

Signal Integrity Toolbox™

Reference



MATLAB®

R2023a



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™ Reference

© COPYRIGHT 2021–2023 by The MathWorks, Inc.

The software described in this document is furnished under a license agreement. The software may be used or copied only under the terms of the license agreement. No part of this manual may be photocopied or reproduced in any form without prior written consent from The MathWorks, Inc.

FEDERAL ACQUISITION: This provision applies to all acquisitions of the Program and Documentation by, for, or through the federal government of the United States. By accepting delivery of the Program or Documentation, the government hereby agrees that this software or documentation qualifies as commercial computer software or commercial computer software documentation as such terms are used or defined in FAR 12.212, DFARS Part 227.72, and DFARS 252.227-7014. Accordingly, the terms and conditions of this Agreement and only those rights specified in this Agreement, shall pertain to and govern the use, modification, reproduction, release, performance, display, and disclosure of the Program and Documentation by the federal government (or other entity acquiring for or through the federal government) and shall supersede any conflicting contractual terms or conditions. If this License fails to meet the government's needs or is inconsistent in any respect with federal procurement law, the government agrees to return the Program and Documentation, unused, to The MathWorks, Inc.

Trademarks

MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See www.mathworks.com/trademarks for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.

Patents

MathWorks products are protected by one or more U.S. patents. Please see www.mathworks.com/patents for more information.

Revision History

September 2021	Online only	New for Version 1.0 (Release 2021b)
March 2022	Online only	Revised for Version 1.1 (Release 2022a)
September 2022	Online only	Revised for Version 1.2 (Release 2022b)
March 2023	Online only	Revised for Version 1.3 (Release 2023a)

1 | _____ Apps

2 | _____ Class

Apps

Parallel Link Designer

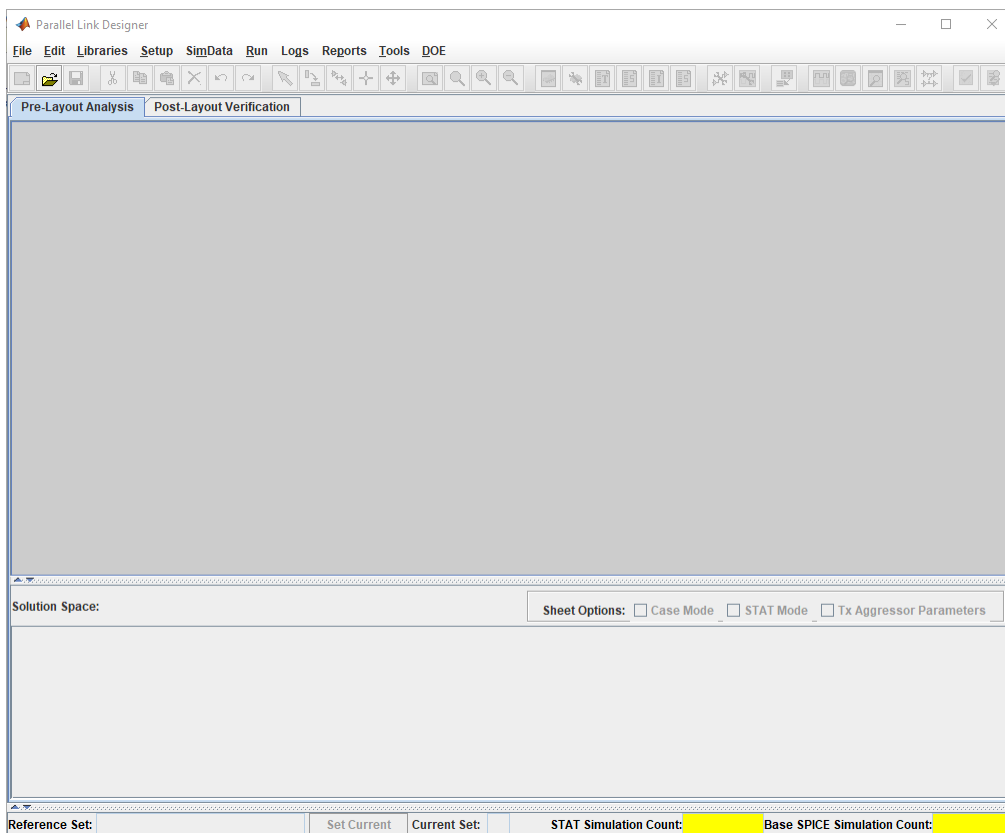
Analyze PCB designs for parallel link applications

