

HyperLynx ActiveX Scripting

Mar 22, 2019

Rev 2.13

HyperLynx ActiveX Scripting	1
Common Objects	2
Application.....	2
Backdrill	3
Collection.....	4
Fabrication.....	4
IcModel.....	4
Jitter.....	5
Layer	6
Material	7
Model	7
ModelSelector	7
Pad.....	8
Padstack.....	9
Rect.....	10
Point	10
SerDesWizard	10
SimSettings	10
SimResult.....	11
Simulator	11
SParamExtractor	12
SpicePort.....	13
Stackup	13
Stimulus	15
VarCollection	15
Viewer.....	16
Waveform.....	17
Pre-layout objects	17

BboxModel	17
BlackboxObject.....	17
Coupling.....	18
DiffIcObject.....	19
IcObject.....	20
ModelPort.....	20
Node	21
ObjectPort	21
PassiveObject	21
ReferenceConductor	21
Schematic	22
SchemNet	24
SchemObject	24
SchemSupplyNet	25
TlineObject	25
ViaObject, DiffViaObject.....	27
VoltageObject.....	28
Post-layout objects.....	28
3DArea	28
Board	28
Component.....	30
CrosstalkSettings	30
Design	31
Net	33
Pin.....	33
Segment.....	34
StackupArea.....	34
Via	35

Common Objects

The section describes objects common for both pre- and post- layout analysis.

Application

Object representing HyperLynx application.

Set App = Application

Properties

Design – Object representing currently loaded post-layout design, HYP or CCE. Valid if some design file is opened manually or programmatically. Read-only. (See also [Design](#)).

Set Design = Application.Design

Schematic – Object representing current FFS schematic. Valid if schematic is loaded or new schematic is created manually or programmatically. Read-only. (See also [Schematic](#)).

Set Schem = Application.Schematic

Simulator – Object representing SI simulator. Read-only. (See also [Simulator](#)).

Set Sim = Application.Simulator

SParamExtractor – Object representing S-parameter extractor. Read-only. (See also [SParamExtractor](#)).

Set Xtractor = Application.SParamExtractor

Visible – Allows to get/set application visibility state. By default, application started from external script (Visual Basic etc.) is running in hidden state.

Application.Visible = True

FileName – Contains path name of currently opened design or schematic or empty string if there is no currently opened file.

MsgBox "Currently opened file: " + Application.FileName

SerDesWizard – Object representing SerDes Wizard. Read-only. (See also [SerDesWizard](#)).

Set serdeswiz = Application.SerDesWizard

Methods

Exit – Close the application.

Application.Exit

NewSchematic – Create new FFS schematic. Return schematic object.

Set Schem = Application.NewSchematic

OpenFile FileName – Opens specified file of any supported type. Returns TRUE in case of success, FALSE otherwise.

*If Application.OpenFile("C:\MentorGraphics\9.1HL\SDD_HOME\hyperlynx\HypFiles\demodiff.hyp")
Then*

Print Message – Prints the message in Output window.

Application.Print("Hello, World")

Backdrill

Object representing via back-drilling parameters (see also [Via](#)).

Properties

ToLayer – Name of layer to which the via should be back-drilled. Read-only.

MsgBox BD.Layer

Diameter – Back-drill hole diameter, in meters. Read-only.

If BD.Diameter = 0.001 Then

Setback – Additional indent from ToLayer, in meters. Read-only.

If BD.Setback = 0.001 Then

Collection

Object representing set of similar items.

Properties

Count – Number of items in the collection. Read-only.

For I = 1 To Nets.Count

Methods

Item Index – Returns object by index. Index is 1-based.

Set Net = Nets.Item(1)

Fabrication

Object representing layer's attributes related to PCB manufacturing (see also [Layer](#)).

Properties

Technology – Represents dielectric layer manufacturing technology. Could be one of the following:

hlTechnologyPrepreg – Pre-impregnated dielectric

hlTechnologyCore – Core dielectric

hlTechnologyFlexCore – Flex dielectric

Layer.Fabrication.Technology = hlTechnologyPrepreg

RoughnessTop – Roughness value at top trace side.

Layer.Fabrication.RoughnessTop = 0.015

RoughnessBottom – Roughness value at bottom trace side.

Layer.Fabrication.RoughnessBottom = 0.015

TrapNarrowSide - Represents narrow side of trapezoidal trace. Could be one of following:

hlLayerSideTop – Trace top is more narrow;

hlLayerSideBottom – Trace bottom is more narrow.

Layer.Fabrication.TrapNarrowSide = hlLayerSideTop

TrapEtchFactor – Represents trapezoidal trace etch factor, calculated as trace thickness divided by half of difference between wide- and narrow- trace side width.

Layer.Fabrication.TrapEtchFactor = 0.7

IcModel

Object representing IC model. (See also [Pin](#)).

Properties

LibFile – Model file name. Read-only.

MsgBox "Model File name = " + Model.LibFile

Device – Model component. Read only.

MsgBox "Model component = " + Model.Device

ModelPin – Model pin. Read-only.

MsgBox "Model pin = " + Model.ModelPin

PinModel – Model currently selected for the pin. Read-only. (See also [ModelSelector](#)).

MsgBox "Current pin model = " + Model.PinModel

DirType – Model deirection type. Possible values:

- hlModDirTypeInput – Input buffer;
- hlModDirTypeOutput – Output buffer;
- hlModDirTypeBidir – Bidirectional buffer;
- hlModDirTypeEnbo – Tri-state buffer;
- hlModDirTypeSpice – SPICE buffer.

Read-only.

If Model.DirType = hlModDirTypeOutput Then

IoDir – Model I/O direction. Possible values:

- hlIoDirInput – Input (Cannot be set to model with DirType = hlModDirTypeOutput);
- hlIoDirOutput – Output (Cannot be set to model with DirType = hlModDirTypeInput).

Model.IoDir = hlIoDirOutput

Invert – Model inverted state. Possible values:

- hlInvertStateDirect – Not inverted;
- hlInvertStateInverted – Inverted.

Model.Invert = hlInvertStateInverted

DrvState – Model driver state. Possible values are:

- hlDrvStateLow – Low state;
- hlDrvStateHigh – High state;
- hlDrvStateSwitching – Switching.

Model.DrvState = hlDrvStateHigh

Selector – Model selector, or null if there is no Model selector for the model. Read-only.
(See also [ModelSelector](#)).

If Not Model.ModelSelector Is Nothing Then

SpicePorts – Collection of SPICE ports, if the model is SPICE model. Read-only. (See also [SpicePort](#)).

For Each Port in Model.SpicePorts

Jitter

Object accumulating stimulus jitter parameters.

Properties

Gaussian – Gaussian jitter magnitude in seconds. = 0, if no gaussian jitter specified.

Jitter.Gaussian = 0.244E-9

GaussianFreq – Gaussian jitter frequency in hertz. Is ignored if property Gaussian = 0.

Jitter.GaussianFreq = 0.1E6

Uniform – Uniform jitter peak-to-peak amount in seconds. = 0, if no uniform jitter specified.

Jitter.Uniform = 0.25E-9

UniformMean – Uniform jitter mean in seconds. Is ignored if property Uniform = 0.

Jitter.UniformMean = 0.1E-9

Sine – half of sinusoidal jitter peak-to-peak amount in seconds. = 0, if no sinusoidal jitter specified.

Jitter.Sine = 0.25E-9

SineInitPhase – Sinusoidal jitter initial phase in degrees. Is ignored if property Sine = 0.

Jitter.SineInitPhase = 0.25

SineFreq – Frequency amount is specified in hertz. Is ignored if property Sine = 0.

Jitter.SineFrequency = 2.1E6

Distortion – Tx duty cycle distortion peak-to-peak amount, specified in seconds. = 0, if not specified.

Jitter.Distortion = 0.5E-9

Repeatable – Generate the same random number sequence for each simulation. TRUE or FALSE. Makes sense for Gaussian and Uniform jitter.

Jitter.Repeatable = TRUE

Methods

LoadJitter JitFileName – Loads jitter from .jit file. Returns TRUE in case of success, FALSE otherwise.

If Jitter.LoadJitter("C:\MentorGraphics\9.1HL\SDD_HOME\hyperlynx\HypFiles\demo.jit") Then

Layer

Object representing single Stackup layer (see also [Stackup](#)).

