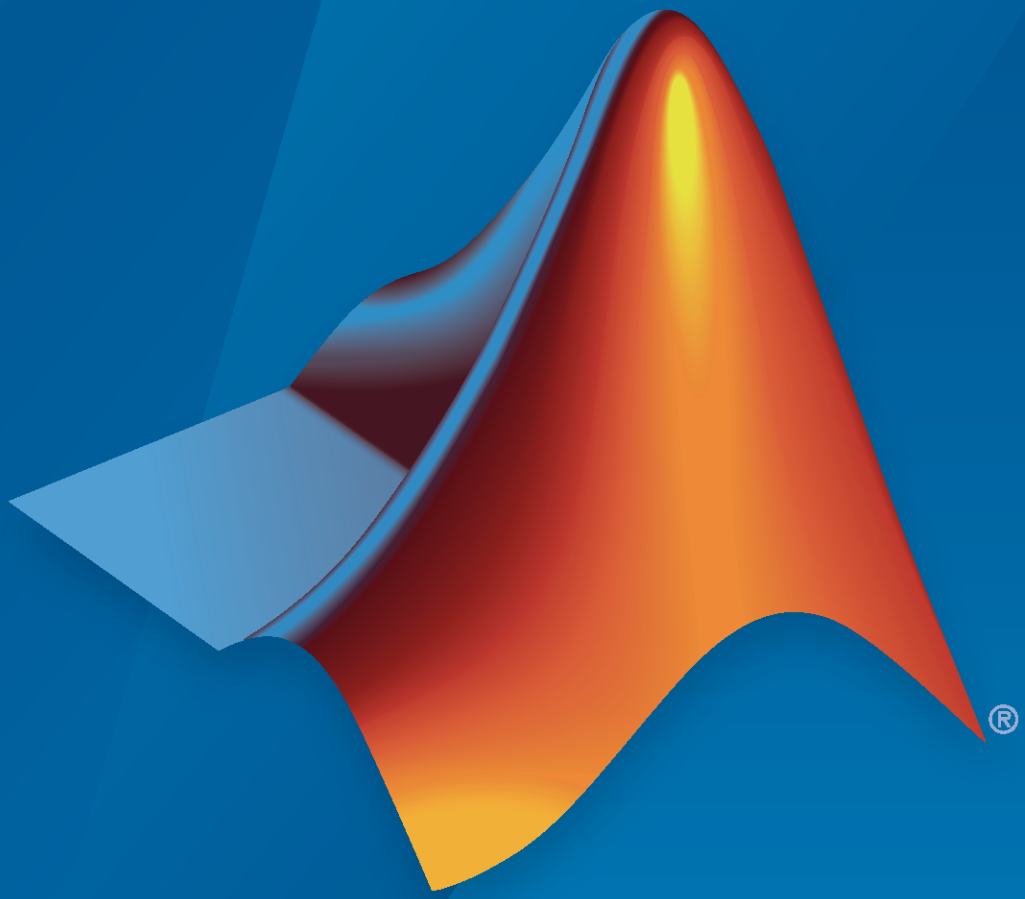
Signal Integrity Toolbox™

User's Guide



MATLAB®

R2021b

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

Revision History

September 2021	Online only	New for Version 1.0 (Release 2021b)

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

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.

