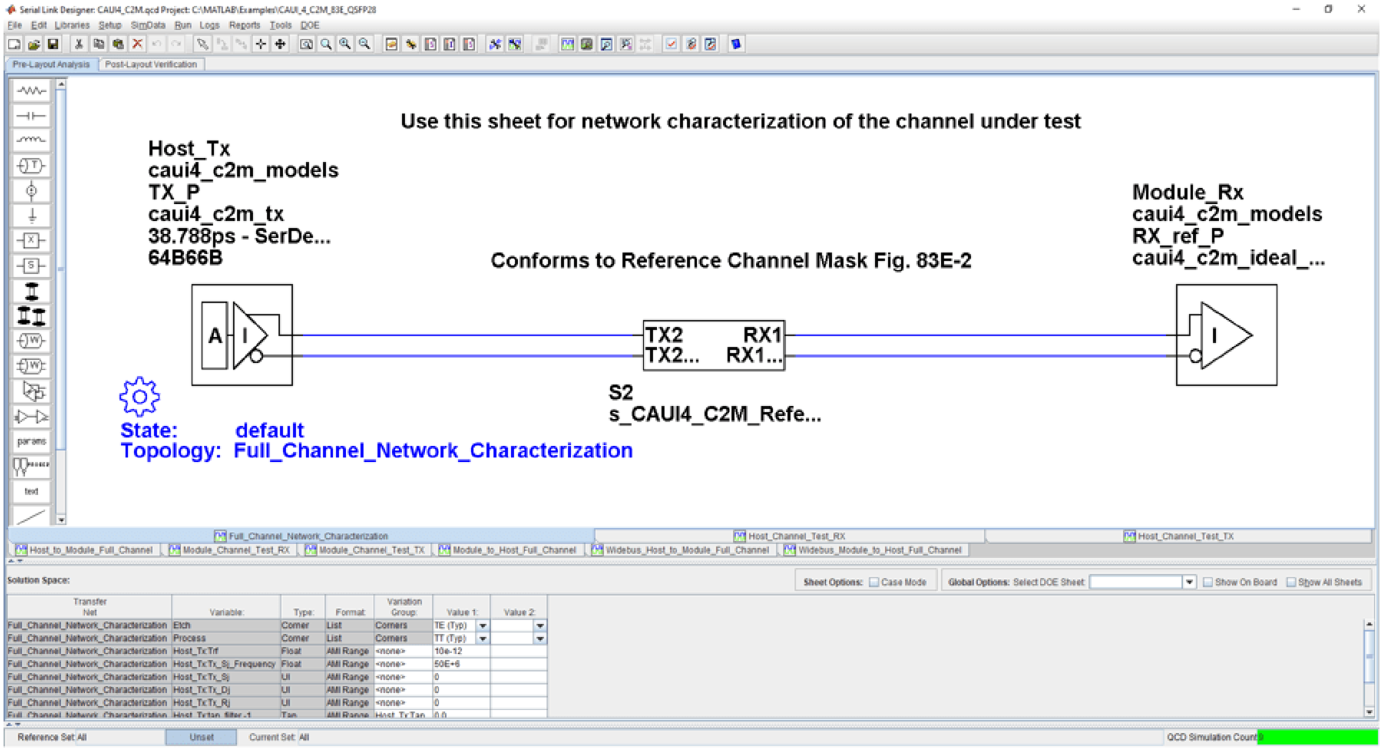
This kit is designed for analysis of a host board and an optical module. The channel consists of a host board connected with a mated connector to a module board. The channels between the transmitting and receiving devices operate at 25.718 Gb/s. The kit has sheets for network characterization to analyze the passive channels (insertion and return). There are single channel sheets to analyze the effects of inter-symbol interference on performance. You can optimize TX and RX equalization and test compliance with a stressed eye. Multi-channel sheets represent the four 25 Gb/s channels to determine the effects of crosstalk on channel eye margin.

This kit enables you to insert a channel design and characterize and validate its performance using the specification masks to determine if the channel has a high confidence of success. If the channel does not meet the compliance masks or BER estimates, further investigation or redesign, along with simulation, will need to be performed to determine possible changes to meet compliance.

Open CAUI-4 C2M Kit

Open the CAUI-4 C2M kit in the **Serial Link Designer** app using the `openSignalIntegrityKit` function.

```
openSignalIntegrityKit("CAUI_4_C2M_83E_QSFP28");
```



Serial Link Designer: CAUI4_C2M.qcd Project: C:\MATLAB\Examples\CAUI_4_C2M_83E_QSFP28

File Edit Libraries Setup SimData Run Logs Reports Tools DOE
Pre-Layout Analysis
Post-Layout Verification
Use this sheet for network characterization of the channel under test
Host_Tx
caui4_c2m_models
TX_P
caui4_c2m_tx
38.788ps - SerDe...
64B66B
Conforms to Reference Channel Mask Fig. 83E-2
Module_Rx
caui4_c2m_models
RX_ref_P
caui4_c2m_ideal_...
A
I
TX2
TX2...
RX1
RX1...
S2
s_CAUI4_C2M_Refe...
I
params
text
Full_Channel_Network_Characterization
Host_Channel_Test_RX
Host_Channel_Test_TX
Host_to_Module_Full_Channel
Module_Channel_Test_RX
Module_Channel_Test_TX
Module_to_Host_Full_Channel
Widebus_Host_to_Module_Full_Channel
Widebus_Module_to_Host_Full_Channel
Unset
QCD Simulation Count 9

State: default
Topology: Full_Channel_Network_Characterization

Solution Space:

Sheet Options: □ Case Mode
Global Options: Select DOE Sheet □ Show On Board □ Show All Sheets

Transfer Net	Variable:	Type:	Format:	Variation Group:	Value 1:	Value 2:
Full_Channel_Network_Characterization	Etch	Corner	List	Corners	TE (Typ)	
Full_Channel_Network_Characterization	Process	Corner	List	Corners	TT (Typ)	
Full_Channel_Network_Characterization	Host_Tx:Trf	Float	AMI Range	<none>	10e-12	
Full_Channel_Network_Characterization	Host_Tx:Tx_Sj_Frequency	Float	AMI Range	<none>	50E+6	
Full_Channel_Network_Characterization	Host_Tx:Tx_Sj	UI	AMI Range	<none>	0	
Full_Channel_Network_Characterization	Host_Tx:Tx_Dj	UI	AMI Range	<none>	0	
Full_Channel_Network_Characterization	Host_Tx:Tx_Rj	UI	AMI Range	<none>	0	
Full_Channel_Network_Characterization	Host_Tx:tap_filter_-1	Tap	AMI Range	Host_Tx:Tap	0.0	

Reference Set: All
Current Set: All

Kit Overview

- Project Name: CAUI_4_C2M_83E_QSFP28

- Interface Name: CAUI4_C2M_83E
- Operating Frequency: 25.78 Gb/s (UI = 38.788 ps)

Schematic sheets are included for testing a CAUI-4 C2M channel with mated connector to a module board. The masks provided in this kit are given in the 28.05 Gb/s CAUI-4 specification.

For more information about the CAUI-4 C2M channel compliance schematics, transfer net properties and compliance rules, refer to the document CAUI4_C2M.pdf that is attached to this example as a supporting file.

References

[1] IEEE 802.3bm-2015 Specification. Annex 83E. 802.3bm-2015.pdf.

[2] IEEE 802.3bj-2014 Specification. Annex 92. 802.3bj-2014.pdf.

See Also

Serial Link Designer

CAUI/XLAUI Chip-to-Chip Compliance Kit

Test the compliance of simulation models and topologies to the CAUI/XLAUI chip-to-chip (C2C) specification.

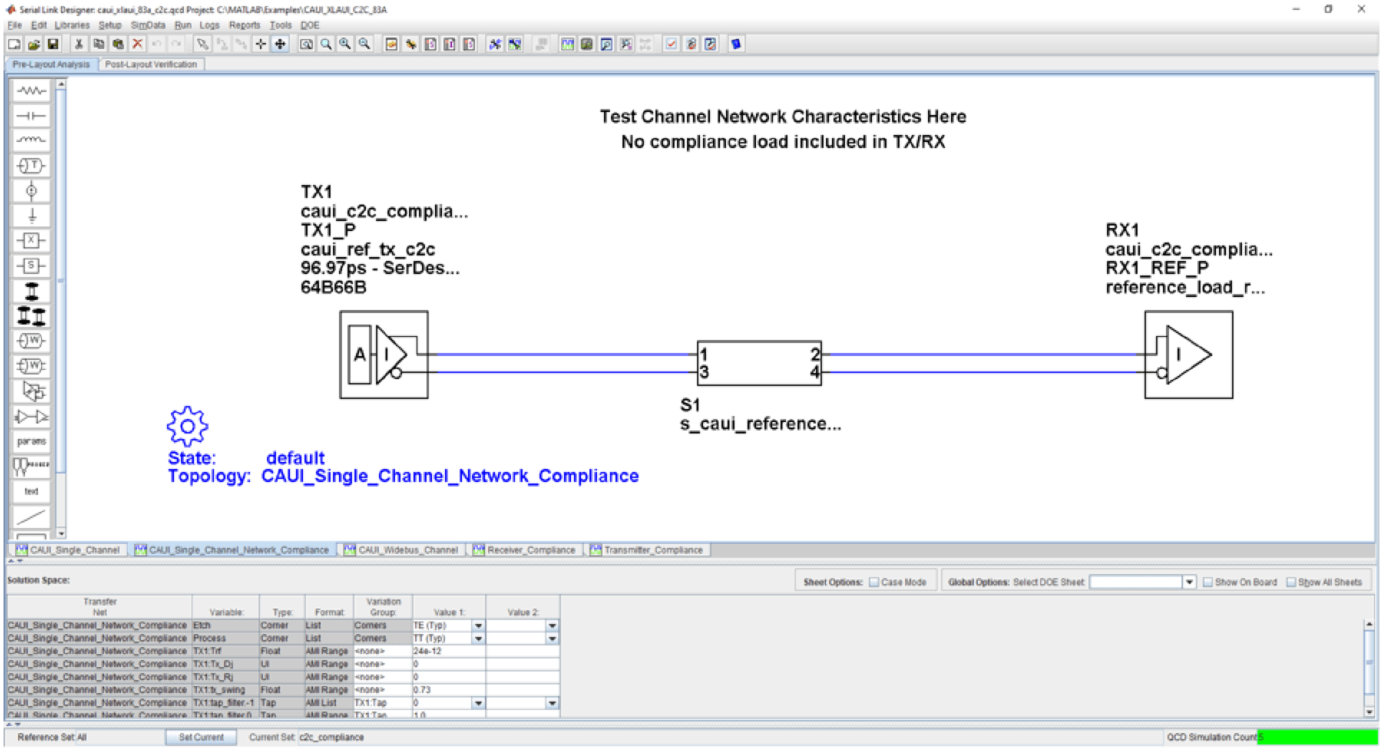
The 802.3ba Annex 83A specification defines physical layer compliance for a CAUI (100 Gb/s) or XLAUI (40 Gb/s) C2C interface.

This kit is designed for compliance of both host and boards. The kit includes IBIS-AMI TX and RX models for reference and compliance testing. You can insert a channel design and test for compliance as specified in the CAUI/XLAUI specification.

Open CAUI/XLAUI C2C Kit

Open the CAUI/XLAUI C2C kit in the **Serial Link Designer** app using the `openSignalIntegrityKit` function.

```
openSignalIntegrityKit("CAUI_XLAUI_C2C_83A");
```


Serial Link Designer: caui_xlaui_83a_c2c.qcd Project: C:\MATLAB\Examples\CAUI_XLAUI_C2C_83A
File Edit Libraries Setup SimData Run Logs Reports Tools DOE
Pre-Layout Analysis
Post-Layout Verification
params
text
Test Channel Network Characteristics Here
No compliance load included in TX/RX
TX1
caui_c2c_complia...
TX1_P
caui_ref_tx_c2c
96.97ps - SerDes...
64B66B
A
I
1
3
2
4
S1
s_caui_reference...
RX1
caui_c2c_complia...
RX1_REF_P
reference_load_r...
I
State: default
Topology: CAUI_Single_Channel_Network_Compliance
CAUI_Single_Channel
CAUI_Single_Channel_Network_Compliance
CAUI_Widebus_Channel
Receiver_Compliance
Transmitter_Compliance

Solution Space:

Sheet Options: □ Case Mode

Global Options: Select DOE Sheet: □ Show On Board □ Show All Sheets

Transfer Net	Variable:	Type:	Format:	Variation Group:	Value 1:	Value 2:
CAUI_Single_Channel_Network_Compliance	Etch	Corner	List	Corners	TE (Typ)	
CAUI_Single_Channel_Network_Compliance	Process	Corner	List	Corners	TT (Typ)	
CAUI_Single_Channel_Network_Compliance	TX1:Tr/f	Float	AMI Range	<none>	24e-12	
CAUI_Single_Channel_Network_Compliance	TX1:Tx_Dj	UI	AMI Range	<none>	0	
CAUI_Single_Channel_Network_Compliance	TX1:Tx_Rj	UI	AMI Range	<none>	0	
CAUI_Single_Channel_Network_Compliance	TX1:tx_swing	Float	AMI Range	<none>	0.73	
CAUI_Single_Channel_Network_Compliance	TX1:tap_filter.-1	Tap	AMI List	TX1:Tap	0	
CAUI_Single_Channel_Network_Compliance	TX1:tap_filter.0	Tap	AMI Range	TX1:Tap	1.0	

Reference Set: All | Set Current | Current Set: c2c_compliance | QCD Simulation Count 5

Kit Overview

- Project Name: CAUI_XLAUI_83A
- Interface Name: CAUI_XLAUI_83A
- Target Operating Frequency: 10.3125 Gb/s (UI = 96.97 ps)

The CAUI/XLAUI C2C kit defines one schematic set.

- c2c_compliance — Used for all compliance testing

For more information about the CAUI/XLAUI C2C channel compliance schematics, transfer net properties, and compliance rules, refer to the document CAUI_XLAUI_C2C.pdf that is attached to this example as a supporting file.

References

[1] IEEE 802.3ba-2010 Annex 83A. 802.3ba-2010.pdf.

See Also

Serial Link Designer

CAUI/XLAUI Chip-To-Module Compliance Kit

Test the compliance of simulation models and topologies to the CAUI/XLAUI chip-to-module (C2M) specification.

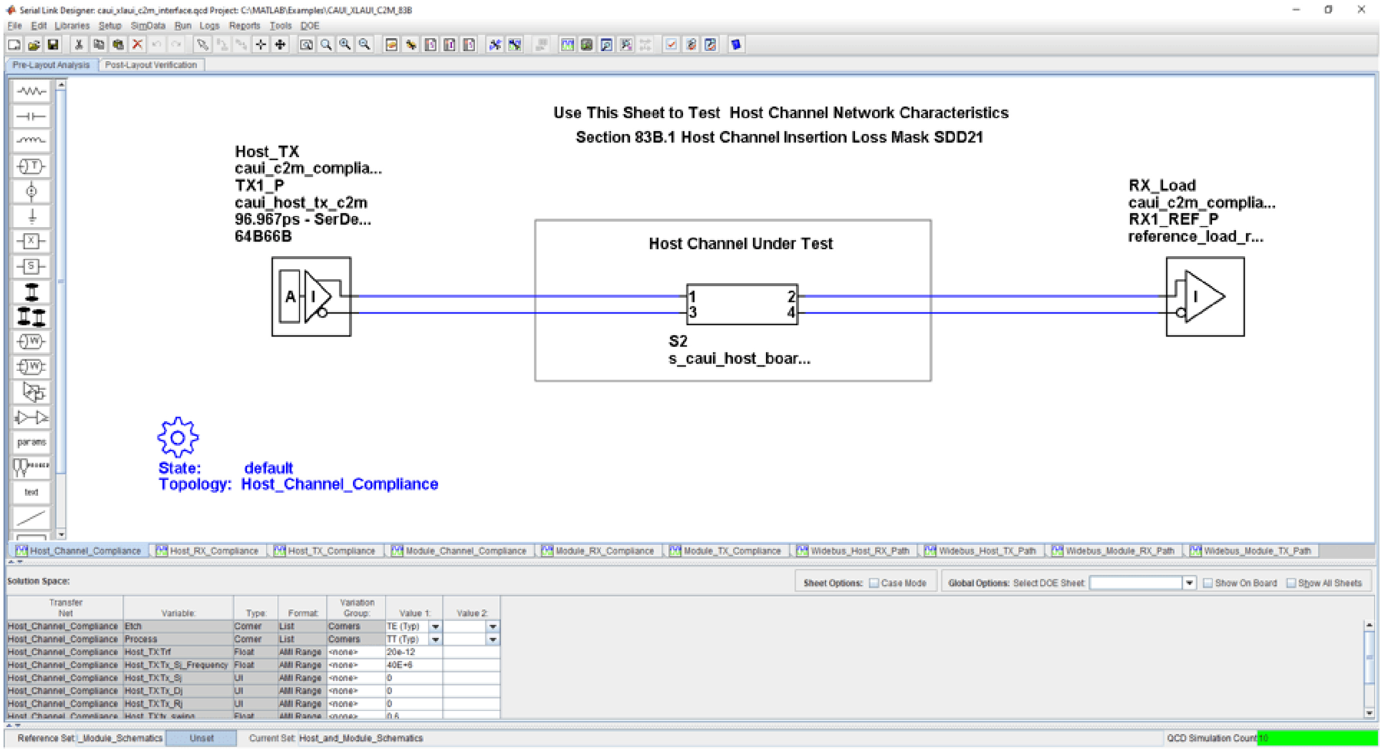
802.3ba Annex 83B specification defines physical layer compliance for a CAUI (100 Gb/s) or XLAUI (40 Gb/s) C2M interface. Host and module board compliance are both detailed including the channel, transmitter and receiver electrical requirements.

This kit is designed for compliance of both host and boards. The kit includes IBIS-AMI TX and RX models for reference and compliance testing. You can insert a channel design and test for compliance as specified in the CAUI/XLAUI specification.

Open CAUI/XLAUI C2M Kit

Open the CAUI/XLAUI C2M kit in the **Serial Link Designer** app using the `openSignalIntegrityKit` function.

```
openSignalIntegrityKit("CAUI_XLAUI_C2M_83B");
```



Use This Sheet to Test Host Channel Network Characteristics
Section 83B.1 Host Channel Insertion Loss Mask SDD21

State: default
Topology: Host_Channel_Compliance

Solution Space:

Transfer Net	Variable:	Type:	Format:	Variation Group:	Value 1:	Value 2:
Host_Channel_Compliance	Etch	Corner	List	Corners	TE (Typ)	
Host_Channel_Compliance	Process	Corner	List	Corners	TT (Typ)	
Host_Channel_Compliance	Host_TXTrf	Float	AMI Range	<none>	20e-12	
Host_Channel_Compliance	Host_TXTx_Sj_Frequency	Float	AMI Range	<none>	40E+6	
Host_Channel_Compliance	Host_TXTx_Sj	UI	AMI Range	<none>	0	
Host_Channel_Compliance	Host_TXTx_Dj	UI	AMI Range	<none>	0	
Host_Channel_Compliance	Host_TXTx_Rj	UI	AMI Range	<none>	0	
Host_Channel_Compliance	Host_TXtx_swing	Float	AMI Range	<none>	0.6	

Kit Overview

- Project Name: CAUI_XLAUI_C2M_83B
- Interface Name: CAUI_XLAUI_C2M_83B
- Target Operating Frequency: 10.3125 Gb/s (UI = 96.97 ps)

The CAUI/XLAUI C2M kit defines four schematic sets.

- Module_Board — Used for module compliance testing
- Host_Board — Used for host compliance testing
- HCB_MCB_Characterization — Used for analyzing HCB and MCB compliance
- All_Project_Schematics — Set of all project schematics

For more information about the CAUI/XLAUI C2M channel compliance schematics, transfer net properties, and compliance rules, refer to the document CAUI_XLAUI_C2M.pdf that is attached to this example as a supporting file.

References

[1] IEEE 802.3ba-2010 Annex 83B. 802.3ba-2010.pdf.

See Also

Serial Link Designer

CEI 25G-LR Compliance Kit

Characterize and validate the performance of a CEI 25G-LR channel design.

CEI 25G-LR is a common electrical interface (CEI) implementation agreement (IA) that supports 25 Gb/s over "Long Reach" (LR) backplane architectures. The CEI-25G-LR Clause is part of the Common Electrical I/O 3.0 Implementation Agreement.

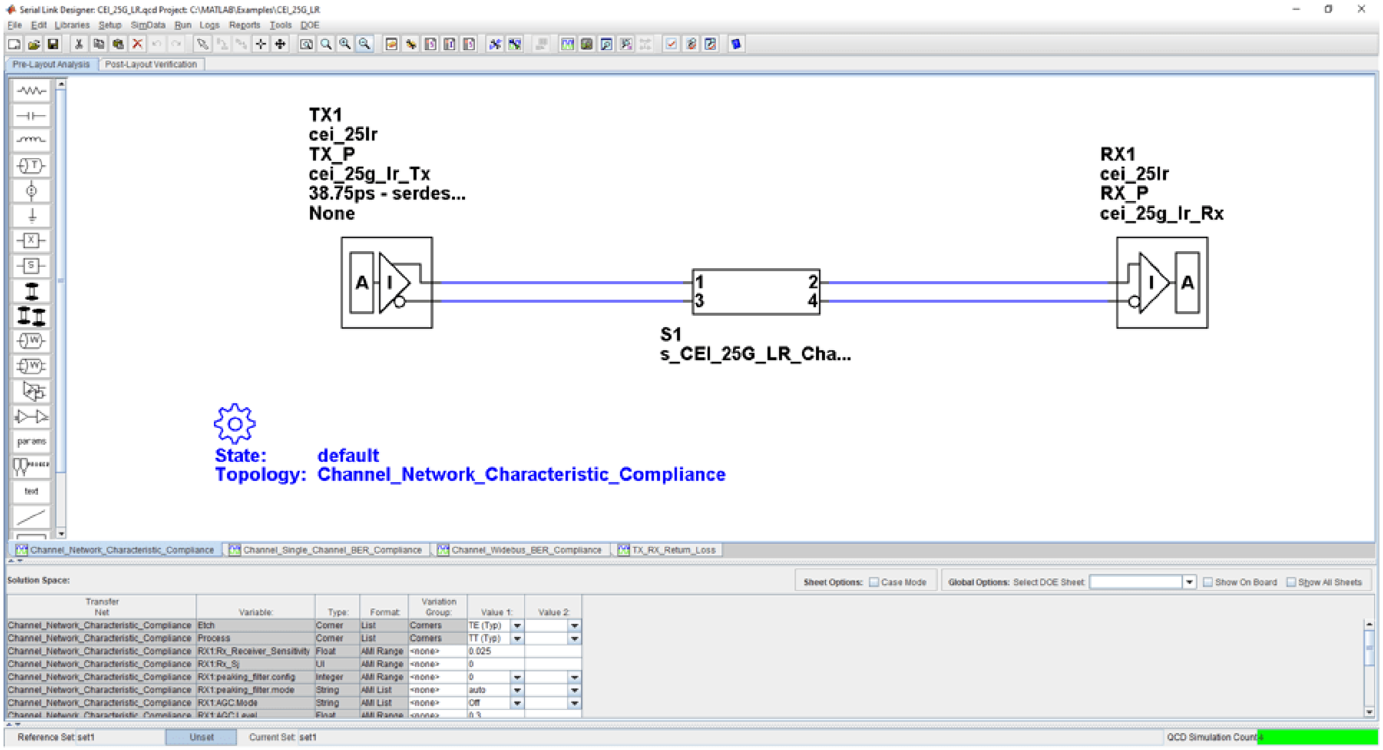
This kit is designed for analysis of a backplane channel design between module boards. The total channel length is approximately 30 inches. It has module boards connected with two mated connectors which represents the interconnect between ASICs transmitting and receiving 25 Gb/s data over the channel. The kit has three sheets: one for single channel BER compliance testing, one for multi-channel FEXT/NEXT crosstalk to measure BER compliance and a sheet for simulating network characteristics for compliance.

This kit enables you to insert a channel design and characterize and validate its performance using the specification masks to determine if the channel has a high confidence of success. If the channel does not meet the compliance masks or BER estimates, further investigation or redesign, along with simulation, will need to be performed to determine possible changes to meet compliance.

Open CEI 25G-LR Kit

Open the CEI 25G-LR kit in the **Serial Link Designer** app using the `openSignalIntegrityKit` function.

```
openSignalIntegrityKit("CEI_25G_LR");
```



Serial Link Designer: CEI_25G_LR.qcd Project: C:\MATLAB\Examples\CEI_25G_LR
File Edit Libraries Setup SimData Run Logs Reports Tools DOE
Pre-Layout Analysis
Post-Layout Verification
TX1
cei_25lr
TX_P
cei_25g_lr_Tx
38.75ps - serdes...
None
A
I
1
3
2
4
S1
s_CEI_25G_LR_Cha...
RX1
cei_25lr
RX_P
cei_25g_lr_Rx
I
A
params
text
Channel_Network_Characteristic_Compliance
Channel_Single_Channel_BER_Compliance
Channel_Widebus_BER_Compliance
TX_RX_Return_Loss
Sheet Options: □ Case Mode
Global Options: Select DOE Sheet:
□ Show On Board
□ Show All Sheets
QCD Simulation Count

State: default
Topology: Channel_Network_Characteristic_Compliance

Solution Space:

Transfer Net	Variable:	Type:	Format:	Variation Group:	Value 1:	Value 2:
Channel_Network_Characteristic_Compliance	Etch	Corner	List	Corners	TE (Typ)	
Channel_Network_Characteristic_Compliance	Process	Corner	List	Corners	TT (Typ)	
Channel_Network_Characteristic_Compliance	RX1:Rx_Receiver_Sensitivity	Float	AMI Range	<none>	0.025	
Channel_Network_Characteristic_Compliance	RX1:Rx_Sj	UI	AMI Range	<none>	0	
Channel_Network_Characteristic_Compliance	RX1:peaking_filter.config	Integer	AMI Range	<none>	0	
Channel_Network_Characteristic_Compliance	RX1:peaking_filter.mode	String	AMI List	<none>	auto	
Channel_Network_Characteristic_Compliance	RX1:AGC.Mode	String	AMI List	<none>	Off	
Channel_Network_Characteristic_Compliance	RX1:AGC.Level	Float	AMI Range	<none>	0.3	

Reference Set: set1 Unset Current Set: set1

Kit Overview

- Project Name: CEI_25G_LR
- Interface Name: CEI_25G_LR
- Operating Frequency: 25.8 Gb/s (UI = 38.75 ps)

The CEI 25G-LR kit defines one schematic set. Schematic sheets are included for testing a CEI 25G-LR channel with mated connectors, and a cable/backplane with two plug-in cards. The masks provided in this kit are given in the 25 Gb/s CEI_25G_LR specification

- **Default** – Schematic sheets focused on channel characterization and BER compliance

For more information about the CEI 25G-LR channel compliance schematics, transfer net properties, and compliance rules, refer to the document CEI_25G_LR.pdf that is attached to this example as a supporting file.

See Also

Serial Link Designer

CEI 28G-SR Compliance Kit

Characterize and validate the performance of a CEI 28G-SR channel design.

CEI 28G-SR is a common electrical interface (CEI) implementation agreement (IA) that supports 28 Gb/s over "Short Reach" (SR) chip-to-chip applications. The CEI-28G-SR Clause is part of the Common Electrical I/O 3.0 Implementation Agreement.

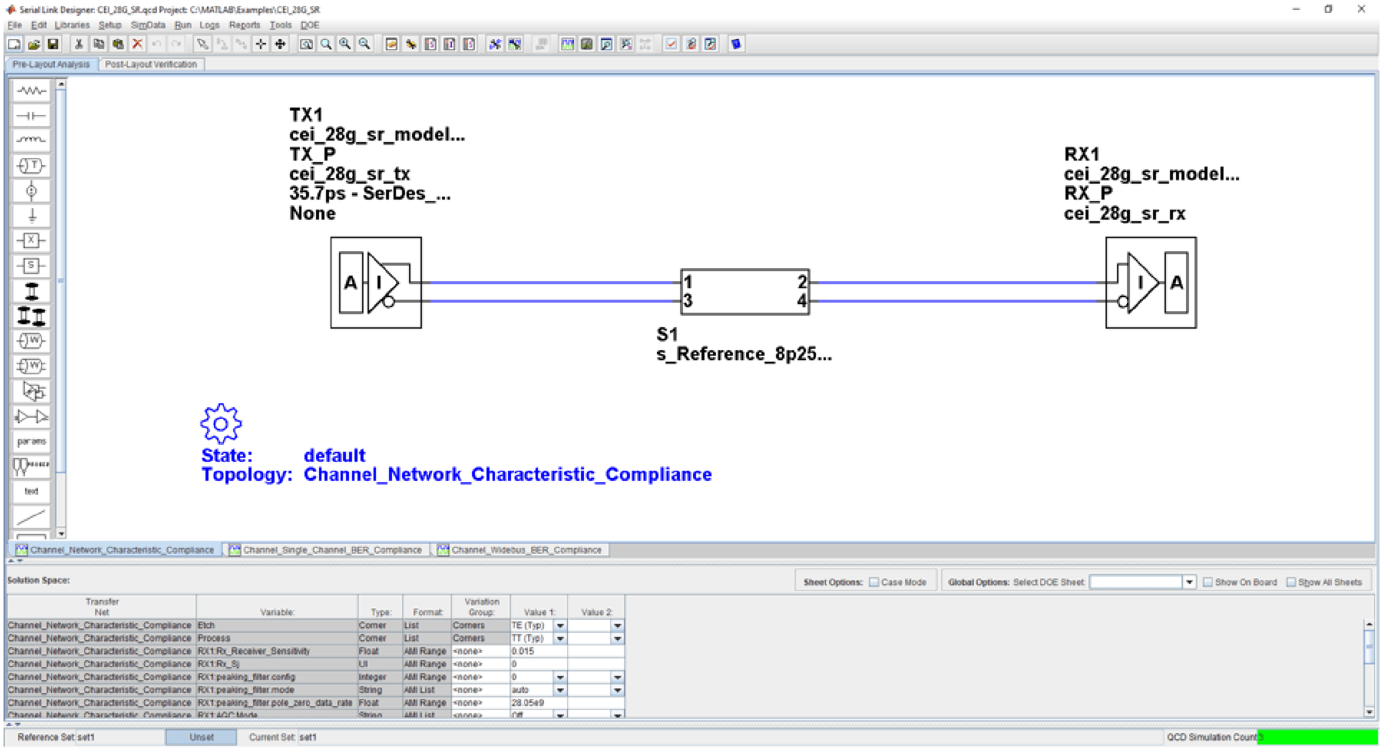
This kit is designed for analysis of a host board and QSFP+ 100G module. The total channel length is approximately 8.25 inches. It consists of a host board connected with a mated connector which represents the interconnect between the transmitting and receiving 28.05 Gb/s data across the channel. The kit has three sheets; one for single channel BER compliance testing, one for multi-channel FEXT/NEXT crosstalk to measure BER compliance and a sheet for simulating network characteristics for compliance.

This kit enables you to insert a channel design and characterize and validate its performance using the specification masks to determine if the channel has a high confidence of success. If the channel does not meet the compliance masks or BER estimates, further investigation or redesign, along with simulation, will need to be performed to determine possible changes to meet compliance.

Open CEI 28G-SR Kit

Open the CEI 28G-SR kit in the **Serial Link Designer** app using the `openSignalIntegrityKit` function.

```
openSignalIntegrityKit("CEI_28G_SR");
```



State: default
Topology: Channel_Network_Characteristic_Compliance

Solution Space:

Transfer Net	Variable:	Type:	Format:	Variation Group:	Value 1:	Value 2:
Channel_Network_Characteristic_Compliance	Etch	Corner	List	Corners	TE (Typ)	
Channel_Network_Characteristic_Compliance	Process	Corner	List	Corners	TT (Typ)	
Channel_Network_Characteristic_Compliance	RX1:Rx_Receiver_Sensitivity	Float	AMI Range	<none>	0.015	
Channel_Network_Characteristic_Compliance	RX1:Rx_Sj	UI	AMI Range	<none>	0	
Channel_Network_Characteristic_Compliance	RX1:peaking_filter.config	Integer	AMI Range	<none>	0	
Channel_Network_Characteristic_Compliance	RX1:peaking_filter.mode	String	AMI List	<none>	auto	
Channel_Network_Characteristic_Compliance	RX1:peaking_filter.pole_zero_data_rate	Float	AMI Range	<none>	28.05e9	
Channel_Network_Characteristic_Compliance	RX1:AGC.Mode	String	AMI List	<none>	Off	

Reference Set: set1

Current Set: set1

Kit Overview

- Project Name: CEI_28G_SR
- Interface Name: CEI_28G_SR
- Operating Frequency: 28.05 Gbps (UI = 35.75ps)

The CEI 28G-SR kit defines one schematic set.

- **Default** – Schematic sheets focused on channel characterization and BER compliance

For more information about the CEI 28G-SR channel compliance schematics, transfer net properties, and compliance rules, refer to the document CEI_28G_SR.pdf that is attached to this example as a supporting file.

See Also

Serial Link Designer

CEI 28G-VSR Compliance Kit

Characterize and validate the performance of a CEI 28G-VSR channel design.

CEI 28G-VSR is a common electrical interface (CEI) implementation agreement (IA) that supports 28 Gb/s over "Very Short Reach" (VSR) optical or electrical chip-to-module applications. The CEI 28G-VSR Clause is part of the Common Electrical I/O 3.0 Implementation Agreement.

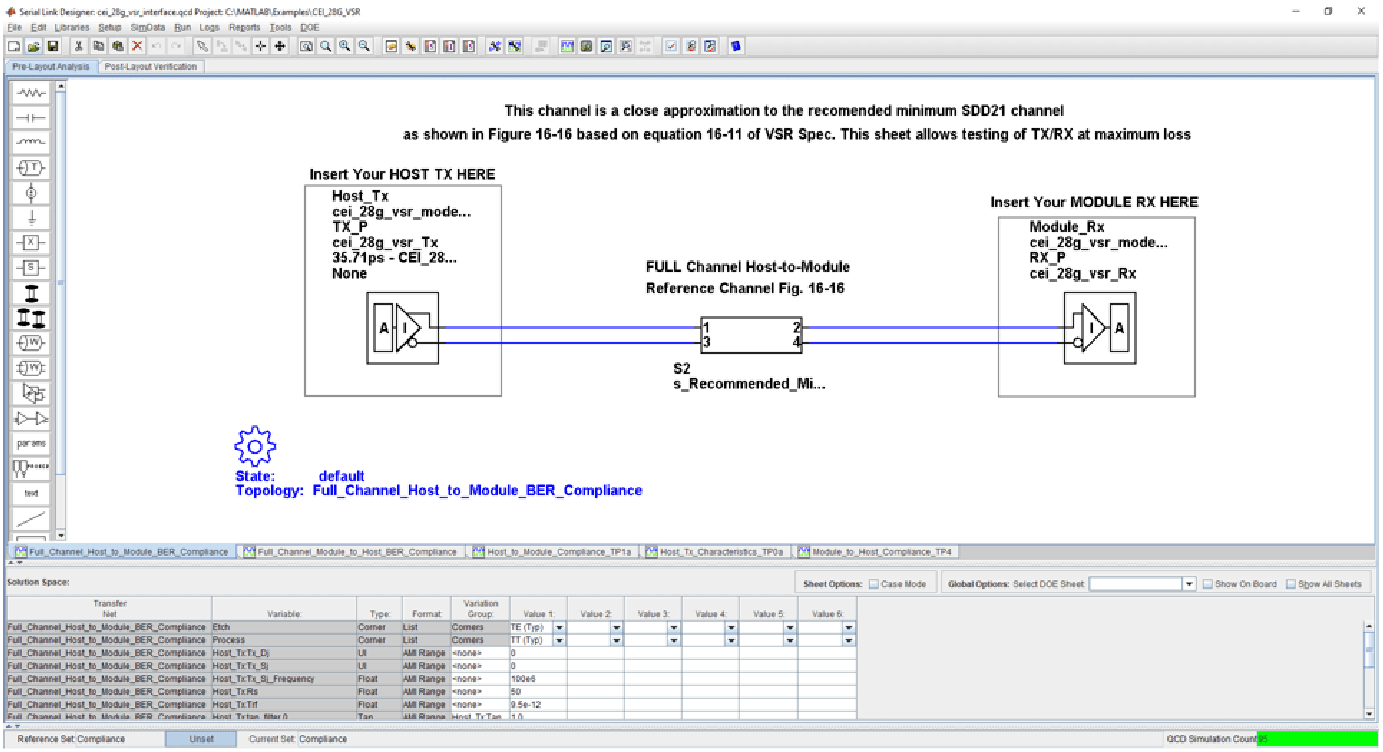
This kit is designed for analysis of a host board and an optical module. The total channel length is approximately 5 inches. The VSR channel consists of a host board connected with a mated connector to a module board which represents the interconnect between the transmitting and receiving 28.05 Gbps data across the channel. The kit has sheets that represent network characterization for insertion and return loss testing, channel FEXT/NEXT crosstalk to measure BER compliance and RX stress testing.

This kit enables you to insert a channel design and characterize and validate its performance using the specification masks to determine if the channel has a “high confidence” of success. If the channel does not meet the compliance masks or BER estimates, further investigation or redesign, along with simulation, will need to be performed to determine possible changes to meet compliance.

Open CEI 28G-VSR Kit

Open the CEI 28G-VSR kit in the **Serial Link Designer** app using the `openSignalIntegrityKit` function.

```
openSignalIntegrityKit("CEI_28G_VSR");
```



Serial Link Designer: cei_28g_vsr_interface.qcd Project: C:\MATLAB\Examples\CEI_28G_VSR

This channel is a close approximation to the recomended minimum SDD21 channel
as shown in Figure 16-16 based on equation 16-11 of VSR Spec. This sheet allows testing of TX/RX at maximum loss

Insert Your HOST TX HERE

Host_Tx
cei_28g_vsr_mode...
TX_P
cei_28g_vsr_Tx
35.71ps - CEI_28...
None

FULL Channel Host-to-Module
Reference Channel Fig. 16-16

S2
s_Recommended_Mi...

Insert Your MODULE RX HERE

Module_Rx
cei_28g_vsr_mode...
RX_P
cei_28g_vsr_Rx

State: default
Topology: Full_Channel_Host_to_Module_BER_Compliance

Solution Space:

Transfer Net	Variable:	Type:	Format:	Variation Group:	Value 1:	Value 2:	Value 3:	Value 4:	Value 5:	Value 6:
Full_Channel_Host_to_Module_BER_Compliance	Etch	Corner	List	Corners	TE (Typ)					
Full_Channel_Host_to_Module_BER_Compliance	Process	Corner	List	Corners	TT (Typ)					
Full_Channel_Host_to_Module_BER_Compliance	Host_TxTx_Dj	UI	AMI Range	<none>	0					
Full_Channel_Host_to_Module_BER_Compliance	Host_TxTx_Sj	UI	AMI Range	<none>	0					
Full_Channel_Host_to_Module_BER_Compliance	Host_TxTx_Sj_Frequency	Float	AMI Range	<none>	100e6					
Full_Channel_Host_to_Module_BER_Compliance	Host_TxRs	Float	AMI Range	<none>	50					
Full_Channel_Host_to_Module_BER_Compliance	Host_TxTrf	Float	AMI Range	<none>	9.5e-12					
Full_Channel_Host_to_Module_BER_Compliance	Host_Txtap_filter 0	Tap	AMI Range	Host_TxTap	1.0					

Reference Set: Compliance
Current Set: Compliance

Kit Overview

- Project Name: CEI_28G_VSR
- Interface Name: CEI_28G_VSR
- Operating Frequency: 28.05 GB/s (UI = 35.75 pS)

The CEI 28G-VSR kit defines two schematic sets. Schematic sheets are included for testing a CEI 28G-VSR channel with mated connector to a module board. The masks provided in this kit are given in the 28.05 Gb/s CEI 28G-VSR specification.

- **Compliance** — All compliance host-to-module or module-to-host simulations
- **MCB_HCB_Characterization** — Compliance board network simulations.

For more information about the CEI 28G-VSR channel compliance schematics, transfer net properties, and compliance rules, refer to the document CEI_28G_VSR.pdf that is attached to this example as a supporting file.

References

[1] CEI-28G-VSR Specification. Part of CEI-4.0 specification IA# OIF-CEI-04.0, December 29, 2017.

[2] CEI-25G-LR and CEI-28G-SR Multi-Vendor Interoperability Testing. March, 2012. 2012_OIF_PLL_White_Paper_Feb29.pdf.

[3] CEI-28G:Paving the Way for 100 Gigabit, OIF Forum Whitepaper. John D’Ambrosia, Force10 Networks, David Stauffer, IBM Microelectronics, Chris Cole, Finisar. OIF_CEI-28G_WP_Final.pdf.

[4] IA Title: Common Electrical I/O (CEI) - Electrical and Jitter Interoperability agreements for 6G+ bps, 11G+ bps and 25G+ bps I/O. (IA # OIF-CEI-03.3). OIF_CEI_03.0.pdf, September 1, 2011.

See Also

Serial Link Designer

CEI 56G-LR Compliance Kit

Characterize and validate the performance of a CEI 56G-LR channel design.

CEI 56G-LR is a common electrical interface (CEI) implementation agreement (IA) that supports 56 Gb/s over "Long Reach" (LR) chip-to-chip applications. The CEI-56G-LR Clause is part of the Common Electrical I/O 3.1 Implementation Agreement.

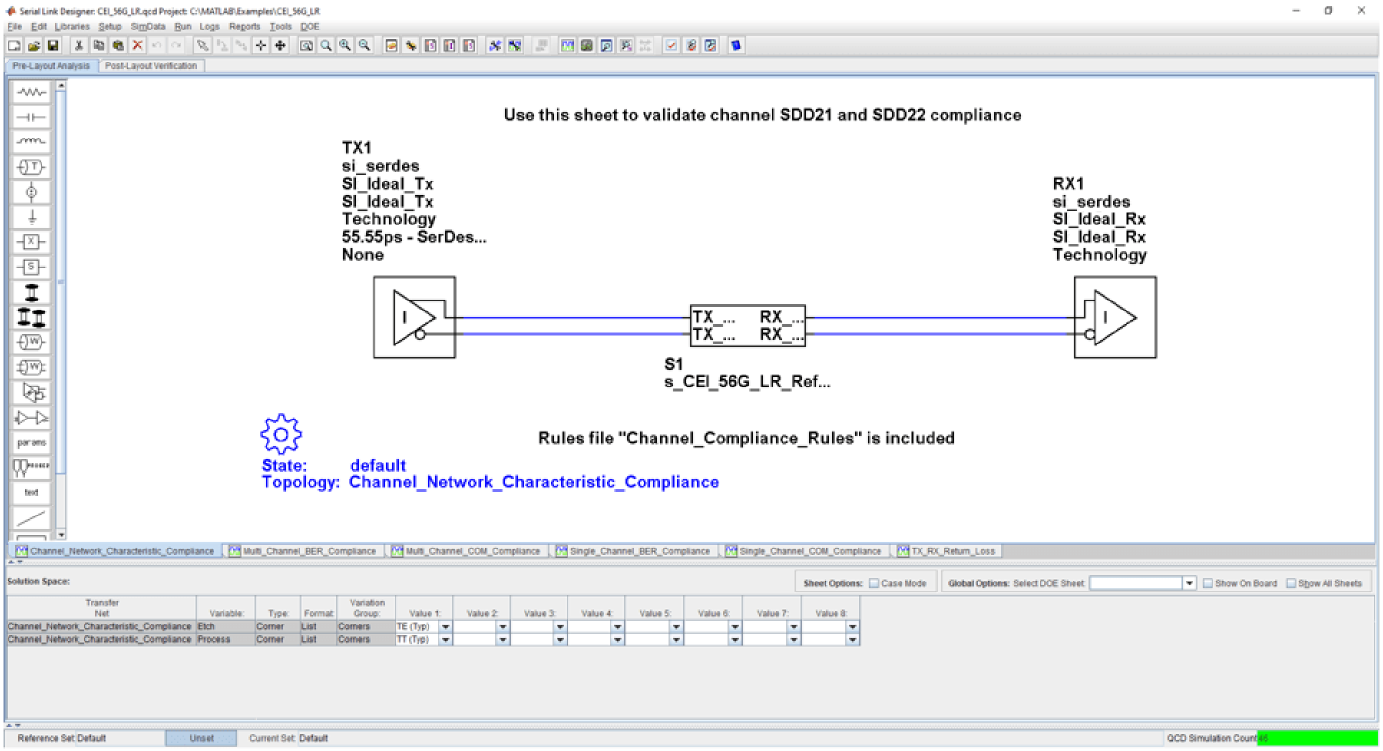
This kit is designed for analysis of a backplane channel design between module boards. The channel model is based on two module boards connected with two mated connectors with PCB trace.

This kit enables you to insert a channel design and characterize and validate its performance using the specification masks to determine if the channel has a high confidence of success. If the channel does not meet the compliance masks, COM, or BER estimates, further investigation or redesign, along with simulation, will need to be performed to determine possible changes to meet compliance.