Properties

Name – Name of layer

MsgBox "Layer.Name is " & Layer.Name

Type – Represents type of layer. Could be one of the following:

hlLayerTypeSignal – Signal layer (metal layer containing traces);

hlLayerTypePlane – Plane layer (metal layer containing areas);

hlLayerTypeSubstrate – Flat (inner) dielectric layer;

hlLayerTypeSolderMask – Conformal (outer) dielectric layer.

Read-Only.

If Layer.Type = hlLayerTypeSignal

Thickness – Thickness of layer, in meters

Dim dTh as Double: dTh = Layer.Thickness

Plating – Plating thickness, in meters. Used for most top or most bottom metal layer.

Dim dPl as Double: sPl = Layer.Plating

IsVisible – TRUE if layer is visible. Otherwise it is FALSE.

Layer.IsVisible = TRUE

Color – Integer representing RGB-value of color used to draw objects on the layer.

Layer.Color = RGB(255, 0, 0)

DrawStyle – Represents style of drawing areas on the layer. Takes one of the following values:

hlDrawStyleNone – Don't draw areas;

hlDrawStyleHatched – Use hatched filling;
hlDrawStyleSolid – Use solid filling;
hlDrawStyleOutline – Draw outlines, don't fill areas.

Layer.DrawStyle = hlDrawStyleHatched

Material – Represents Material object associated with the layer. (See also [Material](#))

Material: Set Material = layer.Material

Fabrication – Represents Fabrication object associated with the layer. (See also [Fabrication](#))

Dim Fbr: Set Fbr = layer.Fabrication

Methods

Copy SrcLayer – Copies parameters of layer SrcLayer to current layer object.

Layer.Copy(Stackup.Layers.Item(1))

Material

Object representing layer's attributes related to material (See also [Layer](#)).

Properties

RelativePermittivity – Represents material dielectric constant.

Layer.Material.RelativePermittivity = 4.3

LossTangent – Represents material Loss Tangent.

Loss = Layer.Material.LossTangent

BulkResistivity – Bulk resistance of conductor, in Ohm * m.

Layer.Material.BulkResistance = 1.724e-8

ResTempCoefficient – Resistivity temperature coefficient of conductor, in 1 / °C.

Layer.Material.ResTempCoefficient = 0.00393

ThermalConductivity – Represent thermal conductivity, W / (m * K).

ThCond = Layer.Material.ThermalConductivity

Model

Object representing passive component model. (See also [Component](#)).

Properties

LibFile – Model file name. Read-only.

MsgBox "Model File name = " + Model.LibFile

Device – Model component. Read only.

MsgBox "Model component = " + Model.Device

ModelSelector

Object representing model selector for IcModel. (See also [IcModel](#)).

Properties

Selection – Currently selected item. Changing of the selection affects all IcModels of the same Component having the same Model selector.

ModelSelector.Selection = "DQ"

Count – Number of items in model selector. Read only.

For I = 1 To ModelSelector.Count

Methods

Item Index – Returns item specified by Index.

MsgBox ModelSelector.Item(2)

Pad

Object representing padstack pad (see also [Padstack](#)).

Properties

Layer – Name of stackup layer where the pad is defined, or empty string for default pad. (Default pad is used on layers where no pad is explicitly specified.) Read-only.

Set Layer = Pad.Layer

Type – Pad type, hIPadTypePad if the pad is metal or hIPadTypeAntipad if the pad is void. Read-only.

If Pad.Type = hIPadTypeAntipad Then

Shape – Shape of the pad. Possible values are:

hIPadShapeRound – round

hIPadShapeRectangle – rectangular

hIPadShapeOval – oval

hIPadShapeOblong – “oblong” (rectangular with rounded corner)

hIPadShapeCustom – arbitrary user-defined

hIPadShapeClearance – anti-pad has the same shape as pad extended by clearance value

hIPadShapeNone – no pad

Read-only for post-layout designs, read-write for pre-layout designs (schematics).

If Pad.Shape = hIPadShapeRectangle Then

Size – Pad size, in meters. For hIPadShapeRound is diameter, for hIPadShapeRectangle, hIPadShapeOval and hIPadShapeOblong is bounding rectangle width, for hIPadShapeClearance – clearance value, for hIPadShapeCustom and hIPadShapeNone is meaningless.

Read-only for post-layout designs, read-write for pre-layout designs (schematics).

Pad.Size = 0.001

Height – Pad height, in meters. For hIPadShapeRectangle, hIPadShapeOval and hIPadShapeOblong is bounding rectangle height, for the rest of pad shapes is meaningless.

Read-only for post-layout designs, read-write for pre-layout designs (schematics).

Pad.Height = 0.001

Angle – Pad rotation angle, in degrees.

Read-only for post-layout designs, read-write for pre-layout designs (schematics).

Pad.Angle = 45

OffsetX – Pad horizontal offset, in meters.

Read-only for post-layout designs, read-write for pre-layout designs (schematics).

Pad.OffsetX = 0.0001

OffsetY – Pad vertical offset, in meters.

Read-only for post-layout designs, read-write for pre-layout designs (schematics).

Pad.OffsetY = 0.0001

Padstack

Object representing via padstack (see also [Via, ViaObject](#)).

Properties

Name – Name of padstack. Read-only.

Set Name = Padstack.Layer

ActualDrillSize – diameter of drill hole, in meters. Affects FinishDrillHole value.

Read-only for post-layout designs, read-write for pre-layout designs (schematics).

Padstack.ActualDrillSize = 0.0005

FinishDrillHole – diameter of drill hole after metallization, in meters. Affects ActualDrillSize value.

Read-only for post-layout designs, read-write for pre-layout designs (schematics).

Padstack.FinishDrillHole = 0.0005

StartLayer – Name of start layer in Padstack's layer span, or empty string for topmost layer.

Read-only for post-layout designs, read-write for pre-layout designs (schematics).

Set Name = Padstack.StartLayer

EndLayer – Name of end layer in Padstack's layer span, or empty string for bottommost layer.

Read-only for post-layout designs, read-write for pre-layout designs (schematics).

Set Name = Padstack.EndLayer

Pads – Collection of Padstack's pads (see also [Pad](#)). Read-only.

Set Pads = Padstack.Pads

Methods

AddPad Layer [, Type, Shape, Size, Height, Angle, OffsetX, OffsetY] – Adds Pad with specified parameters on specified layer (or default pad if layer is empty string). Returns added pad, or null if it cannot be added. Can be used only in pre-layout designs (schematics).

Default parameter values are:

Type = `hIPadTypePad`

Shape = `hIPadShapeRound`

Size = 0.0006096 (24 mils)

Height = 0.0006096 (24 mils)

Angle = 0

OffsetX = 0

OffsetY = 0

Set Pad = Padstack.AddPad "", hIPadTypePad, hIPadShapeRectangle, 0.0012, 0.0016

GetPad Layer, Type - Returns pad of specified type defined on specified layer (or default pad if Layer is empty string), or null if there is no pad defined.

Set Pad = Padstack.GetPad("", hIPadTypeAntipad)

DeletePad Pad - Deletes specified pad.

Can be used only in pre-layout designs (schematics).

Padstack.DeletePad Pad

LockUpdate Lock – Disables/enables Padstack updating to prevent unnecessary operations.
Returns previous LockUpdate state.
Can be used only in pre-layout designs (schematics).
IsLocked = Padstack.LockUpdate(True)

Rect

Object representing rectangle.

Properties

Left – Rectangle left coordinate, in meters

Set L = Rect.Left

Bottom – Rectangle bottom coordinate, in meters

Set B = Rect.Bottom

Right – Rectangle right coordinate, in meters

Set R = Rect.Right

Top – Rectangle top coordinate, in meters

Set T = Rect.Top

Point

Object representing point

Properties

X – x-coordinate, in meters

Set x = Point.X

Y – y-coordinate, in meters

Set y = Point.Y

SerDesWizard

Object providing SerDes Wizard (i.e. extraction / analysis / compliance of differential pairs).

Properties

ResultsFolder – String containing the path to the wizard result files.

If FSO.FolderExists(Serdeswiz.ResultsFolder)

Methods

LoadSettingsFile FileName – Loads sws file to be used when the wizard is Run.

sdw.LoadSettingsFile "c:\projects\bigboard\pcie.sws"

Run Mode – Runs the wizard in the specified mode, using the settings from the specified settings file. If LoadSettingsFile has not been called, the wizard is run interactively. Returns TRUE in case of success, FALSE otherwise.

hlSerDesWizardModeFull – Full wizard mode with S-Param, Advanced Eye, etc.

hlSerDesWizardModeCompliance – Compliance mode

hlSerDesWizardModeMeasured – Measured mode for analysis of 4-port S-Parameter files

If serdeswiz.Run(hlSerDesWizardModeFull)Then

SimSettings

Object representing simulation settings.