Description

The **Parallel Link Designer** analyzes PCB designs for parallel link applications for both pre-layout and post-layout simulations. To perform post-layout analysis, you need a license for RF PCB Toolbox™.

Using pre-layout analysis, you can determine system-level noise and timing margins from integrated signal integrity, waveform, timing, and crosstalk analysis. The pre-layout analysis environment generates design guidelines for your board layouts, package layouts, connectors, and cabling. The post layout environment verifies the actual layout against the design guidelines.

Note Signal Integrity Toolbox™ recommends you to set the Java Heap Memory to at least 8192 MB when using the **Parallel Link Designer** app.



Open the Parallel Link Designer App

- MATLAB® Toolstrip: In the **Apps** tab, under **Signal Processing and Communications**, click the app icon.
- MATLAB command prompt: Enter `parallelLinkDesigner`.

Examples

- “Analyze Parallel Links with Parallel Link Designer”
- “Configure DDR Controller with Two Memory Designators”

Programmatic Use

`parallelLinkDesigner` opens a new session of the **Parallel Link Designer** app, enabling you to design and analyze a parallel link.

`parallelLinkDesigner(path/file.edk)` opens the interface designed by the file.

`parallelLinkDesigner(file.script)` runs a script file and returns the app process handle.

More About

Simulation Parameters

Simulation Parameters control how the analysis is run in the **Parallel Link Designer** app. For detailed information about them, see “Simulation Parameters Used in Parallel Link Design”.

Corner Conditions

There are two types of corner conditions you can vary using the **Parallel Link Designer** app.

- IC Environment Corners — These are the temperature parameter for each corner. They do not affect IBIS buffer models.
- Etch Corners — These are the scaling factors for the Z_0 and T_{pd} parameters of transmission line models. Both ideal and lossy transmission line models are scaled. Lossy transmission line models are scaled by computing the values of Z_0 and T_{pd} from the typical corner L and C values.

For more information, see “Specify Corner Conditions in Parallel Link Design”.

Stimulus Patterns

You can specify the stimulus patterns for the time domain analysis. 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**. For more information, see “Stimulus Patterns in Parallel Link Design”.

Widebus (Pre-Layout Crosstalk Analysis)

In **Parallel Link Designer** app, pre-layout crosstalk analysis is referred to as widebus crosstalk analysis. It consists of single transfer sheets (transfer nets) and a “widebus” or coupled sheet with a victim transfer net and one or more aggressor nets.

Each unique transfer net on a widebus schematic sheet is known as a “widebus group”. That group consists of ONLY the transfer net designators (drivers, receivers or I/O buffers). On a widebus sheet any widebus group can be a victim or aggressor and multiple instances of any widebus group can be placed on the schematic. The widebus groups inherit the designator properties defined in their transfer net sheet. Coupled w-lines, connectors, package models or s-parameter blocks can then be used in the widebus schematic to connect up each widebus group. One of the instances of a widebus group will be designated as the victim net on the schematic.

Widebus analysis can be performed on any project, even previous incarnations of a design. The following requirements are:

- One or more reference transfer nets (sheets) exist. The designator grouping on each unique transfer net are referred to as a “widebus group”.
- Each transfer nets consist of a single driver per transfer (they can have multiple transfers (i.e. memory read/memory write), and can have multiple receivers/loads.
- Transfer net characteristics such as transfers, UI, type, probe points and jitter are defined in the transfer net characteristics (not on a widebus sheet).

To create a widebus sheet there must be at least one transfer net included in the schematic set. Select **File > Schematic Sheet > New Sheet** from the app toolbar. In the newly opened New Sheet dialog box, select **Widebus** and select the widebus groups to be added to the sheet and the number of instances of each.