Open CEI 56G-LR Kit

Open the CEI 56G-LR kit in the **Serial Link Designer** app using the `openSignalIntegrityKit` function.

```
openSignalIntegrityKit("CEI_56G_LR");
```


Serial Link Designer: CEI_56G_LR.qcd Project: C:\MATLAB\Examples\CEI_56G_LR
File Edit Libraries Setup SimData Run Logs Reports Tools DOE
Pre-Layout Analysis
Post-Layout Verification
params
text
Use this sheet to validate channel SDD21 and SDD22 compliance
TX1
si_serdes
SI_Ideal_Tx
SI_Ideal_Tx
Technology
55.55ps - SerDes...
None
RX1
si_serdes
SI_Ideal_Rx
SI_Ideal_Rx
Technology
TX_...
TX_...
RX_...
RX_...
S1
s_CEI_56G_LR_Ref...
Rules file "Channel_Compliance_Rules" is included
State: default
Topology: Channel_Network_Characteristic_Compliance
Channel_Network_Characteristic_Compliance
Multi_Channel_BER_Compliance
Multi_Channel_COM_Compliance
Single_Channel_BER_Compliance
Single_Channel_COM_Compliance
TX_RX_Return_Loss

Solution Space:

Sheet Options: □ Case Mode

Global Options: Select DOE Sheet: □ Show On Board □ Show All Sheets

Transfer Net	Variable:	Type:	Format:	Variation Group:	Value 1:	Value 2:	Value 3:	Value 4:	Value 5:	Value 6:	Value 7:	Value 8:
Channel_Network_Characteristic_Compliance	Etch	Corner	List	Corners	TE (Typ)							
Channel_Network_Characteristic_Compliance	Process	Corner	List	Corners	TT (Typ)							

Reference Set: Default | Unset | Current Set: Default | QCD Simulation Count 46

Kit Overview

- Project Name: CEI_56G_LR
- Interface Name: CEI_56G_LR
- Target Operating Frequency: From 36 Gb/s to 58 Gb/s (UI = 55.55 ps to 34.48 ps)

The CEI 56G-LR kit defines one schematic set. Schematic sheets are included for testing a CEI 56G-LR channel in the form of an S-parameter model. The model represents two mated connectors, a backplane and two plug-in cards. The masks defined for channel losses provided in this kit are given in the CEI 56G-LR specification [1].

- **Default** – Schematic sheets focused on channel characterization and BER compliance.

For more information about the CEI 56G-LR channel compliance schematics, transfer net properties, and compliance rules, refer to the document CEI_56G_LR.pdf that is attached to this example as a supporting file.

References

[1] CEI-56G-LR –PAM4 Long Reach Interface. Contribution Number: OIF2014.380.03. oif2014.380.03-CEI-56G-LR-PAM-4.pdf. June 27, 2016.

[2] Common Electrical I/O (CEI) - Electrical and Jitter Interoperability. IA # OIF-CEI-03.1. February 18, 2014.

See Also

Serial Link Designer

CEI 56G-VSR Compliance Kit

Characterize and validate the performance of a CEI 56G-VSR channel design.

CEI 56G-VSR is a common electrical interface (CEI) implementation agreement (IA) that supports 56 Gb/s over “Very Short Reach” (VSR) optical or electrical chip-to-module applications. The CEI-56G-VSR Clause is part of the Common Electrical I/O 3.1 Implementation Agreement.

The interface relies on PAM4 modulation to increase the bandwidth in 28 GB/s channels. PAM4 modulation can transmit 4-bits per cycle instead of only 2 bits per cycle for NRZ modulation. Theoretically, changing the modulation for signaling will double the bandwidth, so that 28 GB/s compliant channels can run at 56 Gb/s. However, in practice, the design of these interfaces can be challenging when attempting to double the bandwidth using PAM4 modulation.

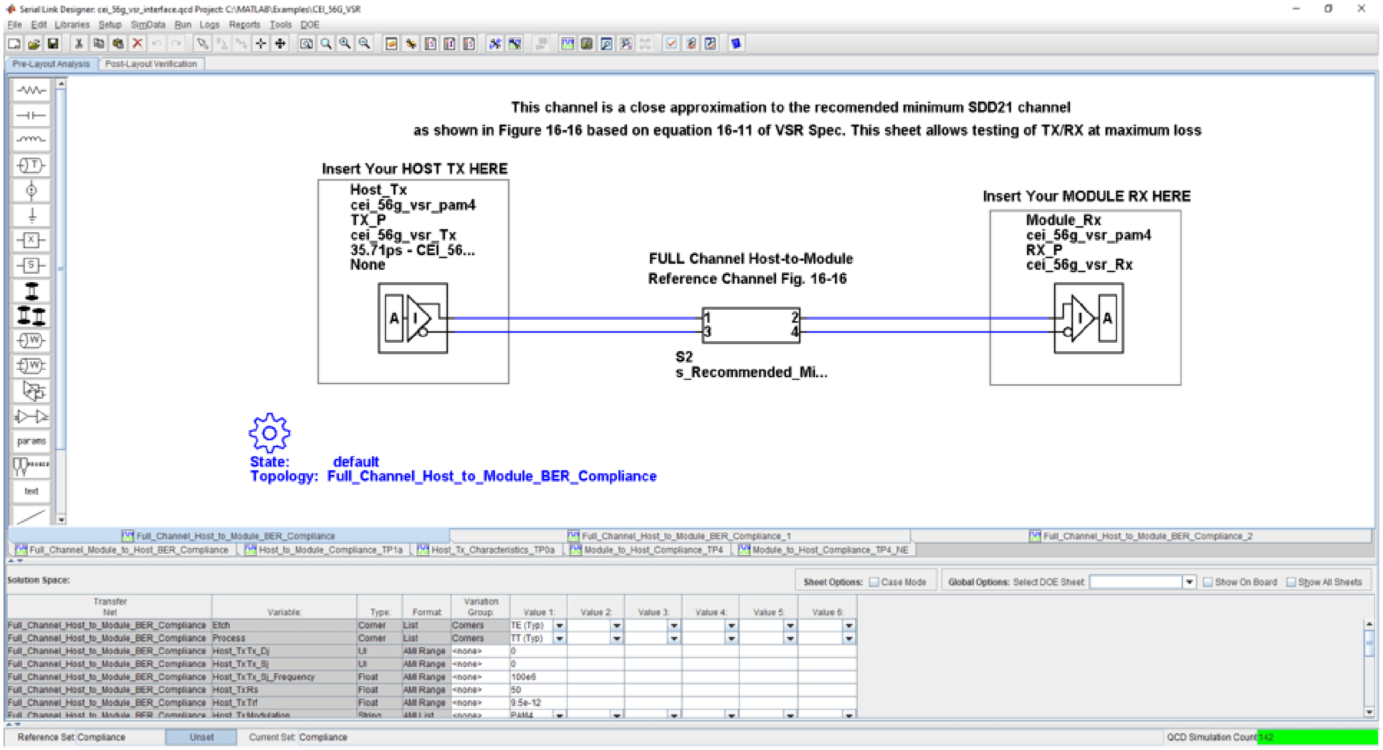
This kit is designed for bidirectional analysis of a host board to an optical module board. The total channel loss at Nyquist or Fb/2 is approximately 10 dB. The VSR channel consists of a host board connected with a mated connector to a module board that represents the interconnection between the transmitting and receiving data across the channel. The kit contains sheets that include the specific host and/or module board design and characterization. Network characterization is set up for insertion and return loss testing to the compliance masks, channel FEXT/NEXT crosstalk is included in multi-channel sheets to measure the effects on BER compliance and RX stress testing.

This kit enables you to insert a channel design and characterize and validate its performance using the specification masks to determine if the channel has a high confidence of success. If the channel does not meet the compliance masks or BER estimates, further investigation or redesign, along with simulation, will need to be performed to determine possible changes to meet compliance. In addition, not all compliance metrics can be simulated and thus will need to be measured in a laboratory environment.

Open CEI 56G-VSR Kit

Open the CEI 56G-VSR kit in the **Serial Link Designer** app using the `openSignalIntegrityKit` function.

```
openSignalIntegrityKit("CEI_56G_VSR");
```


Serial Link Designer: cei_56g_vsr_interface.qcd Project: C:\MATLAB\Examples\CEI_56G_VSR
File Edit Libraries Setup SimData Run Logs Reports Tools DOE
Pre-Layout Analysis
Post-Layout Verification
params
text
Insert Your HOST TX HERE
Host_Tx
cei_56g_vsr_pam4
TX_P
cei_56g_vsr_Tx
35.71ps - CEI_56...
None
A
I
FULL Channel Host-to-Module
Reference Channel Fig. 16-16
1
3
2
4
S2
s_Recommended_Mi...
Insert Your MODULE RX HERE
Module_Rx
cei_56g_vsr_pam4
RX_P
cei_56g_vsr_Rx
I
A
State: default
Topology: Full_Channel_Host_to_Module_BER_Compliance
Full_Channel_Host_to_Module_BER_Compliance
Full_Channel_Host_to_Module_BER_Compliance_1
Full_Channel_Host_to_Module_BER_Compliance_2
Full_Channel_Module_to_Host_BER_Compliance
Host_to_Module_Compliance_TP1a
Host_Tx_Characteristics_TP0a
Module_to_Host_Compliance_TP4
Module_to_Host_Compliance_TP4_NE

This channel is a close approximation to the recomended minimum SDD21 channel as shown in Figure 16-16 based on equation 16-11 of VSR Spec. This sheet allows testing of TX/RX at maximum loss

Solution Space:

Sheet Options: □ Case Mode

Global Options: Select DOE Sheet: ▾ □ Show On Board □ Show All Sheets

Transfer Net	Variable:	Type:	Format:	Variation Group:	Value 1:	Value 2:	Value 3:	Value 4:	Value 5:	Value 6:
Full_Channel_Host_to_Module_BER_Compliance	Etch	Corner	List	Corners	TE (Typ)					
Full_Channel_Host_to_Module_BER_Compliance	Process	Corner	List	Corners	TT (Typ)					
Full_Channel_Host_to_Module_BER_Compliance	Host_TxTx_Dj	UI	AMI Range	<none>	0					
Full_Channel_Host_to_Module_BER_Compliance	Host_TxTx_Sj	UI	AMI Range	<none>	0					
Full_Channel_Host_to_Module_BER_Compliance	Host_TxTx_Sj_Frequency	Float	AMI Range	<none>	100e6					
Full_Channel_Host_to_Module_BER_Compliance	Host_TxRs	Float	AMI Range	<none>	50					
Full_Channel_Host_to_Module_BER_Compliance	Host_TxTrf	Float	AMI Range	<none>	9.5e-12					
Full_Channel_Host_to_Module_BER_Compliance	Host_TxModulation	String	AMI List	<none>	PAM4					

Reference Set: Compliance | Unset | Current Set: Compliance | QCD Simulation Count 142

Kit Overview

- Project Name: CEI_56G_VSR
- Interface Name: CEI_56G_VSR
- Target Operating Frequency: From 36 Gb/s to 58 Gb/s (PAM4 encoding) (UI = 55.55 ps to 34.48 ps)

The CEI 56G-VSR kit defines two schematic sets. Schematic sheets are included for testing a CEI 56G-VSR channel with mated connector to a module board. The masks provided in this kit are given in the CEI 56G-VSR specification [1].

- **Compliance** – All compliance host-to-module or module-to-host simulations
- **MCB_HCB_Characterization** – Compliance board network simulations.

For more information about the CEI 56G-VSR channel compliance schematics, transfer net properties, and compliance rules, refer to the document CEI_56G_VSR.pdf that is attached to this example as a supporting file.

References

[1] CEI-56G: Paving the Way for 100 Gigabit, OIF Forum Whitepaper, OIF_CEI-56G_WP_Final.pdf

See Also

Serial Link Designer

Fibre Channel FC-PI-6 Compliance Kit

Characterize and validate the performance of a Fibre Channel FC-PI-6 channel design.

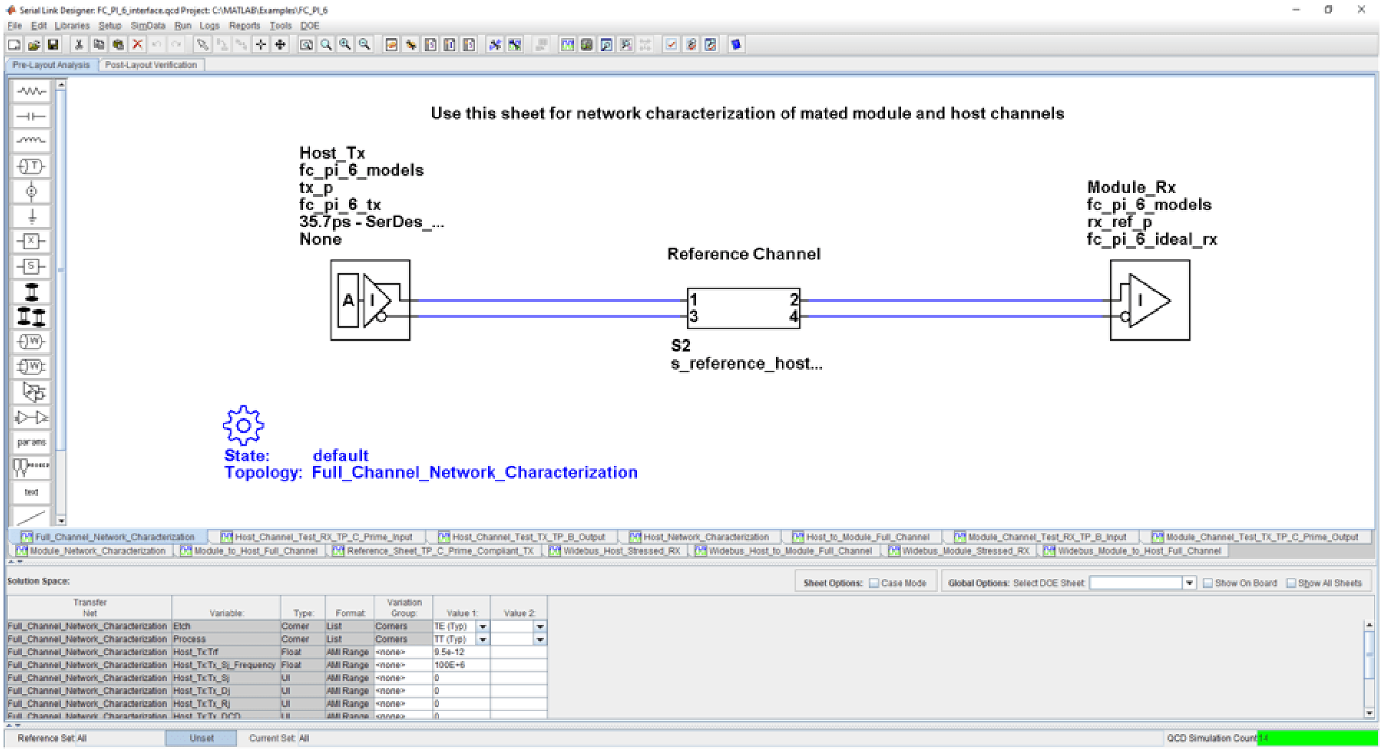
This kit is designed for analysis of a host board and an optical module. The channel consists of a host board connected with a mated connector to a module board which represents the interconnect between the transmitting and receiving 28.05 Gb/s data across the channel. The kit has sheets that represent network characterization for insertion and return loss testing, channel FEXT/NEXT crosstalk to measure BER compliance and RX stress testing.

This kit enables you to insert a channel design and characterize and validate its performance using the specification masks to determine if the channel has a “high confidence” of success. If the channel does not meet the compliance masks or BER estimates, further investigation or redesign, along with simulation, will need to be performed to determine possible changes to meet compliance.

Open FC-PI-6 Kit

Open the FC-PI-6 kit in the **Serial Link Designer** app using the `openSignalIntegrityKit` function.

```
openSignalIntegrityKit("FC_PI_6");
```


Serial Link Designer: FC_PI_6.interface.qcd Project: C:\MATLAB\Examples\FC_PI_6
File Edit Libraries Setup SimData Run Logs Reports Tools DOE
Pre-Layout Analysis
Post-Layout Verification
Use this sheet for network characterization of mated module and host channels
Host_Tx
fc_pi_6_models
tx_p
fc_pi_6_tx
35.7ps - SerDes_...
None
Reference Channel
Module_Rx
fc_pi_6_models
rx_ref_p
fc_pi_6_ideal_rx
S2
s_reference_host...
params
text
Full_Channel_Network_Characterization
Host_Channel_Test_RX_TP_C_Prime_Input
Host_Channel_Test_TX_TP_B_Output
Host_Network_Characterization
Host_to_Module_Full_Channel
Module_Channel_Test_RX_TP_B_Input
Module_Channel_Test_TX_TP_C_Prime_Output
Module_Network_Characterization
Module_to_Host_Full_Channel
Reference_Sheet_TP_C_Prime_Compliant_TX
Widebus_Host_Stressed_RX
Widebus_Host_to_Module_Full_Channel
Widebus_Module_Stressed_RX
Widebus_Module_to_Host_Full_Channel
Sheet Options: □ Case Mode
Global Options: Select DOE Sheet
□ Show On Board
□ Show All Sheets
Reference Set: All
Unset
Current Set: All
QCD Simulation Count 14

State: default
Topology: Full_Channel_Network_Characterization

Solution Space:

Transfer Net	Variable:	Type:	Format:	Variation Group:	Value 1:	Value 2:
Full_Channel_Network_Characterization	Etch	Corner	List	Corners	TE (Typ)	
Full_Channel_Network_Characterization	Process	Corner	List	Corners	TT (Typ)	
Full_Channel_Network_Characterization	Host_Tx:Trf	Float	AMI Range	<none>	9.5e-12	
Full_Channel_Network_Characterization	Host_Tx:Tx_Sj_Frequency	Float	AMI Range	<none>	100E+6	
Full_Channel_Network_Characterization	Host_Tx:Tx_Sj	UI	AMI Range	<none>	0	
Full_Channel_Network_Characterization	Host_Tx:Tx_Dj	UI	AMI Range	<none>	0	
Full_Channel_Network_Characterization	Host_Tx:Tx_Rj	UI	AMI Range	<none>	0	
Full_Channel_Network_Characterization	Host_Tx:Tx_DCD	UI	AMI Range	<none>	0	

Kit Overview

- Project Name: FC_PI_6
- Interface Name: FC_PI_6
- Operating Frequency: 28.05 Gb/s (UI = 35.75 ps)

The FC-PI-6 kit defines three schematic sets. Schematic sheets are included for testing a Fibre Channel FC-PI-6 channel with mated connector to a module board. The masks provided in this kit are given in the Fibre Channel FC-PI-6 specification.

- **ALL** – Contains all project schematics
- **Host_Channel_Simulations** – Host board design schematics
- **Module_Channel_Simulations** – Module board design schematics

For more information about the FC-PI-6 channel compliance schematics, transfer net properties, and compliance rules, refer to the document FibreChannel_FC_PI_6.pdf that is attached to this example as a supporting file.

References

[1] Fibre Channel Physical Interface 6 Rev 1.00 specification. FC-PI-6 Rev 1.00_(13-135v1).pdf. April 26, 2013.

[2] Fibre Channel Methodologies for Signal Quality Specification – MSQS (Rev 0.2). fc_signal_quality_specs_09-263v1.pdf.

[3] IEEE 802.3bj D1.4 Draft Specification. Draft Standard for Ethernet Amendment X:Physical Layer Specifications and Management Parameters for 100 Gb/s Operation Over Backplanes and Copper Cables. P802d3bj_D1p4.pdf. February 21, 2013.

See Also

Serial Link Designer

HMC 15G-SR Compliance Kit

Characterize and validate the performance of an HMC 15G-SR channel design.

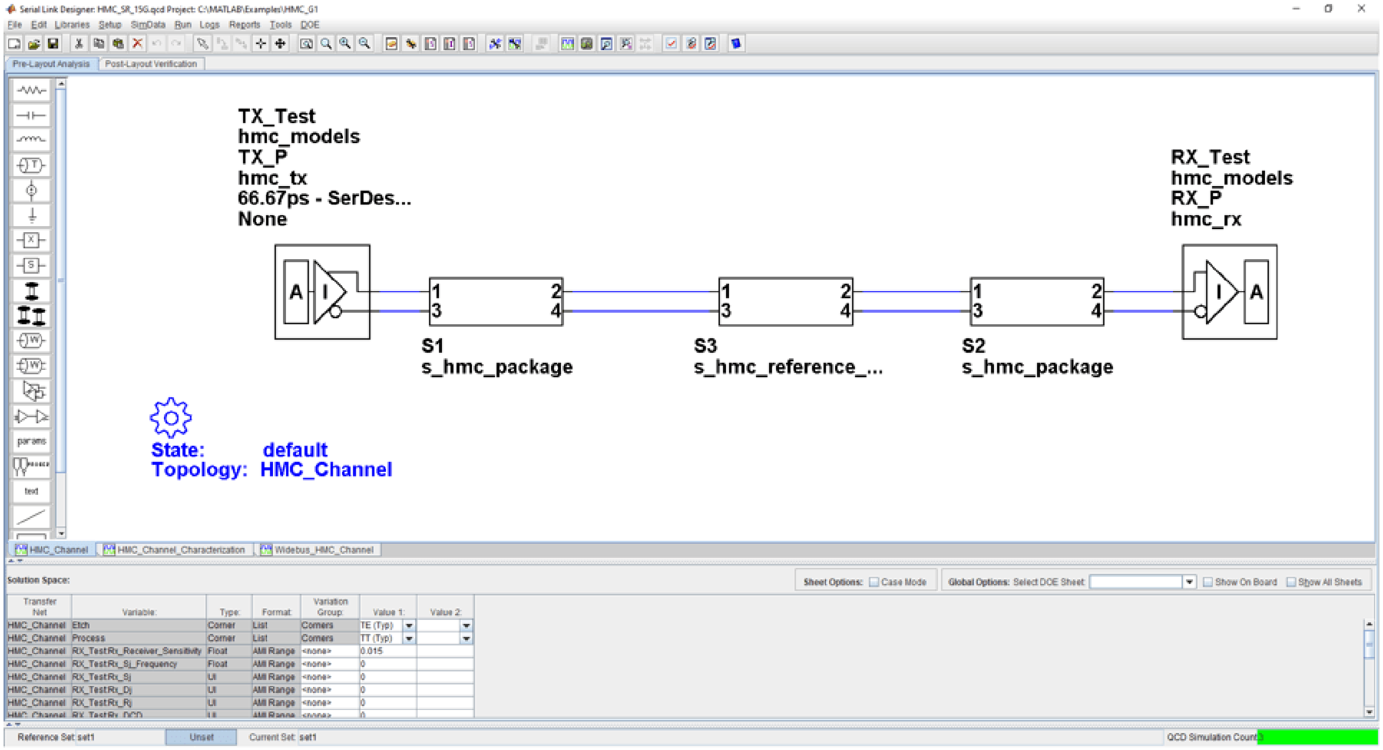
This kit is designed for analysis of an interface between a host ASIC and a Cube. The total channel length between devices is approximately 10 inches. The kit has sheets that represent network characterization for insertion and return loss testing and both single channel and multi-bit channel simulation sheets to measure BER compliance and RX stress testing.

This kit enables you to insert a channel design and characterize and validate its performance using the specification masks to determine if the channel has a “high confidence” of success. If the channel does not meet the compliance masks or BER estimates, further investigation or redesign, along with simulation, will need to be performed to determine possible changes to meet compliance.

Open HMC 15G-SR Kit

Open the HMC 15G-SR kit in the **Serial Link Designer** app using the `openSignalIntegrityKit` function.

```
openSignalIntegrityKit("HMC_G1");
```



Serial Link Designer: HMC_SR_15G.qcd Project: C:\MATLAB\Examples\HMC_G1

State: default
Topology: HMC_Channel

Solution Space:

Sheet Options: □ Case Mode

Global Options: Select DOE Sheet: □ Show On Board □ Show All Sheets

Transfer Net	Variable:	Type:	Format:	Variation Group:	Value 1:	Value 2:
HMC_Channel	Etch	Corner	List	Corners	TE (Typ)	
HMC_Channel	Process	Corner	List	Corners	TT (Typ)	
HMC_Channel	RX_Test.Rx_Receiver_Sensitivity	Float	AMI Range	<none>	0.015	
HMC_Channel	RX_Test.Rx_Sj_Frequency	Float	AMI Range	<none>	0	
HMC_Channel	RX_Test.Rx_Sj	UI	AMI Range	<none>	0	
HMC_Channel	RX_Test.Rx_Dj	UI	AMI Range	<none>	0	
HMC_Channel	RX_Test.Rx_Rj	UI	AMI Range	<none>	0	
HMC_Channel	RX_Test.Rx_DCD	UI	AMI Range	<none>	0	

Reference Set: set1 Unset Current Set: set1 QCD Simulation Count: 3

Kit Overview

- Project Name: HMC_15G_SR
- Interface Name: HMC_15G_SR
- Operating Frequency: 15 Gb/s (UI = 66.67 ps) default setting. 10 Gb/s and 12.5 Gb/s can be selected.

The HMC 15G-SR kit defines one schematic set. The masks provided in this kit are given in the 15 Gb/s HMC-15G-SR specification.

- **Set1** — Contains all project schematics

For more information about the HMC 15G-SR channel compliance schematics, transfer net properties, and compliance rules, refer to the document HMC_15G_SR.pdf that is attached to this example as a supporting file.

References

[1] HMC Specification 1.0. hmc_gen2_hmcc_1.fm - Rev. 1.0 1/13 EN. HMC_Specification_1_0.pdf.

See Also

Serial Link Designer

HMC 30G-VSR Compliance Kit

Characterize and validate the performance of a hybrid memory cube (HMC) 30G-VSR channel design.

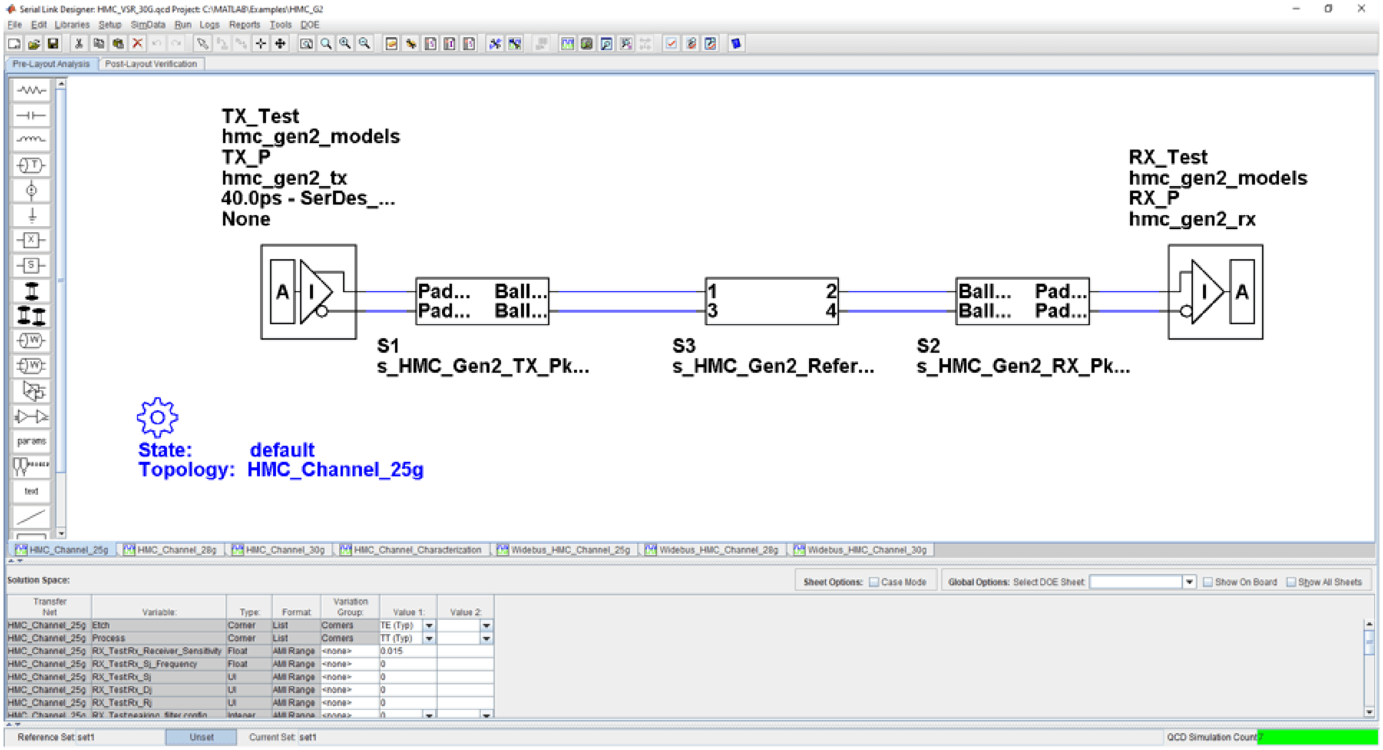
This kit is designed for analysis of an interface between a host ASIC and a Cube. The total channel length between devices is approximately 10 inches. The kit has sheets that represent network characterization for insertion and return loss testing and both single channel and multi-bit channel simulation sheets to measure BER compliance and RX stress testing. The kit includes schematics for testing 30 Gb/s, 28 Gb/s and 25 Gb/s operation including all applicable masks scaled for the representative bit rate.

This kit enables you to insert a channel design and characterize and validate its performance using the specification masks to determine if the channel has a “high confidence” of success. If the channel does not meet the compliance masks or BER estimates, further investigation or redesign, along with simulation, will need to be performed to determine possible changes to meet compliance.

Open HMC 30G-VSR Kit

Open the HMC 30G-VSR kit in the **Serial Link Designer** app using the `openSignalIntegrityKit` function.

```
openSignalIntegrityKit("HMC_G2");
```



Serial Link Designer: HMC_VSR_30G.qcd Project: C:\MATLAB\Examples\HMC_G2

State: default
Topology: HMC_Channel_25g

Solution Space:

Sheet Options: □ Case Mode

Global Options: Select DOE Sheet: □ Show On Board □ Show All Sheets

Transfer Net	Variable:	Type:	Format:	Variation Group:	Value 1:	Value 2:
HMC_Channel_25g	Etch	Corner	List	Corners	TE (Typ)	
HMC_Channel_25g	Process	Corner	List	Corners	TT (Typ)	
HMC_Channel_25g	RX_TestRx_Receiver_Sensitivity	Float	AMI Range	<none>	0.015	
HMC_Channel_25g	RX_TestRx_Sj_Frequency	Float	AMI Range	<none>	0	
HMC_Channel_25g	RX_TestRx_Sj	UI	AMI Range	<none>	0	
HMC_Channel_25g	RX_TestRx_Dj	UI	AMI Range	<none>	0	
HMC_Channel_25g	RX_TestRx_Rj	UI	AMI Range	<none>	0	
HMC_Channel_25g	RX_Testpeaking_filter config	Integer	AMI Range	<none>	0	

Reference Set: set1

Current Set: set1

Kit Overview

- Project Name: HMC_G2
- Interface Name: HMC_VSR_30G
- Operating Frequency: 25 Gb/s, 28 Gb/s, and 30 Gb/s (UI = 33.33ps min)

The HMC 30G-VSR kit defines one schematic set. The masks provided in this kit are given in the HMC-30G-VSR_HMCC_Rev2.0_Public.pdf specification.

- **Set1** — Contains all project schematics

For more information about the HMC 30G-VSR channel compliance schematics, transfer net properties, and compliance rules, refer to the document HMC_30G_VSR.pdf that is attached to this example as a supporting file.

References

[1] HMC Specification 1.0. HMC-30G-VSR_HMCC_Specification_Rev2.0_Public.pdf.

See Also

Serial Link Designer

MIPI D-PHY Serial Link Compliance Kit

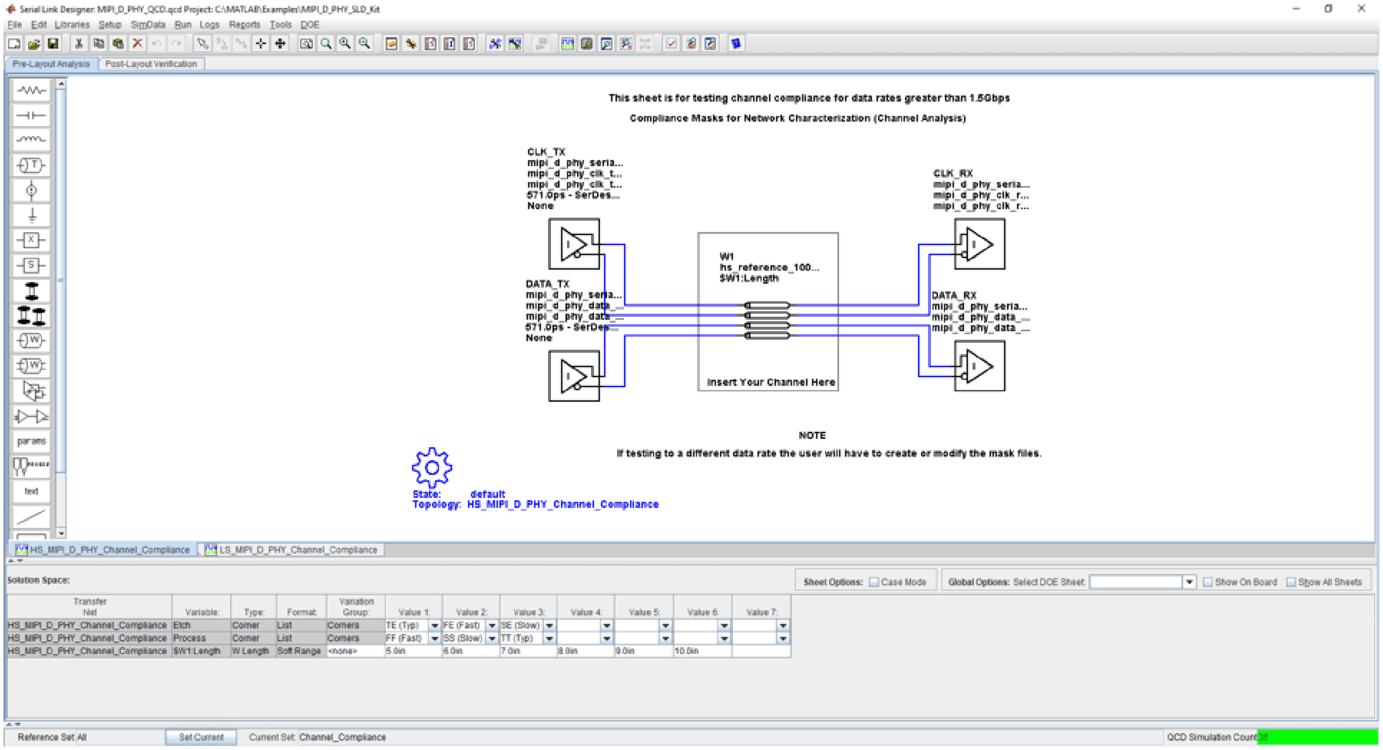
Test the compliance of a channel to the MIPI D-PHY specification using **Serial Link Designer**.

This kit is designed to test MIPI D-PHY channel compliance only. The MIPI D-PHY specification requires channels to meet various mixed mode insertion and return loss characteristics. This kit allows the user to design their channel for compliance with the D-PHY specification. To evaluate the source synchronous timing of the interface using the compliant channel design, use the “MIPI D-PHY Parallel Link Compliance Kit” on page 11-82. In addition, this kit can be used to test the required transmitter and receiver return loss masks.

Open MIPI D-PHY Kit

Open the MIPI D-PHY kit in the **Serial Link Designer** app using the `openSignalIntegrityKit` function.

```
openSignalIntegrityKit("MIPI_D_PHY_SLD_Kit");
```


Serial Link Designer: MIPI_D_PHY_QCD.qcd Project: C:\MATLAB\Examples\MIPI_D_PHY_SLD_Kit

This sheet is for testing channel compliance for data rates greater than 1.5Gbps

Compliance Masks for Network Characterization (Channel Analysis)

NOTE

If testing to a different data rate the user will have to create or modify the mask files.

State: default
Topology: HS_MIPI_D_PHY_Channel_Compliance

Solution Space:

Sheet Options: □ Case Mode

Global Options: Select DOE Sheet: □ Show On Board □ Show All Sheets

Transfer Net	Variable:	Type:	Format:	Variation Group:	Value 1:	Value 2:	Value 3:	Value 4:	Value 5:	Value 6:	Value 7:
HS_MIPI_D_PHY_Channel_Compliance	Etch	Corner	List	Corners	TE (Typ)	FE (Fast)	SE (Slow)				
HS_MIPI_D_PHY_Channel_Compliance	Process	Corner	List	Corners	FF (Fast)	SS (Slow)	TT (Typ)				
HS_MIPI_D_PHY_Channel_Compliance	$W1:Length	W Length	Soft Range	<none>	5.0in	6.0in	7.0in	8.0in	9.0in	10.0in	

Reference Set: All

Current Set: Channel_Compliance

QCD Simulation Count: 36

Kit Overview

- Project Name: MIPI_D_PHY_SLD_Kit
- Interface Name: MIPI_D_PHY_SLD
- Target operating frequencies: 0.9 Gb/s (1.11 ns), 1.5 Gb/s (667 ps), and 1.75 Gb/s (571 ps).

The MIPI D-PHY kit defines three schematic set for each interface. Reference IBIS TX and RX models are included as place holders for compliance testing.

- **ALL** — All sheets in the project

- **Channel_Compliance** — Testing MIPI D-PHY channel compliance
- **TX_RX_Compliance** — Testing TX and RX compliance

For more information about the MIPI D-PHY channel compliance schematics, transfer net properties, and compliance rules, refer to the document MIPI_D_Phy_SLD.pdf that is attached to this example as a supporting file.

References

[1] MIPI Alliance Specification for D-PHY. Version 2.0, 1 August, 2014.

See Also

Serial Link Designer

Related Examples

- “MIPI D-PHY Parallel Link Compliance Kit” on page 11-82

MIPI M-PHY Compliance Kit

Characterize and validate the performance of a MIPI M-PHY channel design.

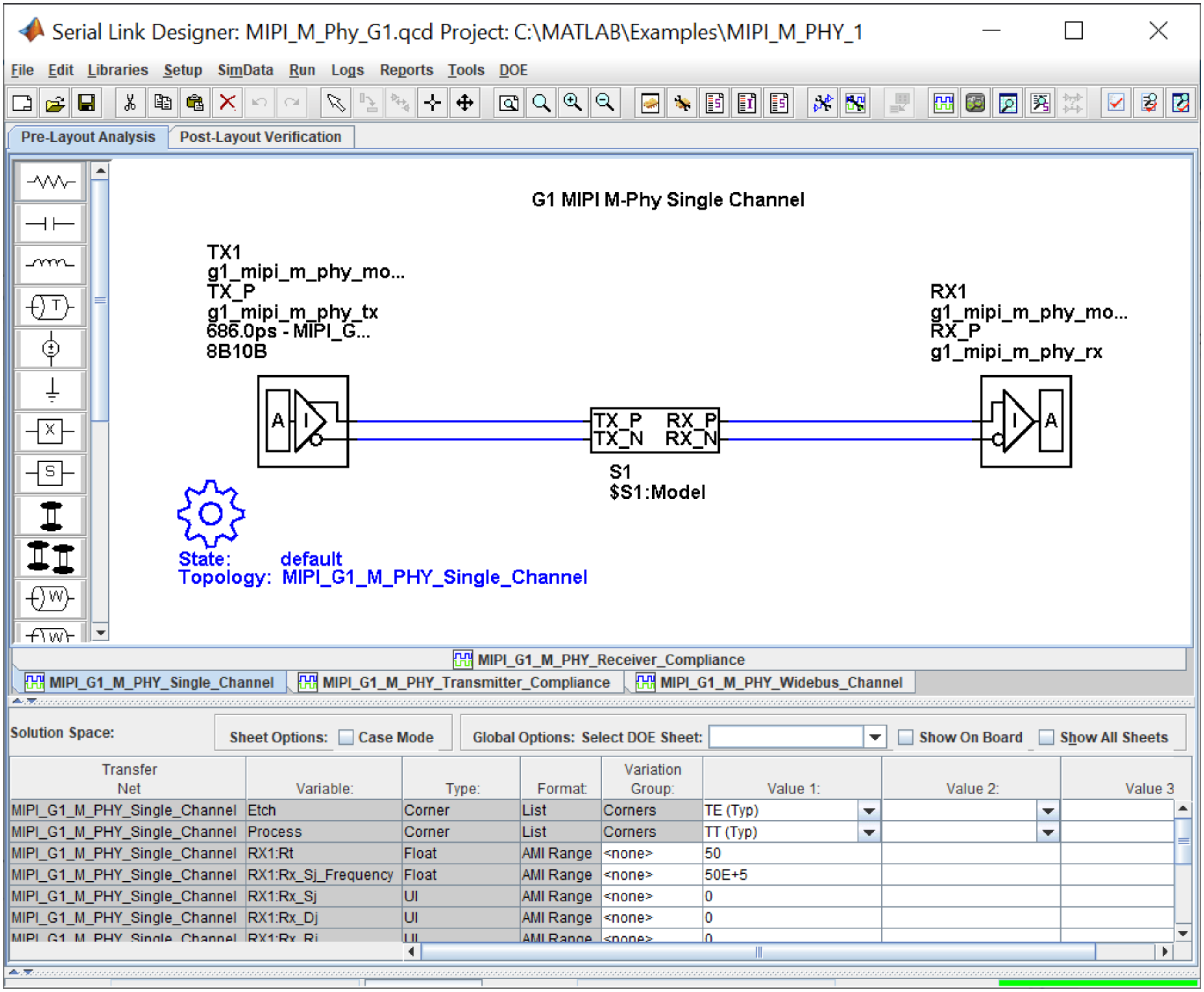
This kit is designed for an interface between system devices with up to 25 inches of PCB etch and one connector. The kit includes IBIS-AMI TX and RX models for reference and compliance testing.

This kit enables you to insert a channel design and characterize and validate its performance using the specification masks to determine if the channel has a “high confidence” of success. If the channel does not meet the compliance masks or BER estimates, further investigation or redesign, along with simulation, will need to be performed to determine possible changes to meet compliance.

Open MIPI M-PHY Kit

Open the MIPI M-PHY kit in the **Serial Link Designer** app using the `openSignalIntegrityKit` function.

```
openSignalIntegrityKit("MIPI_M_PHY");
```


Serial Link Designer: MIPI_M_Phy_G1.qcd Project: C:\MATLAB\Examples\MIPI_M_PHY_1
File Edit Libraries Setup SimData Run Logs Reports Tools DOE
Pre-Layout Analysis
Post-Layout Verification
G1 MIPI M-Phy Single Channel
TX1
g1_mipi_m_phy_mo...
TX_P
g1_mipi_m_phy_tx
686.0ps - MIPI_G...
8B10B
RX1
g1_mipi_m_phy_mo...
RX_P
g1_mipi_m_phy_rx
A
I
TX_P
TX_N
RX_P
RX_N
S1
$S1:Model
State: default
Topology: MIPI_G1_M_PHY_Single_Channel
MIPI_G1_M_PHY_Receiver_Compliance
MIPI_G1_M_PHY_Single_Channel
MIPI_G1_M_PHY_Transmitter_Compliance
MIPI_G1_M_PHY_Widebus_Channel

Solution Space:

Sheet Options: □ Case Mode

Global Options: Select DOE Sheet: □ Show On Board □ Show All Sheets

Transfer Net	Variable:	Type:	Format:	Variation Group:	Value 1:	Value 2:	Value 3
MIPI_G1_M_PHY_Single_Channel	Etch	Corner	List	Corners	TE (Typ)		
MIPI_G1_M_PHY_Single_Channel	Process	Corner	List	Corners	TT (Typ)		
MIPI_G1_M_PHY_Single_Channel	RX1:Rt	Float	AMI Range	<none>	50		
MIPI_G1_M_PHY_Single_Channel	RX1:Rx_Sj_Frequency	Float	AMI Range	<none>	50E+5		
MIPI_G1_M_PHY_Single_Channel	RX1:Rx_Sj	UI	AMI Range	<none>	0		
MIPI_G1_M_PHY_Single_Channel	RX1:Rx_Dj	UI	AMI Range	<none>	0		
MIPI_G1_M_PHY_Single_Channel	RX1:Rx_Rj	UI	AMI Range	<none>	0		

Kit Overview

- Project Name: MIPI_M_PHY
- MIPI_M_Phy_G1 interface: Target operating frequencies of 1.25 Gb/s (800 ps) and 1.46 Gb/s (680 ps)
- MIPI_M_Phy_G2 interface: Target operating frequencies of 255 Gb/s (400 ps) and 2.92 Gb/s (343 ps)
- MIPI_M_Phy_G3 interface: Target operating frequencies of 4.99 Gb/s (200 ps) and 5.83 Gb/s (172 ps)
- MIPI_M_Phy_G4 interface: Target operating frequencies of 9.98 Gb/s (100 ps) and 11.66 Gb/s (86 ps)

The MIPI M-PHY kit defines one schematic set for each interface.

- **MIPI** — Used for all compliance testing

For more information about the MIPI M-PHY channel compliance schematics, transfer net properties, and compliance rules, refer to the document MIPI_M_PHY.pdf that is attached to this example as a supporting file.

References

[1] MIPI Alliance Specification for M-PHY. Version 4.0, 27 Apr-2015.