Properties

Crosstalk – Current crosstalk simulation state (enabled or disabled). Is used for post-layout design only, attempt to access it in pre-layout (schematics) causes error.

SimSettings.Crosstalk = True

Lossy – Current lossy simulation state (enabled or disabled).

SimSettings.Lossy = True

TrapezoidTraces – If True, trapezoidal trace shape is taken into account during simulation. Otherwise, ideal rectangular trace shape is used.

SimSettings.TrapezoidTraces = True

SurfaceRoughness – If True, metal surface roughness is taken into account during simulation.

SimSettings.SurfaceRoughness = True

NonfunctionalPads – If True, non-functional (i.e. non-connected) via pads are presented, otherwise - removed.

SimSettings.NonfunctionalPads = False

SimResult

Object representing simulation results.

Properties

Waveforms – Collection of waveforms. (See also [Waveform](#)).

For I = 1 To SimResult.Waveforms.Count

Methods

Show – Displays simulation results in EZWave viewer.

SimResult.Show

FindWaveform Name – Returns waveform with the specified name or Nothing if it does not exist.

Set TLine = Coupling.FindWaveform("U1.1")

Simulator

Object providing simulation setup/control.

Properties

Stimulus – Object representing simulation stimulus. Read-only. (See also [Stimulus](#)).

Set Stim = Simulator.Stimulus

Type – Simulator type. Could be one of the following values:

hlSimTypeAuto – the best simulator is chosen automatically

hlSimTypeHyperSim – HyperLynx native simulator

hlSimTypeADMS – ELDO/ADMS simulator

hlSimTypeHSPICE – HSPICE simulator

Simulator.Type = hlSimTypeADMS

Operation – Simulator operation mode. Could be one of the following values:

hlSimOpStandard – standard transient simulation

hlSimOpEyeDiagram – Eye Diagram generation mode

Simulator.Operation = hlSimOpEyeDiagram

Corner – IC model simulation conditions. Could be one of the following values:

hlSimCornerWorst – the worst-case (Slow-Weak) model

hlSimCornerTypical – typical case (Normal) model

hlSimCornerBest – the best-case (Fast-Strong) model

Simulator.Corner = hlSimCornerTypical

StepAuto – If True, simulation step is determined automatically.

Simulator.StepAuto = False

Step – Simulation step time (resolution), in seconds.

Simulator.Step = 1e-11

StopAuto – If True, simulation stop time is determined automatically.

Simulator.StopAuto = False

StopTime – Simulation stop time, in seconds. Is used only if StopAuto property is set to False.

Simulator.StopTime = 10e-9

Result – Object representing simulation results. Read-only. (See also [SimResult](#)).

Set Result = Simulator.Result

Target – Allows to choose between EZWave and Oscilloscope waveform when displaying simulation results. Simulation results might be also accessed via SimResult.Waveforms array without displaying waveforms. Could be one of the following values:

hlSimResultTargetOscilloscope – SimResult.Show would open Oscilloscope dialog

hlSimResultTargetEZWave – SimResult.Show would open EZWave

Methods

Run – Performs simulation. Returns TRUE in case of success, FALSE otherwise.

If Simulator.Run Then

SParamExtractor

Object providing S-parameter extraction (i.e. simulation of a network and saving results as Touchstone model).

Properties

Ports – Variable collection of network pins corresponding to Touchstone model ports, specified as full pin names. Order of ports in collection is matching to order of Touchstone model ports.

Xtractor.Ports.Add "UDRV1.2"

MinFreq – Minimum Touchstone model frequency, in Hz. Default value – 100 KHz.

Xtractor.MinFreq = 1e6

MaxFreq – Maximum Touchstone model frequency, in Hz. Default value – 10 GHz.

Xtractor.MaxFreq = 5e9

FreqSweepType – Frequency sweeping type. Possible values are:

hIFreqSweepLinear – Linear frequency sweeping
hIFreqSweepLog – Logarithmic frequency sweeping
hIFreqSweepAdaptive – Adaptive frequency sweeping

Default value is hIFreqSweepAdaptive.

Xtractor.FreqSweepType = hIFreqSweepLinear

StepCount – Number of frequency points. Is used in case of hIFreqSweepLinear sweeping type, meaningless otherwise. Default value – 500.

Xtractor.StepCount = 1000

StepPerDecade – Number of frequency points per one logarithmic decade (1 MHz – 10 MHz etc.). Is used in case of hIFreqSweepLog sweeping type, meaningless otherwise. Default value is 100.

Xtractor.StepPerDecade = 200

Tolerance – Tolerance of adaptive frequency sweeping. Possible values are:

hIFreqSweepToleranceLow – Low tolerance
hIFreqSweepToleranceMedium – Medium tolerance
hIFreqSweepToleranceHigh – High tolerance

Default value is hIFreqSweepToleranceMedium.

Xtractor.Tolerance = hIFreqSweepToleranceHigh

RefImpedance – Reference impedance of load connected to Touchstone model ports, in Ohms. Default value – 50 Ohm.

Xtractor.RefImpedance = 0.01

Methods

Extract FileName – Performs extraction and saves Touchstone model using specified FileName. Returns TRUE in case of success, FALSE otherwise.

If Xtractor.Extract("c:\models\channel.s4p") Then

SpicePort

Object representing SPICE IC Model port. (See also [IcModel](#)).

Properties

Name – SPICE Model port name. Read-only.

MsgBox "Port Name " & Port.Name

CktConn – Circuit connection for the port. May be IC pin reference designator, supply net name, or one of predefined SPICE keywords, including:

NC – the port is not connected
Gnd – the port is connected to global ground
Stimulus – the port is stimulus (for driver ICs only)
Inverted Stimulus – the port is inverted stimulus (for driver ICs only)
Delayed Stimulus – the port is delayed stimulus (for driver ICs only)
Delayed Inverted Stimulus – the port is delayed inverted stimulus (for driver ICs only)

Port.CktConn = "U1.12"

Stackup

Object representing PCB stackup.

Properties

Name – Name of stackup. Name should be no more than 20 symbols length.

MsgBox "Stackup.Name is " & Stackup.Name

Layers – Collection of layers in the stackup. Read-only. (See also [Layer](#))

For I = 1 To Layers.Count

StackupAreas – Collection of rigid-flex areas where this stackup is defined. Read-only. (See also [StackupArea](#)).

Set Areas = Stackup.StackupAreas

Methods

AddLayer Type, Name – Adds layer with specified Type and Name to the end of the stackup.

See Layer.Type for possible layer types.

Returns added layer.

Set Layer = Stackup.AddLayer(hlLayerTypeSignal, "Bottom")

InsertLayer Type, Name, Index – Inserts layer with specified Type and Name to the specified position of the stackup.

Returns inserted layer.

Set Layer = Stackup.InsertLayer(hlLayerTypeSignal, "Inner", 5)

RemoveLayer Layer –Removes specified layer from the stackup.

Returns TRUE if the layer was removed FALSE otherwise. Fails if the layer contains one or more objects (trace segments, FFS transmission lines etc.)

If Not Stackup.Remove(Layer) Then

RemoveLayerAt Index –Removes layer at specified position.

Returns TRUE if the layer was removed FALSE otherwise. Fails if the layer contains one or more objects (trace segments, FFS transmission lines etc.)

If Not Stackup.Remove(5) Then

FindLayer Name –Returns layer with the specified name or Nothing if it does not exist.

Set Layer = Stackup.FindLayer("Bottom")

FindLayerIndex Name – Returns 1-based index of layer with the specified name or 0 if it does not exist.

Set Index = Stackup.FindLayerIndex("Bottom")

FindStackupArea AreaName – Returns rigid-flex area object with specified name and assigned to current stackup (See also [StackupArea](#)). Returns null if area with specified name doesn't exist.

Set Area = Stackup.FindStackupArea("RF_Area_1")

DeleteStackupArea StackupArea – Delete specified rigid-flex area (See also [StackupArea](#))

Set Area = Stackup.FindStackupArea("RF_Area_1")

Stackup.DeleteStackupArea Area

CreateStackupArea AreaName – Creates new rigid-flex area assigned to current stackup with specified name (See also [StackupArea](#)). Returns new area if creation succeeded or null otherwise

Set NewArea = Stackup.CreateStackupArea("stackupArea_1)

LockUpdate Lock – Delays updating database with stackup's changes if Lock is TRUE. Don't forget to call LockUpdate(FALSE) further to force updating database. Returns previous state of LockUpdate.