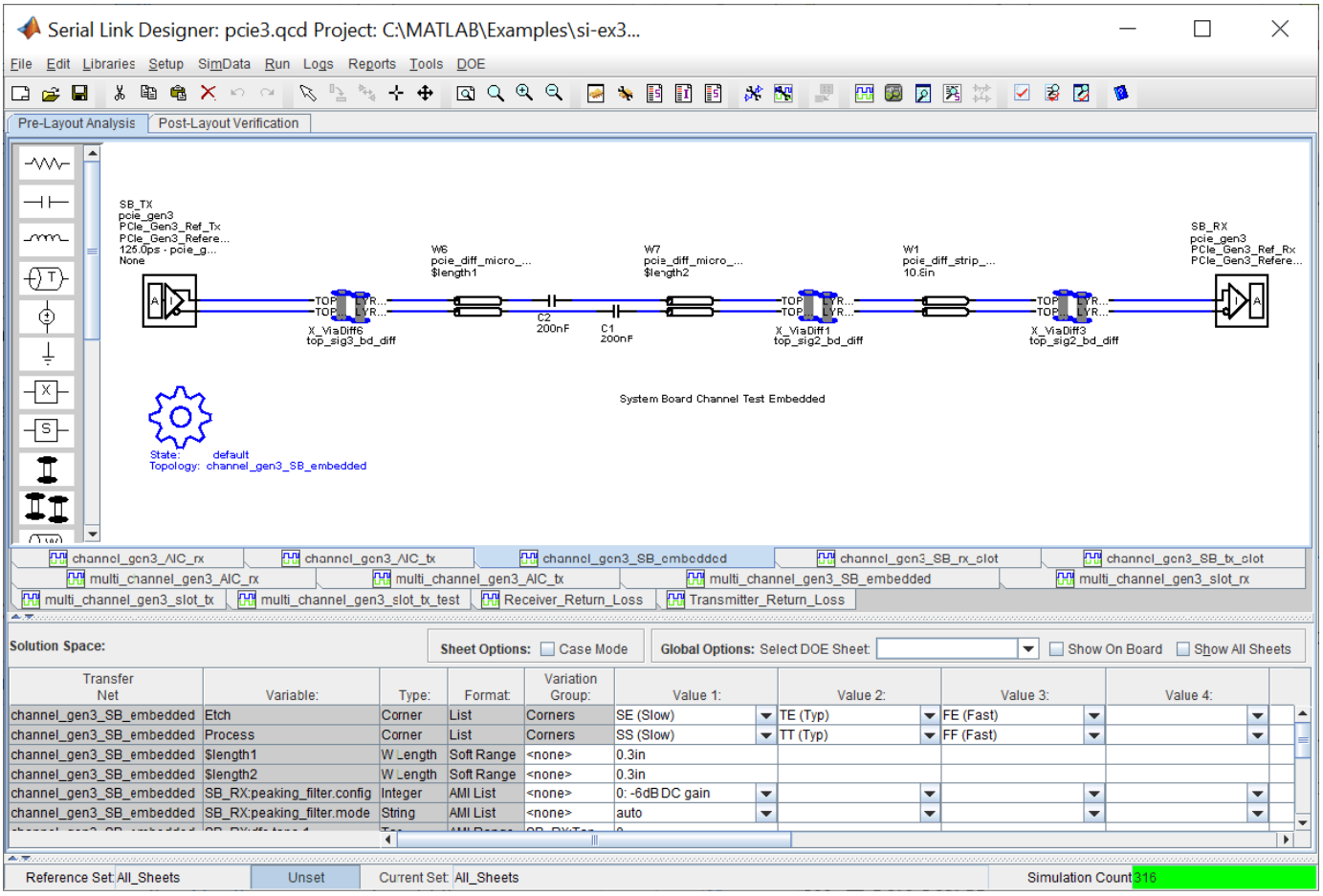
The PCIe-3 signal integrity kit includes all the transfer nets, topologies, generic buffer models and compliance rules for a PCIe-3 high-speed SerDes interface. This includes PCIe-3 technology IBIS-AMI models for the SerDes transmitter and receiver, PCIe-3 compliance masks and transfer nets preconfigured 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");
```



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.
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`.
Record Start	Time to start saving waveforms in a STAT Mode time domain simulation.
Record Bits	Number of bits of the waveform to save.
Waveform Analysis Bits	Number of bits from the STAT Mode simulation to use for Waveform Analysis.
Minimum Ignore Bits	Start time for STAT Mode 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. 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.
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 each waveform in serial link time domain simulation.
Record Bits	Number of bits of the waveform to save.
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.

Parameter	Description

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.

Time Domain Crosstalk Mode	Account for crosstalk in time domain simulation. • Semi-Analytic — Crosstalk is accounted from statistical analysis. • Explicit — Crosstalk is active during time domain simulation.
Enforce Passivity	Make all S-Parameters in the analysis passive.
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 STAT Mode 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.
Conductor Roughness	Surface roughness of conductors in microns (RMS). Used by default for lossy transmission line models created outside Serial Link app.

Parameter	Description
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.
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 the network model, 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 to define all the circuit elements. 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 which ever 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

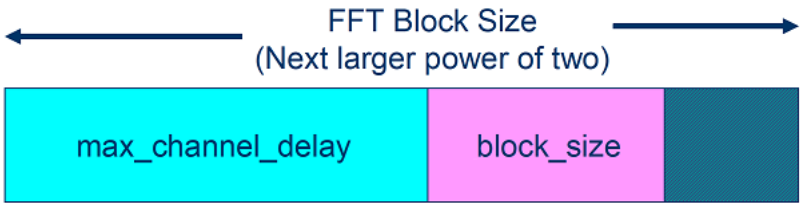
Maximum frequency output for transfer function and S parameters describing the end

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 $1.5 \times DR$, 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.

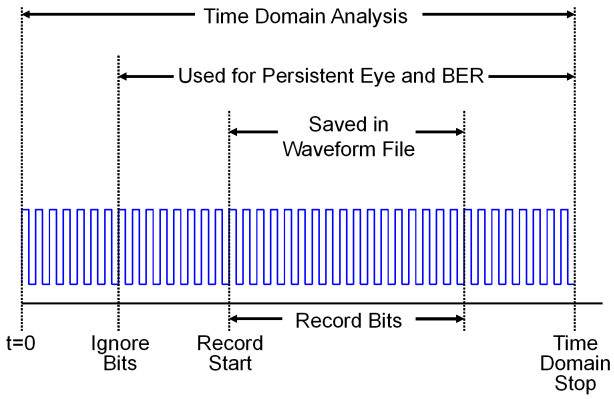


Time Domain Start and Stop Parameters

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 2-6
- “Stimulus Patterns in Serial Link Design” on page 2-8
- “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. 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 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

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 Serial Link Design” on page 2-2
- “Stimulus Patterns in Serial Link Design” on page 2-8
- “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 Pattern

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.

See Also

More About

- “Simulation Parameters Used in Serial Link Design” on page 2-2
- “Specify Corner Conditions in Serial Link Design” on page 2-6
- “Model Jitter and Noise While Designing Serial Link” on page 10-9

Pre-Layout Analysis of Serial Link

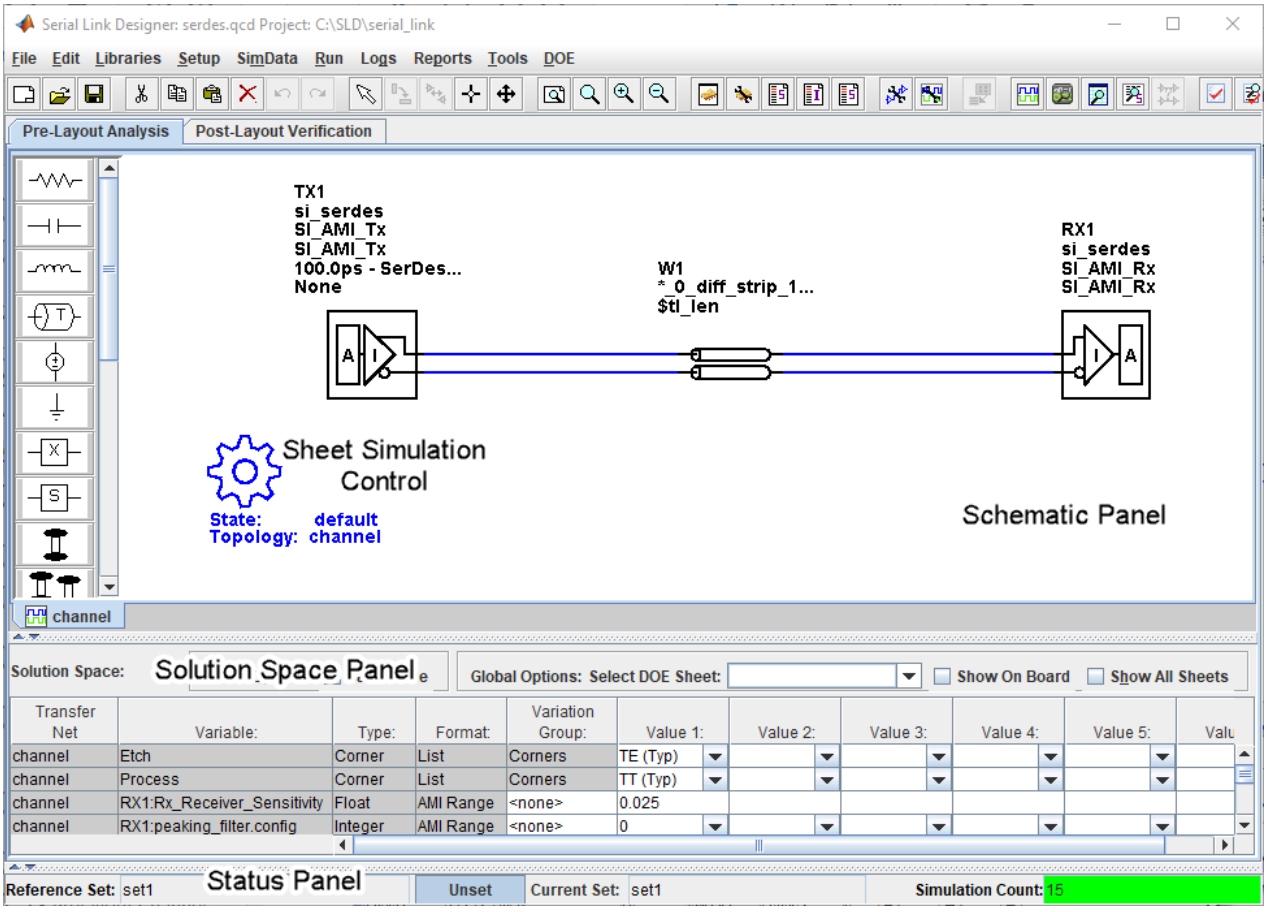
Pre-Layout Analysis of Serial Link

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.



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 2-6 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 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 run,.network characterization, statistical analysis and time domain analysis, and the PDA worst case bit pattern.

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...

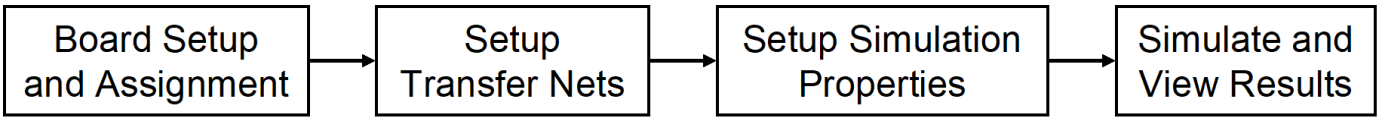
Post-layout verification provides you with an integrated signal integrity and timing environment to verify system-level SI and timing margins for your fully or partially routed PCB design databases.

The post-layout process supports single-board and 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 schematics, post-layout uses the transfer nets from the reference schematic set. If there are no schematic sheets in the reference schematic set of the interface, the **Serial Link Designer** app creates sheets with system transfer nets (STNETs).

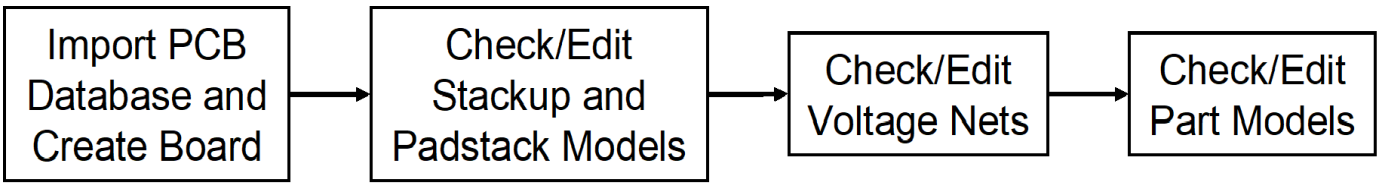
The post-layout verification workflow is the same for each PCB database type. First import the PCB databases, setup the boards, connect the instances if there are multiple boards in the system, run the assignment, setup and analyze the nets, set up simulation properties, then simulate and view the results.



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.



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 and move the PCB database, you cannot re-import the database files.
- **Stackup**

  The **Stackup** tab shows the read stackup from the PCB database and allows control of padstack models. The Stackup Editor on the left side of the tab shows the stackup read from the PCB database and allows you to override the auto-generated trace models. The rightside of the tab controls 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 voltage nets. Non-voltage nets have an NA value in the voltage column.
- **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 will have at least one instance. If you use the same board more than once, you need to create an 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 multiboard 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

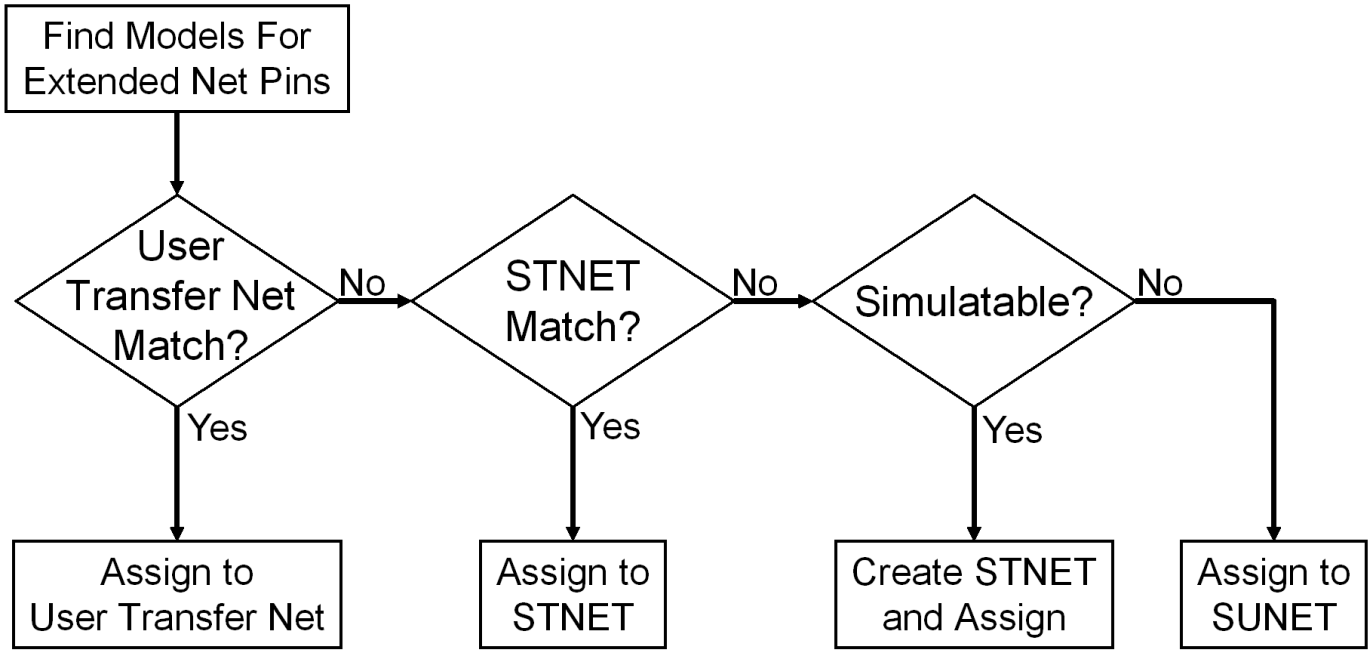
The Assignment process is an automated process for associating nets in the PCB database with transfer nets. This simplifies the setup of the essential net properties in the typical scenarios that you will face:

- **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

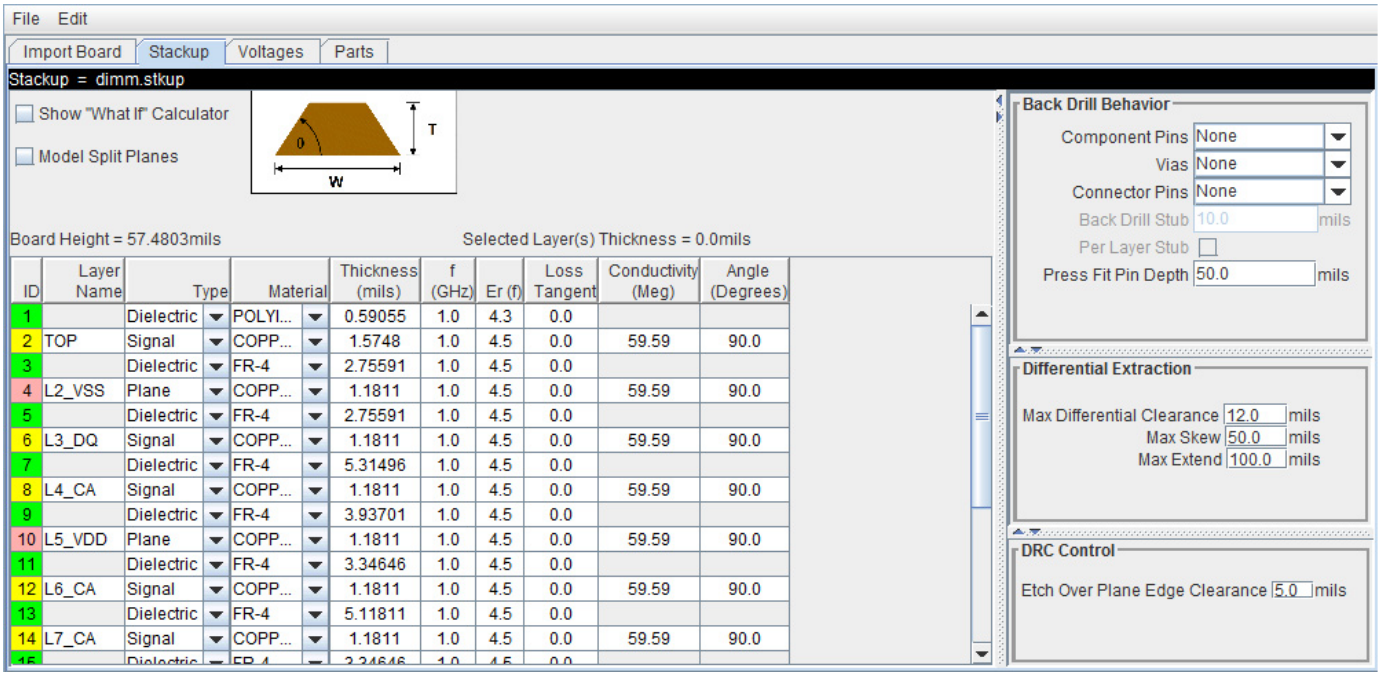
More About

- “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.



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		

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.

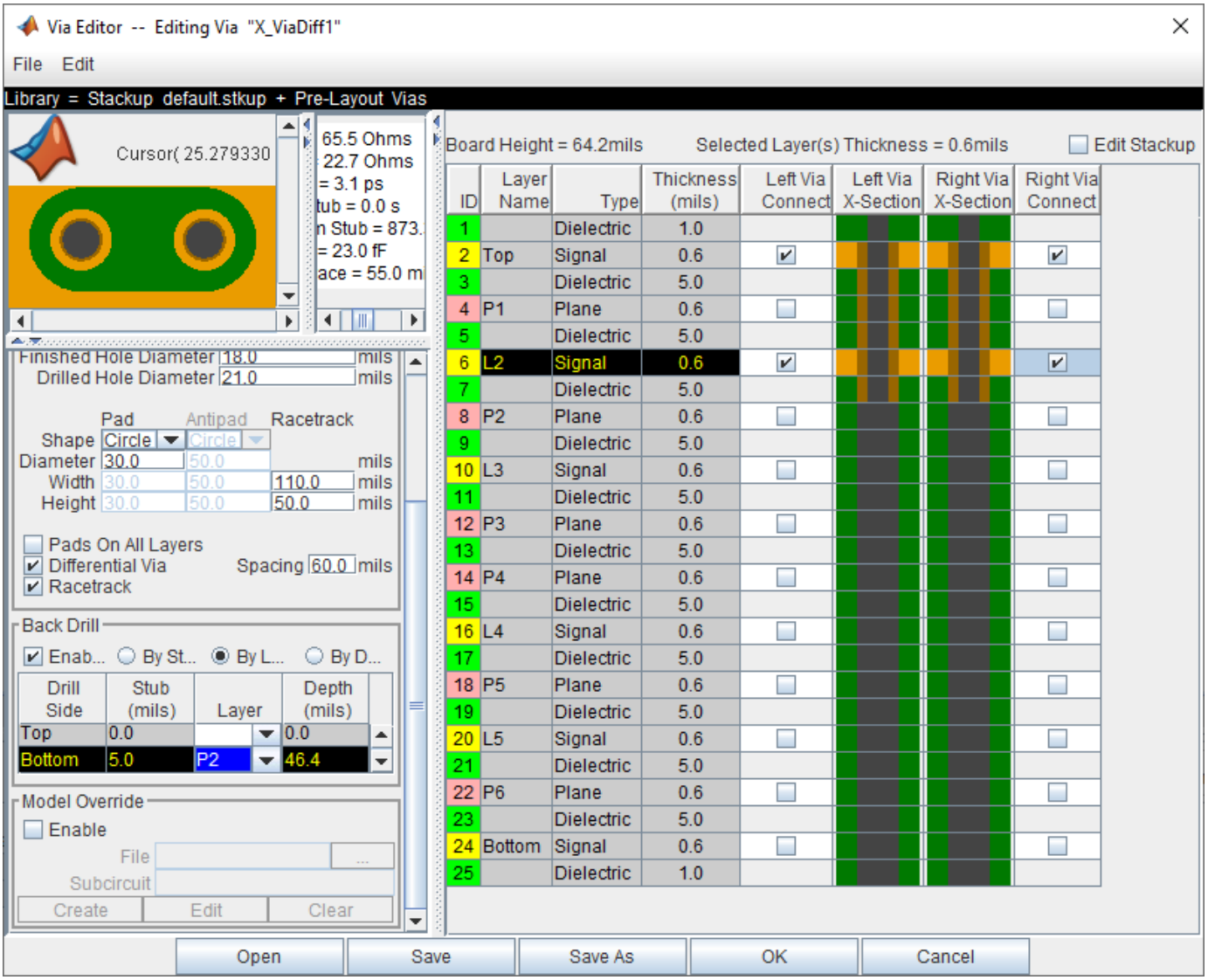
See Also

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 the 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 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 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 During Pre-Layout Simulations

To edit vias during 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 During 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.

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 **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.

You can edit the geometry of a single via, edit the padstack, or override a via model during post-layout.

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.

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.

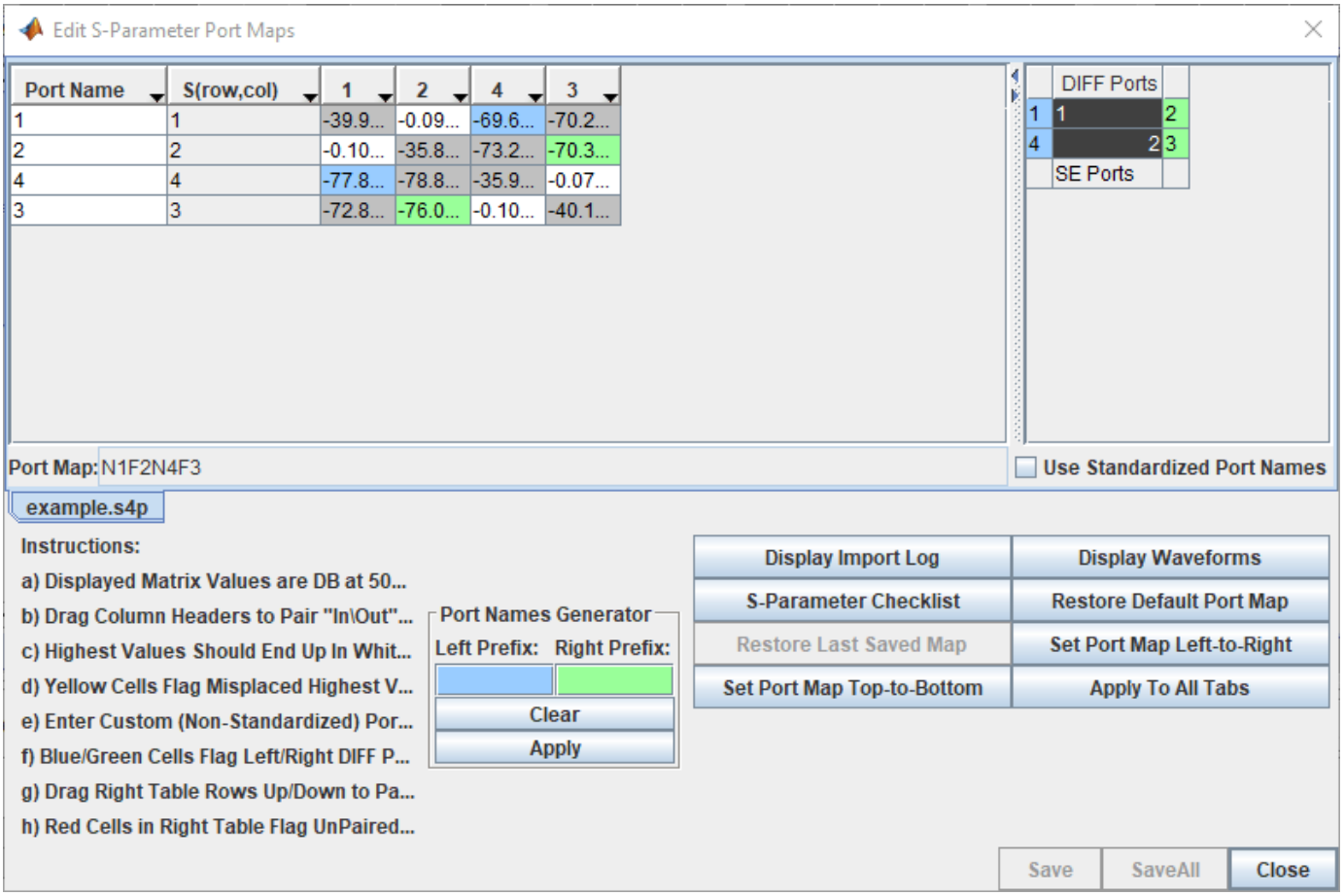
Create New Project

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.

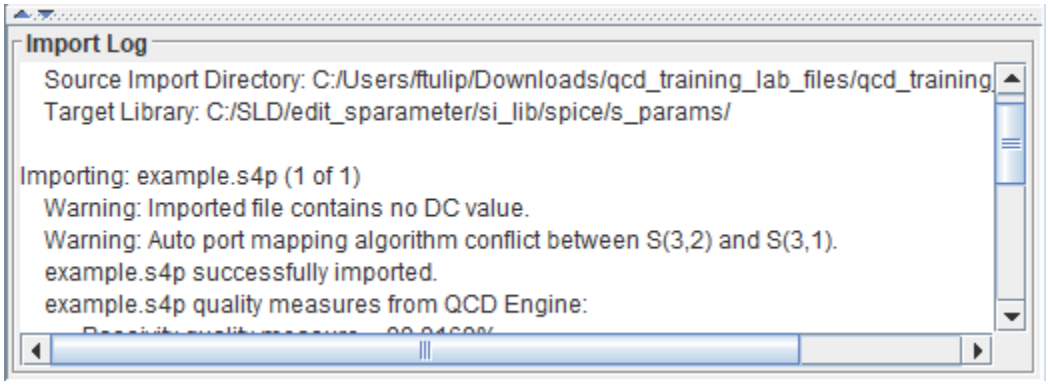
A custom S-Parameter data file (`example.s4p`) is attached as supporting file to this example. This file represents PCB traces on a board. Download the Touchstone® (.s4p) files. To import the S-Parameter data, select **Libraries > Import S-Parameter**. Browse to the location where you saved the downloaded 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.



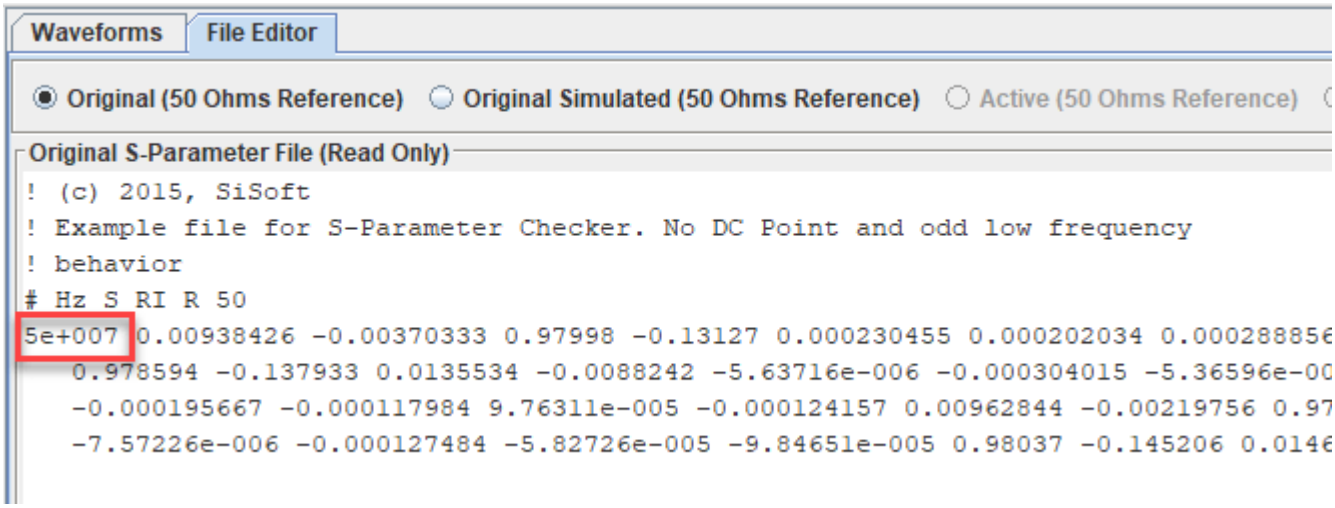
Use S-Parameter Checklist to Analyze S-Parameter Data

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.



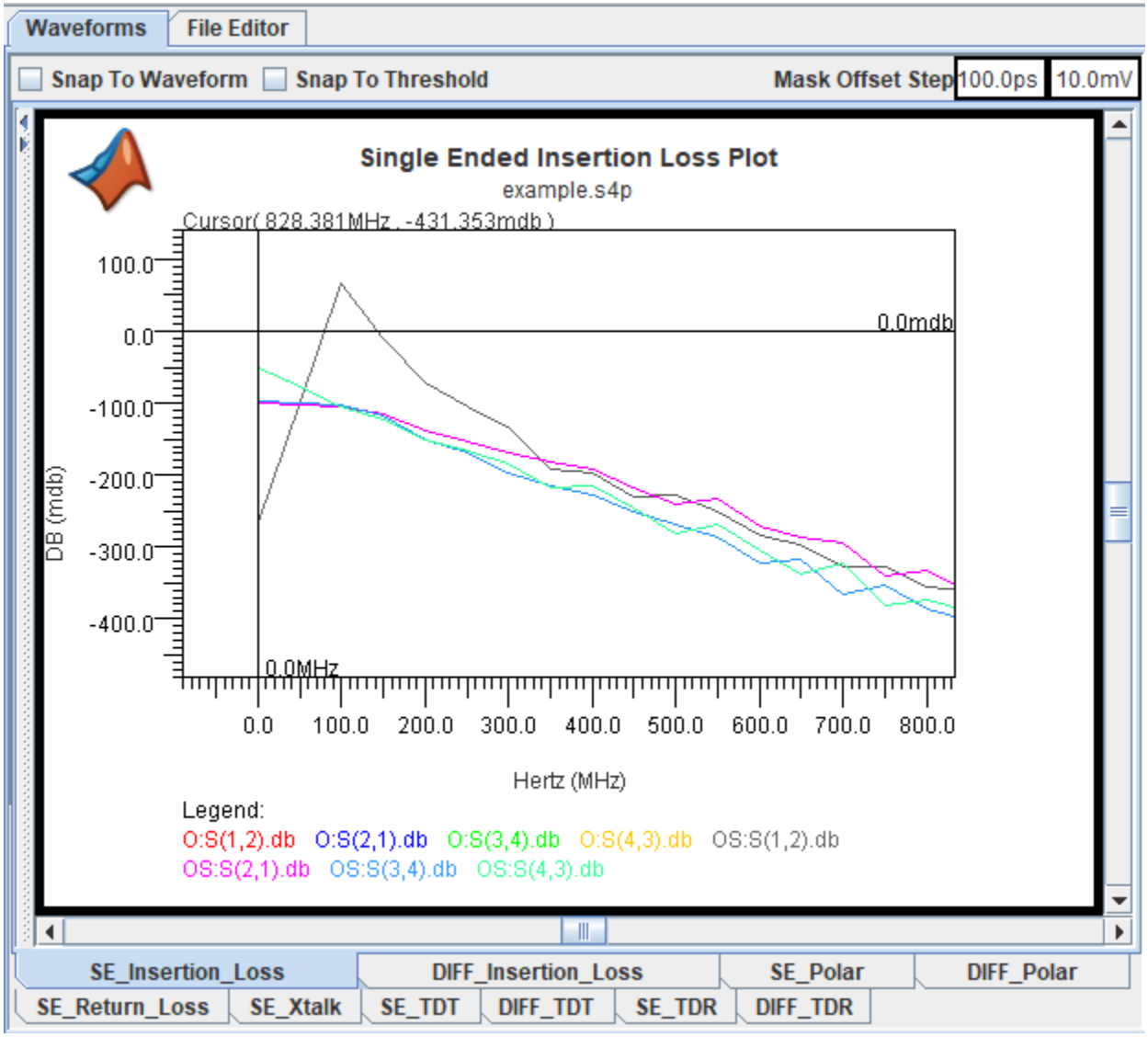
If you look at the **File Editor** tab in the right panel you can see that the lowest frequency point is 50MHz in the S-Parameter file. This is why the import log reported the no DC value warning.



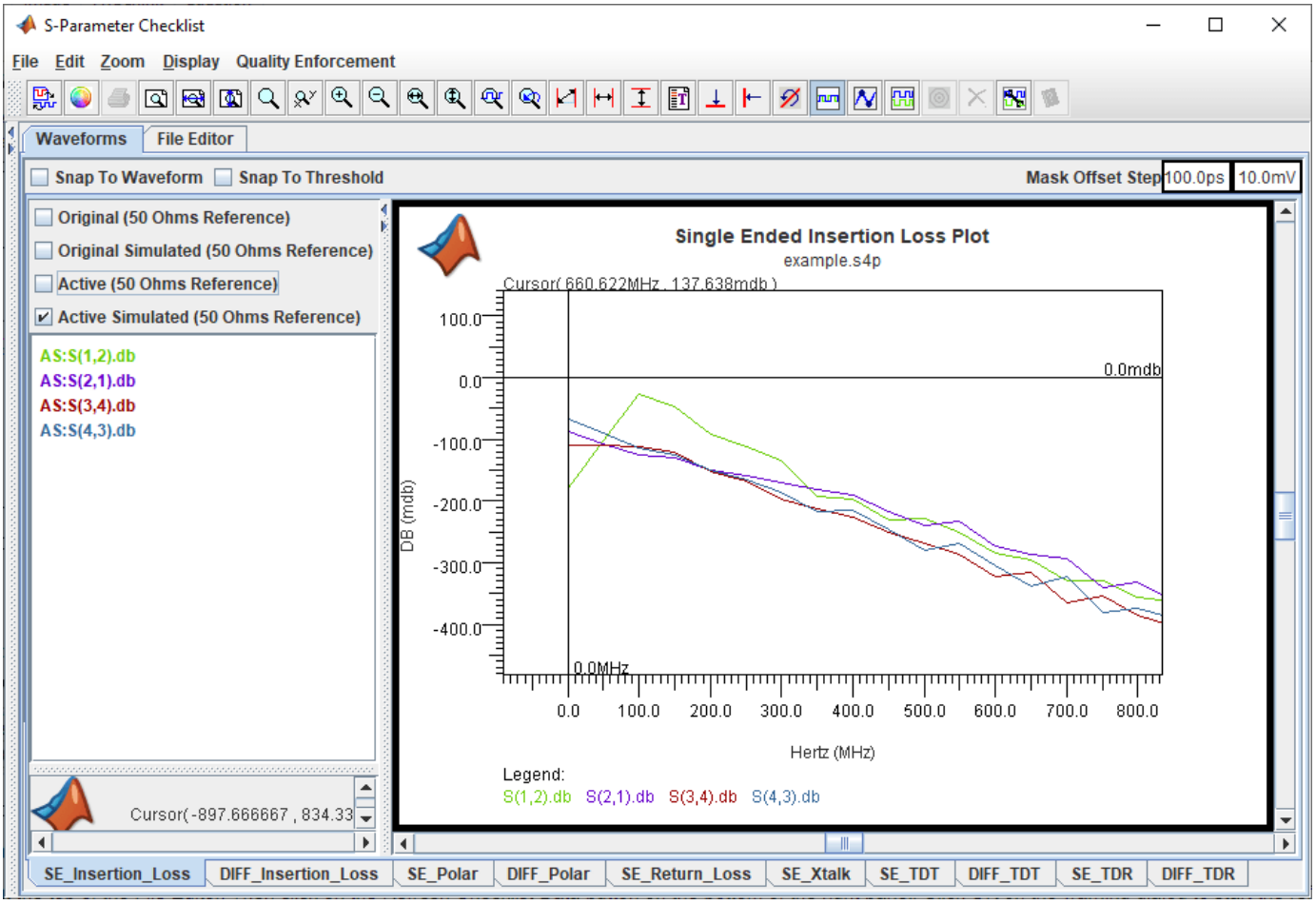
The warning about the port mapping algorithm is from the comparison of the 50MHz data in the file that is used to determine the pinout of the schematic symbol. This is due to the odd nature of the data in this S-Parameter file at low frequencies.

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.



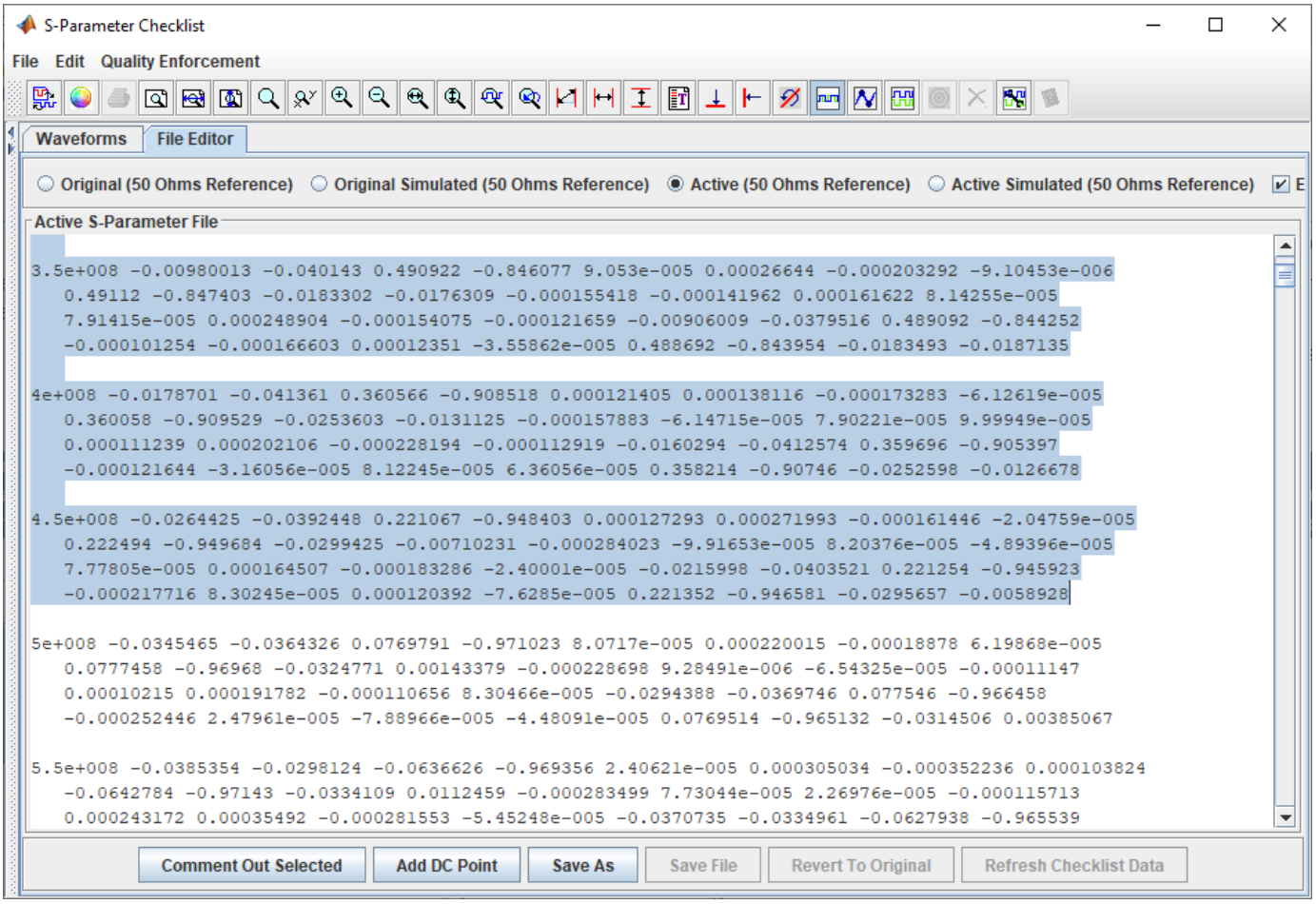
Enforce passivity to investigate further. Select Quality **Enforcement > Make Passive** from the toolstrip. To refresh the data, go back to the **Visual_Inspection** tab. Select **Enable Editing** at the top of the File Editor. Then click on the **Refresh Checklist Data** button on the bottom of the 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.



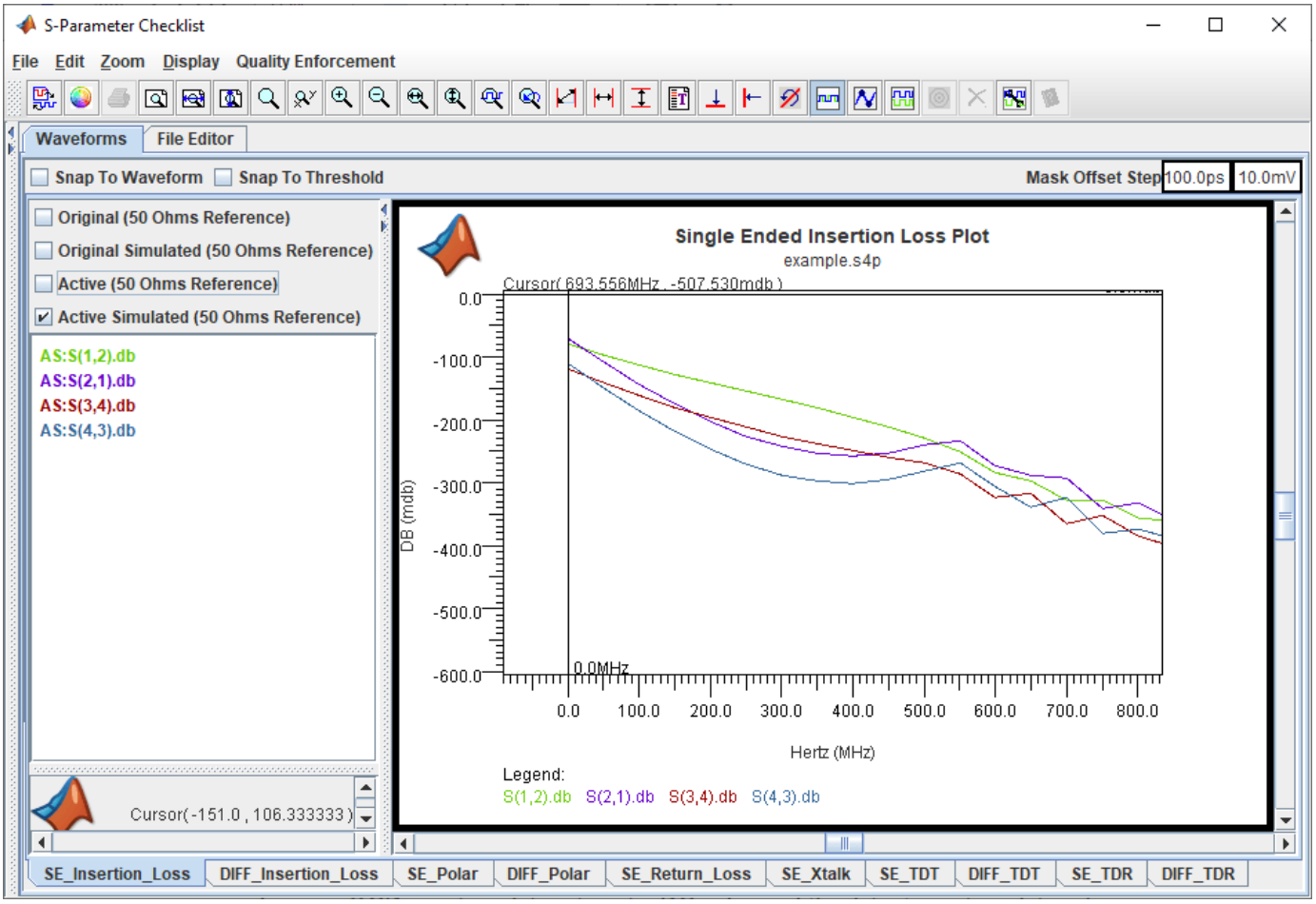
The insertion loss data is now all passive, but there are still some strange issues at the lowest frequencies. The loss is greater at DC than at low frequencies. This does not make sense for PCB traces.

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.



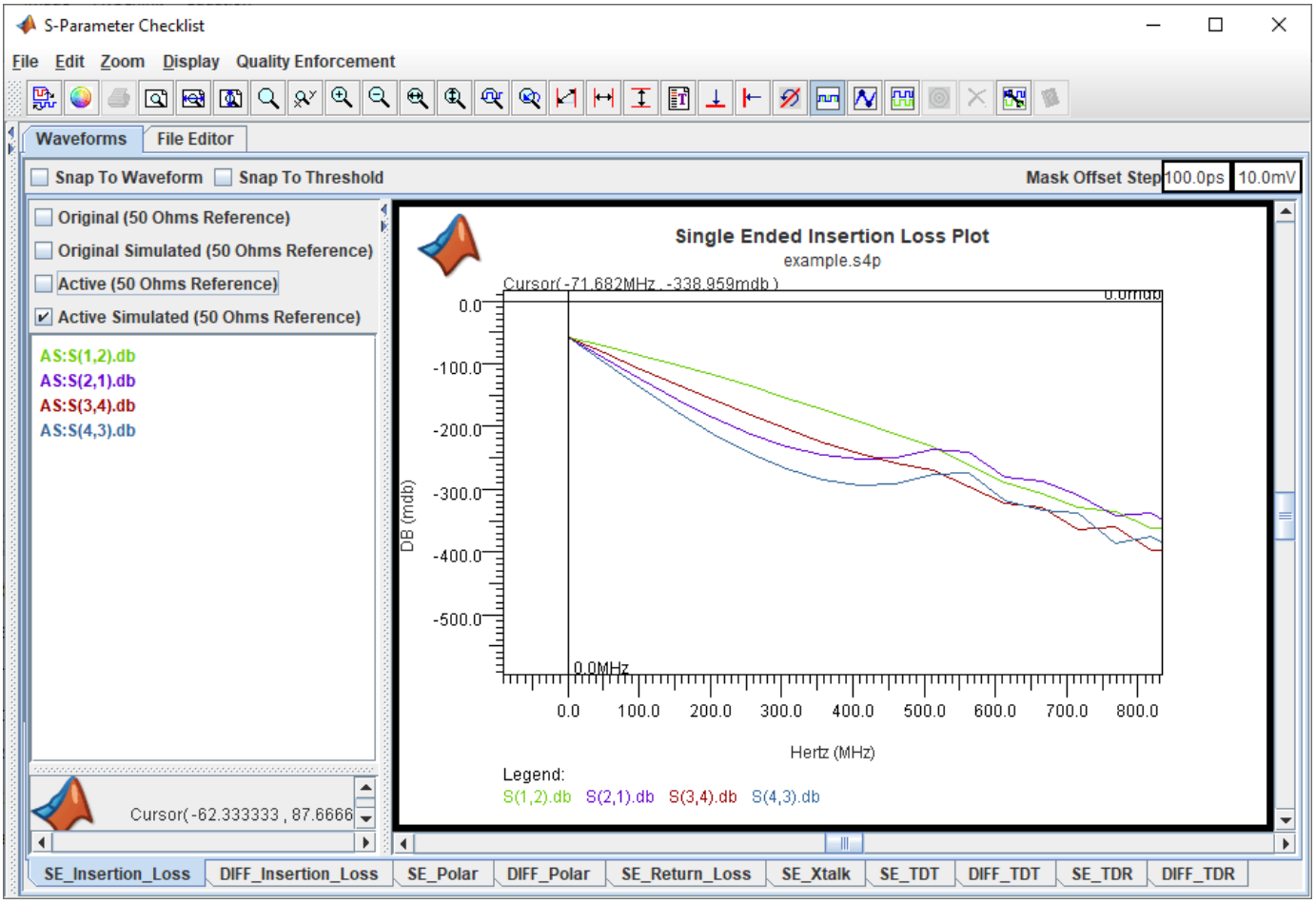
Refresh the checklist data and go back to the **SE_Inserion_Loss** tab in the left panel.The loss is now increasing with frequency. The only issue is that the S(1,2) and S(4,3) data do not have the same DC point. This is because the **Serial Link Designer** app is extrapolating the data to get a DC value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



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.

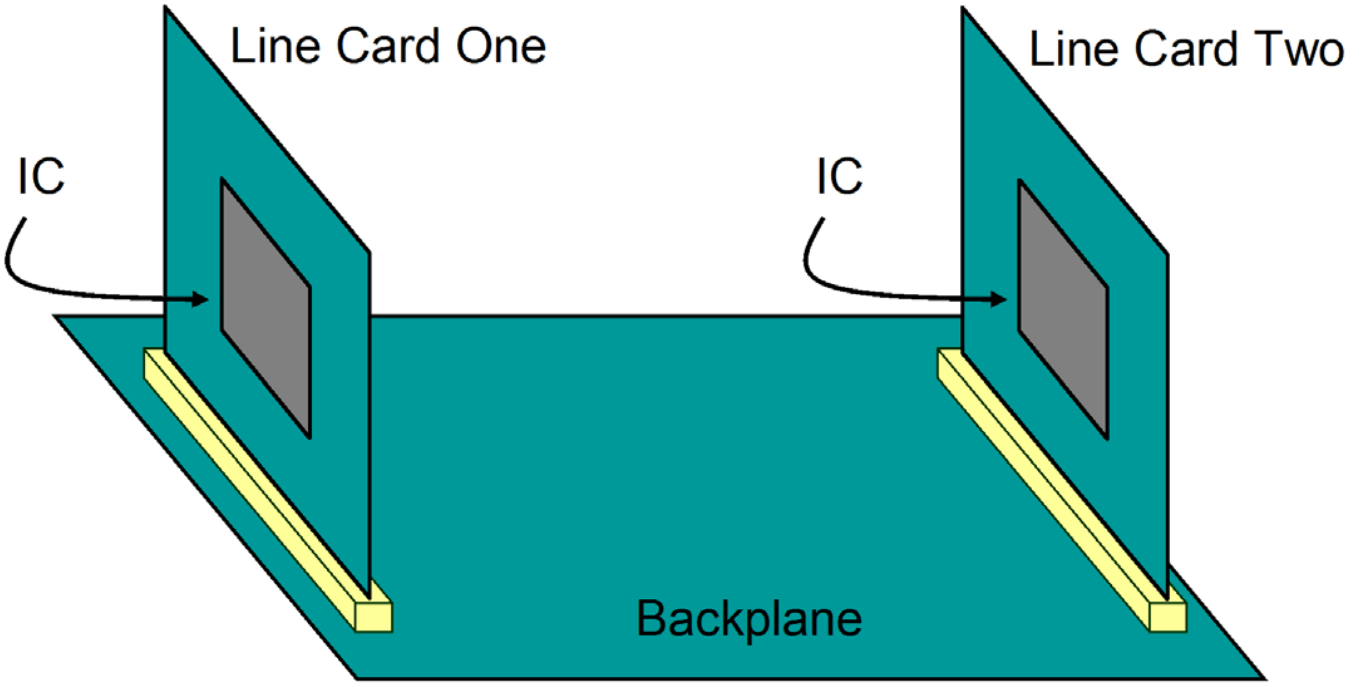
Click **Insert DC Point**, then save the file and refresh the checklist data. Go back to the **SE_Insertion_Loss** tab in the left tab. Now the insertion loss is increasing with frequency and all the data has the same DC point.


S-Parameter Checklist
File Edit Zoom Display Quality Enforcement
Waveforms
File Editor
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
Cursor(-62.333333 , 87.6666
Single Ended Insertion Loss Plot
example.s4p
Cursor(-71.682MHz , -338.959mdb)
0.0mdb
0.0
-100.0
-200.0
-300.0
-400.0
-500.0
DB (mdb)
0.0MHz
0.0
100.0
200.0
300.0
400.0
500.0
600.0
700.0
800.0
Hertz (MHz)
Legend:
S(1,2).db S(2,1).db S(3,4).db S(4,3).db
SE_Insertion_Loss
DIFF_Insertion_Loss
SE_Polar
DIFF_Polar
SE_Return_Loss
SE_Xtalk
SE_TDT
DIFF_TDT
SE_TDR
DIFF_TDR

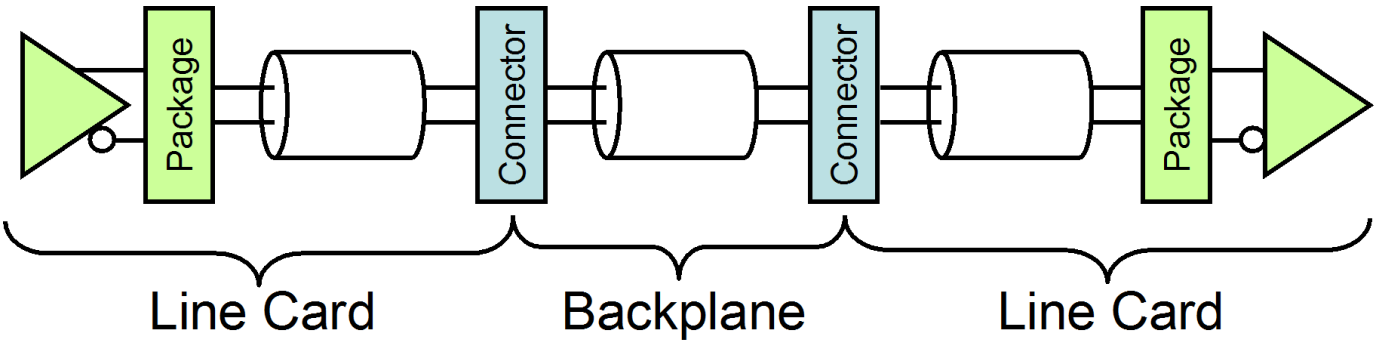
Analyze Backplane with Line Cards

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:



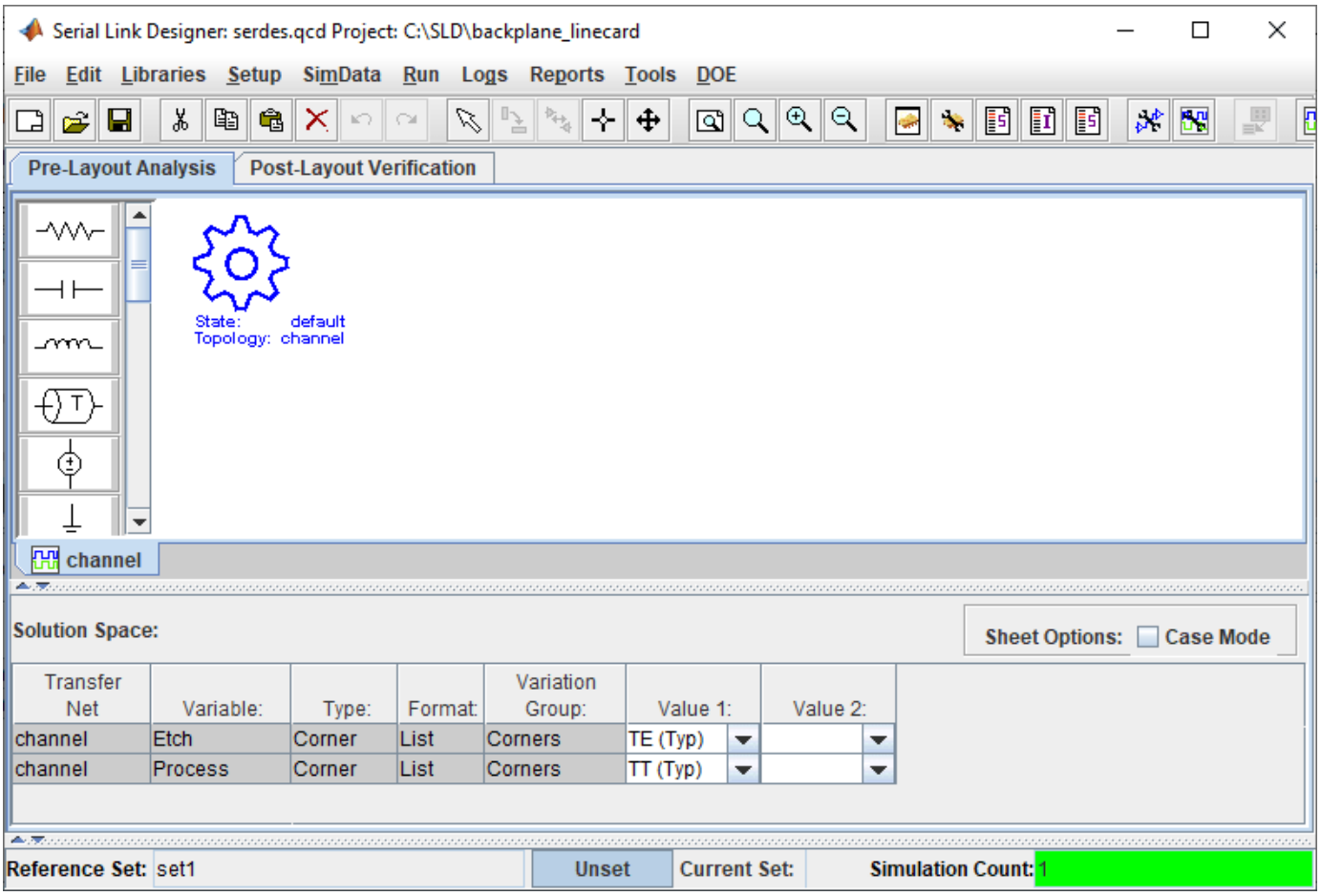
The packages and connectors are modeled with S-Parameters. The traces are modeled with w-lines.

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.



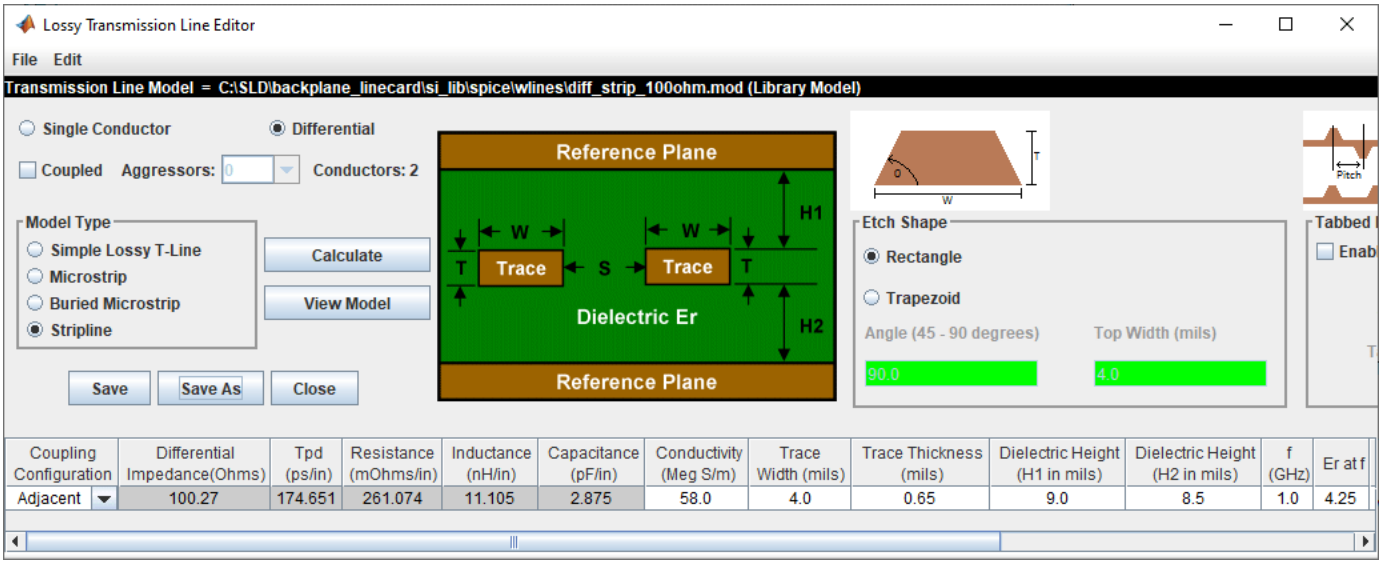
Setup Libraries

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**.

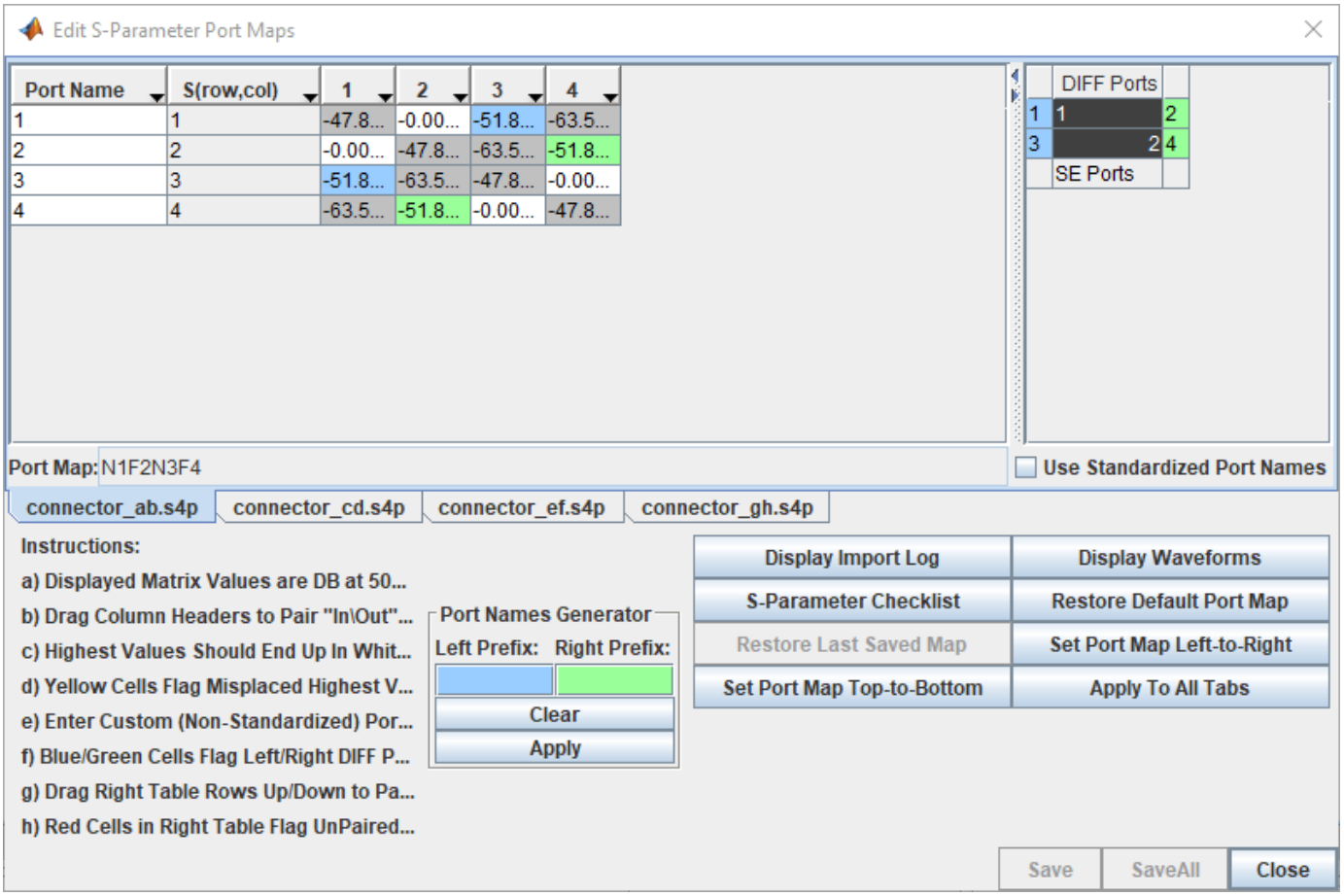
The traces are 4 mils wide and 0.65 mils thick. They are 9.0 mils above and 8.5 mils below planes with a dielectric constant of Er 4.25. The trace separation is 5 mils. So change the parameters **Dielectric Height (H1 in mils)** to 9, **Dielectric Height (H2 in mils)** to `8.5`, and **Differential Separation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



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.

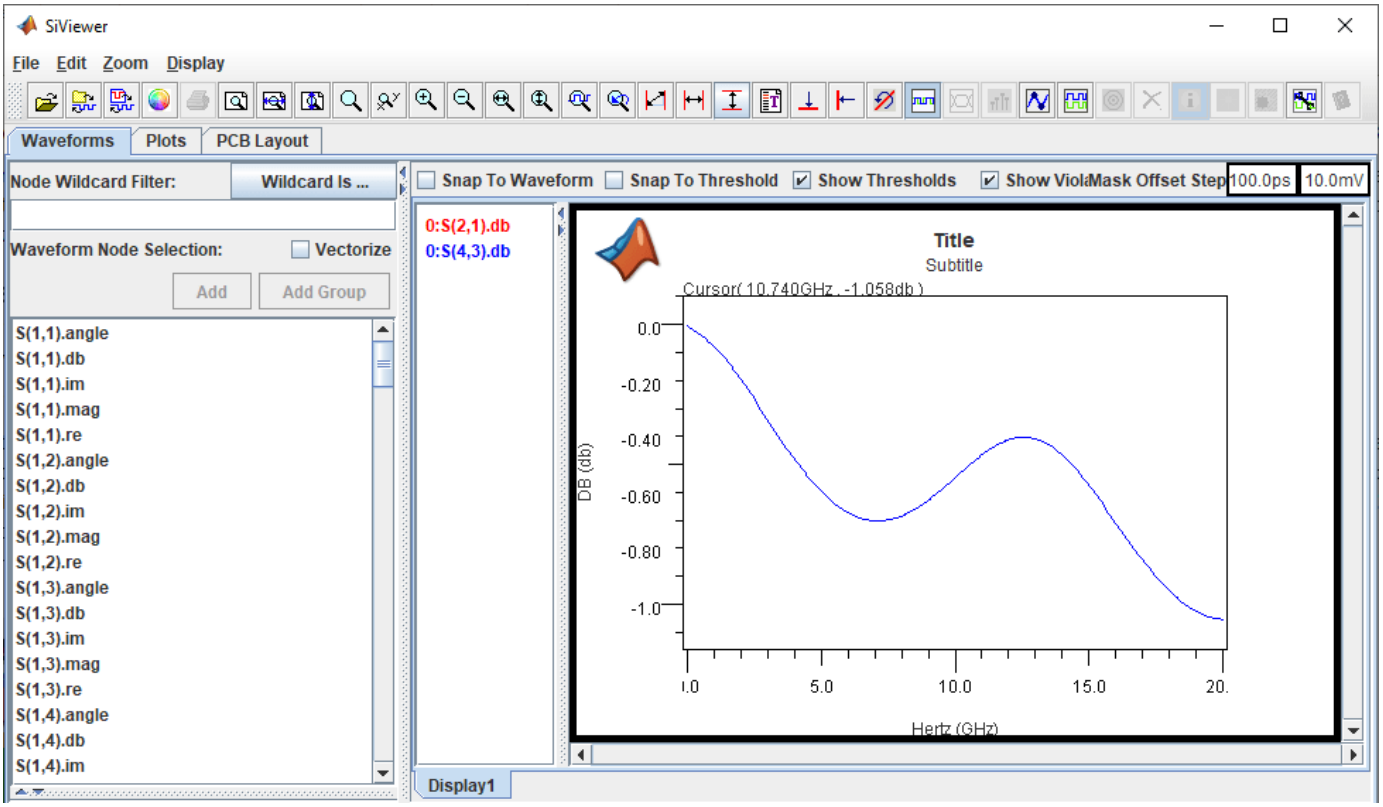
Four custom S-Parameter data files (connector_ab.s4p, connector_cd.s4p, connector_ef.s4p, and connector_gh.s4p) are attached as supporting files to this example. Download all four Touchstone® (.s4p) files. To import the connector S-Parameter data, select **Libraries** > **Import S-Parameter**. Browse 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.



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 the differential ports.

To view the two through path dB vs. frequency responses, click the **Display Waveforms** button. This launches the **Signal Integrity Viewer** app.



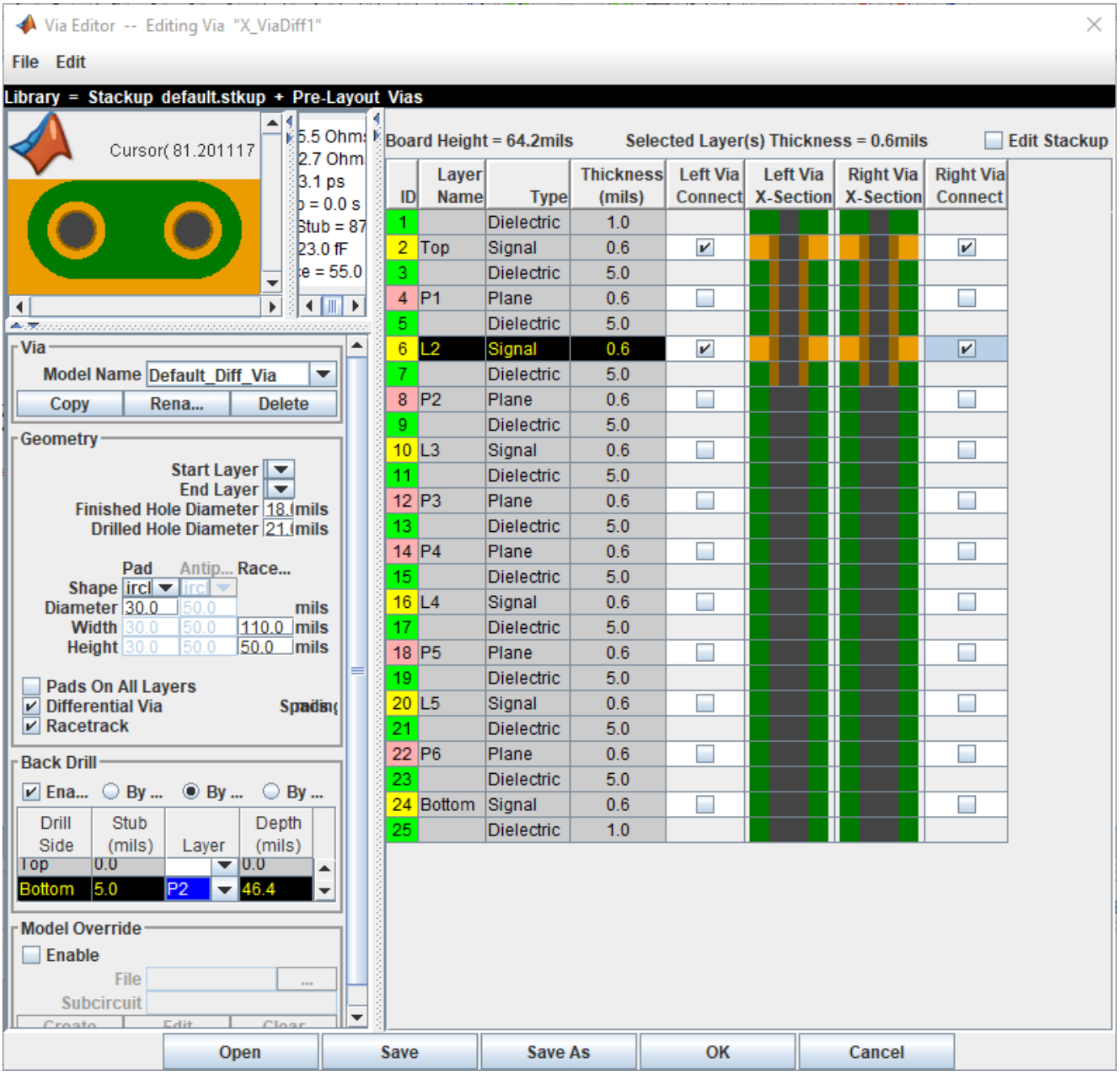
You can add new display to view all the data in real/imaginary, magnitude/angle and dB. Close the **Signal Integrity Viewer** app, Edit S-Parameter Port Maps dialog box, and Import S-Parameter File(s) dialog box.

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** in the . Switch to `<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.



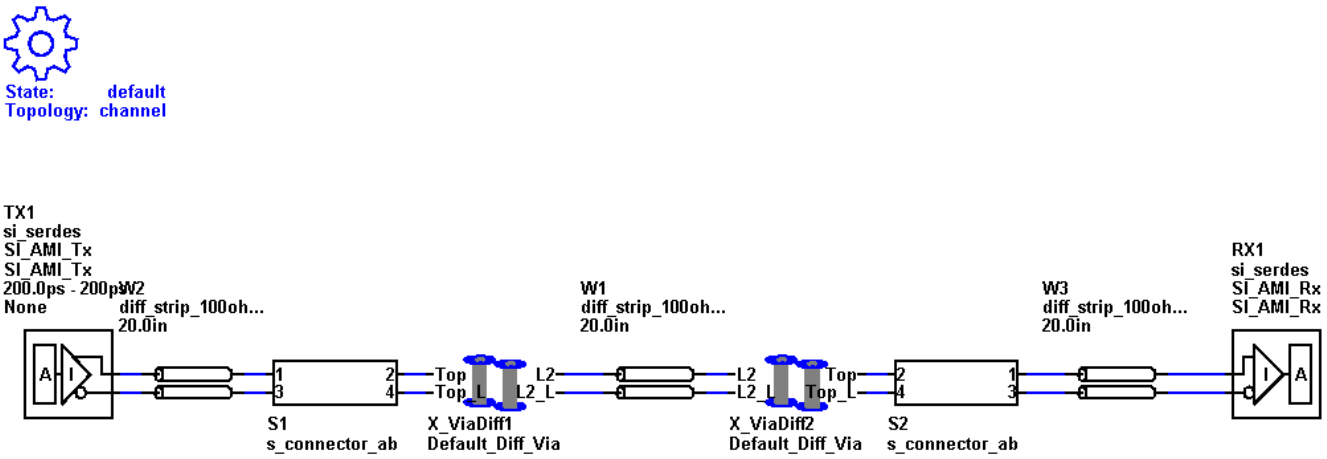
Save and close the Via Editor dialog box. Copy, paste, and mirror another via to the right of the transmission line.

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.

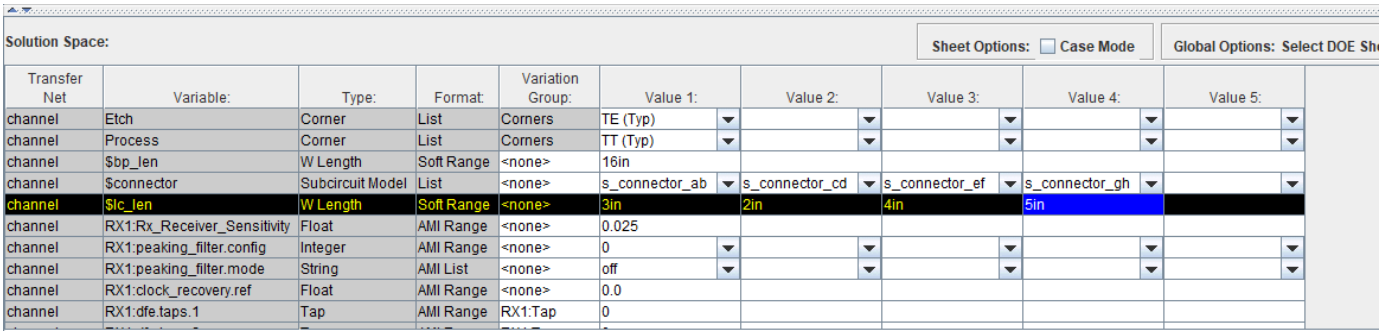
Validate the schematic set by selecting **Run > Validate Current Schematic Set**. The validation should run without warning or errors.

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.

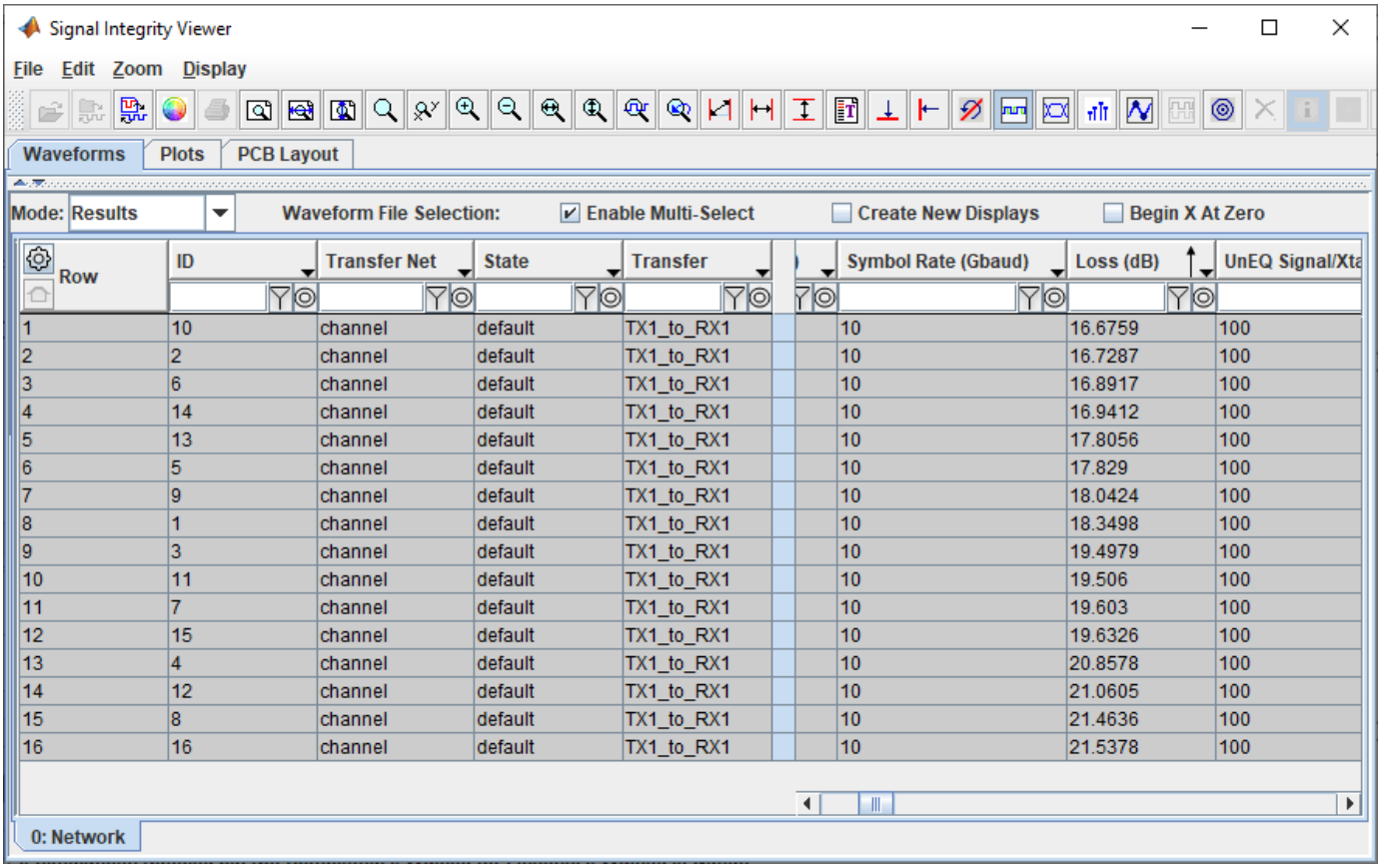
To sweep the line card length, select the `$lc_len` **Variable** and add the values 2in, 4in, and 5in. Save the changes.



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				

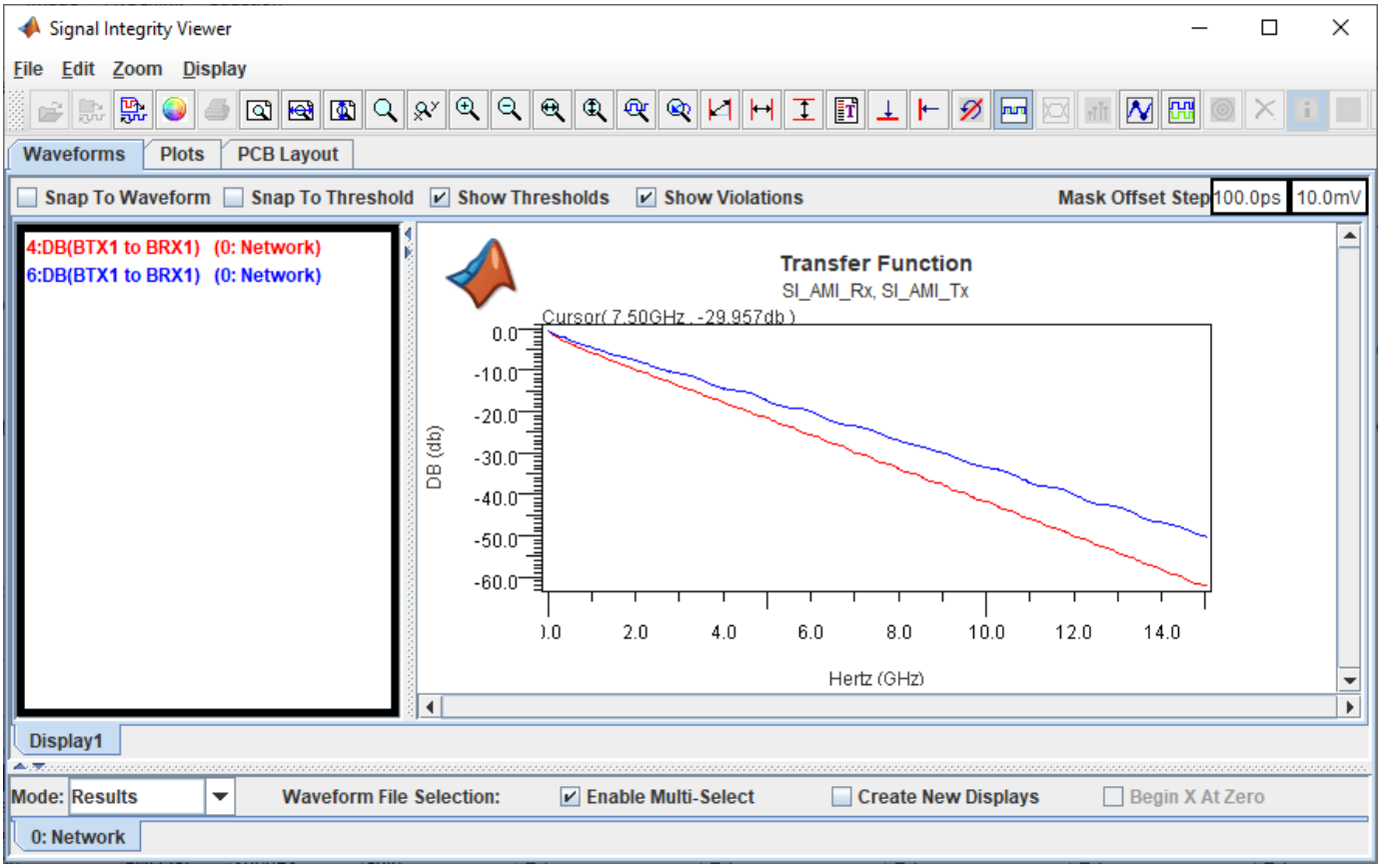
Run the simulation by selecting **Run > Simulate Selected**. In the Prelayout Channel Analysis dialog box, select **Validate**, **Generate Netlists**, **Perform Channel Analysis**, and **Autoload Results**. Make sure **Include Statistical Analysis** and **Include Time Domain Analysis** are unchecked, 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.



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)**.



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.

Signal Integrity Viewer

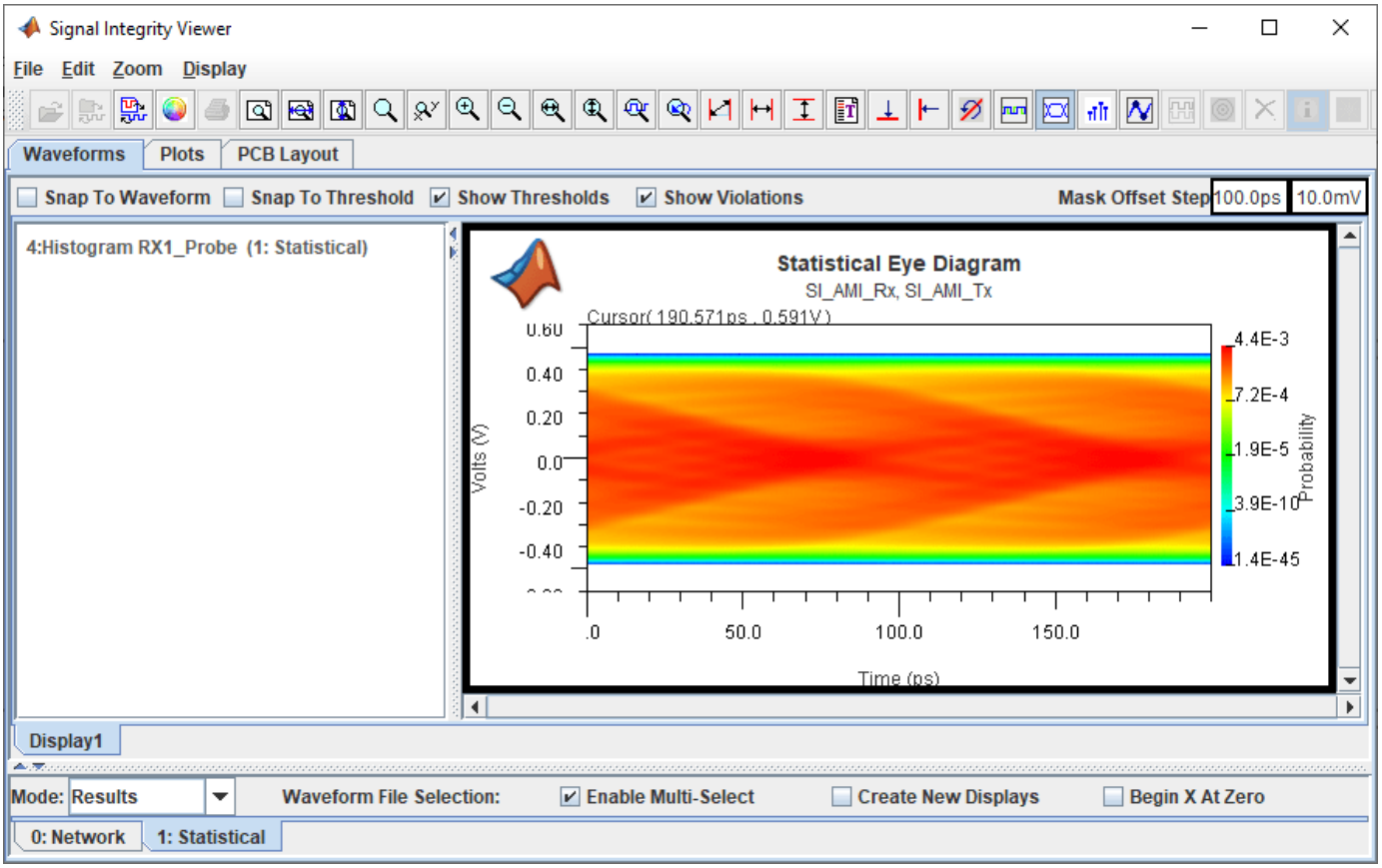
File Edit Zoom Display

Waveforms | Plots | PCB Layout

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

0: Network | 1: Statistical



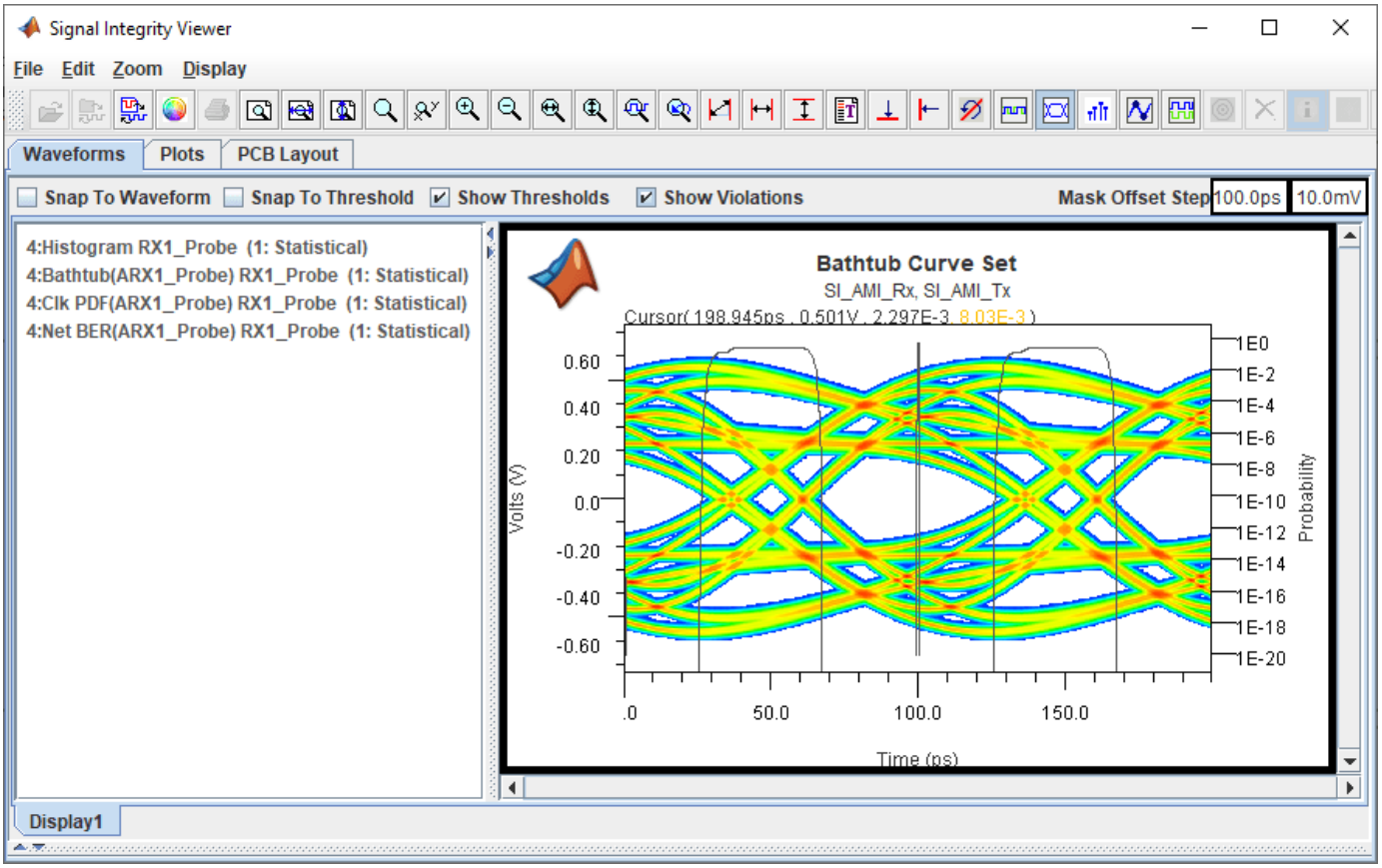
Click on the **Stat BER** header to get the smallest BER (4.64e-10 in this example) at the top. To see the tap settings for the top row right-click on the row in the table and select **Show Solution Space**. In the dialog that appears you can see the tap settings: `TX1.tap_filter.0 = 0.7` and `TX1.tap_filter.1 = -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 run now has positive statistical eye margin. Select one of the rows with positive margin, right click and select **Show BER**. **Y**ou can see the statistical eye, the bathtub curve and the clock PDF.



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.

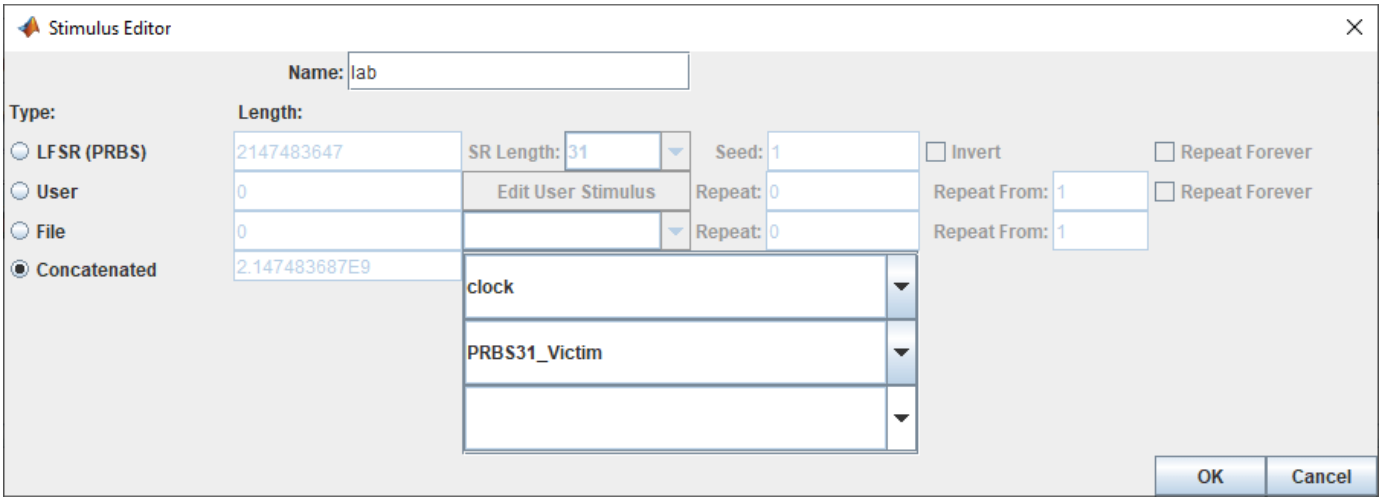
To set up the time domain analysis, in the Solution Space panel of the **Serial Link Designer** app, delete the **Value 2** (2in) of the `$lc_len` **Variable**. Set the values of **Variable** `RX1:peaking_filter.mode` **Value 1** to `auto` and **Value 2** to blank. Change the **Variation Group** of TX1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

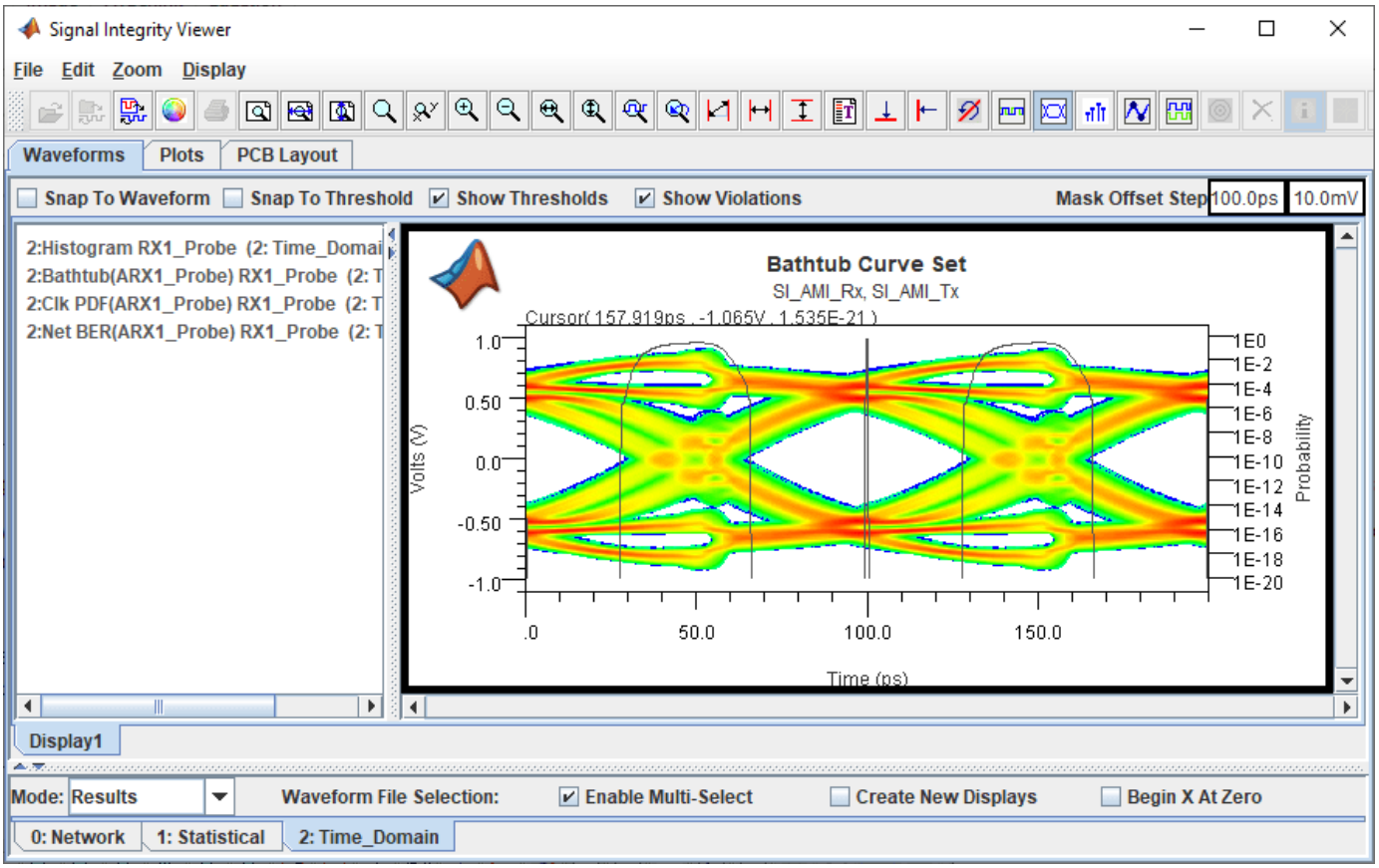
Select **Setup > Simulation Parameters** and check that the **Time Domain Stop** is set to `1,000,000 UI` and the **Record Bits** is set to `2,500 UI`. Right-click on the RX symbol on the schematic and select **Edit AMI File(s)**. In the AMI file that opens the **Ignore_Bits** parameter is set to `500,000 UI`. So the simulation runs 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.

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.

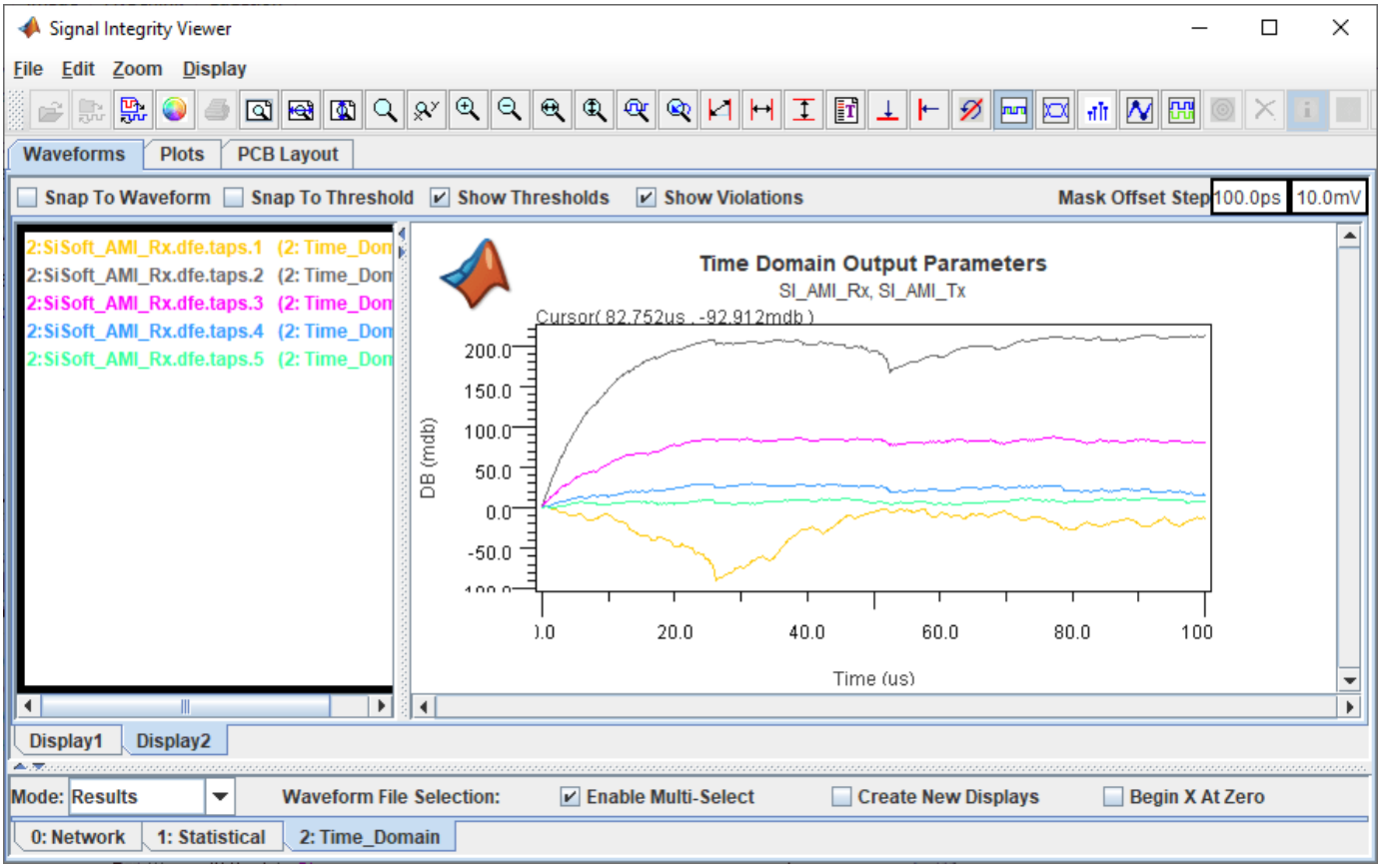


On the Designator Element Properties dialog box, select the **Stimulus** as `lab`. Save the changes. Run the simulation and select Include Time Domain Analysis in the Prelayout Channel Analysis dialog box.

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**.



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.



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

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.

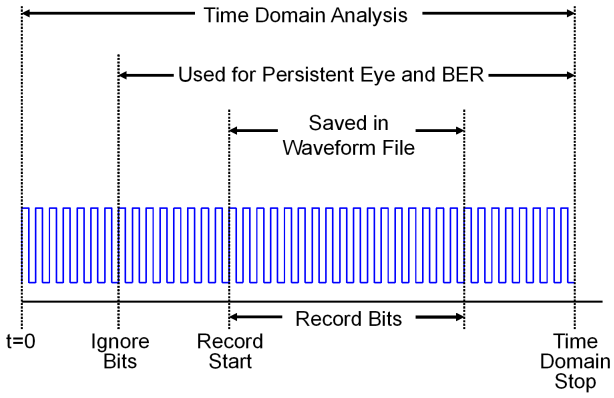
Parameter	Description
Samples Per Bit	The number of time steps in a bit time. Defines the time step used in the STAT Mode “.tran” statement.
Max Channel Delay	User-supplied value for the maximum length of the channel impulse response. Goes into FFT block size calculation, which also defines the message 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.

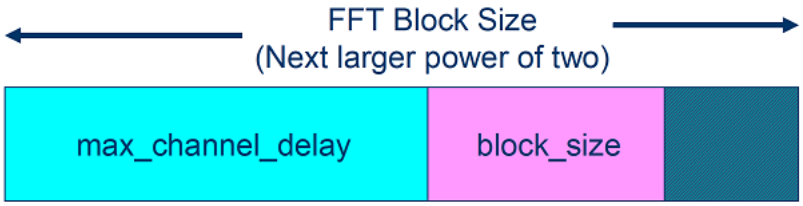
STATify	Control how statistical techniques are applied to time domain simulations and Getwave-only models. The values are:
	• TD_Extrapolation: Extrapolates the bathtub curve to account for the effects of ISI at lower probabilities than can be derived from the time domain simulation alone. When this 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. cannot be deleted or renamed. The victim and aggressor patterns are used in pre-layout coupled/SSO simulation mode.

When defining a new stimulus, you have two choices:

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

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

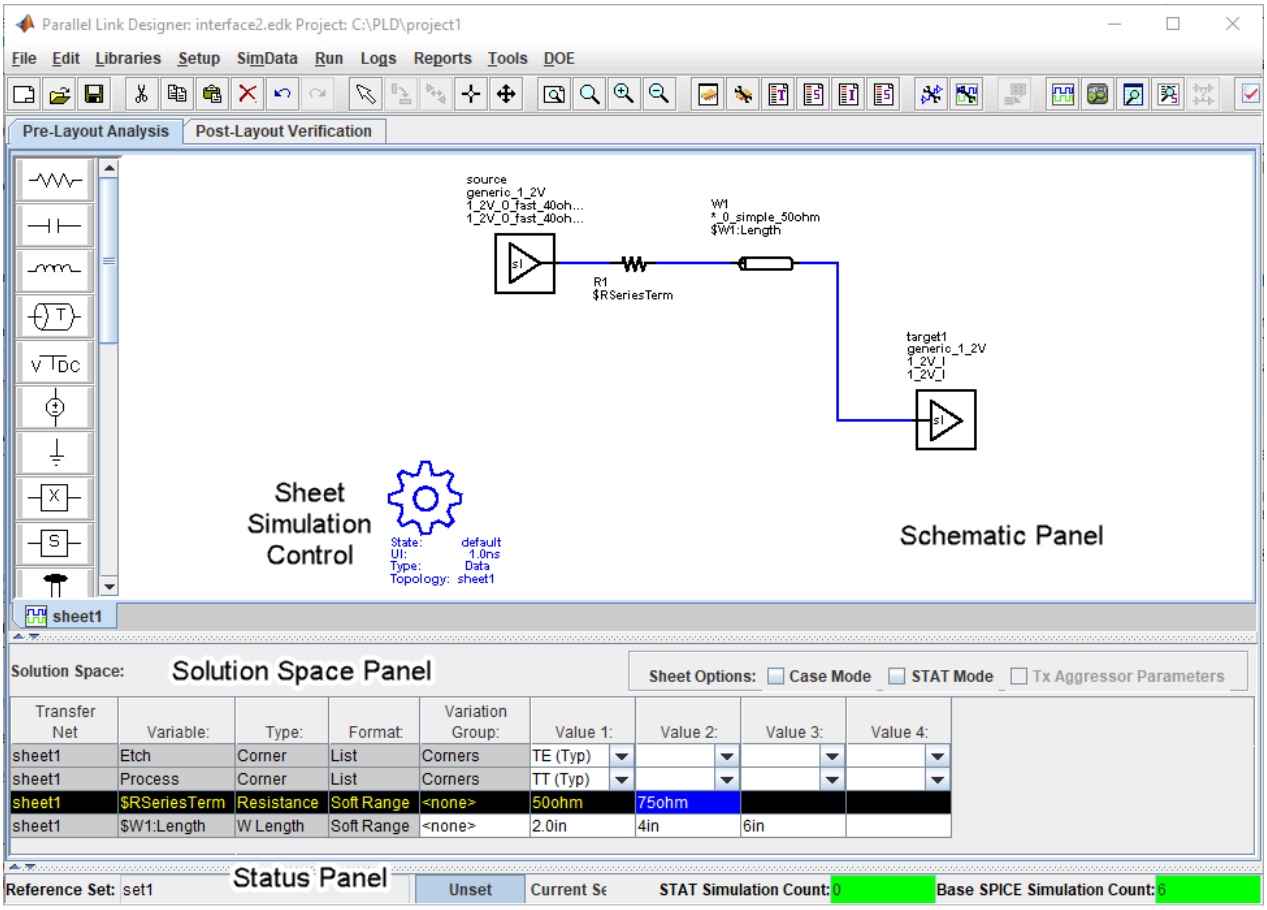
Pre-Layout Analysis of Parallel Link

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.



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 indicates syntax errors in the data and a summary of the HSPICE waveform analysis failures for 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.
Mask By Channel	Available when there is a DDR4 DQ/DQS analysis.

Report	Description
Mask By Receiver Corner	Available when there is a DDR4 DQ/DQS analysis.
Mask by Driver Receiver Corner	Available when there is a DDR4 DQ/DQS analysis.
Mask Training Details	Available in post-layout when there is a DDR4 DQ/DQS analysis.
Mask Eye Details	Available when there is a DDR4 DQ/DQS analysis.
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.
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 HSPICE generation report contains the HSPICE generation log with information about the Spice decks generated and any errors. 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 HSPICE generation report after running a pre-layout or post-layout simulation, select **Reports > HSPICE 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

Post-Layout Verification of Parallel Link

Post-Layout Verification of Parallel Link

In this section...

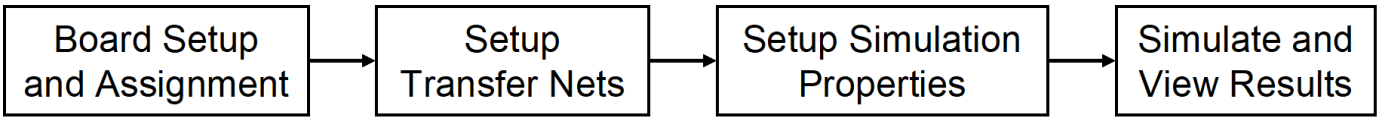
Post-layout verification provides you with an integrated signal integrity and timing environment to verify system-level SI and timing margins for your fully or partially routed PCB design databases.

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 parallel link design project. If the interface you are working in has pre-layout schematics, post-layout uses the transfer nets from the reference schematic set. If there are no schematic sheets in the reference schematic set of the interface, the **Parallel Link Designer** app creates sheets with system transfer nets (STNETs).

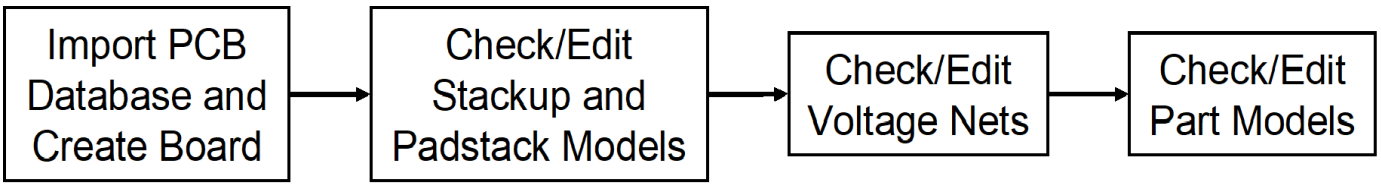
The post-layout verification workflow is the same for each PCB database type. First import the PCB databases, setup the boards, connect the instances if there are multiple boards in the system, run the assignment, setup and analyze the nets, set up simulation properties, then simulate and view the results.



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.



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 and move the PCB database, you cannot re-import the database files.
- **Stackup**

  The **Stackup** tab shows the read stackup from the PCB database and allows control of padstack models. The Stackup Editor on the left side of the tab shows the stackup read from the PCB database and allows you to override the auto-generated trace models. The rightside of the tab controls 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 voltage nets. Non-voltage nets have an NA value in the voltage column.
- **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 will have at least one instance. If you use the same board more than once, you need to create an 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 multiboard 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

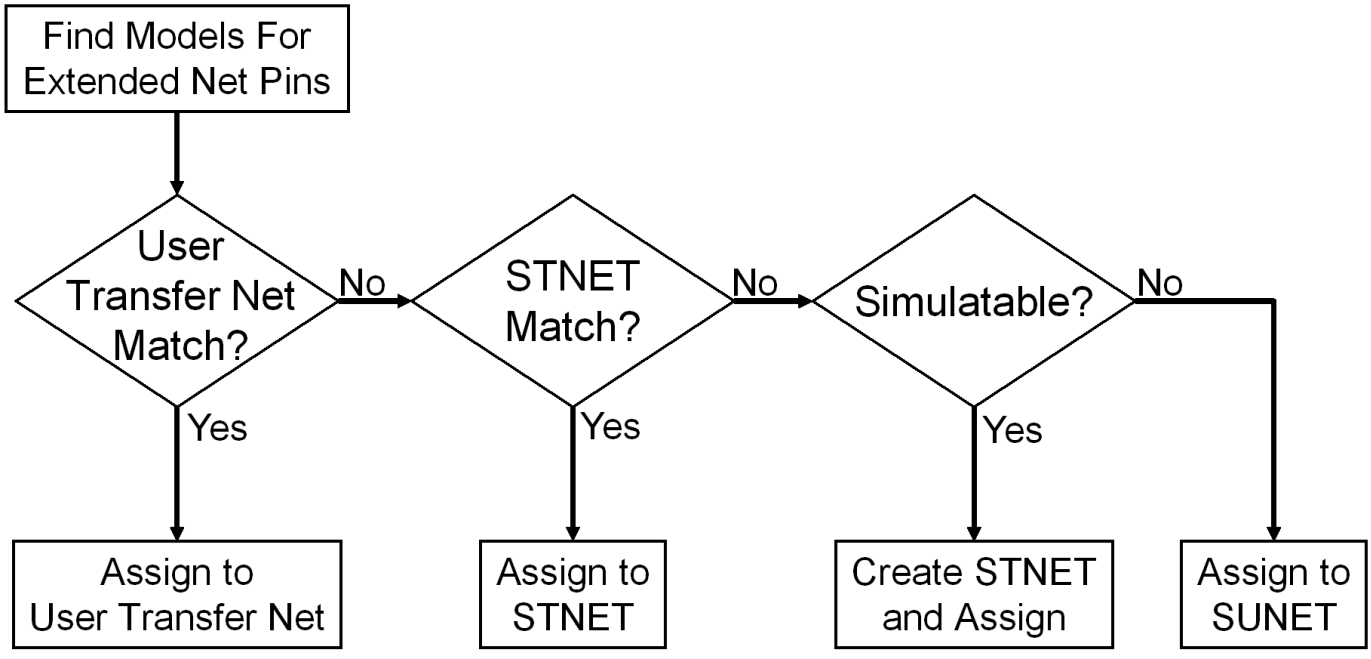
The Assignment process is an automated process for associating nets in the PCB database with transfer nets. This simplifies the setup of the essential net properties in the typical scenarios that you will face:

- **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

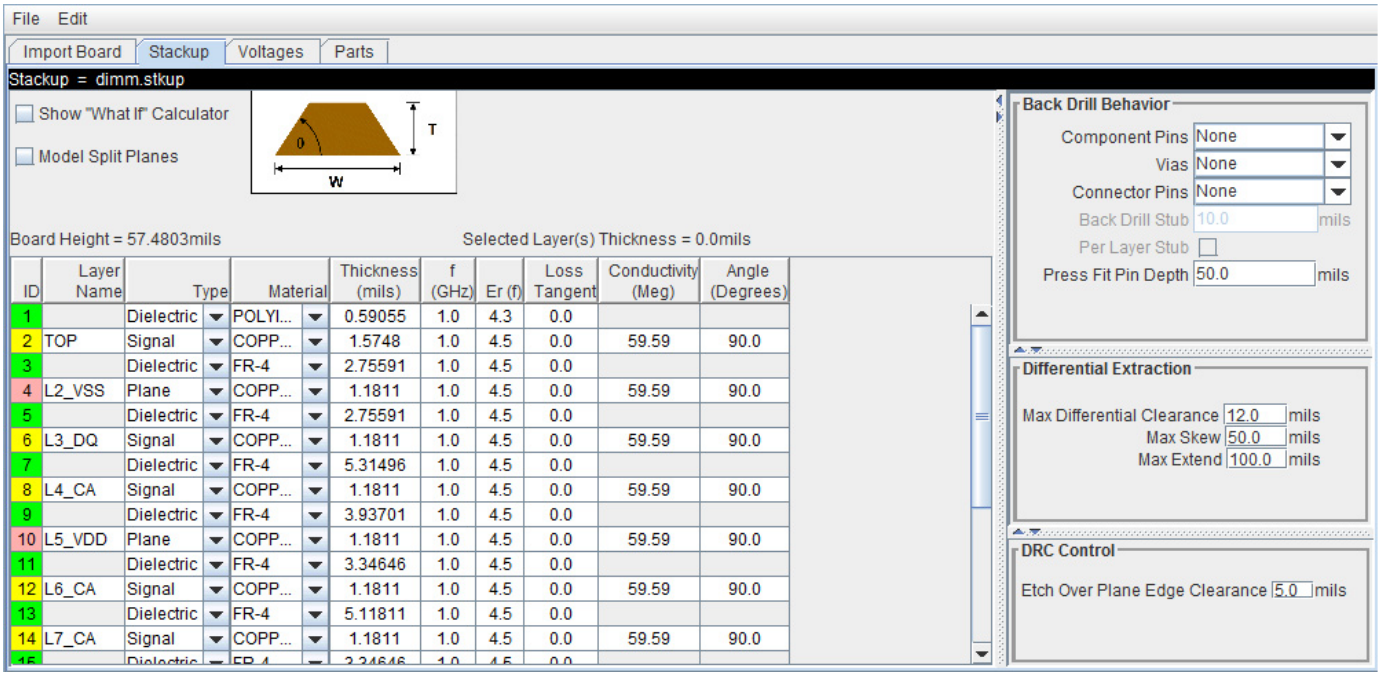
More About

- “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 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.

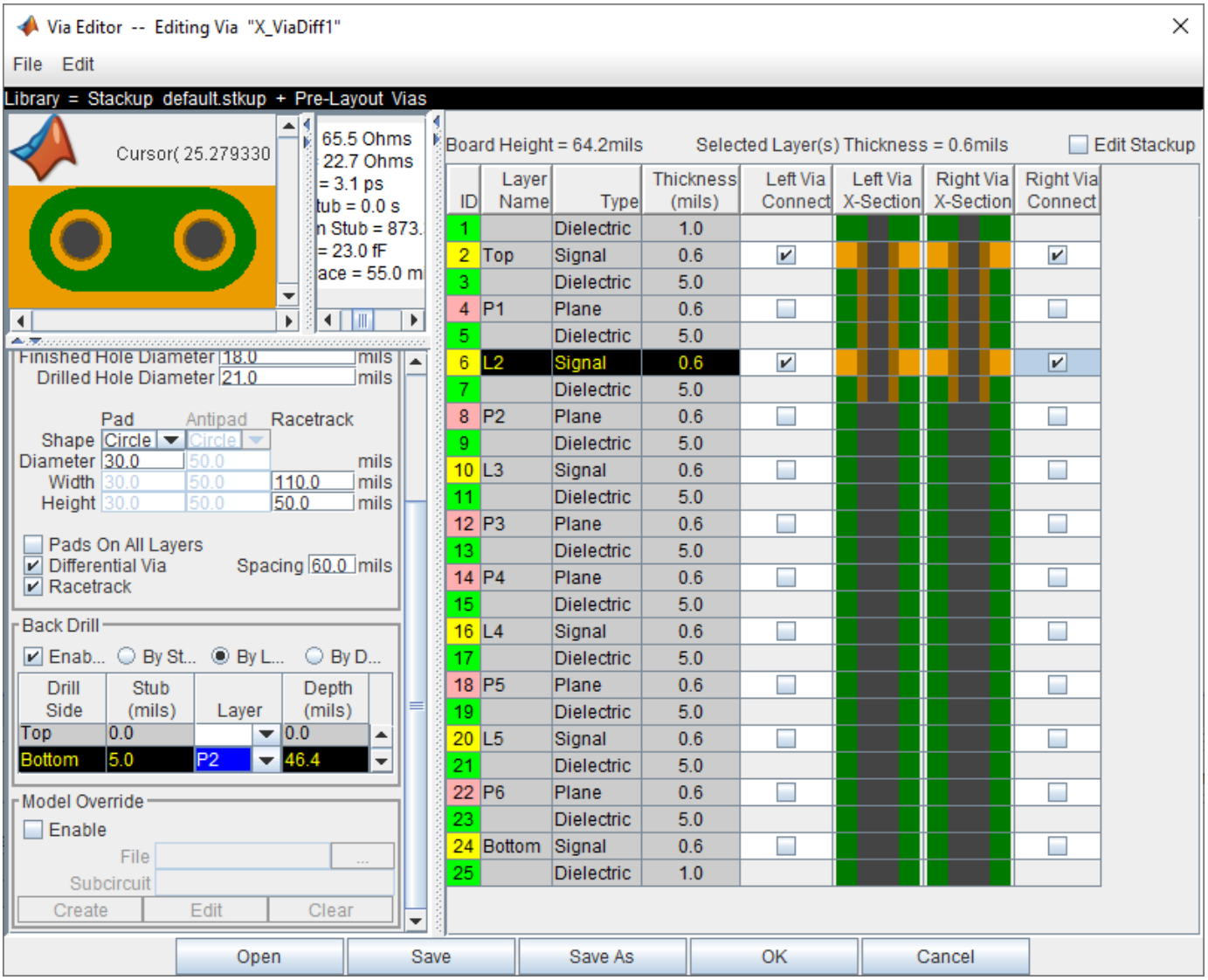
See Also

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 the 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 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 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 During Pre-Layout Simulations

To edit vias during 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 During 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.

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 **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.

You can edit the geometry of a single via, edit the padstack, or override a via model during post-layout.

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.

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

Parallel Link Featured Examples

Configure DDR2 Controller with Two Memory Designators

This example shows how you can configure a DDR2 controller with two custom memory designators.

Create New Project

Open the **Parallel Link Designer** app.