[2] IEEE 802.3bj-2014 (CJPAT and CRPAT reference). 802.3bj-2014.pdf.

See Also

Serial Link Designer

PCIe-2 Compliance Kit

Test the compliance of simulation models and topologies to the PCI Express generation 2 (PCIe-2) specification.

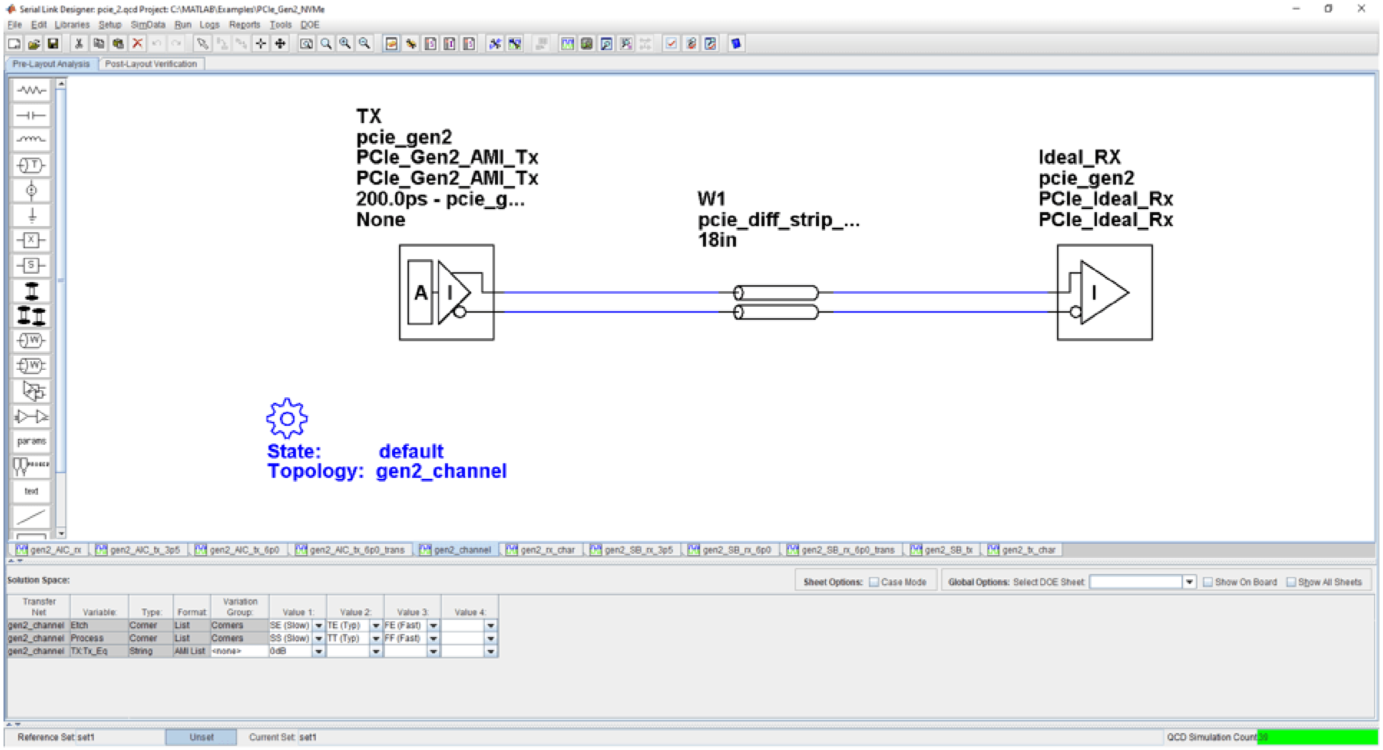
This PCIe signal integrity kit includes all the transfer nets, topologies, generic buffer models and compliance rules for a PCIe-2 high-speed SerDes interface. This includes PCIe-2 technology IBIS-AMI models for the SerDes transmitter and receiver, PCIe-2 compliance masks and transfer nets preconfigured for TX and RX characterization that are customizable for a PCIe-2 embedded channel.

This kit enables you to insert a channel design and characterize and validate its performance using the specification masks to determine if the channel has a high confidence of success. If the channel does not meet the compliance masks or BER estimates, further investigation or redesign, along with simulation, will need to be performed to determine possible changes to meet compliance.

Open PCIe-2 Kit

Open the PCIe-2 kit in the **Serial Link Designer** app using the `openSignalIntegrityKit` function.

```
openSignalIntegrityKit("PCIe_Gen2_NVMe");
```



Serial Link Designer: pcie_2.qcd Project: C:\MATLAB\Examples\PCIe_Gen2_NVMe

State: default
Topology: gen2_channel

Solution Space:

Sheet Options: □ Case Mode

Global Options: Select DOE Sheet: □ Show On Board □ Show All Sheets

Transfer Net	Variable:	Type:	Format:	Variation Group:	Value 1:	Value 2:	Value 3:	Value 4:
gen2_channel	Etch	Corner	List	Corners	SE (Slow)	TE (Typ)	FE (Fast)	
gen2_channel	Process	Corner	List	Corners	SS (Slow)	TT (Typ)	FF (Fast)	
gen2_channel	TX:Tx_Eq	String	AMI List	<none>	0dB			

Reference Set: set1 Unset Current Set: set1 QCD Simulation Count: 30

Kit Overview

- Project Name: PCIe_Gen2_NVMe
- Interface Name: PCIe_Gen2
- Target Operating Frequency: 5.0 Gb/s (UI = 200 ps)

For more information about the PCIe-2 channel compliance schematics, transfer net properties and compliance rules, refer to the document PCIe_gen2.pdf that is attached to this example as a supporting file.

See Also

Serial Link Designer

PCIe-3 Compliance Kit

Test the compliance of simulation models and topologies to the PCI Express generation 3 (PCIe-3) specification.

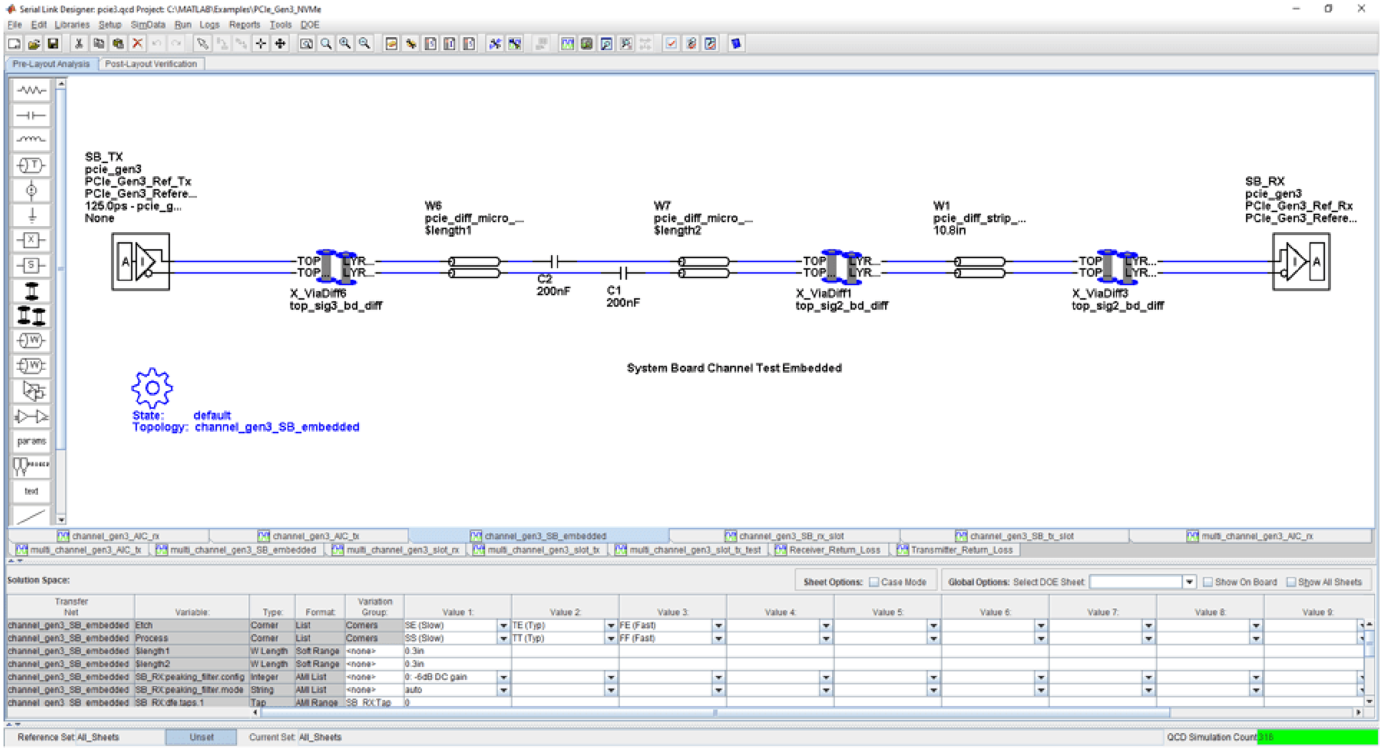
The PCIe-3 signal integrity kit includes all the transfer nets, topologies, generic buffer models and compliance rules for a PCIe-3 high-speed SerDes interface. This includes PCIe-3 technology IBIS-AMI models for the SerDes transmitter and receiver, PCIe-3 compliance masks and transfer nets preconfigured for TX and RX characterization that are customizable for a specific PCIe-3 add-in card (AIC), system board (SB), and PCIe-3 embedded channel.

This kit enables you to insert a channel design and characterize and validate its performance using the specification masks to determine if the channel has a high confidence of success. If the channel does not meet the compliance masks or BER estimates, further investigation or redesign, along with simulation, will need to be performed to determine possible changes to meet compliance.

Open PCIe-3 Kit

Open the PCIe-3 kit in the **Serial Link Designer** app using the `openSignalIntegrityKit` function.

```
openSignalIntegrityKit("PCIe_Gen3_NVMe");
```



Solution Space:

Sheet Options: □ Case Mode

Global Options: Select DOE Sheet □ Show On Board □ Show All Sheets

Transfer Net	Variable:	Type:	Format:	Variation Group:	Value 1:	Value 2:	Value 3:	Value 4:	Value 5:	Value 6:	Value 7:	Value 8:	Value 9:
channel_gen3_SB_embedded	Etch	Corner	List	Corners	SE (Slow)	TE (Typ)	FE (Fast)						
channel_gen3_SB_embedded	Process	Corner	List	Corners	SS (Slow)	TT (Typ)	FF (Fast)						
channel_gen3_SB_embedded	$length1	W Length	Soft Range	<none>	0.3in								
channel_gen3_SB_embedded	$length2	W Length	Soft Range	<none>	0.3in								
channel_gen3_SB_embedded	SB_RX:peaking_filter.config	Integer	AMI List	<none>	0: -6dB DC gain								
channel_gen3_SB_embedded	SB_RX:peaking_filter.mode	String	AMI List	<none>	auto								
channel_gen3_SB_embedded	SB_RX:dfe.taps.1	Tap	AMI Range	SB_RX:Tap	0								

Reference Set: All_Sheets Unset Current Set: All_Sheets

QCD Simulation Count: 316

Kit Overview

- Project Name: PCIe_Gen3_NVMe
- Interface Name: PCIe_Gen3
- Target Operating Frequency: 8.0 Gb/s, 4.0 GHz (Nyquist) (UI = 125 ps)

The PCIe-3 kit defines four schematic sets:

- **All_Sheets**: All schematic sheets
- **AIC**: Schematic sheets for add-in card design
- **SB_Slot**: Schematic sheets for system board with slot design
- **SB_Emb**: Schematic sheets for system board embedded design

For more information about the PCIe-3 channel compliance schematics, transfer net properties and compliance rules, refer to the document PCIe_gen3.pdf that is attached to this example as a supporting file.

See Also

Serial Link Designer

PCIe-4 Compliance Kit

Test the compliance of simulation models and topologies to the PCI Express generation 4 (PCIe-4) specification.

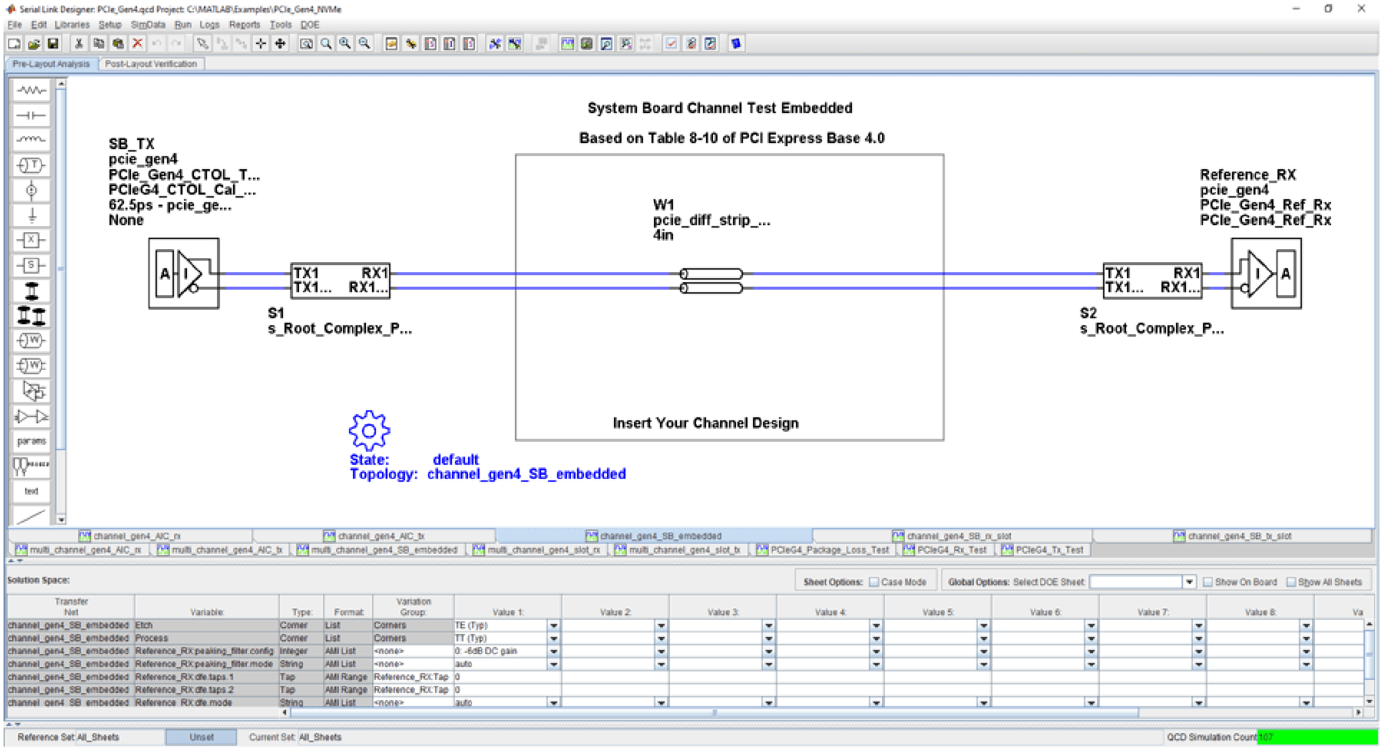
This PCIe compliance signal integrity kit includes all the transfer nets, topologies, generic buffer models and compliance rules for a PCIe-4 high-speed SerDes interface. This includes PCIe-4 technology IBIS-AMI models for the SerDes transmitter and receiver, PCIe-4 compliance masks and transfer nets preconfigured for TX and RX characterization that are customizable for a PCIe-4 embedded channel.

This kit enables you to insert a channel design and characterize and validate its performance using the specification masks to determine if the channel has a high confidence of success. If the channel does not meet the compliance masks or BER estimates, further investigation or redesign, along with simulation, will need to be performed to determine possible changes to meet compliance.

Open PCIe-4 Kit

Open the PCIe-4 kit in the **Serial Link Designer** app using the `openSignalIntegrityKit` helper function.

```
openSignalIntegrityKit("PCIe_Gen4_NVMe");
```



Solution Space:

Transfer Net:	Variable:	Type:	Format:	Variation Group:	Value 1:	Value 2:	Value 3:	Value 4:	Value 5:	Value 6:	Value 7:	Value 8:	Va
channel_gen4_SB_embedded	Etch	Corner	List	Corners	TE (Typ)								
channel_gen4_SB_embedded	Process	Corner	List	Corners	TT (Typ)								
channel_gen4_SB_embedded	Reference_RX.peaking_filter.config	Integer	AMI List	<none>	0: -6dB DC gain								
channel_gen4_SB_embedded	Reference_RX.peaking_filter.mode	String	AMI List	<none>	auto								
channel_gen4_SB_embedded	Reference_RX.dfe.taps.1	Tap	AMI Range	Reference_RX.Tap	0								
channel_gen4_SB_embedded	Reference_RX.dfe.taps.2	Tap	AMI Range	Reference_RX.Tap	0								
channel_gen4_SB_embedded	Reference_RX.dfe.mode	String	AMI List	<none>	auto								

Kit Overview

- Project Name: PCIe_Gen4_NVMe
- Interface Name: PCIe_Gen4
- Target Operating Frequency: 16.0 Gb/s, 8.0 GHz (Nyquist) (62.5ps)

The PCIe-4 kit defines five schematic sets:

- **All_Sheets**: All schematic sheets
- **AIC**: Add-In Card schematics only
- **SB_EMB**: System Board embedded schematics only
- **SB_Slot**: Slot configuration schematics
- **Tx_and_Rx_Tests**: Return loss and package loss

For more information about the PCIe-4 channel compliance schematics, transfer net properties and compliance rules, refer to the document PCIe_gen4.pdf that is attached to this example as a supporting file.

See Also

Serial Link Designer

PCIe-5 Compliance Kit

Test the compliance of simulation models and topologies to the PCI Express generation 5 (PCIe-5) specification.

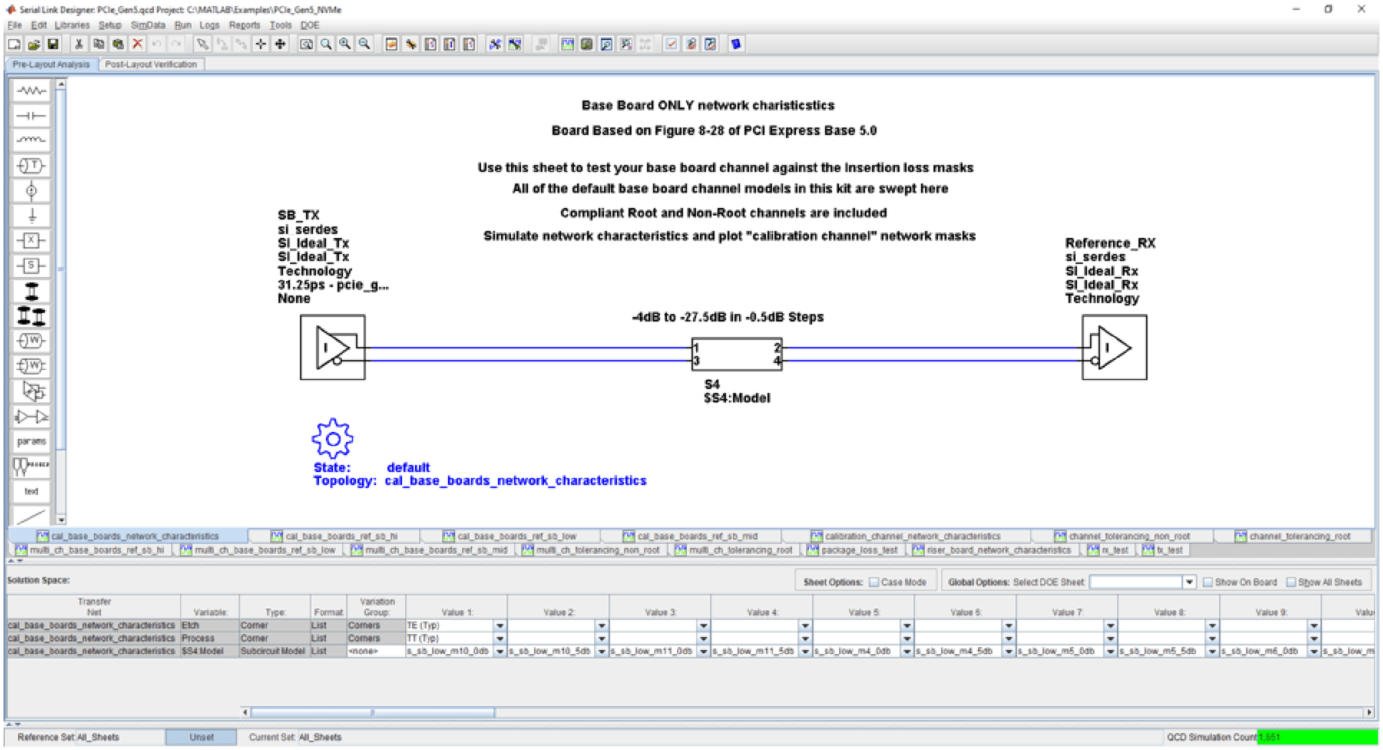
This PCIe compliance signal integrity kit includes all the transfer nets, topologies, generic buffer models and compliance rules for a PCIe-5 high-speed SerDes interface. This includes PCIe-5 technology IBIS-AMI models for the SerDes transmitter and receiver, PCIe-5 compliance masks and transfer nets preconfigured for TX and RX characterization that are customizable for a PCIe-5 embedded channel.

This kit enables you to insert a channel design and characterize and validate its performance using the specification masks to determine if the channel has a high confidence of success. If the channel does not meet the compliance masks or BER estimates, further investigation or redesign, along with simulation, will need to be performed to determine possible changes to meet compliance.

Open PCIe-5 Kit

Open the PCIe-5 kit in the **Serial Link Designer** app using the `openSignalIntegrityKit` helper function.

```
openSignalIntegrityKit("PCIe_Gen5_NVMe");
```


Serial Link Designer: PCIe_Gen5.qcd Project: C:\MATLAB\Examples\PCIe_Gen5_NVMe
File Edit Libraries Setup SimData Run Logs Reports Tools DOE
Pre-Layout Analysis
Post-Layout Verification
params
text
SB_TX
si_serdes
SI_Ideal_Tx
SI_Ideal_Tx
Technology
31.25ps - pcie_g...
None
Reference_RX
si_serdes
SI_Ideal_Rx
SI_Ideal_Rx
Technology
1
3
2
4
S4
$S4:Model
cal_base_boards_network_characteristics
cal_base_boards_ref_sb_hi
cal_base_boards_ref_sb_low
cal_base_boards_ref_sb_mid
calibration_channel_network_characteristics
channel_tolerancing_non_root
channel_tolerancing_root
multi_ch_base_boards_ref_sb_hi
multi_ch_base_boards_ref_sb_low
multi_ch_base_boards_ref_sb_mid
multi_ch_tolerancing_non_root
multi_ch_tolerancing_root
package_loss_test
riser_board_network_characteristics
rx_test
tx_test
Sheet Options: □ Case Mode
Global Options: Select DOE Sheet
□ Show On Board
□ Show All Sheets
Reference Set: All_Sheets
Unset
Current Set: All_Sheets
QCD Simulation Count 1,651

Base Board ONLY network charisticstics

Board Based on Figure 8-28 of PCI Express Base 5.0

Use this sheet to test your base board channel against the insertion loss masks

All of the default base board channel models in this kit are swept here

Compliant Root and Non-Root channels are included

Simulate network characteristics and plot "calibration channel" network masks

-4dB to -27.5dB in -0.5dB Steps

State: default
Topology: cal_base_boards_network_characteristics

Solution Space:

Transfer Net	Variable:	Type:	Format:	Variation Group:	Value 1:	Value 2:	Value 3:	Value 4:	Value 5:	Value 6:	Value 7:	Value 8:	Value 9:	Valu
cal_base_boards_network_characteristics	Etch	Corner	List	Corners	TE (Typ)									
cal_base_boards_network_characteristics	Process	Corner	List	Corners	TT (Typ)									
cal_base_boards_network_characteristics	$S4 Model	Subcircuit Model	List	<none>	s_sb_low_m10_0db	s_sb_low_m10_5db	s_sb_low_m11_0db	s_sb_low_m11_5db	s_sb_low_m4_0db	s_sb_low_m4_5db	s_sb_low_m5_0db	s_sb_low_m5_5db	s_sb_low_m6_0db	s_sb_low_m

Kit Overview

- Project Name: PCIe_Gen5_NVMe
- Interface Name: PCIe_Gen5
- Target Operating Frequency: 32.0 Gb/s; 16.0 GHz (Nyquist) (UI = 31.25 ps)

The PCIe-5 kit defines five schematic sets:

- **All_Sheets**: All schematic sheets
- **Cal_Channel_Ref_Design_32Gbps**: Base Specification Reference Design for 32 Gbps
- **Channel_Tolerancing**: Calibration Channel Stressed RX Testing
- **Compliance_Board_Testing**: Network Characteristics for Calibration Channel Models and Reference Design Models
- **Tx_and_Rx_Pkg_Tests**: Testing of Tx and Rx Characteristics and Package Model

For more information about the PCIe-5 channel compliance schematics, transfer net properties and compliance rules, refer to the document PCIe_gen5.pdf that is attached to this example as a supporting file.

See Also

Serial Link Designer

QSFP+ Compliance Kit

Test the channel design of a host board for compliance to the QSFP+ specification.

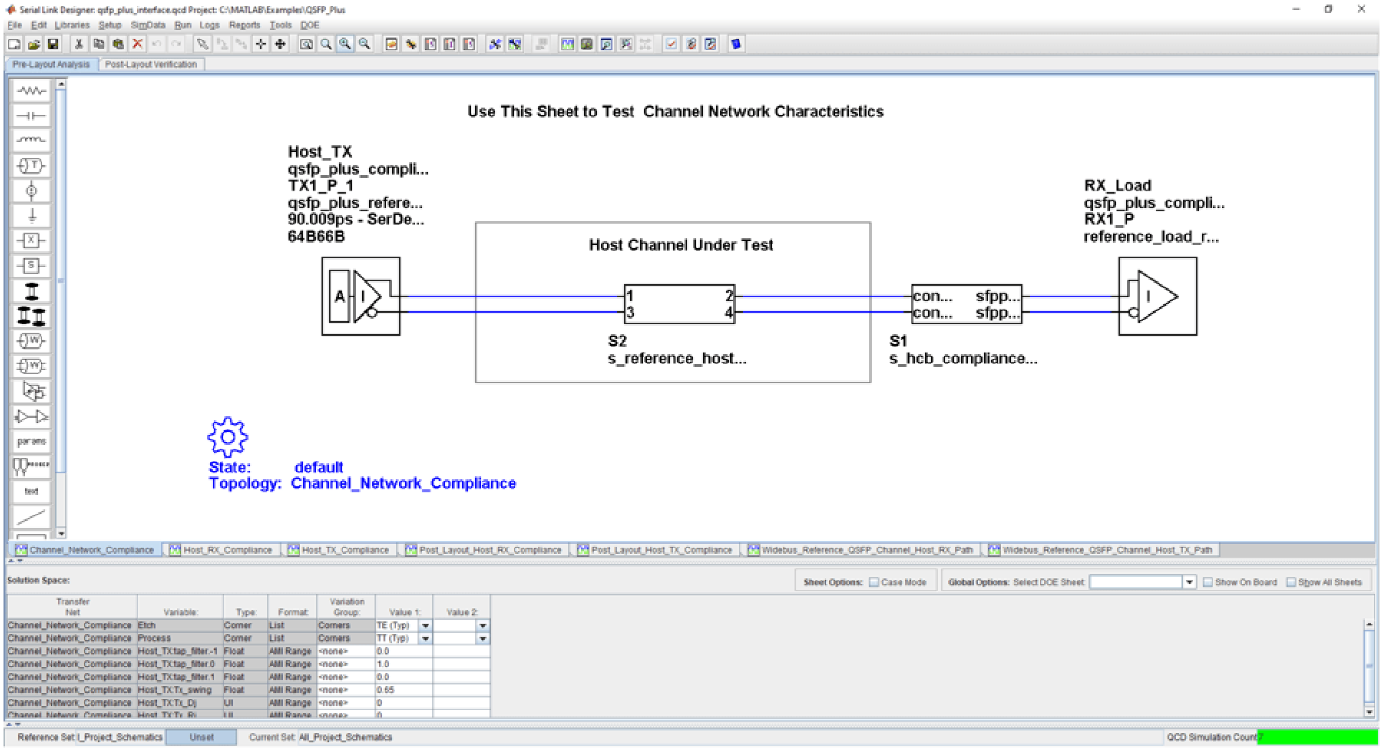
A QSFP+ link is made up of four SFP+ channels that are synchronized within the receiving module to support an aggregate 40 Gb/s link.

This kit is designed for compliance of the host board only. Module compliance is not currently supported in this kit. The kit includes IBIS-AMI TX and RX models for reference and compliance testing. Pre-layout and post-layout schematic sheets are provided. Package models are included only for post-layout reference IBIS-AMI models. These package models contain the QSFP+ connector, HCB (host compliance board), MCB, and SMA connector S-parameters. You can insert a channel design and test for compliance as specified in the QSFP+ specification (SFF-8431 for SerDes channel).

Open QSFP+ Kit

Open the QSFP+ kit in the **Serial Link Designer** app using the `openSignalIntegrityKit` function.

```
openSignalIntegrityKit("QSFP_Plus");
```



Serial Link Designer: qsfp_plus_interface.qcd Project: C:\MATLAB\Examples\QSFP_Plus

Solution Space:

Transfer Net	Variable:	Type:	Format:	Variation Group:	Value 1:	Value 2:
Channel_Network_Compliance	Etch	Corner	List	Corners	TE (Typ)	
Channel_Network_Compliance	Process	Corner	List	Corners	TT (Typ)	
Channel_Network_Compliance	Host_TX.tap_filter.-1	Float	AMI Range	<none>	0.0	
Channel_Network_Compliance	Host_TX.tap_filter.0	Float	AMI Range	<none>	1.0	
Channel_Network_Compliance	Host_TX.tap_filter.1	Float	AMI Range	<none>	0.0	
Channel_Network_Compliance	Host_TX.Tx_swing	Float	AMI Range	<none>	0.65	
Channel_Network_Compliance	Host_TX.Tx_Dj	UI	AMI Range	<none>	0	
Channel_Network_Compliance	Host_TX.Tx_Rj	UI	AMI Range	<none>	0	

Reference Set: I_Project_Schematics   Unset   Current Set: All_Project_Schematics

Kit Overview

- Project Name: QSFP_Plus
- Interface Name: QSFP_Plus
- Target Operating Frequency: 11.1 Gb/s (UI = 90.009 ps)

The QSFP+ kit defines three schematic sets.

- **Pre_Layout_Reference_Schematics** – Used for Pre-Layout testing of channel
- **Post_Layout_Reference_Schematics** – Used for reference sheets when doing Post-Layout
- **All_Project_Schematics** – Set of all project Schematics

For more information about the QSFP+ channel compliance schematics, transfer net properties, and compliance rules, refer to the document QSFP_Plus.pdf that is attached to this example as a supporting file.

References

[1] SFF Committee: SFF-8431 Specifications for Enhanced Small Form Factor Pluggable Module QSFP+. Revision 4.1, 6th of July 2009.

See Also

Serial Link Designer

SAS 3.0 Compliance Kit

Characterize and validate the performance of an SAS 3.0 channel design.

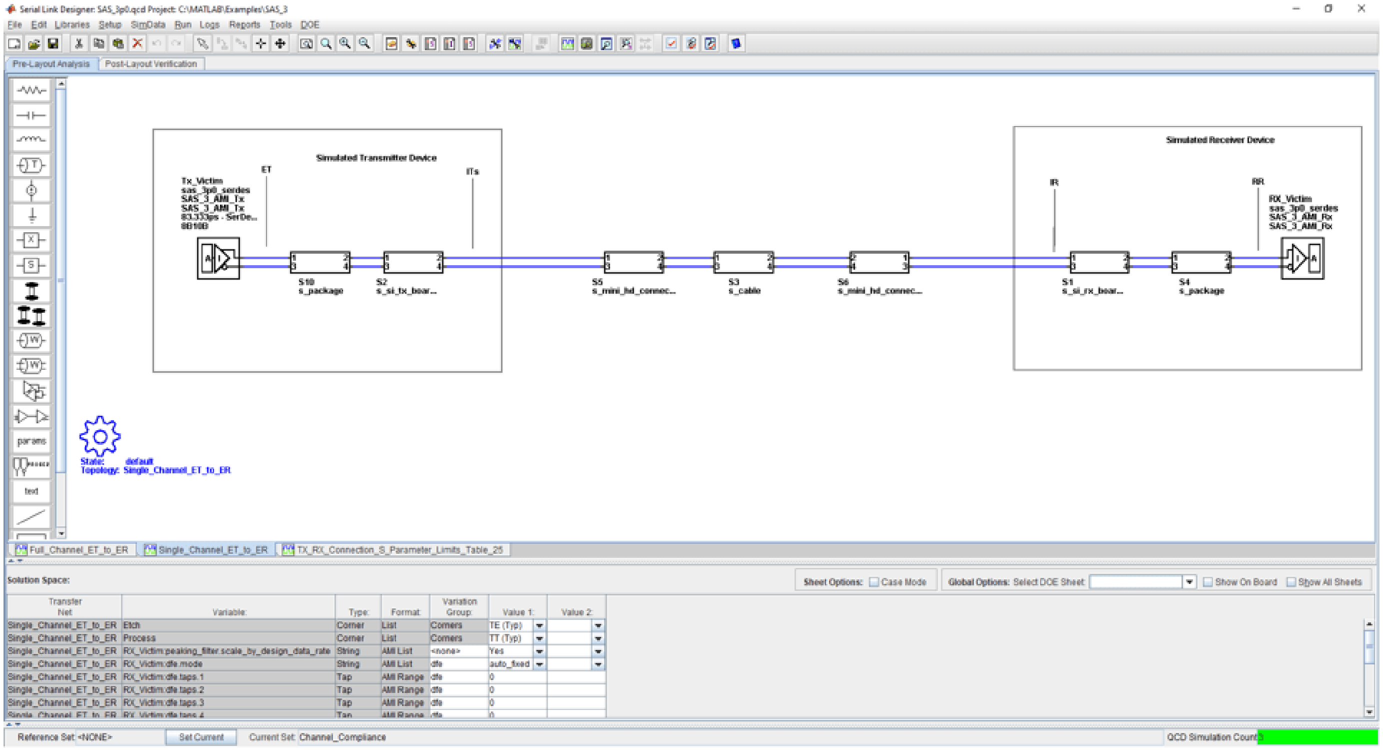
This kit is designed for analysis of a channel design with two mated connectors as provided on the current SAS 3.0 specification provided on the T10 website. The kit also includes sheets containing the backplane/cable and connectors with two plug-in cards attached. In addition, SAS3 reference IBIS-AMI TX and RX models are provided with representative package models. Widebus sheets in this kit are included for crosstalk simulations for full channel and receiver stress tests.

This kit enables you to insert a channel design and characterize and validate its performance using the specification masks to determine if the channel has a high confidence of success. If the channel does not meet the compliance masks or BER estimates, further investigation or redesign, along with simulation, will need to be performed to determine possible changes to meet compliance.

Open SAS 3.0 Kit

Open the SAS 3.0 kit in the **Serial Link Designer** app using the `openSignalIntegrityKit` function.

```
openSignalIntegrityKit("SAS_3");
```



Solution Space:

Transfer Net	Variable:	Type:	Format:	Variation Group:	Value 1:	Value 2:
Single_Channel_ET_to_ER	Etch	Corner	List	Corners	TE (Typ)	
Single_Channel_ET_to_ER	Process	Corner	List	Corners	TT (Typ)	
Single_Channel_ET_to_ER	RX_Victim:peaking_filter.scale_by_design_data_rate	String	AMI List	<none>	Yes	
Single_Channel_ET_to_ER	RX_Victim:dfe.mode	String	AMI List	dfe	auto_fixed	
Single_Channel_ET_to_ER	RX_Victim:dfe.taps.1	Tap	AMI Range	dfe	0	
Single_Channel_ET_to_ER	RX_Victim:dfe.taps.2	Tap	AMI Range	dfe	0	
Single_Channel_ET_to_ER	RX_Victim:dfe.taps.3	Tap	AMI Range	dfe	0	
Single_Channel_ET_to_ER	RX_Victim:dfe.taps.4	Tap	AMI Range	dfe	0	

Reference Set: <NONE>   Set Current   Current Set: Channel_Compliance

Kit Overview

- Project Name: SAS_3
- Interface Name: SAS_3p0
- Target Operating Frequency: 12 Gb/s (UI = 83.333 ps)

The SAS 3.0 kit defines three schematic sets. Schematic sheets are included for testing a SAS3 channel with mated connectors, and a cable/backplane with two plug-in cards. The masks provided in this kit are provided in the 12 GB/s SAS3 specification.

- **Channel_Compliance** – Schematic sheets focused on channel end-to-end compliance
- **Stressed_Receiver** – Stressed receiver tests based on specification requirements and ISI generation
- **Transmitter_Compliance** – Compliance tests for transmitter and transmitter device characteristics

For more information about the SAS 3.0 channel compliance schematics, transfer net properties, and compliance rules, refer to the document SAS3.pdf that is attached to this example as a supporting file.

References

[1] Serial Attached SCSI -3 (SAS-3). Working_draft_15_nov_2012.pdf (Revision 04).

[2] 21250-WTP-001-A_mindspeed_sas_info.pdf. SAS Info from Mindspeed.

[3] SAS Protocol Layer - 2 (SPL-2). T10/2228-D (Revision 05, 10 Nov. 2012).

See Also

Serial Link Designer

SATA 3.0 Compliance Kit

Characterize and validate the performance of a SATA 3.0 channel design.

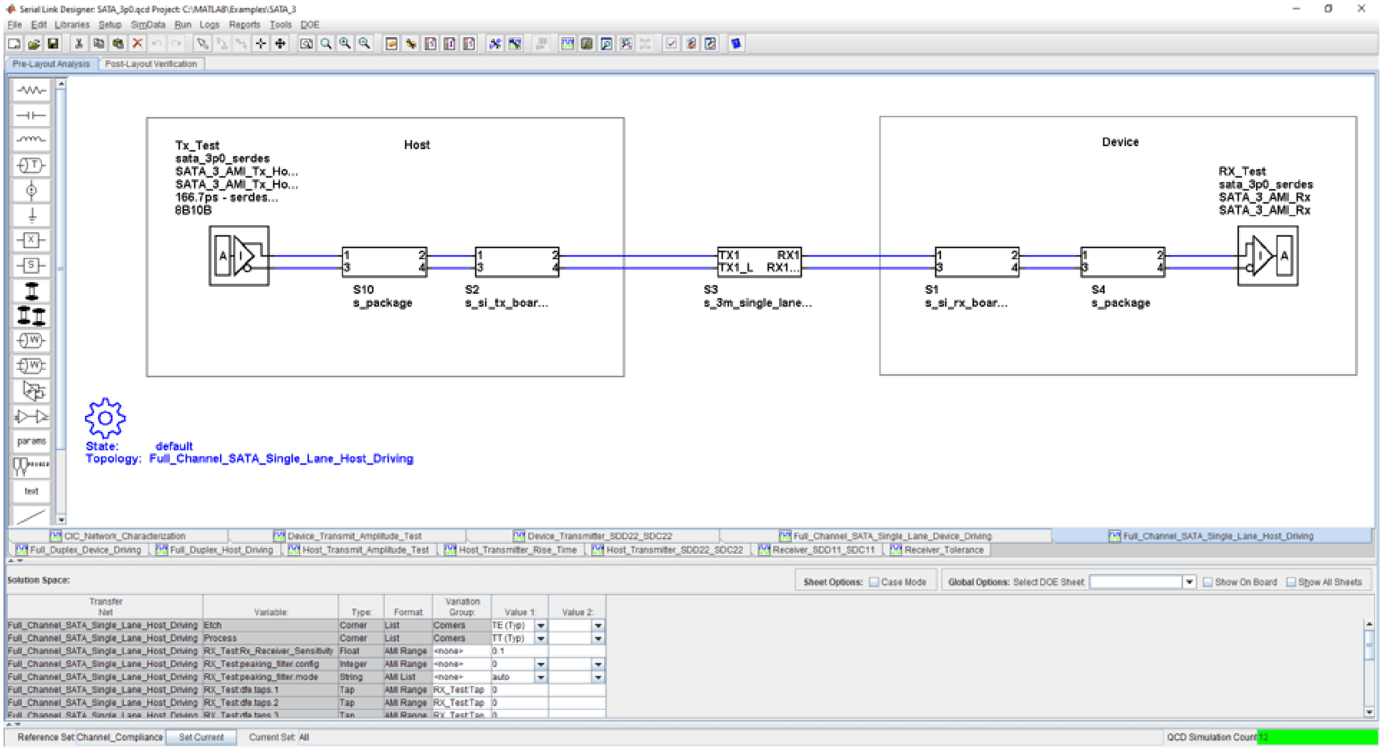
This kit is designed for analysis of a channel design between the SATA 3.0 host and a SATA 3.0 device. The channel consists of a host board and a device board connected by a SATA cable with two mated connectors consistent with the SATA 3.0 specification.

This kit enables you to insert a channel and/or cable design and characterize and validate its performance using the specification masks or other specification requirements to determine if the channel has a high confidence of success. If the channel does not meet the compliance masks or BER estimates, further investigation or redesign, along with simulation, will need to be performed to determine possible changes to meet compliance.

Open SATA 3.0 Kit

Open the SATA 3.0 kit in the **Serial Link Designer** app using the `openSignalIntegrityKit` function.

```
openSignalIntegrityKit("SATA_3");
```


Serial Link Designer: SATA_3p0.qcd Project: C:\MATLAB\Examples\SATA_3
File Edit Libraries Setup SimData Run Logs Reports Tools DOE
Pre-Layout Analysis
Post-Layout Verification
Tx_Test
sata_3p0_serdes
SATA_3_AMI_Tx_Ho...
SATA_3_AMI_Tx_Ho...
166.7ps - serdes...
8B10B
Host
S10
s_package
S2
s_si_tx_boar...
TX1 RX1
TX1_L RX1...
S3
s_3m_single_lane...
Device
RX_Test
sata_3p0_serdes
SATA_3_AMI_Rx
SATA_3_AMI_Rx
S1
s_si_rx_boar...
S4
s_package
params
text
State: default
Topology: Full_Channel_SATA_Single_Lane_Host_Driving
CIC_Network_Characterization
Device_Transmit_Amplitude_Test
Device_Transmitter_SDD22_SDC22
Full_Channel_SATA_Single_Lane_Device_Driving
Full_Channel_SATA_Single_Lane_Host_Driving
Full_Duplex_Device_Driving
Full_Duplex_Host_Driving
Host_Transmit_Amplitude_Test
Host_Transmitter_Rise_Time
Host_Transmitter_SDD22_SDC22
Receiver_SDD11_SDC11
Receiver_Tolerance
Sheet Options: □ Case Mode
Global Options: Select DOE Sheet
□ Show On Board
□ Show All Sheets
Set Current
QCD Simulation Count: 12

Solution Space:

Transfer Net	Variable:	Type:	Format:	Variation Group:	Value 1:	Value 2:
Full_Channel_SATA_Single_Lane_Host_Driving	Etch	Corner	List	Corners	TE (Typ)	
Full_Channel_SATA_Single_Lane_Host_Driving	Process	Corner	List	Corners	TT (Typ)	
Full_Channel_SATA_Single_Lane_Host_Driving	RX_Test:Rx_Receiver_Sensitivity	Float	AMI Range	<none>	0.1	
Full_Channel_SATA_Single_Lane_Host_Driving	RX_Test:peaking_filter.config	Integer	AMI Range	<none>	0	
Full_Channel_SATA_Single_Lane_Host_Driving	RX_Test:peaking_filter.mode	String	AMI List	<none>	auto	
Full_Channel_SATA_Single_Lane_Host_Driving	RX_Test:dfe.taps.1	Tap	AMI Range	RX_Test:Tap	0	
Full_Channel_SATA_Single_Lane_Host_Driving	RX_Test:dfe.taps.2	Tap	AMI Range	RX_Test:Tap	0	
Full_Channel_SATA_Single_Lane_Host_Driving	RX_Test:dfe.taps.3	Tap	AMI Range	RX_Test:Tap	0	

Reference Set: Channel_Compliance

Current Set: All

Kit Overview

- Project Name: SATA_3
- Interface Name: SATA_3p0
- Target Operating Frequency: 6 Gb/s (UI = 166.7 ps)

The SATA 3.0 kit defines three schematic sets. The first focuses channel compliance, second is for transmitter compliance and the third is for receiver compliance. A full-duplex channel is provided for aggressor crosstalk between the TX and RX channels of the full duplex structure.

- **Channel_Compliance** – Schematic sheets focused on channel end-to-end compliance. SerDes and widebus sheets.
- **Transmitter Compliance** – Schematic sheets for compliance and calibration testing of the TX
- **Receiver Compliance** – Schematic sheets for compliance and tolerance testing of the RX

For more information about the SATA 3.0 channel compliance schematics, transfer net properties, and compliance rules, refer to the document SATA_3p0.pdf that is attached to this example as a supporting file.

References

[1] Serial ATA Revision 3.1 (July 18, 2011). SerialATA_Revision_3_1_Gold.pdf.