OldLock = Stackup.LockUpdate(TRUE)

Stimulus

Object representing simulation stimulus.

Properties

BitSequenceType – Specifies bit sequence to use for simulation. Possible values are:

hlBitSequenceEdge – Rising or Falling edge

hlBitSequenceOscillator – Oscillator

hlBitSequencePBRs – Pseudo-random bit sequence

hlBitSequence10B8B – 8b/10b encoding

hlBitSequence64B66B – 64b/66b encoding

hlBitSequence128B130B – 128b/130b encoding

Stimulus.BitSequenceType = hlBitSequenceOscillator

BitOrder – Specifies bit order for pseudo-random bit sequence. Possible values are 3 – 15. Is ignored for other sequence types.

Stimulus.BitOrder = 6

EdgeIsRising – Specifies edge for hlBitSequenceEdge sequence. Edge is rising if the value is True or Falling edge otherwise. Is ignored for other sequence types.

Stimulus.EdgeIsRising = True

Frequency – Specifies oscillation frequency, in Hz.

Stimulus.Frequency = 133e6

DutyCycle – Specifies duty cycle, i.e. ratio of bit length to period, for hlBitSequenceOscillator sequence, in percent. Is ignored for other sequence types.

Stimulus.DutyCycle = 50

Repetitions – Number of sequence repetitions, for Eye Diagram generation.

Stimulus.Repetitions = 32

Jitter – Specifies jitter, for Eye Diagram generation.

Set Jitter = Stimulus.Jitter

VarCollection

Object representing variable (editable) set of similar items.

Properties

Count – Number of items in the collection. Read-only.

For I = 1 To Ports.Count

Methods

Item Index – Returns item by index. Index is 1-based.

Set Port = Port.Item(1)

Add – Adds item to the end of collection.

Ports.Add("UDRV1.2")

Insert Index, Value – Inserts new item into specified position. Index is 1-based.

Ports.Insert(2, "UDRV1.3")

Remove Index – Removes item in specified position. Index is 1-based.

Ports.Remove 2

Clear – Removes all items from collection.

Ports.Clear

Viewer

Object representing Post-layout design (Board) or Pre-layout (Schematic) Viewer.

Properties

DesignExtents – Rect object containing design “physical” coordinates. For post-layout design – in meters, for pre-layout (schematic) in “abstract” units. Read-only. (See also [Rect](#))

Set Rect = Viewer.DesignExtents

ZoomExtents – Rect object containing coordinates of currently displayed areas, in “physical” coordinates. Read-only. (See also [Rect](#))

Set Rect = Viewer.ZoomExtents

WindowWidth – Viewer window width, in screen pixels. Read-only.

Set W = Viewer.WindowWidth

WindowHeight – Viewer window height, in screen pixels. Read-only.

Set H = Viewer.WindowHeight

Methods

GetDesignExtents Left, Bottom, Right, Top – Returns design “physical” coordinates. For post-layout design – in meters, for pre-layout (schematic) in “abstract” units.

Viewer.GetDesignExtents Left, Bottom, Right, Top

GetZoomExtents Left, Bottom, Right, Top – Returns coordinates of currently displayed area, in “physical” coordinates.

Viewer.GetZoomExtents Left, Bottom, Right, Top

GetWindowSize Width, Height – Returns Viewer window dimensions, in pixels.

Viewer.GetWindowSize Width, Height

ZoomTo Left, Bottom, Right, Top – Displays specified rectangular area in “physical” coordinates to be fitted/centered in Viewer window.

Viewer.ZoomTo Left, Bottom, Right, Top

FitToWindow – Fits whole design to Viewer window.

Viewer.FitToWindow

DesignToWindowX(X) – Returns window horizontal position, in pixels, corresponding to specified physical X coordinate.

WinX = Viewer.DesignToWindowX(0.01)

DesignToWindowY(Y) – Returns window vertical position, in pixels, corresponding to specified physical Y coordinate.

WinY = Viewer.DesignToWindowY(0.01)

WindowToDesignX(X) – Returns physical X coordinate corresponding to specified window horizontal position, in pixels.

X = Viewer.WindowToDesignX(100)

WindowToDesignY(Y) – Returns physical Y coordinate corresponding to specified window vertical position, in pixels.

Y = Viewer.WindowToDesignY(100)

ScaleToWindow(Distance) – Returns value, in pixels, corresponding to specified physical distance.

Dist = Viewer.ScaleToWindow(0.01)

ScaleToDesign(Distance) – Returns value, in physical coordinates, corresponding to specified distance between two window points, in pixels.

Dist = Viewer.ScaleToDesign(100)

Waveform

Object representing single simulation waveform

Properties

Name – Waveform name. Read-only.

MsgBox "Waveform name = " + Waveform.Name

Methods

Save FileName – Save the waveform into specified text file. Returns TRUE in case of success, FALSE otherwise.

Waveform.Save "c:/waveform1.txt"

Pre-layout objects

The section describes objects available in pre-layout design (schematic)

BboxModel

Object representing Touchstone or SPICE model assigned to FFS Black Box (see also BlackboxObject).

Properties

LibFile – Model file name. Read-only.

MsgBox "Model File name = " + Model.LibFile

Subckt – SPICE model sub-circuit. Meaningless for Touchstone models. Read only.

MsgBox "Model sub-circuit = " + Model.Subckt

Params – Additional SPICE simulator parameters. Contains parameters as specified in the model itself as well as specifying in BlackBox.AssignModel method. Read only.

MsgBox "Model parameters = " + Model.Params

BlackboxObject

Object representing arbitrary part of FFS schematic described by Touchstone or SPICE model.

Properties

Model – Object representing black-box model (see also [BboxModel](#)). Read-only.

Set Model = Res.Model

ModelPorts –Collection of ports provided by assigned model (see also [ModelPort](#)). Read-only.

For I = 1 To BBox.ModelPorts.Count

Methods

AssignModel File, Subckt, Params – Assigns model specified by file, sub-circuit (SPICE model only) and additional SPICE simulator parameters. Returns TRUE in case of success, FALSE otherwise.

Result = Res.AssignModel("simple_diff.sp", "Tx_diff", "")

Coupling

Object representing transmission lines coupling.

Properties

Name – Coupling name.

MsgBox Coupling.Name

Length – Coupled region length, in meters. By default – equal to length of first added transmission line.

Coupling.Length = 0.0025

TLines – Collection of transmission lines included in coupling (see also [TlineObject](#)). Read-only.

For Each TLine in Coupling.TLines

FindTLine Name – Returns t-line with the specified name or Nothing if it does not exist.

Set TLine = Coupling.FindTLine("TL1")

RefConductors – Collection of Reference Conductors included in coupling (see also [ReferenceConductor](#)). Read-only.

For Each RefConductor in Coupling.RefConductors

Methods

Include TLine, Pos – Includes specified transmission line in coupling in specified horizontal (X) position, in meters.

Coupling.Include TLine, 0.0005

Exclude TLine – Excludes specified transmission line from coupling.

Coupling.Exclude TLine

GetPos TLine – Returns current horizontal (X) position in coupling of specified transmission line, in meters.

Pos = Coupling.GetPos(TLine)

SetPos TLine, Pos – Moves transmission line to specified horizontal (X) position in coupling, in meters.

Coupling.SetPos TLine, 0.0005

GetLeftSeparation TLine – Returns distance between specified transmission line and one to the left, or zero, in meters.

Left = Coupling.GetLeftSeparation(TLine)

SetLeftSeparation TLine, Left – Moves specified transmission line and all lines to the right, to specify distance to the left one transmission line, in meters.

Coupling.SetLeftSeparation TLine, 0.0005

GetRightSeparation TLine – Returns distance between specified transmission line and one to the right, or zero, in meters.

Pos = Coupling.GetRightSeparation(TLine)

SetRightSeparation TLine, Right – Moves all transmission lines to the right, to specify distance to the next to the right transmission line, in meters.

Coupling.SetRightSeparation TLine, 0.0005

AddRefConductor Type, LayerName, Pos, Width – Adds in coupling a reference conductor of specified type, located on specified layer. Optionally sets the horizontal (X) conductor position (meaningful if conductor type is not h1RefConductorPlane or h1RefConductorSpacer) and width (meaningful if conductor type is h1RefConductorTrace), in meters.

Possible types are:

Returns added ReferenceConductor (see also [ReferenceConductor](#)).

h1RefConductorPlane – Solid plane

h1RefConductorHalfPlaneLeft – Left half-plane

h1RefConductorHalfPlaneRight – Right half-plane

h1RefConductorTrace – Trace

h1RefConductorSpacer – Spacer

Returns added ReferenceConductor (see also [ReferenceConductor](#)).