Clock Forwarding

The **Parallel Link Designer** app supports the clock forwarding methodology defined in IBIS BIRD 204. This BIRD describes a mechanism by which the clock waveform, or recovered clock times, can be passed to the data AMI receiver model to allow the model maker access to the clock. This enables the receiver model to include clock phase noise, clock tree delays, phase interpolators, etc. IBIS BIRD204 has been approved by the IBIS committee and will be included in the IBIS 7.1.

A serial-link channel uses a CDR (clock data recovery) circuit in the receiver to create a clock signal that is then used to latch each individual data bit coming into the core of an IC. The IBIS-AMI standard was created to model this operation and a CDR is required to be present, either in the receiver model itself (the norm) or in the EDA tool that is running the simulation. By contrast, in a parallel-link channel no CDR is present. Instead, a separate clock (or strobe) signal is sent along with the data signals, which is used to latch sets of data bits at one time. This is known as clock-forwarding.

Clock forwarding is accessed by selection of a clock group in the Widebus Groups dialog box. The clock group identifies the clock and which respective data signals are associated with it. This relationship is used for analyzing clock recovery in the parallel interface.

Any widebus group transfer net that has been set up as a clock or strobe will be considered to be a clock group on the widebus sheet. The clock group drop down menu selection will show all strobe/clock groups on the current widebus sheet, plus “<none>”. Clock transfer nets will default to “<none>”. Data transfer nets will default to first clock or strobe associated sheet. Each data group can be associated with a different clock group which also supports sweeping of victim group.

Version History

Introduced in R2021b

See Also

Apps

SI Viewer

Topics

“Analyze Parallel Links with Parallel Link Designer”

“Configure DDR Controller with Two Memory Designators”

Serial Link Designer

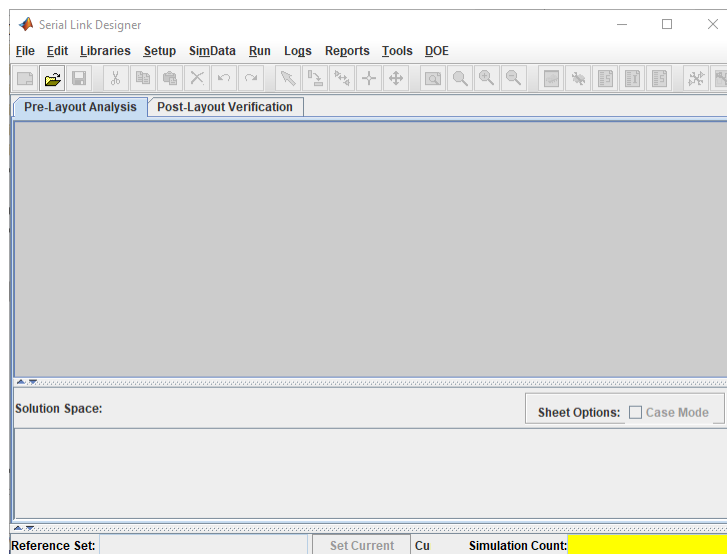
Analyze PCB designs for serial link applications

Description

The **Serial Link Designer** app analyzes PCB designs for serial link applications for both pre-layout and post-layout simulations. To perform post-layout analysis, you need license to RF PCB Toolbox.

Using pre-layout analysis, you can run simulations using IBIS-AMI models with models of your system channels to determine equalization gamut, plot IBIS Statistical and Time Domain eye diagrams at a specified BER, measure channel response characteristics vs. specification masks, as well as perform crosstalk analysis. The pre-layout analysis environment can help you to generate design guidelines for board layouts, package layouts, connectors, and cabling. The post-layout environment verifies the actual layout against design guidelines or specification rules provided in a signal integrity kit.

Note Signal Integrity Toolbox recommends you to set the Java Heap Memory to at least 8192 MB when using the **Serial Link Designer** app.



Open the Serial Link Designer App

- MATLAB Toolstrip: In the **Apps** tab, under **Signal Processing and Communications**, click the app icon.
- MATLAB command prompt: Enter serialLinkDesigner.