[2] SATA-IO Interoperability and Technical Training (November 15, 2010). SATA-IO-Tech-Training-Master_v2_PostedNoDigital.pdf.

See Also

Serial Link Designer

SFP+ Compliance Kit

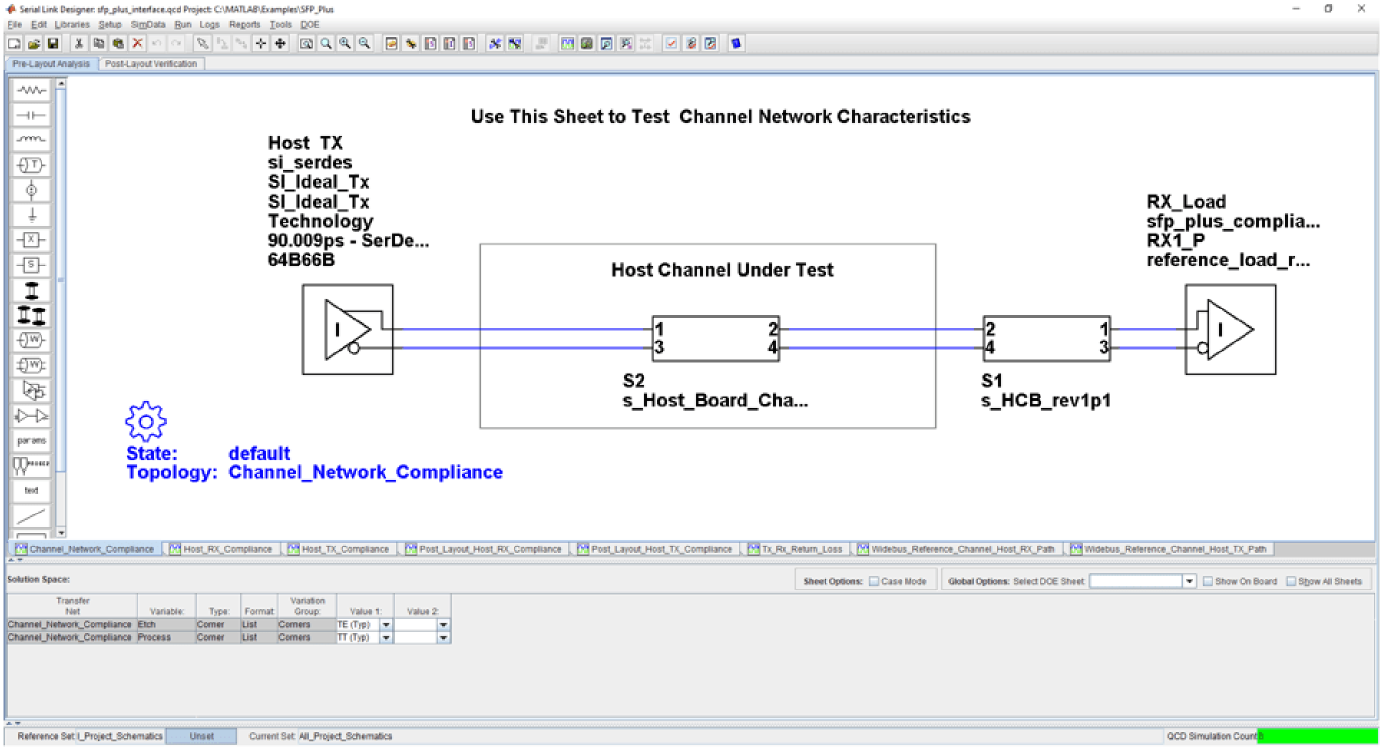
Test the channel design of a host board for compliance to the SFP+ specification.

This kit is designed for compliance of the host board only. Module compliance is not currently supported in this kit. The kit includes IBIS-AMI TX and RX models for reference and compliance testing. Pre-layout and post-layout schematic sheets are provided. Package models are included only for post-layout reference IBIS-AMI models. These package models contain the SFP+ connector, HCB (host compliance board), MCB, and SMA connector S-parameters. You can insert a channel design and test for compliance as specified in the SFP+ specification (SFF-8431 for SerDes channel).

Open SFP+ Kit

Open the SFP+ kit in the **Serial Link Designer** app using the `openSignalIntegrityKit` function.

```
openSignalIntegrityKit("SFP_Plus");
```


Serial Link Designer: sfp_plus_interface.qcd Project: C:\MATLAB\Examples\SFP_Plus

File Edit Libraries Setup SimData Run Logs Reports Tools DOE
Pre-Layout Analysis
Post-Layout Verification
params
text
Use This Sheet to Test Channel Network Characteristics
Host TX
si_serdes
SI_Ideal_Tx
SI_Ideal_Tx
Technology
90.009ps - SerDe...
64B66B
Host Channel Under Test
1
3
2
4
S2
s_Host_Board_Cha...
2
4
1
3
S1
s_HCB_rev1p1
RX_Load
sfp_plus_complia...
RX1_P
reference_load_r...
State: default
Topology: Channel_Network_Compliance
Channel_Network_Compliance
Host_RX_Compliance
Host_TX_Compliance
Post_Layout_Host_RX_Compliance
Post_Layout_Host_TX_Compliance
Tx_Rx_Return_Loss
Widebus_Reference_Channel_Host_RX_Path
Widebus_Reference_Channel_Host_TX_Path

Solution Space:

Sheet Options: □ Case Mode Global Options: Select DOE Sheet: ▼ □ Show On Board □ Show All Sheets

Transfer Net	Variable:	Type:	Format:	Variation Group:	Value 1:	Value 2:
Channel_Network_Compliance	Etch	Corner	List	Corners	TE (Typ) ▼	▼
Channel_Network_Compliance	Process	Corner	List	Corners	TT (Typ) ▼	▼

Reference Set: l_Project_Schematics Unset Current Set: All_Project_Schematics QCD Simulation Count: 0

Kit Overview

- Project Name: SFP_Plus
- Interface Name: SFP_Plus
- Target Operating Frequency: 11.1 Gb/s (UI = 90.009 ps)

The SFP+ kit defines three schematic sets.

- **Pre_Layout_Reference_Schematics** – Used for pre-layout testing of channel
- **Post_Layout_Reference_Schematics** – Used for reference sheets when doing post-layout

- **All_Project_Schematics** – Set of all project schematics

For more information about the SFP+ channel compliance schematics, transfer net properties, and compliance rules, refer to the document SFP_Plus.pdf that is attached to this example as a supporting file.

References

[1] SFF Committee: SFF-8431 Specifications for Enhanced Small Form Factor Pluggable Module SFP +. Revision 4.1, 6th of July 2009. SFF-8431-(SFP+%20MSA).pdf.

See Also

Serial Link Designer

USB 3.0 Compliance Kit

Characterize and validate the performance of a USB 3.0 channel design.

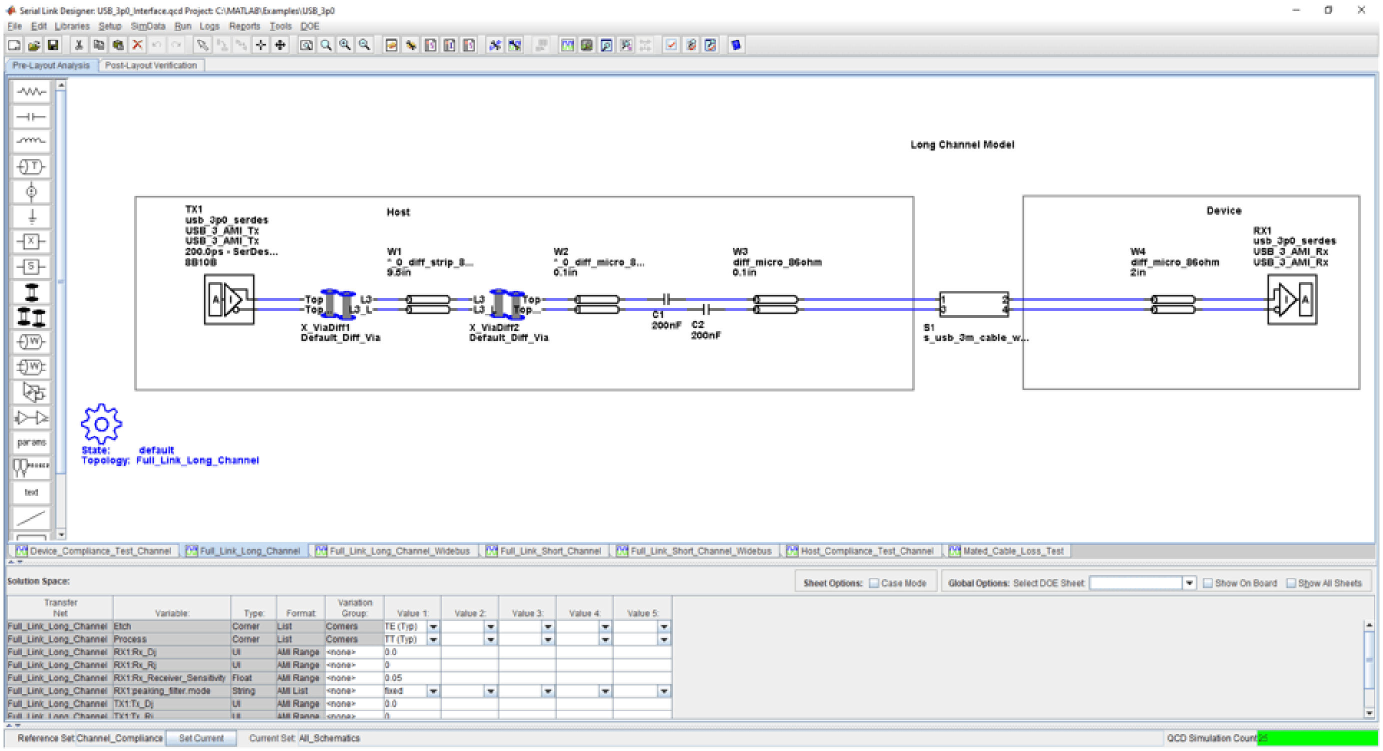
This kit is designed for analysis of a channel design between the USB 3.0 host and a USB hub, or between a USB 3.0 hub and a USB 3.0 device. The channel consists of a host board and a device board (hub or peripheral device) connected by a USB cable with two mated connectors consistent with the USB 3.0 specification.

This kit enables you to insert a channel design and characterize and validate its performance using the specification masks to determine if the channel has a high confidence of success. If the channel does not meet the compliance masks or BER estimates, further investigation or redesign, along with simulation, will need to be performed to determine possible changes to meet compliance.

Open USB 3.0 Kit

Open the USB 3.0 kit in the **Serial Link Designer** app using the `openSignalIntegrityKit` function.

```
openSignalIntegrityKit("USB_3p0");
```



Transfer Net:	Variable:	Type:	Format:	Variation Group:	Value 1:	Value 2:	Value 3:	Value 4:	Value 5:
Full_Link_Long_Channel	Etch	Corner	List	Corners	TE (Typ)				
Full_Link_Long_Channel	Process	Corner	List	Corners	TT (Typ)				
Full_Link_Long_Channel	RX1:Rx_Dj	UI	AMI Range	<none>	0.0				
Full_Link_Long_Channel	RX1:Rx_Rj	UI	AMI Range	<none>	0				
Full_Link_Long_Channel	RX1:Rx_Receiver_Sensitivity	Float	AMI Range	<none>	0.05				
Full_Link_Long_Channel	RX1:peaking_filter.mode	String	AMI List	<none>	fixed				
Full_Link_Long_Channel	TX1:Tx_Dj	UI	AMI Range	<none>	0.0				
Full_Link_Long_Channel	TX1:Tx_Rj	UI	AMI Range	<none>	0				

Kit Overview

- Project Name: USB_3p0
- Interface Name: USB_3p0
- Operating Frequency: 5 GB/s (UI = 200 ps)

The USB 3.0 kit defines three schematic sets. One schematic set focuses on channel compliance and one schematic set is for device compliance. Both long and short channels are modeled along with FEXT/NEXT aggressor characteristics for crosstalk simulations.

- **Channel_Compliance** – Schematic sheets focused on channel end-to-end compliance. Serdes and widebus sheets
- **Device Compliance** – Schematic sheets for DUT compliance driving either device or host boards. Widebus sheets only.
- **All_Schematics** – All project schematic sheets

For more information about the USB 3.0 channel compliance schematics, transfer net properties, and compliance rules, refer to the document USB_3p0.pdf that is attached to this example as a supporting file.

References

[1] Universal Serial Bus 3.0 (May 1, 2011). USB3_r1.0_06_06_2011.pdf.

[2] USB_Superspeed_Equalizer_Design_Guidelines. USB_Superspeed_Equalizer_Design_Guidelines_2011-06-10.pdf.

[3] Simplifying Validation and Debug of USB 3.0 Designs (Tektronix). www.tektronix.com/applications/serial_data/usb.html.

[4] USB 3.0 Electrical Compliance Methodology White Paper (Revision 0.5). USB_3_0_e-Compliance_methodology_0p5_whitepaper.pdf.

See Also

Serial Link Designer

USB 3.1 Compliance Kit

Characterize and validate the performance of a USB 3.1 channel design.

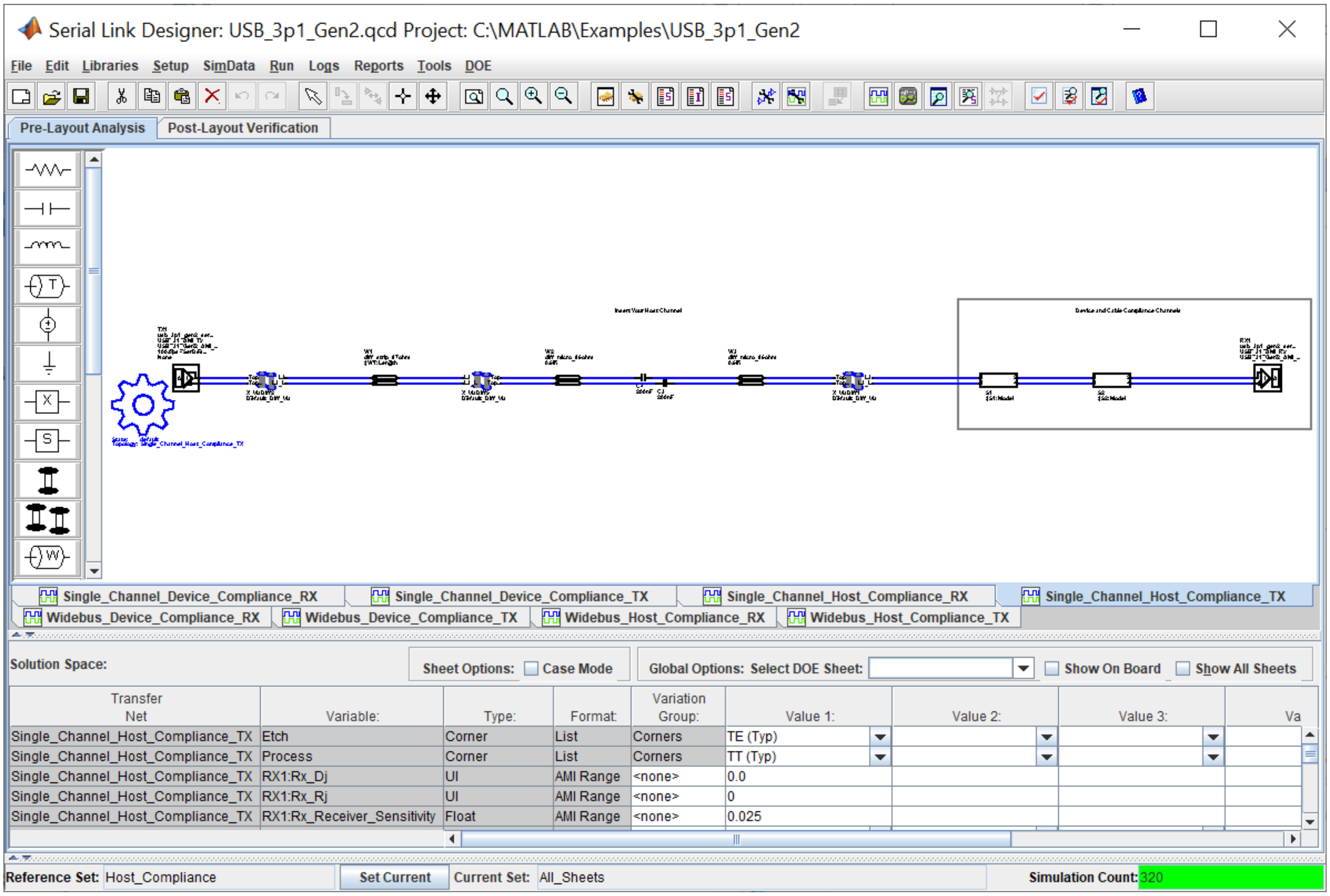
This kit is designed for analysis of a channel design between the USB 3.1 host and device. One schematic set focuses on host compliance and one schematic set focuses on device compliance. Both single channel schematics and multi-channel schematics are provided for host and device compliance. Reference S-parameter models from the USB 3.1 website are included for compliance testing. Each schematic directs you where to place your channel design.

This kit enables you to insert a channel design and characterize and validate its performance using the specification masks to determine if the channel has a high confidence of success. If the channel does not meet the compliance masks or BER estimates, further investigation or redesign, along with simulation, will need to be performed to determine possible changes to meet compliance.

Open USB 3.1 Kit

Open the USB 3.1 kit in the **Serial Link Designer** app using the `openSignalIntegrityKit` function.

```
openSignalIntegrityKit("USB_3p1_Gen2");
```


Serial Link Designer: USB_3p1_Gen2.qcd Project: C:\MATLAB\Examples\USB_3p1_Gen2
File Edit Libraries Setup SimData Run Logs Reports Tools DOE
Pre-Layout Analysis
Post-Layout Verification
Insert Your Host Channel
Device and Cable Compliance Channels
Single_Channel_Device_Compliance_RX
Single_Channel_Device_Compliance_TX
Single_Channel_Host_Compliance_RX
Single_Channel_Host_Compliance_TX
Widebus_Device_Compliance_RX
Widebus_Device_Compliance_TX
Widebus_Host_Compliance_RX
Widebus_Host_Compliance_TX

Solution Space: Sheet Options: □ Case Mode Global Options: Select DOE Sheet: □ Show On Board □ Show All Sheets

Transfer Net	Variable:	Type:	Format:	Variation Group:	Value 1:	Value 2:	Value 3:	Va
Single_Channel_Host_Compliance_TX	Etch	Corner	List	Corners	TE (Typ)			
Single_Channel_Host_Compliance_TX	Process	Corner	List	Corners	TT (Typ)			
Single_Channel_Host_Compliance_TX	RX1:Rx_Dj	UI	AMI Range	<none>	0.0			
Single_Channel_Host_Compliance_TX	RX1:Rx_Rj	UI	AMI Range	<none>	0			
Single_Channel_Host_Compliance_TX	RX1:Rx_Receiver_Sensitivity	Float	AMI Range	<none>	0.025			

Reference Set: Host_Compliance Set Current Current Set: All_Sheets Simulation Count: 320

Kit Overview

- Project Name: USB_3p1_Gen2
- Interface Name: USB_3p1_Gen2
- Operating Frequency: 10 Gb/s (UI = 100 ps)

The USB 3.1 kit defines three schematic sets.

- **Host_Compliance** – Schematic sheets focused on channel end-to-end compliance. Serdes and widebus sheets.
- **Device Compliance** – Schematic sheets for DUT compliance driving either device or host boards. Widebus sheets only.
- **All_Schematics** – All project schematic sheets.

For more information about the USB 3.1 channel compliance schematics, transfer net properties, and compliance rules, refer to the document USB_3p1.pdf that is attached to this example as a supporting file.

References

[1] Universal Serial Bus 3.1 (July 26, 2013). USB_3_1_r1.0.pdf.

[2] USB 3.0 Electrical Compliance Methodology White Paper (Revision 0.5). USB_3_0_e-Compliance_methodology_0p5_whitepaper.pdf.

See Also

Serial Link Designer

XAUI Compliance Kit

Characterize and validate the performance of a 10 Gigabit Attachment Unit Interface (XAUI) channel design.

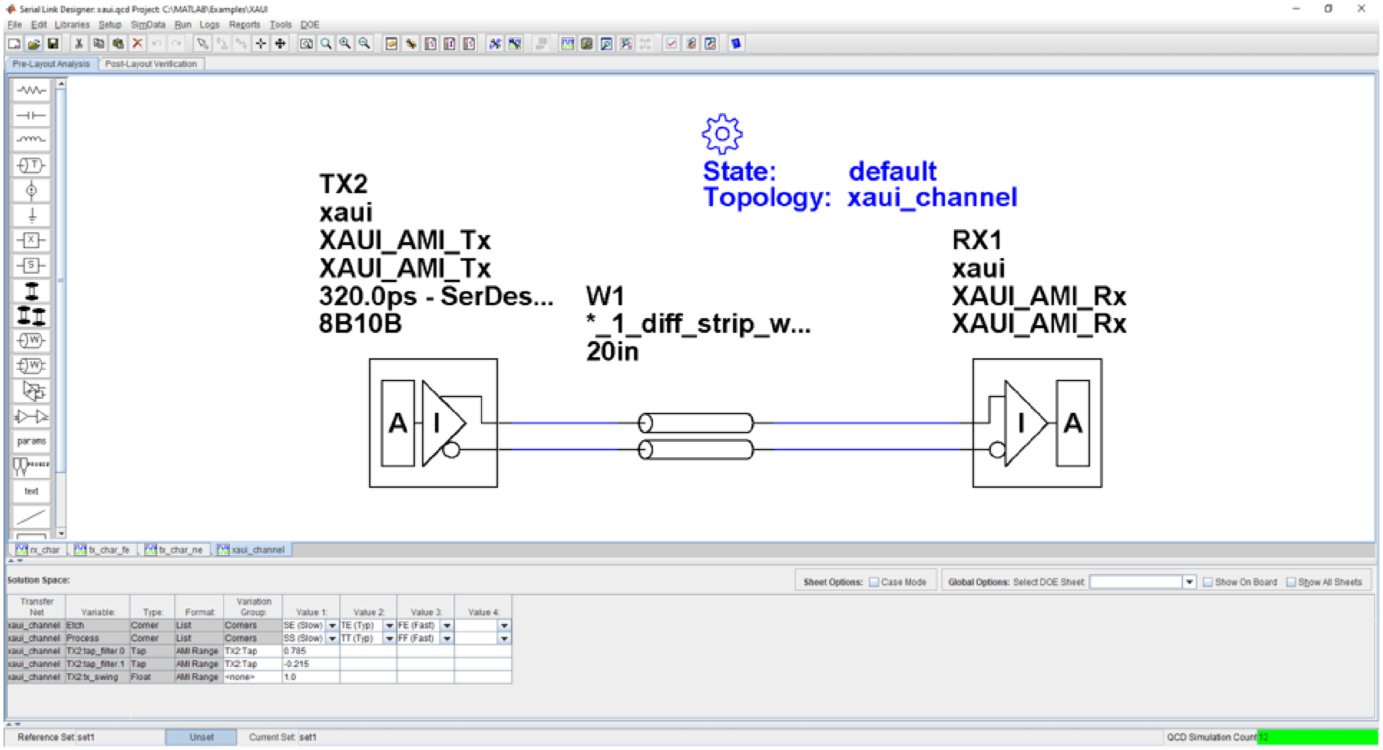
This XAUI compliance signal integrity kit includes all the transfer nets, generic buffer models, and eye masks for an XAUI high-speed SerDes interface. This includes XAUI technology IBIS-AMI models for the SerDes transmitter and receiver, XAUI eye masks, transfer nets preconfigured for TX and RX characterization, and an easily customizable end-to-end transfer net for a full XAUI link.

This kit enables you to insert a channel design and characterize and validate its performance using the specification masks to determine if the channel has a high confidence of success. If the channel does not meet the compliance masks or BER estimates, further investigation or redesign, along with simulation, will need to be performed to determine possible changes to meet compliance.

Open XAUI Kit

Open the XAUI kit in the **Serial Link Designer** app using the `openSignalIntegrityKit` function.

```
openSignalIntegrityKit("XAUI");
```



Solution Space:

Transfer Net	Variable:	Type:	Format:	Variation Group:	Value 1:	Value 2:	Value 3:	Value 4:
xaui_channel	Etch	Corner	List	Corners	SE (Slow)	TE (Typ)	FE (Fast)	
xaui_channel	Process	Corner	List	Corners	SS (Slow)	TT (Typ)	FF (Fast)	
xaui_channel	TX2:tap_filter.0	Tap	AMI Range	TX2:Tap	0.785			
xaui_channel	TX2:tap_filter.1	Tap	AMI Range	TX2:Tap	-0.215			
xaui_channel	TX2:tx_swing	Float	AMI Range	<none>	1.0			

Reference Set: set1

Current Set: set1

QCD Simulation Count: 12

Kit Overview

- Project Name: XAUI
- Interface Name: xaui
- Target Operating Frequency: 3.125 Gb/s (320 ps)

The XAUI kit defines one schematic set.

- **set1** — Used for all compliance testing

For more information about the XAUI channel compliance schematics, transfer net properties, and compliance rules, refer to the document XAUI.pdf that is attached to this example as a supporting file.

References

[1] "IEEE Standard for Information Technology - Local and Metropolitan Area Networks - Part 3: CSMA/CD Access Method and Physical Layer Specifications - Media Access Control (MAC) Parameters, Physical Layer, and Management Parameters for 10 Gb/s Operation." *IEEE Std 802.3ae-2002 (Amendment to IEEE Std 802.3-2002)*, August 2002, 1–544. https://doi.org/10.1109/IEEESTD.2002.94131.

See Also

Serial Link Designer

Registered DDR2 Architectural Kit

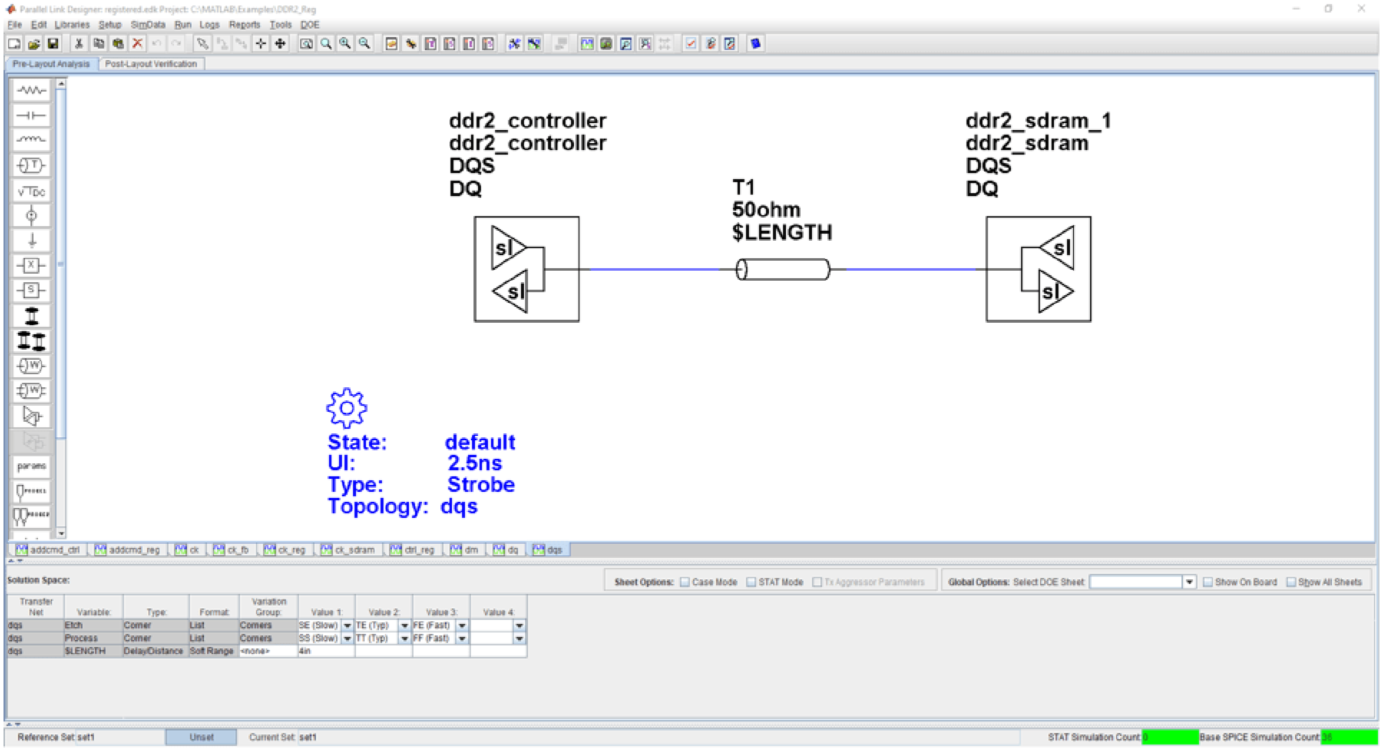
Implement a registered DDR2 interface for pre-layout analysis or post-layout verification.

This registered DDR2 architectural signal integrity kit includes all transfer nets, timing models, waveform processing levels and generic models for a registered DDR2 interface. This includes generic buffer models for the DDR2 controller, PLL, register, and SDRAM, along with fully functional timing models and complete waveform processing levels. You can modify the kit to match your exact implementation. Then, perform complete pre-layout solution space analysis and/or full post-layout verification for waveform quality and timing margins.

Open Registered DDR2 Kit

Open the registered DDR2 kit in the **Parallel Link Designer** app using the `openSignalIntegrityKit` function.

```
openSignalIntegrityKit("DDR2_Reg");
```



State: default
UI: 2.5ns
Type: Strobe
Topology: dqs

Solution Space:

Sheet Options: □ Case Mode □ STAT Mode □ Tx Aggressor Parameters

Global Options: Select DOE Sheet: □ Show On Board □ Show All Sheets

Transfer Net:	Variable:	Type:	Format:	Variation Group:	Value 1:	Value 2:	Value 3:	Value 4:
dqs	Etch	Corner	List	Corners	SE (Slow)	TE (Typ)	FE (Fast)	
dqs	Process	Corner	List	Corners	SS (Slow)	TT (Typ)	FF (Fast)	
dqs	$LENGTH	Delay/Distance	Soft Range	<none>	4in			

Reference Set: set1

Current Set: set1

STAT Simulation Count: 0

Base SPICE Simulation Count: 36

Kit Overview

For more information about the registered DDR2 architectural signal integrity kit, including block diagrams, system configurations, transfer nets and libraries, along with instructions on how to customize the kit for a specific implementation, refer to the document DDR2_Registered.pdf that is attached to this example as a supporting file.

See Also

Parallel Link Designer

Unbuffered DDR2 Architectural Kit

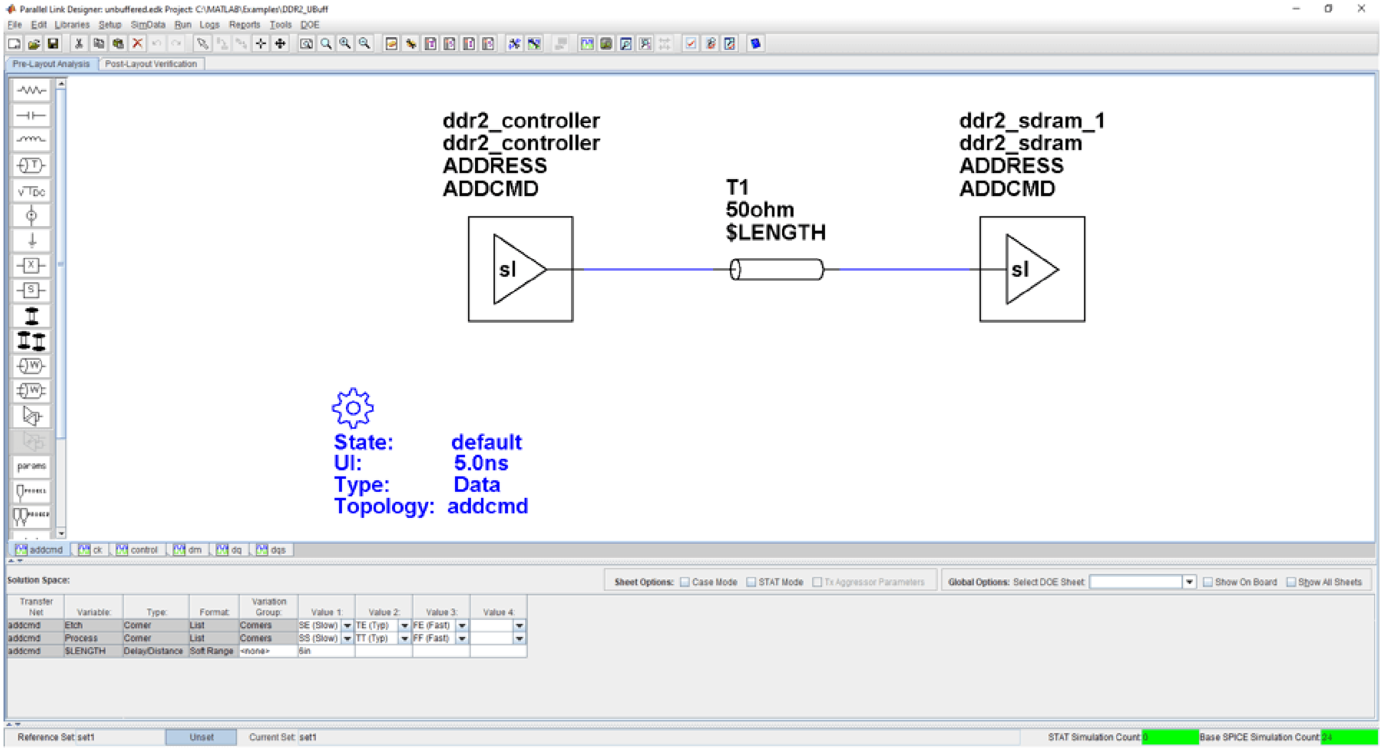
Implement a registered DDR2 interface for pre-layout analysis or post-layout verification.

This unbuffered DDR2 architectural signal integrity kit includes all transfer nets, timing models, waveform processing levels and generic models for an unbuffered DDR2 interface. This includes generic buffer models for the DDR2 controller and SDRAM, along with fully functional timing models and complete waveform processing levels. You can modify the kit to match your exact implementation. Then, perform complete pre-layout solution space analysis and/or full post-layout verification for waveform quality and timing margins.

Open Unbuffered DDR2 Kit

Open the unbuffered DDR2 kit in the **Parallel Link Designer** app using the `openSignalIntegrityKit` function.

```
openSignalIntegrityKit("DDR2_UBuff");
```



State: default
UI: 5.0ns
Type: Data
Topology: addcmd

Solution Space:

Sheet Options: □ Case Mode □ STAT Mode □ Tx Aggressor Parameters

Global Options: Select DOE Sheet: □ Show On Board □ Show All Sheets

Transfer Net	Variable:	Type:	Format:	Variation Group:	Value 1:	Value 2:	Value 3:	Value 4:
addcmd	Etch	Corner	List	Corners	SE (Slow)	TE (Typ)	FE (Fast)	
addcmd	Process	Corner	List	Corners	SS (Slow)	TT (Typ)	FF (Fast)	
addcmd	$LENGTH	Delay/Distance	Soft Range	<none>	6in			

Reference Set: set1 Unset Current Set: set1 STAT Simulation Count: 0 Base SPICE Simulation Count: 24

Kit Overview

For more information about the unbuffered DDR2 architectural signal integrity kit, including block diagrams, system configurations, transfer nets and libraries, along with instructions on how to customize the kit for a specific implementation, refer to the document DDR2_Unbuffered.pdf that is attached to this example as a supporting file.

See Also

Parallel Link Designer

Unbuffered DDR2 with PLL Architectural Kit

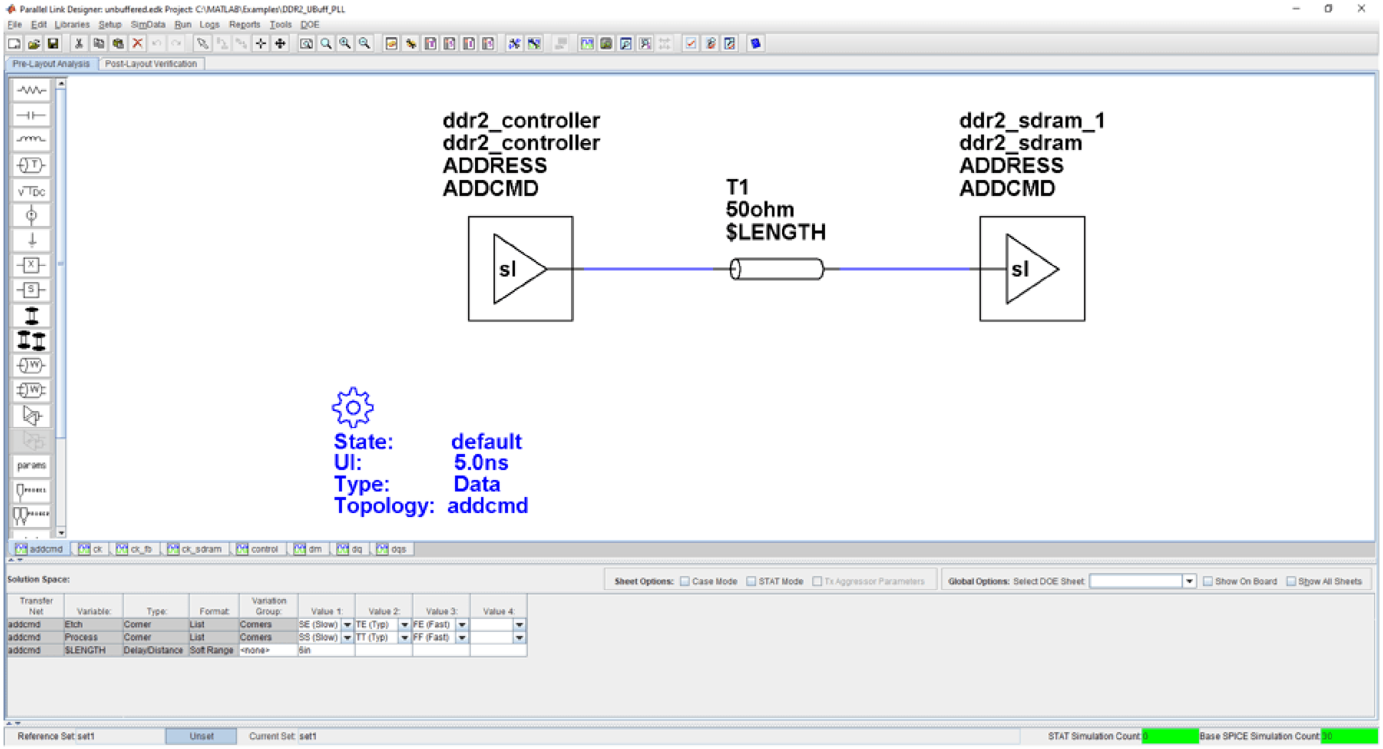
Implement an unbuffered DDR2 interface with PLL clock buffer for pre-layout analysis or post-layout verification.

This unbuffered DDR2 with PLL architectural signal integrity kit includes all transfer nets, timing models, waveform processing levels and generic models for an unbuffered DDR2 interface with PLL clock buffer. This includes generic buffer models for the DDR2 controller, PLL and SDRAM, along with fully functional timing models and complete waveform processing levels. You can modify the kit to match your exact implementation. Then, perform complete pre-layout solution space analysis and/or full post-layout verification for waveform quality and timing margins.

Open Unbuffered DDR2 with PLL Kit

Open the unbuffered DDR2 with PLL kit in the **Parallel Link Designer** app using the `openSignalIntegrityKit` function.

```
openSignalIntegrityKit("DDR2_UBuff_PLL");
```



State: default
UI: 5.0ns
Type: Data
Topology: addcmd

Solution Space:

Sheet Options: □ Case Mode □ STAT Mode □ Tx Aggressor Parameters

Global Options: Select DOE Sheet: □ Show On Board □ Show All Sheets

Transfer Net	Variable:	Type:	Format:	Variation Group:	Value 1:	Value 2:	Value 3:	Value 4:
addcmd	Etch	Corner	List	Corners	SE (Slow)	TE (Typ)	FE (Fast)	
addcmd	Process	Corner	List	Corners	SS (Slow)	TT (Typ)	FF (Fast)	
addcmd	$LENGTH	Delay/Distance	Soft Range	<none>	6in			

Reference Set: set1 Unset Current Set: set1

STAT Simulation Count: 0 Base SPICE Simulation Count: 30

Kit Overview

For more information about the unbuffered DDR2 with PLL architectural signal integrity kit, including block diagrams, system configurations, transfer nets, and libraries, along with instructions on how to customize the kit for a specific implementation, refer to the document DDR2_Unbuffered_With_PLL.pdf that is attached to this example as a supporting file.

See Also

Parallel Link Designer

Registered DDR3 Architectural Kit

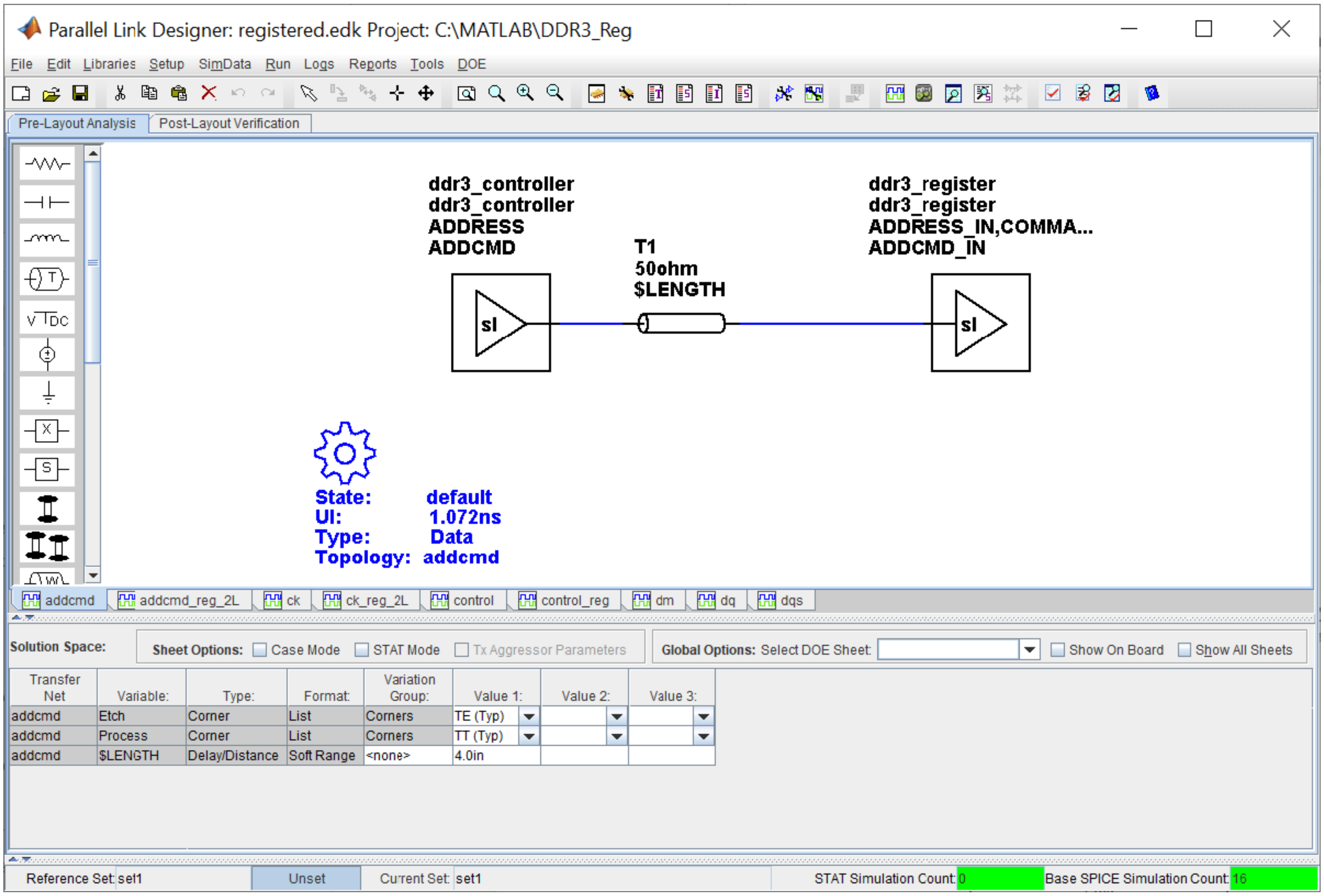
Implement a Registered DDR3 interface for pre-layout analysis or post-layout verification.

This Registered DDR3 architectural signal integrity kit includes the transfer nets, timing models, waveform processing levels and generic models for a registered DDR3 interface. This includes generic buffer models for the DDR3 controller, register and SDRAM, along with fully functional timing models and complete waveform processing levels. You can modify the kit to match your exact implementation. Then, perform complete pre-layout solution space analysis and/or full post-layout verification for waveform quality and timing margins.