RefConductor = Coupling.AddRefConductor(h1RefConductorTrace, "GND", 0.001, 0.000254)

DelRefConductor RefConductor – Deletes specified reference conductor from coupling.

Coupling.DelRefConductor RefConductor

DifficObject

Object representing FFS IC buffer.

Properties

RefDes – IC buffer reference designator. Read-only.

MsgBox IC.RefDes

Pin1Name – First pin name. Read-only.

MsgBox IC.Pin1Name

Pin2Name – Second pin name. Read-only.

MsgBox IC.Pin2Name

Model1 – Object representing IC model assigned to the first port of the element or null.

Read-only. (See also [IcModel](#)).

Set Model = Ic.Model1

Model2 – Object representing IC model assigned to the second port of the element or null.

Read-only. (See also [IcModel](#)).

Set Model = Ic.Model2

Methods

AssignModel PortIndex, File, Device, ModelPin – Assigns model specified by file, device (model component) and model pin to specified pin of the element. Valid PinIndex values are 1 and 2. Returns TRUE if the model is successfully assigned, FALSE otherwise

Result = Ic.AssignModel(1, "demo1.ibs", "COMP1", "1")

Rename RefDes, Pin1Name, Pin2Name – Assign new reference designator and pin names to the object. Returns TRUE in case of success, or FALSE if an object with the same reference designator and pin name already exists.

Result = IC.Rename "UDRV", "OUT+", "OUT-"

IcObject

Object representing FFS IC buffer.

Properties

RefDes – IC buffer reference designator. Read-only.

MsgBox IC.RefDes

PinName – Pin name. Read-only.

MsgBox IC.PinName

Model – Object representing IC model assigned to element or null. Read-only. (See also [IcModel](#)).

Set Model = Ic.Model

Methods

AssignModel File, Device, ModelPin – Assigns model specified by file, device (model component) and model pin. Returns TRUE if the model is successfully assigned, FALSE otherwise.

Result = Ic.AssignModel("demo1.ibs", "COMP1", "1")

Rename RefDes, PinName – Assign new reference designator and pin name to the object. Returns TRUE in case of success, or FALSE if an object with the same reference designator and pin name already exists.

Result = IC.Rename "UDRV", "A1"

ModelPort

Object representing single port provided by Touchstone or SPICE model assigned to FFS Black Box object (see [BlackboxObject](#)).

Properties

Name –Port name. Meaningless for Touchstone models. Read-only.

If ModelPort.Name = "input" Then

ConnType –Connection type for the port. Possible values are:

hlModelPortConnSchem – Port is connected to FFS schematic. Default value;

hlModelPortConnUser – Connection for the port is specified by user.

ModelPort.ConnType = hlModelPortConnUser

ConnNet –Net connected to the port. Possible values are:

One of predefined net names ("Gnd" for grounded port, "NC" for non-connected port);

Name of any power-supply net presented in the schematic.
Meaningless for ports with connection type `hlModelPortConnSchem`.
ModelPort.ConnNet = "NC"

Side – Black box symbol side where port is located. Possible values are:
 `hlSideLeft` – Left side of symbol;
 `hlSideRight` – Right side of symbol.
Meaningless for ports with connection type `hlModelPortConnSchem`.
ModelPort.Side = hlSideRight

Node

Object representing schematic connection (Node).

Properties

Objects – Collection of schematic elements connected to the node. Read-only.
Set Objects = Node.Objects

ObjectPort

Object representing schematic object port. Is used to represent all objects connected to a schematic net (see also [SchemNet](#)).

Properties

Object – Object the port belongs to. Read-only.
Set Objs = ObjPort.Object

Index – Port index in object's port collection. Read-only.
Set Idx = ObjPort.Index

PassiveObject

Object representing FFS passive component, including Resistor, Capacitor and Inductance.

Properties

Value – Passive component value, depending on particular component type, in base units (Ohm for resistance, F for capacitance and H for inductance).
Cap.Value = 1e-6

Model – Object representing passive component model. Read-only. (see also [Model](#)).
Set Model = Res.Model

Methods

AssignModel File, Device – Assigns model specified by file and device (model component).
Returns TRUE in case of success, FALSE otherwise.
Result = Res.AssignModel("resistor.ibs", "resistor")

ReferenceConductor

Object representing reference (ground) conductor in transmission lines coupling (see also [Coupling](#)).

Properties

Type – Reference conductor type. Read-only. One of
 `hlRefConductorPlane` – Solid plane
 `hlRefConductorHalfPlaneLeft` – Left half-plane
 `hlRefConductorHalfPlaneRight` – Right half-plane

hlRefConductorTrace – Trace

hlRefConductorSpacer – Spacer

If RefConductor.Type = hlRefConductorPlane Then

Layer – Name of stackup layer where the conductor is located. Read-only.

MsgBox RefConductor.Layer

Pos – Conductor horizontal (X) position, in meters. Read-only. Meaningful if conductor type is not hlRefConductorPlane and not hlRefConductorSpacer.

If RefConductor.Pos < 0.001 Then

Width – Conductor width, in meters. Read-only. Meaningful if conductor type is hlRefConductorTrace.

If RefConductor.Width < 0.000254 Then

Schematic

Object representing FFS schematic.

Properties

Stackup – PCB <master> stackup of the schematic. Read-only. (See also [Stackup](#)).

Set Stackup = Schematic.Stackup

Stackups – Collection of PCB stackups of the board. Read-only. (See also [Stackup](#)).

Set Stackups = Schematic.Stackups

Objects - Collection of schematic elements. Read-only. (See also [SchemObject](#)).

Set Objects = Schematic.Objects

Nets - Collection of schematic nets. Read-only. (See also [SchemNet](#)).

Set Nets = Schematic.Nets

SupplyNets - Collection of schematic power-supply nets. Read-only. (See also [SchemSupplyNet](#)).

Set Supplies = Schematic.SupplyNets

Couplings - Collection of coupled transmission line groups. Read-only. (See also [Coupling](#)).

Set Couplings = Schematic.Couplings

Padstacks - Collection of padstacks. Read-only. (See also [Padstack](#)).

Set Padstacks = Schematic.Padstacks

SimSettings – Simulation settings. Read-only. (See also [SimSettings](#)).

Set SimSettings = Schematic.SimSettings

Viewer – Schematic Viewer. Read-only. (See also [Viewer](#)).

Set Viewer = Schematic.Viewer

Methods

AddObject Type – Add object of a specified type to the schematic (see [SchemObject](#) for the list of available types). Returns added object.

Set Ic = Schematic.AddObject(hlScmObjTypeIc)

FindObject ObjectName – Returns object with specified name or null if the object does not exist.

Set TL = Schematic.FindObject("TL1")

Delete Object – Deletes specified schematic object or connection (node).

Schematic.Delete Object

AddPadstack Name – Adds empty padstack with specified name. Returns added padstack.

Set Padstack = Schematic.AddPadstack("NewPadstack")

FindPadstack Name – Returns padstack with the specified name, or null if it's not exist.

Set Padstack = Schematic.FindPadstack("NewPadstack")

DeletePadstack Padstack – Deletes specified padstack.

Schematic.DeletePadstack Padstack

GetNet Object, Port – Returns net containing specified port of specified object (see also [ObjectPort](#), [SchemNet](#)).

Set Net = Schematic.GetNet(Obj, 1)

Connect Object1, Port1, Object2, Port2 – Connects specified ports of specified objects.

Returns TRUE in case of success, FALSE otherwise.

Result = Schematic.Connect(Res, 2, Cap, 1)

Disconnect Object, Port – Disconnects specified port of specified object.

Schematic.Disconnect Object, 1

FindCoupling Name – Returns coupling with the specified name or Nothing if it does not exist.

Set Coupling = Coupling.FindCoupling("TestCoupling")

CreateCoupling – Returns new empty coupled group. (See also [Coupling](#)).

Set Coupling = Schematic.CreateCoupling

DiscardCoupling – Removes specified coupled group. (See also [Coupling](#)).

Schematic.DiscardCoupling Coupling1

AutoPlace – Performs automatic elements placement.

Schematic.AutoPlace

Save FileName – Saves schematic using specified file name. Returns True in case of success, False otherwise.