```
parallelLinkDesigner
```

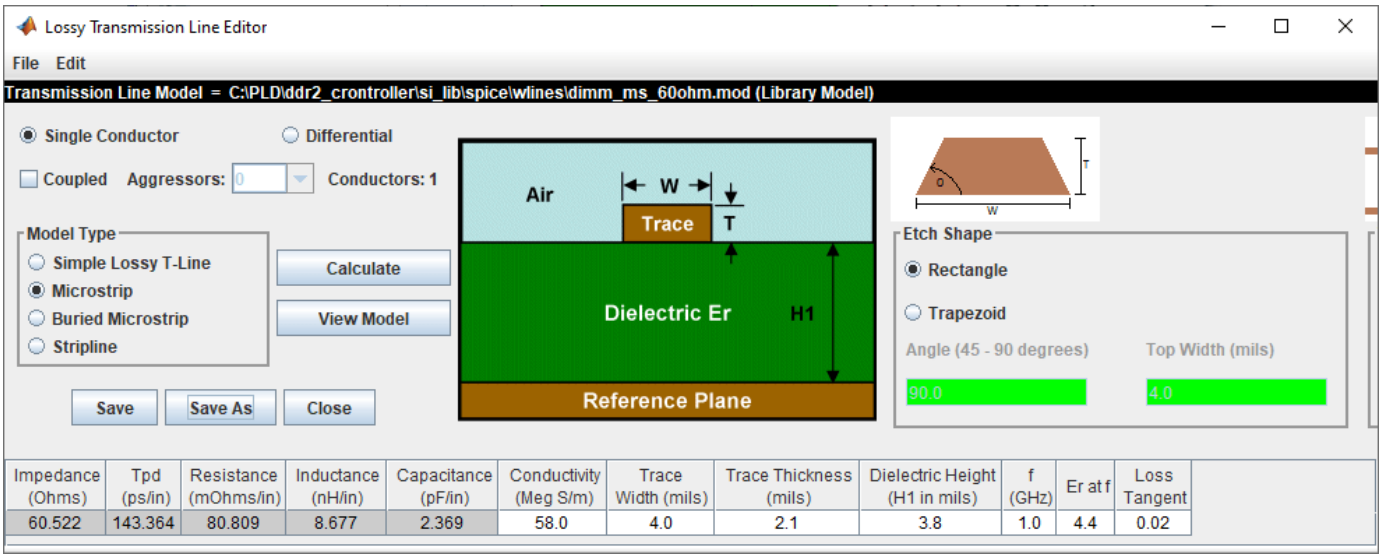
Create a new project by selecting **File > Project > New Project**. In the newly opened dialog box, name the project as `ddr2_controller`, the interface as `ddr2`, and the schematic sheet as `dq`. The **Pre-Layout Analysis** tab shows the blank schematic sheet.

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 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-ended** 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.



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.

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.

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:

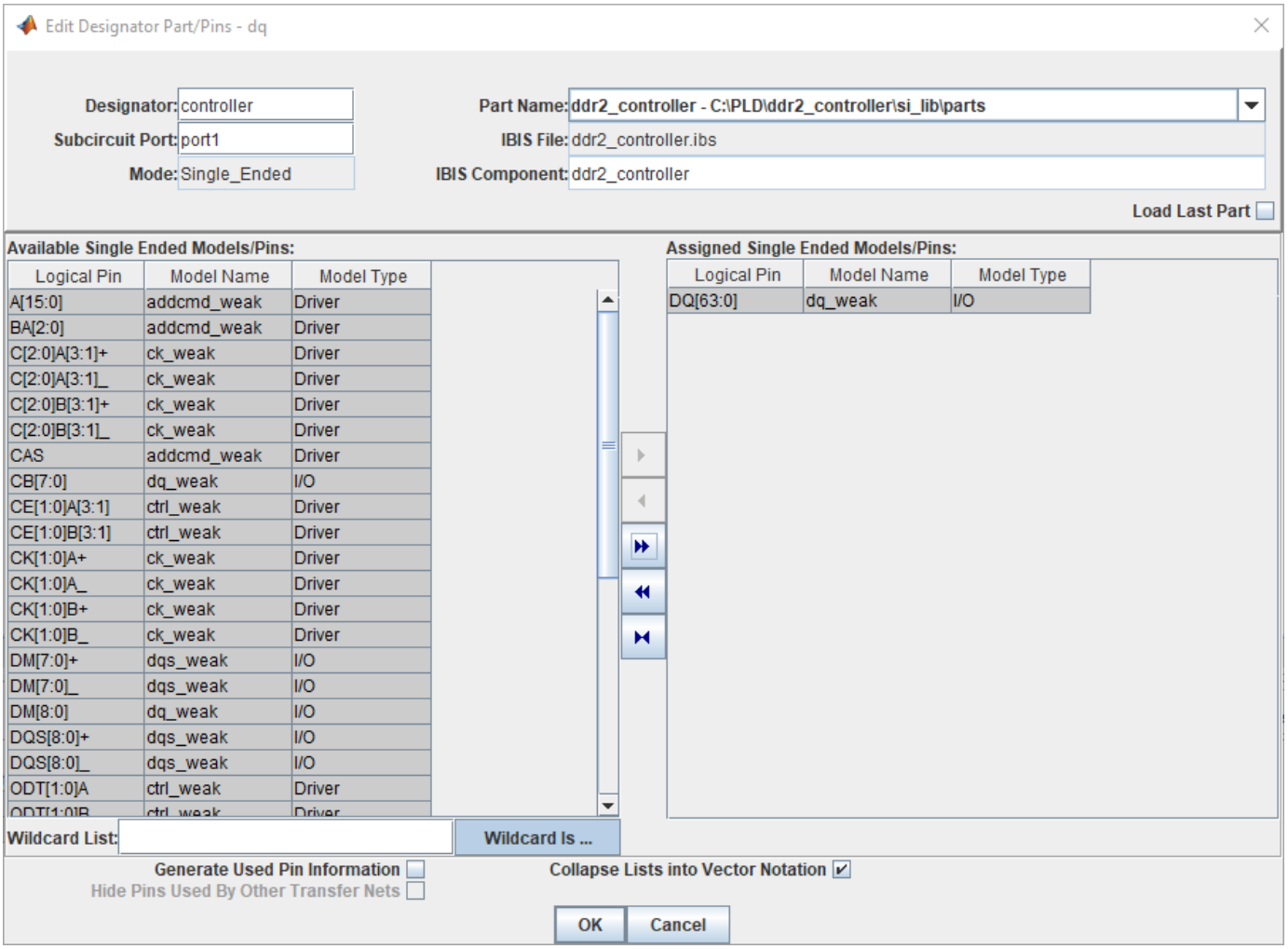
```
ddr2_ck_period = 5.0