Examples

- “Analyze Serial Links with Serial Link Designer”
- “Analyze Backplane with Line Cards”
- “Edit Imported S-Parameter Data”

Programmatic Use

`serialLinkDesigner` opens a new session of the **Serial Link Designer** app, enabling you to design and analyze a serial link PCB.

`serialLinkDesigner('file.qcd')` opens the interface designated by `file.qcd`.

`serialLinkDesigner('file.script')` runs a script file and returns the app process handle.

More About

Simulation Parameters

Simulation Parameters control how the analysis is run in the **Serial Link Designer** app. For more information, see “Simulation Parameters Used in Serial Link Design”.

Corner Conditions

There are two types of corner conditions you can vary using the **Serial Link Designer** app.

- IC Environment Corners — These are the temperature and voltage factor parameters that can be set for each IC corner and used in simulation. The temperature parameter has no effect on IBIS buffer models because those are pre-generated by the vendor and are already included in the model.
- Etch Corners — These are the scaling factors for the characteristic impedance (Z_0) and propagation delay (T_{pd}) parameters of transmission line models. Both ideal and lossy transmission line models are scaled. Lossy transmission line models are scaled by computing the values of Z_0 and T_{pd} from the typical corner L and C values.

For more information, see “Specify Corner Conditions in Serial Link Design”.

Stimulus Patterns

You can specify the stimulus patterns used by the simulator for time domain analysis. 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**. For more information, see “Stimulus Patterns in Serial Link Design”.

Version History

Introduced in R2021b

See Also

Apps
SI Viewer

Topics

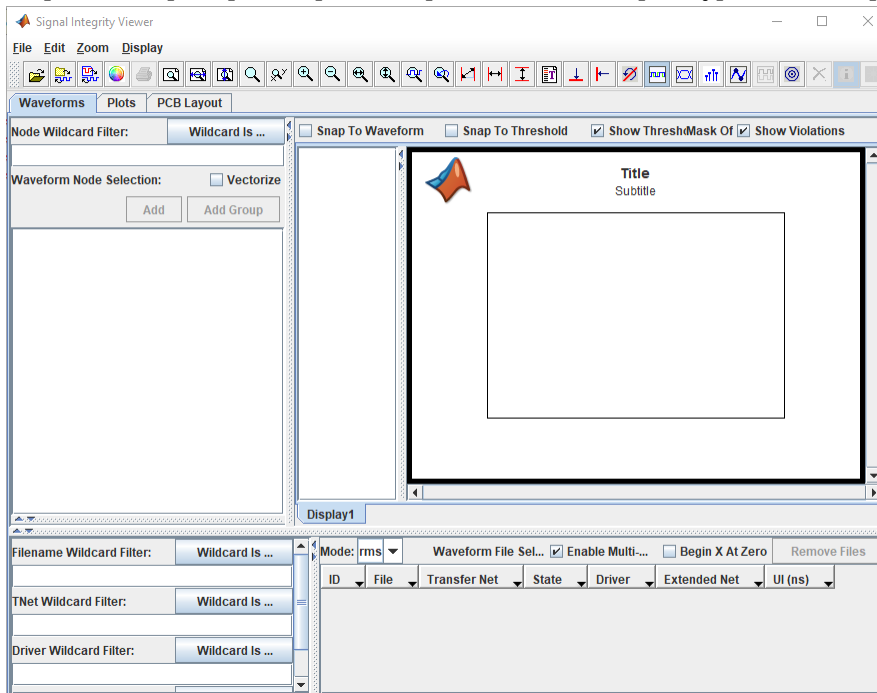
“Analyze Serial Links with Serial Link Designer”
“Analyze Backplane with Line Cards”
“Edit Imported S-Parameter Data”

Signal Integrity Viewer

View the signal integrity results of Serial Link Designer or Parallel Link Designer app

Description

Use the **Signal Integrity Viewer** app to view the results of the simulations done in **Serial Link Designer** or **Parallel Link Designer** app. This app provides a fast way to view waveforms, eye diagrams, TDR (time domain reflectometry)/ TDT (time domain transmission), s-parameters, impulse response, step response, pulse response, and other plot types for data produced by the two apps.