Open Registered DDR3 Kit

Open the Registered DDR3 kit in the **Parallel Link Designer** app using the `openSignalIntegrityKit` function.

```
openSignalIntegrityKit("DDR3_Reg");
```



Solution Space: Sheet Options: □ Case Mode □ STAT Mode □ Tx Aggressor Parameters Global Options: Select DOE Sheet: □ Show On Board □ Show All Sheets

Transfer Net	Variable:	Type:	Format:	Variation Group:	Value 1:	Value 2:	Value 3:
addcmd	Etch	Corner	List	Corners	TE (Typ)		
addcmd	Process	Corner	List	Corners	TT (Typ)		
addcmd	$LENGTH	Delay/Distance	Soft Range	<none>	4.0in		

Reference Set: set1 | Unset | Current Set: set1 | STAT Simulation Count: 0 | Base SPICE Simulation Count: 16

Kit Overview

For more information about the Registered DDR3 architectural signal integrity kit, including block diagrams, system configurations, transfer nets and libraries, along with instructions on how to customize the kit for a specific implementation, refer to the document DDR3_Registered.pdf that is attached to this example as a supporting file.

References

[1] JEDEC – DDR3 SDRAM Standard. JESD79-3E, July 2010.

[2] JEDEC – Definition of the SSTE32882 Registering Clock Driver with Parity and Quad Chip Selects for DDR3/DDR3L/DDR3U RDIMM 1.5V/1.35V/1.25V Applications. JESD82-29A, December 2010.

[3] JEDEC – Proposed DDR3-800/1066/1333/1600 tDS, TDH VIH.DQ, VIL.DQ and tVAC AC135 Spec. Committee: JC-42.3C. Committee Item Number: 1680.22.

See Also

Parallel Link Designer

Unbuffered DDR3 Architectural Kit

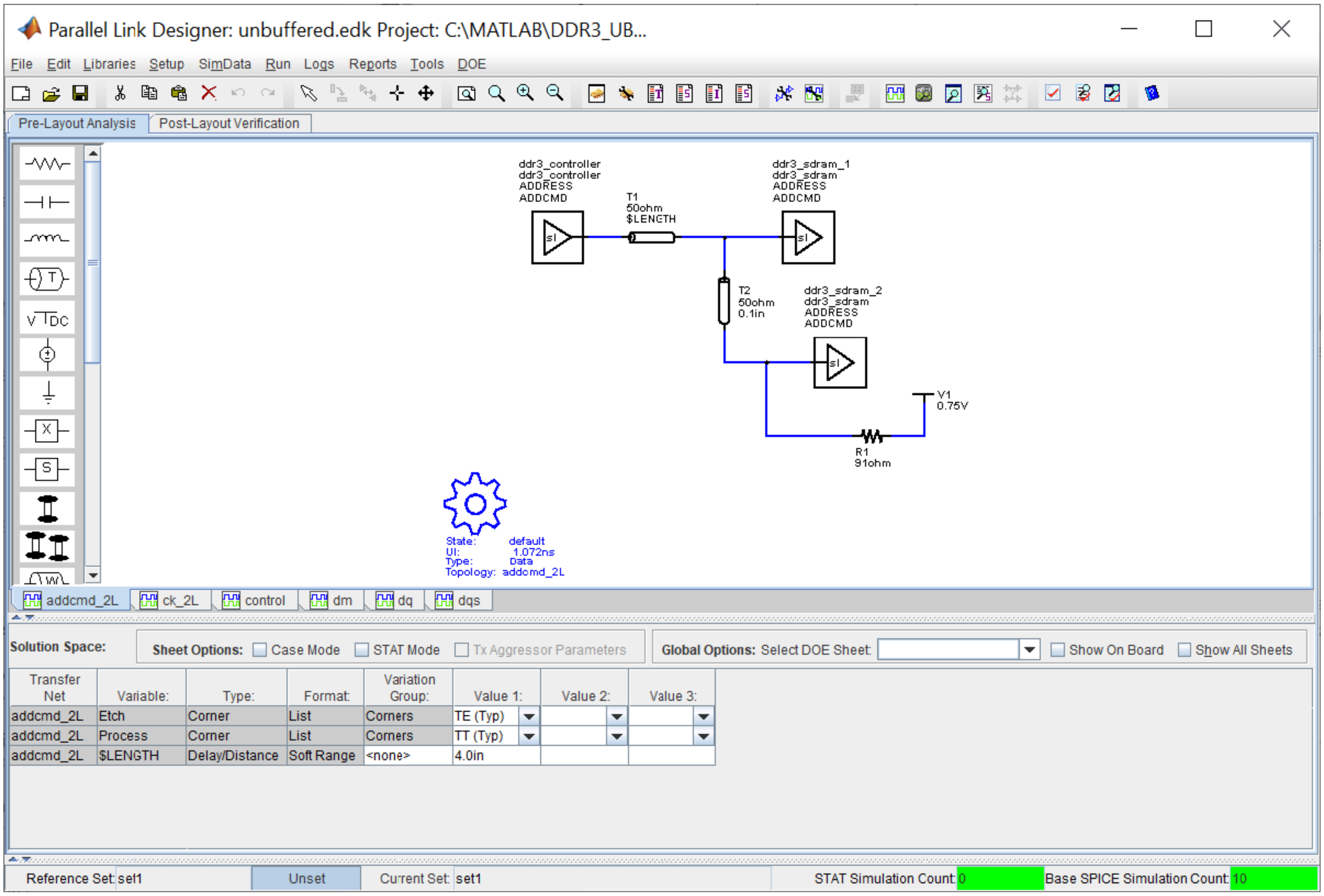
Implement an unbuffered DDR3 interface for pre-layout analysis or post-layout verification.

This unbuffered DDR3 architectural signal integrity kit includes all the transfer nets, timing models, waveform processing levels and generic models for an unbuffered DDR3 interface. This includes generic buffer models for the DDR3 controller and SDRAM, along with fully functional timing models and complete waveform processing levels. You can modify the kit to match your exact implementation. Then, perform complete pre-layout solution space analysis and/or full post-layout verification for waveform quality and timing margins.

Open Unbuffered DDR3 Kit

Open the unbuffered DDR3 kit in the **Parallel Link Designer** app using the `openSignalIntegrityKit` function.

```
openSignalIntegrityKit("DDR3_UBuff");
```



Solution Space:

Sheet Options: □ Case Mode □ STAT Mode □ Tx Aggressor Parameters

Global Options: Select DOE Sheet: □ Show On Board □ Show All Sheets

Transfer Net	Variable:	Type:	Format:	Variation Group:	Value 1:	Value 2:	Value 3:
addcmd_2L	Etch	Corner	List	Corners	TE (Typ)		
addcmd_2L	Process	Corner	List	Corners	TT (Typ)		
addcmd_2L	$LENGTH	Delay/Distance	Soft Range	<none>	4.0in		

Reference Set: set1 Unset Current Set: set1 STAT Simulation Count: 0 Base SPICE Simulation Count: 10

Kit Overview

For more information about the unbuffered DDR3 architectural signal integrity kit, including block diagrams, system configurations, transfer nets and libraries, along with instructions on how to customize the kit for a specific implementation, refer to the document DDR3_Unbuffered.pdf that is attached to this example as a supporting file.

References

[1] JEDEC – DDR3 SDRAM Standard. JESD79-3E, July 2010.

[2] JEDEC – Proposed DDR3-800/1066/1333/1600 tDS, TDH VIH.DQ, VIL.DQ and tVAC AC135 Spec. Committee: JC-42.3C. Committee Item Number: 1680.22.

See Also

Parallel Link Designer

Unbuffered DDR3L Architectural Kit

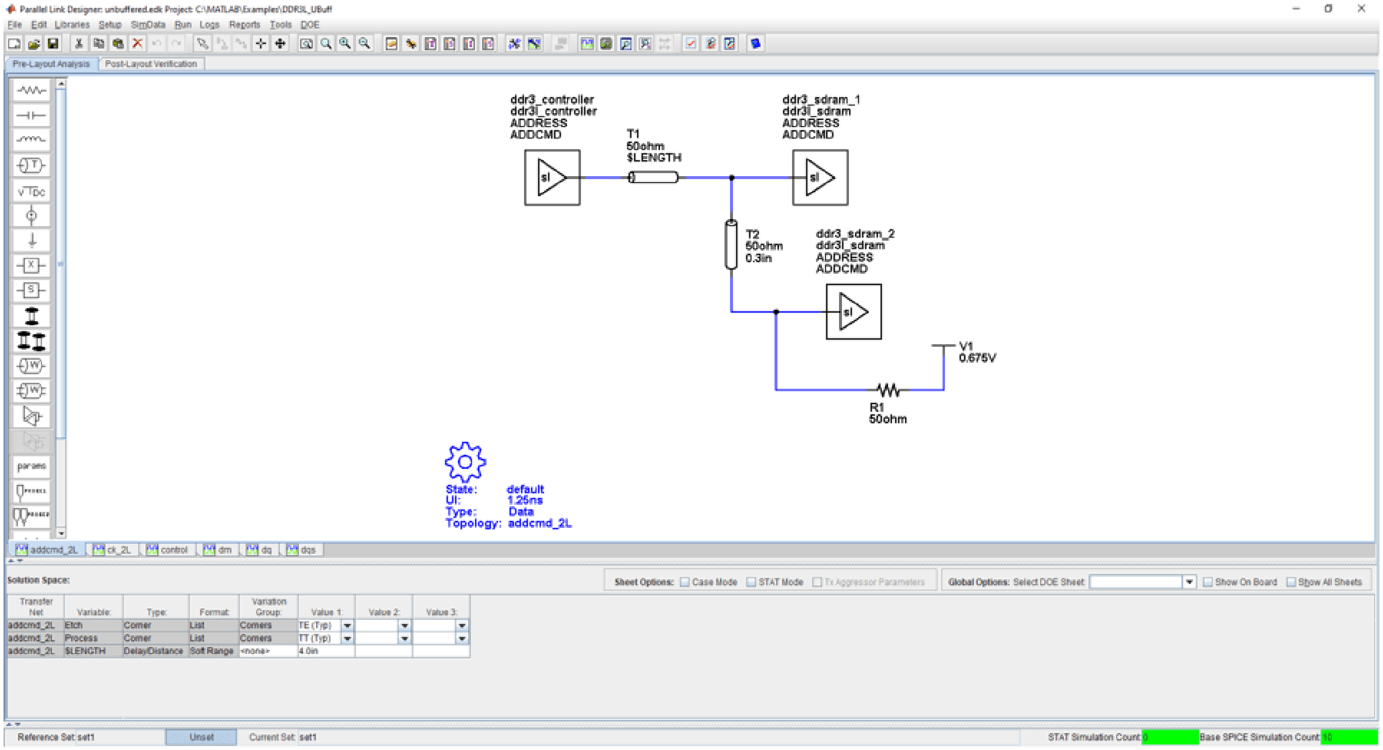
Implement an unbuffered DDR3L interface for pre-layout analysis or post-layout verification.

DDR3L is a lower voltage version of standard DDR3 that utilizes a 1.35V I/O voltage instead of 1.5V. This unbuffered DDR3L architectural signal integrity kit includes all the transfer nets, timing models, waveform processing levels and generic models for an unbuffered DDR3L interface. This includes generic buffer models for the DDR3L controller and SDRAM, along with fully functional timing models and complete waveform processing levels. You can modify the kit to match your exact implementation. Then, perform complete pre-layout solution space analysis and/or full post-layout verification for waveform quality and timing margins.

Open Unbuffered DDR3L Kit

Open the unbuffered DDR3L kit in the **Parallel Link Designer** app using the `openSignalIntegrityKit` function.

```
openSignalIntegrityKit("DDR3L_UBuff");
```



Solution Space:

Transfer Net	Variable:	Type:	Format:	Variation Group:	Value 1:	Value 2:	Value 3:
addcmd_2L	Etch	Corner	List	Corners	TE (Typ)		
addcmd_2L	Process	Corner	List	Corners	TT (Typ)		
addcmd_2L	$LENGTH	Delay/Distance	Soft Range	<none>	4.0in		

Reference Set: set1 Unset Current Set: set1

Kit Overview

For more information about the unbuffered DDR3L architectural signal integrity kit, including block diagrams, system configurations, transfer nets and libraries, along with instructions on how to customize the kit for a specific implementation, refer to the document DDR3L_Unbuffered.pdf that is attached to this example as a supporting file.

References

[1] JEDEC – DDR3 SDRAM Standard. JESD79-3E, July 2010.

[2] JEDEC – Addendum No. 1 to JESD79-3 – 1.35 V DDR3L-800, DDR3L-1066, DDR3L-1333, and DDR3L-1600. JESD79-3-1, July 2010.

See Also

Parallel Link Designer

DDR4 Implementation Kit for JEDEC Raw Card B

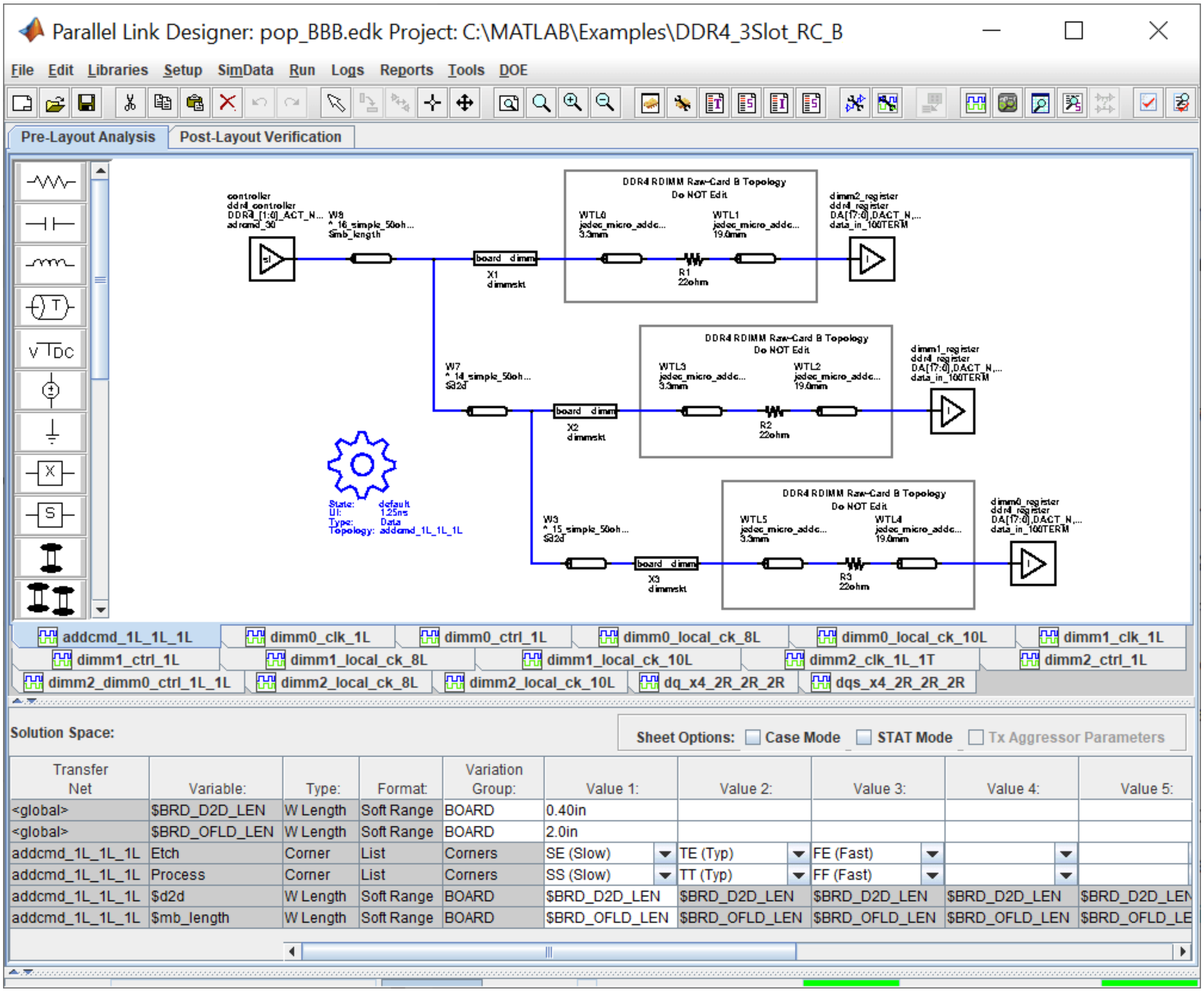
Implement a 3-slot DDR4 Raw Card B RDIMM interface for pre-layout analysis or post-layout verification.

This DDR4 Raw Card B RDIMM implementation signal integrity kit includes block diagrams, system configurations, transfer nets and libraries, which can be easily modified to match your exact implementation. You can modify the kit to match your exact DDR4 implementation. Then, perform complete pre-layout solution space analysis and/or full post-layout verification for waveform quality and timing margins.

Open DDR4 Raw Card B Kit

Open the DDR4 Raw Card B kit in the **Parallel Link Designer** app using the `openSignalIntegrityKit` function.

```
openSignalIntegrityKit("DDR4_3Slot_RC_B");
```



Solution Space:

Sheet Options: □ Case Mode □ STAT Mode □ Tx Aggressor Parameters

Transfer Net	Variable:	Type:	Format:	Variation Group:	Value 1:	Value 2:	Value 3:	Value 4:	Value 5:
<global>	$BRD_D2D_LEN	W Length	Soft Range	BOARD	0.40in				
<global>	$BRD_OFLD_LEN	W Length	Soft Range	BOARD	2.0in				
addcmd_1L_1L_1L	Etch	Corner	List	Corners	SE (Slow)	TE (Typ)	FE (Fast)		
addcmd_1L_1L_1L	Process	Corner	List	Corners	SS (Slow)	TT (Typ)	FF (Fast)		
addcmd_1L_1L_1L	$d2d	W Length	Soft Range	BOARD	$BRD_D2D_LEN	$BRD_D2D_LEN	$BRD_D2D_LEN	$BRD_D2D_LEN	$BRD_D2D_LEN
addcmd_1L_1L_1L	$mb_length	W Length	Soft Range	BOARD	$BRD_OFLD_LEN	$BRD_OFLD_LEN	$BRD_OFLD_LEN	$BRD_OFLD_LEN	$BRD_OFLD_LE

Kit Overview

- Project name: DDR4_3Slot_RC_B
- pop_BBB interface: A 3-slot DDR4 interface with all 3 slots populated with RDIMM modules
- pop_XBB interface: A 3-slot DDR4 interface with 2 slots populated with RDIMM modules
- pop_XXB interface: A 3-slot DDR4 interface with 1 slot populated with RDIMM modules

There are two independent DDR4 channels in the generic controller: DDR4_0 and DDR4_1. Only one channel represented in pre-layout analysis. Post-layout analysis automatically extracts and simulates all channels. This is a 288-pin buffered DDR4 RDIMM. There are 72-bits per channel (64 data, 8 ECC) and 3 RDIMM slots per channel: dimm0, dimm1 and dimm2. Each slot can be populated with Raw Card B DDR4 Registered DIMM modules.

This kit supports both HSPICE and IsSpice4 simulators. No specific version of either simulator is required when running this kit.

For more information about the 3-slot DDR4 Raw Card B implementation signal integrity kit, including block diagrams, system configurations, transfer nets and libraries, refer to the document DDR4_3slot_rcB.pdf that is attached to this example as a supporting file.

See Also

Parallel Link Designer

DDR4 Memory Down Implementation Kit

Implement a DDR4 memory down (MD) interface for pre-layout analysis or post-layout verification.

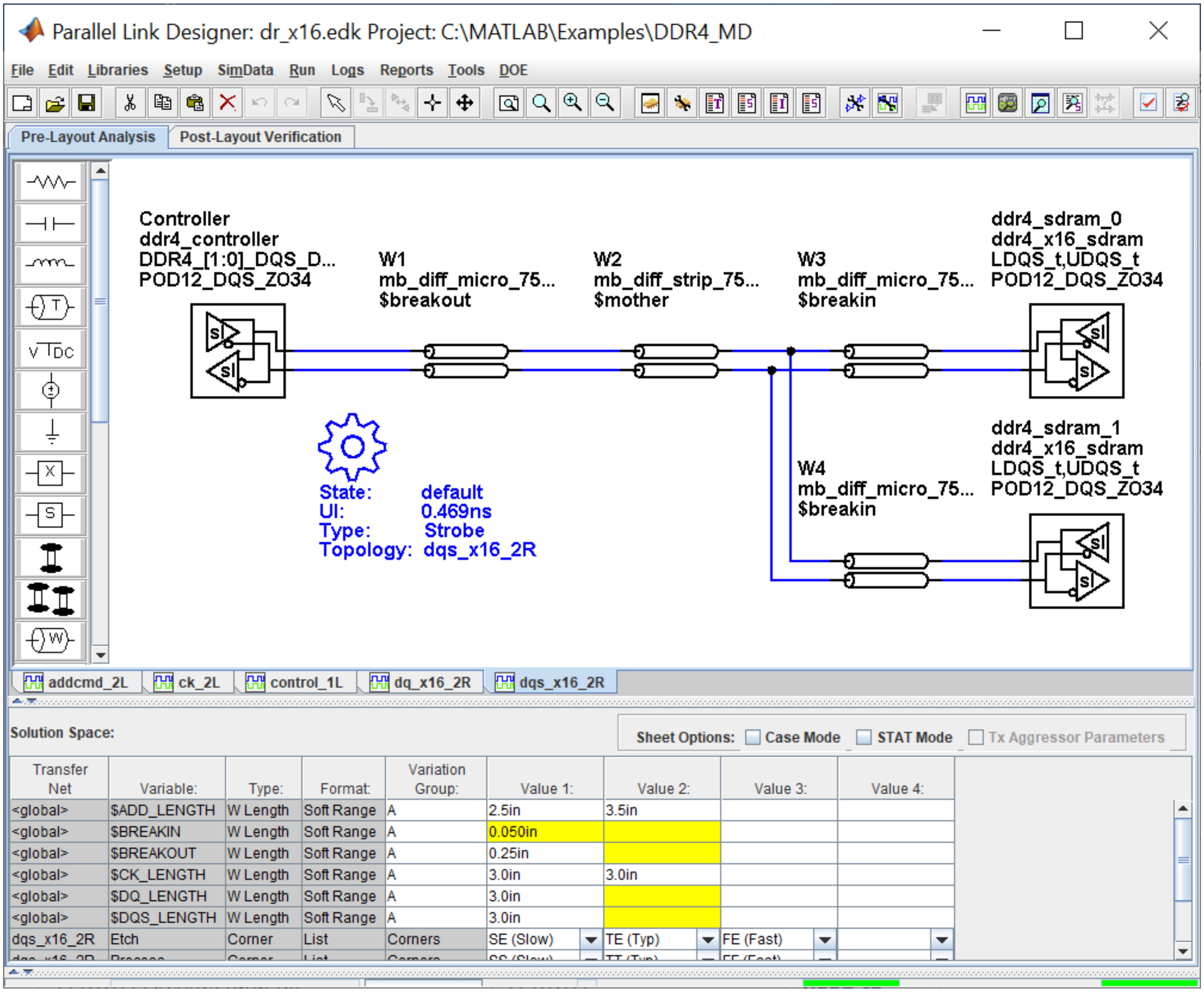
This DDR4 MD implementation signal integrity kit includes all the transfer nets, timing models, waveform processing levels, and simulation models for both single and dual rank memory down (discrete) configurations. This includes buffer models for the DDR4 memory controller as well as Micron SDRAMs. Also included are timing models with complete waveform processing levels. This kit implements x16 SDRAM configurations only.

You can modify the kit to match your exact DDR4 implementation. Then, perform complete pre-layout solution space analysis and/or full post-layout verification for waveform quality and timing margins.

Open DDR4 MD Kit

Open the DDR4 MD kit in the **Parallel Link Designer** app using the `openSignalIntegrityKit` function.

```
openSignalIntegrityKit("DDR4_MD");
```


Parallel Link Designer: dr_x16.edk Project: C:\MATLAB\Examples\DDR4_MD

State: default
UI: 0.469ns
Type: Strobe
Topology: dqs_x16_2R

Solution Space:

Sheet Options: □ Case Mode □ STAT Mode □ Tx Aggressor Parameters

Transfer Net	Variable:	Type:	Format:	Variation Group:	Value 1:	Value 2:	Value 3:	Value 4:
<global>	$ADD_LENGTH	W Length	Soft Range	A	2.5in	3.5in		
<global>	$BREAKIN	W Length	Soft Range	A	0.050in			
<global>	$BREAKOUT	W Length	Soft Range	A	0.25in			
<global>	$CK_LENGTH	W Length	Soft Range	A	3.0in	3.0in		
<global>	$DQ_LENGTH	W Length	Soft Range	A	3.0in			
<global>	$DQS_LENGTH	W Length	Soft Range	A	3.0in			
dqs_x16_2R	Etch	Corner	List	Corners	SE (Slow)	TE (Typ)	FE (Fast)	
dqs_x16_2R	Process	Corner	List	Corners	SS (Slow)	TT (Typ)	FF (Fast)	

Kit Overview

- Project name: DDR4_MD
- Interface names: dr_x16 and sr_x16

This kit supports both HSPICE and IsSpice4 simulators. No specific version of either simulator is required when running this kit.

For more information about the DDR4 MD implementation signal integrity kit, including block diagrams, system configurations, transfer nets and libraries, refer to the document DDR4_MD.pdf that is attached to this example as a supporting file.

See Also

Parallel Link Designer

DDR5 Implementation Kit

Implement a 1-slot generic DDR5 RDIMM interface for pre-layout analysis or post-layout verification.

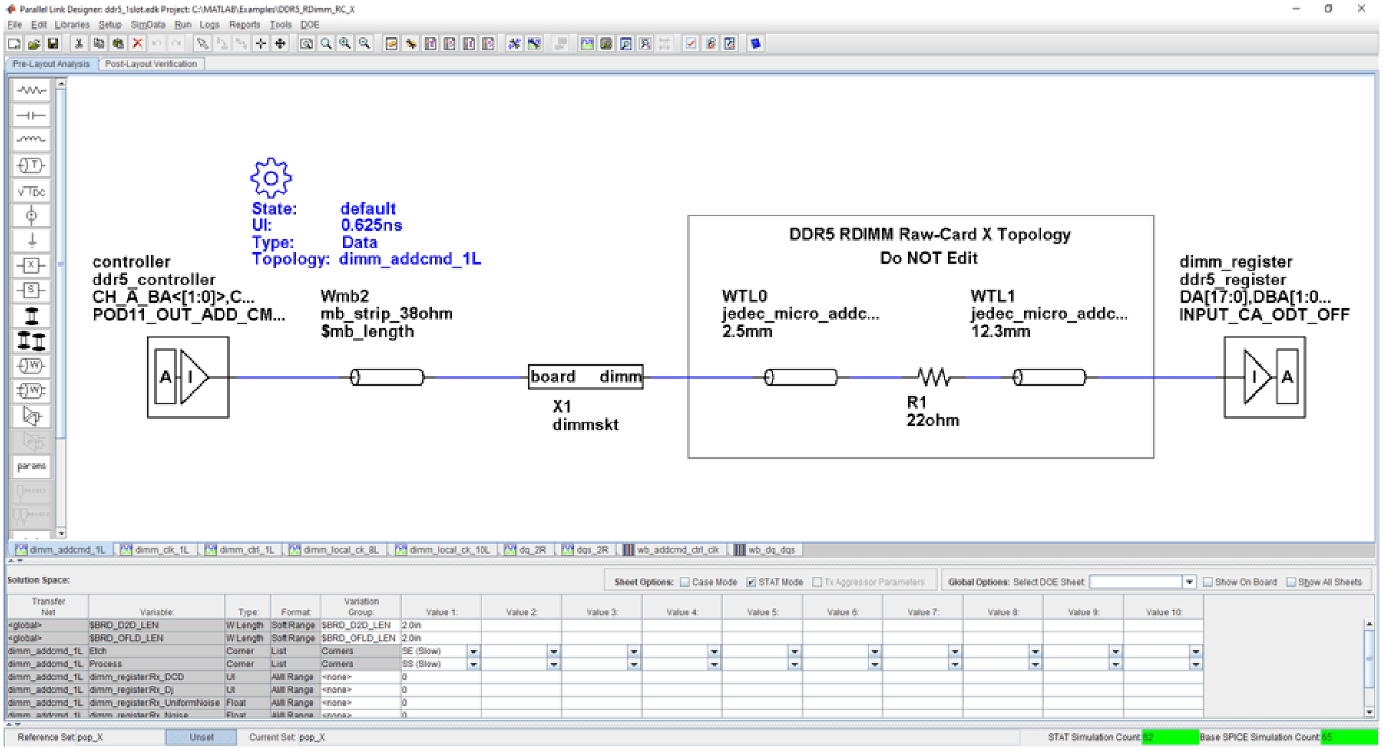
DDR5 is an industry standard dynamic memory format operating at a maximum of 6400M transfers per second. The standard is defined by JEDEC in the DDR5 JEDEC Specification JESD79-5.

This DDR5 implementation signal integrity kit includes all the transfer nets, waveform processing levels and simulation models for a 1-slot generic DDR5 RDIMM interface. This includes buffer models for a generic DDR5 controller, register and SDRAM, along with complete waveform processing levels. You can modify the kit to match your exact DDR5 implementation. Then, perform complete pre-layout solution space analysis and/or full post-layout verification for waveform quality and timing margins.

Open 1-Slot DDR5 Kit

Open the 1-slot DDR5 kit in the **Parallel Link Designer** app using the `openSignalIntegrityKit` function.

```
openSignalIntegrityKit("DDR5_RDimm_RC_X");
```


Parallel Link Designer: ddr5_1slot.edk Project: C:\MATLAB\Examples\DDR5_RDimm_RC_X
File Edit Libraries Setup SimData Run Logs Reports Tools DOE
Pre-Layout Analysis Post-Layout Verification
params
State: default
UI: 0.625ns
Type: Data
Topology: dimm_addcmd_1L
controller
ddr5_controller
CH_A_BA<[1:0]>,C...
POD11_OUT_ADD_CM...
A I
Wmb2
mb_strip_38ohm
$mb_length
board dimm
X1
dimmskt
DDR5 RDIMM Raw-Card X Topology
Do NOT Edit
WTL0
jedec_micro_addc...
2.5mm
R1
22ohm
WTL1
jedec_micro_addc...
12.3mm
dimm_register
ddr5_register
DA[17:0],DBA[1:0...
INPUT_CA_ODT_OFF
I A
dimm_addcmd_1L dimm_clk_1L dimm_ctrl_1L dimm_local_ck_8L dimm_local_ck_10L dq_2R dqs_2R wb_addcmd_ctrl_clk wb_dq_dqs

Solution Space:

Sheet Options: □ Case Mode ☑ STAT Mode □ Tx Aggressor Parameters

Global Options: Select DOE Sheet: □ Show On Board □ Show All Sheets

Transfer Net	Variable:	Type:	Format:	Variation Group:	Value 1:	Value 2:	Value 3:	Value 4:	Value 5:	Value 6:	Value 7:	Value 8:	Value 9:	Value 10:
<global>	$BRD_D2D_LEN	W Length	SoftRange	$BRD_D2D_LEN	2.0in									
<global>	$BRD_OFLD_LEN	W Length	SoftRange	$BRD_OFLD_LEN	2.0in									
dimm_addcmd_1L	Etch	Corner	List	Corners	SE (Slow)									
dimm_addcmd_1L	Process	Corner	List	Corners	SS (Slow)									
dimm_addcmd_1L	dimm_registerRx_DCD	UI	AMI Range	<none>	0									
dimm_addcmd_1L	dimm_registerRx_Dj	UI	AMI Range	<none>	0									
dimm_addcmd_1L	dimm_registerRx_UniformNoise	Float	AMI Range	<none>	0									
dimm_addcmd_1L	dimm_registerRx_Noise	Float	AMI Range	<none>	0									

Reference Set: pop_X Unset Current Set: pop_X

STAT Simulation Count: 62 Base SPICE Simulation Count: 65

Kit Overview

- Project name: DDR5_RDimm_RC_X
- Interface name: ddr5_1slot
- Two independent DDR5 channels in the generic controller: DDR5_0 and DDR5_1
- 72-bits per channel (64 data, 8 ECC)
- 288-pin buffered DDR5 RDIMM

- Address/Command 1N timing utilized (can be set up for 2N if desired)
- Data mask (DM) not used

The slot is populated with Raw Card “X” DDR5 Registered DIMM modules. Raw Card X used for setup and is not a JEDEC specified Raw Card.

- Number of SDRAMs: 18
- Package Type: Planar, 78 ball FPGA
- Number of Ranks: 2
- Width: x4

This kit supports both HSPICE and IsSpice4 simulators. No specific version of either simulator is required when running this kit.

For more information about the generic DDR5 using a mock Raw Card X RDIMM implementation signal integrity kit, including block diagrams, system configurations, transfer nets and libraries, refer to the document DDR5_RDimm_RC_X.pdf that is attached to this example as a supporting file.

References

[1] JEDEC: *DDR5 SDRAM*. JESD79-5, July 2020.

See Also

Parallel Link Designer

GDDR5 x32 Implementation Kit

Implement a 32-bit GDDR5 interface for pre-layout analysis or post-layout verification.

GDDR5 (double data rate type five) SGRAM (synchronous graphics random access memory) is a high bandwidth interface designed for use in graphics cards, game consoles and high-performance computing. GDDR5 interfaces are capable of speeds of 7 Gb/s, with the goal of reaching 8 Gb/s.

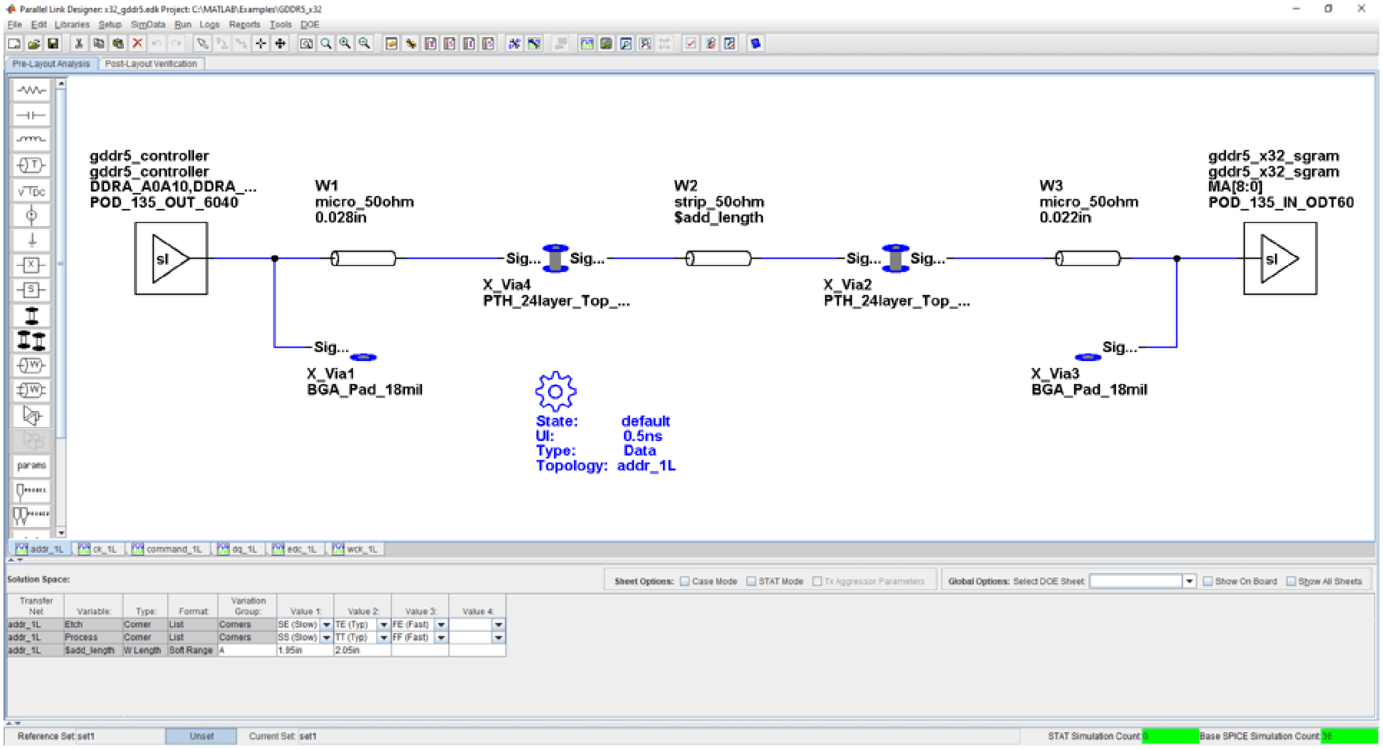
This GDDR5 implementation signal integrity kit includes all the transfer nets, timing models, waveform processing levels and simulation models for a GDDR5 x32 memory down interface. This includes buffer models for a generic GDDR5 controller and Micron x32 8 Gb SGRAM, along with timing models and complete waveform processing levels.

You can modify the kit to match your exact DDR5 implementation. Then, perform complete pre-layout solution space analysis and/or full post-layout verification for waveform quality and timing margins.

Open GDDR5 x32 Kit

Open the GDDR5 x32 kit in the **Parallel Link Designer** app using the `openSignalIntegrityKit` function.

```
openSignalIntegrityKit("GDDR5_x32");
```



Parallel Link Designer: x32_gddr5.edk Project: C:\MATLAB\Examples\GDDR5_x32

Solution Space:

Transfer Net	Variable:	Type:	Format:	Variation Group:	Value 1:	Value 2:	Value 3:	Value 4:
addr_1L	Etch	Corner	List	Corners	SE (Slow)	TE (Typ)	FE (Fast)	
addr_1L	Process	Corner	List	Corners	SS (Slow)	TT (Typ)	FF (Fast)	
addr_1L	$add_length	W Length	Soft Range	A	1.95in	2.05in		

Reference Set: set1 Unset Current Set: set1 STAT Simulation Count: 0 Base SPICE Simulation Count: 36

Kit Overview

- Project name: GDDR5_x32
- Interface name: x32_gddr5
- Target data rate: 4 Gb/s (UI = 250 ps)

This kit supports both HSPICE and IsSpice4 simulators. No specific version of either simulator is required when running this kit.

For more information about the generic GDDR5 x32 implementation signal integrity kit, including block diagrams, system configurations, transfer nets and libraries, refer to the document GDDR5_x32.pdf that is attached to this example as a supporting file.

References

[1] JEDEC: Graphics Double Data Rate (GDDR5) SGRAM Standard. JESD212B.01. December 2013.

[2] JEDEC: POD135 – 1.35V Pseudo Open Drain I/O. JESD8-21A. September 2013.

[3] Micron: GDDR5 SGRAM for Networking – MT51K256M32 – 16Meg x32 I/O x16 Banks, 32Meg x16 I/O x16 Banks. Rev. A 5/14 EN.

[4] Micron: Technical Note: GDDR5 SGRAM Introduction. Rev. A 2/14 EN.

See Also

Parallel Link Designer

GDDR6 x32 Architectural Kit

Implement a 32-bit GDDR6 interface for pre-layout analysis or post-layout verification.

GDDR6 (double data rate type six) SGRAM (synchronous graphics random access memory) is a high bandwidth interface designed for use in graphics cards, game consoles and high-performance computing. GDDR6 interfaces are capable of speeds up to 16 Gb/s.

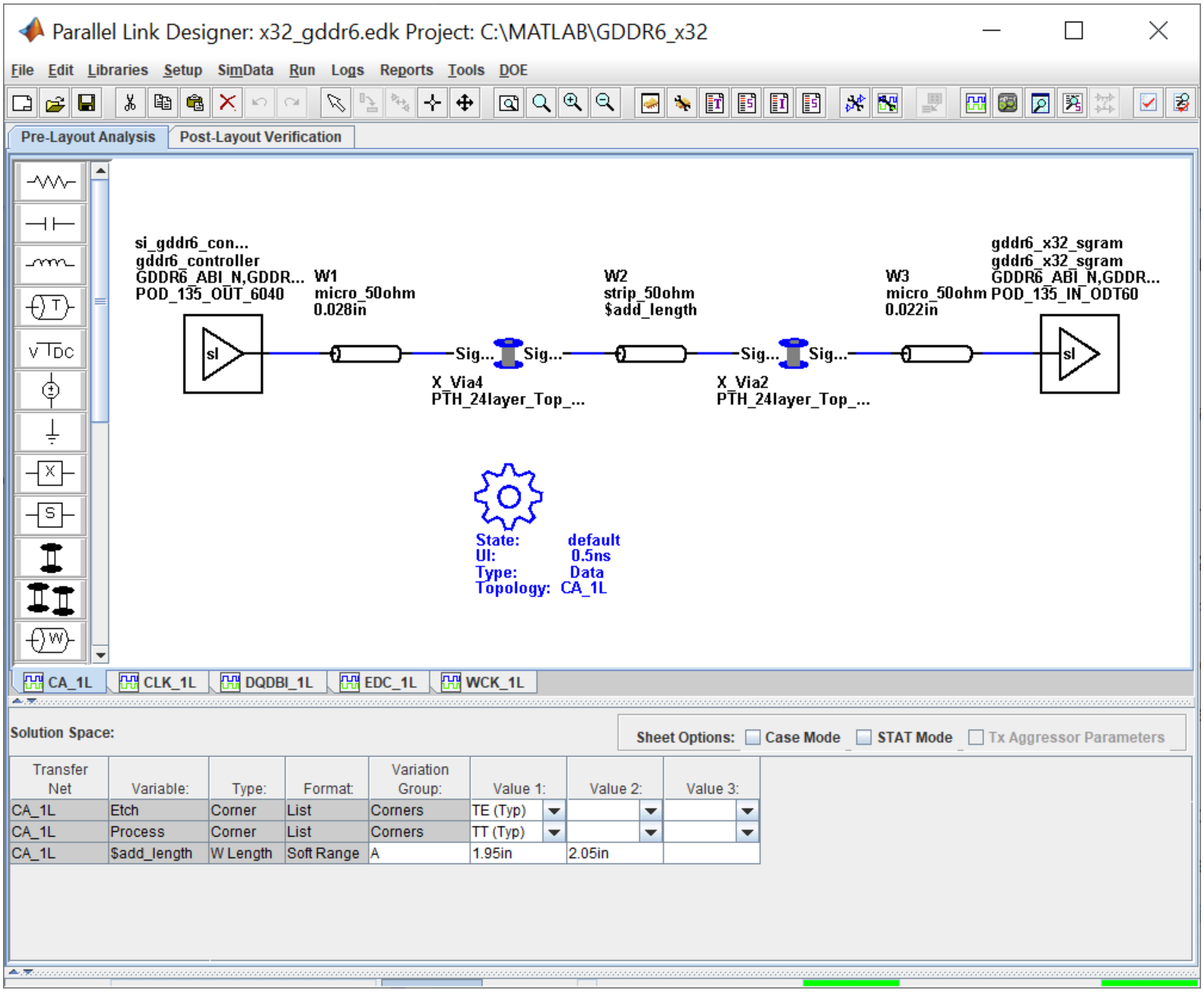
This GDDR6 architectural signal integrity kit includes all the transfer nets, waveform processing levels, generic timing and simulation models for a GDDR6 interface. This includes generic buffer models for the GDDR6 controller and SGRAM, along with functional timing models and complete waveform processing levels, all of which are easily customizable.

You can modify the kit to match your exact DDR6 implementation. Then, perform complete pre-layout solution space analysis and/or full post-layout verification for waveform quality and timing margins.

Open GDDR6 x32 Kit

Open the GDDR6 x32 kit in the **Parallel Link Designer** app using the `openSignalIntegrityKit` function.

```
openSignalIntegrityKit("GDDR6_x32");
```



Solution Space:

Sheet Options: □ Case Mode □ STAT Mode □ Tx Aggressor Parameters

Transfer Net	Variable:	Type:	Format:	Variation Group:	Value 1:	Value 2:	Value 3:
CA_1L	Etch	Corner	List	Corners	TE (Typ)		
CA_1L	Process	Corner	List	Corners	TT (Typ)		
CA_1L	$add_length	W Length	Soft Range	A	1.95in	2.05in	

Kit Overview

- Project name: GDDR6_x32
- Interface name: x32_gddr6
- Target data rate: 4 Gb/s (UI = 250 ps)
- 40 data bits per channel (32 data, 4 DBI and 4 EDC)
- 1.25V or 1.35V signaling selectable

This kit supports both HSPICE and IsSpice4 simulators. No specific version of either simulator is required when running this kit.

For more information about the generic GDDR6 x32 implementation signal integrity kit, including block diagrams, system configurations, transfer nets and libraries, refer to the document GDDR6_x32.pdf that is attached to this example as a supporting file.

References

[1] Graphics Double Data Rate 6 (GDDR6) SGRAM Standard." JEDEC. JESD250C. February 2021. https://www.jedec.org/standards-documents/docs/jesd250c.