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
```

Save and close the clock domain file.

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.

Create Schematic

Use three single-ended designators, one controller designator on the left and otwo memory designators on the right. Select the controller designator on the left and set the **Designator** parameter to `controller` and **Part Name** parameter to `ddr2_controller` from the dropdown menu. You need to all data pins of the controller in 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.

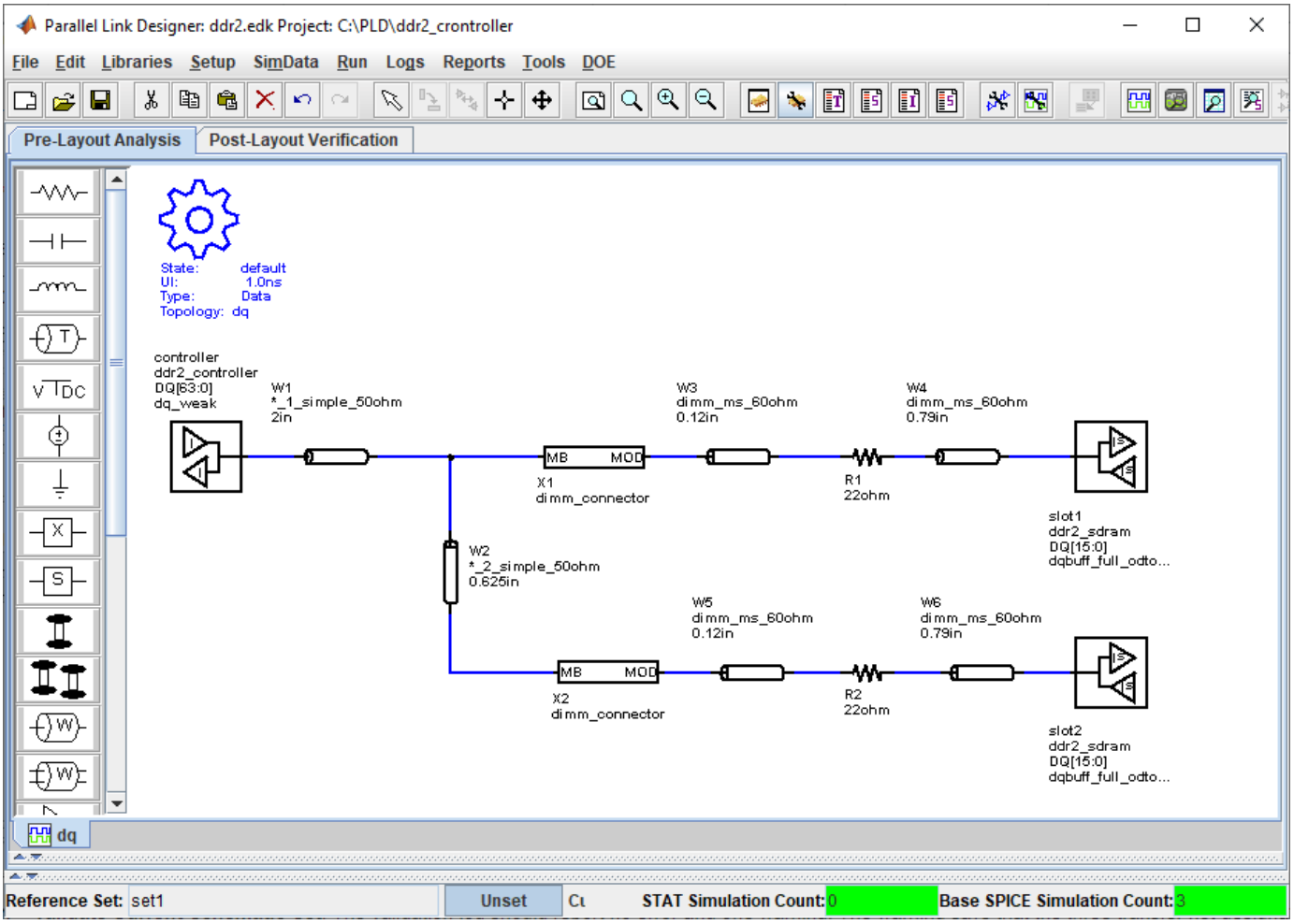


Edit the two memory designators. Name the top one `slot1_dram` and the bottom one `slot2_dram`. For both designators use the `ddr2_sdram` part and include pins `DQ[15:0]`.

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 lines symbols to change their lengths. Change the controller to slot one t-line length to 4.2in and the slot one to slot two t-line length to 0.625 inches.

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

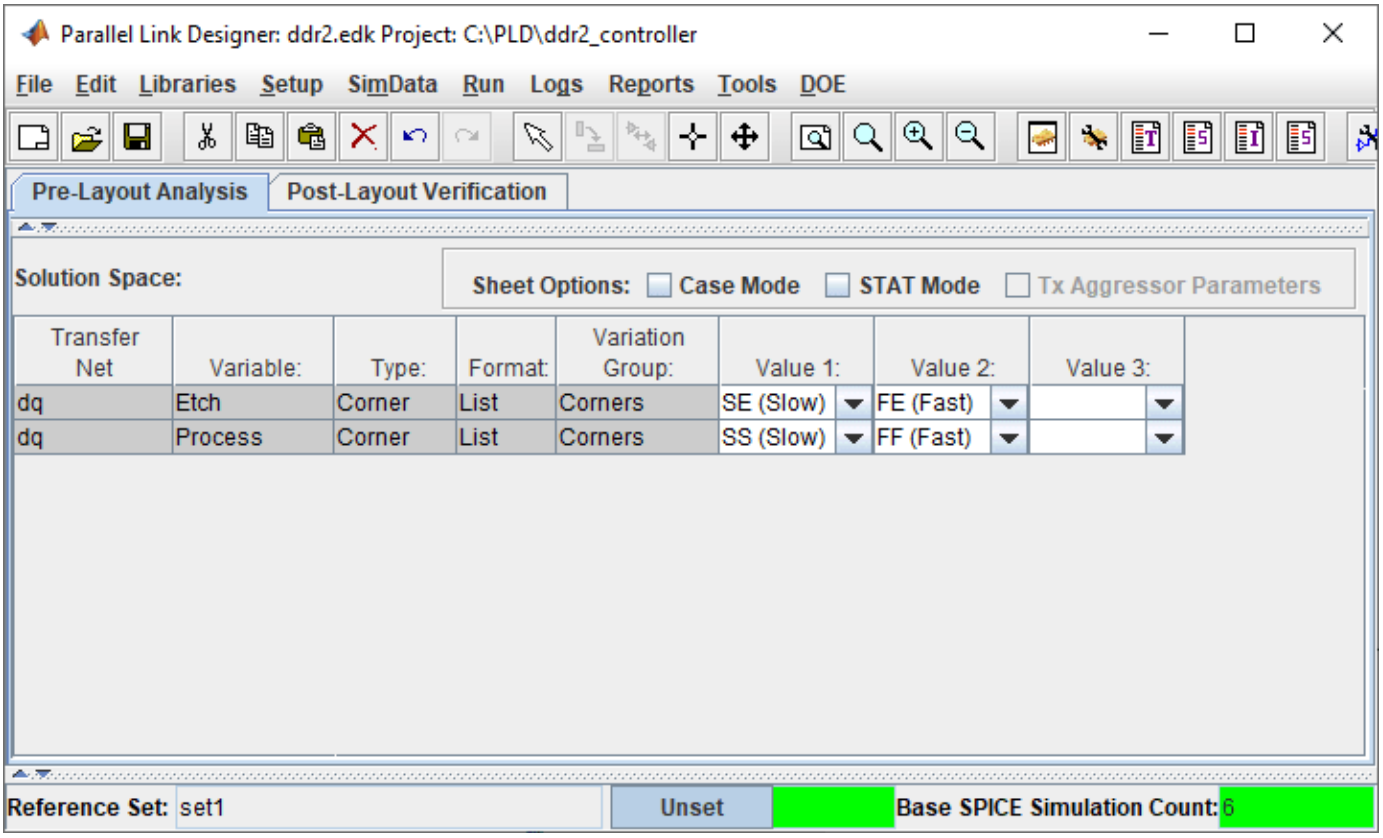
There is a series resistor on each DIMM with a transmission line segment on each side. Add the resistor and a transmission line element. Double click and change the resistor value to 22 ohm. Right click on the transmission line element for one side of the DIMM, select **Select T-Line Mode**l from the menu, go to directory <Project Library>\spice\wlines and select `micro_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 to complete the schematic.



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.



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™ 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: The focus of this example is to illustrate how to setup a post-layout project in **Parallel Link Designer** or **Serial Link Designer**. While there are DDR5 IBIS models in the support project "DDRx_CPU_Dimm_Postlayout," the circuit topologies do not represent a 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 and DIMM are provided as attachments to this example (provided in "Neutral" format). You can acquire these by clicking the button to download the attachments. 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

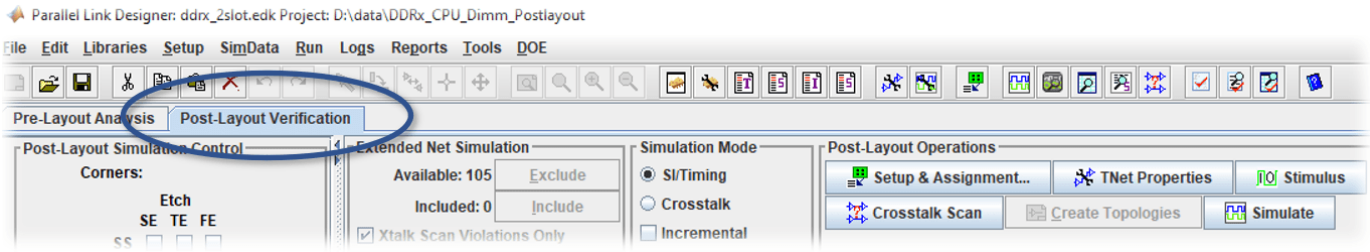
- Configure source-destination bus transactions in Transfers dialog
- Set IBIS models for IO drive strength (ODS) and on-die termination (ODT)

Run Post-Layout Simulation

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

Set up Post-Layout Verification

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

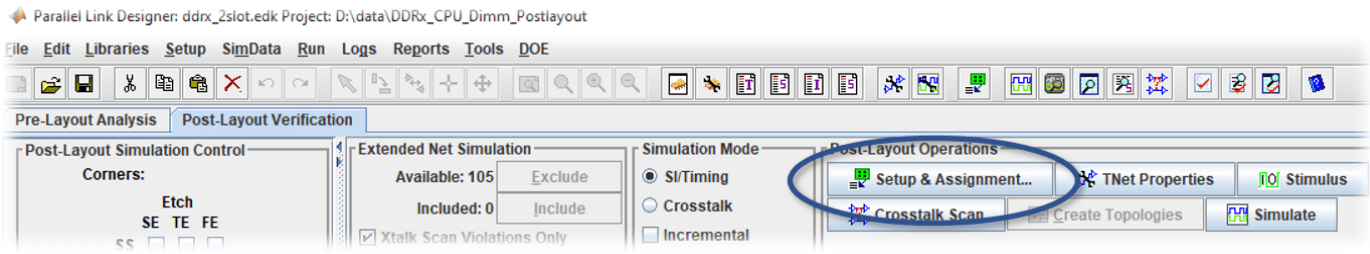


PCB databases are imported into Parallel Link Designer. Then you can configure the PCB stackup information, voltage nets and models assigned and are then placed into a board library which can be used across multiple interfaces. Boards from the library are instantiated and connected to enable Signal Integrigy 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.

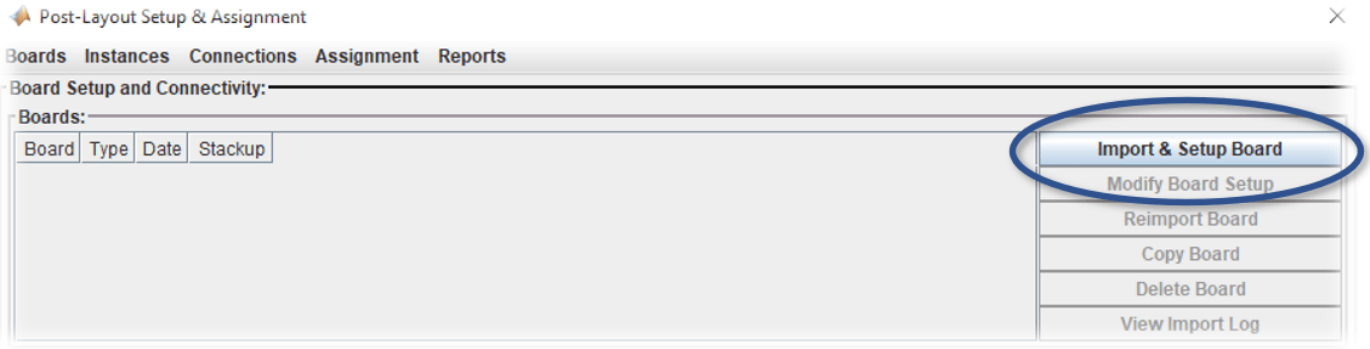
Import and Setup Boards

Import Main Board

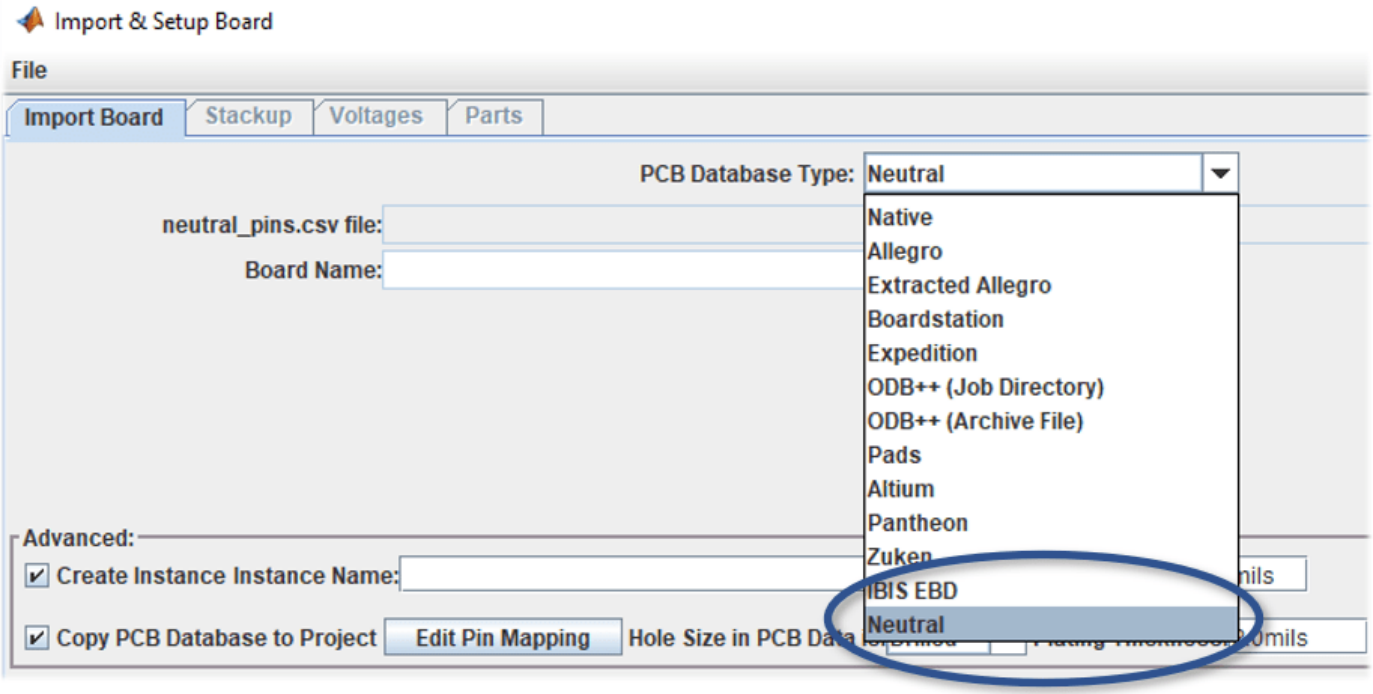
PCB databases are imported from the Setup & Assignment dialog. Click the Setup & Assignment button to launch the dialog.



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

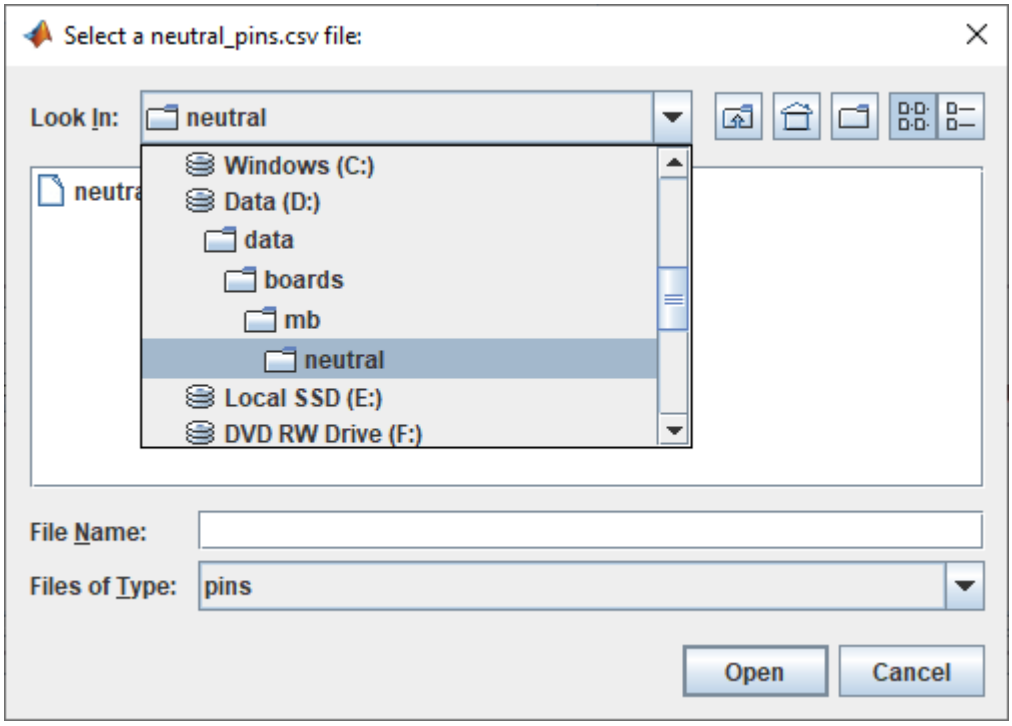


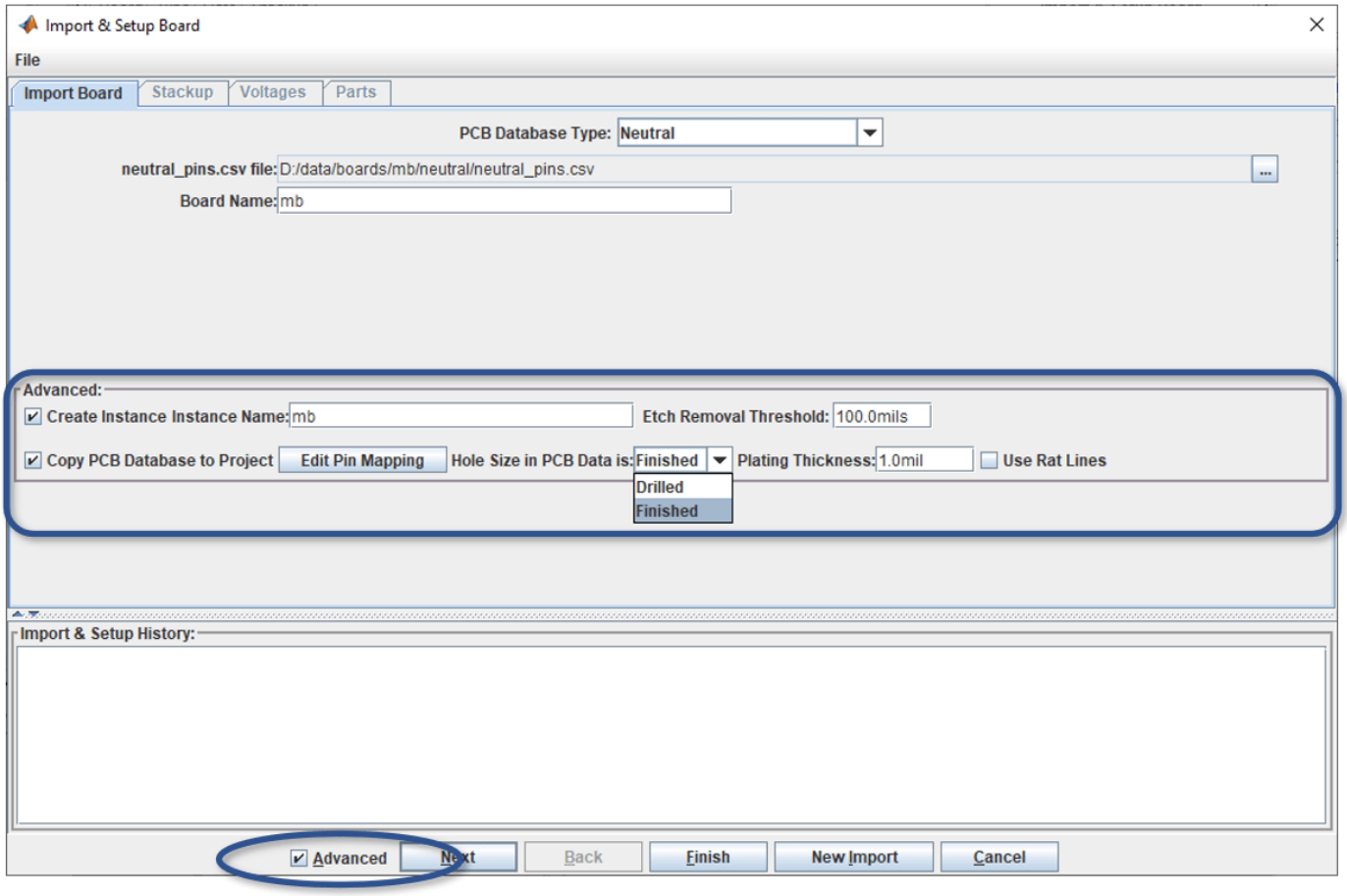
This will launch the Import & Setup Board dialog with the Import Board tab active. You can select from a number of different formats of PCB database from the **PCB Database Type** dropdown menu. The PCB databases of the Main Board and DIMM attached to this example are in "Neutral" format, so select this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



Attached to this example are two PCB databases, which are provided in Neutral format under the folders "boards\mb" for the Main Board PCB and "boards\dimm" for the DIMM.

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
Windows (C:)
Data (D:)
data
boards
mb
neutral
Local SSD (E:)
DVD RW Drive (F:)
neutra
File Name:
Files of Type:
pins
Open
Cancel

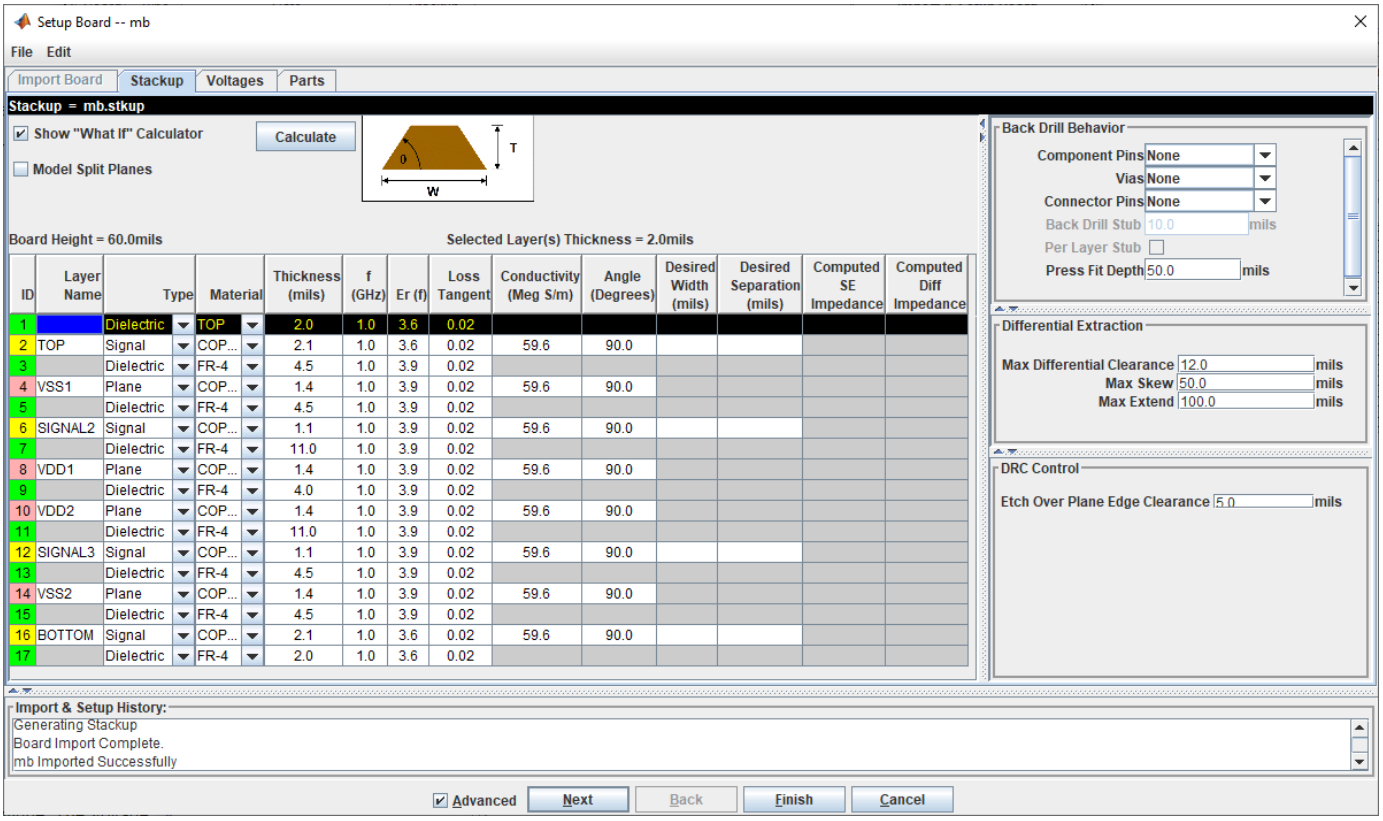


Before clicking the **Next** button to import a board, you may wish to select the "Advanced" option. This displays options for:

- 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.

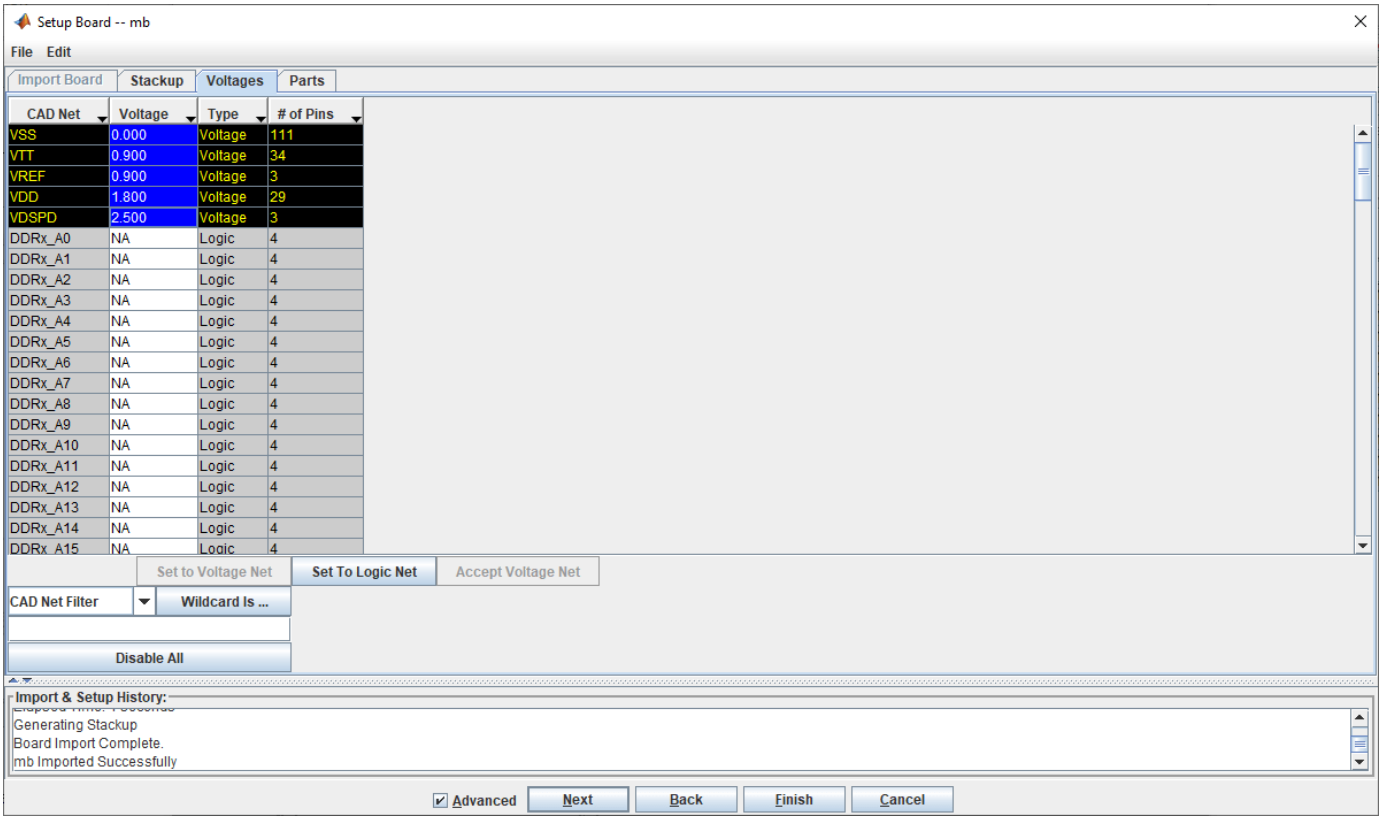
When the PCB database has completed importing, the Stackup tab will appear. This shows the PCB stackup information that Parallel Link Designer read from the PCB database.



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						

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.

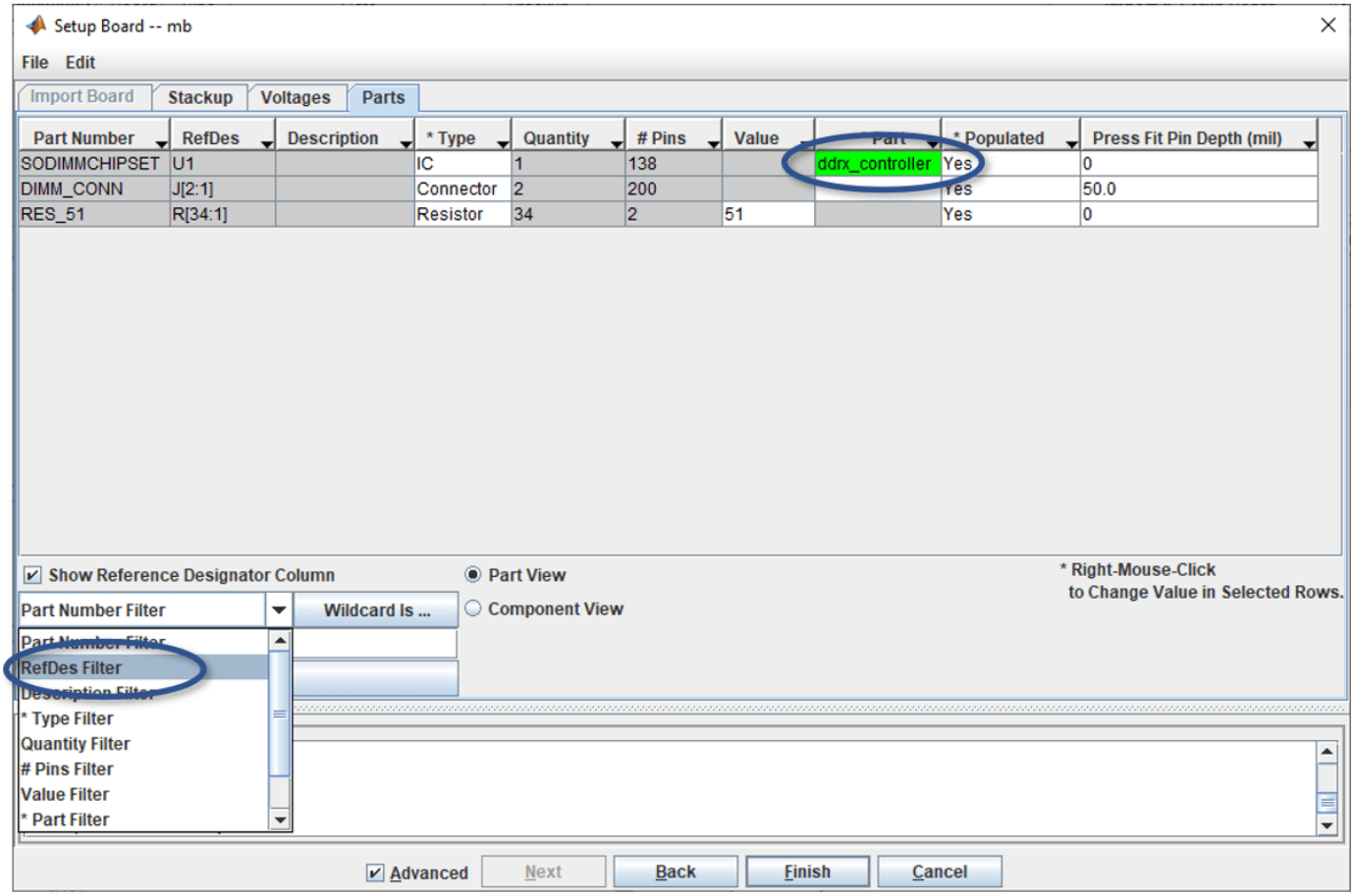
In this case, no changes to the PCB stackup are needed, so click the **Next** button at the bottom of the Import & Setup Board dialog to continue. The Voltage Nets tab will appear (See Figure).



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.



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.

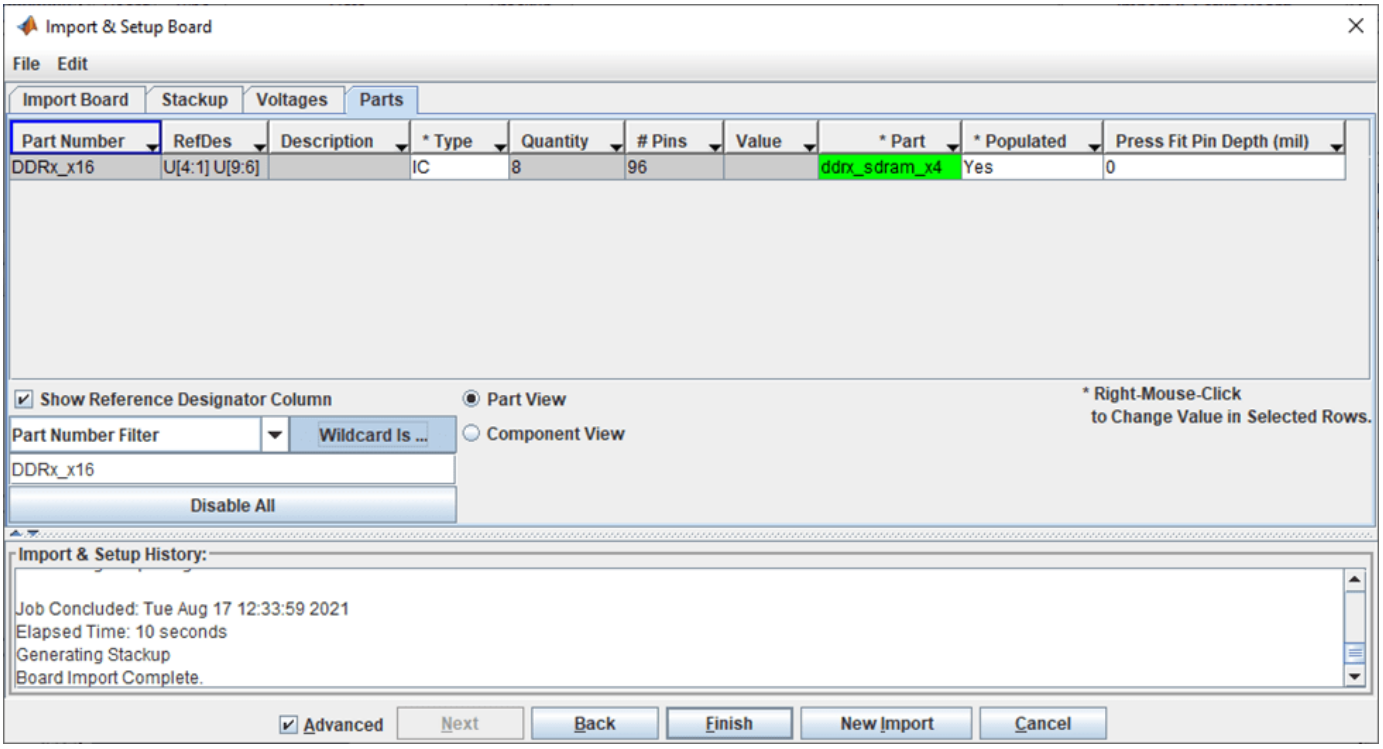
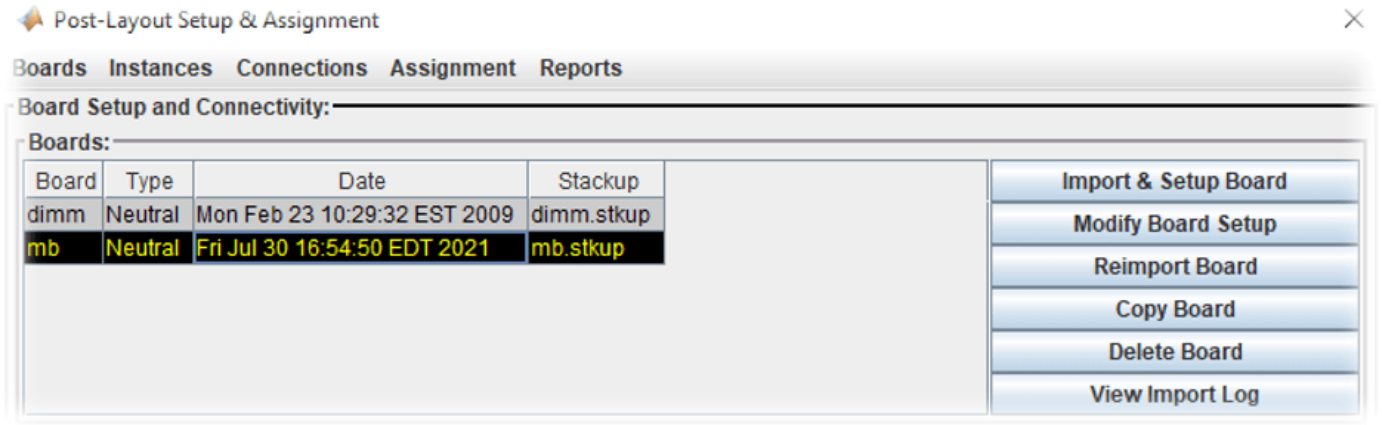


Figure: Set the DIMM Part Number DDRx_x16 to Part ddrx_sdram_x4 from the project library.

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



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.

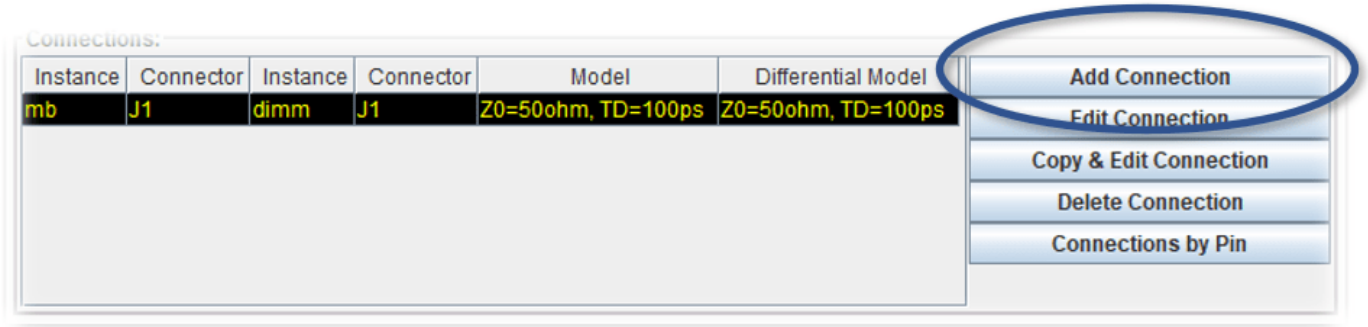
Create New DIMM Instance

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.

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.



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.

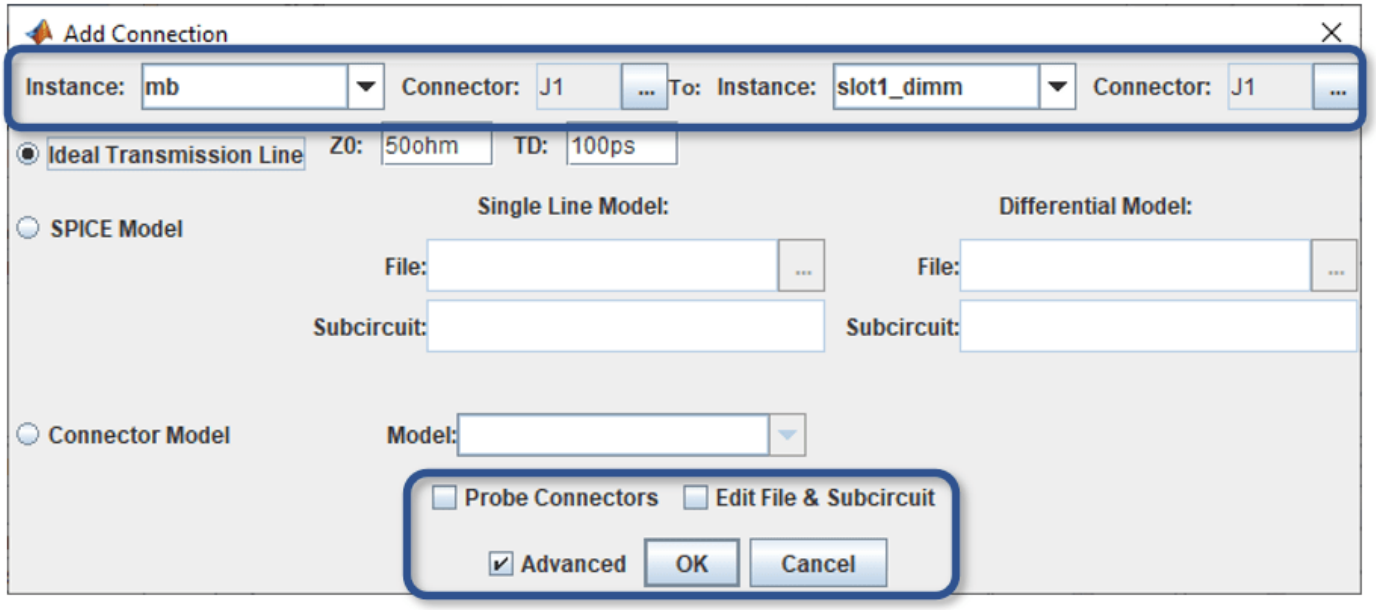
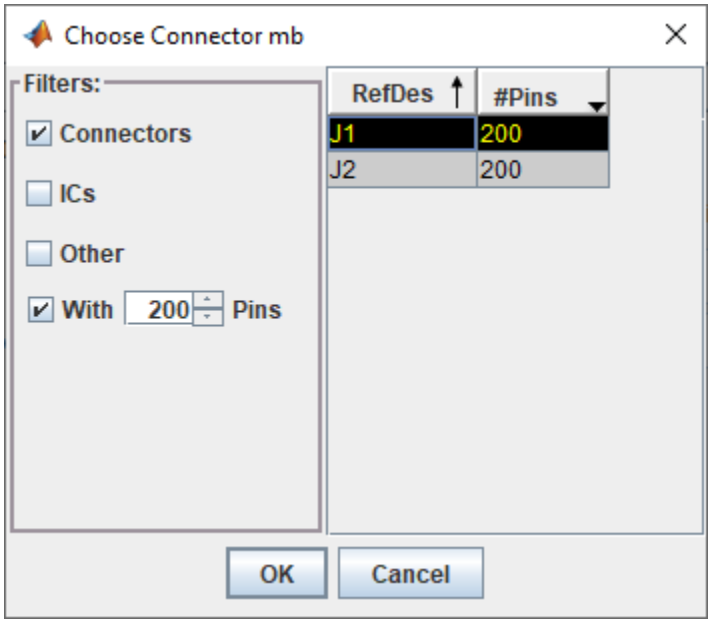


Figure : Add Connection Dialog: enable the Advanced checkbox to view options for Probe Connectors and Edit File & Subcircuit

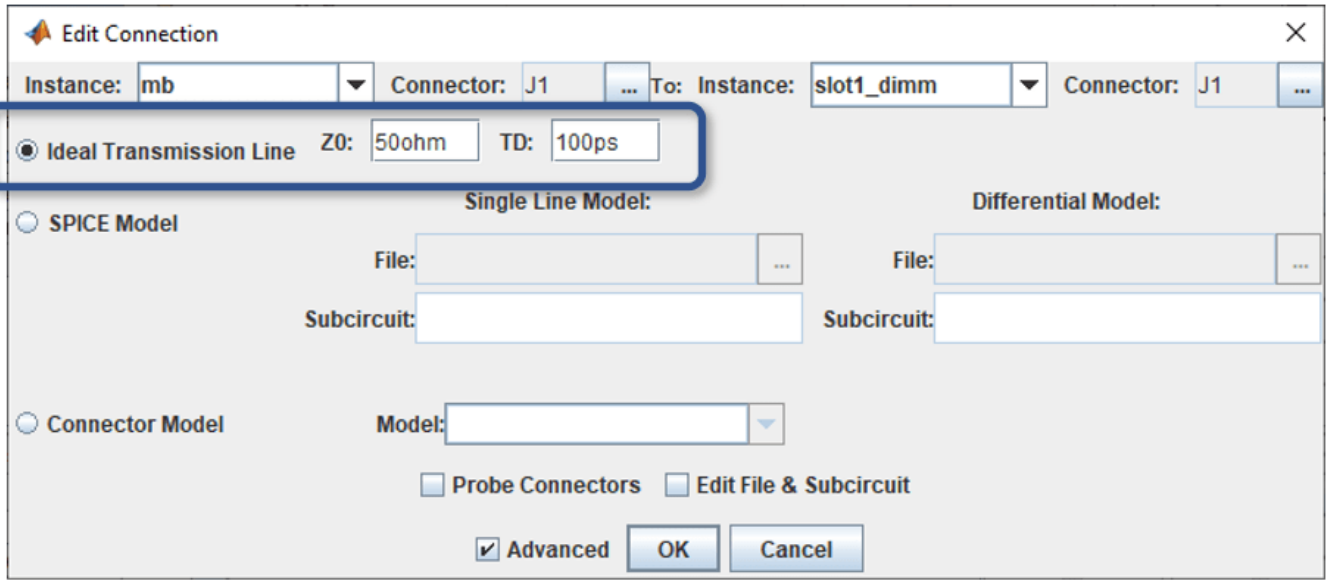
Click the **Browse** button to select a connector on the Instance mb. The Choose Connector dialog lets you select the connector on the Instance mb that will be connected to the DIMM. Select J1.



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.



Click **OK** to complete the connection.

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

1. Click Add Connection.

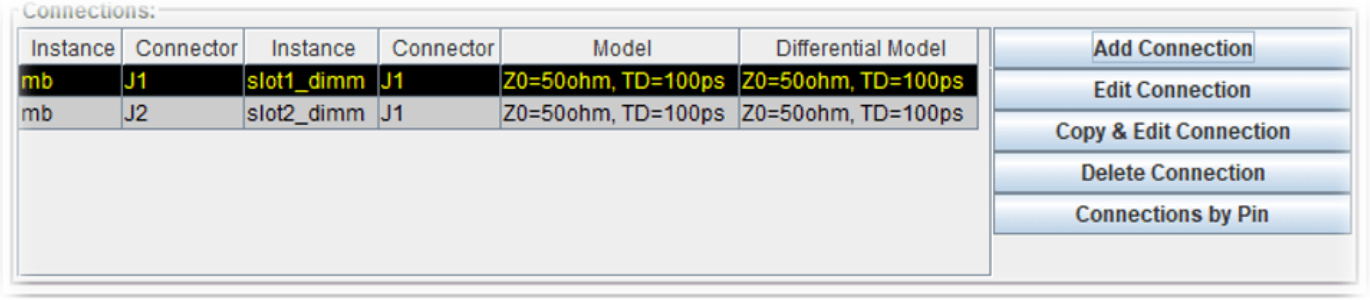
2. Select Instance mb and connector J2 on the left.

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.



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.

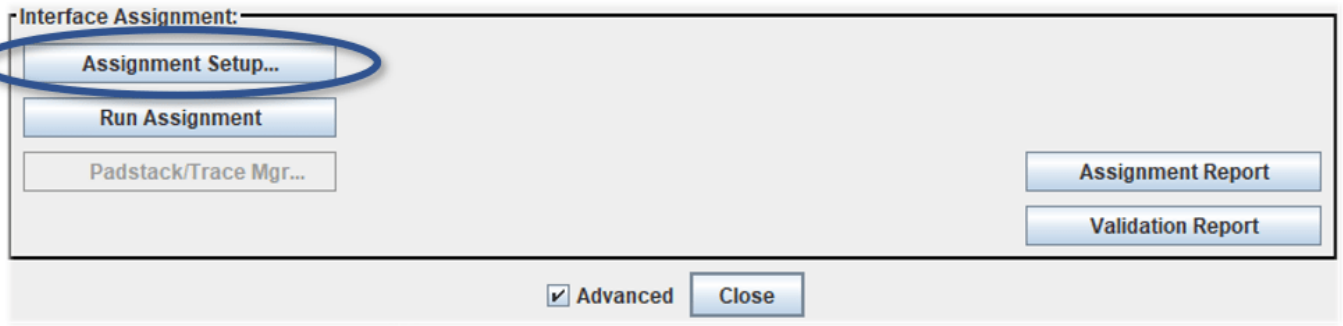


Figure: Assignment Setup Button

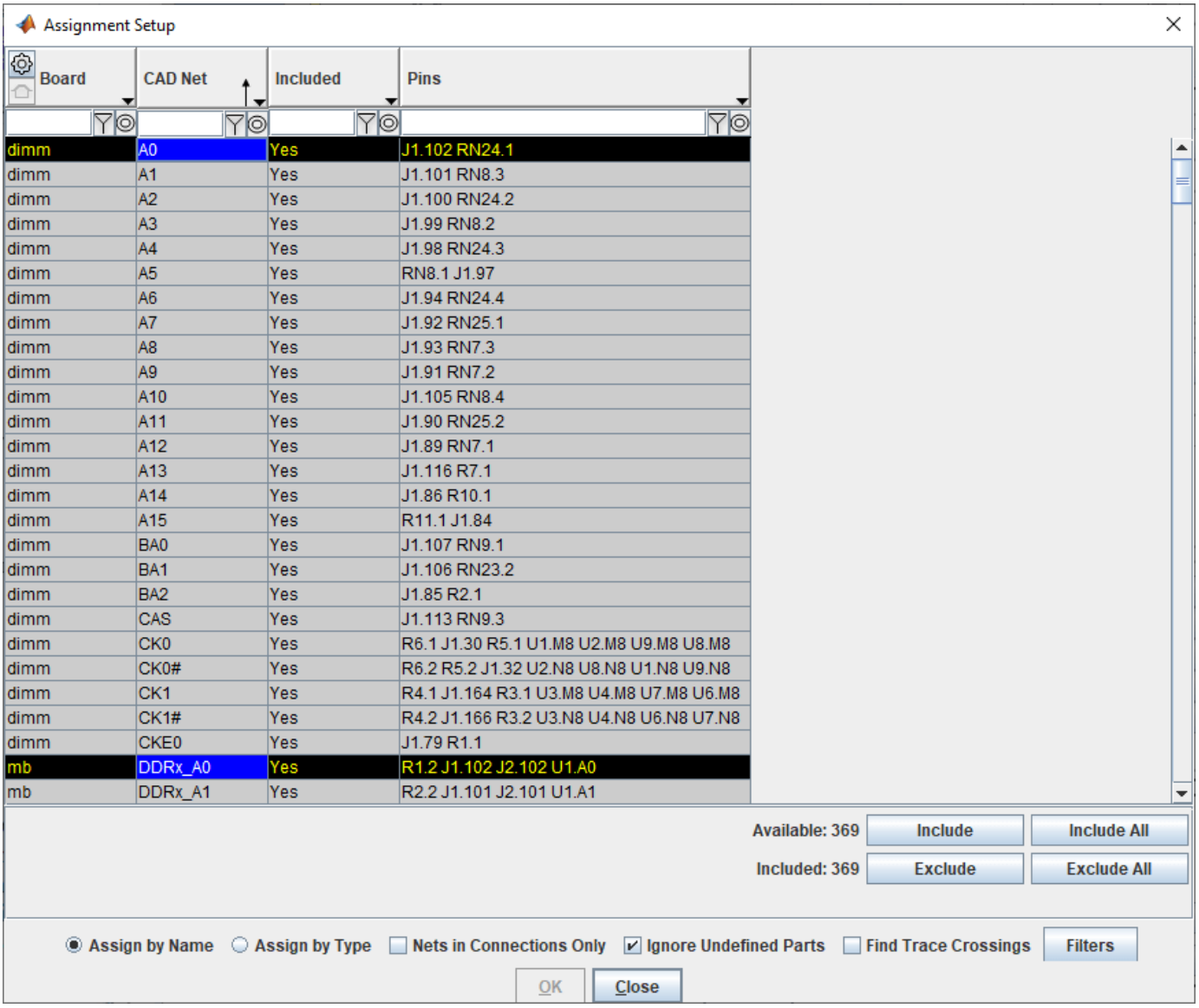


Figure: Select Nets to Include or Exclude for Assignment

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.

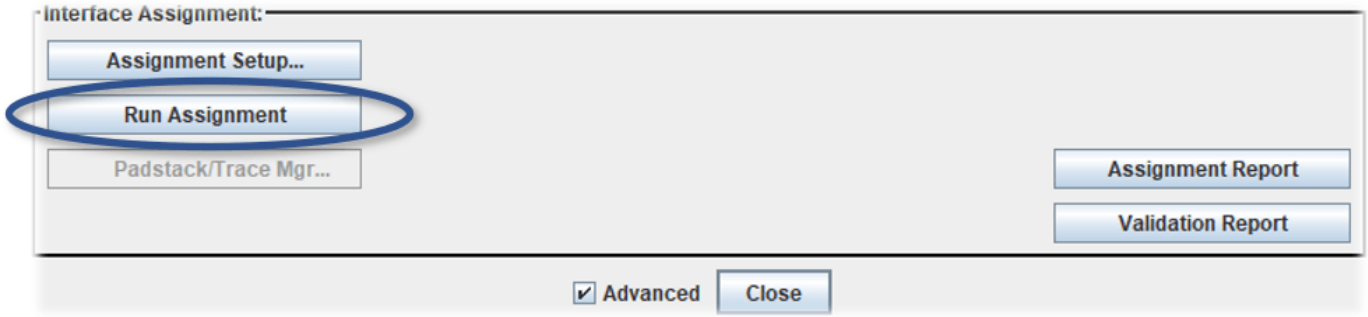
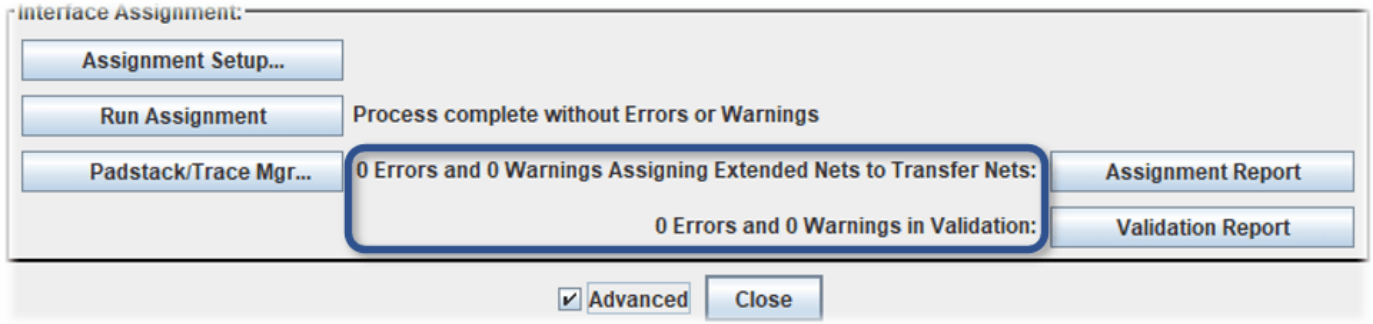


Figure: Run Assignment Button

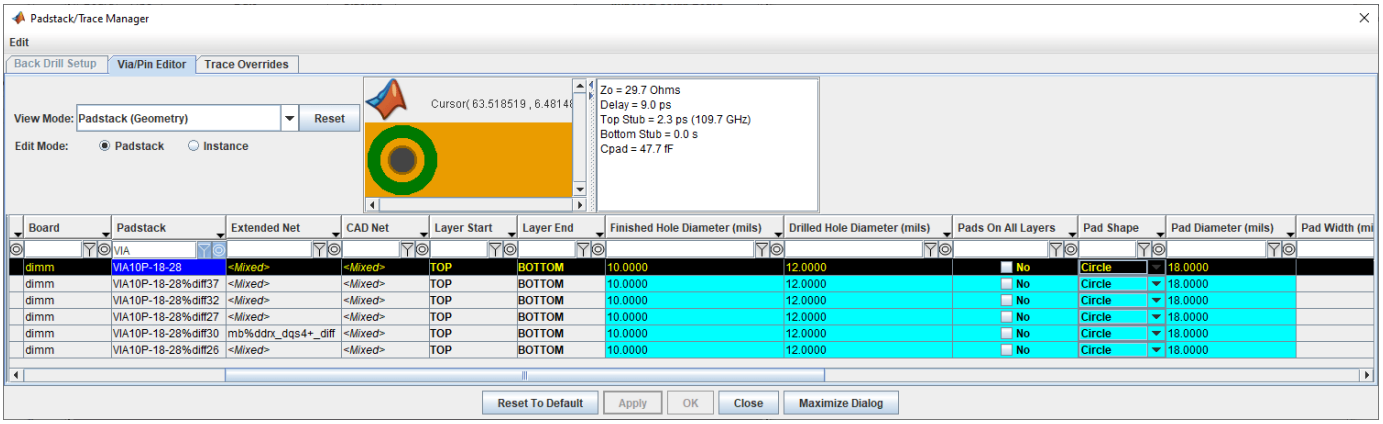
When the Assignment process is complete the Assignment and Validation Reports are created.



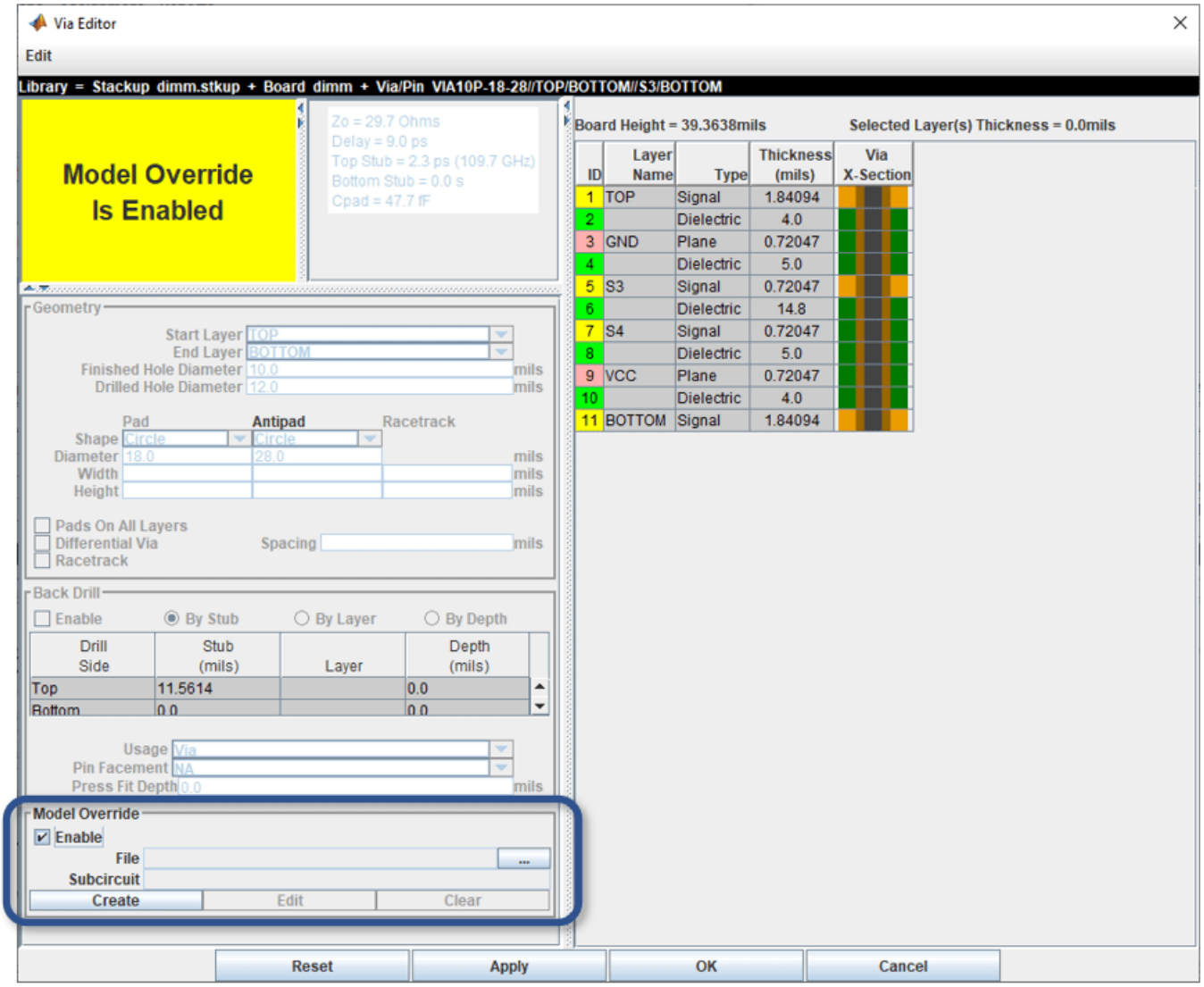
It is always a good idea to check the status after running Assignment. It can alert you to problems that may be difficult to debug without seeing the error messages. For example, if there are warnings listed they may represent unused parts. Even if you see no errors or warnings it is wise to review the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



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).



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.

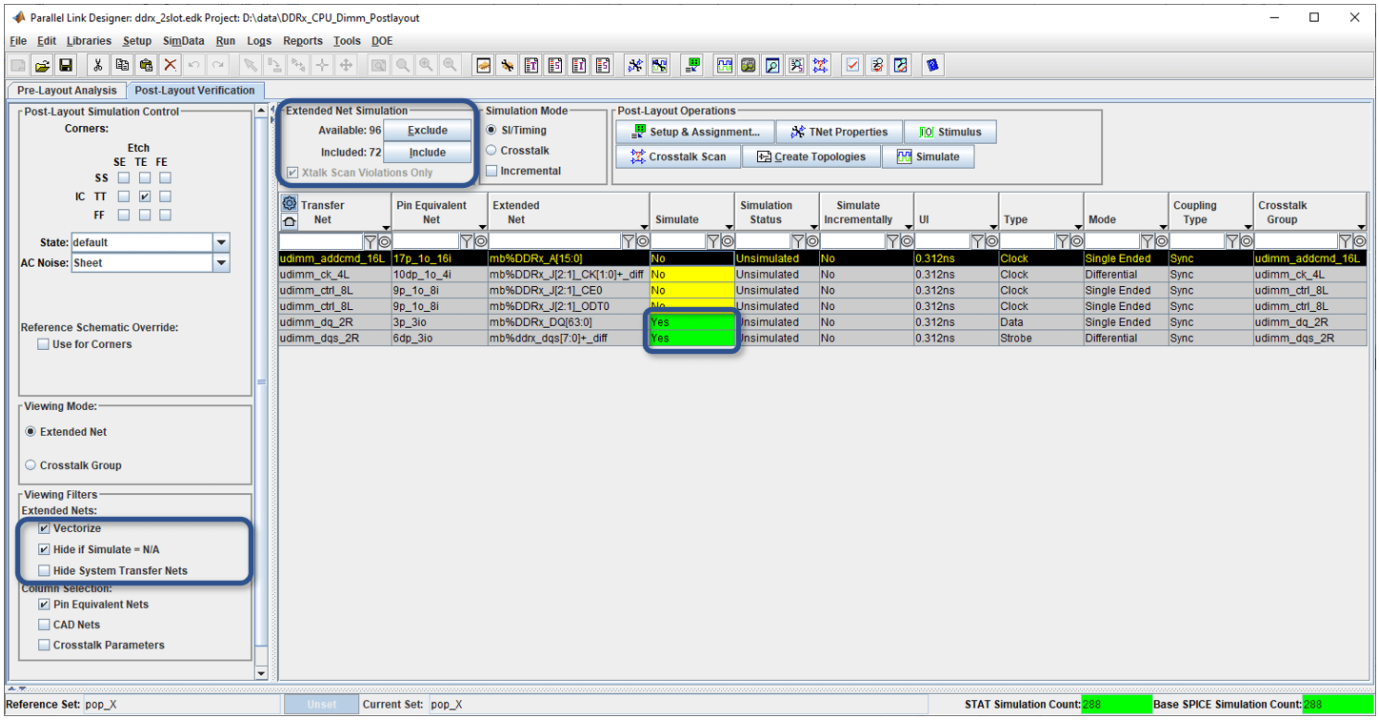
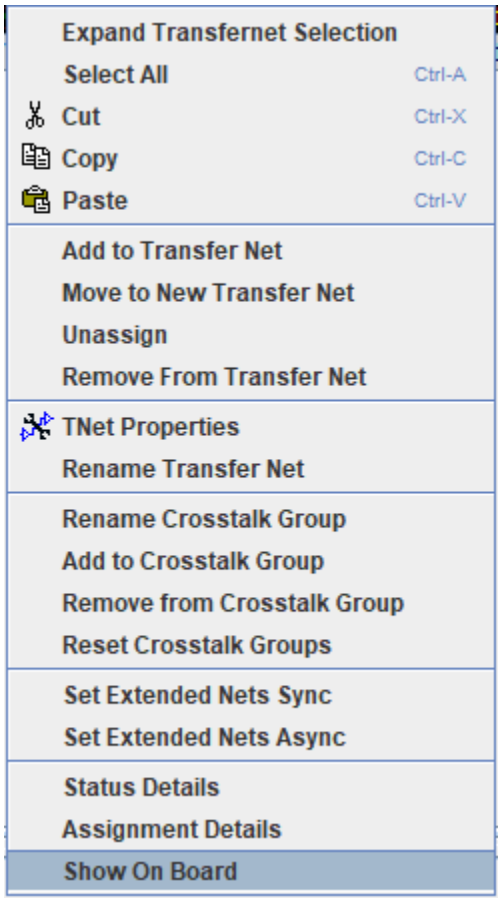


Figure: Post-Layout Tab After Assignment. You can use the Viewing Filters to Vectorize nets by group and to Hide if Simulate = N/A

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.



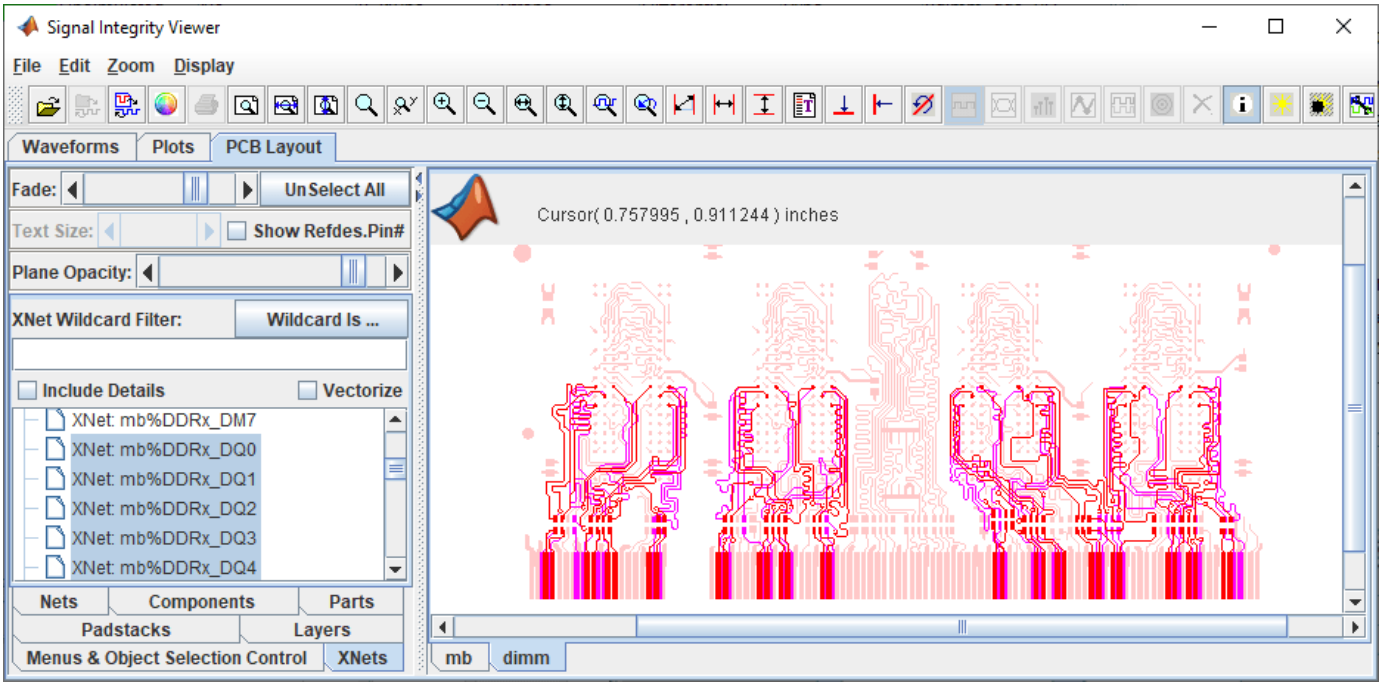


Figure: Signal Integrity Viewer opens with all selected Transfer Nets highlighted.

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).

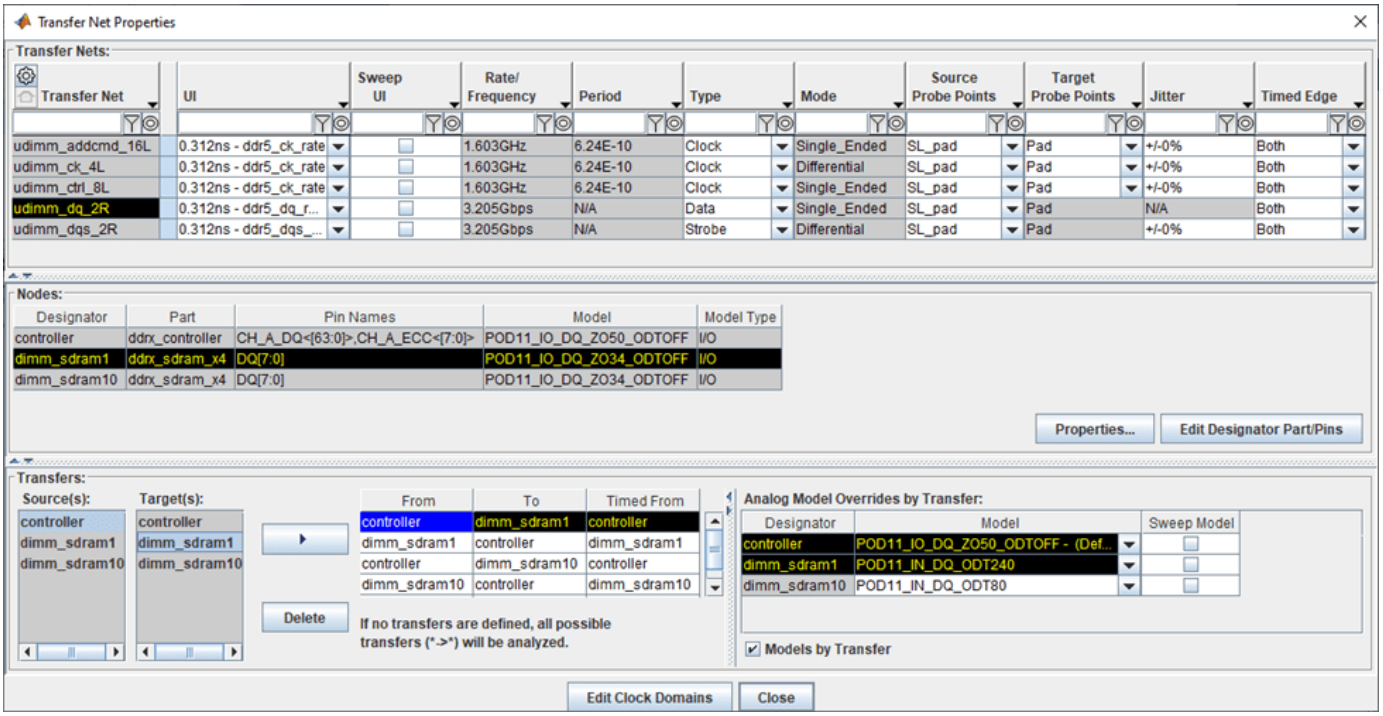


Figure: Transfer Net Properties Tab after Assignment. You can configure Transfers and Analog Model Overrides by Transfer (e.g. configure IBIS model ODS and ODT)

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.

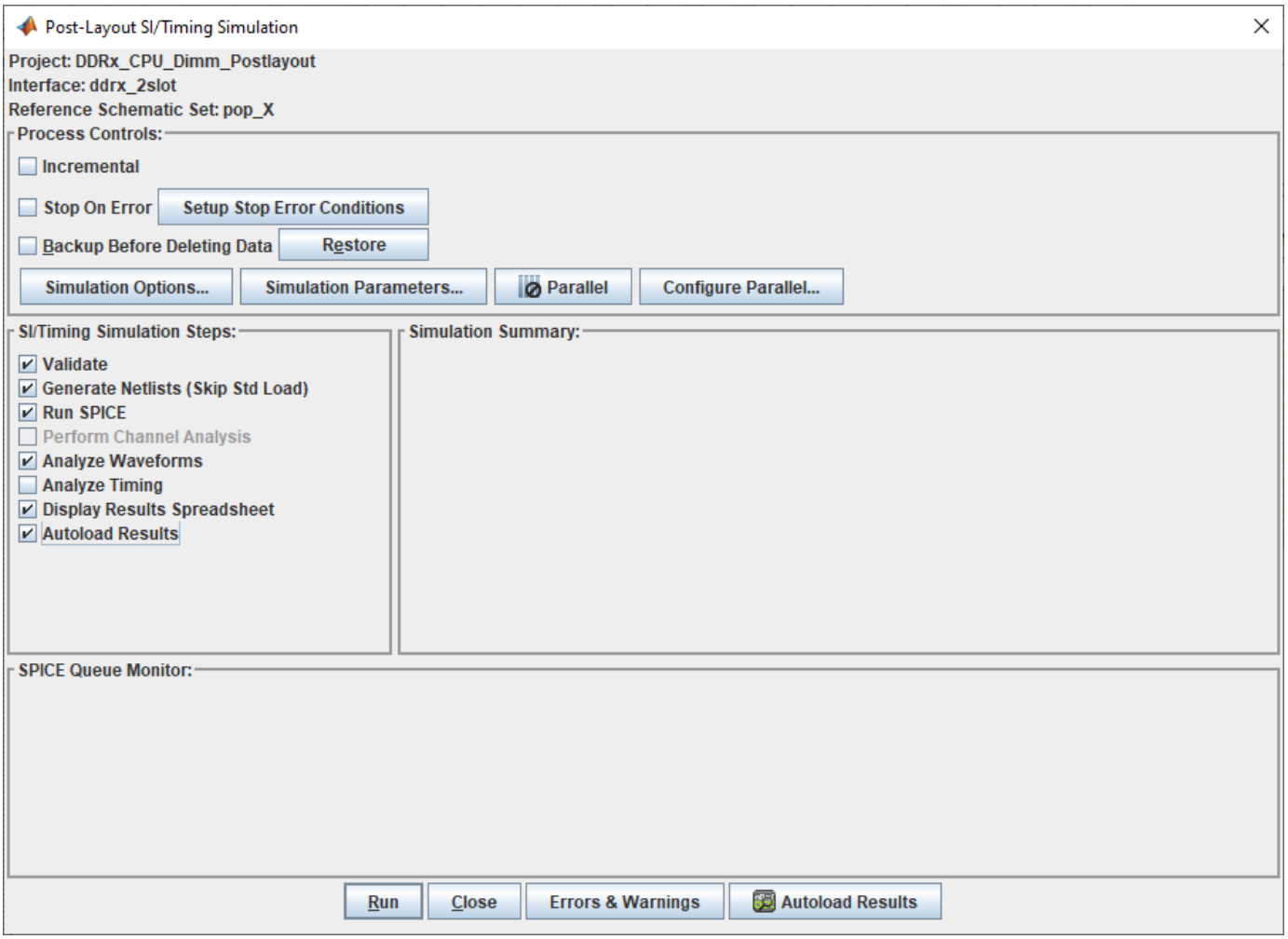
Before selecting the nets to simulate, hide the nets that are from the serial interface or are used for multi-rank DIMMs by checking the Hide if Simulate=N/A checkbox (see Figure).

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).



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

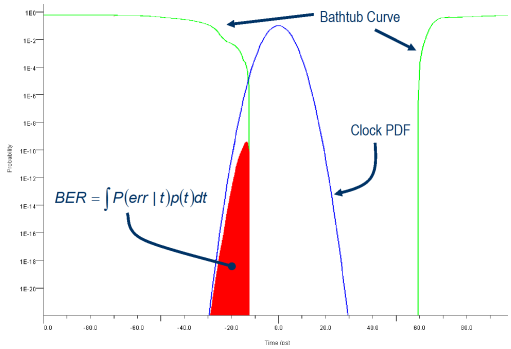
- “Results of Pre-Layout Analysis in Parallel Link” on page 7-8
- “DDR5 Implementation Kit” on page 11-71
- “GDDR5 x32 Implementation Kit” on page 11-73
- “Low-Power DDR5 Architectural Kit” on page 11-80

Jitter and Noise

Model Jitter and Noise While Designing Parallel Link

You can model three major sources of jitter using the **Parallel Link Designer**: TX clock jitter, RX clock jitter, and RX clock recovery jitter. You can also add RX noise.

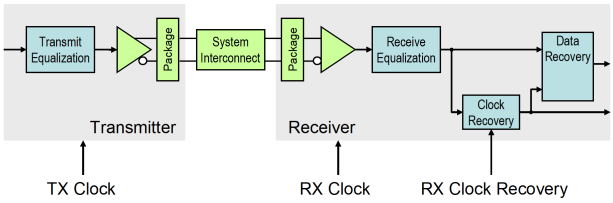
Jitter and noise affect the bit error rate (BER) of a serial channel. Some sources of jitter affect the data bathtub curve and some affect the clock PDF. The data bathtub and clock PDF are used in the BER calculation, so changing either one will change BER.



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 the setting of the **Clock Mode** parameter.

- **Clock Mode** Normal — RX clock recovery jitter affects the clock PDF.
- **Clock Mode** Clocked — RX clock recovery jitter affects the data eye and bathtub.
- **Clock Mode** 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**.



TX Clock Jitter

TX clock jitter is modeled using five parameters. The parameters modify the Tx stimulus (Time Domain analysis) or are added in post processing (Statistical Analysis). View and modify these parameters in the TX Jitter dialog, accessible through the Designator Element Properties dialog.

TX jitter will always change 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 in this way: $Time(n) = n \times UI + \texttt{Tx\_Rj} \times \texttt{randn}$ $Time(n)$ 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 only applicable 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 in this way: $Time(n) = n \times UI + \texttt{Tx\_Dj} \times \texttt{rand}$ `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 in this way: $Time(n) = n \times UI + \texttt{Tx\_Sj} \times \sin(n \times UI \times 2\pi \times \texttt{Tx\_Sj\_Frequency})$ 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), 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 in this way: $Time(n) = n \times UI + \texttt{Tx\_DCD} \times (-1)^n$

Note If a TX jitter parameter is set in the AMI file, the field in the TX Jitter dialog will be non-editable. To change the jitter in that case the AMI file must be edited.

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 post-processing the results from the model and are not passed to the model. These parameters can be viewed and modified in the RX Jitter dialog.

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* + `Rx_Rj × 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* + `Rx_Dj × 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 an RX Jitter parameter is set in the AMI file the field in the RX Jitter dialog will be non-editable. To change the jitter in that case the AMI file must be edited.

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 post-processing the results from the model. In statistical analysis these parameters are always used. In time domain analysis these parameters are used 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. These parameters must be added to the AMI file using a text editor as shown in “10.4.1.3 AMI File Usage” on page 361.

In the definition of these jitter parameters, *ideal_time* is 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* + `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* + `Rx_Clock_Recovery_Rj` × `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* + `Rx_Clock_Recovery_Sj` × sin(π/2 × `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), 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* + `Rx_Clock_Recovery_DCD` × $(-1)^n$

RX Noise

RX noise parameters modify the statistics associated with the data input to the receiver's sampling latch. This data is used by **Parallel Link Designer** when post-processing 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. Note that it is seldom if ever accurate to model crosstalk or power supply noise as a Gaussian distributed process. Note also that in order to supply an accurate value for this parameter, it may 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, either can be used interchangeably.
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 as shown in “10.4 AMI Control of Jitter/Noise Parameters”

Note If an RX Noise parameter is set in the AMI file the field in the RX Jitter dialog will be non-editable. To change the jitter in that case the AMI file must be edited.

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 will have the value shown but have editing disabled to indicate that the value is controlled by the AMI file

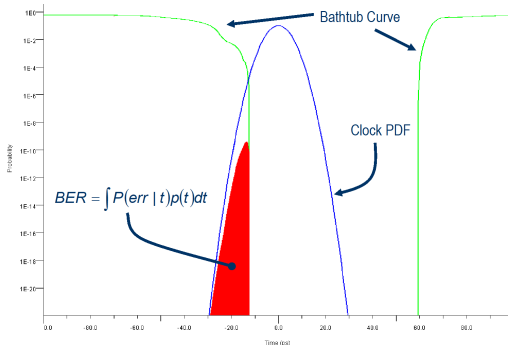
Parameter Type	Sample Entry and Description
Corner	`(Tx_Rj (Usage Info)(Corner 0.005 0.006 0.004)(Type UI) (Description "TX Random Jitter in UI"))` The value used in the analysis is based on the IC process corner selected in the GUI (see “5.4.1 Corners” on page 123 for information on setting IC corners. The Transfer Net Properties dialog will have <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 will have <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.

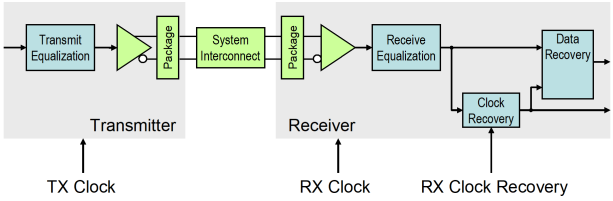
Jitter and noise affect the bit error rate (BER) of a serial channel. Some sources of jitter affect the data bathtub curve and some affect the clock PDF. The data bathtub and clock PDF are used in the BER calculation, so changing either one will change BER.



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 the setting of the **Clock Mode** parameter.

- **Clock Mode** Normal — RX clock recovery jitter affects the clock PDF.
- **Clock Mode** Clocked — RX clock recovery jitter affects the data eye and bathtub.
- **Clock Mode** 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**.



TX Clock Jitter

TX clock jitter is modeled using five parameters. The parameters modify the Tx stimulus (Time Domain analysis) or are added in post processing (Statistical Analysis). View and modify these parameters in the TX Jitter dialog, accessible through the Designator Element Properties dialog.

TX jitter will always change 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 in this way: $Time(n) = n \times UI + \texttt{Tx\_Rj} \times \texttt{randn}$ $Time(n)$ 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 only applicable 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 in this way: $Time(n) = n \times UI + \texttt{Tx\_Dj} \times \texttt{rand}$ `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 in this way: $Time(n) = n \times UI + \texttt{Tx\_Sj} \times \sin(n \times UI \times 2\pi \times \texttt{Tx\_Sj\_Frequency})$ 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), 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 in this way: $Time(n) = n \times UI + \texttt{Tx\_DCD} \times (-1)^n$

Note If a TX jitter parameter is set in the AMI file, the field in the TX Jitter dialog will be non-editable. To change the jitter in that case the AMI file must be edited.

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 post-processing the results from the model and are not passed to the model. These parameters can be viewed and modified in the RX Jitter dialog.

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* + `Rx_Rj` × `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* + `Rx_Dj` × `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 an RX Jitter parameter is set in the AMI file the field in the RX Jitter dialog will be non-editable. To change the jitter in that case the AMI file must be edited.

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 post-processing the results from the model. In statistical analysis these parameters are always used. In time domain analysis these parameters are used 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. These parameters must be added to the AMI file using a text editor as shown in “10.4.1.3 AMI File Usage” on page 361.

In the definition of these jitter parameters, *ideal_time* is 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* + `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* + `Rx_Clock_Recovery_Rj` × `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* + `Rx_Clock_Recovery_Sj` × sin(π/2 × `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), 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* + `Rx_Clock_Recovery_DCD` × $(-1)^n$

RX Noise

RX noise parameters modify the statistics associated with the data input to the receiver's sampling latch. This data is used by **Serial Link Designer** when post-processing 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. Note that it is seldom if ever accurate to model crosstalk or power supply noise as a Gaussian distributed process. Note also that in order to supply an accurate value for this parameter, it may 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, either can be used interchangeably.
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 as shown in “10.4 AMI Control of Jitter/Noise Parameters”

Note If an RX Noise parameter is set in the AMI file the field in the RX Jitter dialog will be non-editable. To change the jitter in that case the AMI file must be edited.

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 will have the value shown but have editing 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 (see “5.4.1 Corners” on page 123 for information on setting IC corners. The Transfer Net Properties dialog will have <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 will have <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.

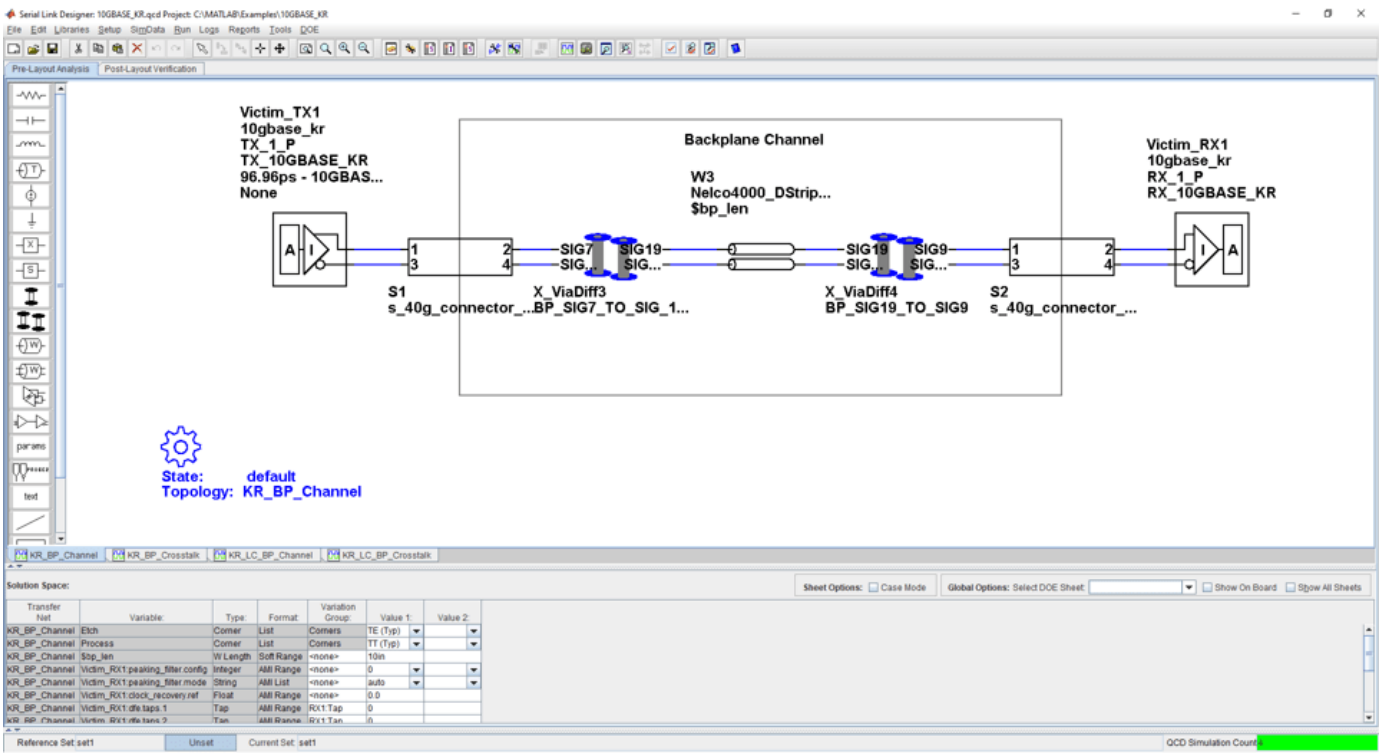
This kit is designed for analysis of a backplane design with two mated connectors as given in the Annex 69B section of the 10GBASE-KR specification. The kit also includes sheets containing the backplane and connectors with two plug-in line cards attached. IBIS-AMI TX and RX models are provided with representative 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");
```