Open the Signal Integrity Viewer App

- MATLAB Toolstrip: In the **Apps** tab, under **Signal Processing and Communications**, click the app icon.
- MATLAB command prompt: Enter `signalIntegrityViewer`.

Examples

- “Analyze Serial Links with Serial Link Designer”
- “Analyze Parallel Links with Parallel Link Designer”

Programmatic Use

`signalIntegrityViewer` opens a new session of the **Signal Integrity Viewer** app, enabling you to view the results of a serial link or parallel link analysis.

`signalIntegrityViewer('file.csv')` opens the waveform files. Waveform files can be `.csd` from IsSpice or `.tr0` files from HSPICE.

`signalIntegrityViewer('file.cfg')` opens a saved configuration file.

Version History

Introduced in R2021b

See Also

Apps

Serial Link Designer | Parallel Link Designer

Topics

"Analyze Serial Links with Serial Link Designer"

"Analyze Parallel Links with Parallel Link Designer"

"Managing Simulation Data and Results"

"Advanced Visualization Using Signal Integrity Viewer"

Class

SignalIntegrityInterface

Return interface object of given Signal Integrity Toolbox project object

Description

The `SignalIntegrityInterface` class object returns an interface object from a given `SignalIntegrityProject` object. You can get the current interface, any specific interface in the project, or all the project interfaces.

Creation

Syntax

```
int=sip.CurrentInterface  
int=sip.Interfaces(i)  
int=sip.Interfaces
```

Description

`int=sip.CurrentInterface` returns a `SignalIntegrityInterface` class object corresponding to the current interface of the `SignalIntegrityProject` object `sip`.

`int=sip.Interfaces(i)` returns a `SignalIntegrityInterface` class object corresponding to the i -th interface of the `SignalIntegrityProject` object `sip`.

`int=sip.Interfaces` returns the `SignalIntegrityInterface` class objects corresponding to all the interfaces of the `SignalIntegrityProject` object `sip`.

Properties

Name — Name of current interface

string

Name of the current interface, specified as a string.

CurrentSchematicSet — Name of current schematic set

string

Name of the current schematic set, specified as a string.

Results — Results of simulations ran in current interface

table

Results of the simulations that ran in the current interface being observed, specified as a $[N \times (M+4)]$ table. N and M represent the number of simulations and the number of result variables, respectively.

Sheets — Object(s) representing all sheets in current interface

class object

SignalIntegritySheet object(s) representing all the sheets in the current interface, specified as [1xN] class object. *N* represents the number of sheets.

ParentProject — Parent project object

class object

SignalIntegrityProject object representing the parent project, specified as a class object.

Version History

Introduced in R2023a

See Also

SignalIntegrityProject | SignalIntegritySheet | SignalIntegrityState |
SignalIntegritySimulation | SignalIntegrityWaveform

Topics

“Access Project Data from MATLAB Command Window”

SignalIntegrityProject

Create Signal Integrity Toolbox project object

Description

The `SignalIntegrityProject` class creates a project object to access the read-only data in a Signal Integrity Toolbox project.

Creation

Syntax

```
sip=SignalIntegrityProject(projectFolder)
```

Description

`sip=SignalIntegrityProject(projectFolder)` returns a `SignalIntegrityProject` class object from the data contained in the `projectFolder`.

Example: `sip=SignalIntegrityProject('MIPI_M_PHY')` creates a `SignalIntegrityProject` class object from the `MIPI_M_PHY` project in the current working directory.

Properties

ProjectFolder — Full path to project folder

string

Full path to the project folder, specified as a string.

CurrentInterface — Object representing current interface

class object

`SignalIntegrityInterface` object representing the current interface, specified as a class object.

Type — Type of current interface

serial | parallel

Type of the current interface, specified as either a serial interface or a parallel interface.

Sheets — Object(s) representing all sheets in current interface

class object

`SignalIntegritySheet` object(s) representing all the sheets in the current interface, specified as `[1xN]` class object. N represents the number of sheets.