See Also

Parallel Link Designer

Low-Power DDR4 Architectural Kit

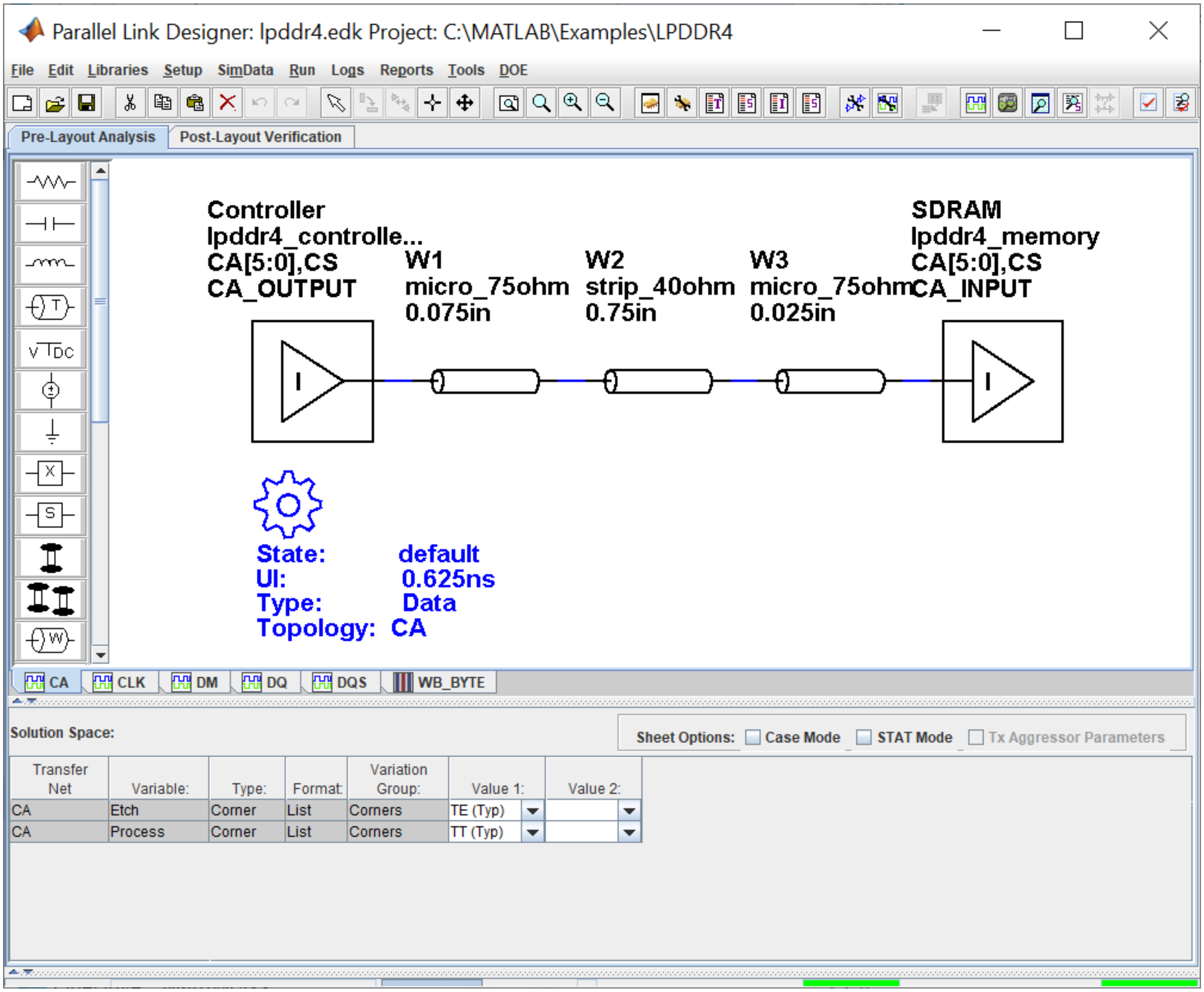
Implement a low-power DDR4 (LPDDR4) interface for pre-layout analysis or post-layout verification.

This LPDDR4 architectural signal integrity kit includes all the transfer nets, mask compliance checks, waveform processing levels and generic models for a LPDDR4 interface. This includes generic buffer models for the LPDDR4 controller and SDRAM, along with fully functional timing models and complete waveform processing levels. You can modify the kit to match your exact LPDDR4 implementation. Then, perform complete pre-layout solution space analysis and/or full post-layout verification for waveform quality and timing margins.

Open LPDDR4 Kit

Open the LPDDR4 kit in the **Parallel Link Designer** app using the `openSignalIntegrityKit` function.

```
openSignalIntegrityKit("LPDDR4");
```



State: default
UI: 0.625ns
Type: Data
Topology: CA

Solution Space:

Sheet Options: □ Case Mode □ STAT Mode □ Tx Aggressor Parameters

Transfer Net	Variable:	Type:	Format:	Variation Group:	Value 1:	Value 2:
CA	Etch	Corner	List	Corners	TE (Typ)	
CA	Process	Corner	List	Corners	TT (Typ)	

Kit Overview

For more information about the LPDDR4 architectural signal integrity kit, including block diagrams, system configurations, transfer nets and libraries, refer to the document LPDDR4.pdf that is attached to this example as a supporting file.

See Also

Parallel Link Designer

Low-Power DDR5 Architectural Kit

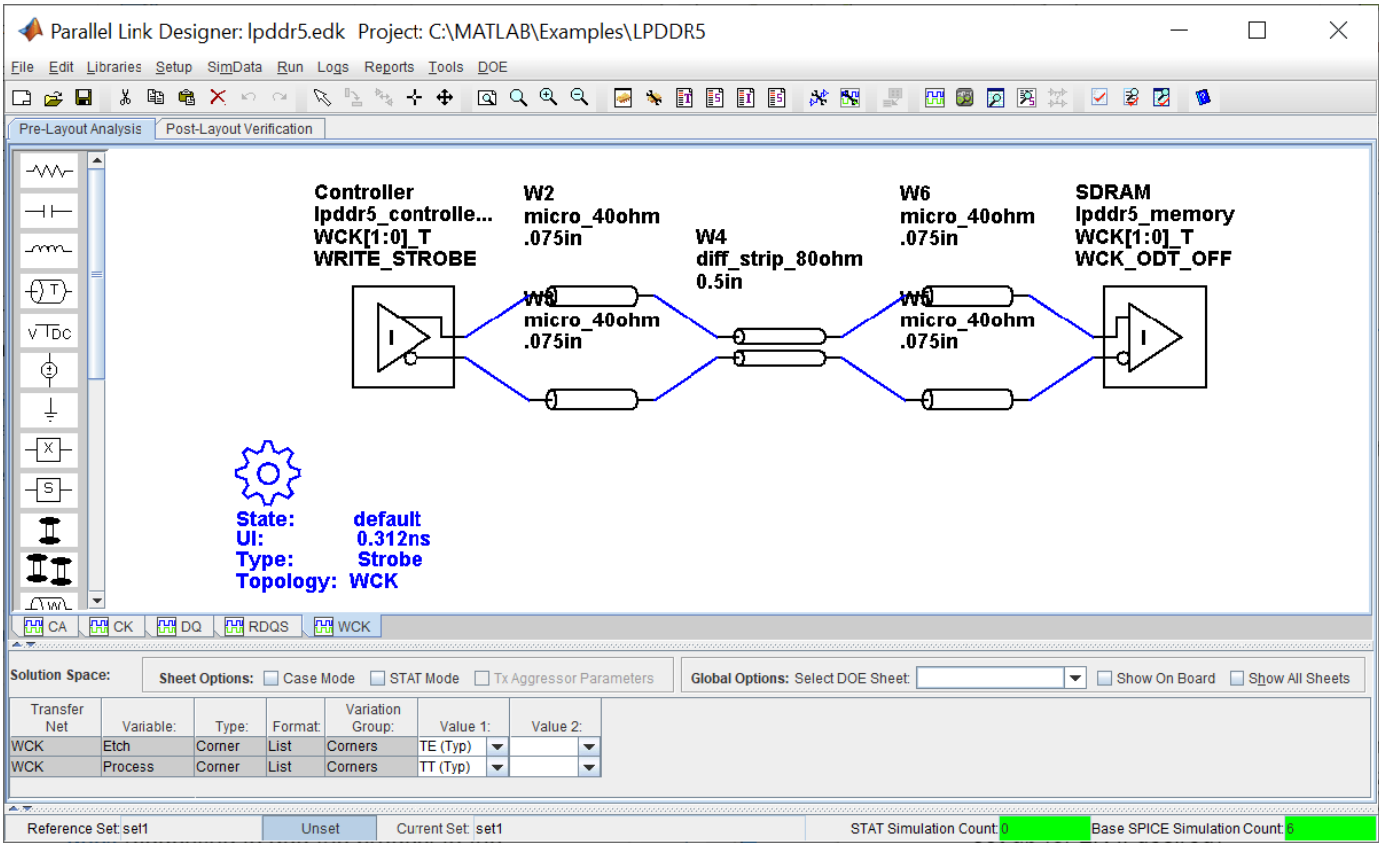
Implement a low-power DDR5 (LPDDR5) interface for pre-layout analysis or post-layout verification.

This LPDDR5 architectural signal integrity kit includes all the transfer nets, mask compliance checks, waveform processing levels and generic models for a LPDDR5 interface. This includes generic buffer models for the LPDDR5 controller and SDRAM, along with fully functional timing models and complete waveform processing levels. You can modify the kit to match your exact LPDDR5 implementation. Then, perform complete pre-layout solution space analysis and/or full post-layout verification for waveform quality and timing margins.

Open LPDDR5 Kit

Open the LPDDR5 kit in the **Parallel Link Designer** app using the `openSignalIntegrityKit` function.

```
openSignalIntegrityKit("LPDDR5");
```



Sheet Options: □ Case Mode □ STAT Mode □ Tx Aggressor Parameters

Global Options: Select DOE Sheet: □ Show On Board □ Show All Sheets

Transfer Net	Variable:	Type:	Format:	Variation Group:	Value 1:	Value 2:
WCK	Etch	Corner	List	Corners	TE (Typ)	
WCK	Process	Corner	List	Corners	TT (Typ)	

Reference Set: set1 | Unset | Current Set: set1 | STAT Simulation Count: 0 | Base SPICE Simulation Count: 6

Kit Overview

For more information about the LPDDR5 architectural signal integrity kit, including block diagrams, system configurations, transfer nets and libraries, refer to the document LPDDR5.pdf that is attached to this example as a supporting file.

See Also

Parallel Link Designer

MIPI D-PHY Parallel Link Compliance Kit

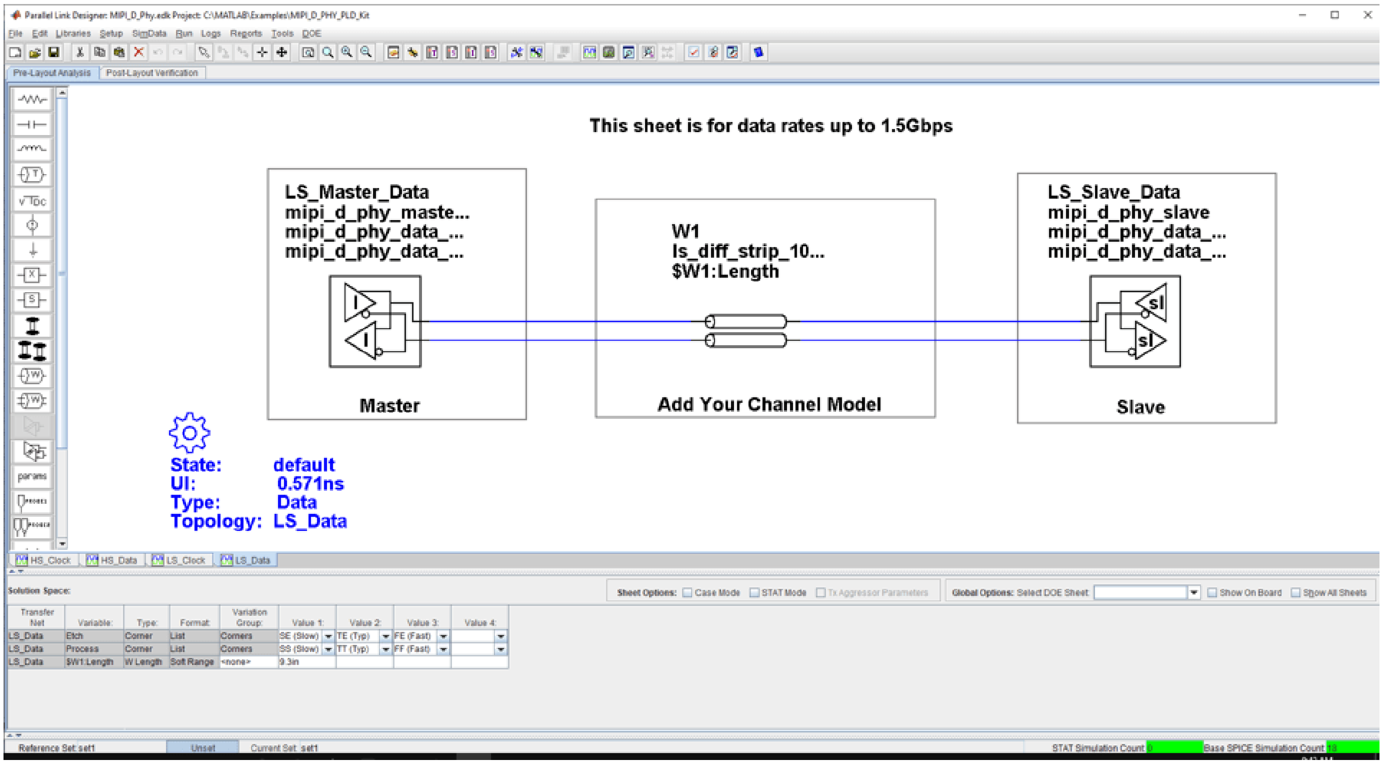
Test the compliance to the MIPI D-PHY specification with respect to clock-to-data timing in the forward direction and waveform quality in the reverse transmission using **Parallel Link Designer**.

This kit can be used for testing compliance with respect to the clock-to-data timing in the forward direction (Master to Slave). The reverse transmission (Slave to Master) operates at ¼ the data rate of the forward transmission. The specification is vague on timing requirements for these transfers and expects the logic implementation of the Master to train itself during the synchronous calibration mode to ensure timing is met for reverse transmission. Thus this kit is configured to test waveform quality only for reverse transmission. Lastly, this kit does not include any LP mode functionality.

Open MIPI D-PHY Kit

Open the MIPI D-PHY kit in the **Parallel Link Designer** app using the `openSignalIntegrityKit` function.

```
openSignalIntegrityKit("MIPI_D_PHY_PLD_Kit");
```



State: default
UI: 0.571ns
Type: Data
Topology: LS_Data

Sheet Options: □ Case Mode □ STAT Mode □ Tx Aggressor Parameters

Global Options: Select DOE Sheet □ Show On Board □ Show All Sheets

Solution Space:

Transfer Net	Variable:	Type:	Format:	Variation Group:	Value 1:	Value 2:	Value 3:	Value 4:
LS_Data	Etch	Corner	List	Corners	SE (Slow)	TE (Typ)	FE (Fast)	
LS_Data	Process	Corner	List	Corners	SS (Slow)	TT (Typ)	FF (Fast)	
LS_Data	$W1:Length	W Length	Soft Range	<none>	9.3in			

Kit Overview

- Project name: MIPI_D_PHY_PLD_Kit
- Interface name: MIPI_D_Phy

The MIPI D-PHY kit defines schematic sheets for high speed data rates above 1.5 Gb/s and low speed data rates less than or equal to 1.5 Gb/s. The kit can be simulated with either IsSPICE (the default) or HSPICE.

For more information about the MIPI D-PHY channel compliance schematics, transfer net properties, and compliance rules, refer to the document MIPI_D_Phy_PLD_Kit.pdf that is attached to this example as a supporting file.

References

[1] MIPI D-PHY Specification. mipi_D-PHY_specification_v1-2.pdf. Rev. 1.2.

See Also

Parallel Link Designer

Related Examples

- “MIPI D-PHY Serial Link Compliance Kit” on page 11-31

CIO RLDRAM II Architectural Kit

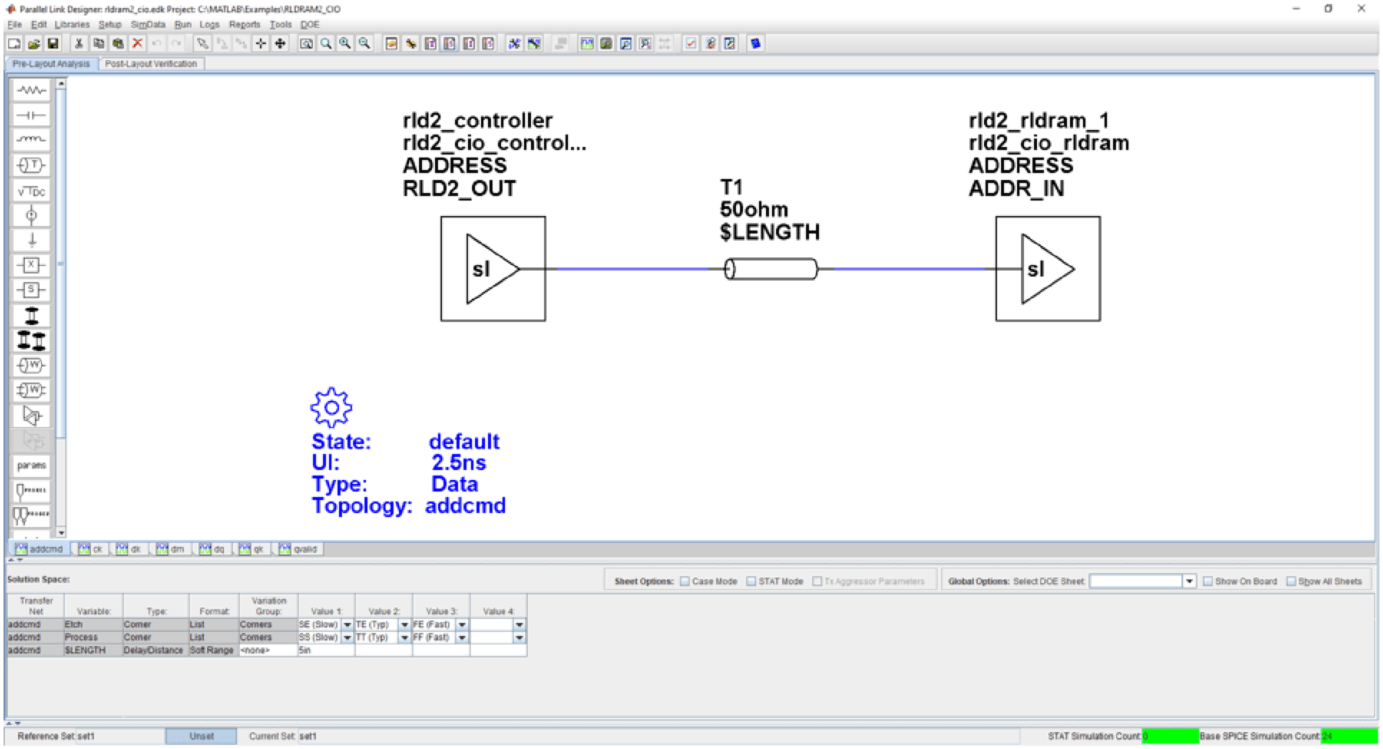
Implement a common I/O (CIO) RLDRAM II interface for pre-layout analysis or post-layout verification.

This CIO RLDRAM II architectural signal integrity kit includes all transfer nets, timing models, waveform processing levels and generic models for a CIO RLDRAM II interface. This includes generic buffer models for the controller and RLDRAM along with fully functional timing models and complete waveform processing levels. You can modify the kit to match your exact implementation. Then, perform complete pre-layout solution space analysis and/or full post-layout verification for waveform quality and timing margins.

Open CIO RLDRAM II Kit

Open the CIO RLDRAM II kit in the **Parallel Link Designer** app using the `openSignalIntegrityKit` function.

```
openSignalIntegrityKit("RLDRAM2_CIO");
```



State: default
UI: 2.5ns
Type: Data
Topology: addcmd

Solution Space:

Sheet Options: □ Case Mode □ STAT Mode □ Tx Aggressor Parameters

Global Options: Select DOE Sheet: □ Show On Board □ Show All Sheets

Transfer Net	Variable:	Type:	Format:	Variation Group:	Value 1:	Value 2:	Value 3:	Value 4:
addcmd	Etch	Corner	List	Corners	SE (Slow)	TE (Typ)	FE (Fast)	
addcmd	Process	Corner	List	Corners	SS (Slow)	TT (Typ)	FF (Fast)	
addcmd	$LENGTH	Delay/Distance	Soft Range	<none>	5in			

Reference Set: set1 Unset Current Set: set1

STAT Simulation Count: 0 Base SPICE Simulation Count: 24

Kit Overview

For more information about the CIO RLDRAM II architectural signal integrity kit, including block diagrams, system configurations, transfer nets and libraries, along with instructions on how to customize the kit for a specific implementation, refer to the document RLDRAM2_CIO.pdf that is attached to this example as a supporting file.

See Also

Parallel Link Designer

SIO RLDRAM II Architectural Kit

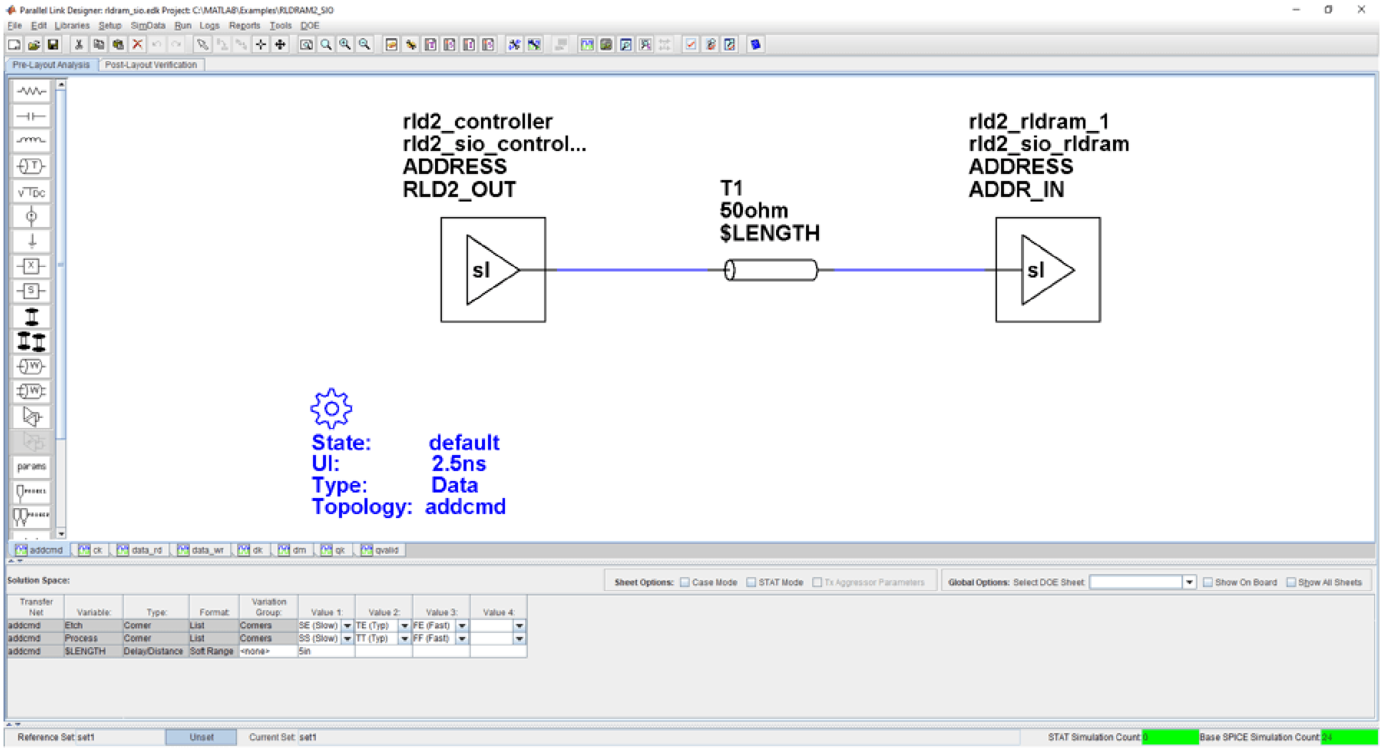
Implement a separate I/O (SIO) RLDRAM II interface for pre-layout analysis or post-layout verification.

This SIO RLDRAM II architectural signal integrity kit includes all transfer nets, timing models, waveform processing levels and generic models for an SIO RLDRAM II interface. This includes generic buffer models for the controller and RLDRAM along with fully functional timing models and complete waveform processing levels. You can modify the kit to match your exact implementation. Then, perform complete pre-layout solution space analysis and/or full post-layout verification for waveform quality and timing margins.

Open SIO RLDRAM II Kit

Open the SIO RLDRAM II kit in the **Parallel Link Designer** app using the `openSignalIntegrityKit` function.

```
openSignalIntegrityKit("RLDRAM2_SIO");
```


Parallel Link Designer: rldram_sio.edk Project: C:\MATLAB\Examples\RLDRAM2_SIO
File Edit Libraries Setup SimData Run Logs Reports Tools DOE
Pre-Layout Analysis
Post-Layout Verification
rld2_controller
rld2_sio_control...
ADDRESS
RLD2_OUT
sl
T1
50ohm
$LENGTH
rld2_rldram_1
rld2_sio_rldram
ADDRESS
ADDR_IN
sl
State: default
UI: 2.5ns
Type: Data
Topology: addcmd
params
addcmd
ck
data_rd
data_wr
dk
dm
qk
qvalid
Sheet Options: □ Case Mode □ STAT Mode □ Tx Aggressor Parameters
Global Options: Select DOE Sheet □ Show On Board □ Show All Sheets

Solution Space:

Transfer Net	Variable:	Type:	Format:	Variation Group:	Value 1:	Value 2:	Value 3:	Value 4:
addcmd	Etch	Corner	List	Corners	SE (Slow)	TE (Typ)	FE (Fast)	
addcmd	Process	Corner	List	Corners	SS (Slow)	TT (Typ)	FF (Fast)	
addcmd	$LENGTH	Delay/Distance	Soft Range	<none>	5in			

Reference Set: set1 Unset Current Set: set1 STAT Simulation Count: 0 Base SPICE Simulation Count: 24

Kit Overview

For more information about the SIO RLDRAM II architectural signal integrity kit, including block diagrams, system configurations, transfer nets and libraries, along with instructions on how to customize the kit for a specific implementation, refer to the document RLDRAM2_SIO.pdf that is attached to this example as a supporting file.

See Also

Parallel Link Designer

RLDRAM III Architectural Kit

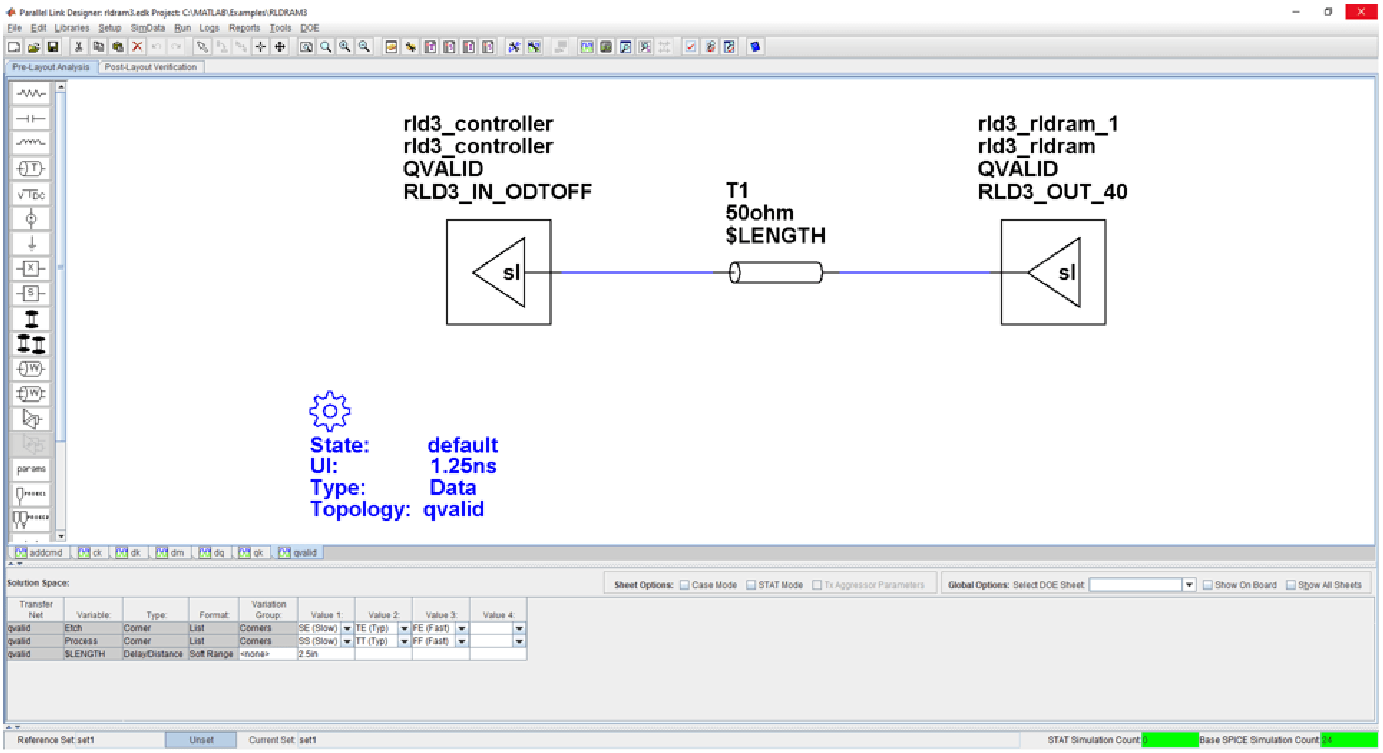
Implement a RLDRAM III interface for pre-layout analysis or post-layout verification.

This RLDRAM III architectural signal integrity kit includes all transfer nets, timing models, waveform processing levels and generic models for a RLDRAM III interface. This includes generic buffer models for the controller and RLDRAM along with fully functional timing models and complete waveform processing levels. You can modify the kit to match your exact implementation. Then, perform complete pre-layout solution space analysis and/or full post-layout verification for waveform quality and timing margins.

Open RLDRAM III Kit

Open the RLDRAM III kit in the **Parallel Link Designer** app using the `openSignalIntegrityKit` function.

```
openSignalIntegrityKit("RLDRAM3");
```



State: default
UI: 1.25ns
Type: Data
Topology: qvalid

Solution Space:

Sheet Options: □ Case Mode □ STAT Mode □ Tx Aggressor Parameters

Global Options: Select DOE Sheet □ Show On Board □ Show All Sheets

Transfer Net	Variable:	Type:	Format:	Variation Group:	Value 1:	Value 2:	Value 3:	Value 4:
qvalid	Etch	Corner	List	Corners	SE (Slow)	TE (Typ)	FE (Fast)	
qvalid	Process	Corner	List	Corners	SS (Slow)	TT (Typ)	FF (Fast)	
qvalid	$LENGTH	Delay/Distance	Soft Range	<none>	2.5in			

Reference Set: set1

Current Set: set1

STAT Simulation Count: 0

Base SPICE Simulation Count: 24

Kit Overview

For more information about the RLDRAM III architectural signal integrity kit, including block diagrams, system configurations, transfer nets and libraries, along with instructions on how to customize the kit for a specific implementation, refer to the document RLDRAM3.pdf that is attached to this example as a supporting file.

References

[1] Micron – 576Mb: x18, x36 RLDRAM3 Features (Advance datasheet). 576_rldram3.pdf – Rev. B 1/12 EN.

[2] Micron – TN-44-01: Technical Note, RLDRAM3 Design Guide.
TN_44_01_RLDRAM_3_Design_Guide.fm – Rev. A 8/11 EN.

See Also

Parallel Link Designer

Run Parallel Simulations in Signal Integrity Toolbox

You can easily generate many thousands of SPICE or Channel Analysis simulations using the **Serial Link Designer** and **Parallel Link Designer** apps. By default, the apps run all simulations sequentially on the local computer, which can take a significant amount of time to complete. However, if you have a Parallel Computing Toolbox license, then you can run multiple simulations in parallel and considerably reduce the time required to run the complete set of simulations.

Using the default Parallel Computing Toolbox settings, most users can run parallel simulations quickly and efficiently without making any changes to the settings. However, in some cases, running simulations with the default settings on your local machine can reduce the interactive performance and impede your ability to do other work at the same time. Likewise, running too many simulations at once on a multiuser machine can negatively impact other users. If you find that the default settings negatively impact you or other users on a shared machine, then you can modify the parallel computing settings.

Here are some useful parallel computing concepts:

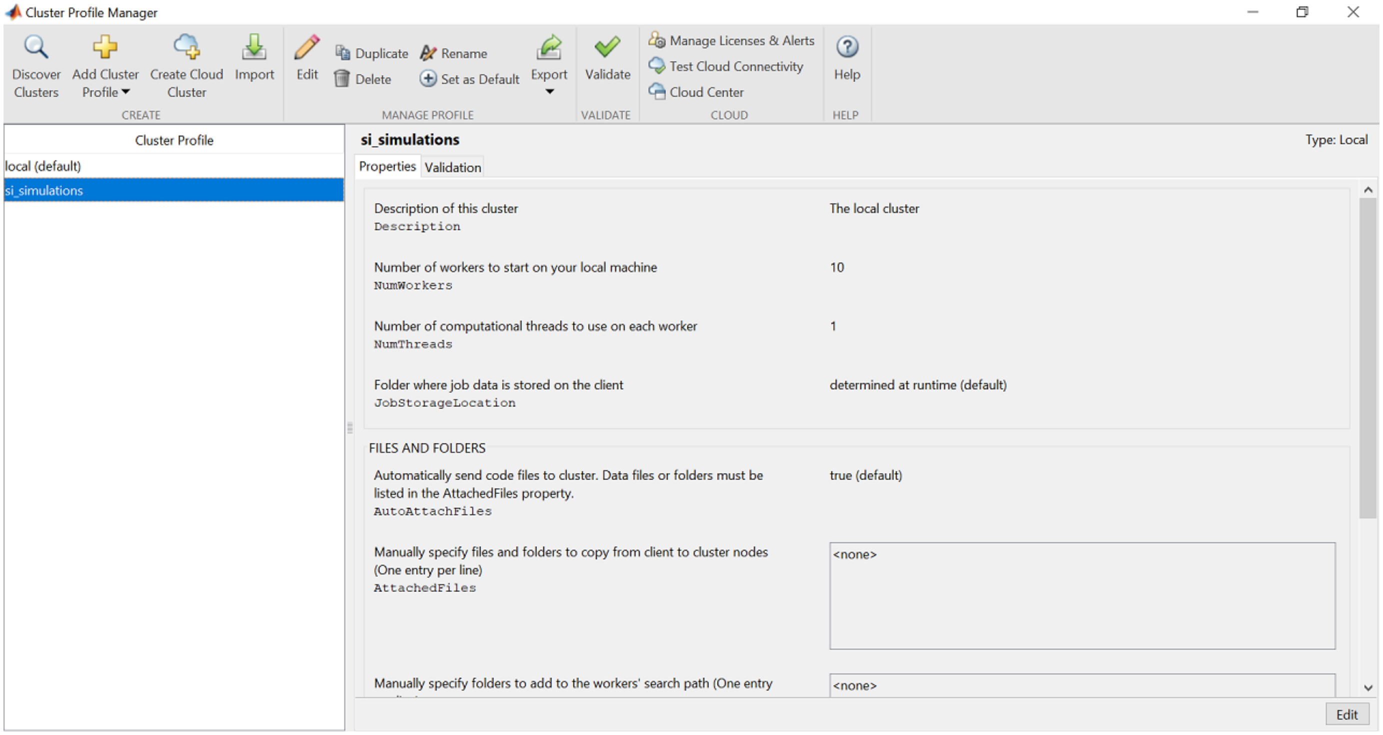
- Task: A list of operations. In Signal Integrity Toolbox™, these operations are individual simulations. Each task can consist of one or multiple simulations run sequentially.
- Cluster: The location where tasks will be performed. A cluster can be a single machine that can execute multiple threads simultaneously, such as a multiprocessor or multicore system, or a system with one or more CPUs. A cluster can also consist of a group of remote machines.
- Worker: A MATLAB® computational engine that runs in the background without a graphical desktop.
- Parallel pool: A set of MATLAB workers running in parallel on a cluster.

Configure Local MATLAB Cluster for Parallel Simulations

Using the default Parallel Computing Toolbox settings, the built-in local MATLAB cluster uses all available cores (or logical processors) on a machine.

When performing local simulations, Parallel Computing Toolbox uses a parallel pool. Each worker in a parallel pool launches its own instance of MATLAB. Therefore, the workers can use a significant amount of memory when running. Make sure that a minimum of 4 GB RAM per worker is available to avoid an impact on the overall performance of the target machine.

You can adjust the number of workers used in parallel simulations using the Cluster Profile Manager. This figure shows Cluster Profile Manager with a custom cluster called **si_simulations** that uses ten workers.



si_simulations

Type: Local

Description of this cluster
`Description`
The local cluster

Number of workers to start on your local machine
`NumWorkers`
10

Number of computational threads to use on each worker
`NumThreads`
1

Folder where job data is stored on the client
`JobStorageLocation`
determined at runtime (default)

FILES AND FOLDERS

Automatically send code files to cluster. Data files or folders must be listed in the AttachedFiles property.
`AutoAttachFiles`
true (default)

Manually specify files and folders to copy from client to cluster nodes (One entry per line)
`AttachedFiles`
<none>

Manually specify folders to add to the workers' search path (One entry
<none>

Open the Cluster Profile Manager from the MATLAB toolstrip by selecting the **Parallel** drop-down list from the **Environment** tab, then select **Create and Manage Clusters**. Follow these steps to add and configure a cluster suitable for your signal integrity simulations in the Cluster Profile Manager:

1. Highlight the **local** Cluster Profile
2. Select Duplicate from the toolbar menu. This action creates a copy of the local profile named **local_Copy**.
3. Rename the **local_Copy** profile to **si_simulations**or to a name you prefer by double-clicking the profile name and editing the text box.
4. With the **si_simulations** profile highlighted, click the **Edit** button to modify the **si_simulations** profile.
5. Change the number of workers (`NumWorkers`) text field to the desired number of workers. A good starting point is 4 GB per worker. For example, on a machine with 12 logical processors and 64 GB of memory, setting the number of workers to 10 should allow for good interactive performance without using all the resources on the machine. However, on the same machine with only 32 GB of memory, setting the number of workers to 6 will keep you from running out of memory.

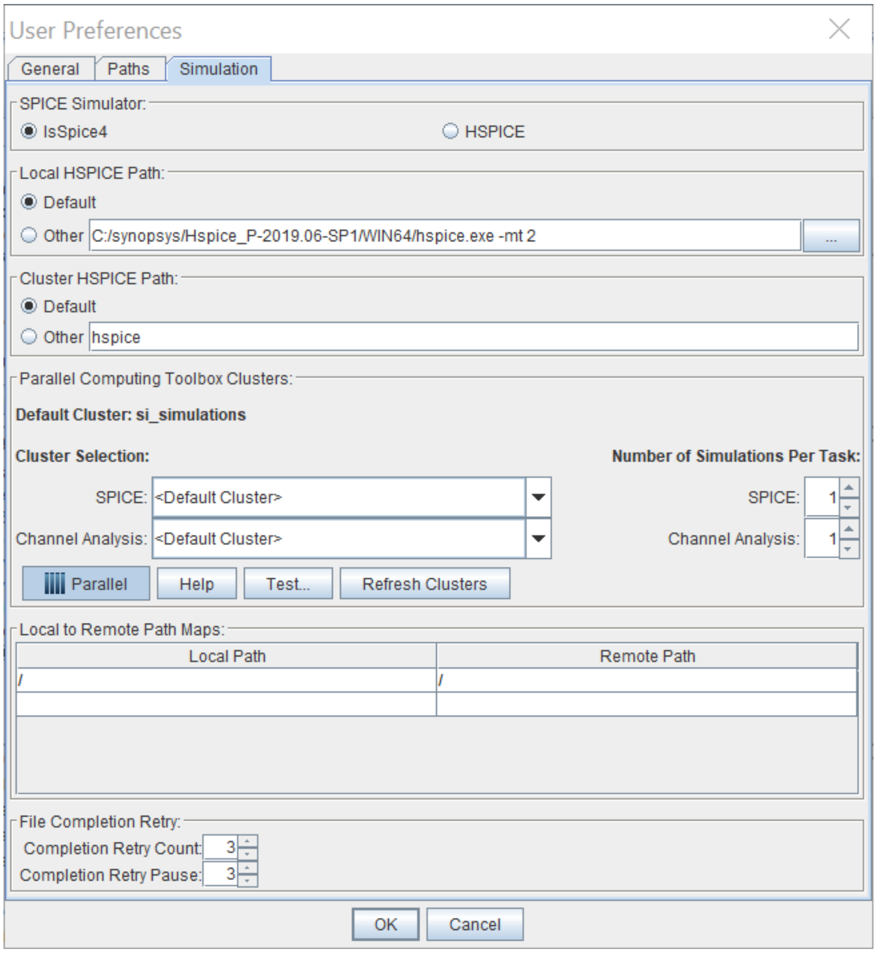
   When running on a remote cluster, use a similar approach. However, if the remote machine is shared across multiple users, reduce the number of workers to allow good performance for all users. You may need to adjust this number in your cluster based on the memory and CPU demands on your cluster and the machine hosting your cluster.

   Click **Done** to save the changes.
6. With **si_simulations** highlighted, click the **Set as Default** button in the toolbar menu.
7. Finally, test that everything is working correctly by clicking the **Validate** button. If everything works, you can close the Cluster Profile Manager.

Parallel Computing Toolbox supports many different cluster types such as Microsoft® Windows® HPC Server or IBM Spectrum LSF®. For more information about configuring these clusters, see “Discover Clusters and Use Cluster Profiles” (Parallel Computing Toolbox) and “Get Started with MATLAB Parallel Server” (MATLAB Parallel Server).

Adjust Cluster Settings for Signal Integrity Toolbox

To edit the cluster settings in the **Serial Link Designer** and **Parallel Link Designer** apps, select **Setup > User Preferences**, then select the **Simulation** tab.



SPICE Simulator:

● IsSpice4 ○ HSPICE

Local HSPICE Path:

● Default

○ Other C:/synopsys/Hspice_P-2019.06-SP1/WIN64/hspice.exe -mt 2

Cluster HSPICE Path:

● Default

○ Other hspice

Parallel Computing Toolbox Clusters:

Default Cluster: si_simulations

Cluster Selection:

SPICE: <Default Cluster>

Channel Analysis: <Default Cluster>

Number of Simulations Per Task:

SPICE: 1

Channel Analysis: 1

Local to Remote Path Maps:

Local Path	Remote Path
/	/

File Completion Retry:

Completion Retry Count: 3

Completion Retry Pause: 3

SPICE Simulator

For **Parallel Link Designer**, select either the IsSpice4 or HSPICE simulator using the radio buttons. **Serial Link Designer** only supports HSPICE simulations.

Signal Integrity Toolbox provides an unlimited number of IsSpice4 licenses. The only limit on the number of IsSpice4 and Channel Analysis simulations that can be run in parallel is the size of your cluster.

Running HSPICE simulations requires a separate HSPICE license and installation. The number of HSPICE simulations is limited to the number of HSPICE licenses that you have.

HSPICE Paths

You can specify the path to the HSPICE executable whether running a single simulation at a time locally or many simulations in parallel on a cluster. The **Default** setting picks the version of HSPICE specified by the HSPICE system environment variables. The **Other** setting allows a specific version of HSPICE to be used when multiple versions are installed.

You can add additional HSPICE flags using the **Other** path. For example, to enable multithreading, use: `C:/synopsys/Hspice_P-2019.06-SP1/WIN64/hspice.exe -mt 2`.

Parallel Computing Toolbox Clusters

You can specify different clusters for SPICE and Channel Analysis simulations. If you do not require different clusters for SPICE and Channel Analysis and will not concurrently run other MATLAB functions (such as `parfor`) on a parallel cluster while you are running simulations, then select **<Default Cluster>** for both SPICE and Channel Analysis.

Choose the default cluster from the MATLAB toolstrip by selecting the **Parallel** > **Select a Default Cluster** drop-down list.

The **Parallel** button enables and disables parallel simulations. The **Test** button is similar to the Validate function in the Cluster Profile Manager, but also includes some additional tests specific to Signal Integrity Toolbox. Test the final setup to verify that everything is set up and working correctly.

Number of Simulations Per Task

The **Number of Simulations Per Task** specifies how many simulations are submitted to a worker. By default, a single simulation is sent to each worker. When a worker completes a simulation, a new single simulation is sent to that worker. Although the overhead of this process is low, when running very fast simulations (less than 1 second per simulation), it can be advantageous to submit multiple simulations to a worker. Unless you are consistently running very fast simulations, leave this setting at `1`.

Local to Remote Path Maps

The local path refers to a network drive as seen from the local machine. The remote path refers to the path as seen from the cluster machine. Map the local path to the remote path when using a queuing system such as MATLAB Job Scheduler (MJS) or IBM Spectrum LSF®. The project must be on a network drive that is accessible by both the local machine and the remote machine. For example, a network drive mapped as `Z:/` on a local Windows machine might be seen as `/hw/projects` from a remote Linux machine that runs the remote simulation.