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.

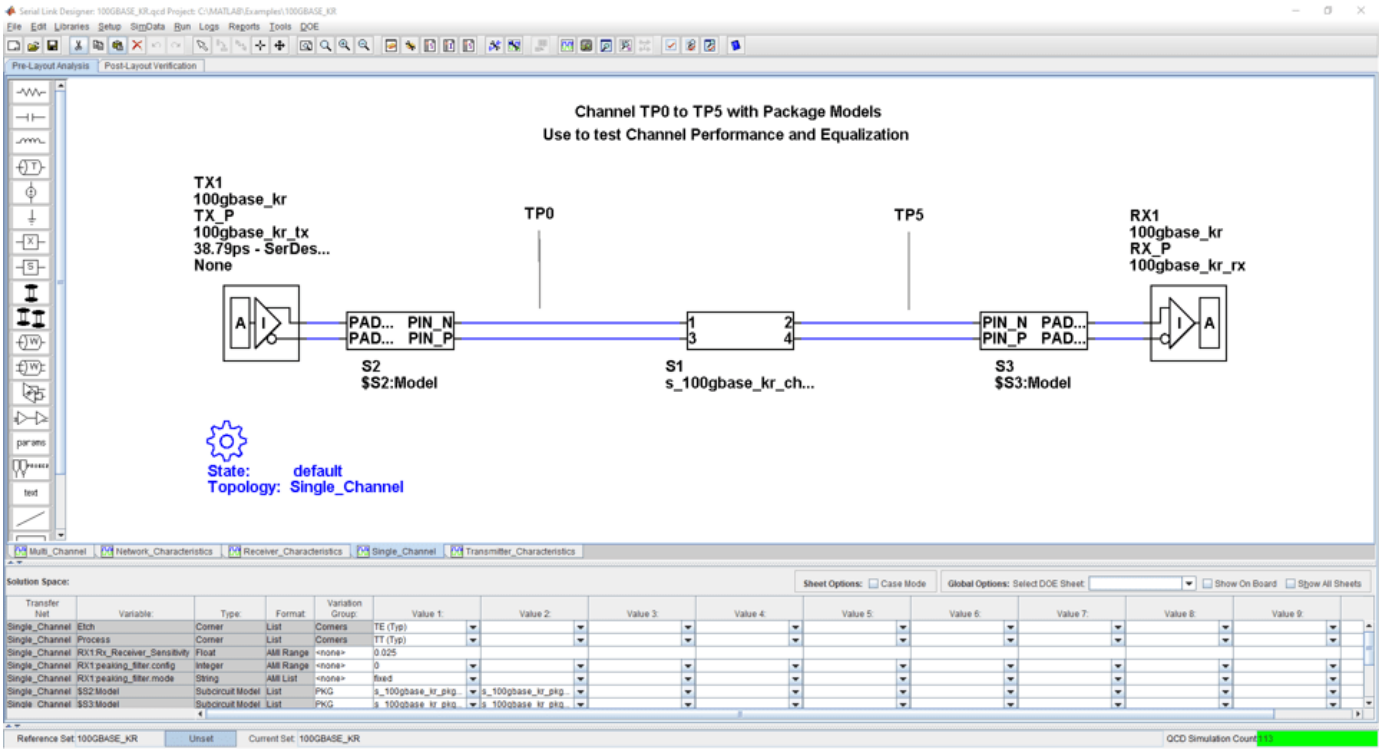
This kit is designed for analysis of a 25.78 Gb/s channel based on 802.3bj compliance. The kit provides a test environment for designing and analyzing channels and their performance prior to COM testing. You can also run COM on channel designs. The kit implements most of the characteristics and parameters 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");
```



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

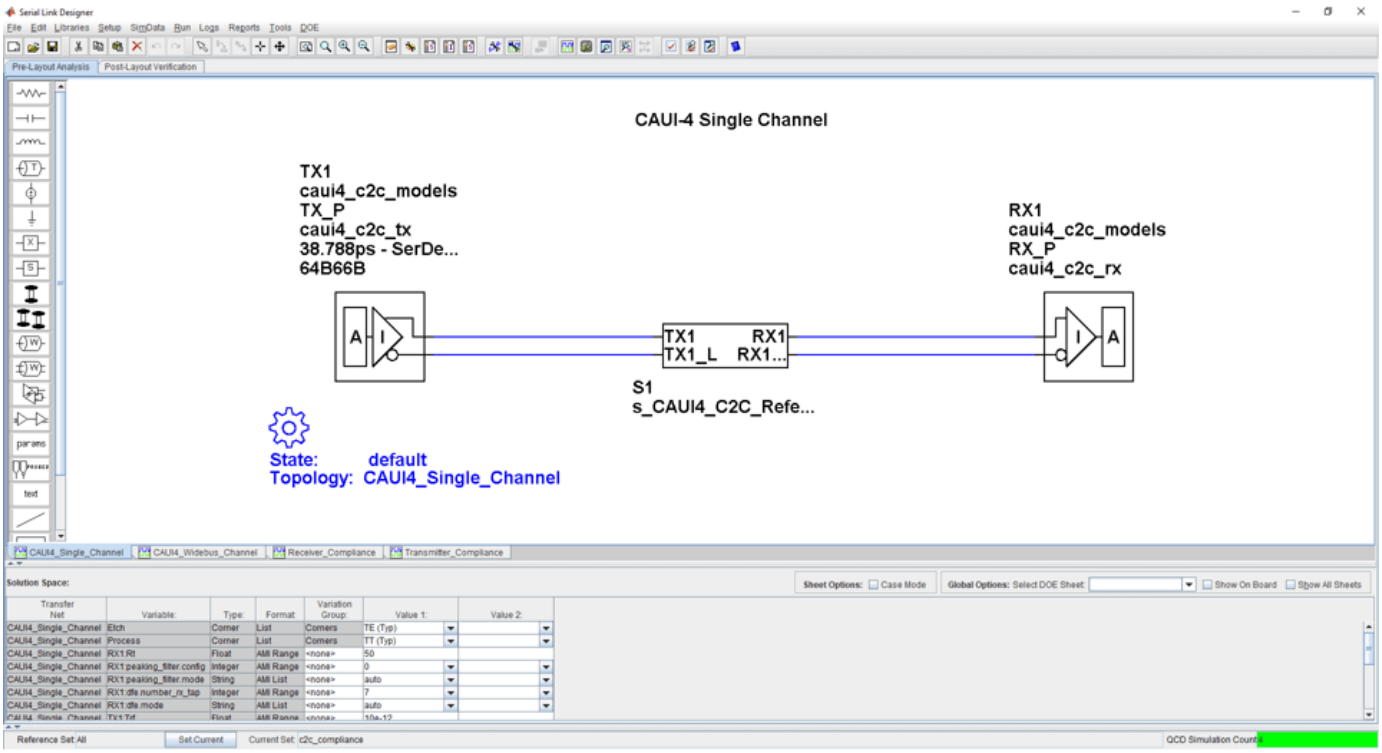
Test the compliance of simulation models and topologies to the CAUI-4 chip-to-chip (C2C) specification.

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");
```



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.

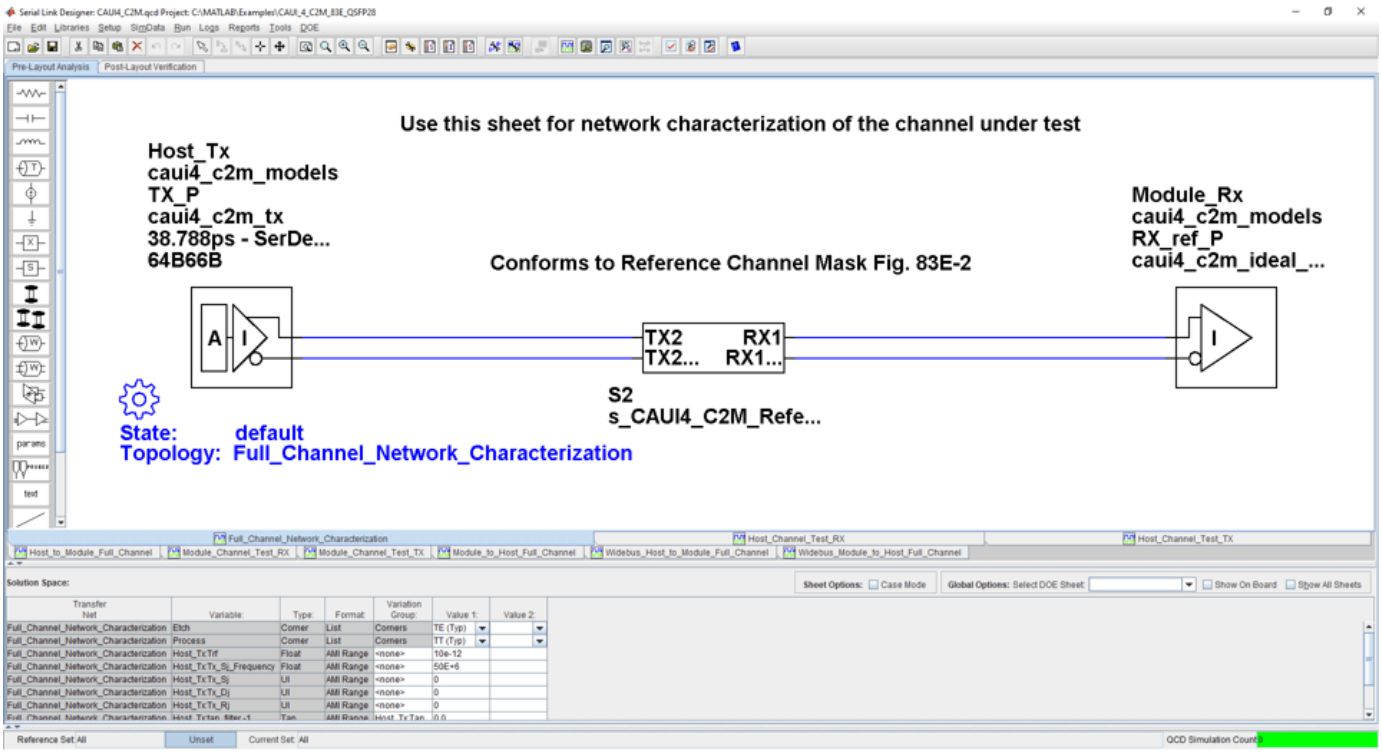
This kit is designed for analysis of a host board and an optical module. The channel consists of a host board connected with a mated connector to a module board. The channels between the transmitting and receiving devices operate at 25.718 Gb/s. The kit has sheets for network characterization to analyze 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");
```



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.

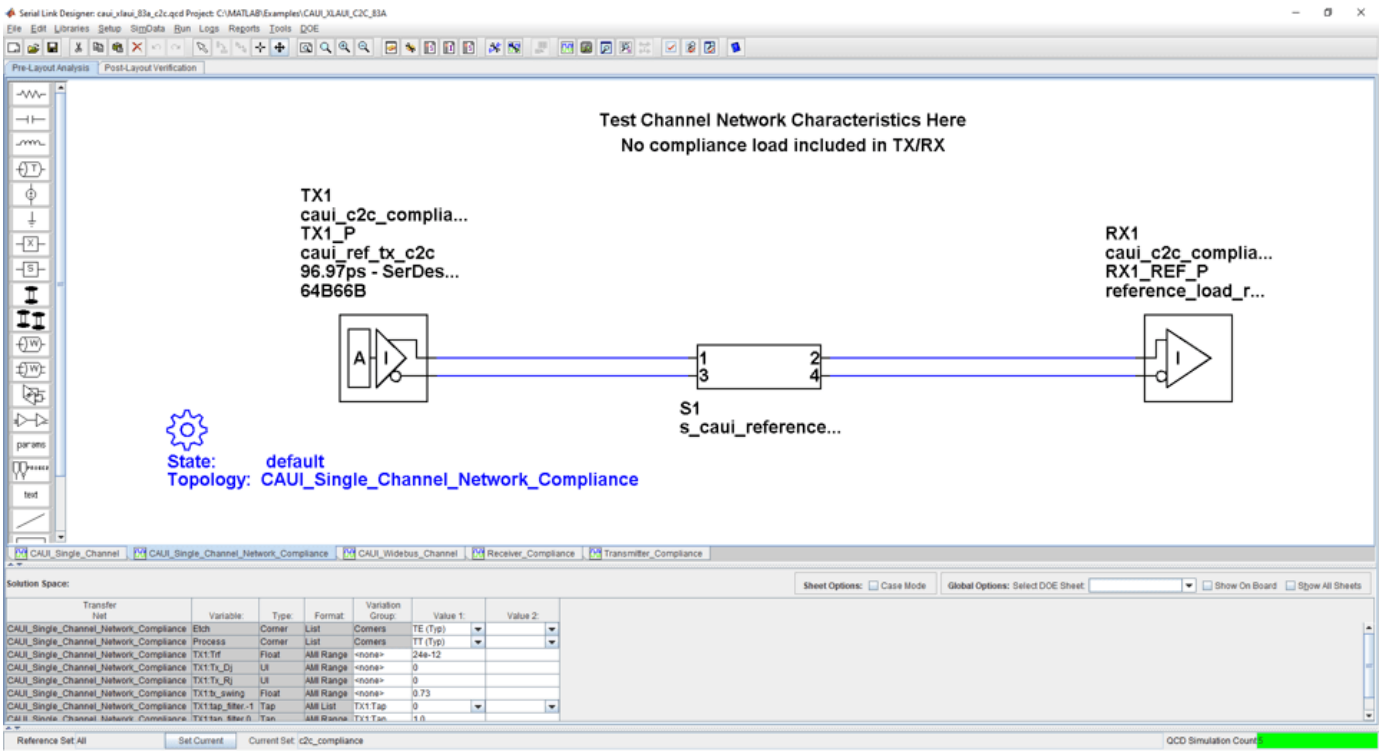
The 802.3ba Annex 83A specification defines physical layer compliance for a CAUI (100 Gb/s) or XLAUI (40 Gb/s) C2C interface.

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");
```



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.

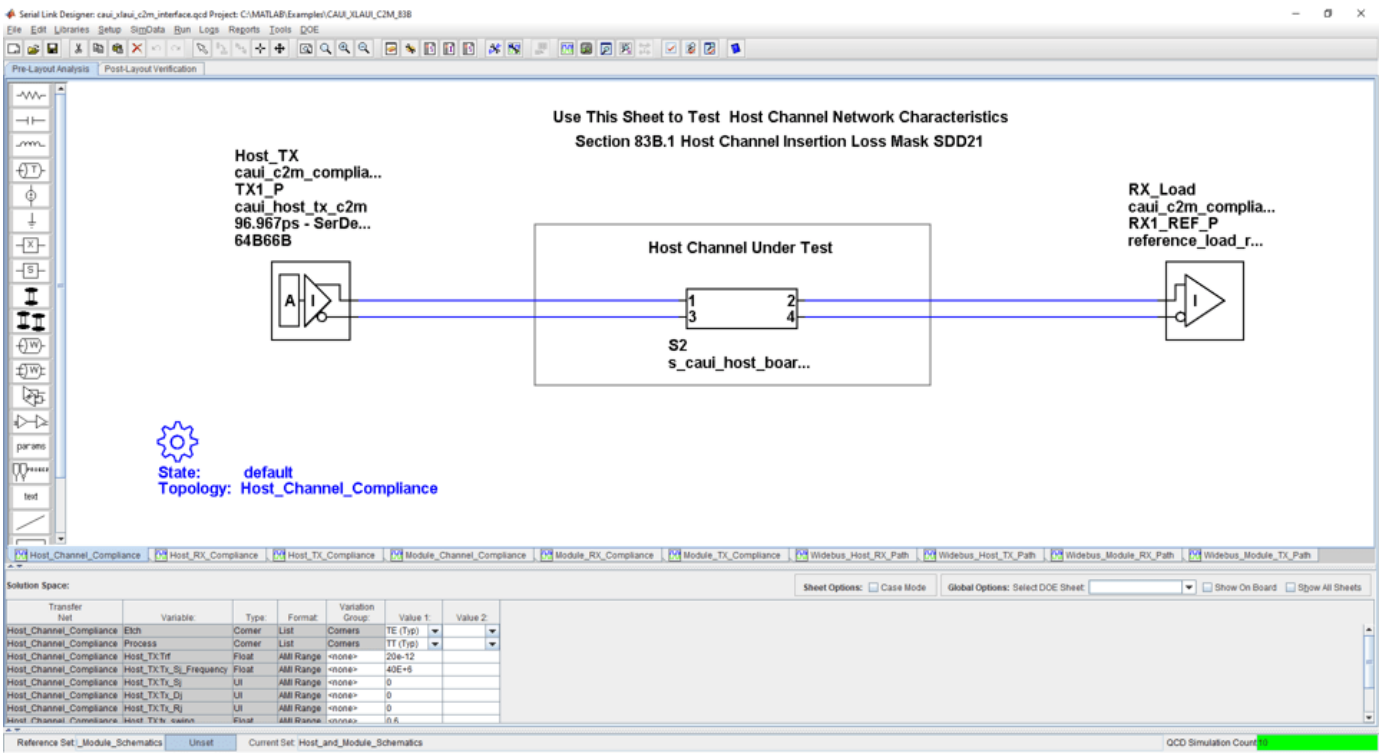
802.3ba Annex 83B specification defines physical layer compliance for a CAUI (100 Gb/s) or XLAUI (40 Gb/s) C2M interface. Host and module board compliance are both detailed including the channel, transmitter and receiver electrical requirements.

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");
```



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.

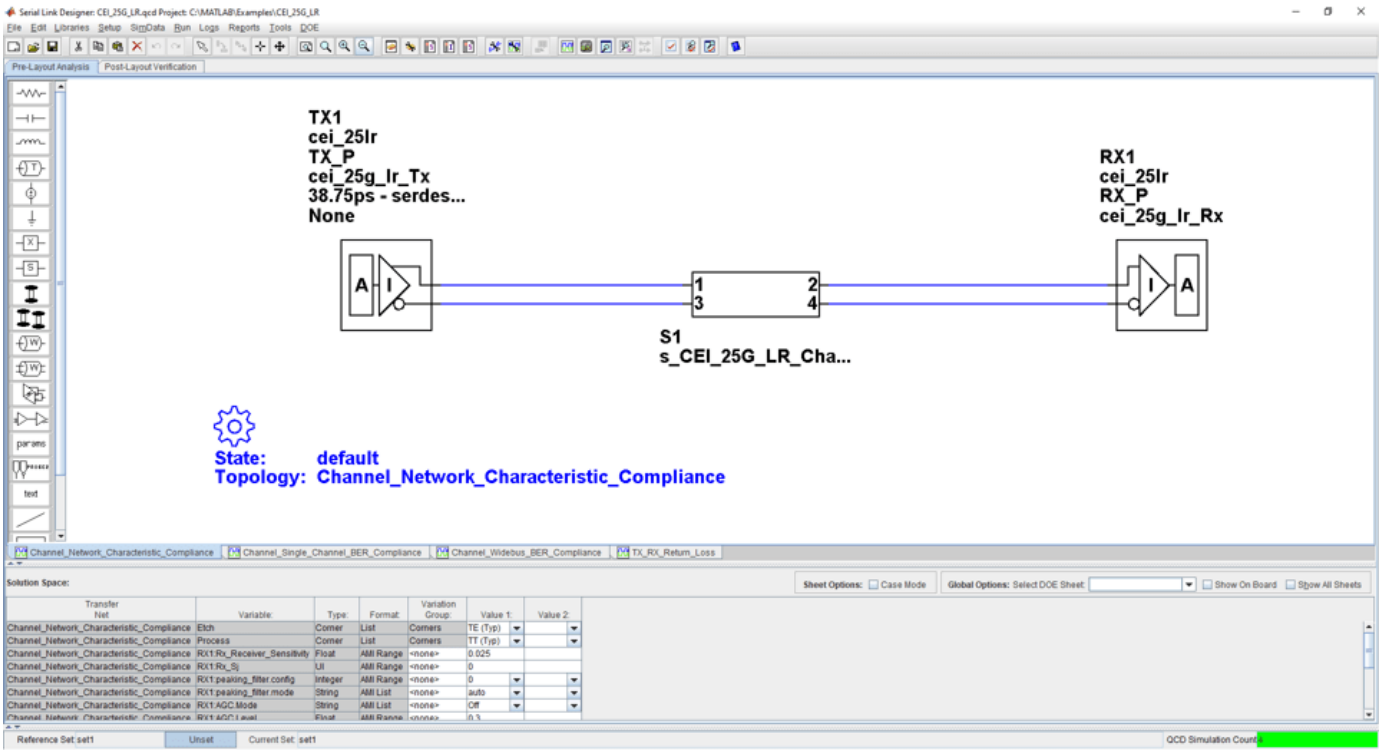
This kit is designed for analysis of a backplane channel design between module boards. The total channel length is approximately 30 inches. It has module boards connected with two mated connectors which represents the interconnect between ASICs transmitting and receiving 25 Gb/s data over the channel. 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");
```



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.