Schematic.Save "c:\schematics\myschematic.fff"

FindStackup StackupName – Returns stackup object with specified name. Returns null if stackup doesn't exist. Returns <master> stackup if StackupName is empty

Set Stackup = Schematic.FindStackup("stackup_1")

CreateStackup Name, Stackup – Creates new stackup that will be copy of existing stackup if second parameter is not null. New stackup will have specified name. Returns created stackup or null if provided name for new stackup is not unique or creation failed.

Set NewStackup = Design.CreateStackup("stackup_1", 0)

DeleteStackup Stackup – Delete specified PCB stackup

Schematic.DeleteStackup Schematic.FindStackup("stackup_1")

AddSupplyNet Name, Voltage – Creates new supply net with specified name and voltage, returns created net. (See also [ShemSupplyNet](#)).

Set PWR17 = Schematic.AddSupplyNet("PWR", 1.7)

FindSupplyNet Name – Returns supply net by specified name, or null object if the net does not exist or is not supply.

Set VCC = Schematic.FindSupplyNet("VpullUp")

DeleteSupplyNet Name – Deletes supply net by specified name, if the net exist, is supply and not in use. (See also [ShemSupplyNet](#).)

Schematic.DeleteSupplyNet Schematic.FindSupplyNet("VpullUp")

SchemNet

Object representing schematic nets, that is, a set of objects ports connected directly and/or by transmission lines (see also [ObjectPort](#)).

Properties

Name – Net name.

Net.Name = "TX"

Items – Collection of object ports included to the net. Read-only.

Set Items = Net.Items

SchemObject

Object representing generic schematic element. All schematic elements support properties and methods of SchemObject.

Properties

Name – Schematic element name. Read- only.

MsgBox "Object name = " + Object.Name

Type – Schematic element type. Possible values are:

hlScmObjTypeIc – IC Buffer

hlScmObjTypeDiffIc – Differential IC Buffer

hlScmObjTypeResistor – Resistor

hlScmObjTypeInductance – Inductance

hlScmObjTypeCapacitor – Capacitor

hlScmObjTypeFerrBead – Ferrite Bead;

hlScmObjTypeTline – Transmission line

hlScmObjTypeGround – Ground

hlScmObjTypeVoltage – Power

hlScmObjTypeBlackBox – Black-box

hlScmObjTypeVia – Via

hlScmObjTypeDiffVia – Differential Via

hlScmObjTypeEntry – Entry point (S-parameter port)

hlScmObjTypeMosfet – Series MOSFET component

hlScmObjTypeComment – Text Comment

Read-only.

If Object.Type = hIScmObjTypeResistor Then

PortCount – Number of object ports. Read-only.

For I = 1 To Object.PortCount

X – Object horizontal position, in Schematic coordinates. Read-only.

X = Object.X

Y – Object vertical position, in Schematic coordinates. Read-only.

Y = Object.Y

Methods

GetNode PortIndex – Returns object representing connection (Schematic Node) for the specified port, or null if the port is not connected. Port index is 1-based. (See also [Node](#)).

Set Node = Objec.GetNodet(1)

GetPortIndex Node – Returns index of port connected to the specified node. Index is 1-based.

Index = Object.GetPortIndex(Node)

GetCoord X, Y – Return object position in Schematic coordinates

Object.GetCoord X, Y

SetCoord X, Y – Move object to specified position in Schematic coordinates.

Object.SetCoord 100, 200

SchemSupplyNet

Objects representing power-supply (power or ground) schematic net.

Properties

Name – Net name. Read-only.

MsgBox "Supply Net name: " & SupplyNet.Name

IsUse – True if the net is used (i.e. is one of predefined supply nets, or is assigned to VoltageObject object. See also [VoltageObject](#)).

If SupplyNet.InUse Then

Voltage – Power voltage of the supply net. Read-write.

SupplyNet.Voltage = 3.3

TlineObject

Object representing FFS transmission line.

Properties

LineType – Transmission line type. Possible values are:

hITLTypeSimple – Simple transmission line. Simulation-related parameters (e.g. impedance, delay, etc.) of the transmission line are specified by user.

hITLTypeStackup – Trace segment located on stackup layer. Simulation-related parameters are calculated automatically on the base of PCB Stackup data. Default value for new transmission lines.

hITLTypeMicrostrip – Trace segment located on top of stackup, above reference plane.

hITLTypeBuriedMicrostrip – Trace segment located inside dielectric, above reference plane.

hITLTypeStripline – Trace segment located inside dielectric, between upper and lower reference planes.

hITLTypeWireOverGround – Round wire located inside dielectric, above ground plane.

hITLTypeMicrostripOverHatch – Trace segment located on top of stackup, above reference plane containing metal hatch.

hITLTypeStriplineBetweenHatch - Trace segment located inside dielectric, between upper and lower reference planes containing metal hatch.

TLine.Type = hITLTypeStackup

Z0 – Transmission line impedance, in Ohms. For transmission line type hITLTypeSimple is read-write, for other types is read-only.

TLine.Z0 = 50

Delay – Transmission line delay, in seconds. For transmission line type hITLTypeSimple is read-write, for other types is read-only.

If TLine.Delay > 1e-6 Then

R – Transmission line DC resistance, in Ohms. For transmission line type hITLTypeSimple is read-write, for other types is read-only.

TLine.R = 0.001

Layer – Name of layer where transmission line located. Meaningful only for transmission line type hITLTypeStackup.

TLine.Layer = "Top"

Length – Transmission line length, in meters. Meaningful for all transmission line types except hITLTypeSimple.

TLine.Length = 0.002

Width – Transmission line width, in meters. Meaningful for all transmission line types except hITLTypeSimple.

TLine.Width = 2.54e-4

Thickness – Transmission line thickness, in meters. Meaningful for all transmission line types except hITLTypeSimple and hITLTypeStackup.

TLine.Thickness = 2.54e-4

Plating – Thickness of additional metal plating, in meters. Meaningful for transmission line types hITLTypeMicrostrip and hITLTypeMicrostripOverHatch.

TLine.Plating = 2.54e-5

DielAboveHeight – Thickness of dielectric layer above transmission line, in meters.

Meaningful for transmission line types hITLTypeBuriedMicrostrip, hITLTypeStripline and hITLTypeStriplineBetweenHatch.

TLine.DielAboveHeight = 2.54e-3

DielBelowHeight – Thickness of dielectric layer below transmission line, in meters.

Meaningful for all transmission line types except hITLTypeSimple and hITLTypeStackup.

TLine.DielBelowHeight = 2.54e-3

DielConstant – Dielectric constant of adjacent dielectric layer(s). Meaningful for all transmission line types except hTLTypeSimple and hTLTypeStackup.

TLine.DielConstant = 4.3

LossTangent – Loss tangent of adjacent dielectric layer(s). Meaningful for all transmission line types except hTLTypeSimple and hTLTypeStackup.

TLine.LossTangen = 0.01

SolderMaskHeight – Thickness of dielectric solder mask covering trace segment. Meaningful for transmission line types hTLTypeMicrostrip and hTLTypeMicrostripOverHatch.

TLine.SolderMaskHeight = 2.54e-5

SolderMaskDielConstant – Dielectric constant of dielectric solder mask covering trace segment. Meaningful for transmission line types hTLTypeMicrostrip and hTLTypeMicrostripOverHatch.

TLine.SolderMaskDielConstant = 3.0

BulkResistivity – Bulk resistivity of transmission line metal, Ohm - meter. Meaningful for all transmission line types except hTLTypeSimple and hTLTypeStackup.

TLine.BulkResistivity = 1e-8

TemperatureCoef – Temperature coefficient of transmission line metal, 1 / degree. Meaningful for all transmission line types except hTLTypeSimple and hTLTypeStackup.

TLine.TemperatureCoef = 4e-3

HatchWidth – Width of hatch lines, in meter. Meaningful for transmission line types hTLTypeMicrostripOverHatch and hTLTypeStriplineBetweenHatch.

TLine.HatchWidth = 2.54e-4

HatchPitch – Distance between hatch lines, in meter. Meaningful for transmission line types hTLTypeMicrostripOverHatch and hTLTypeStriplineBetweenHatch.

TLine.HatchPitch = 2.54e-4

HatchAngle – Angle of hatch lines, in degree. Meaningful for transmission line types hTLTypeMicrostripOverHatch and hTLTypeStriplineBetweenHatch.

TLine.HatchAngle = 45.0

HatchOffset – Distance from transmission line to nearest hatch lines crossing, in meter. Meaningful for transmission line types hTLTypeMicrostripOverHatch and hTLTypeStriplineBetweenHatch.