Path maps are not required when running local parallel pools or Microsoft Windows HPC Server clusters.

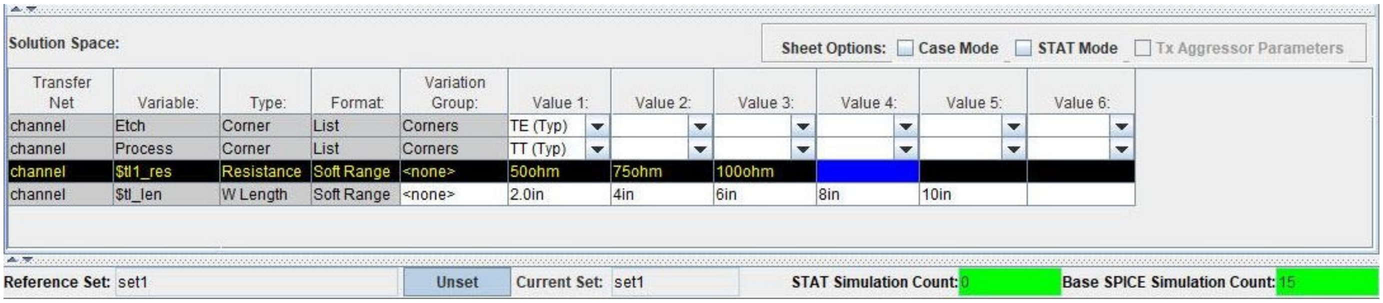
File Completion Retry

Completion Retry Count specifies how many times Signal Integrity Toolbox will retry its completion check before flagging a simulation as incomplete and failed. **Completion Retry Pause** specifies the delay, in seconds, between each completion check retry.

In some remote server and remote disk environments, the simulation output files are not completely written to disk when the simulation completes. If you see simulation errors when simulations appear to have run to completion, try increasing the value of these two parameters.

Run Parallel Simulations

First, set up the simulation parameters to populate the solution space. For example, this figure shows the Solution Space panel with parameters set up according to the example “Analyze Parallel Links with Parallel Link Designer”. The **Base SPICE Simulation Count** shows that there are 15 simulations.



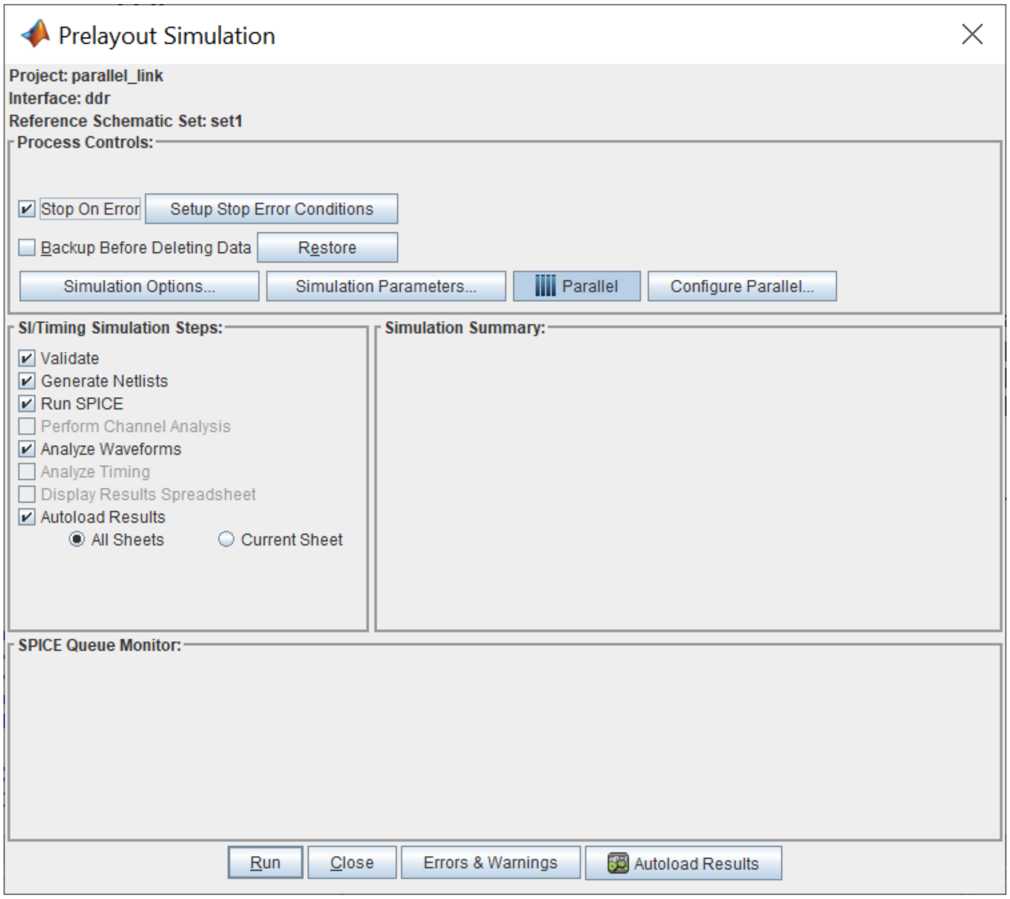
Solution Space:

Sheet Options: □ Case Mode □ STAT Mode □ Tx Aggressor Parameters

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	$tl1_res	Resistance	Soft Range	<none>	50ohm	75ohm	100ohm			
channel	$tl_len	W Length	Soft Range	<none>	2.0in	4in	6in	8in	10in	

Reference Set: set1 Unset Current Set: set1 STAT Simulation Count: 0 Base SPICE Simulation Count: 15

To run the simulations, select **Run > Simulated Selected**. The Prelayout Simulation dialog box opens.



Prelayout Simulation

Project: parallel_link
Interface: ddr
Reference Schematic Set: set1

Process Controls:

☑ Stop On Error
□ Backup Before Deleting Data

SI/Timing Simulation Steps:

☑ Validate
☑ Generate Netlists
☑ Run SPICE
□ Perform Channel Analysis
☑ Analyze Waveforms
□ Analyze Timing
□ Display Results Spreadsheet
☑ Autoload Results
● All Sheets ○ Current Sheet

Simulation Summary:

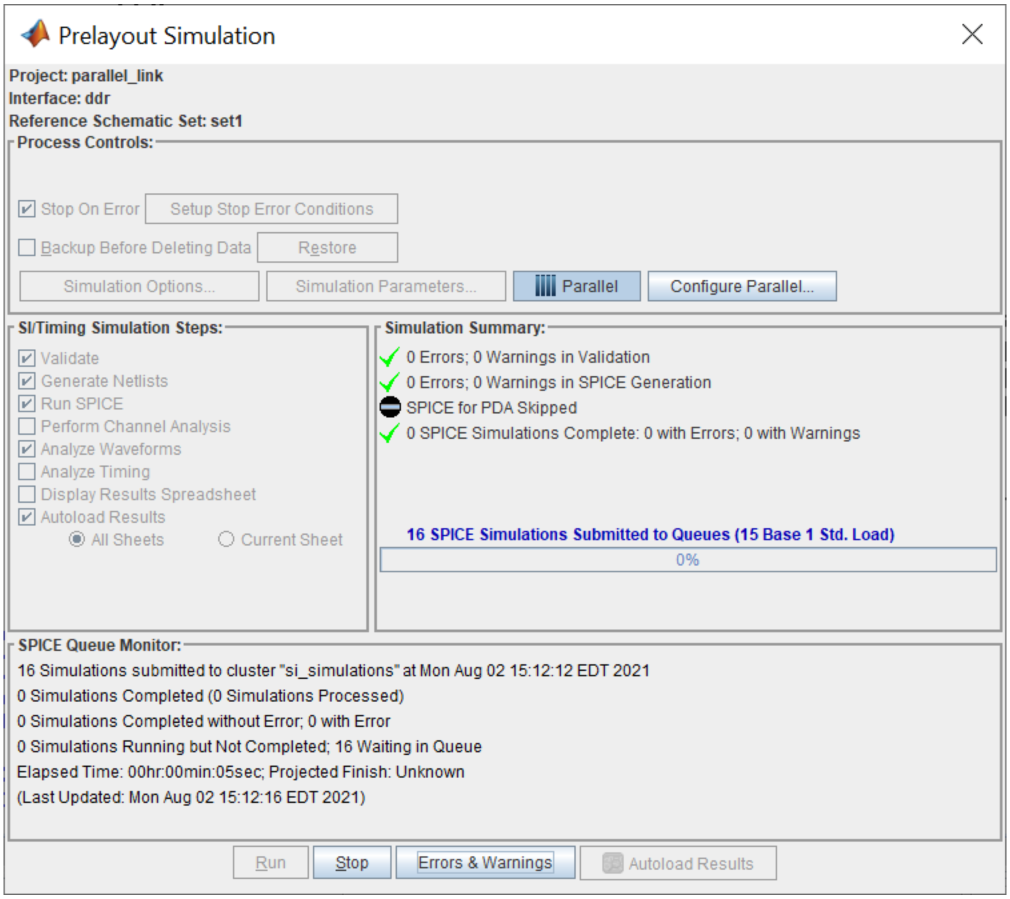
SPICE Queue Monitor:

If you have Parallel Computing Toolbox, then the **Parallel** button should already be selected. Toggle this selection to enable and disable the parallel simulations. With the **Parallel** button enabled, launch the simulations by clicking **Run**. There is a short delay while the parallel pool starts.

Note: The startup time of the parallel pool is only required for the first set of simulations. After the parallel pool is up and running, subsequent simulations will launch immediately. By default, the parallel pool will remain up for 30 minutes. This value can be adjusted using the Parallel Computing Toolbox Preferences dialog box.

You can monitor the simulations using the SPICE Queue Monitor panel of the Prelayout Simulation dialog box. The SPICE Queue Monitor shows this information:

- Total number of simulations submitted
- Number of simulations completed
- Number of simulations completed without errors
- Number of simulations currently running
- Number of simulations still waiting in the queue
- Total elapsed time
- Project finish time based on the number of simulations submitted and the time taken by the already completed simulations to run



Project: parallel_link
Interface: ddr
Reference Schematic Set: set1

Process Controls:

☑ Stop On Error
□ Backup Before Deleting Data

SI/Timing Simulation Steps:

☑ Validate
☑ Generate Netlists
☑ Run SPICE
□ Perform Channel Analysis
☑ Analyze Waveforms
□ Analyze Timing
□ Display Results Spreadsheet
☑ Autoload Results
● All Sheets ○ Current Sheet

Simulation Summary:

✓ 0 Errors; 0 Warnings in Validation
✓ 0 Errors; 0 Warnings in SPICE Generation
⊖ SPICE for PDA Skipped
✓ 0 SPICE Simulations Complete: 0 with Errors; 0 with Warnings

16 SPICE Simulations Submitted to Queues (15 Base 1 Std. Load)
0%

SPICE Queue Monitor:

16 Simulations submitted to cluster "si_simulations" at Mon Aug 02 15:12:12 EDT 2021
0 Simulations Completed (0 Simulations Processed)
0 Simulations Completed without Error; 0 with Error
0 Simulations Running but Not Completed; 16 Waiting in Queue
Elapsed Time: 00hr:00min:05sec; Projected Finish: Unknown
(Last Updated: Mon Aug 02 15:12:16 EDT 2021)

You can also track the status of simulations using the Job Monitor in MATLAB. The Job Monitor shows the number of tasks that can currently be run in the parallel pool (equivalent to the number of workers here) and the current state of the parallel pool. The description in this example reads Interactive Pool and the State reflects the current state of the parallel pool and not the state of the current set of simulations. Entries in the Job Monitor are never automatically purged and will accumulate over time. You can periodically delete old entries by selecting them, right-clicking, and selecting **Delete**. For more information, see “Job Monitor” (Parallel Computing Toolbox).

See Also

Related Examples

- “Analyze Serial Links with Serial Link Designer”
- “Analyze Parallel Links with Parallel Link Designer”

External Websites

- Product Requirements & Platform Availability for Parallel Computing Toolbox

12

Signal Integrity Topics

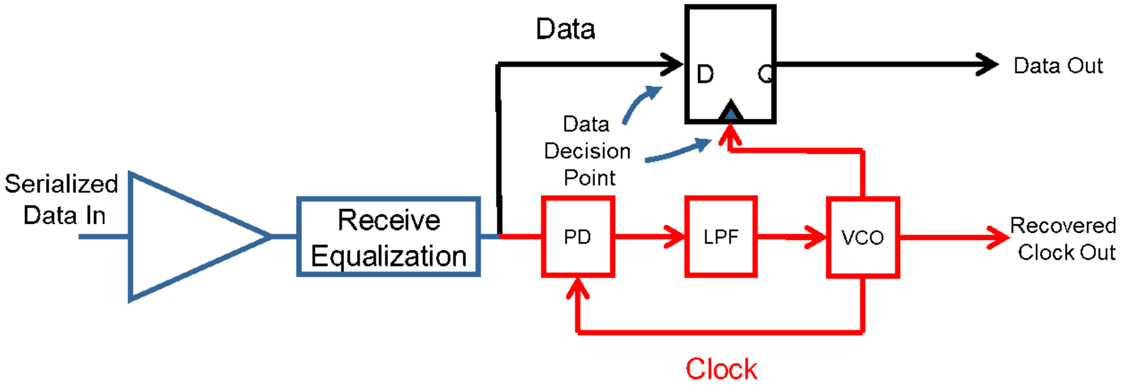
Clock Modes

The IBIS-AMI specification standardizes the way AMI models return waveform and clock data to an AMI simulator. But the standard says nothing about how the simulator should process that data. You can select from three different clock modes. Each clock mode is based on how the recovered clock and data distributions are used in the simulation to produce eye diagrams and predict bit error rates. Understanding and applying each of these modes effectively provides additional insight as to how, and where, a design's operating margins are being affected and what to do about it. Each of the three clock modes are defined and presented in both time domain and statistical analysis along with their benefits and applicability.

Note In **Parallel Link Designer** app, clock modes are only selectable when simulating in STAT mode. As such, the STAT mode box must be checked on all sheets that are to be simulated using a particular clock mode.

Clock and Data Paths in Serial Link Receiver

A serial link receiver design is the starting point to the understanding of QCD clock modes. The focus is on the relationship of the clock and data paths leading to the data decision latch.

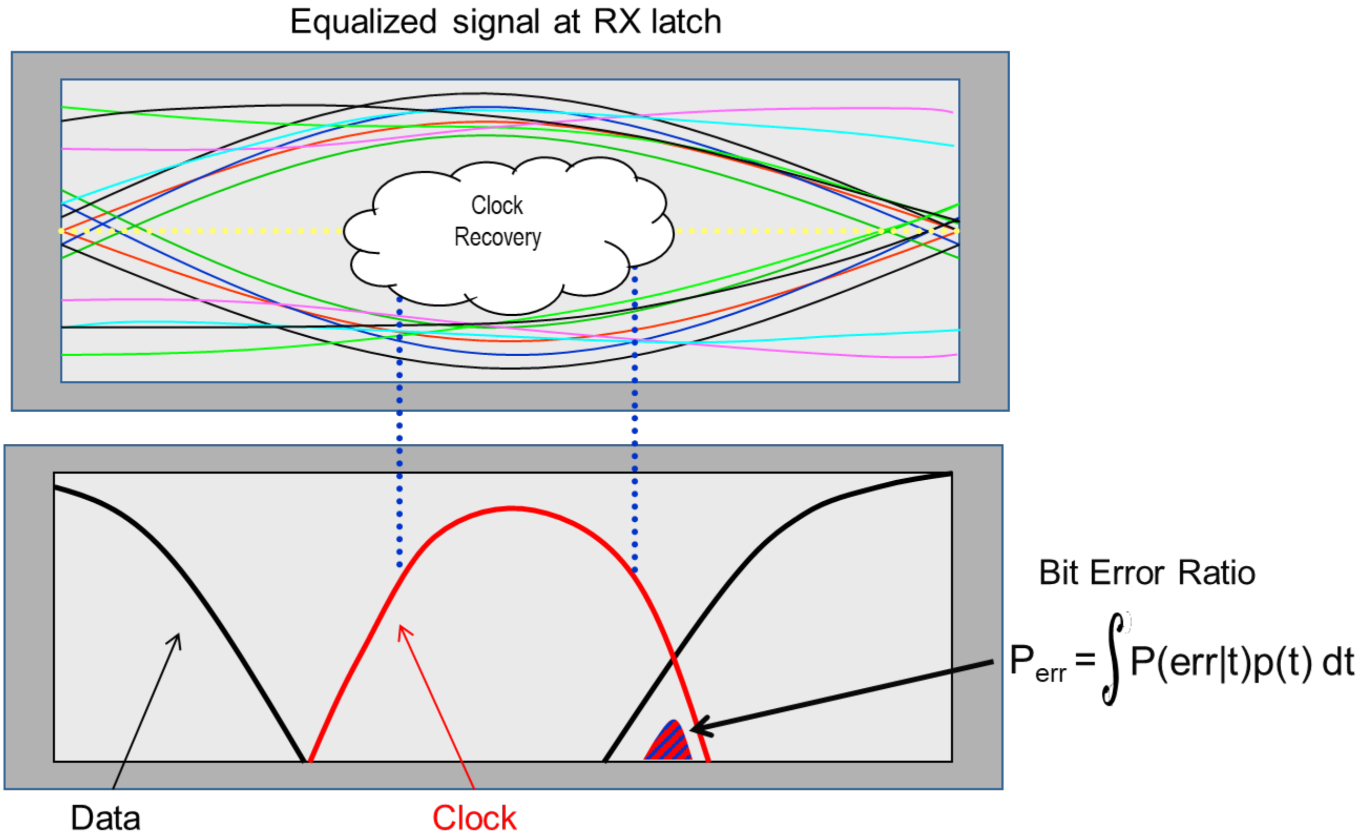


In a typical serial link receiver, the CDR is used to recover the clock signal from the serialized data stream. The recovered clock from the CDR does two things: it provides the clock signal that the sampling latch uses to capture the data, and it tells the DFE when to perform corrections to the data.

An ideal CDR having infinite bandwidth would provide a clock tick to the latch that is based on the data bit being latched. The dependency between the data and clock signals in this ideal CDR would be total. However, real world CDRs have limited bandwidth. Thus, the recovered clock being fed to the latch from the CDR is based upon a number of previous data bits. That means the clock tracks variations in the input signal, but only if they occur slowly enough for the CDR to respond to them. If the variations in the input edges are too fast for the CDR to respond, the CDR will track the only the average input rate and not respond to high frequency jitter. In this case, the variations in data and CDR behavior will be independent, as they both may vary but they won't vary in unison. So, data and clock under the right circumstances can be treated as independent variables.

Using Data and Clock Distribution to Predict Bit-Error Rate

The standard metric used to describe channel performance is the bit-error ratio (also known as BER). The clock and data distributions are what is used to determine the channel bit-error rate.


Equalized signal at RX latch
Clock Recovery
Bit Error Ratio
$P_{err} = \int P(err|t)p(t)\,dt$
Data
Clock

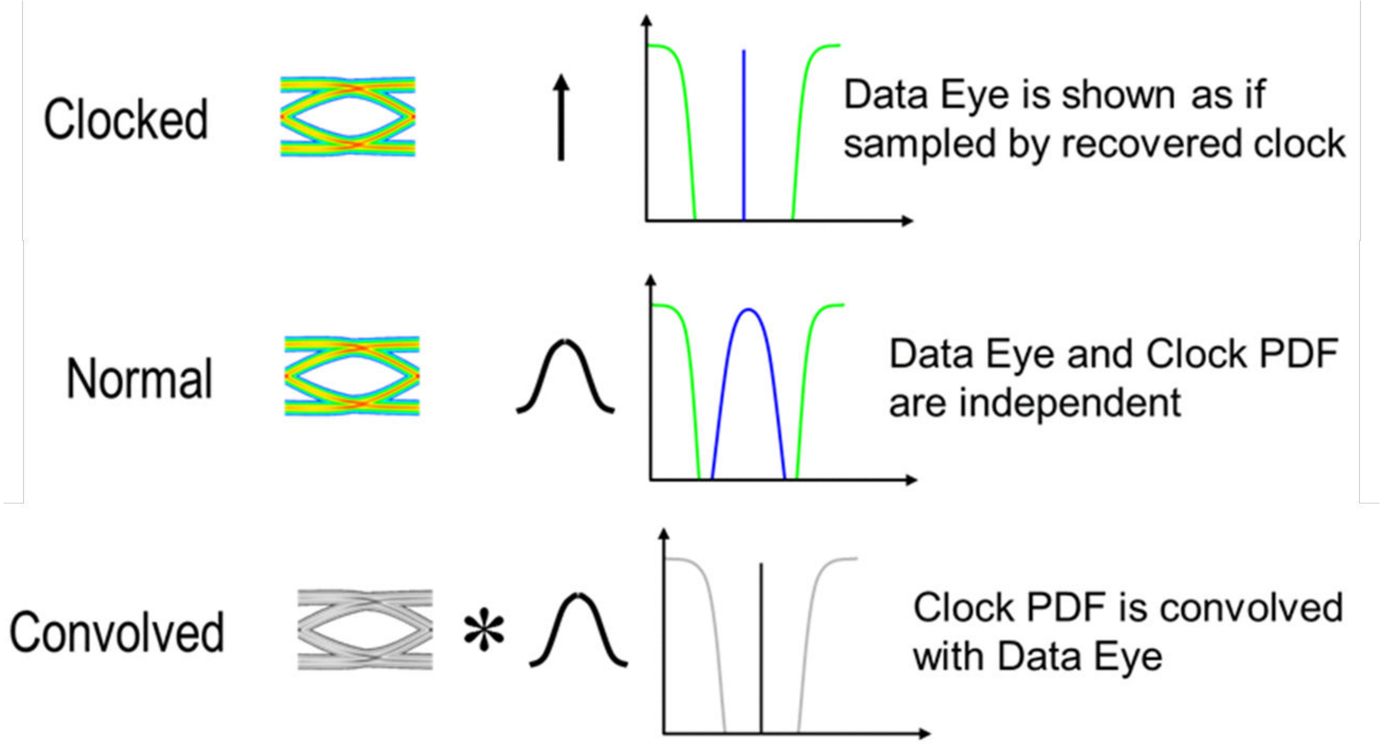
The data bathtub curve (marked in black) is the data distribution at the data input to the Rx latch. The recovered clock (marked in red) is probed at the clock input to the Rx latch and shows a different distributed behavior. The sampling clock would ideally be in the center of the eye and its distribution is the result of multiple jitter sources. The bathtub curve is compared with the Rx clock PDF and the overlap between the curves determines the bit-error rate.

The BER equation is based on the product of the probability of error with a particular clock position and the probability of that clock position occurring. The values for all possible clock positions are then summed and the BER curve is plotted as shown. The highest probability calculated is the reported BER of the channel.

This view of the clock PDF and data bathtub can only be seen when looking at the sampling clock and data from the perspective of an ideal reference clock, such that the behaviors of clock and data are displayed independently. If clock and data are considered dependent, QCD will display the clock PDF as a delta function in the center of the UI because variation of the clock will be reflected in the eye diagram (data eye).

Clock Modes

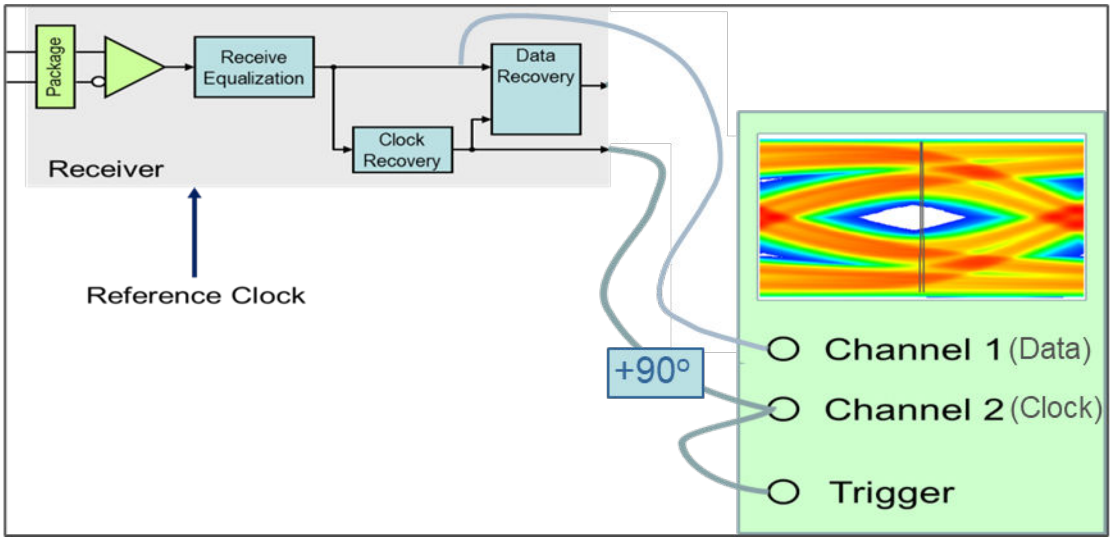
The three clock modes are: clocked, normal and convolved. These modes differ depending on the nature of the Rx clock recovery and assumptions of dependence, or independence between the clock and data.


Clocked
Data Eye is shown as if sampled by recovered clock
Normal
Data Eye and Clock PDF are independent
Convolved
*
Clock PDF is convolved with Data Eye

Clocked mode shows the data as it would be captured by the recovered clock. Normal mode al-lows the user to view both the data and clock independently with respect to an ideal external reference clock. Convolved mode combines the normal mode eye and clock PDF through con-volution to produce an eye diagram, and clock PDF that look as though the simulation were run in clocked mode. These clock modes are primarily designed for use in time domain simulations where the IBIS AMI model returns both the data waveform and clock ticks. Each of the modes will be discussed with regards to both time-domain and statistical analysis highlighting their benefits and when they are applicable for use.

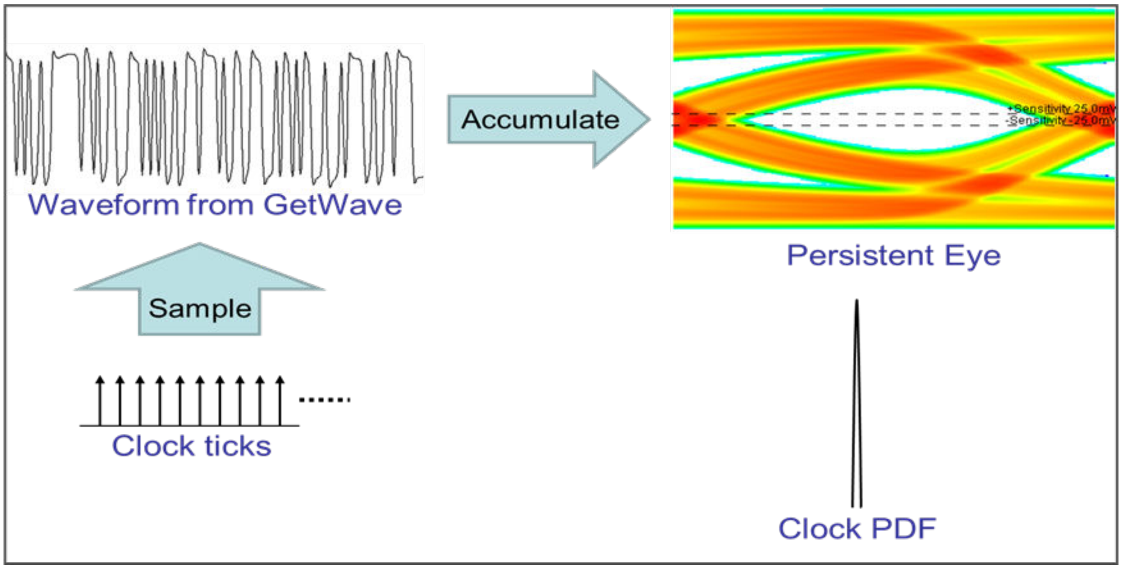
Time Domain Clock Mode: Clocked

The system representation of the clocked mode operation represents how the data is captured by the recovered clock from a system perspective.



The data is captured at the input to the decision latch using the clock from the output of the clock recovery circuit. Note that the time values of AMI clocks are actually ½ UI before the data is sampled (at the start of the data bit), which makes them perfect to trigger waveform acquisition in the scope diagram. A ½ UI delay is added to the input of channel 2 to show the clock in proper relation to the input waveform.

The simulator operation in the clocked mode is shown:



The simulated data waveform and clock is on the left and the resulting eye diagram and clock PDF that would be displayed as an output from the simulator is on the right. The clock PDF is displayed as a delta function in the center of the UI and the recovered clock variation due to noise and jitter is thus reflected in the data eye diagram. Because of this when in clocked mode it is not possible to distinguish where eye closure is coming from, whether it is in the data path or in the clock path. In this mode, clock and data are being treated as dependent variables, meaning the acquisition of waveform data is entirely dependent on the clock signals coming from the AMI model.

The clocked mode approach is intuitive from a systems perspective, because it emulates the way actual systems work. This is the way many AMI models have been designed to be used, and it is the only mode most other simulators support.

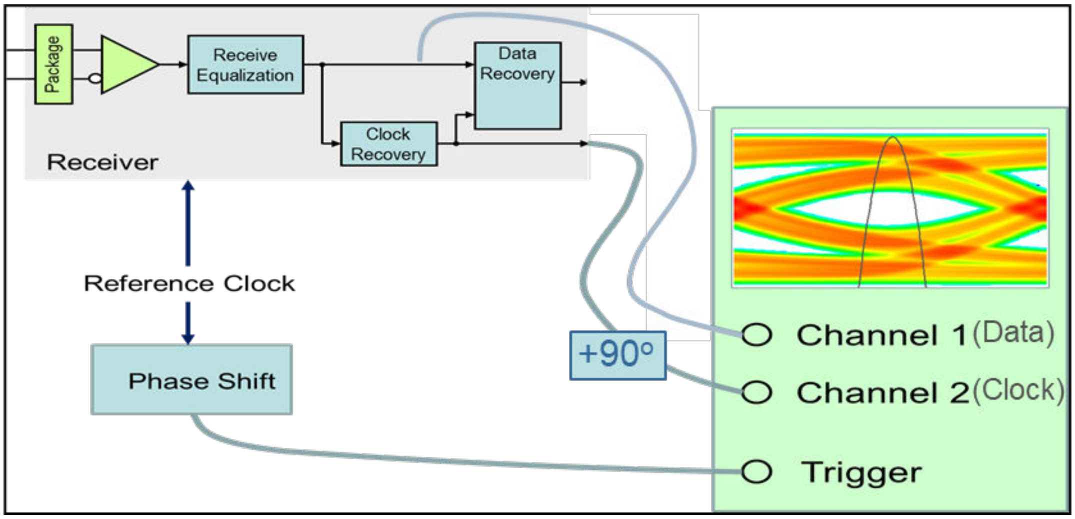
As one might expect, the calculation of BER in clocked mode is limited to the number of bits simulated. If there are only 1e6 samples of a process, one can only talk about probabilities down to 1e-6. This is the limitation of statistical significance that a user will incur in time domain simulations. To illustrate this the following example can be used:

If one could expect a time domain simulation to run at approximately one million bits per minute, after a minute a BER of 1e-6 could be produced. From a statistical significance standpoint this would be an insufficient sample size for most interface standards and proprietary designs. In general, a minimum sample size of 1e12, or a trillion bits, would be required to satisfy most current specifications. To simulate this many bits at a rate of one million bits per minute, the simulation would take almost 2 years. Under the right circumstances this limitation can be overcome by using normal clocked mode.

Time Domain Clock Mode: Normal

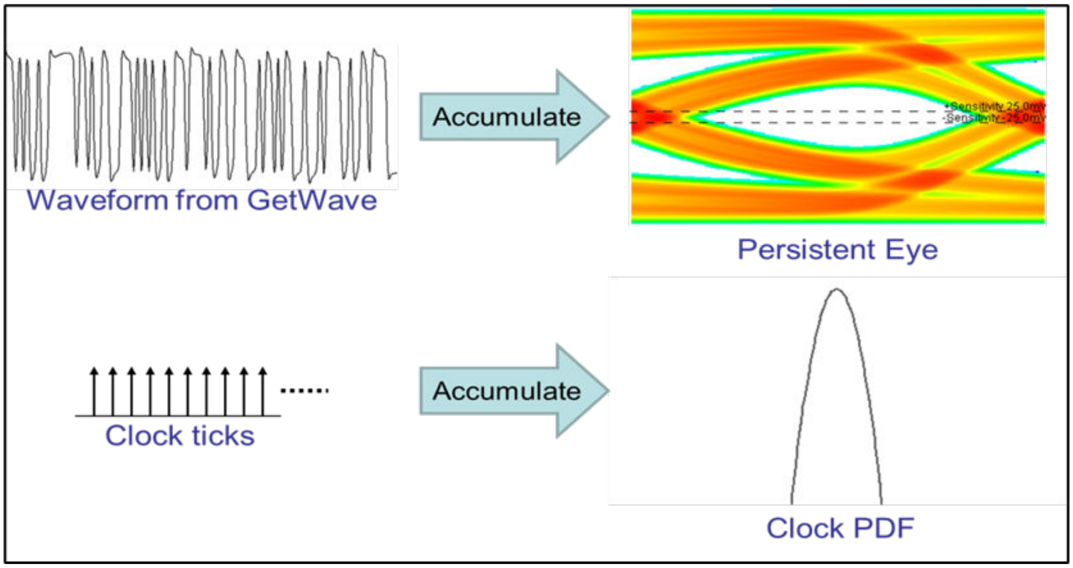
Normal mode provides a method to extend the BER probabilities of time domain simulations into the ranges of most industry standards for channel design, and it does so without needing to use extrapolation algorithms. This is because normal mode is based on an assumption of independence between the recovered clock and the data at the Rx latch. While clocked mode treats clock and data as dependent processes, normal mode treats these distributions as independent. Because of this, the two independent distributions can be used to predict a significantly lower BER by estimating the probability of different interactions.

In normal mode, the data and clock are captured using an ideal reference clock source.



This method of capture keeps the clock jitter and noise from affecting the data bathtub, and thus the data eye consists of the persistent waveform along with any channel ISI, and Tx jitter. The clock PDF starts with the accumulated probabilities of clock ticks returned by the model, which is then convolved with any jitter budgets from the model to create the presented clock PDF.

The simulator operation in the normal clocked mode is shown:



The simulated data waveform and clock ticks is on the left and the resulting eye diagram and clock PDF that would be displayed is on the right. Unlike clocked mode, the data and clock distributions are accumulated independently. This provides a level of insight into model behavior that clocked mode cannot, namely how much of the eye closure is coming from the data path, and how much of the closure is due to jitter in the recovered clock signal. Also, jitter and noise sources are presented on the data bathtub and clock PDF separately, unlike clocked mode where clock jitter and noise distribution are only presented in the data bathtub.

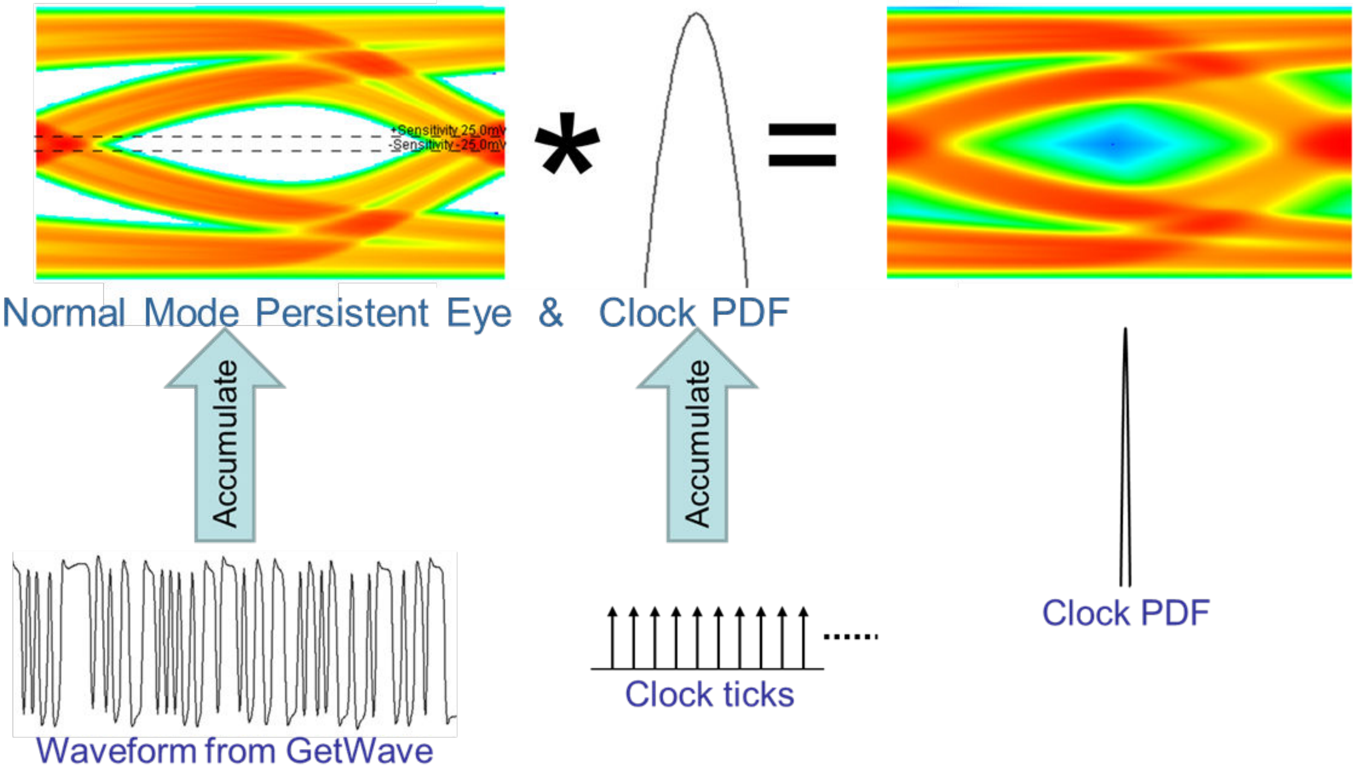
Normal mode provides a method to predict lower bit-error-ratios by improving statistical significance of a time-domain simulation. In comparison with clocked mode, the assumption of independent clock and data allow the statistical distributions to be combined to predict overall probability. Referring back to the example of a time domain simulation taking one minute to predict a BER of 1e-6 in clocked mode, the combined distributions of a million data bits and a million clock bits would allow a BER prediction of 1e-12 in the same simulation time.

On the surface, this assumption of independence between clock and data may seem crazy. How could the recovered clock and data possibly be independent, when both come from the same source? It’s clear that they can’t be completely independent, otherwise the CDR would have no purpose. When independence is discussed in this context, what is really meant is jitter.

Specifically, this is the kind of low frequency, repetitive (usually sinusoidal) jitter that a CDR can track out. If the frequency of the jitter in the incoming signal is high or random enough, the CDR won’t follow it and the variations in data jitter and the recovered clock will be effectively statistically independent. Even when track-able jitter is present in the input signal, if its magnitude is low enough (say 1% of the UI or less) it’s impact on eye closure will be small enough that the additional statistical significance of normal mode may be worth the price. When there is significant track-able sinusoidal jitter, the CDR’s ability to remove track-able jitter is not accounted for, and the effect of that jitter is effectively double counted, making the simulation results pessimistic. How pessimistic depends on the magnitude and frequency of the incoming sinusoidal jitter. Under these conditions normal mode would not be a useful predictor of BER.

Time Domain Clock Mode: Convolved

The third clock mode is convolved. The primary function of convolved mode is to validate assumptions of data and clock independence. This is basically a validation of the normal mode simulation.



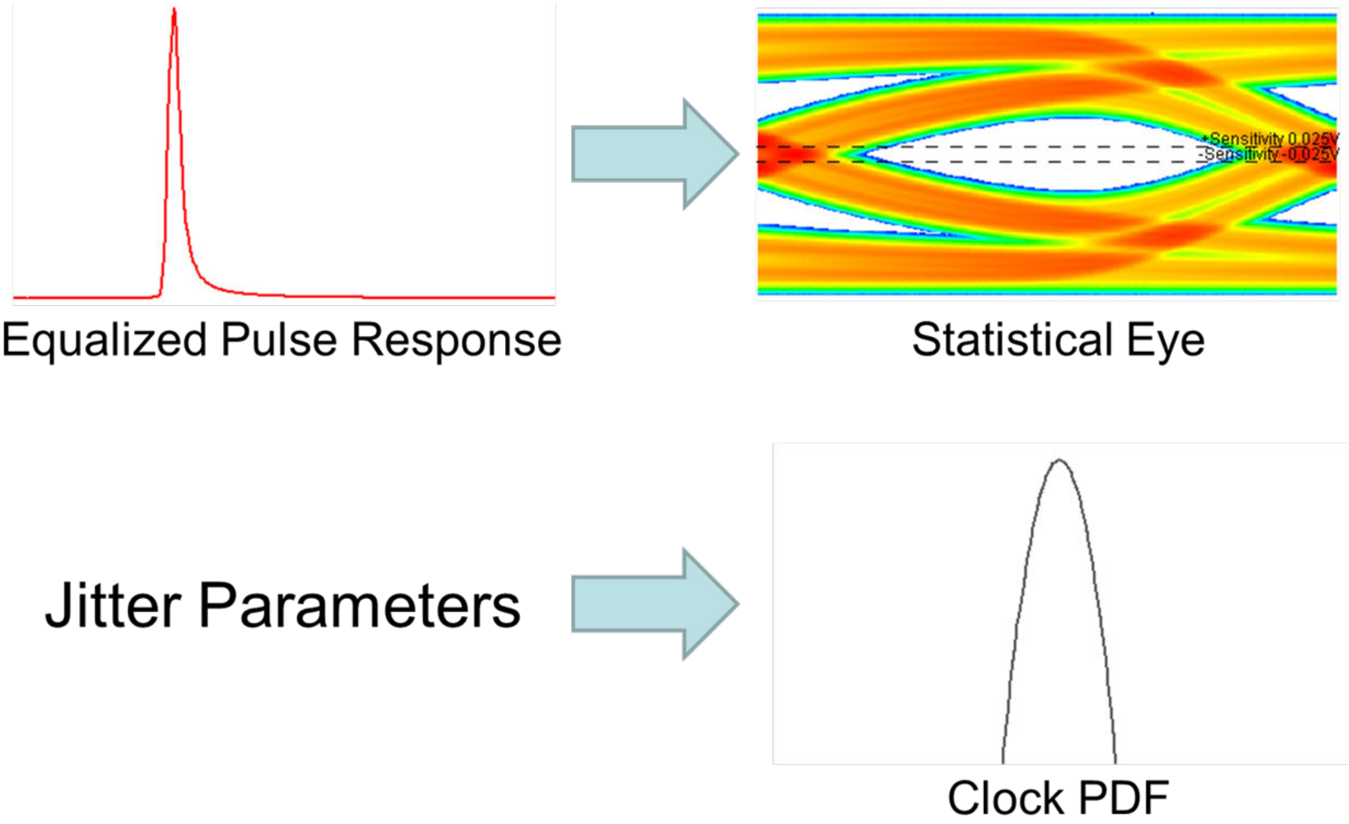
In time domain with the clock mode set to convolved, the data eye and clock PDF captured in normal mode are convolved together to present an eye and clock that look as though the simulation had been run in clocked mode. If the jitter processes in the data path and the clock recovery path are reasonably independent, the eye diagram from a convolved mode simulation will look the same as the eye from a clocked mode simulation. This means that the clock recovery process essentially involves no tracking of the data, the clock and data are independent, and normal mode can be used to reliably predict BERs lower than are possible by clocked mode.

Statistical Simulations and Clock Modes

Statistical simulations are based solely on the impulse response of the channel and data pattern dependencies. There is no waveform nor returned clock ticks from the model in statistical simulations. Because no clock is returned from the model clocked mode and convolved mode are identical.