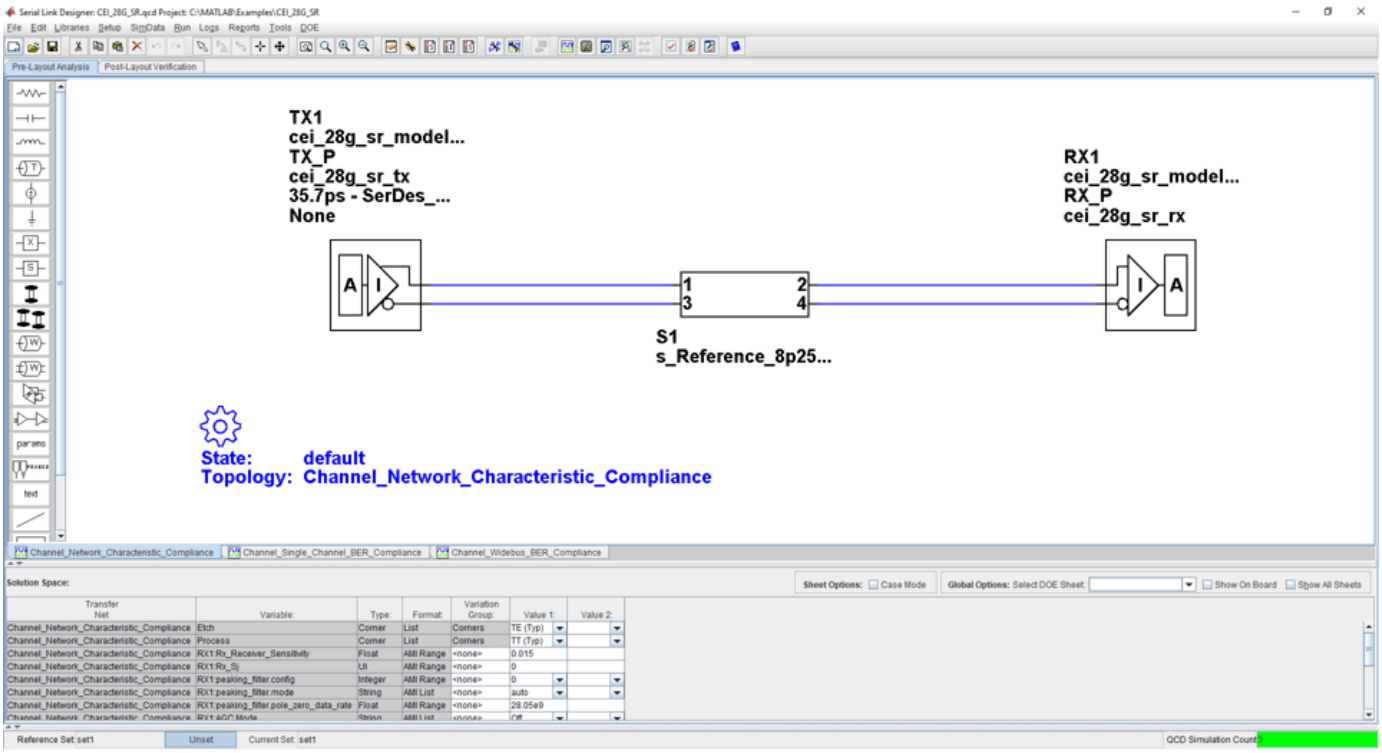
This kit is designed for analysis of a host board and QSFP+ 100G module. The total channel length is approximately 8.25 inches. It consists of a host board connected with a mated connector which represents the interconnect between the transmitting and receiving 28.05 Gb/s data across the channel. The 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");
```



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.

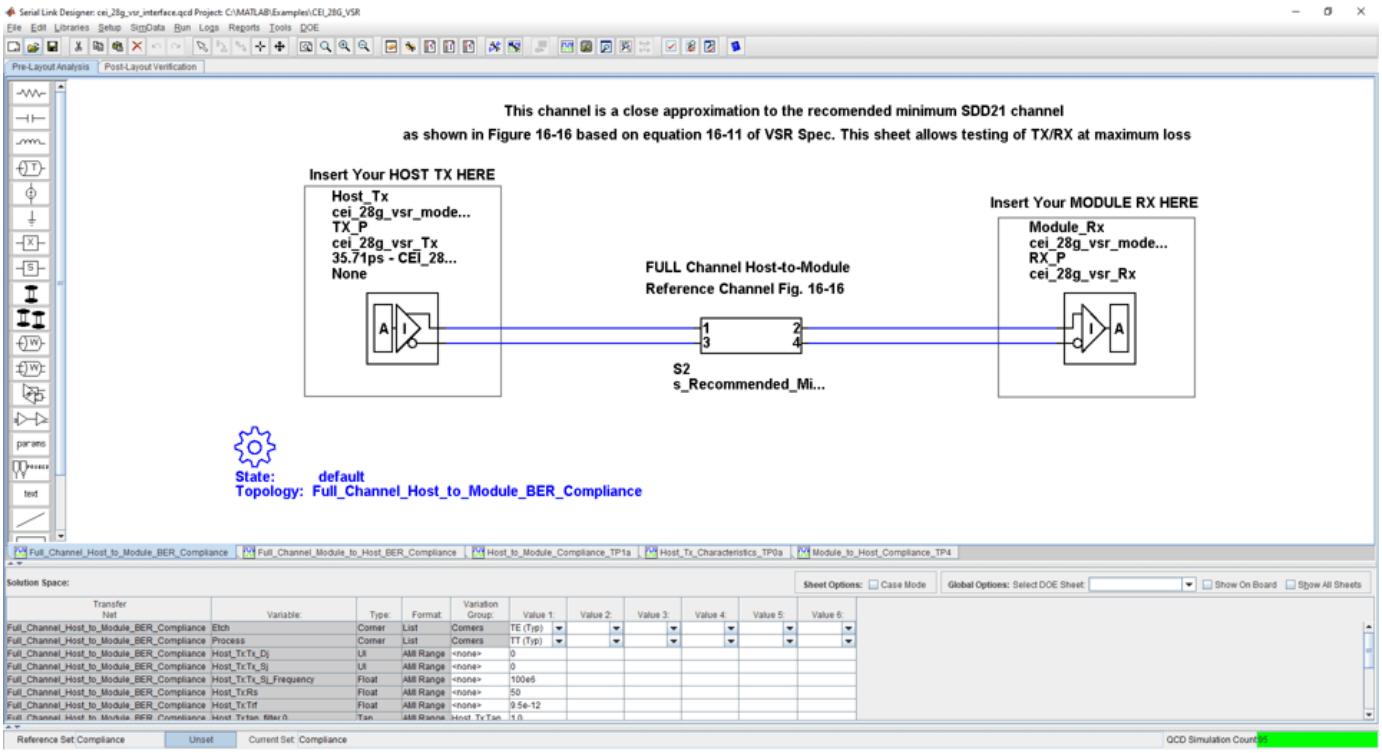
This kit is designed for analysis of a host board and an optical module. The total channel length is approximately 5 inches. The VSR channel consists of a host board connected with a mated connector to a module board which represents the interconnect between the transmitting and receiving 28.05 Gbps 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");
```



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.

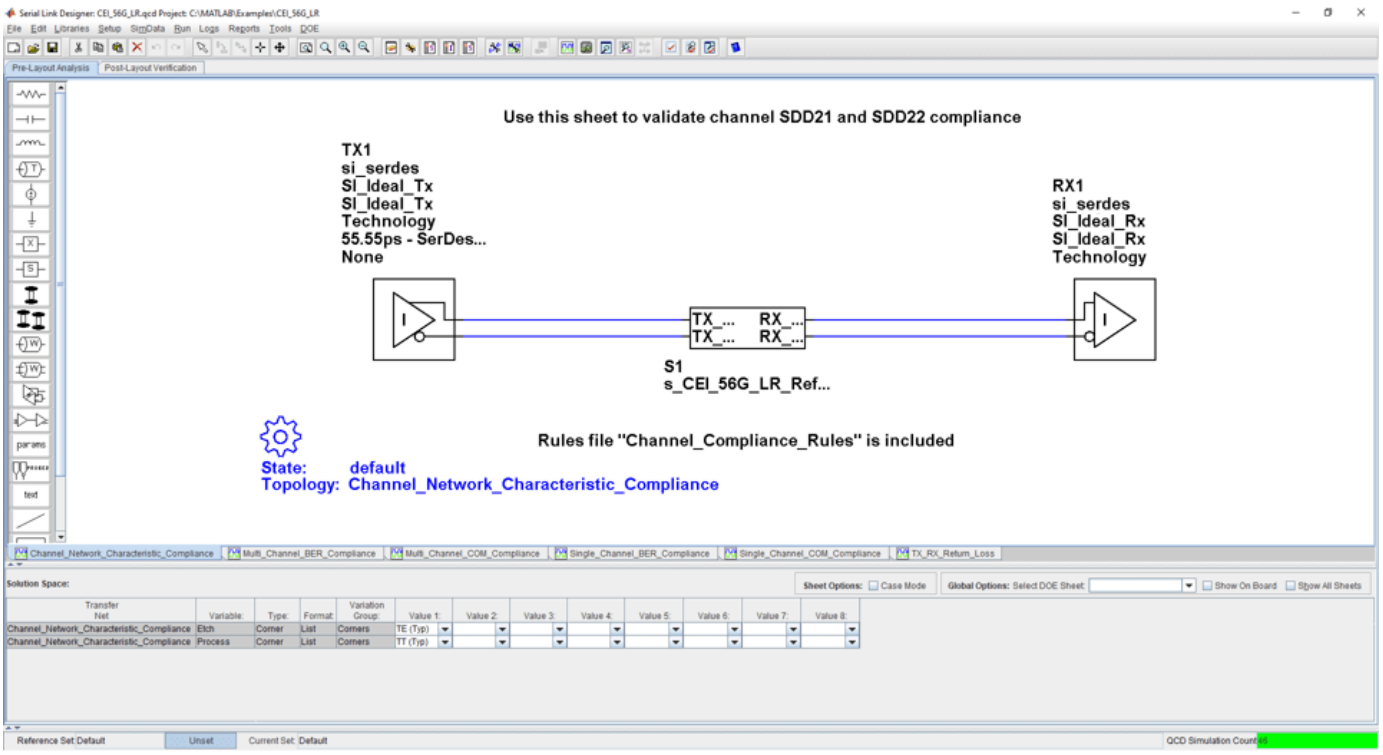
This kit is designed for analysis of a backplane channel design between module boards. The channel model is based on two module boards connected with two mated connectors with PCB trace.

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");
```



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.

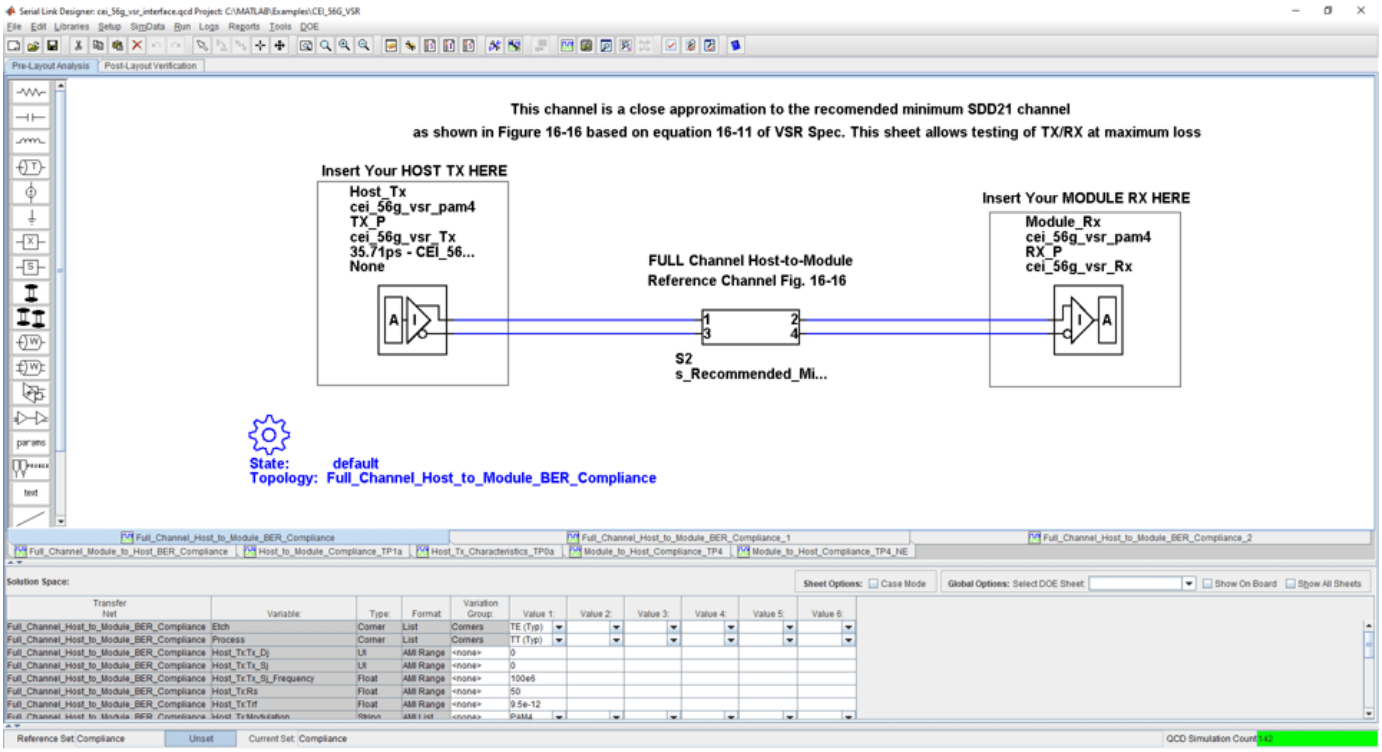
This kit is designed for bidirectional analysis of a host board to an optical module board. The total channel loss at Nyquist or Fb/2 is approximately 10 dB. The VSR channel consists of a host board connected with a mated connector to a module board that represents the interconnection between the transmitting 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");
```



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.

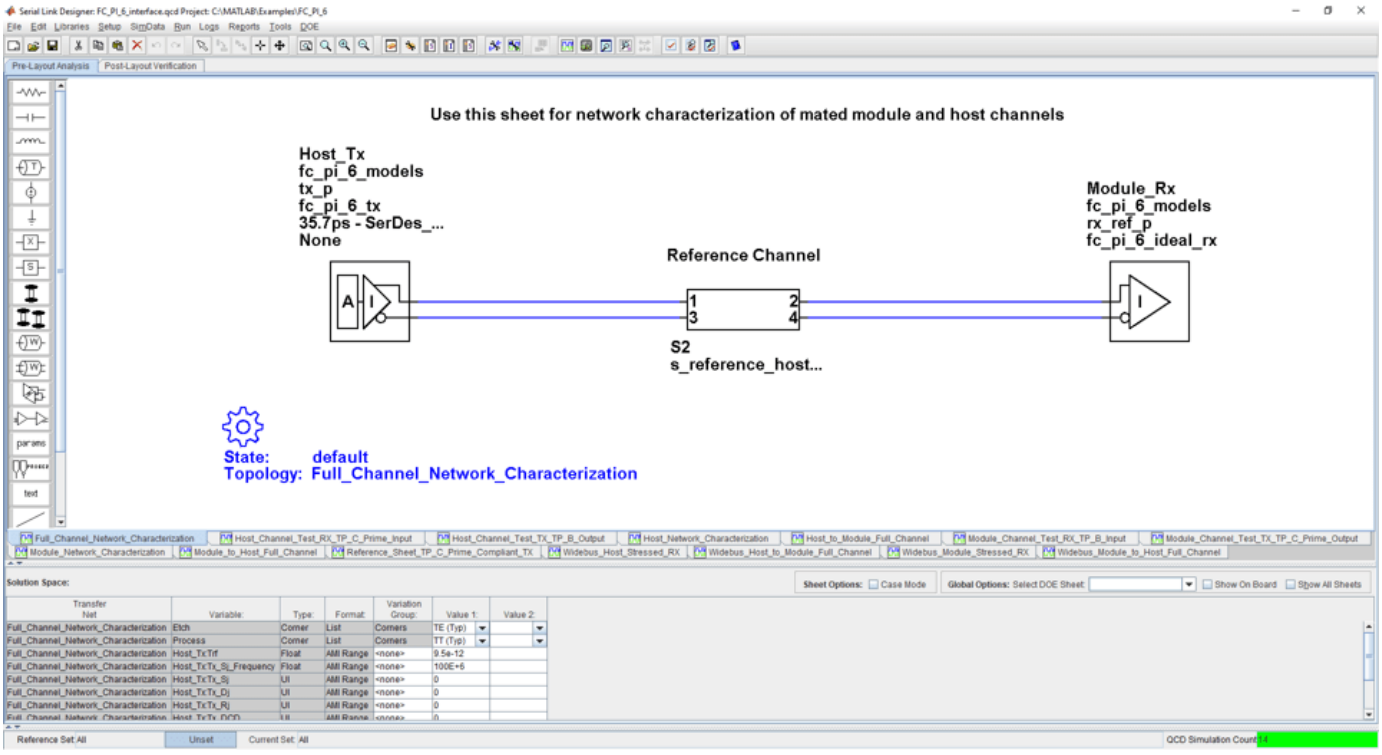
This kit is designed for analysis of a host board and an optical module. The channel consists of a host board connected with a mated connector to a module board which represents the interconnect between the transmitting and receiving 28.05 Gb/s data across the channel. The kit has sheets that represent 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");
```



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.

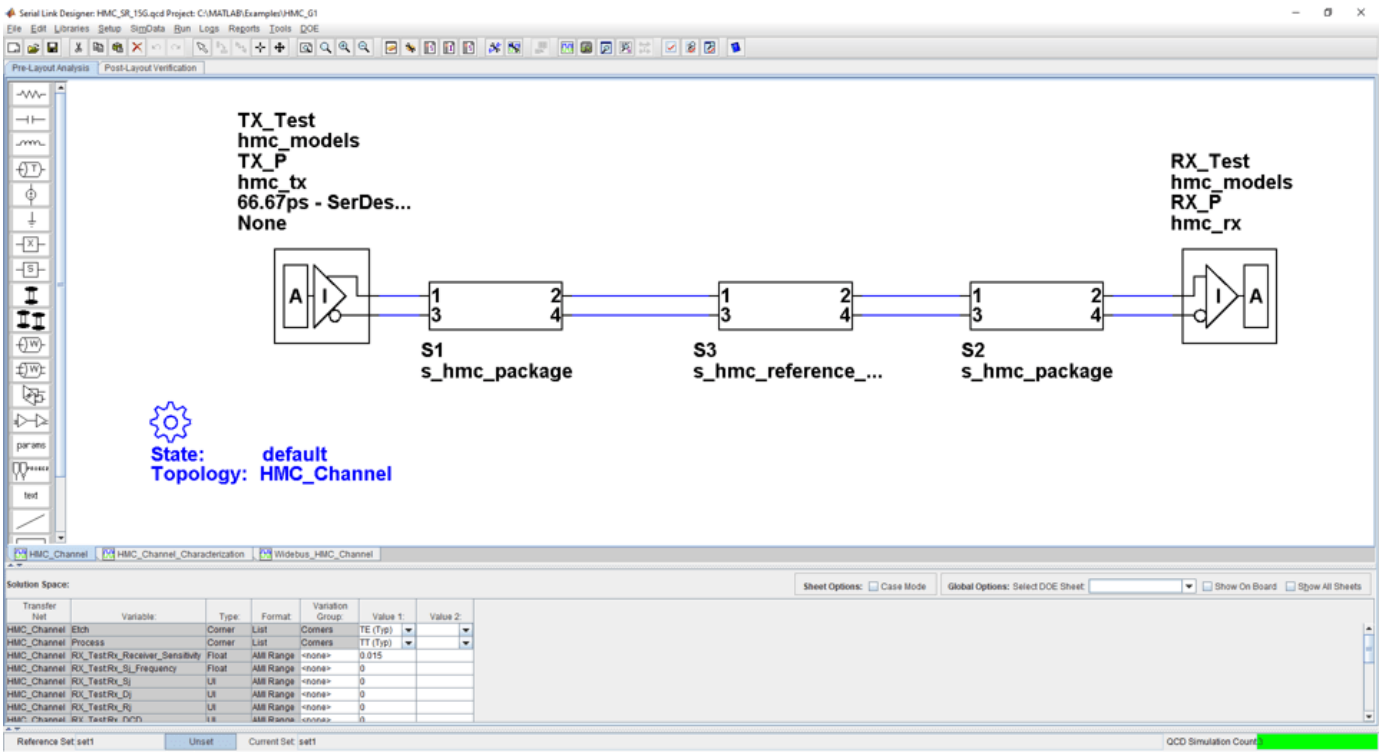
This kit is designed for analysis of an interface between a host ASIC and a Cube. The total channel length between devices is approximately 10 inches. The kit has sheets that represent network characterization for insertion and return loss testing and both single channel and multi-bit channel simulation 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");
```



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.

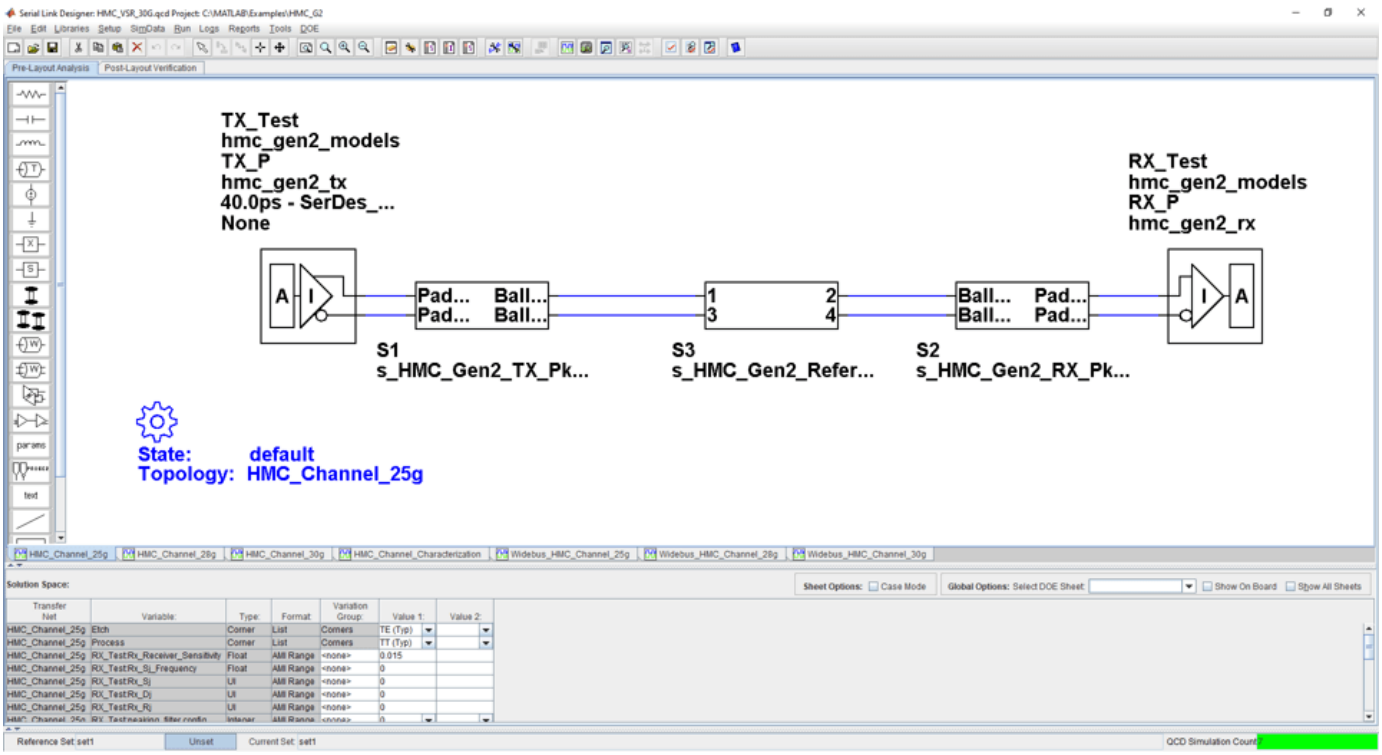
This kit is designed for analysis of an interface between a host ASIC and a Cube. The total channel length between devices is approximately 10 inches. The kit has sheets that represent network characterization for insertion and return loss testing and both single channel and multi-bit channel simulation 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");
```



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

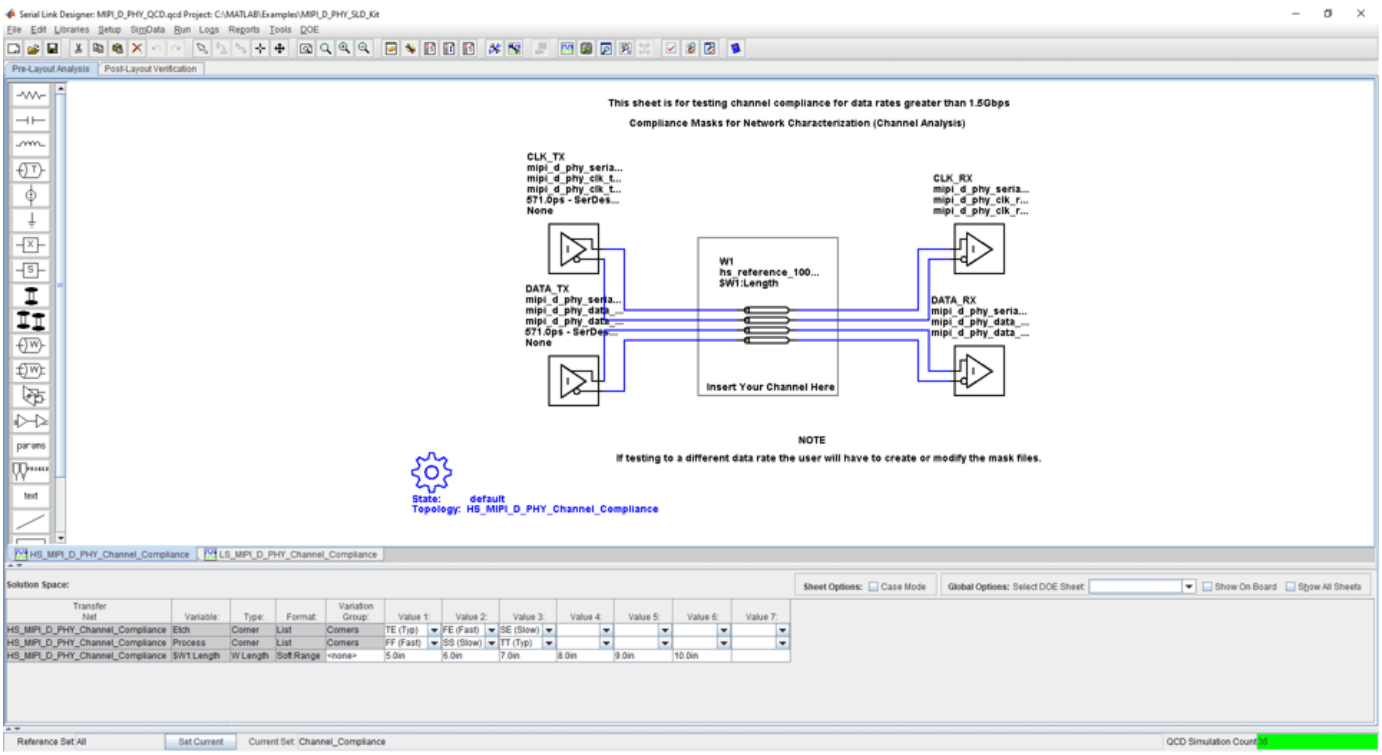
Test the compliance of a channel to the MIPI D-PHY specification using **Serial Link Designer**.

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");
```



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.

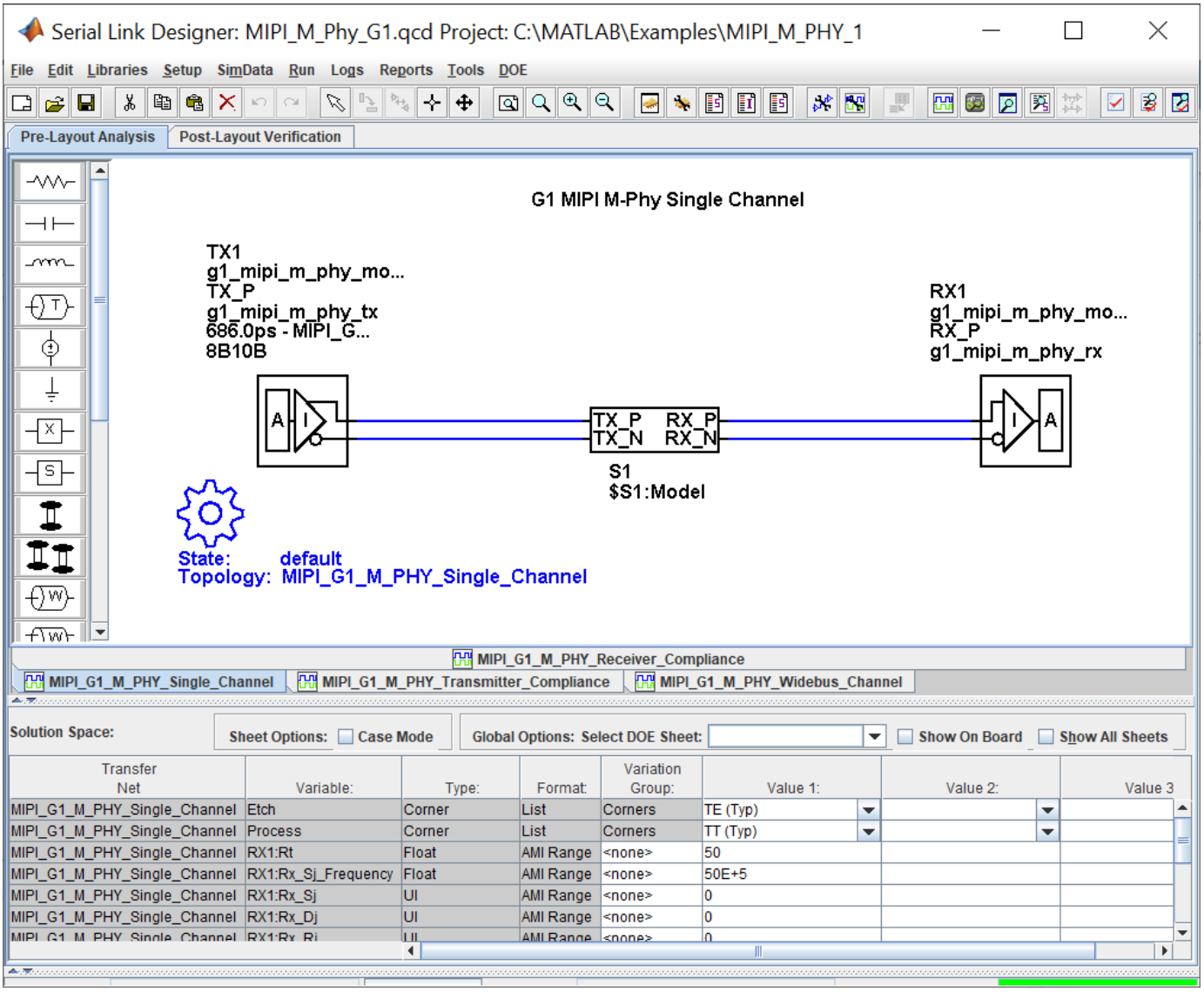
This kit is designed for an interface between system devices with up to 25 inches of PCB etch and one connector. The kit includes IBIS-AMI TX and RX models for reference and compliance testing.

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");
```



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.

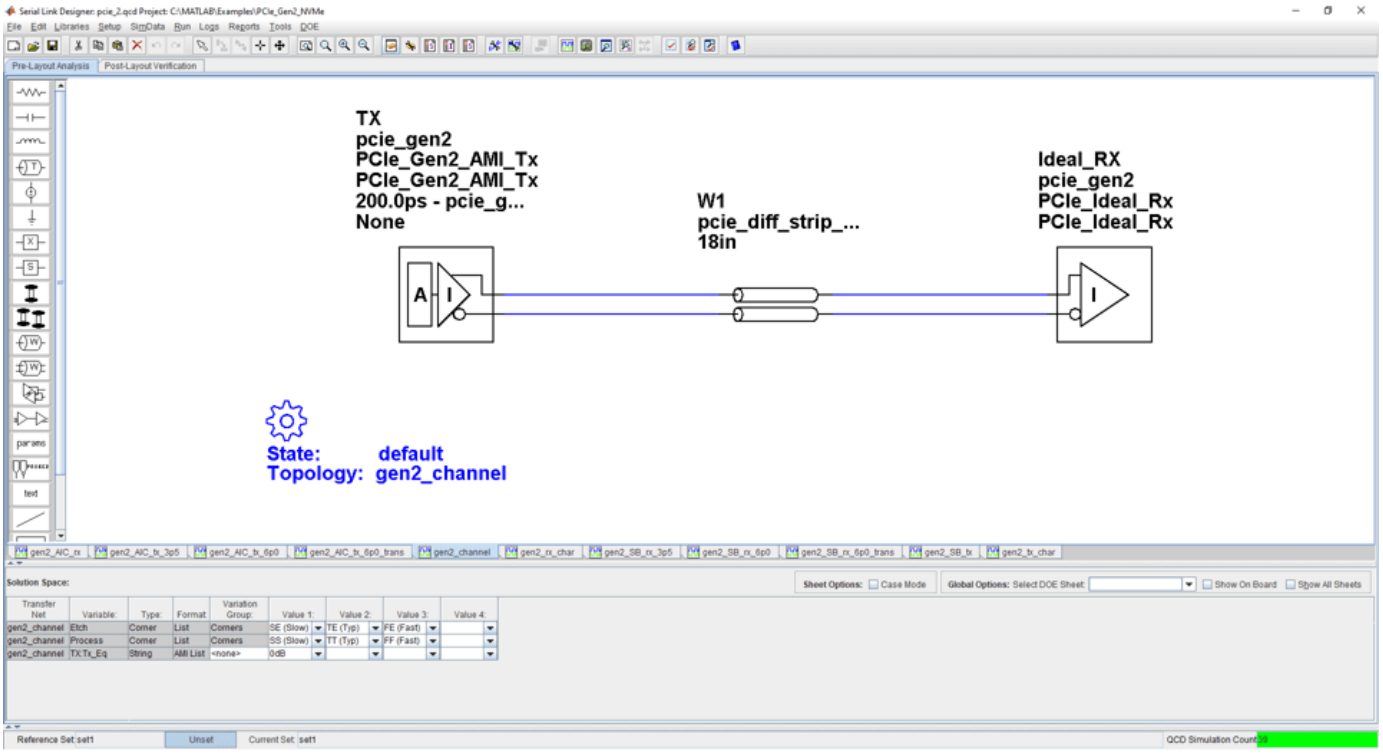
This PCIe signal integrity kit includes all the transfer nets, topologies, generic buffer models and compliance rules for a PCIe-2 high-speed SerDes interface. This includes PCIe-2 technology IBIS-AMI models for the SerDes transmitter and receiver, PCIe-2 compliance masks and transfer nets preconfigured 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");
```



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.

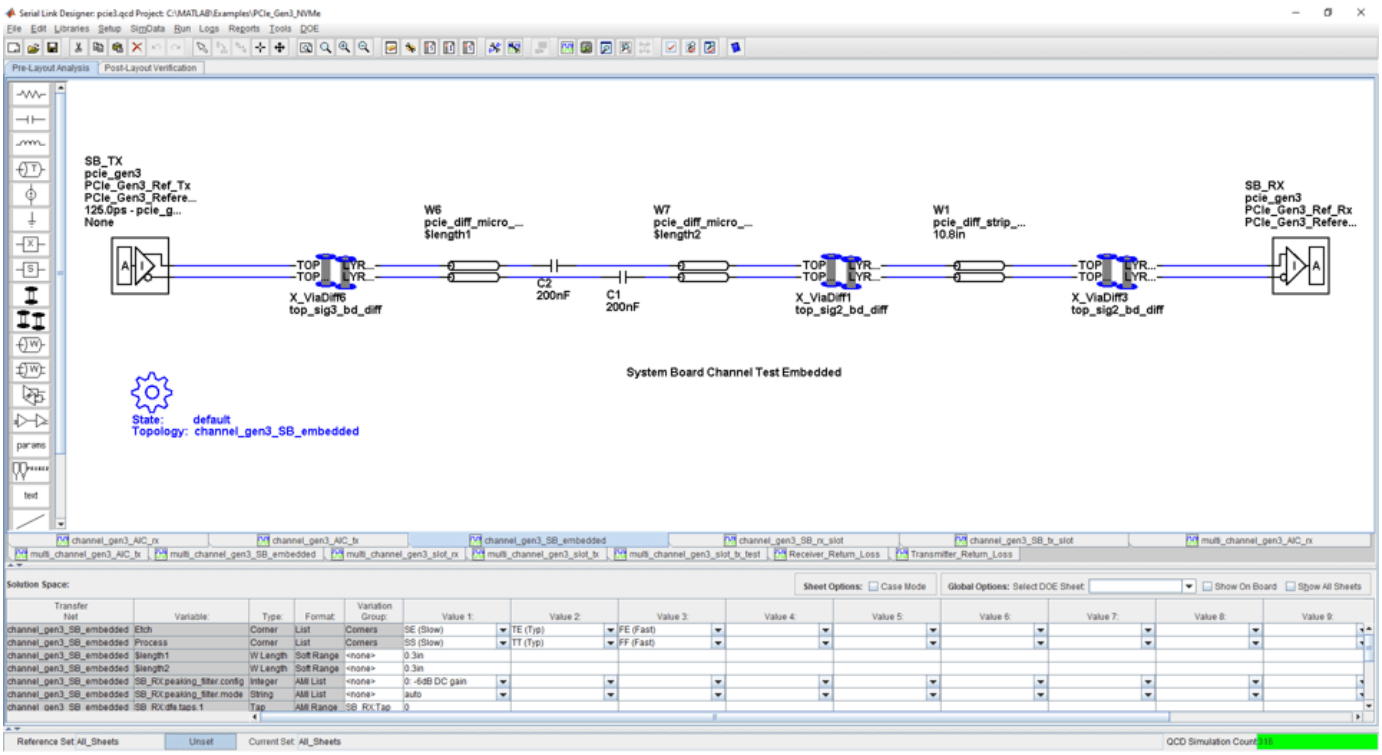
The PCIe-3 signal integrity kit includes all the transfer nets, topologies, generic buffer models and compliance rules for a PCIe-3 high-speed SerDes interface. This includes PCIe-3 technology IBIS-AMI models for the SerDes transmitter and receiver, PCIe-3 compliance masks and transfer nets preconfigured 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");
```



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.

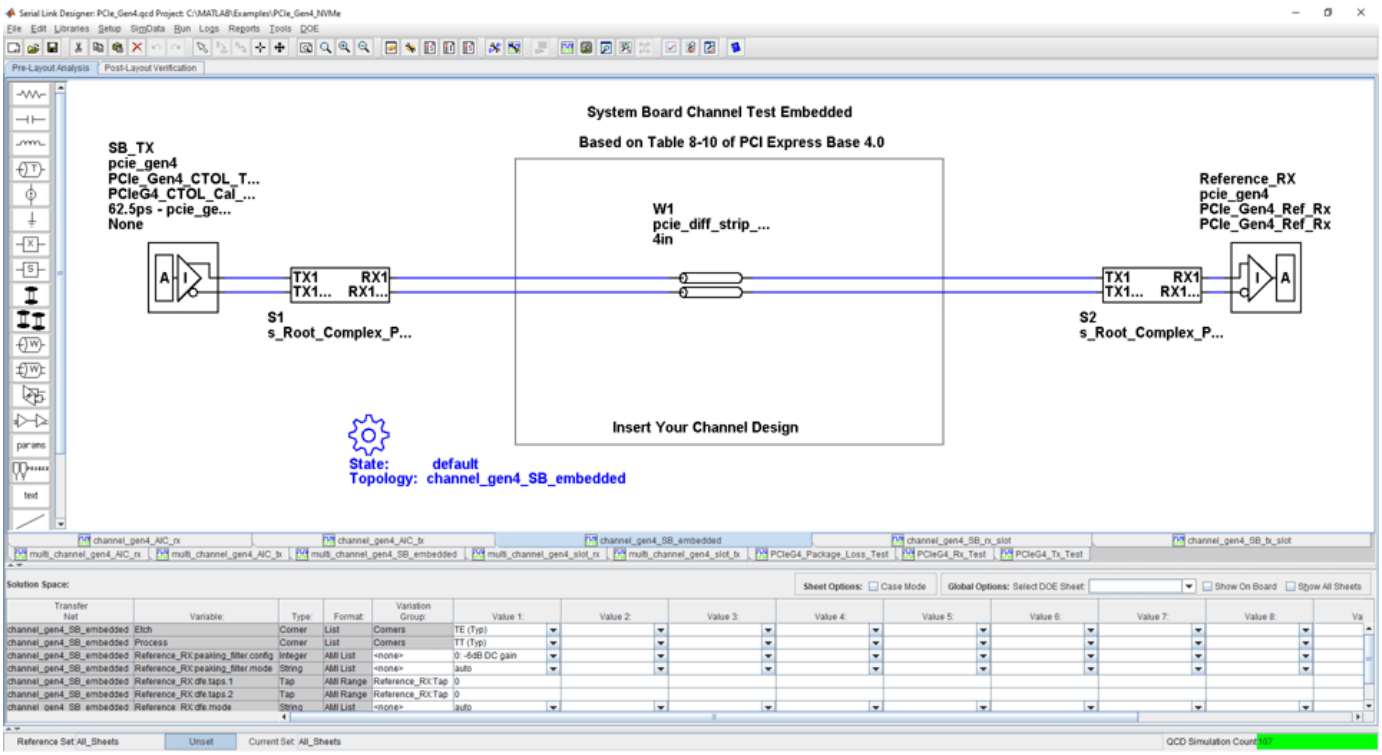
This PCIe compliance signal integrity kit includes all the transfer nets, topologies, generic buffer models and compliance rules for a PCIe-4 high-speed SerDes interface. This includes PCIe-4 technology IBIS-AMI models for the SerDes transmitter and receiver, PCIe-4 compliance masks and transfer nets 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");
```



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.

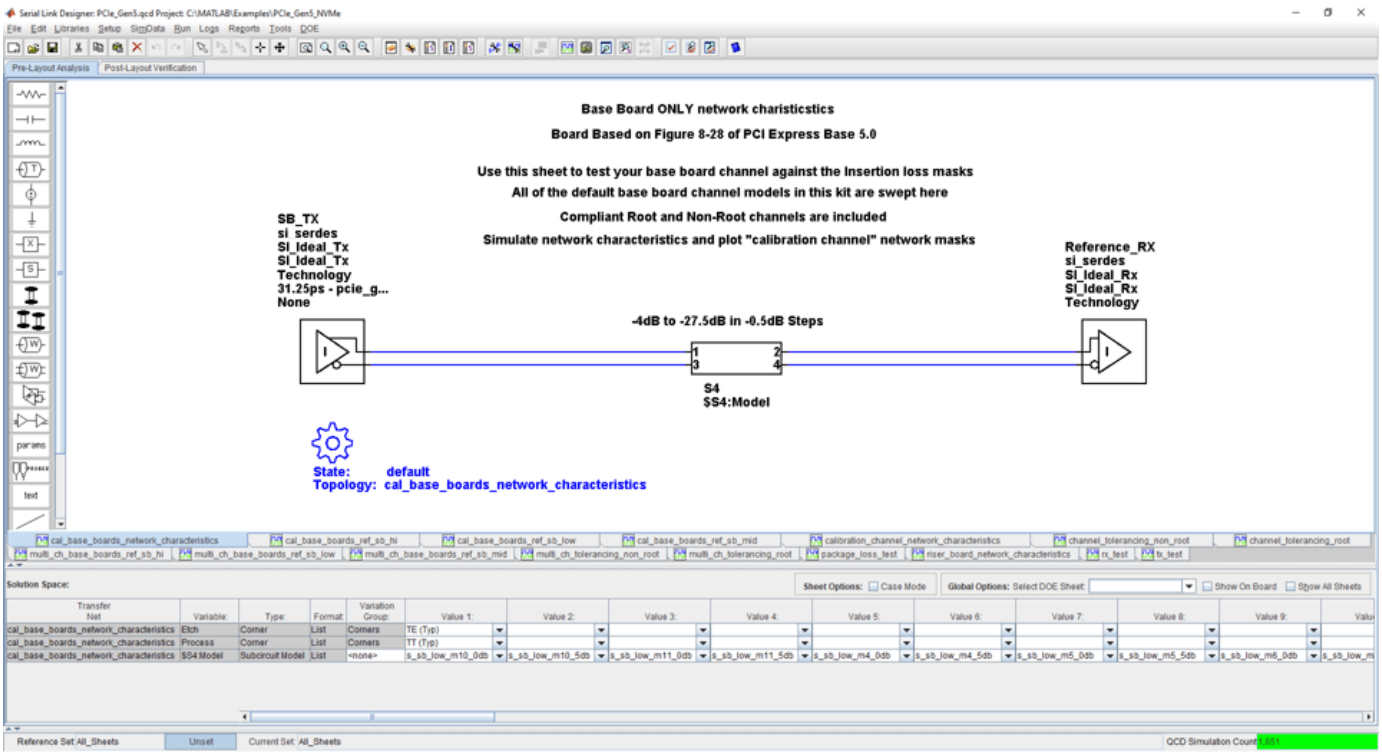
This PCIe compliance signal integrity kit includes all the transfer nets, topologies, generic buffer models and compliance rules for a PCIe-5 high-speed SerDes interface. This includes PCIe-5 technology IBIS-AMI models for the SerDes transmitter and receiver, PCIe-5 compliance masks and transfer nets 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");
```



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.

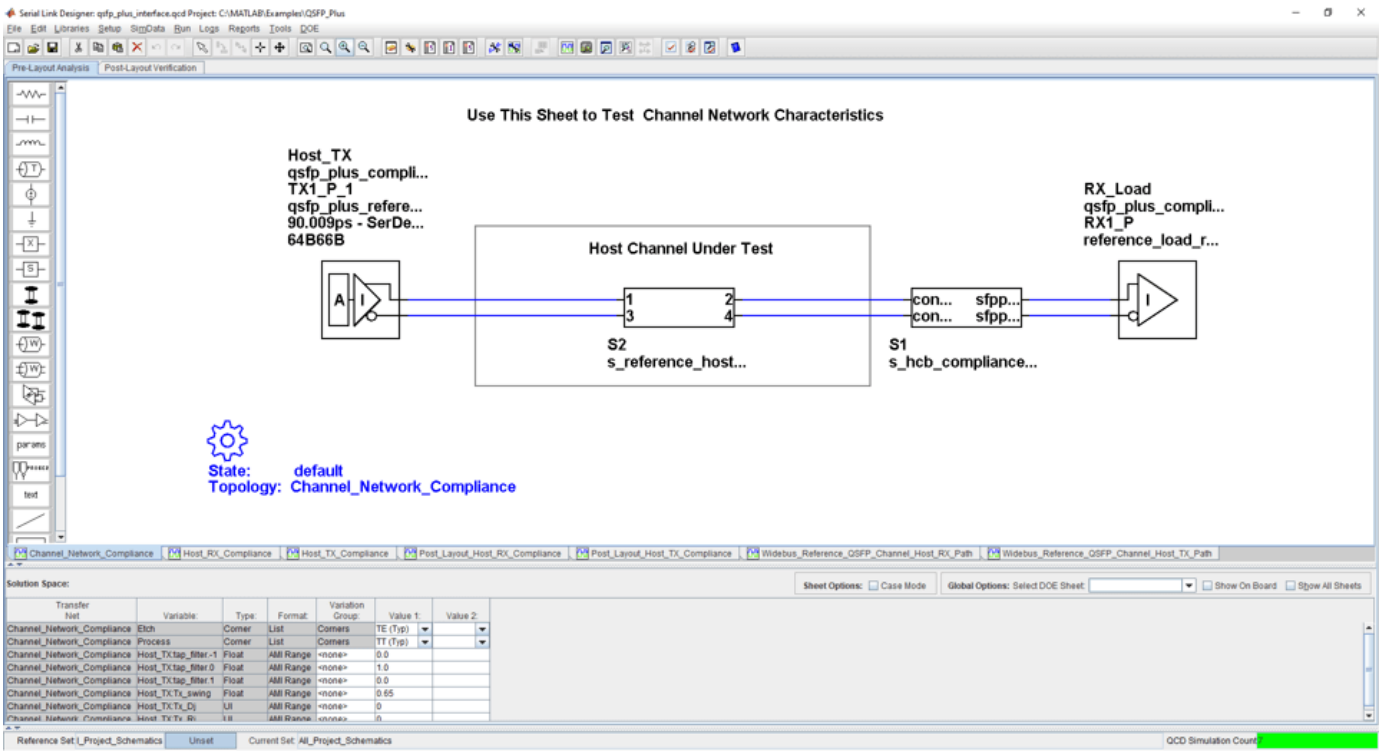
A QSFP+ link is made up of four SFP+ channels that are synchronized within the receiving module to support an aggregate 40 Gb/s link.

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");
```



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.