TLine.HatchOffset = 1.00e-4

Stackup – Name of stackup associated with TlineObject. Empty name if master stackup is associated. Read-write.

TLine.Stackup = "stackup_1"

ViaObject, DiffViaObject

Objects representing FFS Via and Differential Via respectively.

Properties

MaxConnectedLayers – Maximum number of layers where via can be connected to transmission line. By default – 2.

Via.MaxConnectedLayer = 4

Padstack– Name of Padstack assigned to via (See also [Padstack](#)).

Via.Padstack = NewPadstack.Name

Stackup – Name of stackup associated with TlineObject. Empty name if master stackup is associated. Read-write.

TLine.Stackup = "stackup_1"

VoltageObject

Object representing connection to power-supply net.

Properties

SupplyNet – Name of power-supply net associated with the object. Net should exist in the schematic (See also [ShemSupplyNet](#)).

Voltage.SupplyNet = "VCC"

Post-layout objects

The section describes objects available in post-layout designs.

3DArea

Object representing board area selected for 3D EM simulation.

Properties

Active – Defines 3D Area status. If True, the area is included in simulation, if False – does not.

Area.Active = True

Name – 3D Area name.

MsgBox Area.Name

Nets – Collection containing all nets affected by the 3D Area. Read-only. (See also [Search3DAreas](#) – Automatically searches 3D Areas on currently selected nets. Returns True in case of success, False otherwise.

If Design.Search3DAreas() = True Then

Delete3DArea Area – Deletes specified 3D Area.

Design.DeleteDAreas Area

Net).

Set Nets = Area.Nets

ModelFile – File name of S-parameter model corresponding to the 3DArea. Maybe absolute or relative. Specified file name may not contain extension, it will be automatically generated.

Area.ModelFile = "my3DAreaModel"

OutputFile – Output (CCE) file name of the 3DArea. Maybe absolute or relative. Specified file name may not contain extension, it will be automatically generated.

Area.OutputFile = "my3DArea.cce"

PortCount – Number of 3DArea ports. Read-only.

If Area.PortCount = 2 Than

Methods

Solve – Performs Full-Wave Solver simulation to create S-parameter model of the 3D Area. Returns True in case of success, False otherwise.

If Area.Solve() = False Than

Board

Object representing single board of currently loaded post-layout multi-board design (PJH).

Properties

Stackup – <master> PCB stackup of the board. Read-only. (See also [Stackup](#)).

Set Stackup = Board.Stackup

Stackups – Collection of PCB stackups of the board. Read-only. (See also [Stackup](#)).

Set Stackups = Board.Stackups

Nets – Collection containing all nets of the board. Read-only. (See also [Search3DAreas](#) – Automatically searches 3D Areas on currently selected nets. Returns True in case of success, False otherwise.

If Design.Search3DAreas() = True Than

Delete3DArea Area – Deletes specified 3D Area.

Design.DeleteDAreas Area

Net).

Set Nets = Board.Nets

Components – Collection containing all components of the board. Read-only. (See also [Component](#)).

Set Comps = Board.Components

CurSelNet – Object representing currently selected net. May be null if there is no currently selected net.

Board.CurSelNet = Design.FindNet("clk")

FileName – Design file corresponding to the particular board of multi-board project. May be absolute or relative to multi-board project file.

MsgBox "Board file " & Board.FileName

Suffix – Suffix used to identify objects of the particular board in multi-board design (_B00, _B01, etc.)

If Right(Net.Name, 4) = Board.Suffix Then

Methods

FindComponent RefDes – Returns component object with specified reference designator or null if the component does not exist in the board.

Set Comp = Board.FindComponentt("U11")

If Not Comp Is Nothing Then

FindNet NetName – Returns net object with specified name or null if the net does not exist in the design.

Set Clk = Board.FindNet("clk")

If Not Clk Is Nothing Then

FindStackup StackupName – Returns stackup object with specified name. Returns null if stackup doesn't exist. Returns <master> stackup if StackupName is empty. StackupName is supposed to be without board suffix ("stackup_1" not "stackup_1_B01")

Set Stackup = Board.FindStackup("stackup_1")

CreateStackup Name, Stackup – Creates new stackup that will be copy of existing stackup if second parameter is not null. New stackup will have specified name. Returns created stackup or null if provided name for new stackup is not unique or creation failed.

Set NewStackup = Design.CreateStackup("stackup_1", 0)

DeleteStackup Stackup – Delete specified PCB stackup

Board.DeleteStackup Board.FindStackup("stackup_1")

Component

Object representing component.

Properties

RefDes – Component reference designator. Read-only.

MsgBox "Component name = " + Comp.RefDes

Type – Component type. Possible values are:

hlCompTypeIC – IC

hlCompTypeResistor – Resistor

hlCompTypeCapacitor – Capacitor

hlCompTypeConnector – Connector

hlCompTypeInductance – Inductance

hlCompTypeDiode – Diode

hlCompTypeFerrite – Ferrite Bead

hlCompTypeRC – RC

hlCompTypeTestPoint – Test point

Read-only.

If Comp.Type = hlCompTypeResistor Then

PartType – Component Part Type name.

Read-only.

If Comp.PartType = "10K" Then

Value – Component value, depending on particular component type, in base units (Ohm for resistance, F for capacitance and H for inductance). Valid only for passive components (hlCompTypeResistor, hlCompTypeCapacitor, hlCompTypeInductance).

Comp.Value = 10000

Model – Component model. Valid only for passive components (hlCompTypeResistor, hlCompTypeCapacitor, hlCompTypeInductance). Read-only. (See also [Model](#)).

Set Model = Comp.Model

Pins – Collection of component pins. Read-only. (See also [Pin](#)).

For I = 1 To Comp.Pins.Count

Methods

FindPin Name – Returns pin object with specified reference designator or null if the pin does not exist in the design. (See also [Pin](#)).

Set Pin = Comp.FindPin("11")

AssignModel File, Device – Assigns model specified by file and device (model component). Returns TRUE in case of success, FALSE otherwise. Valid only for passive components (hlCompTypeResistor, hlCompTypeCapacitor, hlCompTypeInductance).

Result = Res.AssignModel("resistor.ibs", "resistor")

CrosstalkSettings

Object representing crosstalk simulation settings.

Properties

Method – Crosstalk estimation method. Read-only. Possible values are:

hlCrosstalkElectrical – electrical estimation

hlCrosstalkGeometric – geometric estimation

If CrosstalkSettings.Method = hlCrosstalkGeometric Then

Threshold – Crosstalk threshold, in Volts, for electrical crosstalk estimation. Read-only.

MsgBox "Crosstalk Threshold = " + CStr(CrosstalkSettings.Threshold)

MaxDistToAgressor – Maximum distance to aggressor net, in meters, for geometric crosstalk estimation. Read-only.

MsgBox "Maximum distance to aggressor = " + CStr(CrosstalkSettings.MaxDistToAgressor)

MinCoupleSegLen – Minimum length of coupled segment, in meters, for geometric crosstalk estimation. Read-only.

MsgBox "Minimum length of coupled segment = " + CStr(CrosstalkSettings.MinCoupleSegLen)

HorzNeighborLim – Maximum number of neighbors in horizontal direction, in meters, for geometric crosstalk estimation. Read-only.

MsgBox "Max number of neighbors X = " + CStr(CrosstalkSettings.HorzNeighborLim)

VertNeighborLim – Maximum number of neighbors in vertical direction, in meters, for geometric crosstalk estimation. Read-only.

MsgBox "Max number of neighbors Y = " + CStr(CrosstalkSettings.VertNeighborLim)

Methods

SetElectrical Threshold – Sets crosstalk estimation method to hlCrosstalkElectrical, assigns specified Threshold value.

CrosstalkSettings.SetElectrical 0.1

SetGeometric MaxDistToAgressor, MinCoupleSegLen, HorzNeighborLim, VertNeighborLim – Sets crosstalk estimation method to hlCrosstalkGeometric, assigns specified parameters.

CrosstalkSettings.SetGeometric 0.01, 0.0025, 1, 1

Design

Object representing currently loaded post-layout design (HYP or CCE).

Properties

Boards – Collection containing all boards of multi-board design. Read-only. (See also [Board](#)).

Set Boards = Design.Boards

Components – Collection containing all components of the design. Read-only. (See also [Component](#)).

Set Comps = Design.Components

Stackup – PCB <master> stackup. Read-only. (See also [Stackup](#)).

Set Stackup = Design.Stackup

Stackups – Collection of PCB stackups of the board. Read-only. (See also [Stackup](#)).