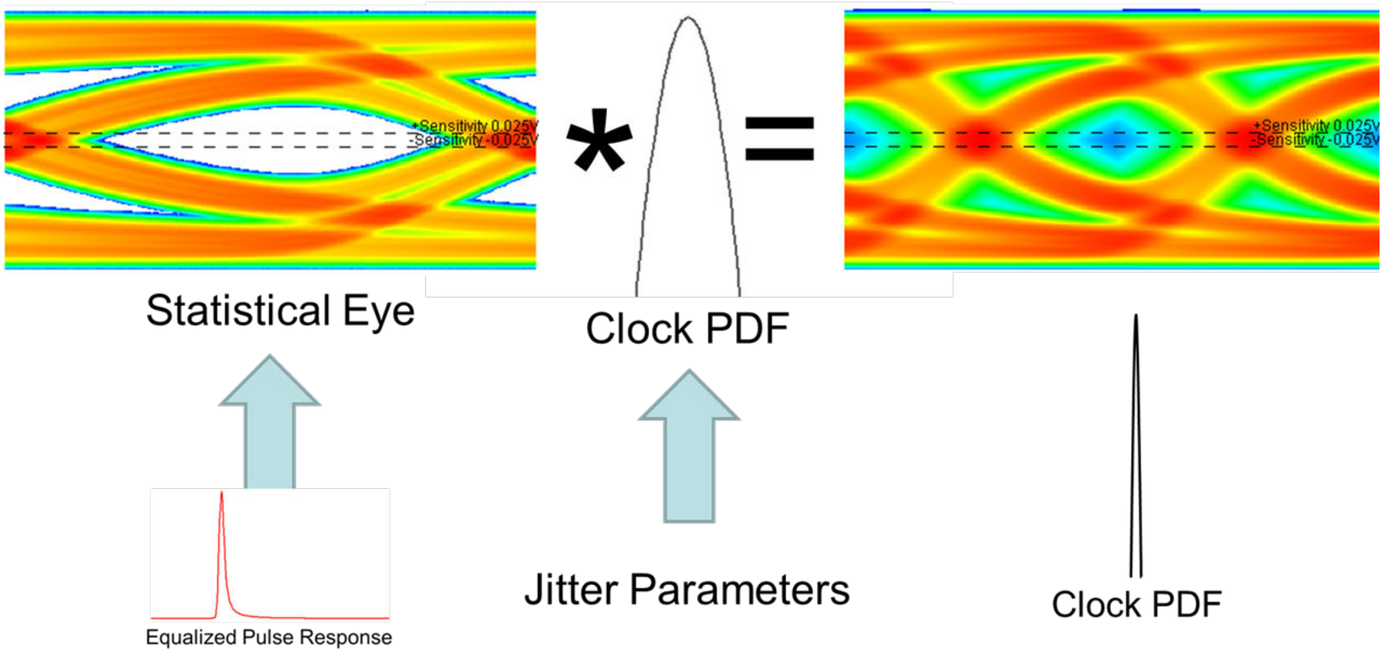
The clock position in a statistical analysis simulation is determined by an algorithm called “Hula Hoop”. This algorithm places a ring 1UI wide over the equalized pulse response of the channel. The data eye is represented by the time points where the ring touches the pulse response. The center point is where the clock PDF is placed.

Statistical analysis in normal mode produces an eye diagram that is the product of the equalized pulse response and probabilities of bit switching.



The ISI of the channel is determined directly from the equalized pulse response and through mathematical analysis the eye diagram can be obtained which includes any Tx jitter. Rx Jitter and Rx clock recovery jitter are IBIS-AMI parameters that are specified or embedded in the AMI model and will be imposed on the clock PDF.

Clocked and convolved modes are performed the same way in statistical analysis. The statistical eye generated using the equalized pulse response is convolved with the clock PDF which is based on the Rx jitter and Rx clock recovery jitter parameters.



Setting Clock Mode

The clock mode can be set from the Serial Link Designer or Parallel Link Designer by accessing the Designator Element properties dialog box. To open the dialog box, double click on either a Tx or Rx designator on the schematic sheet, The clock mode column in the dialog window is a pull-down menu allowing you to select a clock mode for the Rx. You can opt to use the check box next to the pull-down selector to sweep clock modes. If this box is checked, the clock mode becomes a variable in the solution space for the schematic sheet. You can then set and sweep the various modes.

See Also

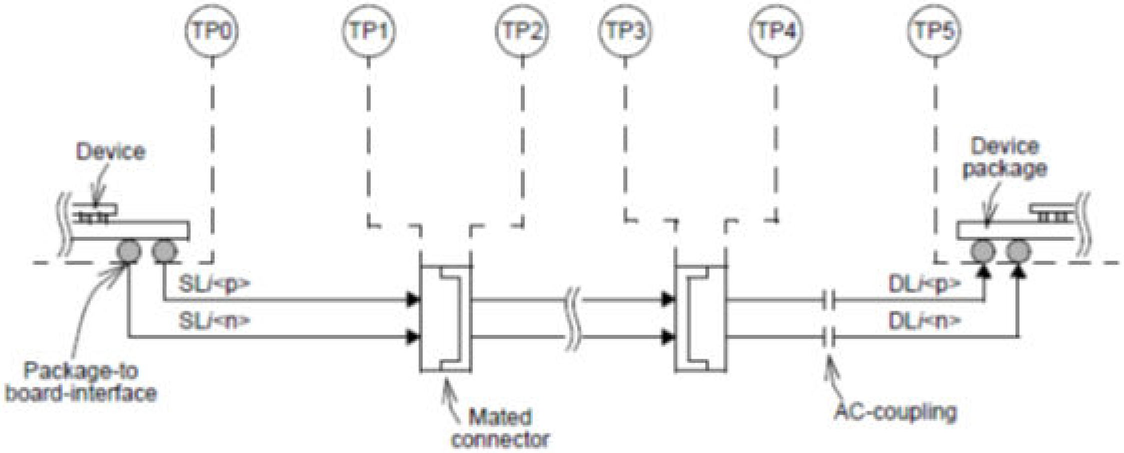
“Eye Measurement and Reporting” on page 12-17

Channel Operating Margin (COM)

Channel Operating Margin (COM) is a figure of merit for a passive channel expressed in decibels. COM gives insight about the quality of the passive channel design. It is calculated using the ratio of signal amplitude factors to noise amplitude factors. Channel bit rate, insertion loss, return loss, cross-coupling, transmitter and receiver equalization, and IC package models are some of the factors applied to determine COM. While it is required for compliance in some applications, COM can also be a valuable part of channel design methodology in general.

The IEEE 802.3bj 100GBASE specification defines the 100GBASE interface to consist of four channels each operating at 25.78125Gbps. These channel designs can involve PCB only, backplane or copper cables. Signaling is accomplished with either NRZ (Non Return to Zero) or PAM4 (Pulse Amplitude Modulation). Encoding the packets with forward error correction (FEC) is optional but can greatly improve a channel BER (Bit Error Ratio). Testing the compliance of the passive electrical channel to the specification requires it to meet or exceed what is known as COM (or Channel Operating Margin) as measured in decibel units.

The 802.3bj channel model with associated test points is shown:



The passive channel referenced is between TP0 and TP5:

COM is a figure of merit derived from the scattering parameters of the passive channel. l. Its calculation relies on the s-parameter models of the victim and aggressor channels along with user specified information about the TX, RX and their equalization characteristics. COM can also account for the package characteristics of the TX and RX or you can choose to define generic package characteristics to be used in the COM calculation.

A series of MATLAB application scripts have been developed to automate the COM calculations. The scripts are offered for download on the IEEE website (e.g. com_ieee8023_93a.m). To maintain continuity for those utilizing this application, Signal Integrity Toolbox integrated the capability of running COM through the **Serial Link Designer** app. Its implementation provides direct interface with MATLAB and a seamless process of passing the appropriate s-parameter channel models. All COM results are consolidated into one report and select COM results from the report can be added to the Channel Analysis report and thus are directly loaded into the **Signal Integrity Viewer** tool after a COM simulation. You can then use these metrics in the analysis of the channel.

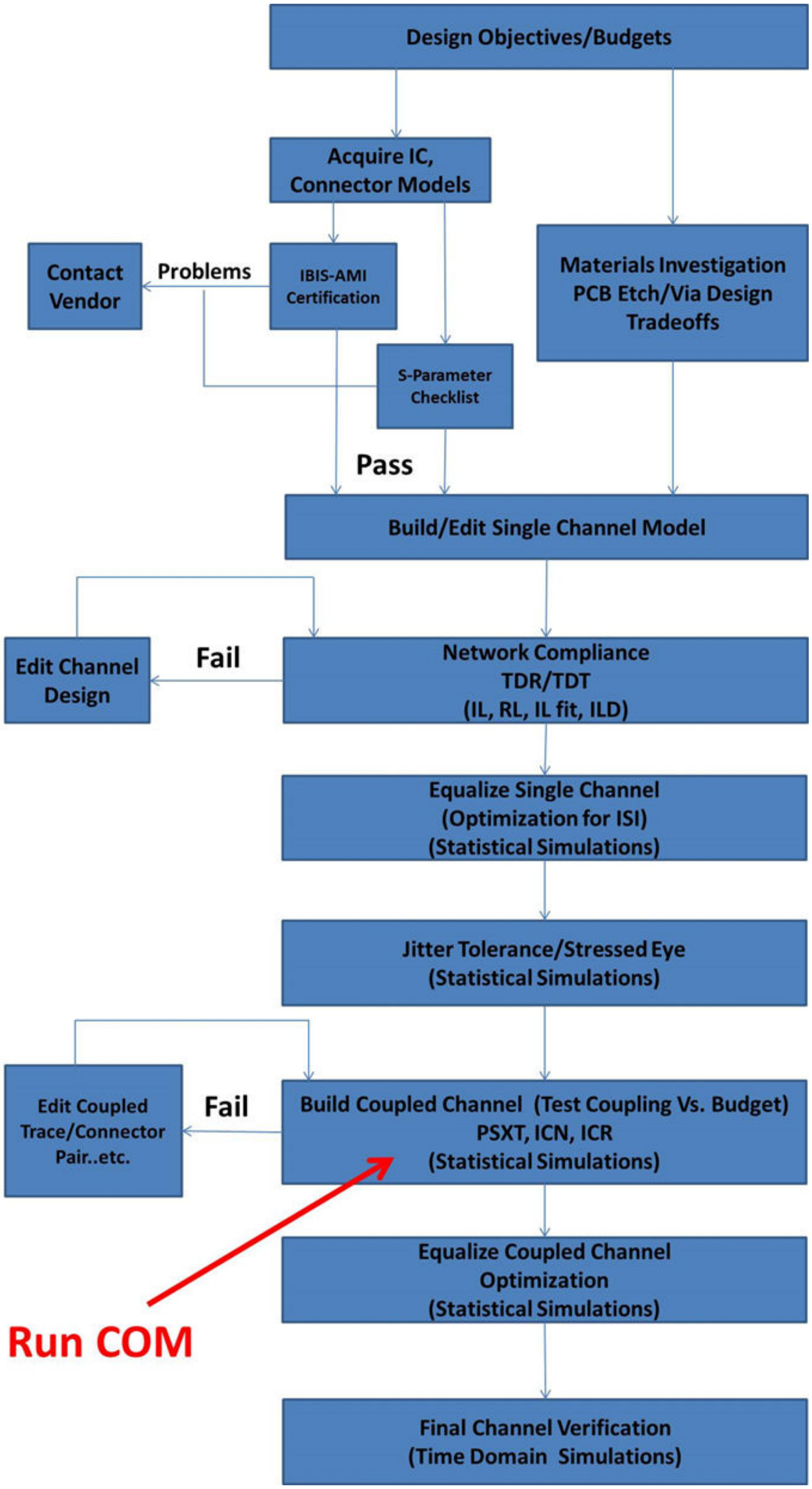
The COM code is a statistical based algorithm which is based on linear time invariant (or LTI) assumptions about the channel, transmitter and receiver. A complete channel design flow would include simulating the interface using TX and RX IBIS-AMI and package models of the actual silicon both statistically and in time domain with an objective of meeting a target BER (Bit-Error-Ratio). This can only be accomplished using a channel simulator such as Serial Link Designer.

Signal Integrity Toolbox offers a variety of compliance and design kits including 802.3bj.

Channel Design Methodology

COM can be used as a fast and easy way test signal-to-noise performance of a channel regardless of compliance to a specification. The channel being designed can be an 802.3bj implementation or a custom design. It could implement NRZ or PAM4 signaling. You can set a multitude of parameters with limits unique to your design and use those values to determine COM.

An example of channel design methodology can be shown as:


Design Objectives/Budgets
Acquire IC, Connector Models
Contact Vendor
Problems
IBIS-AMI Certification
S-Parameter Checklist
Materials Investigation PCB Etch/Via Design Tradeoffs
Pass
Build/Edit Single Channel Model
Edit Channel Design
Fail
Network Compliance TDR/TDT (IL, RL, IL fit, ILD)
Equalize Single Channel (Optimization for ISI) (Statistical Simulations)
Jitter Tolerance/Stressed Eye (Statistical Simulations)
Edit Coupled Trace/Connector Pair..etc.
Fail
Build Coupled Channel (Test Coupling Vs. Budget) PSXT, ICN, ICR (Statistical Simulations)
Run COM
Equalize Coupled Channel Optimization (Statistical Simulations)
Final Channel Verification (Time Domain Simulations)

The highlighted area can benefit from using COM as part of design analysis.

COM Setup

The COM configuration spreadsheet is a specially formatted document and is used to define each of the variables required to run COM. Some of the settings are unique to the channel and others are user preferences or control settings.

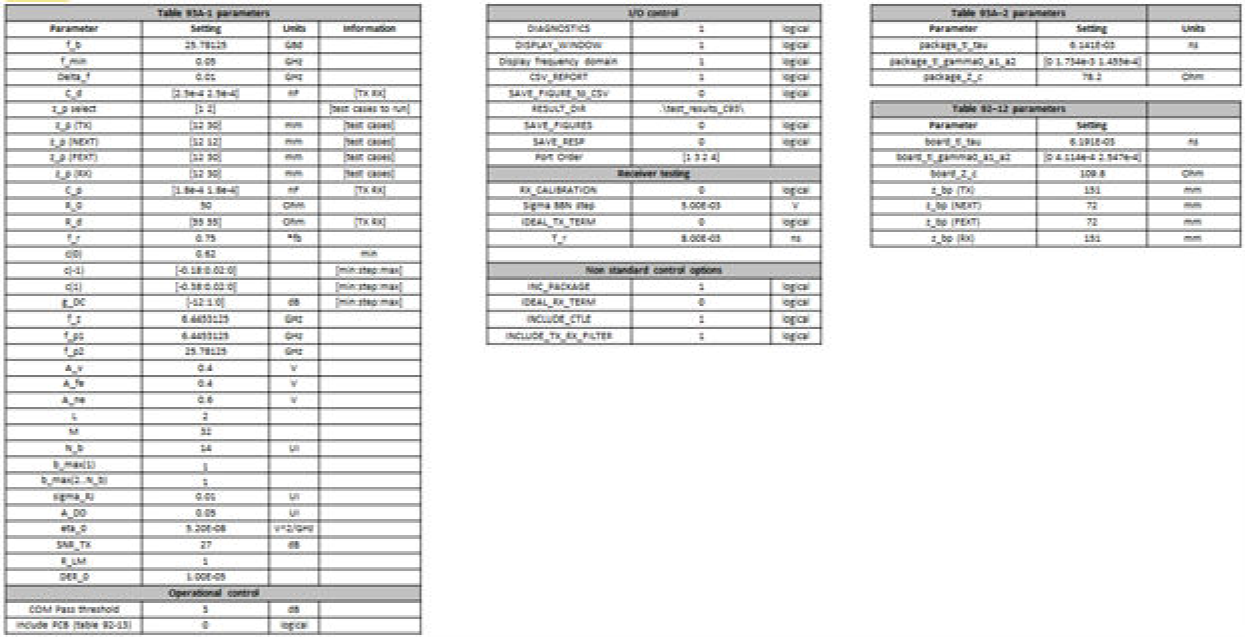


Table 93A-1 parameters			
Parameter	Setting	Units	Information
f_b	25.78125	GBd	
f_min	0.05	GHz	
Delta_f	0.01	GHz	
C_d	[2.5e-4 2.5e-4]	nF	[TX RX]
z_p select	[1 2]		[test cases to run]
z_p (TX)	[12 30]	mm	[test cases]
z_p (NEXT)	[12 12]	mm	[test cases]
z_p (FEXT)	[12 30]	mm	[test cases]
z_p (RX)	[12 30]	mm	[test cases]
C_p	[1.8e-4 1.8e-4]	nF	[TX RX]
R_0	50	Ohm	
R_d	[55 55]	Ohm	[TX RX]
f_r	0.75	*fb	
c(0)	0.62		min
c(-1)	[-0.18:0.02:0]		[min:step:max]
c(1)	[-0.38:0.02:0]		[min:step:max]
g_DC	[-12:1:0]	dB	[min:step:max]
f_z	6.4453125	GHz	
f_p1	6.4453125	GHz	
f_p2	25.78125	GHz	
A_v	0.4	V	
A_fe	0.4	V	
A_ne	0.6	V	
L	2		
M	32		
N_b	14	UI	
b_max(1)	1		
b_max(2..N_b)	1		
sigma_RJ	0.01	UI	
A_DD	0.05	UI	
eta_0	5.20E-08	V^2/GHz	
SNR_TX	27	dB	
R_LM	1		
DER_0	1.00E-05		
Operational control			
COM Pass threshold	3	dB	
Include PCB (table 92-13)	0	logical	

I/O control		
DIAGNOSTICS	1	logical
DISPLAY_WINDOW	1	logical
Display frequency domain	1	logical
CSV_REPORT	1	logical
SAVE_FIGURE_to_CSV	0	logical
RESULT_DIR	.\test_results_C85\	
SAVE_FIGURES	0	logical
SAVE_RESP	0	logical
Port Order	[1 3 2 4]	
Receiver testing		
RX_CALIBRATION	0	logical
Sigma BBN step	5.00E-03	V
IDEAL_TX_TERM	0	logical
T_r	8.00E-03	ns
Non standard control options		
INC_PACKAGE	1	logical
IDEAL_RX_TERM	0	logical
INCLUDE_CTLE	1	logical
INCLUDE_TX_RX_FILTER	1	logical

Table 93A-2 parameters		
Parameter	Setting	Units
package_tl_tau	6.141E-03	ns
package_tl_gamma0_a1_a2	[0 1.734e-3 1.455e-4]	
package_Z_c	78.2	Ohm

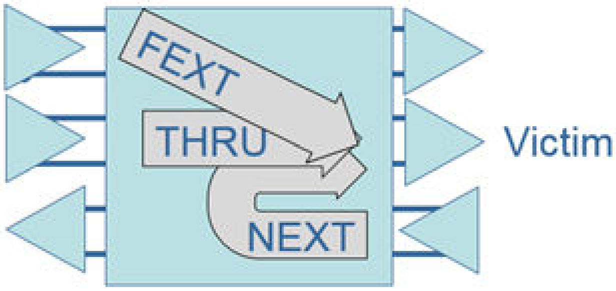
Table 92-12 parameters		
Parameter	Setting	
board_tl_tau	6.191E-03	ns
board_tl_gamma0_a1_a2	[0 4.114e-4 2.547e-4]	
board_Z_c	109.8	Ohm
z_bp (TX)	151	mm
z_bp (NEXT)	72	mm
z_bp (FEXT)	72	mm
z_bp (RX)	151	mm

Example spreadsheets for each particular application covered by IEEE (802.3bj 100GBASE-KR4, 100GBASE-CR4, 100GBASE-KP4 and CAUI-4 C2C) are available with the COM download.

Note The COM script is a continuously changing piece of software as new specifications are adopting this methodology and new metrics and tests are being added. It is recommended that the user visit the IEEE website (https://www.ieee802.org/3/bj/public/tools.html) to download the latest COM code, example spreadsheets, and documentation

Channel S-parameters for COM

Along with the spreadsheet, the other input to the COM code is the measured or simulated s-parameter models of the victim channel and its aggressors. A coupled channel model is shown:



There is a victim channel (Thru) and two aggressors showing the far-end (FEXT) and a near end (NEXT) coupling to the victim. Models of the victim and each aggressor are passed into COM as 4-port S-parameter models. There is virtually no limit to the number of aggressor channels passed into COM. However, from a practical standpoint 2-3 aggressors on each side of the victim will account for most of the crosstalk. **Serial Link Designer** automatically extracts these models during the "Network Analysis" simulation and when running COM simulation. The appropriate s-parameters and parameters are automatically passed into the COM script.

Viewing COM Results

There are many results outputted by COM. All of the COM results from a simulation are available in the **Collected COM Results** report. To access the report, in the **Serial Link Designer**, select **Reports > Collected COM Results Report**.

Any of the results from the COM simulation can be preselected for display in the **Channel Analysis Report**. This loads the results in the **Signal Integrity Viewer** after simulation and allows them to be analyzed along with the standard simulation results. You can select the COM results that you wish to add to the report by editing the "com_columns.txt" file which resides in the Serial Link Designer installation area. You can copy this file to your site configuration directory if you have one defined.

An example `com_columns.txt` file that demonstrates adding parameters to the channel analysis report is shown:

Other than reporting COM the other parameters were chosen for demonstration purposes only. The parameters are defined as follows:

- **COM**: COM value in dB for this case.
- **COM fit loss (dB)**: Fitted insertion loss at half baud rate.
- **COM FOM ILD**: RMS over half baud rate span of insertion loss deviation. This may be used in the diagnosis of a channel design.
- **COM ICN (mV)**: RMS over half baud rate span of power sum of the crosstalk. This may be used in the diagnosis of a channel design.
- **COM TXLE taps**: List of transmitter FFE taps used for the CTLE in the COM calculations.
- **COM FOM**: Best figure of merit result from the CTLE and Tx FFE optimization.
- **COM DFE taps**: Adapted DFE tap values.
- **CTLE DC gain (dB)**: DC gain of the CTLE after adaptation 0 to -12dB.

References

[1] IEEE Standard for Ethernet: Amendment 2, *Physical Layer Specifications and Management Parameters for 100 Gb/s Operation Over Backplanes and Copper Cables*, located at https://standards.ieee.org/standard/802_3bj-2014.html.

[2] R. Mellitz, Adee Ran, *COM Quick Guide for April 2014*, located at https://www.ieee802.org/3/bj/public/tools/mellitz_3bj_01_0414.pdf.

[3] COM Configuration Documentation (config_com_ieee8023_93a_doc.pdf).

See Also

More About

- “Channel Operating Margin (COM) for Serial Link” on page 5-56

External Websites

- https://www.ieee802.org/3/bj/public/tools.html

Eye Measurement and Reporting

The standard of performance for a high-speed serial link is bit-error-ratio (BER). BER is estimated based on a number of factors, one of which is the inner eye contour of an eye diagram. Simulation results, both statistical and time domain, contain eye width and height measurements, along with calculated margin to a target BER.

Note The methods and reporting of these metrics pertain to **Parallel Link Designer** only when simulating in **STAT** mode.

The inner contour of the eye diagram at the **Target BER** is used to estimate the channel BER. Understanding the derivation of eye height, width and voltage margin as reported in statistical and time domain simulations provides insight to the BER estimate. For compliance to standards, inner and outer eye masks can be applied to simulation results and the available margin is reported. Knowing how the margin is calculated and reported is beneficial to debugging potential problems within a design.

Parameters Used in Eye Measurement

Eye metrics such as height, width and margin along with the BER for the channel are determined based on three metrics; the eye contours, the clock PDF and the sensitivity of the receiver.

Eye Contours

Eye contours are plots of the amplitude associated with fixed probabilities as a function of sampling time. They indicate the shape of the inner and outer boundaries of the eye diagram for each of a number of different probabilities. The eye contours for a given simulation are based on the Target BER.

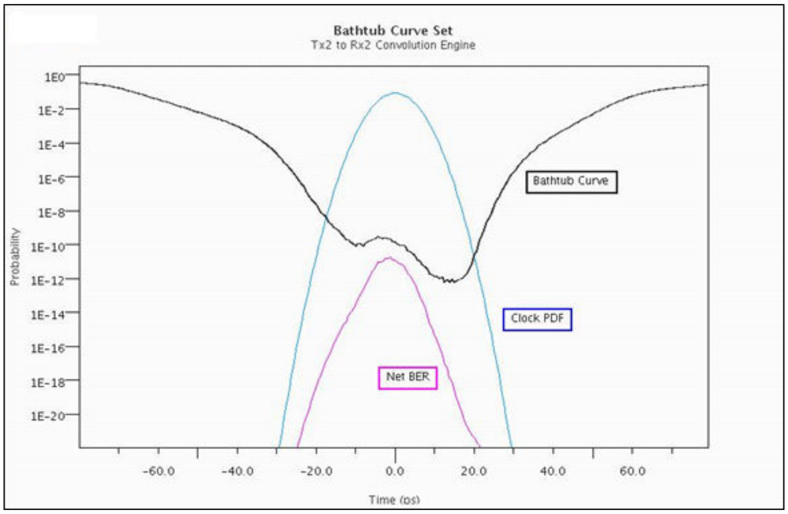
To set the target BER, open the Simulation Parameters dialog box by selecting **Setup** > **Simulation Parameters** from the **Serial Link Designer** or **Parallel Link Designer** app.

The default value is `1e-12`, but you can set this based on the BER requirement of the channel design. Target BER defines the contours that is generated and reported after rolling up the simulation results.

Four eye contours are generated after the simulation, for the Target BER, Target BER + `1e3`, Target BER + `1e6` and Target BER + `1e9`. For example, if the Target BER is set at `1e-12`, the contours are displayed at: `1e-12`, `1e-9`, `1e-6` and `1e-3`. Regardless of the Target BER setting, a `0` contour is also generated which represents the BER = `0` point.

Clock PDF

The clock PDF is the probability density function (PDF) of the phase difference between the clock at the receiver decision point and an ideal transmitter symbol clock. It is represented as a Gaussian probability density function.

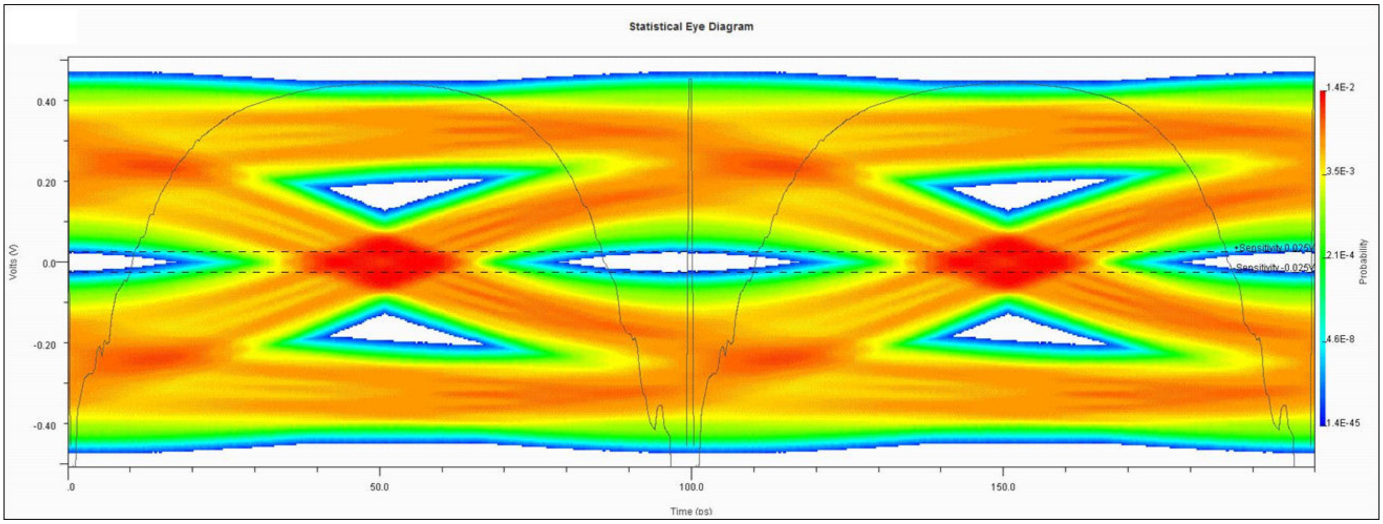


You can determine the net BER from the interaction between the bathtub curves and the clock pdf. The bathtub curve is the probability of error as a function of the time that the data is actually sampled. The net BER is the probability of an error occurring at a given sampling time given the probability of sampling at that time. This curve is the area under the product of the bathtub curve and the clock PDF.

Receiver Sensitivity

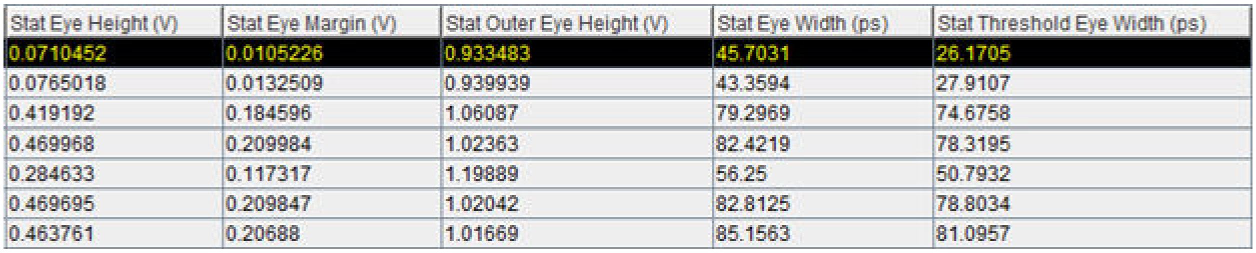
Sensitivity is a keyword that is part of the IBIS-AMI specification for receiver models. It is defined as the minimum latch overdrive voltage at the data decision point of the receiver after equalization. For example if sensitivity is defined as 25mV, the latch would require +/- 25mV for switching. The default sensitivity used in the **Serial Link Designer** and **Parallel Link Designer** is `0`.

A statistical eye diagram with the bathtub curve set and the receiver sensitivity marked +/- 25mV (dashed lines) is shown:



Eye Reporting

Signal Integrity Viewer reports the results of statistical simulation.

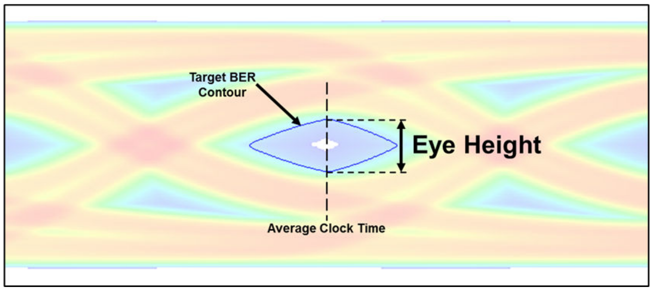


Stat Eye Height (V)	Stat Eye Margin (V)	Stat Outer Eye Height (V)	Stat Eye Width (ps)	Stat Threshold Eye Width (ps)
0.0710452	0.0105226	0.933483	45.7031	26.1705
0.0765018	0.0132509	0.939939	43.3594	27.9107
0.419192	0.184596	1.06087	79.2969	74.6758
0.469968	0.209984	1.02363	82.4219	78.3195
0.284633	0.117317	1.19889	56.25	50.7932
0.469695	0.209847	1.02042	82.8125	78.8034
0.463761	0.20688	1.01669	85.1563	81.0957

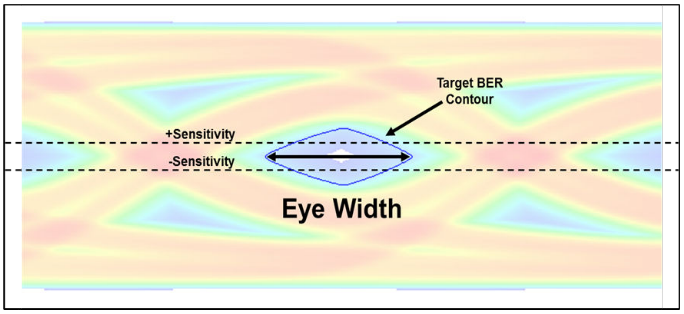
These results include: Stat Eye Height, Stat Eye Width, Stat Eye Margin, Stat Outer Eye Height and Stat Threshold Eye Width. These results are all determined from the Target BER contour (1E-12) and the receiver sensitivity (Stat Threshold Eye Width, Sensitivity +/- 25mV).

Additional reported parameters are:

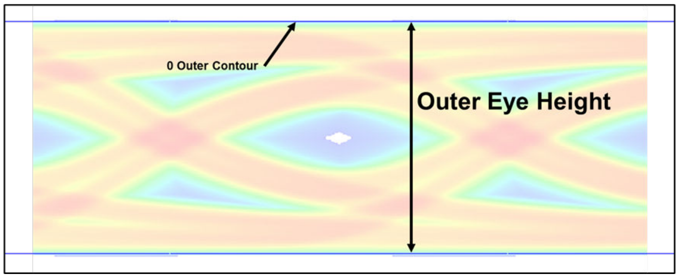
- **Stat Eye Height(V)** — The height of the target bit error rate contour at the average clock time.



- **Stat Eye Width(ps)** — The width of the eye measured at the 0V crossing.

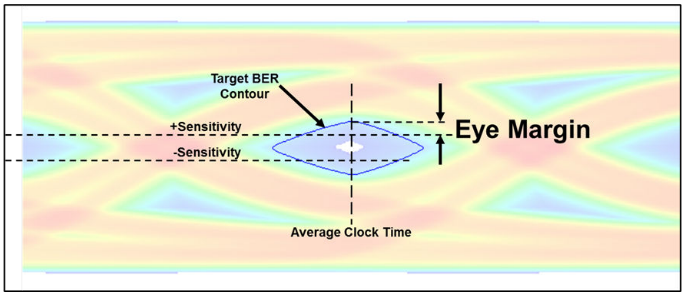


- **Stat Eye Outer Height (V)** — The maximum voltage measured on the outer eye. This is the maximum voltage measured on the 0 outer contour.

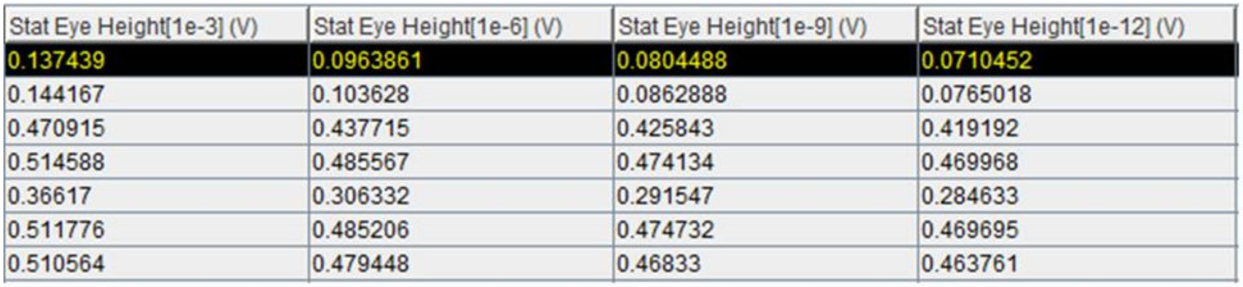


- **Stat Threshold Eye Width (ps)** — The eye width measured at the intersection of the inner eye and the receiver sensitivity

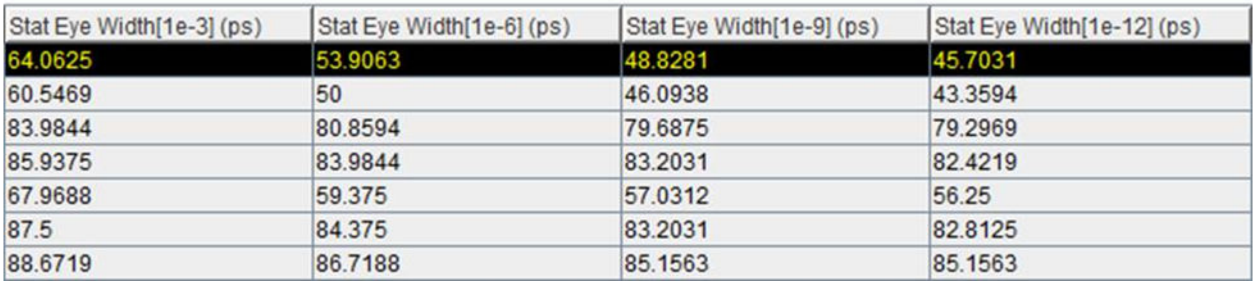
- **Stat Eye Margin (V)** — Voltage measured from the sensitivity threshold to the target BER contour at the average clock time.



Eye height and eye width are reported for all contours generated in the simulation based on the Target BER.



Stat Eye Height[1e-3] (V)	Stat Eye Height[1e-6] (V)	Stat Eye Height[1e-9] (V)	Stat Eye Height[1e-12] (V)
0.137439	0.0963861	0.0804488	0.0710452
0.144167	0.103628	0.0862888	0.0765018
0.470915	0.437715	0.425843	0.419192
0.514588	0.485567	0.474134	0.469968
0.36617	0.306332	0.291547	0.284633
0.511776	0.485206	0.474732	0.469695
0.510564	0.479448	0.46833	0.463761



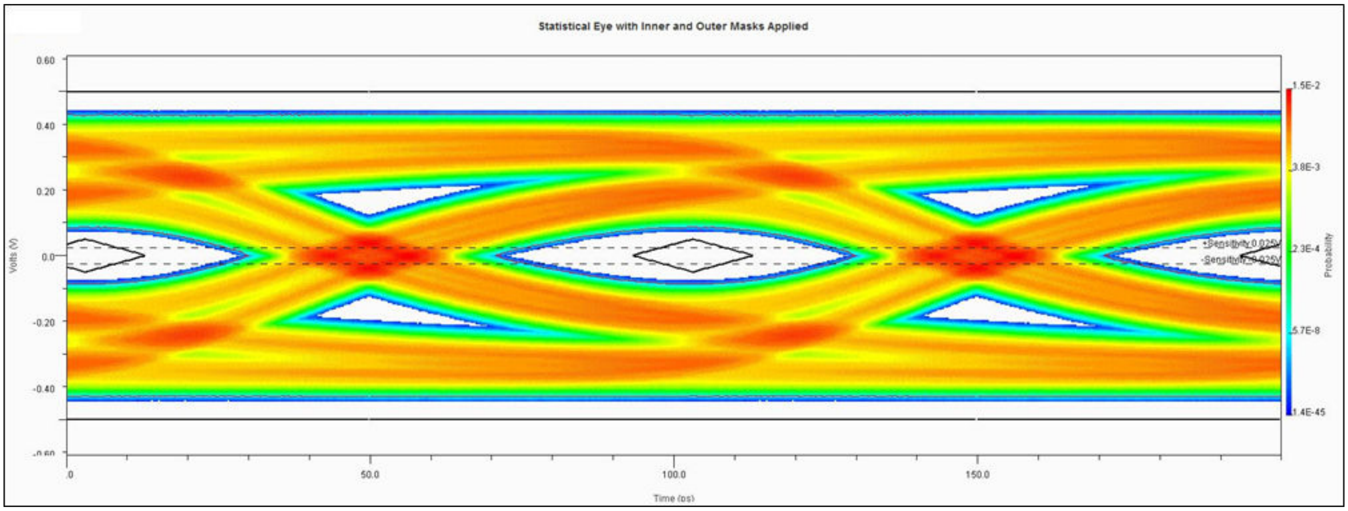
Stat Eye Width[1e-3] (ps)	Stat Eye Width[1e-6] (ps)	Stat Eye Width[1e-9] (ps)	Stat Eye Width[1e-12] (ps)
64.0625	53.9063	48.8281	45.7031
60.5469	50	46.0938	43.3594
83.9844	80.8594	79.6875	79.2969
85.9375	83.9844	83.2031	82.4219
67.9688	59.375	57.0312	56.25
87.5	84.375	83.2031	82.8125
88.6719	86.7188	85.1563	85.1563

For example, if the target BER is `1e-12`, statistical eye heights and widths for `1e-3`, `1e-6`, `1e-9` and `1e-12` BER are reported.

Note When comparing the reported results to measurements made manually in **Signal Integrity Viewer**, an error is introduced from the samples per bit selection. To determine the amount of error when making a manual measurement, divide the UI (unit interval) of a bit by the number of samples per bit (UI/SPB). Using a higher number of samples per bit results in a smaller error.

Calculating Eye Margin from Simulation Results to Eye Mask

When an eye mask is defined and applied to a sheet being simulated, the margin between the mask and the Target BER contour are reported. An eye mask can be defined as either an inner mask, outer mask or both. A statistical eye diagram with inner and outer masks applied is shown:



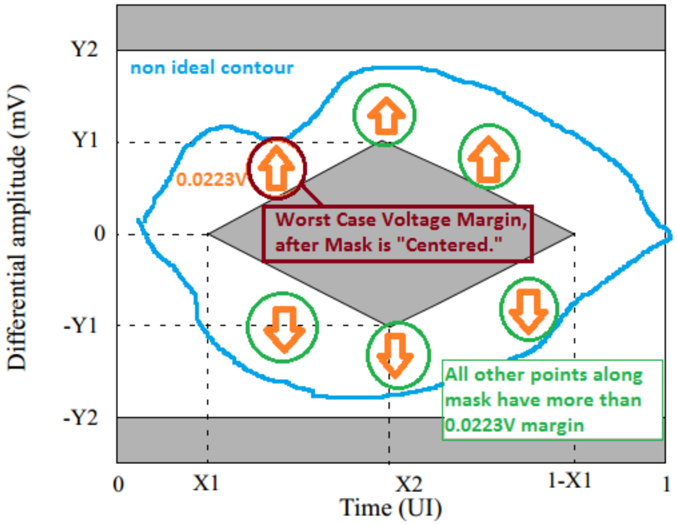
If both inner and outer masks are defined, the smallest margin at any point of the two is reported. The worst eye height margin and worst eye width margin at any point for a given eye mask. Only one result is reported so it is important know which masks are being applied to best identify violations.

You can define and use two types of eye masks: static and skew eye masks. A static eye mask is centered at the 0.5UI point of the bit time. The margin to the mask can then be determined based on its static position. The skew eye mask is positioned by the simulator after simulation to maximize eye margin and place the mask at its optimal point in the eye

To obtain the mask margin numbers using a skew eye mask:

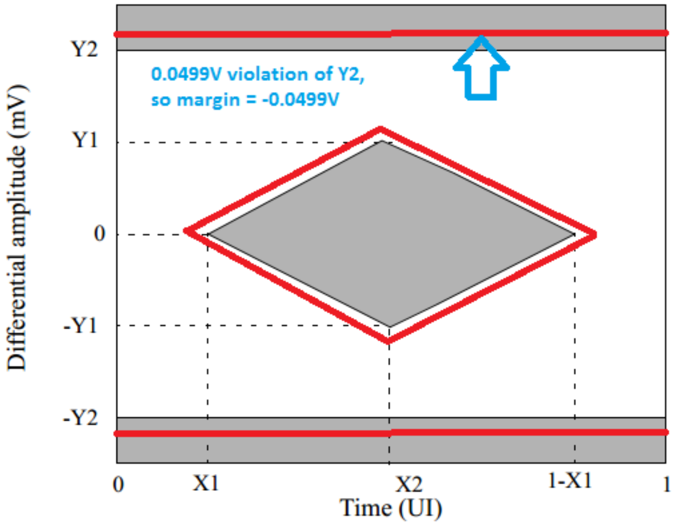
1. Slice UI into discrete time slices (for example, 256 slices per UI).
2. Place mask edge at zero UI and obtain margins for all time slices that cross mask (margins are positive and negative).
3. Increment mask to next time slice and recapture margins for all slices that cross mask.
4. Continue this process until right side of mask hits 1 UI.

Taking all of this data, obtain the best position for the mask that maximizes the worst case margin (looking for most positive result) across all time slices. Then report the worst case margin. The detemination of mask margins is shown as:

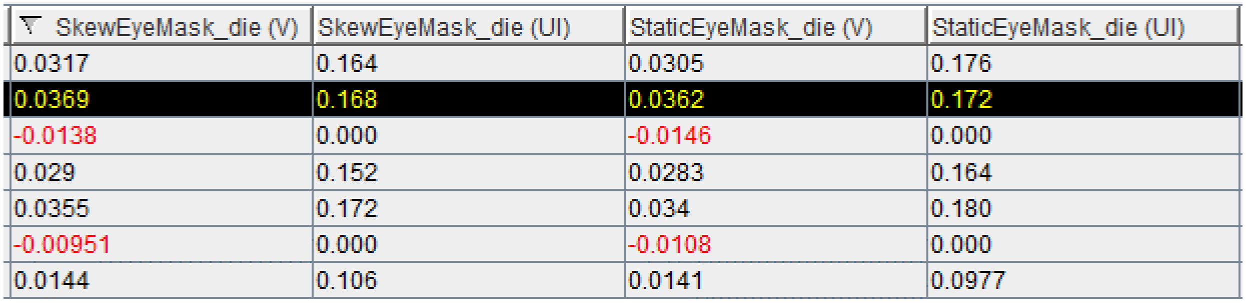

Differential amplitude (mV)
Y2
Y1
0
-Y1
-Y2
non ideal contour
0.0223V
Worst Case Voltage Margin, after Mask is "Centered."
All other points along mask have more than 0.0223V margin
0
X1
X2
1-X1
1
Time (UI)

The worst case margin is shown to be between a perturbation of the eye contour and the mask. In this case no outer mask is assumed.

If both an outer mask and inner mask are applied in a simulation and a violation is reported, you need to determine where the violation comes from. A violation to the outer eye mask when both inner and outer eye masks applied is shown:


Differential amplitude (mV)
Y2
Y1
0
-Y1
-Y2
0.0499V violation of Y2, so margin = -0.0499V
0
X1
X2
1-X1
1
Time (UI)

After simulation, the results for eye mask margin is reported:



SkewEyeMask_die (V)	SkewEyeMask_die (UI)	StaticEyeMask_die (V)	StaticEyeMask_die (UI)
0.0317	0.164	0.0305	0.176
0.0369	0.168	0.0362	0.172
-0.0138	0.000	-0.0146	0.000
0.029	0.152	0.0283	0.164
0.0355	0.172	0.034	0.180
-0.00951	0.000	-0.0108	0.000
0.0144	0.106	0.0141	0.0977

In this case both skew and static masks are applied. Eye height and width margins are reported in the individual columns. Comparing static and skew mask margins in the table below show slightly more margin when applying the skew mask. The red entries represent violations to either the upper or lower mask. To identify the violation the Target BER contour and the mask can be plotted.

See Also

“Clock Modes” on page 12-2 |