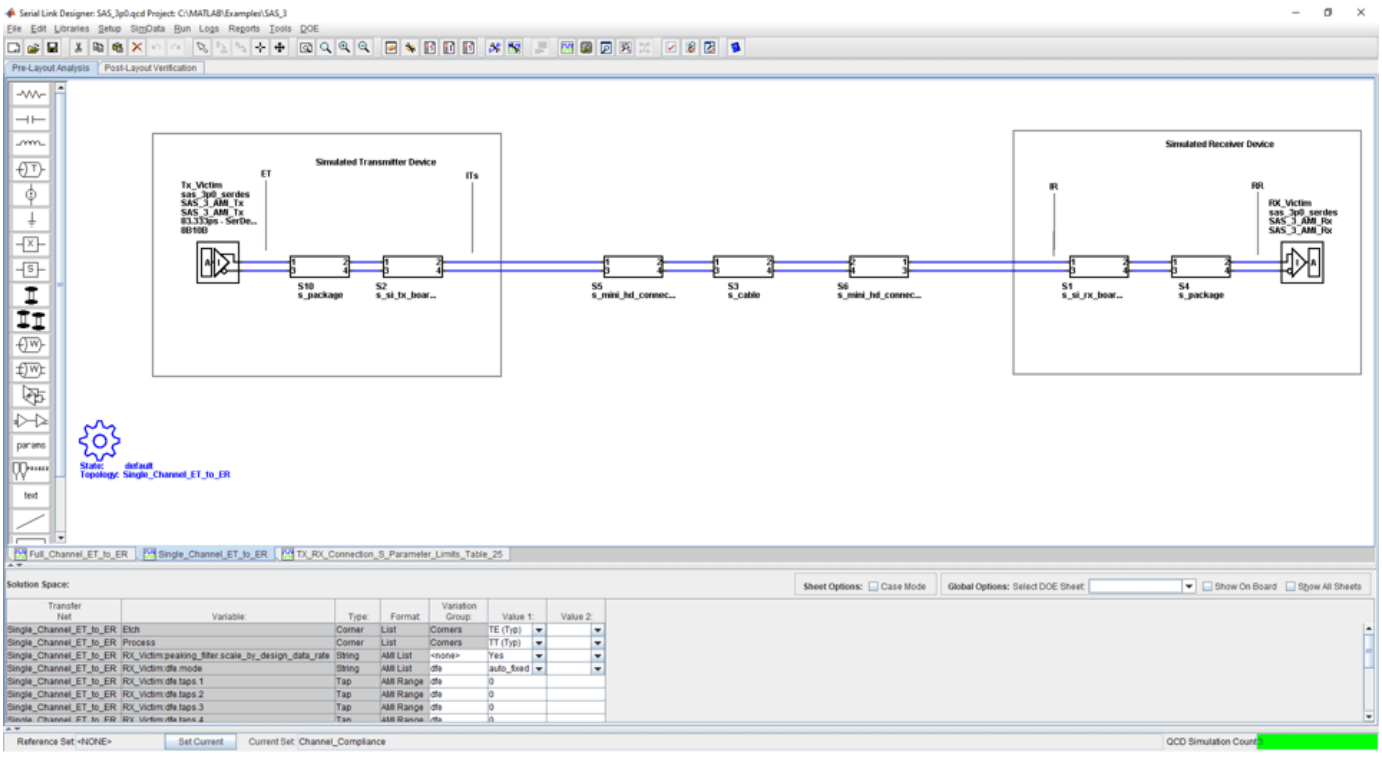
This kit is designed for analysis of a channel design with two mated connectors as provided on the current SAS 3.0 specification provided on the T10 website. The kit also includes sheets containing the backplane/cable and connectors with two plug-in cards attached. In addition, SAS3 reference IBIS-AMI 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");
```



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.

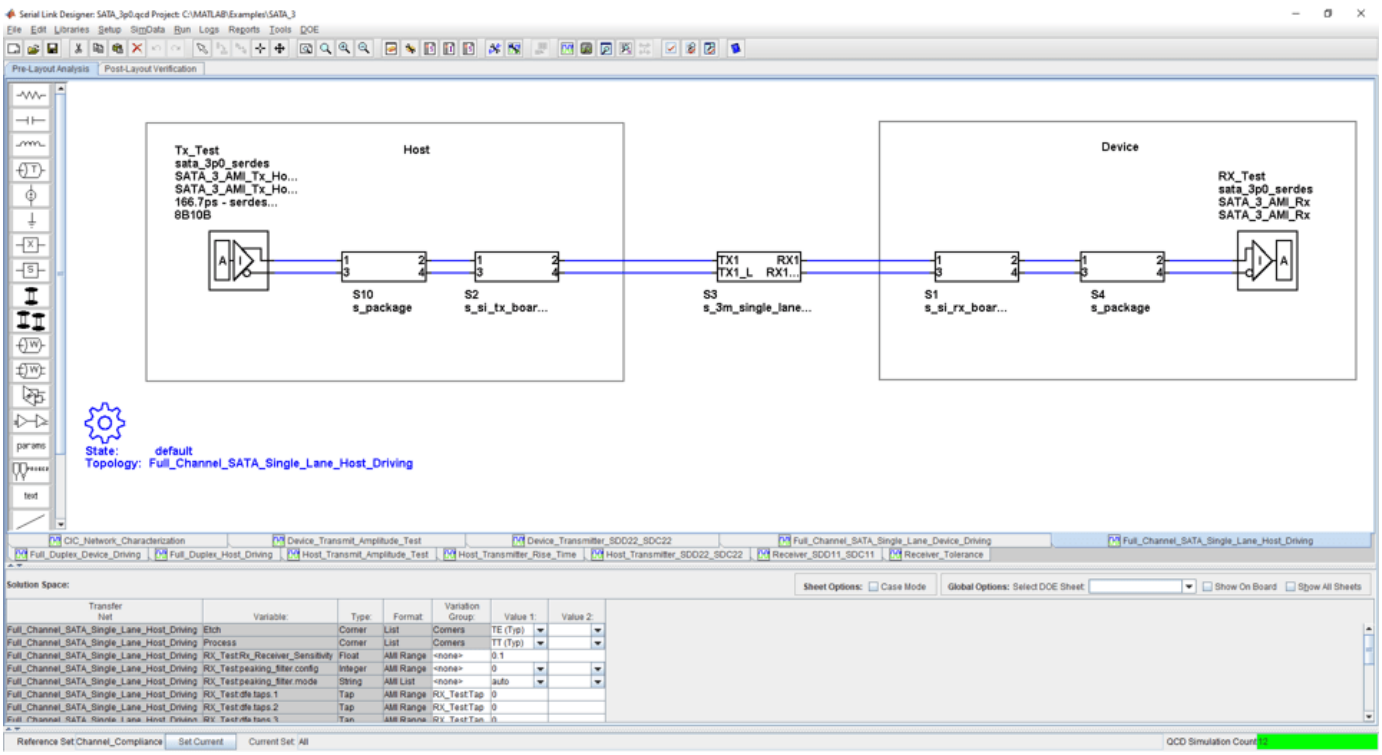
This kit is designed for analysis of a channel design between the SATA 3.0 host and a SATA 3.0 device. The channel consists of a host board and a device board connected by a SATA cable with two mated connectors consistent with the SATA 3.0 specification.

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");
```



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

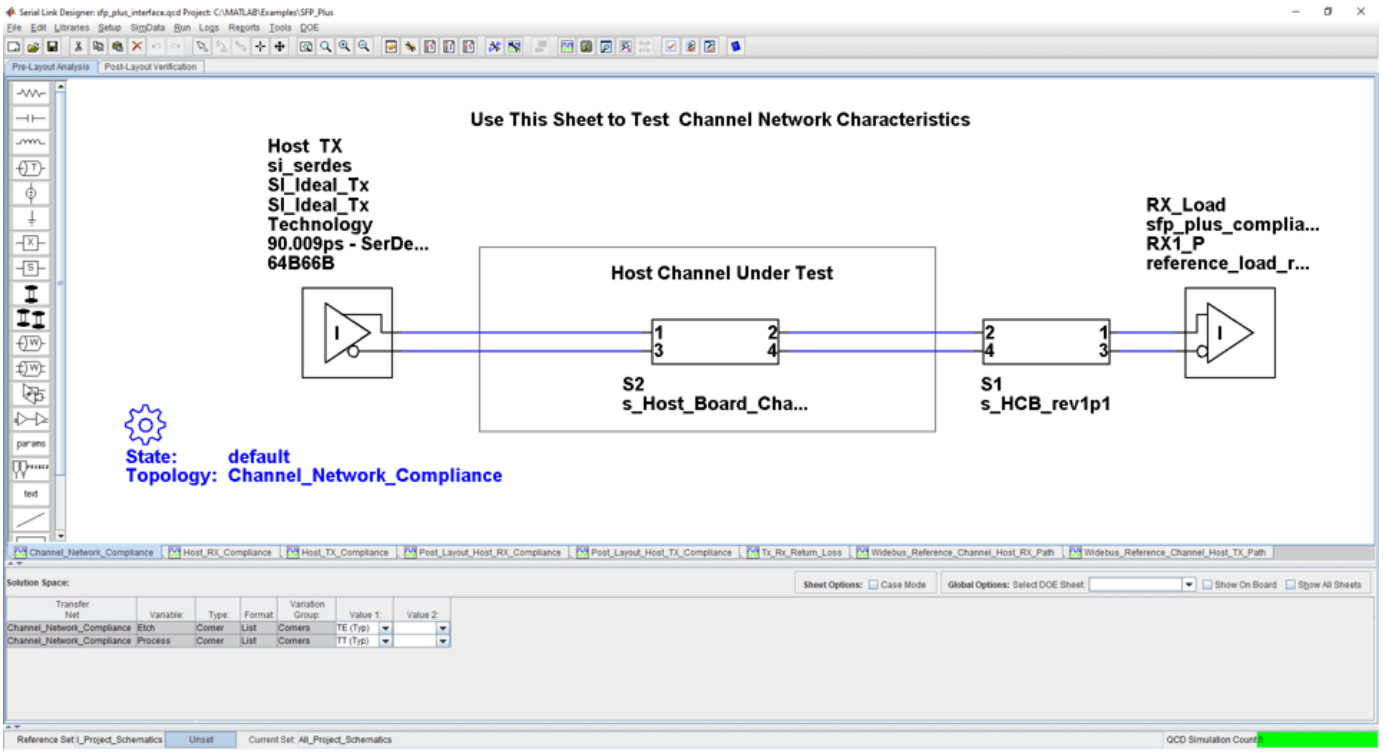
Test the channel design of a host board for compliance to the SFP+ specification.

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");
```



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.

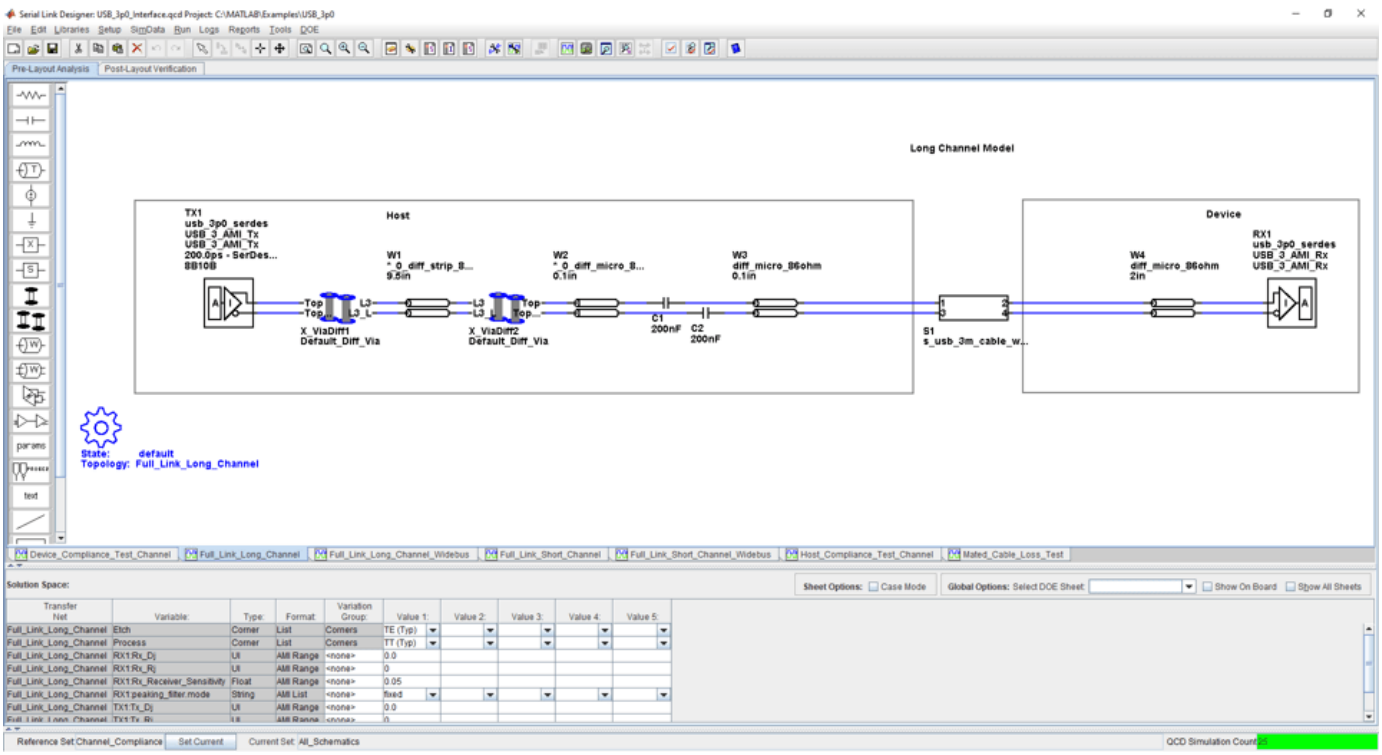
This kit is designed for analysis of a channel design between the USB 3.0 host and a USB hub, or between a USB 3.0 hub and a USB 3.0 device. The channel consists of a host board and a device board (hub or peripheral device) connected by a USB cable with two mated connectors consistent with the USB 3.0 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");
```



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.

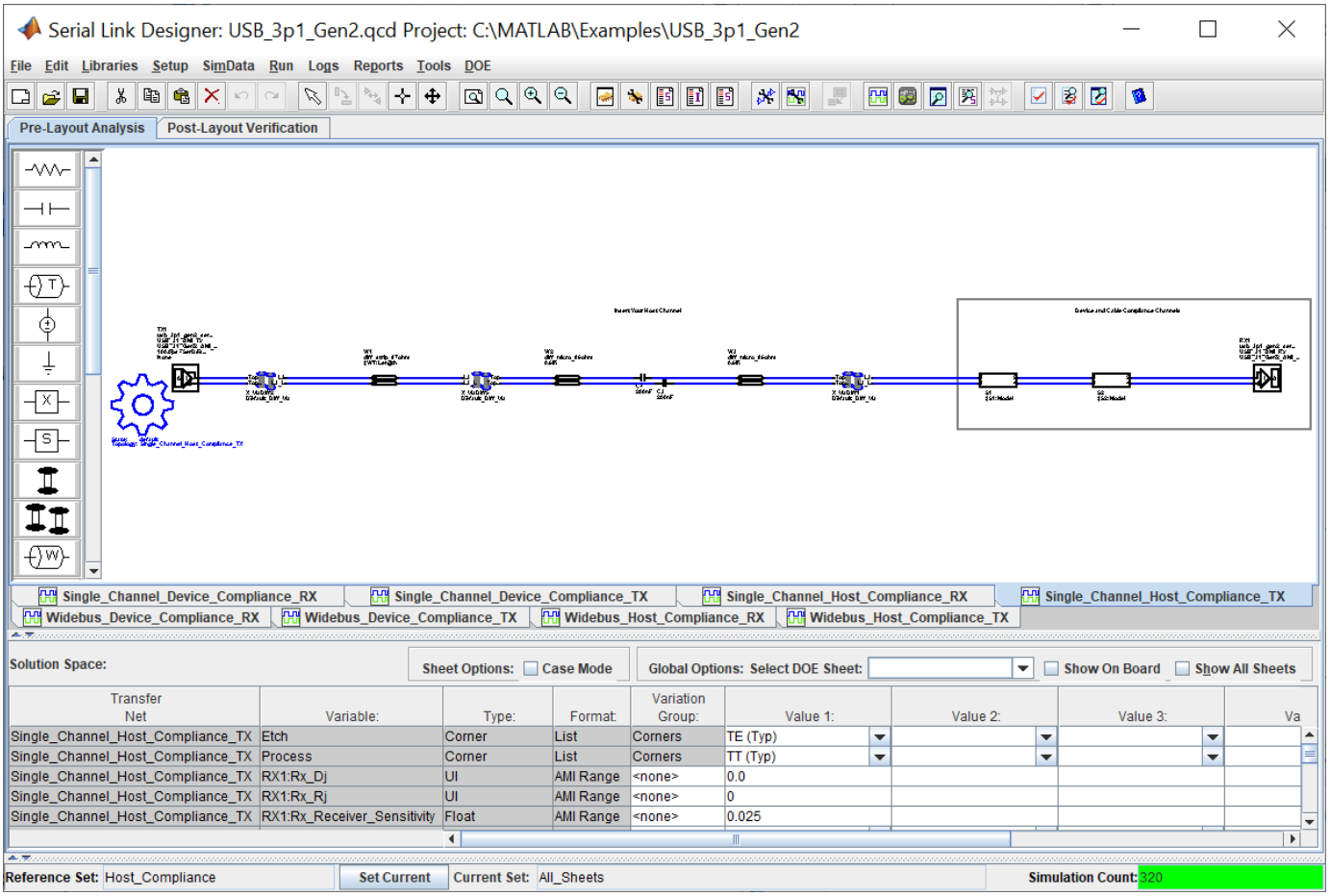
This kit is designed for analysis of a channel design between the USB 3.1 host and device. One schematic set focuses on host compliance and one schematic set focuses on device compliance. Both single channel schematics and multi-channel schematics are provided for host and device compliance. Reference 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");
```



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.

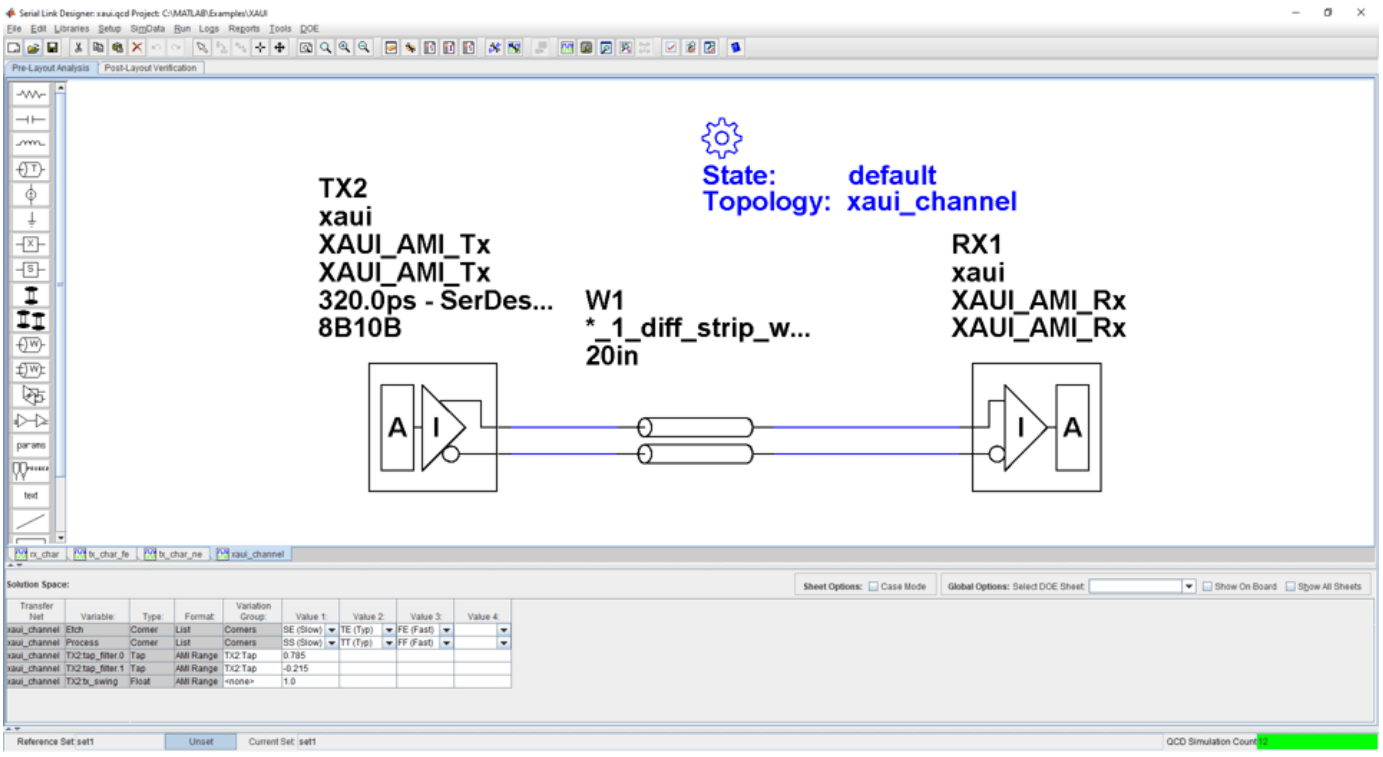
This XAUI compliance signal integrity kit includes all the transfer nets, generic buffer models, and eye masks for an XAUI high-speed SerDes interface. This includes XAUI technology IBIS-AMI models for the SerDes transmitter and receiver, XAUI eye masks, transfer nets preconfigured for TX and RX characterization, 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");
```



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

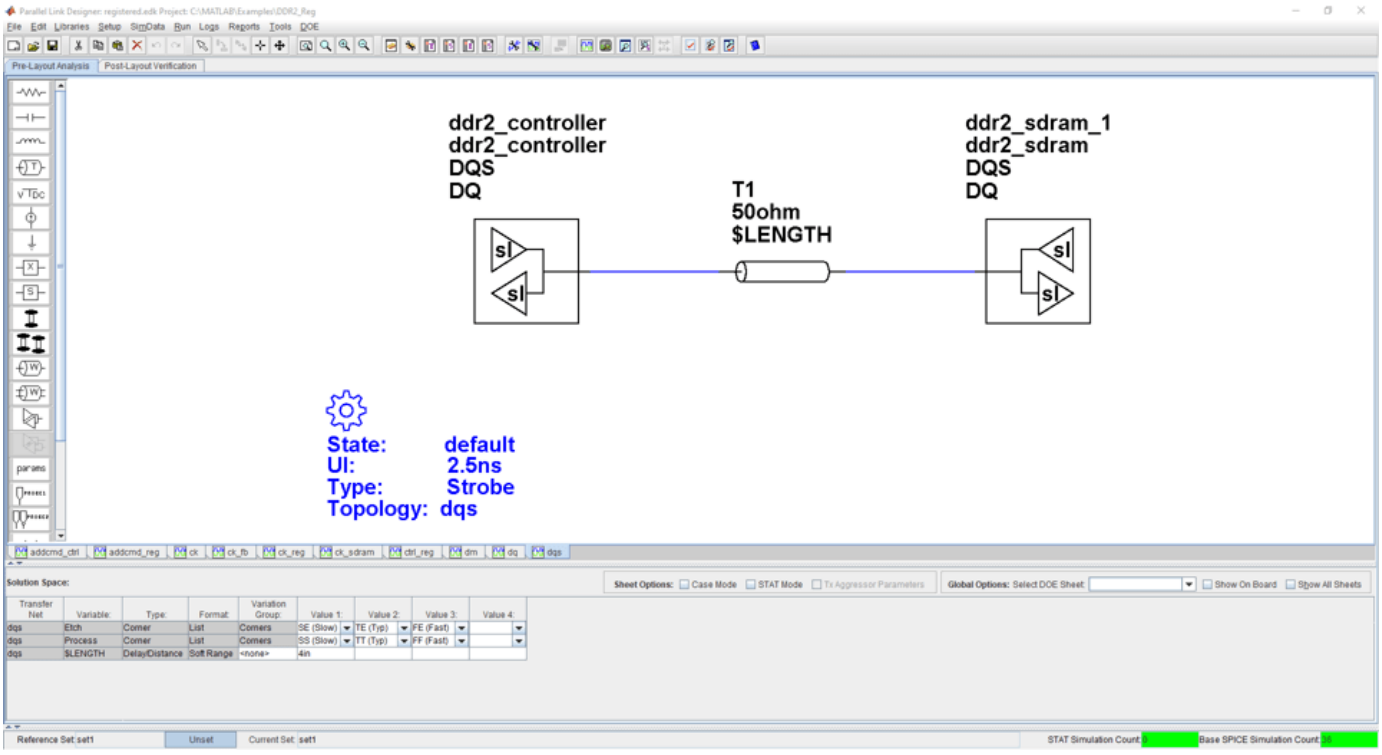
Implement a registered DDR2 interface for pre-layout analysis or post-layout verification.

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");
```



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

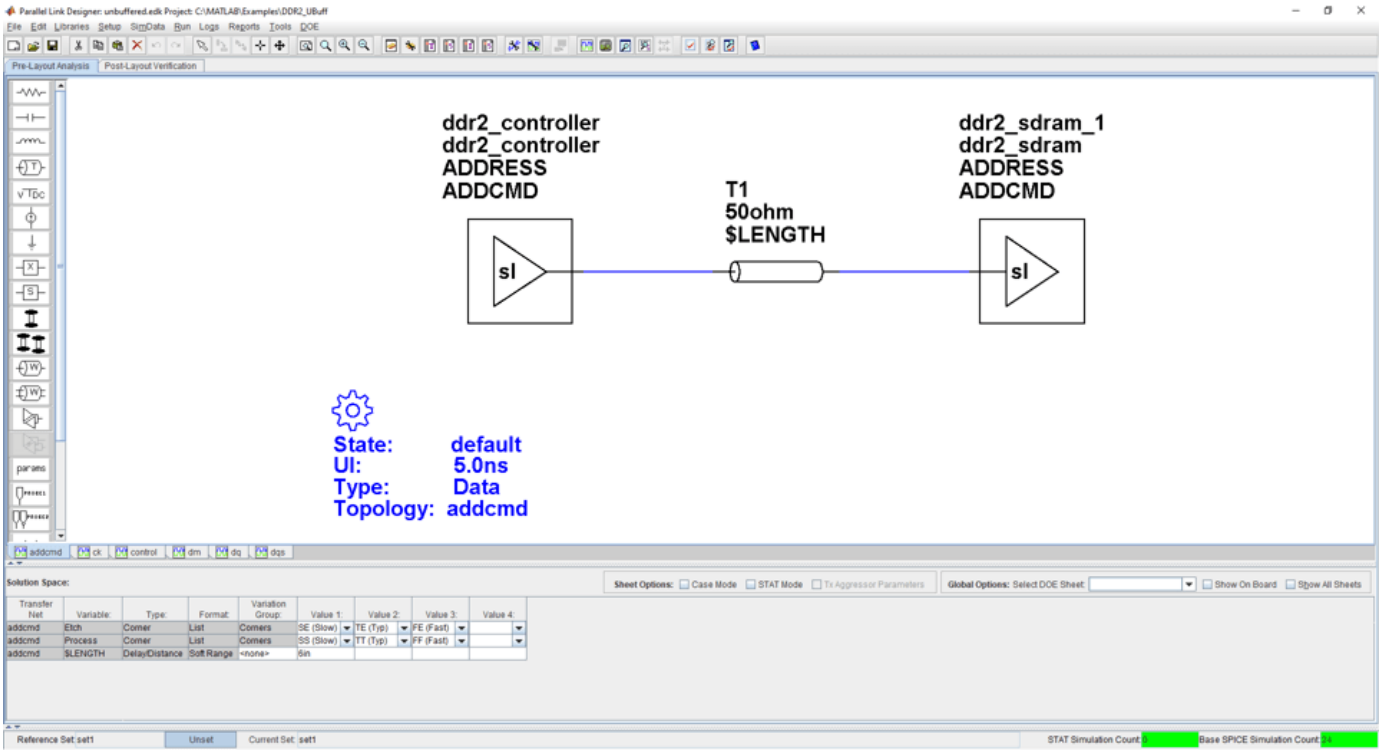
Implement a registered DDR2 interface for pre-layout analysis or post-layout verification.

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");
```



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

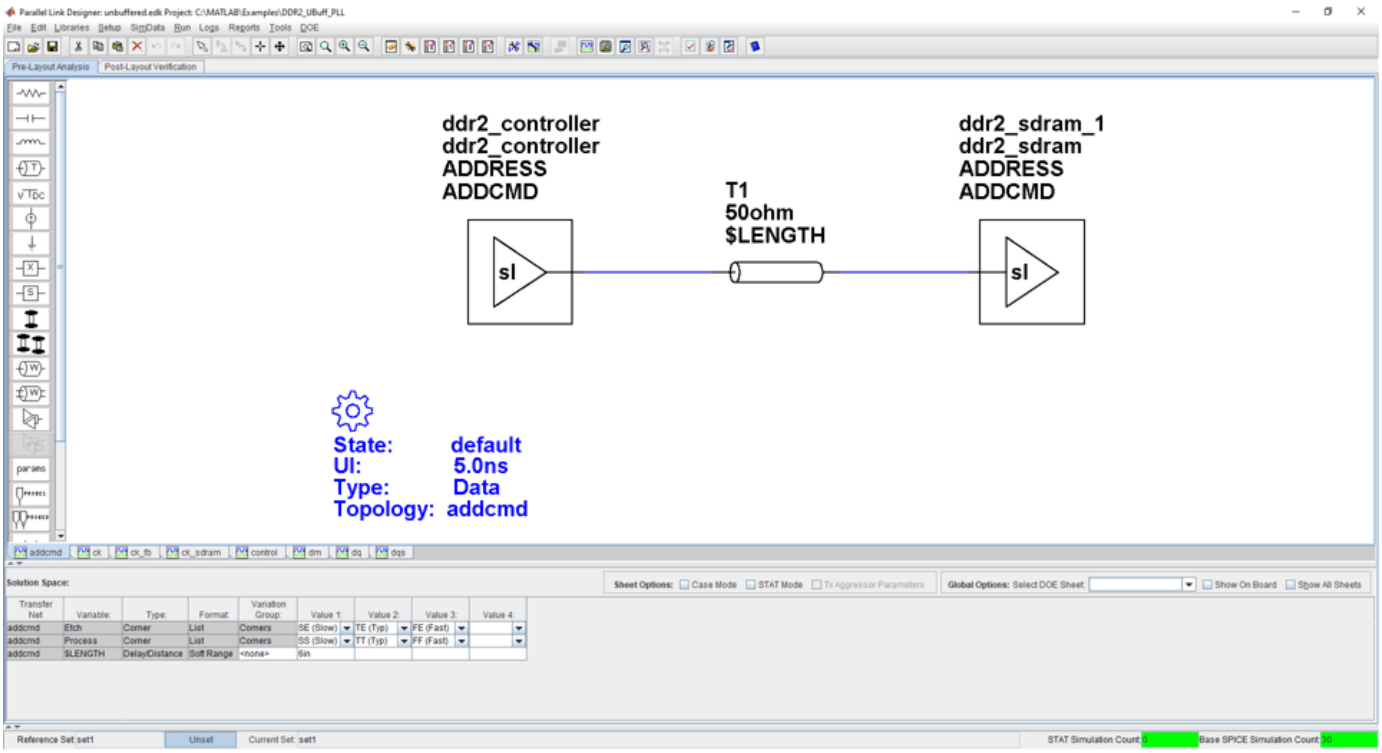
Implement an unbuffered DDR2 interface with PLL clock buffer for pre-layout analysis or post-layout verification.

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");
```



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

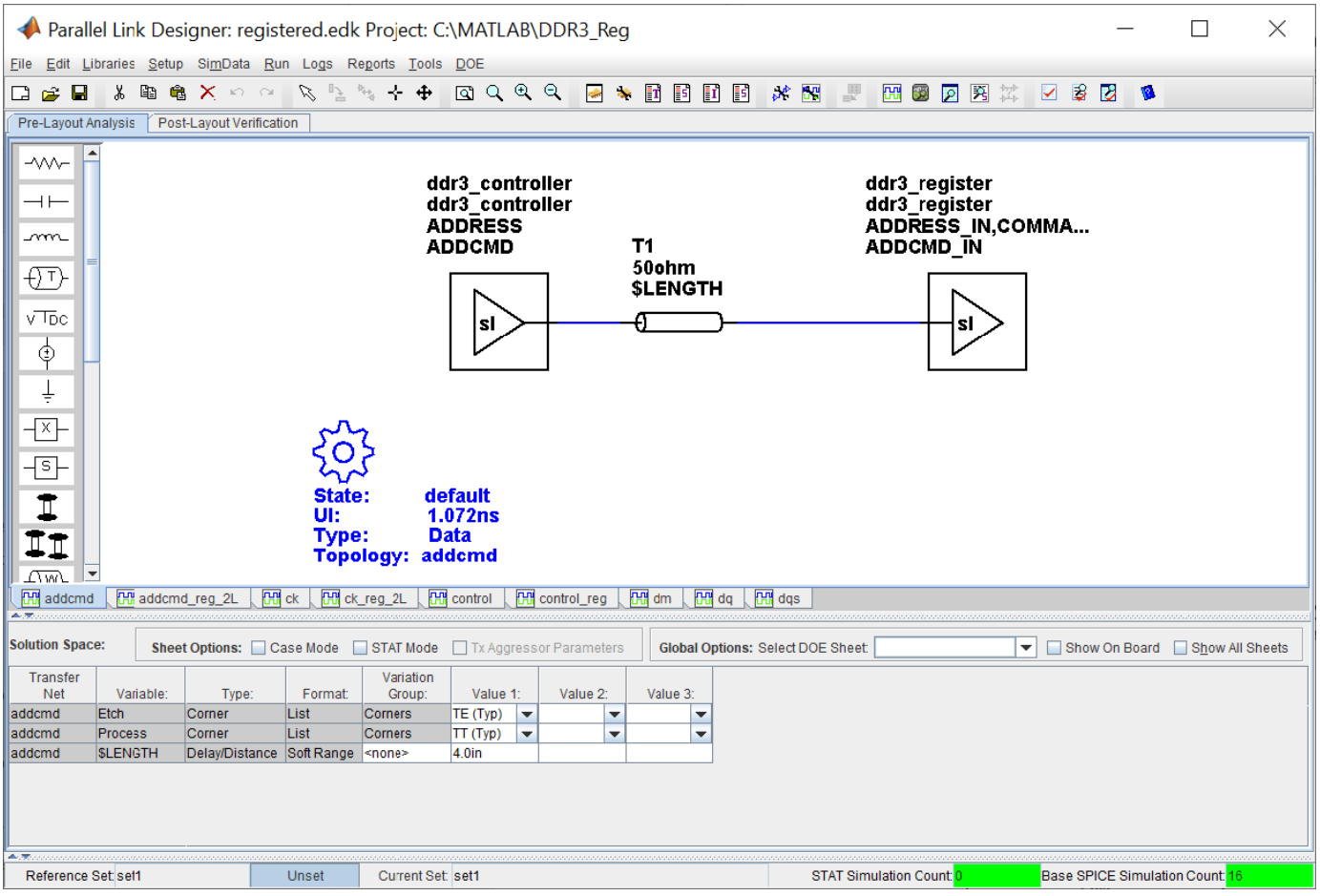
Implement a Registered DDR3 interface for pre-layout analysis or post-layout verification.

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");
```



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

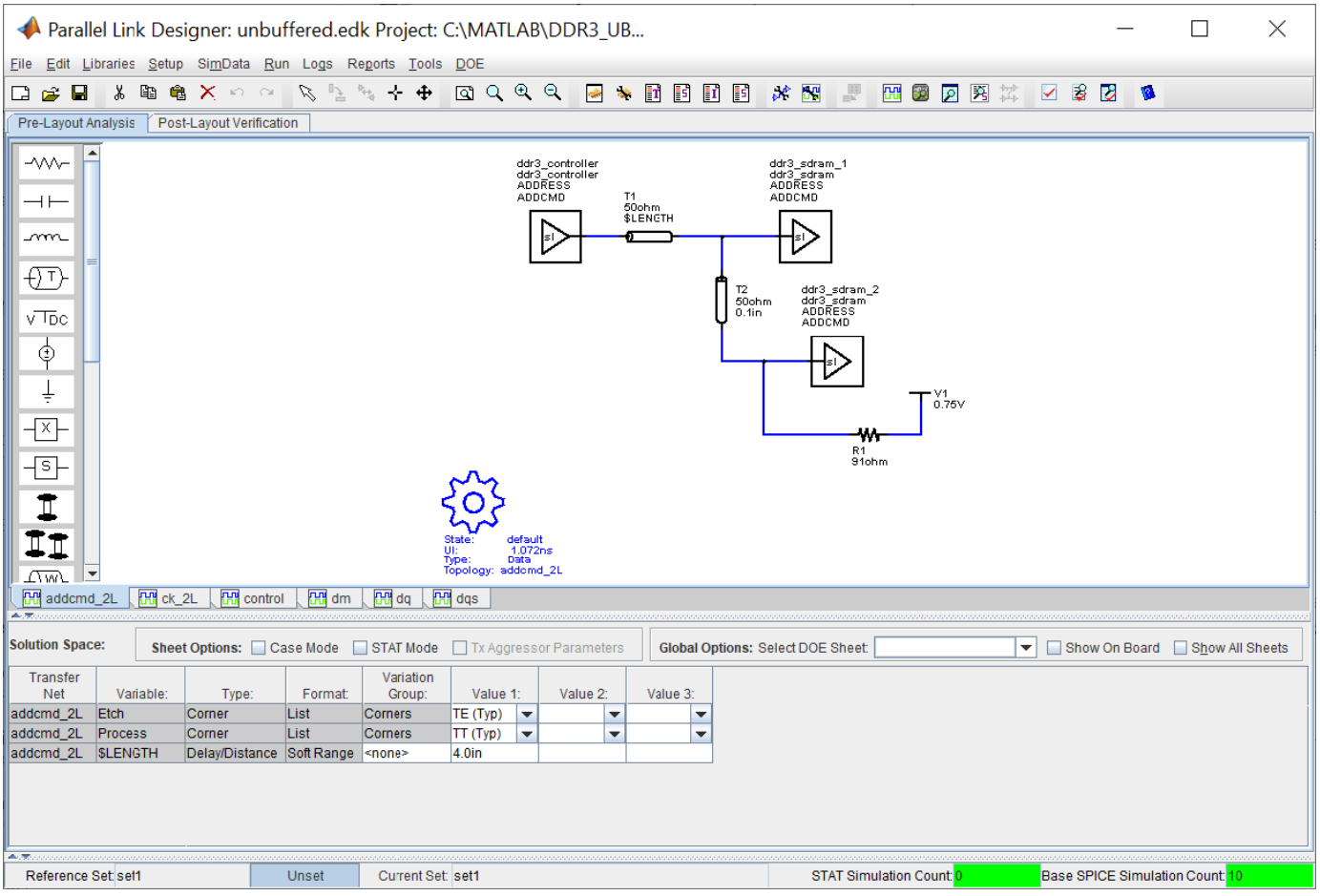
Implement an unbuffered DDR3 interface for pre-layout analysis or post-layout verification.

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");
```



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

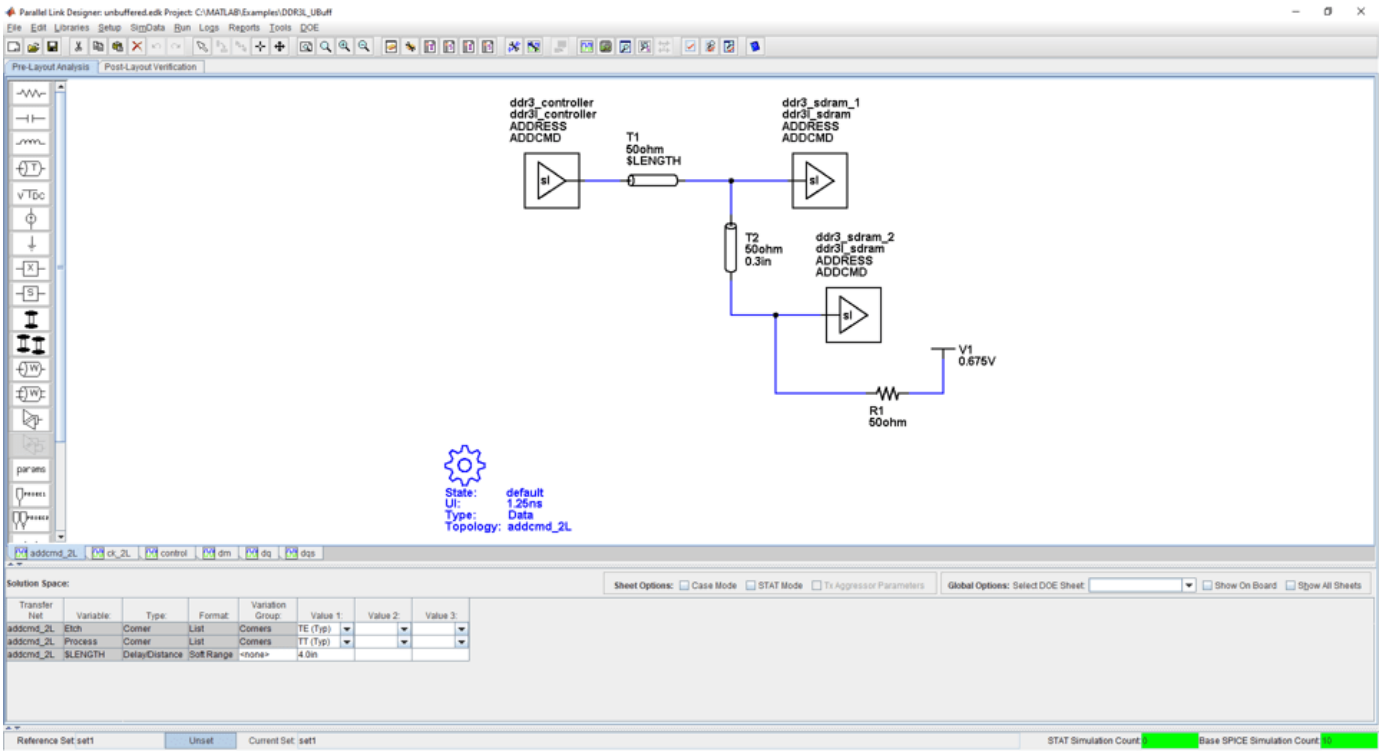
Implement an unbuffered DDR3L interface for pre-layout analysis or post-layout verification.

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");
```



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

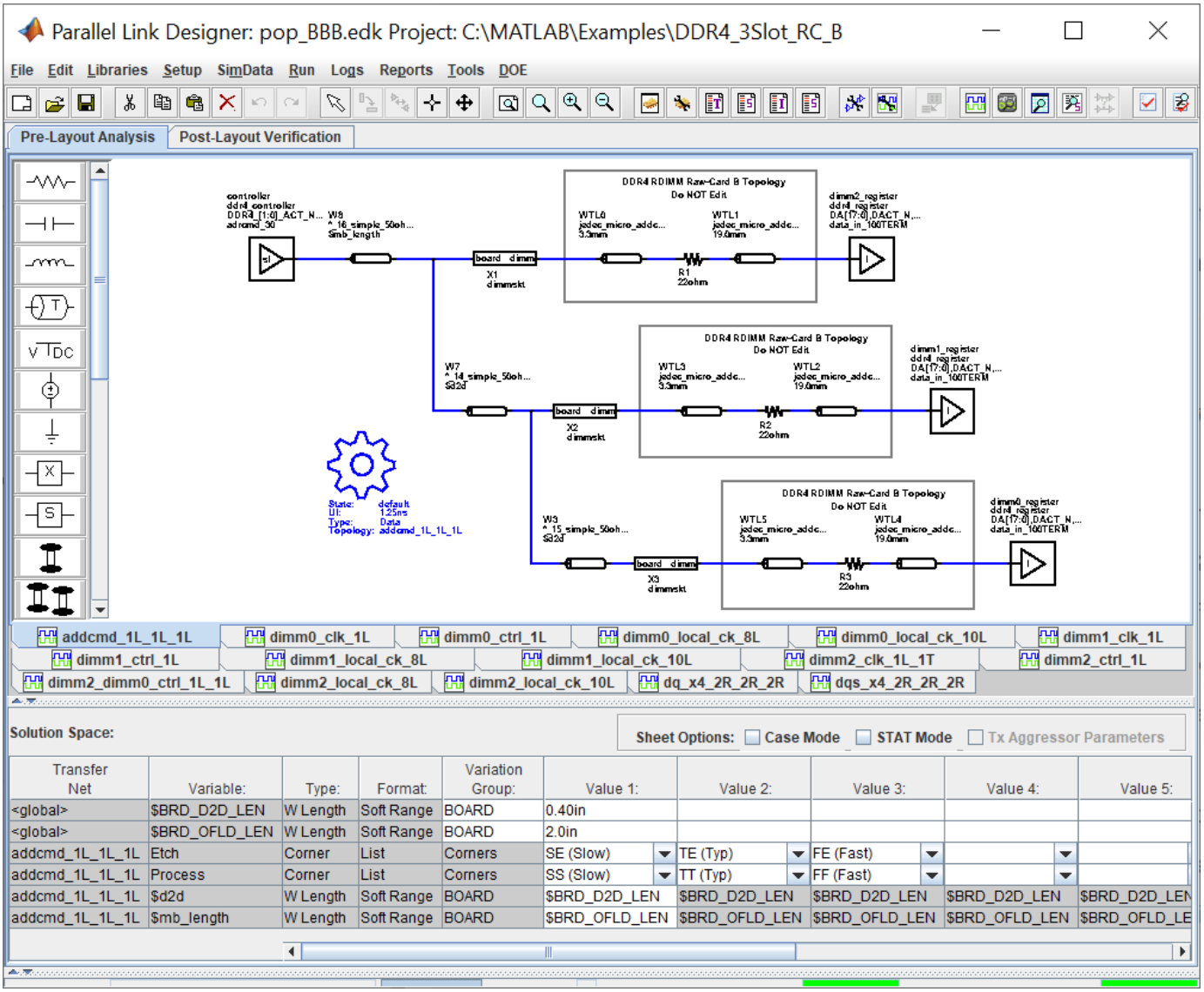
Implement a 3-slot DDR4 Raw Card B RDIMM interface for pre-layout analysis or post-layout verification.