Set Stackups = Design.Stackups

Padstacks – Collection of padstacks. Read-only. (See also [Padstack](#)).

Set Padstacks = Design.Padstacks

Nets – Collection containing all nets in the design. Read-only. (See also [Search3DAreas](#) – Automatically searches 3D Areas on currently selected nets. Returns True in case of success, False otherwise.

If Design.Search3DAreas() = True Than

Delete3DArea Area – Deletes specified 3D Area.

Design.DeleteDAreas Area

Net).

Set Nets = Design.Nets

TriDAreas – Collection containing all 3D Areas of the board. Read-only. (See also [3DArea](#)).

Set Areas = Design.TriDAreas

CurSelNet – Object representing currently selected net. May be null if there is no currently selected net.

Design.CurSelNet = Design.FindNet("clk")

SelectedNets – Object representing collection of currently selected nets. Read-only.

Set SelNest = Design.SelectedNets

SimSettings – Simulation settings. Read-only. (See also [SimSettings](#)).

Set SimSettings = Design.SimSettings

CrosstalkSettings – Crosstalk simulation settings. Read-only. (See also [CrosstalkSettings](#)).

Set CrosstalkSettings = Design.CrosstalkSettings

Viewer – Design Viewer. Read-only. (See also [Viewer](#)).

Set Viewer = Design.Viewer

Methods

FindComponent RefDes – Returns component object with specified reference designator or null if the component does not exist in the design.

Set Comp = Design.FindComponentt("U11")

FindNet NetName – Returns net object with specified name or null if the net does not exist in the design.

Set Clk = Design.FindNet("clk")

SelectNet NetName – Set specified net as currently selected and returns corresponding net object or null if the net does not exist in the design.

Set Clk = Design.SelectNet("clk")

SelectNets NetNames – Selects specified set of nets, passed as single string containing net names delimited by whitespace character.

Set Clk = Design.SelectNet("clk clk in lhrs2")

FindPadstack Name – Returns padstack with the specified name, or null if it's not exist.

Set Padstack = Design.FindPadstack("NewPadstack")

Save – Saves user-definitions (.BUD) file for the design. Returns True in case of success, False otherwise.

Design.Save

FindStackup StackupName – Returns stackup object with specified name. Returns null if stackup doesn't exist. Returns <master> stackup if StackupName is empty

Set Stackup = Design.FindStackup("stackup_1")

CreateStackup Name, Stackup – Creates new stackup that will be copy of existing stackup if second parameter is not null. New stackup will have specified name. Returns created stackup or null if provided name for new stackup is not unique or creation failed.

Set NewStackup = Design.CreateStackup("stackup_1", 0)

DeleteStackup Stackup – Deletes specified PCB stackup.

Design.DeleteStackup Design.FindStackup("stackup_1")

Search3DAreas – Automatically searches 3D Areas on currently selected nets. Returns True in case of success, False otherwise.

If Design.Search3DAreas() = True Then

Delete3DArea Area – Deletes specified 3D Area.

Design.DeleteDAreas Area

Net

Object representing physical (CAD) net.

Properties

Name – Net name. Read-only.

MsgBox "Net Name = " + Net.Name

Length – Net length. Read-only.

MsgBox "Net Length = " + CStr(Net.Length)

IsSupply – TRUE if the net is supply, FALSE otherwise.

Net.IsSupply = TRUE

Voltage – Supply net voltage, in Volts. Meaningless for non-supply nets.

Net.Voltage = 3.3

Pins – Collection of pins belonging to the net. Read-only. (See also [Pin](#)).

Set Pins = Net.Pins

Segments – Collection of segments belonging to the net. Read-only. (See also [Segment](#)).

Set Segs = Net.Segments

Vias – Collection of vias belonging to the net. Read-only. (See also [Via](#)).

Set Vias = Net.Vias

RelatedNets – Collection of nets related to the net. Relation may be as conductive (connection via passive component etc.) or non-conductive (coupling). Read-only.

Set Related = Net.RelatedNets

Pin

Object representing component pin.

Properties

Name – Pin name. Read only.

MsgBox "Pin name = " + Pin.Name

FullName – Full pin name in form <Component Reference Designator>.<Pin Name>. Read-only.

MsgBox "Full Pin name = " + Pin.FullName

Component – Object representing component to which the pin belongs. Read-only. (See also [Component](#)).

Set Comp = Pin.Comp

Net – Object representing component to which the pin belongs. Read-only. (See also [Net](#)).

Set Net = Pin.Net

Model – Object representing IC model assigned to pin or null. Read-only. (See also [Model](#), [IcModel](#)).

Set Model = Pin.Model

Methods

AssignModel File, Device, ModelPin – Assigns model specified by file, device (model component) and model pin. Returns TRUE if the model is successfully assigned, FALSE otherwise. (See also [Model](#), [IcModel](#)).

Result = Pin.AssignModel("demo1.ibs", "COMP1", "1")

Segment

Object representing trace segment.

Properties

Layer – Layer where the segment is located. Read only.

Set Layer = Seg.Layer

X1 – Horizontal coordinate of first segment end, meters. Read-only.

If Seg.X1 = 0.01 Then

Y1 – Vertical coordinate of first segment end, meters. Read-only.

If Seg.Y1 = 0.01 Then

X2 – Horizontal coordinate of second segment end, meters. Read-only.

If Seg.X1 = 0.02 Then

Y2 – Vertical coordinate of second segment end, meters. Read-only.

If Seg.Y1 = 0.02 Then

Width – Segment width, meters. Read-only.

If Seg.Width = 0.00254 Then

Stackup – Name of stackup associated with TlineObject. Empty name if master stackup is associated. Read-only.

MsgBox "Stackup = " + Seg.Stackup

StackupArea

Object representing rigid-flex areas.

Properties

Name – Area name. Read-write.

MsgBox Area.Name

Points – Collection of polygon vertices that define area. (See also [Point](#)). Read-only.

Set Points = Area.Points

Rect – Rectangular that describes the area or null if the area is described by polygon (See also [Rect](#)). Read-only.

Set Rect = Area.Rect

Methods

SetRect left, top, right, bottom – Defines area as rectangular with specified coordinates in meters. If stackup object is retrieved from Board (not Design object) that these should be local board coordinates.

Area.SetRect 0, 0, 0.05, 0.05

AddVertex x, y – Add vertex in polygon which defines rigid-flex area. Should be used in combination with ClosePolyline. It is expected no less than 3 vertices are added. Coordinates are to be in meters and are local if area was retrieved from Board object

Area.AddVertex 0.05, 0.05

ClosePolyline – Completes creating polygon that defines area by closing it

Area.ClosePolyline

Via

Object representing via.

Properties

X – Horizontal coordinate, meters. Read only.

If Via.X = 0.01 Then

Y – Vertical coordinate, meters. Read only.

If Via.Y = 0.01 Then

DrillSize – Drill hole diameter, meters. Read only.

If Via.DrillSize = 0.001 Then

Padstack – Name of via padstack. Read-only.

If Via.Padstack = "PADSTACK_01" Then

TopLayer – Topmost layer in via layer span. Read-only.

If Via.Layer.Name = "TOP" Then

BottomLayer – Bottommost in via layer span. Read-only.

If Via.Layer.Name = "BOTTOM" Then

ConnectedLayers – Collection of layers where via is connected to trace or area. Read-only.

For Each Layer In Via.ConnectedLayers

Stackup – Name of stackup associated with TlineObject. Empty name if master stackup is associated. Read-only.

MsgBox "Stackup = " + Via.Stackup

Methods

ClearBackdrill LayerPos – Discards back-drilled state for layer specified by LayerPos where possible LayerPos values are:

hlLayerPosTopmost – Topmost stackup layer

hlLayerPosBottommost – Bottommost stackup layer

Via.ClearBackdrill hlLayerPosBottommost

SetBackdrill LayerPos, Diameter, Setback, ToLayer – Sets back-drilled state for via side specified by LayerPos (see ClearBackdrill method for possible LayerPos values), assigns specified back-drilling parameters (see [Backdrill](#) object for descriptions of Diameter, Setback and ToLayer parameters). If optional ToLayer parameter is missed, the first connected layer is used as ToLayer.

Via.SetBackdrill hlLayerPosBottommost, 0.0005, 0.0001

GetBackdrill LayerPos – Returns Backdrill object for via side specified by LayerPos (see ClearBackdrill method for possible LayerPos values), or Null if the back-drilling is not defined. (See also [Backdrill](#)).

If Not Via.GetBackdrill(hlLayerPosBottommost) Is Nothing Then