Interfaces — Object(s) representing all interfaces in project

class object

SignalIntegrityInterface object(s) representing all the interfaces in the current project, specified as [1xN] class object. *N* represents the number of interfaces.

Version History

Introduced in R2023a

See Also

SignalIntegrityInterface | SignalIntegritySheet | SignalIntegrityState |
SignalIntegritySimulation | SignalIntegrityWaveform

Topics

“Access Project Data from MATLAB Command Window”

SignalIntegritySheet

Return sheet object of given Signal Integrity Toolbox interface object

Description

The `SignalIntegritySheet` class object returns the sheet object from a given `SignalIntegrityInterface` object. You can get the current sheet, any specific sheet in the interface, or all the interface sheets.

Creation

Syntax

```
sh=int.CurrentSheet
sh=int.Sheets(i)
sh=int.Sheets
```

Description

`sh=int.CurrentSheet` returns a `SignalIntegritySheet` class object corresponding to the current sheet of the `SignalIntegrityInterface` object `int`.

`sh=int.Sheets(i)` returns a `SignalIntegritySheet` class object corresponding to the *i*-th sheet of the `SignalIntegrityInterface` object `int`.

`sh=int.Sheets` returns the `SignalIntegritySheet` class objects corresponding to all the sheets of the `SignalIntegrityInterface` object `int`.

Properties

Name — Name of current sheet

`string`

Name of the current sheet, specified as a string.

Included — Determine whether sheet is included in simulation

`true | false`

Determine whether the sheet is included in the simulation, specified as either `true` or `false`

CurrentState — Object representing current state

`class object`

`SignalIntegrityState` object representing current state in the sheet, specified as a class object.

Simulations — Objects representing all simulations in sheet

`class object`

SignalIntegritySimulation object(s) representing all the simulations that ran in the sheet, specified as a [1xN] class object. N represents the number of simulations.

SolutionSpace — Pre-layout solution space for current state

table

Pre-layout solution space for the current state, specified as a [N x M] table. N represents the number of solution space variables.

Results — Results of simulations ran in current sheet

table

Results of the simulations that ran in the current sheet being observed, specified as a [N x (M+4)] table. N and M represent the number of simulations and the number of result variables, respectively.

SimulationParameters — Simulation parameters currently set

table

Simulation parameters currently set, specified as a [1 x N] table. N represents the number of simulation parameters.

States — Object(s) representing all states in current sheet

class object

SignalIntegrityState object(s) representing all the states in the current sheet, specified as [1 x N] class object. N represents the number of states.

ParentInterface — Parent interface object

class object

SignalIntegrityInterface object representing the parent interface, specified as a class object.

Version History

Introduced in R2023a

See Also

SignalIntegrityProject | SignalIntegrityInterface | SignalIntegrityState | SignalIntegritySimulation | SignalIntegrityWaveform

Topics

“Access Project Data from MATLAB Command Window”

SignalIntegritySimulation

Return simulation object of given Signal Integrity Toolbox sheet object

Description

The `SignalIntegritySimulation` class object returns the simulation object from a given `SignalIntegritySheet` object or `SignalIntegrityState` object. You can get any specific simulation in the sheet or all the simulations in the sheet.

Creation

Syntax

```
sim=sh.Simulations(i)  
sim=sh.Simulations
```

Description

`sim=sh.Simulations(i)` returns a `SignalIntegritySimulation` class object corresponding to the i -th simulation of the `SignalIntegritySheet` object `sh`.

`sim=sh.Simulations` returns the `SignalIntegritySimulation` class objects corresponding to all the simulations of the `SignalIntegritySheet` object `sh`.

Properties

Name — Name of simulation

string

Name of the simulation, specified as a string.

Transfer — Name of transfer net

string

Name of the transfer net, specified as a string.

Results — Results of simulations

table

Results of the simulations, specified as a $[N \times (M+4)]$ table. N and M represent the number of simulations and the number of result variables, respectively.