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");
```



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.

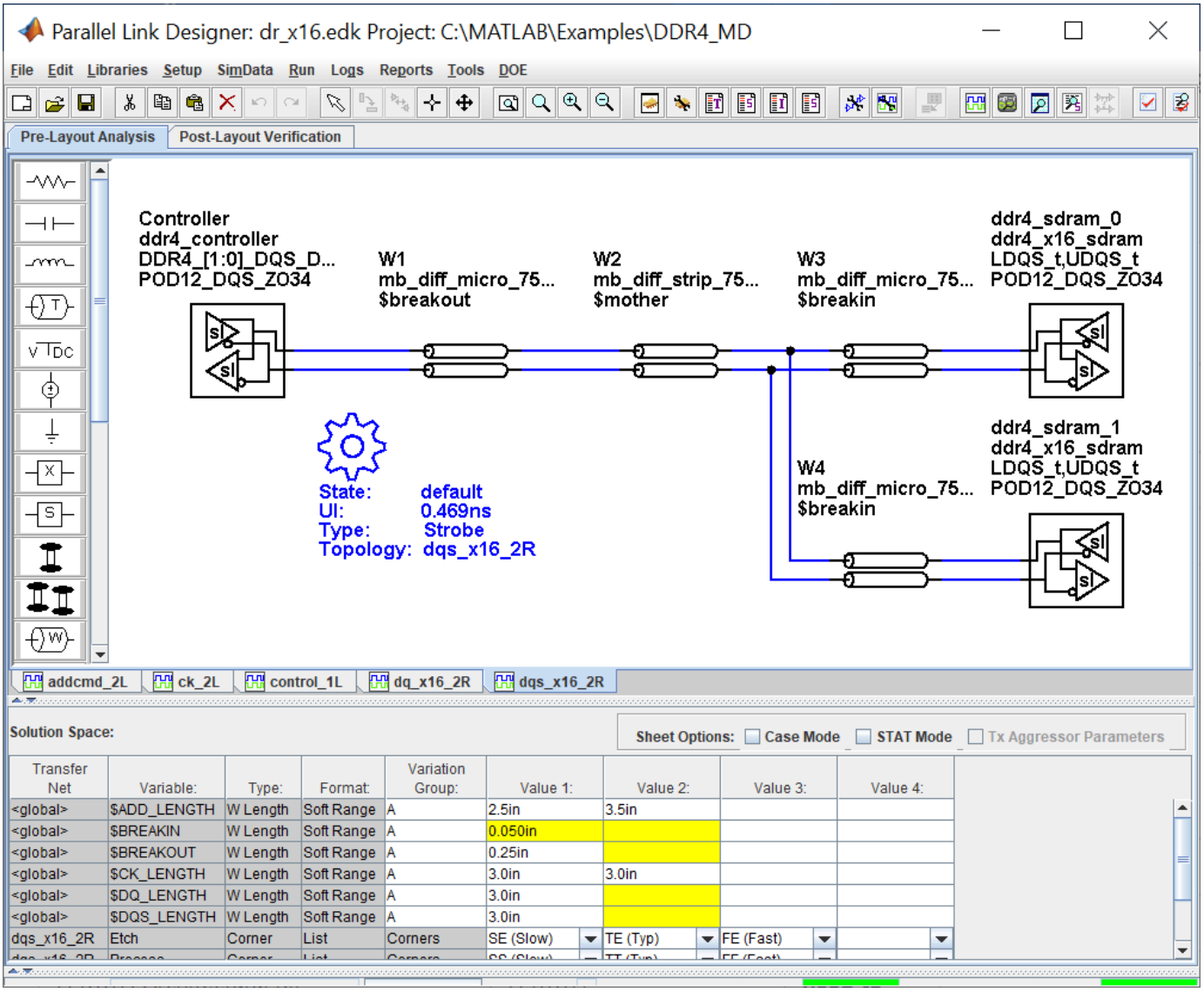
This DDR4 MD implementation signal integrity kit includes all the transfer nets, timing models, waveform processing levels, and simulation models for both single and dual rank memory down (discrete) configurations. This includes buffer models for the DDR4 memory controller as well as Micron SDRAMs. Also 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");
```



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.

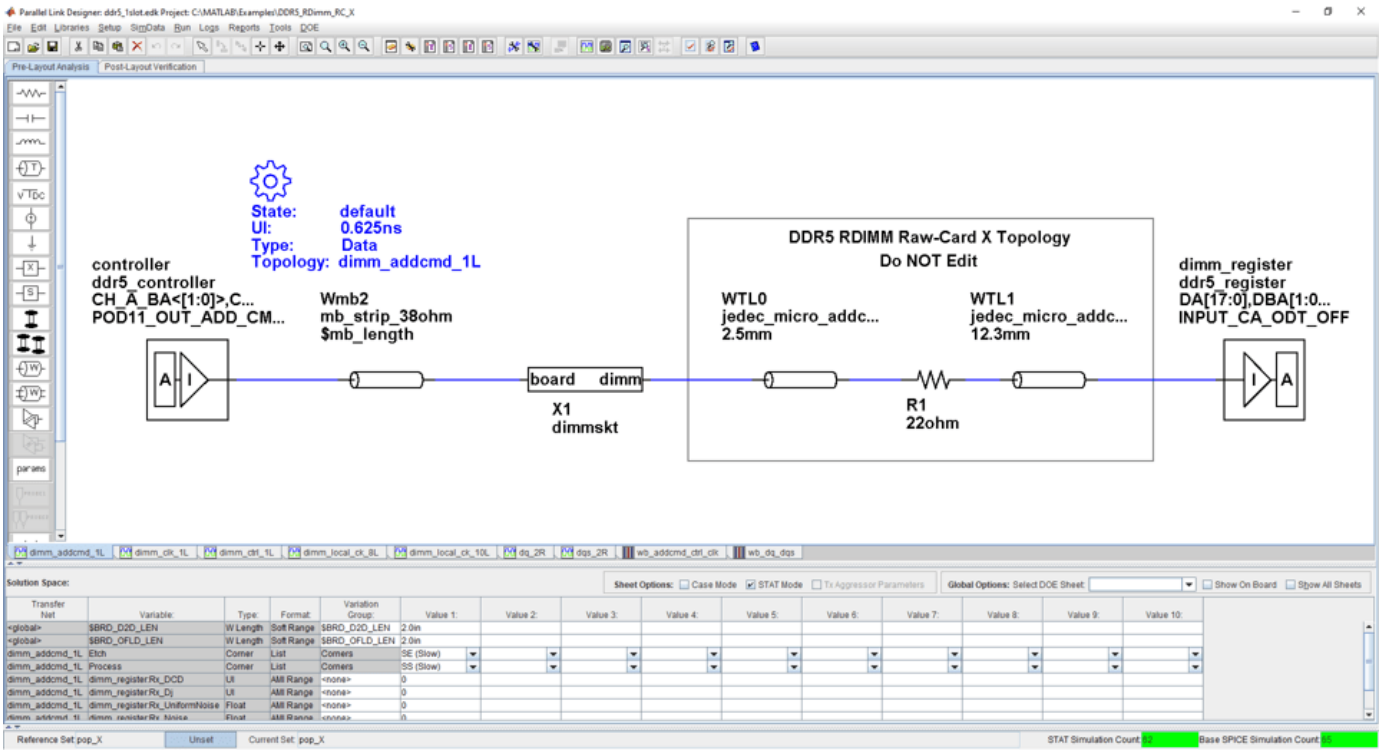
DDR5 is an industry standard dynamic memory format operating at a maximum of 6400M transfers per second. The standard is defined by JEDEC in the DDR5 JEDEC Specification JESD79-5.

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");
```



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.

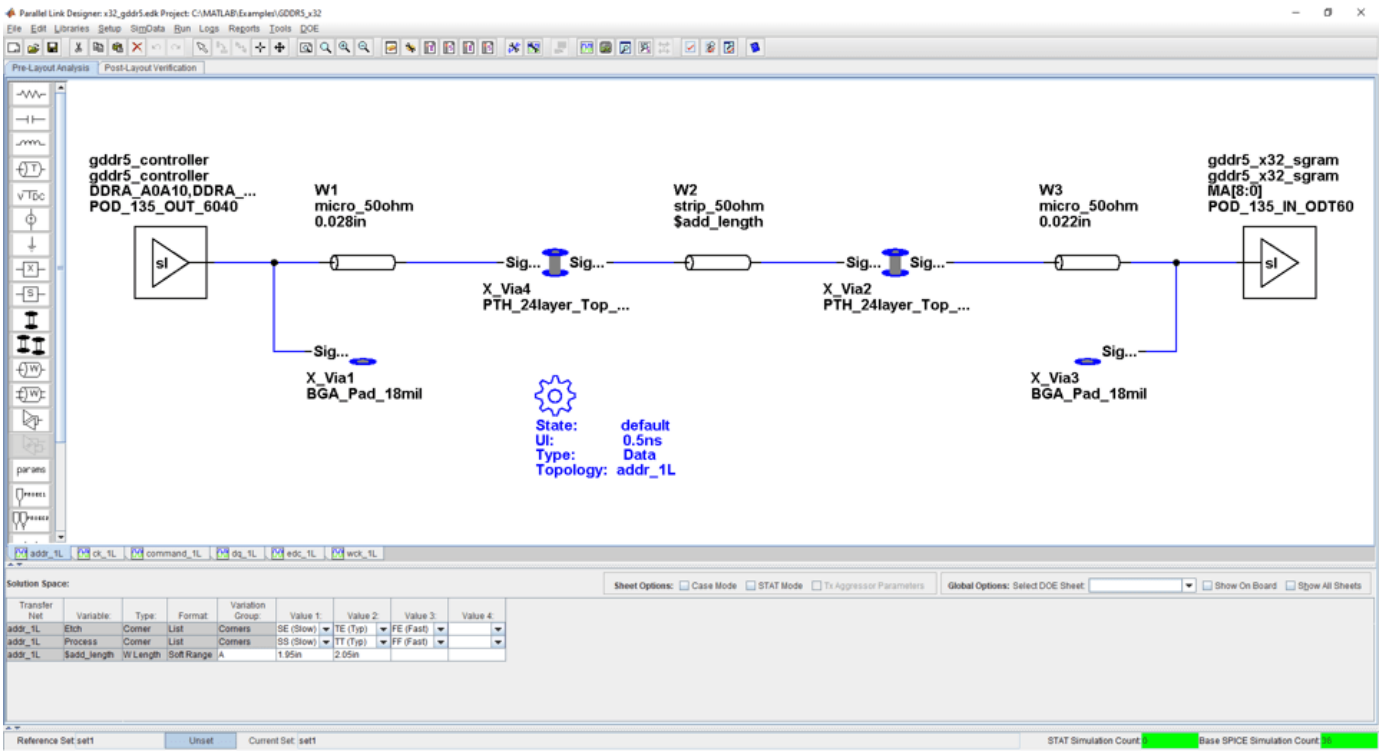
This GDDR5 implementation signal integrity kit includes all the transfer nets, timing models, waveform processing levels and simulation models for a GDDR5 x32 memory down interface. This includes buffer models for a generic GDDR5 controller and Micron x32 8 Gb SGRAM, along with timing models and complete 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");
```



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.

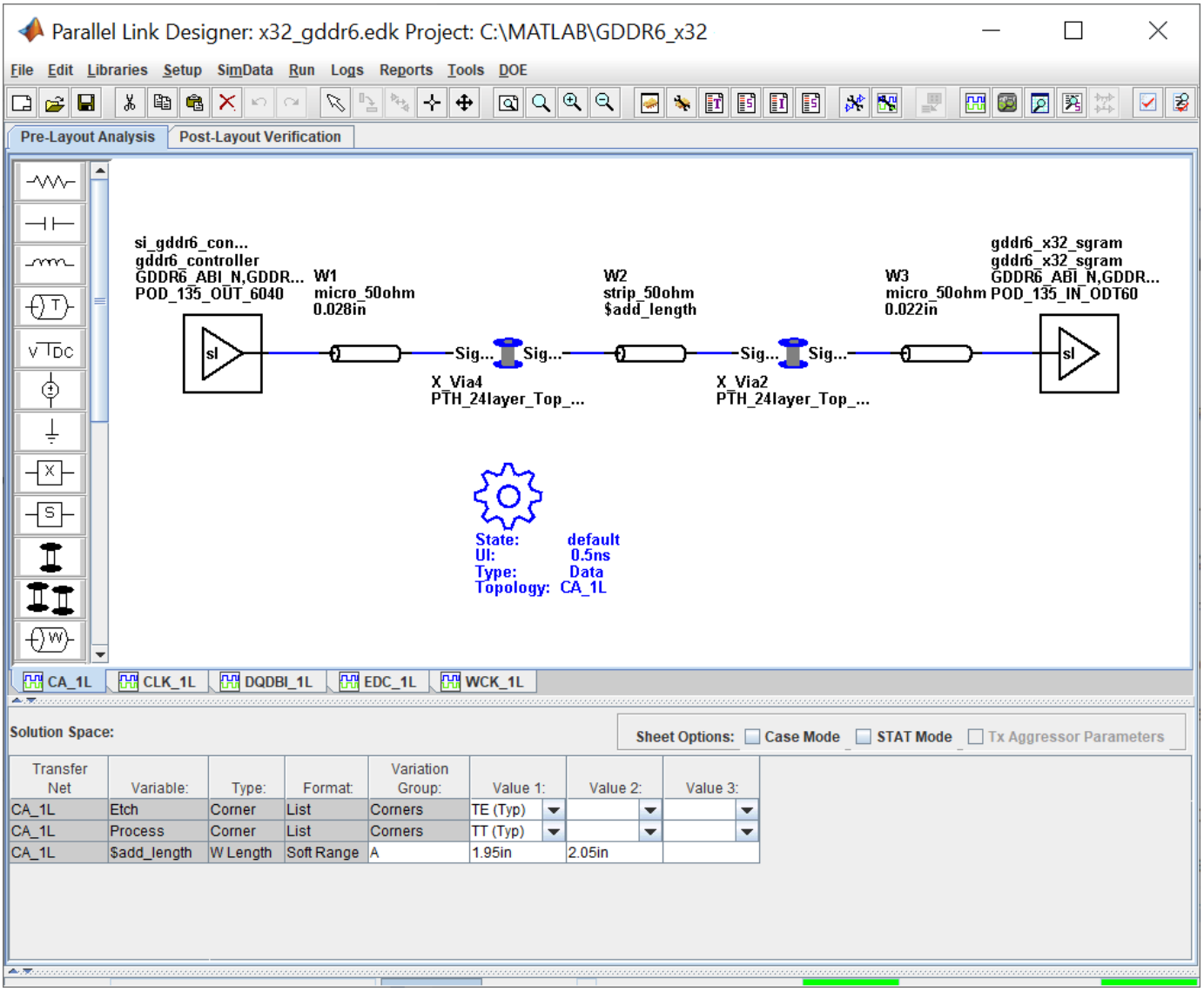
This GDDR6 architectural signal integrity kit includes all the transfer nets, waveform processing levels, generic timing and simulation models for a GDDR6 interface. This includes generic buffer models for the GDDR6 controller and SGRAM, along with functional timing models and complete waveform processing 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");
```



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

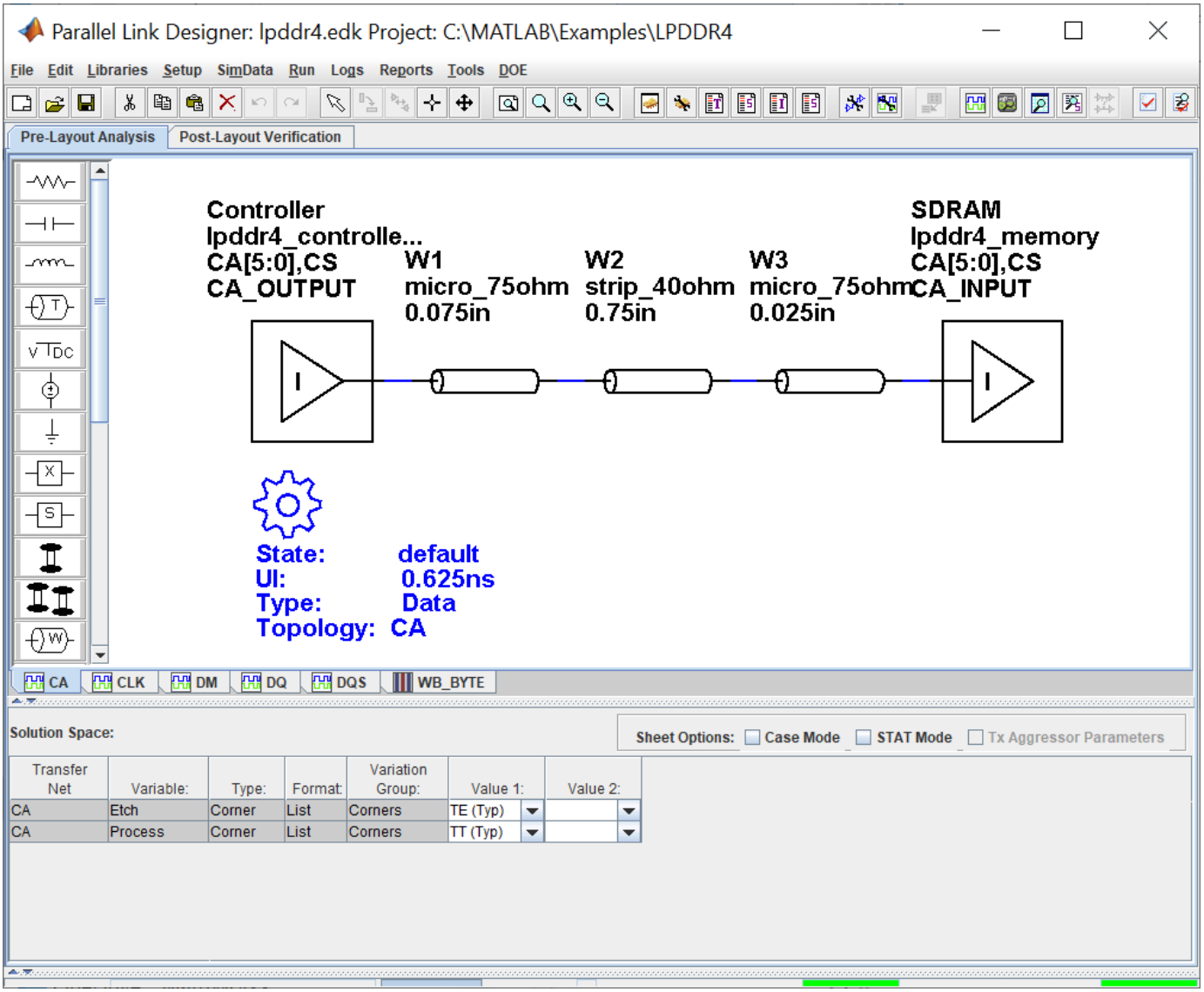
Implement a low-power DDR4 (LPDDR4) interface for pre-layout analysis or post-layout verification.

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");
```



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

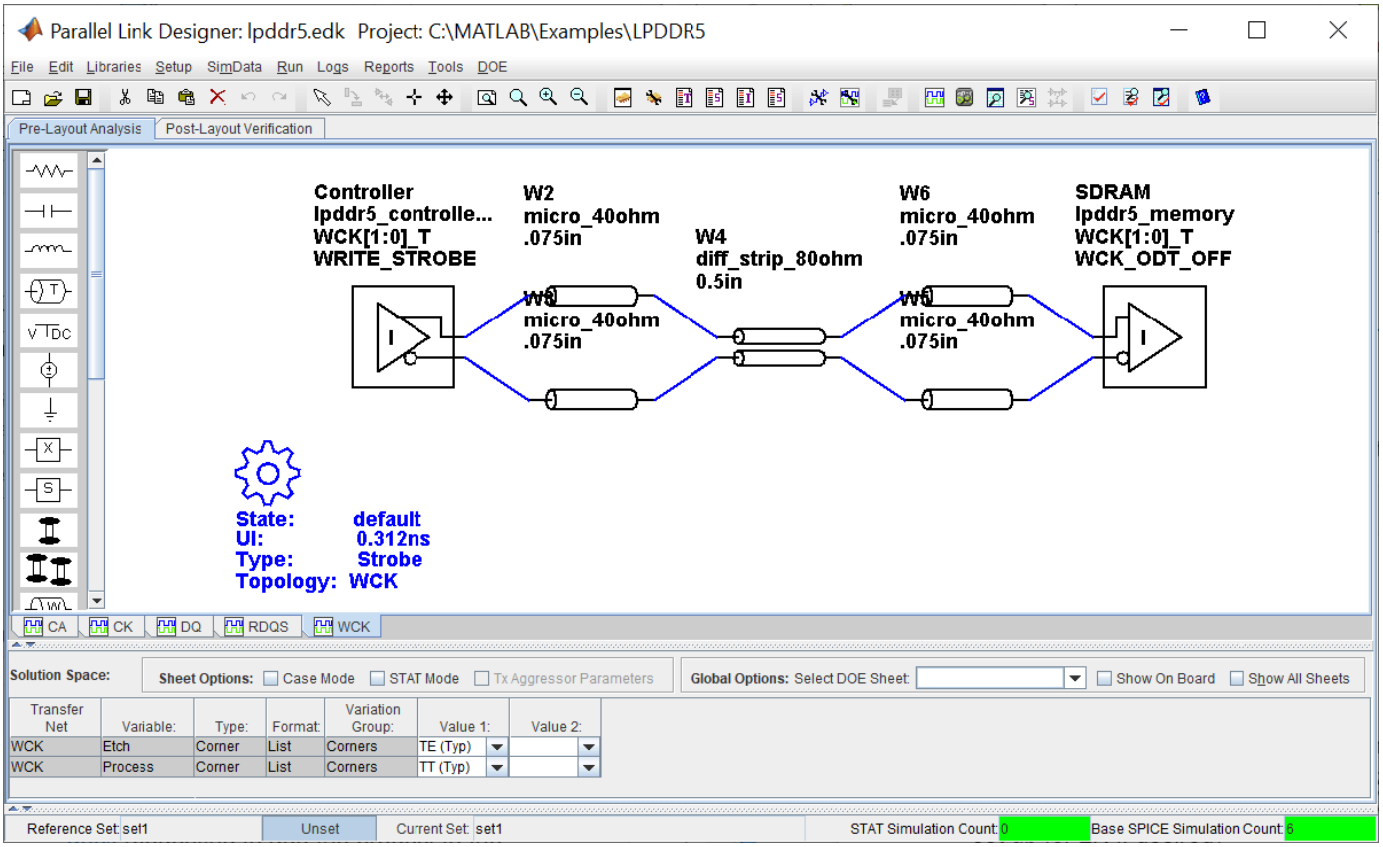
Implement a low-power DDR5 (LPDDR5) interface for pre-layout analysis or post-layout verification.

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");
```



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

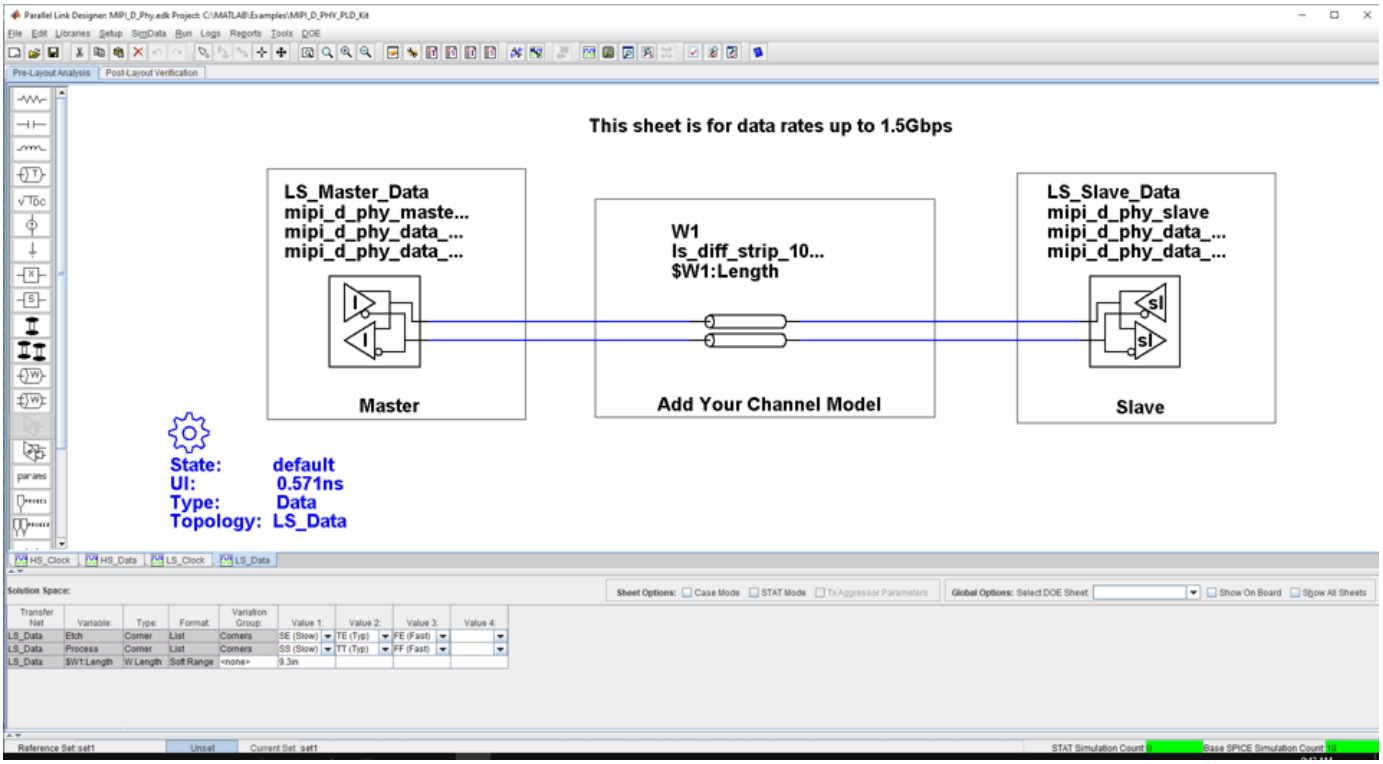
Test the compliance to the MIPI D-PHY specification with respect to clock-to-data timing in the forward direction and waveform quality in the reverse transmission using **Parallel Link Designer**.

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");
```



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

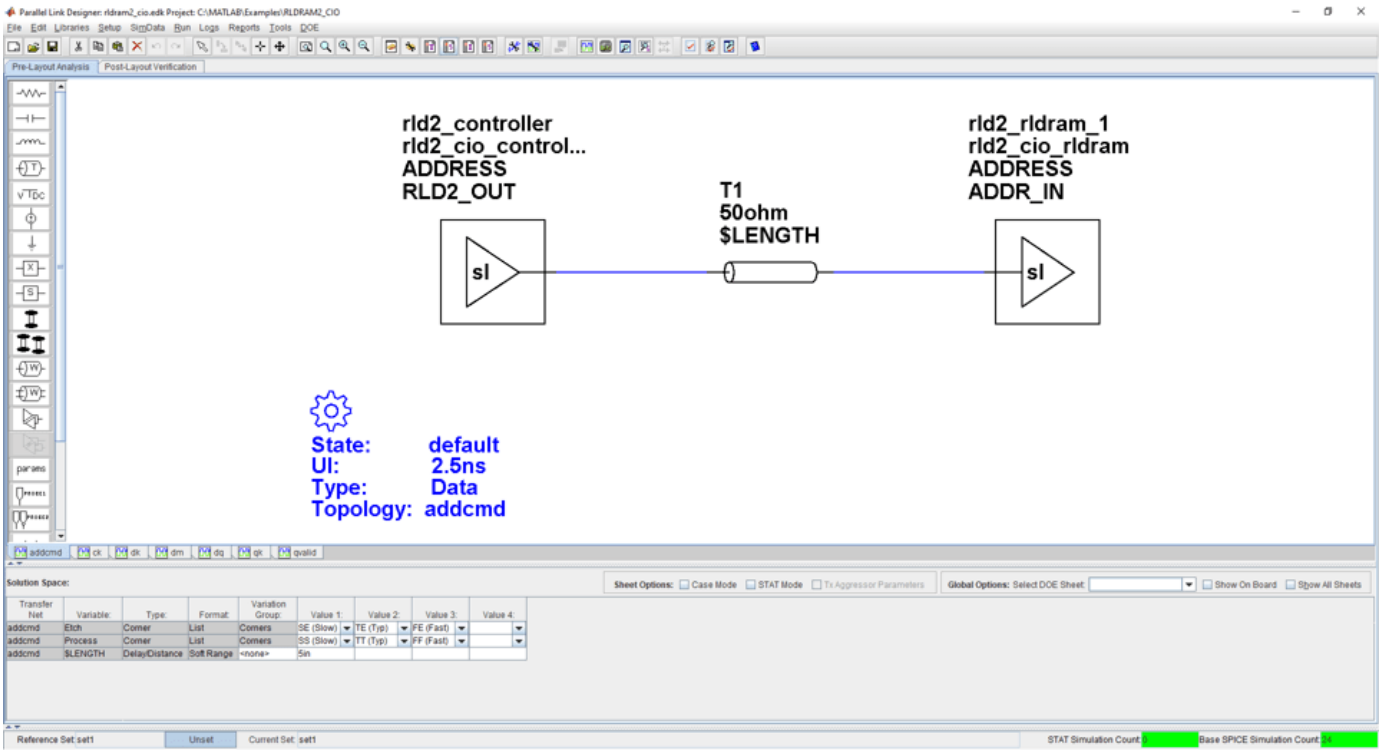
Implement a common I/O (CIO) RLDRAM II interface for pre-layout analysis or post-layout verification.

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");
```



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

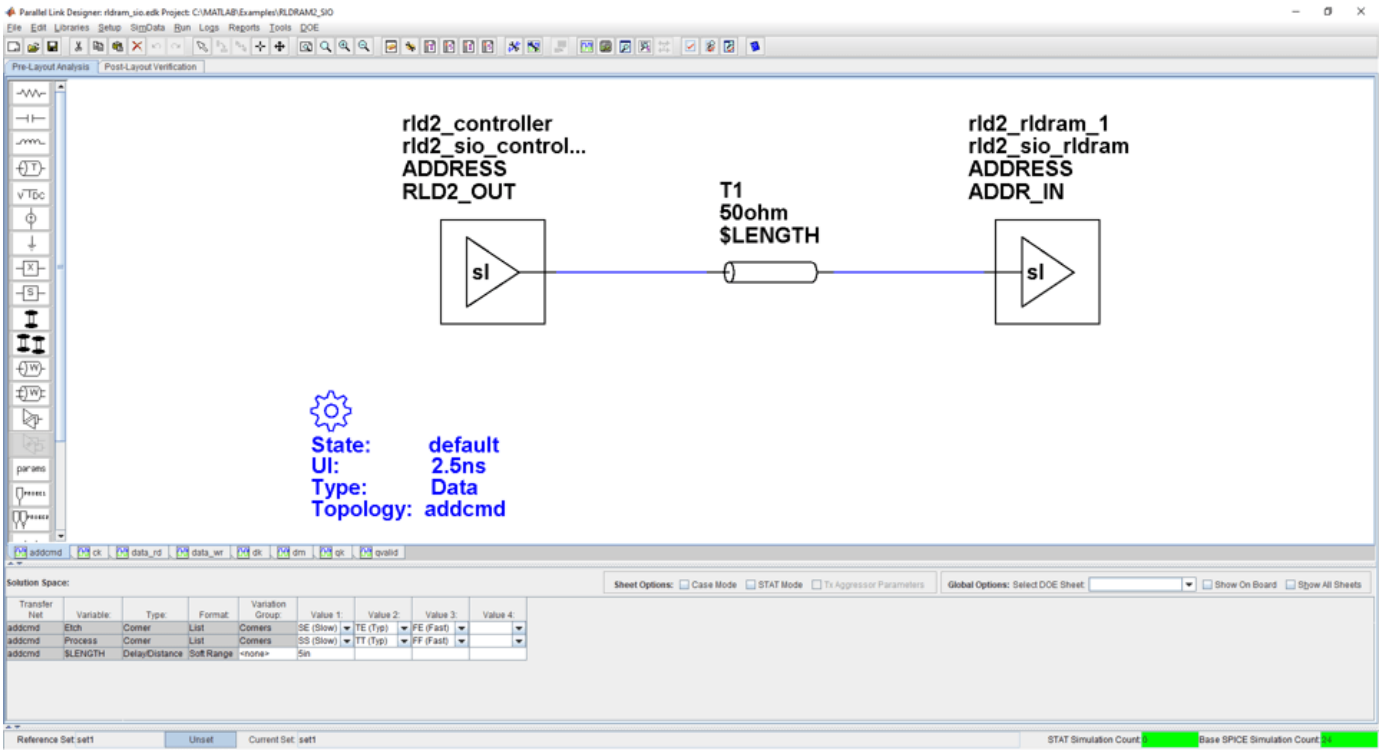
This example shows how to implement a separate I/O (SIO) RLDRAM II interface for pre-layout analysis or post-layout verification.

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");
```



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

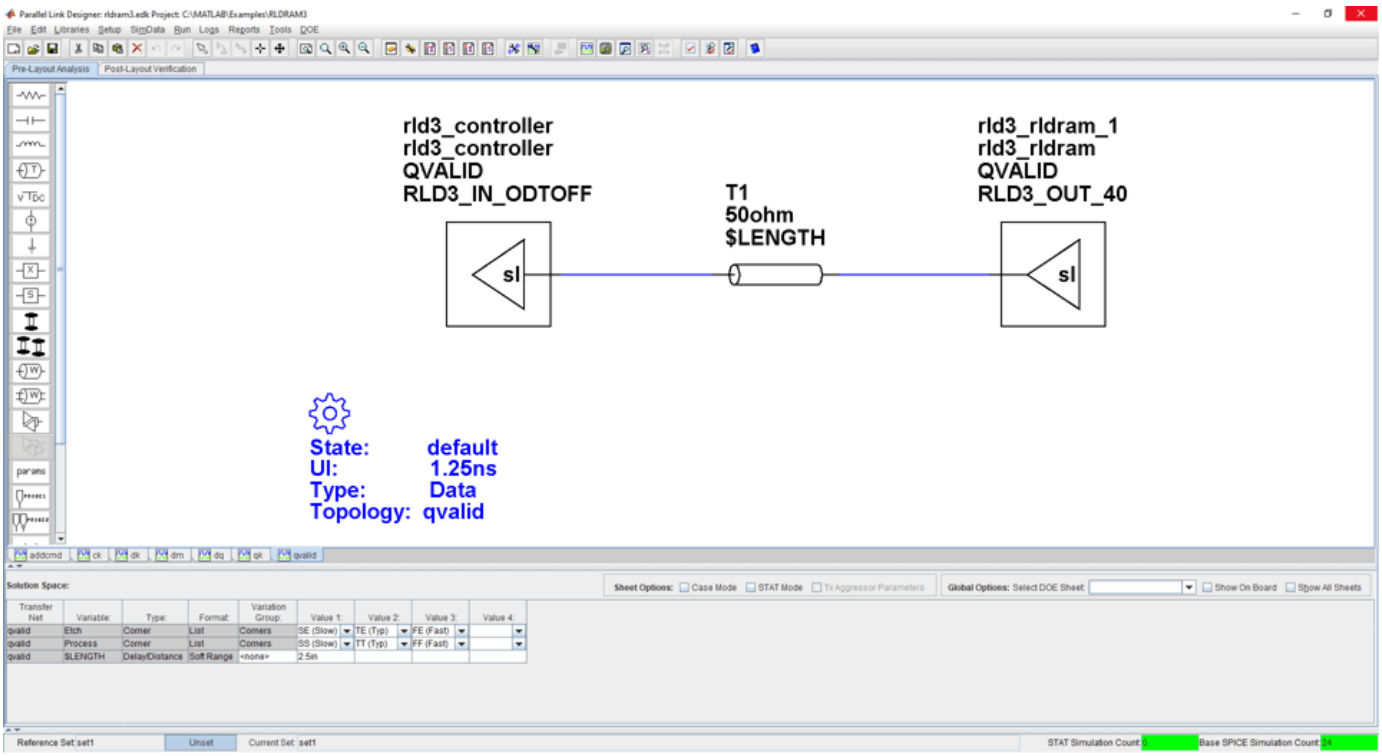
Implement a RLDRAM III interface for pre-layout analysis or post-layout verification.

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");
```



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:

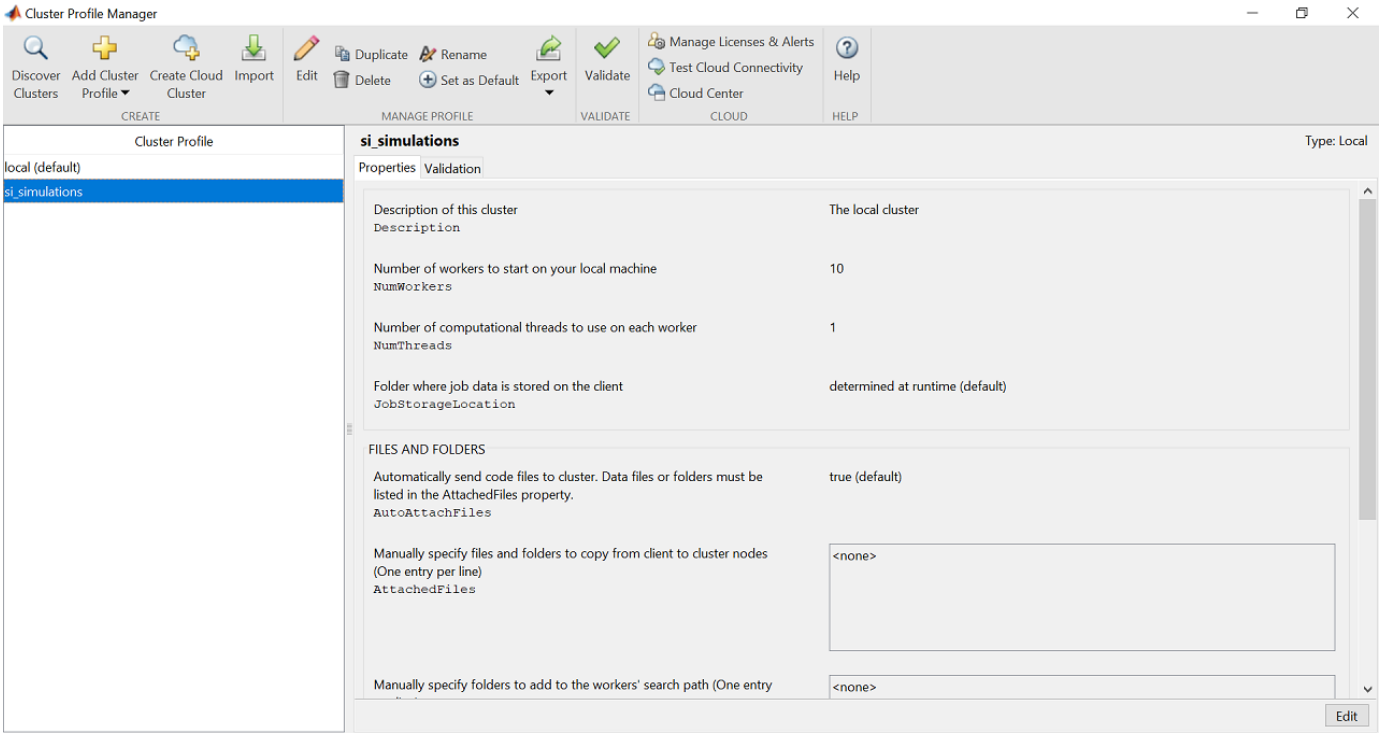
- Task: A list of operations. In Signal Integrity Toolbox™, these operations are individual simulations. Each task can consist of one or multiple simulations run sequentially.
- Cluster: The location where tasks will be performed. A cluster can be a single machine that can execute multiple threads simultaneously, 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.



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.

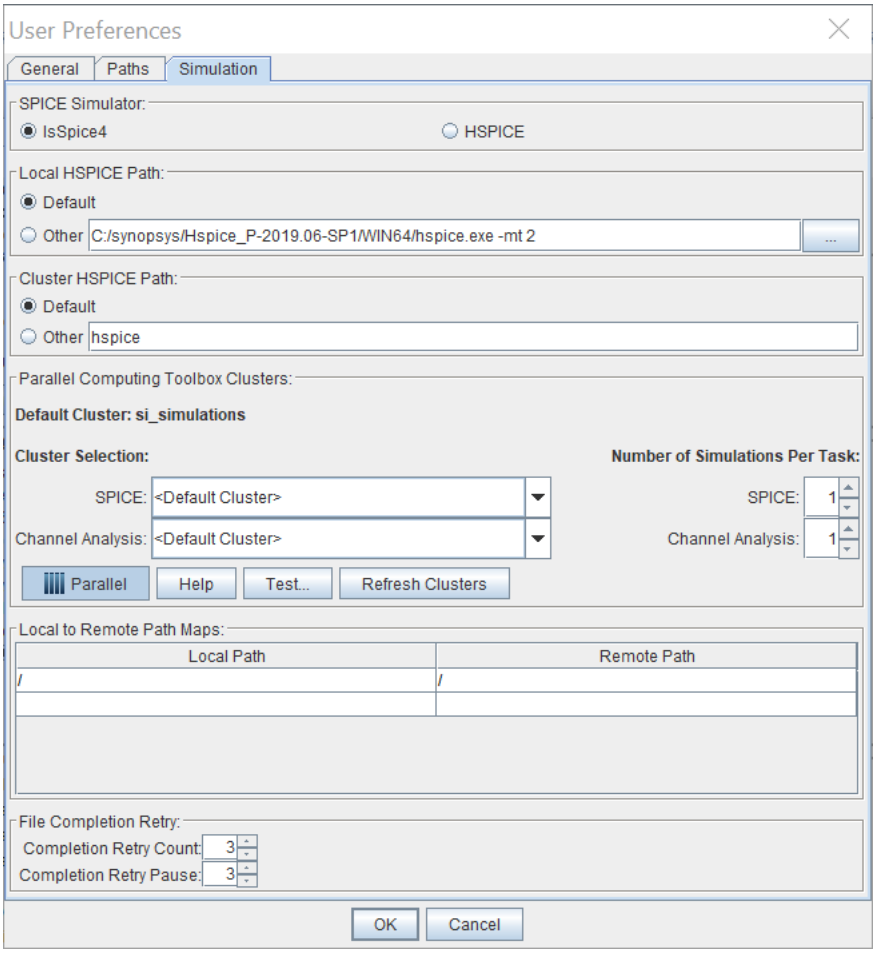
   When running on a remote cluster, use a similar approach. However, if the remote machine is shared across multiple users, reduce the number of workers to allow good performance for all users. You may need to adjust this number in your cluster based on the memory and CPU demands on your cluster and the 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

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.

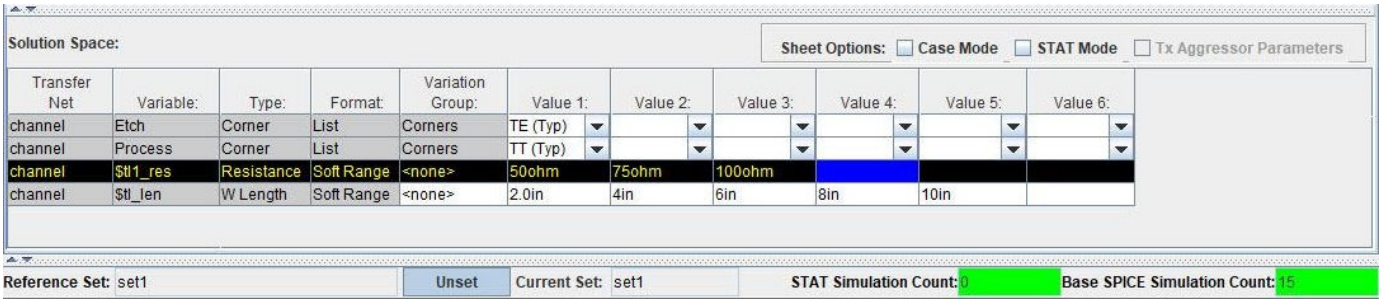
File Completion Retry

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.

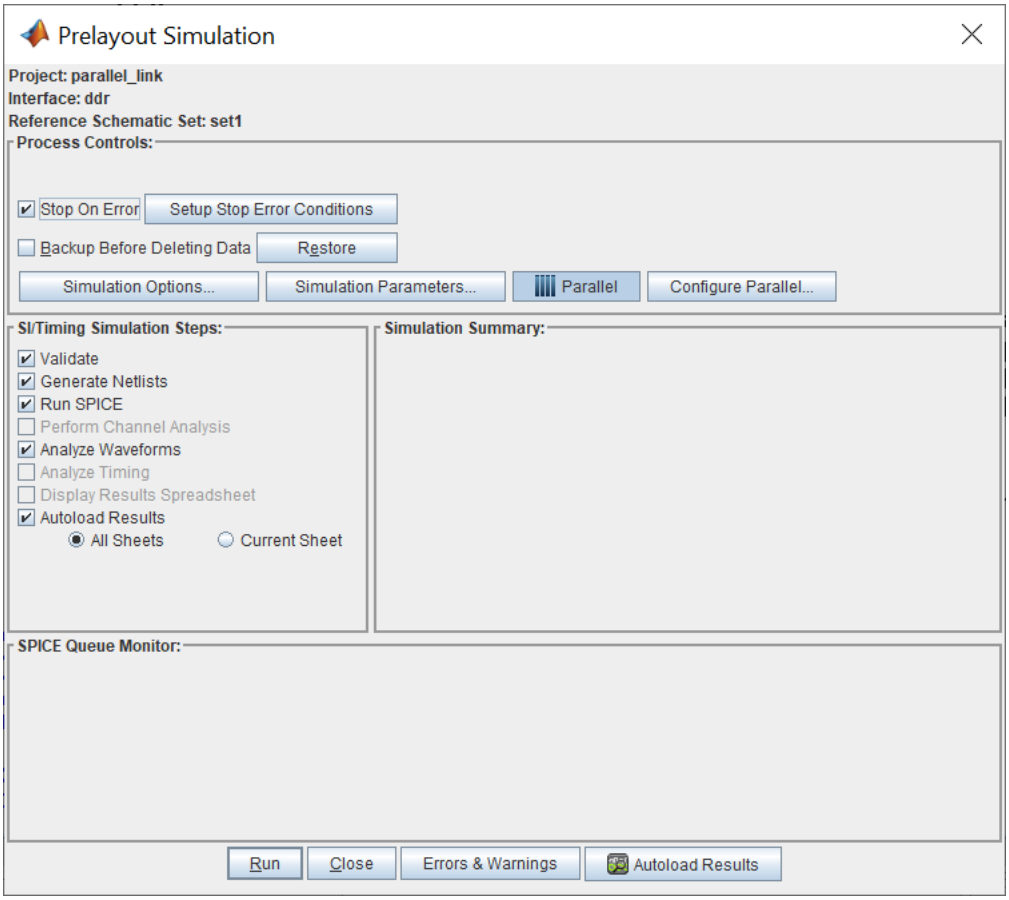
In some remote server and remote disk environments, the simulation output files are not completely written to disk when the simulation completes. If you see simulation errors when simulations appear to have run to completion, try increasing the value of these two parameters.

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.



To run the simulations, select **Run > Simulated Selected**. The Prelayout Simulation dialog box opens.

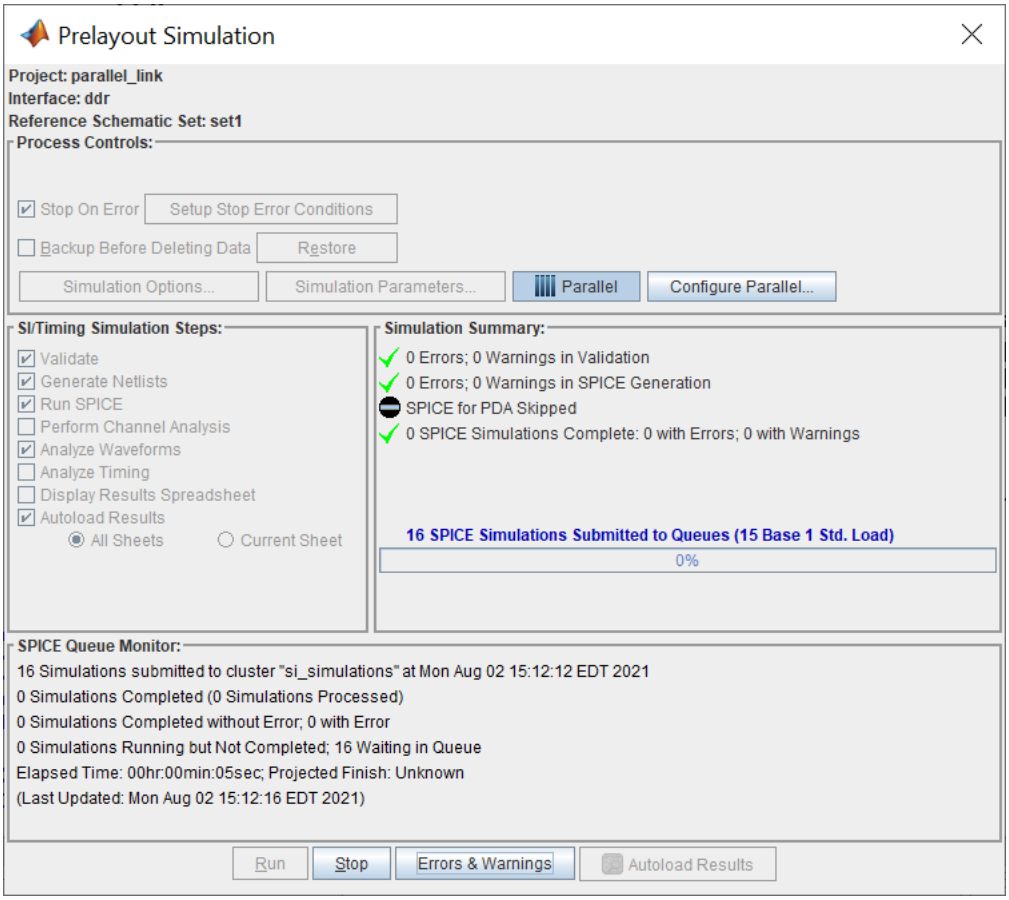


If you have Parallel Computing Toolbox, then the **Parallel** button should already be selected. Toggle this selection to enable and disable the parallel simulations. With the **Parallel** button enabled, launch the simulations by clicking **Run**. There is a short delay while the parallel pool starts.

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



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