Waveforms — Objects representing all waveforms in current simulation

class object

`SignalIntegrityWaveform` object(s) representing all the waveforms in the current simulation being observed, specified as a $[1 \times N]$ class object. N represents the number of waveforms.

ParentSheet — Parent sheet object

class object

SignalIntegritySheet object representing the parent interface, specified as a class object.

ParentState — Parent state object

class object

SignalIntegrityState object representing the parent state, specified as a class object.

ProjectFolder — Full path to project folder

string

Full path to the project folder, specified as a string.

Number — Simulation number

positive integer

Simulation number used for indexing, specified as a positive integer.

InterfaceName — Name of simulation interface

string

Name of the interface to which the simulation belongs, specified as a string.

Version History

Introduced in R2023a**See Also**

SignalIntegrityProject | SignalIntegrityInterface | SignalIntegritySheet | SignalIntegrityState | SignalIntegrityWaveform

Topics

“Access Project Data from MATLAB Command Window”

SignalIntegrityState

Return state object of given Signal Integrity Toolbox sheet object

Description

The `SignalIntegrityState` class object returns the state object from a given `SignalIntegritySheet` object. You can get the current state, any specific state in the sheet, or all the states in the sheet.

Creation

Syntax

```
st=sh.CurrentState
st=sh.States(i)
st=sh.States
```

Description

`st=sh.CurrentState` returns a `SignalIntegrityState` class object corresponding to the current state of the `SignalIntegritySheet` object `sh`.

`st=sh.States(i)` returns a `SignalIntegrityState` class object corresponding to the i -th state of the `SignalIntegritySheet` object `sh`.

`st=sh.States` returns the `SignalIntegrityState` class objects corresponding to all the states of the `SignalIntegritySheet` object `sh`.

Properties

Name — Name of current state

string

Name of the current state, specified as a string.

SolutionSpace — Pre-layout solution space for current state

table

Pre-layout solution space for the current state being observed, specified as a $[N \times M]$ table. N represents the number of solution space variables.

Simulations — Objects representing all simulations in state

class object

`SignalIntegritySimulation` object(s) representing all the simulations in the state being observed, specified as a $[1 \times N]$ class object. N represents the number of simulations.

ParentSheet — Parent sheet object

class object

SignalIntegritySheet object representing the parent sheet, specified as a class object.

Version History

Introduced in R2023a

See Also

SignalIntegrityProject | SignalIntegrityInterface | SignalIntegritySheet | SignalIntegritySimulation | SignalIntegrityWaveform

Topics

“Access Project Data from MATLAB Command Window”

SignalIntegrityWaveform

Return waveform object of given Signal Integrity Toolbox simulation object

Description

The `SignalIntegrityWaveform` class object returns the waveform object from a given `SignalIntegritySimulation` object. You can get any specific waveform in the simulation or all the waveforms in the simulation.

Creation

Syntax

```
w=sim.Waveforms(i)
w=sim.Waveforms
```

Description

`w=sim.Waveforms(i)` returns a `SignalIntegrityWaveform` class object corresponding to the *i*-th waveform of the `SignalIntegritySimulation` object `sim`.

`w=sim.Waveforms` returns the `SignalIntegrityWaveform` class objects corresponding to all the waveforms of the `SignalIntegritySimulation` object `sm`.

Properties

Title — Waveform title

string

Title of the waveform, specified as a string.

File — Full path to waveform file

string

Full path to the waveform file, specified as a string.

Domain — Simulation domain

string

Simulation domain, specified as network characterization, statistical, or time domain simulation.

Display — Waveform type

string

Type of the waveform, specified as BER, impulse, step, eye, and so on.

ParentSimulation — Parent simulation object

class object

SignalIntegritySimulation object representing the parent interface, specified as a class object.

More About

Methods

SignalIntegrityWaveform class object also contains these methods:

Name	Description	Access
table	Return waveform points in table form.	public
plot	Plot the waveforms.	public

Version History

Introduced in R2023a

See Also

SignalIntegrityProject | SignalIntegrityInterface | SignalIntegritySheet | SignalIntegrityState | SignalIntegritySimulation

Topics

“Access Project Data from MATLAB Command Window”

