Virtuoso Basic Library Reference

Product Version ICADVM20.1
October 2020

Contents

Preface

The `basic` library is a default reference library provided within the Virtuoso `cds.lib` file.

This reference contains basic information about each symbol within the basic library.

This preface contains the following topics:

Scope

Unless otherwise noted, the functionality described in this guide can be used in both mature node (for example, IC6.1.8) and advanced node and methodologies (for example, ICADVM18.1) releases.

Label	Meaning
(ICADVM18.1 Only)	Features supported only in the ICADVM18.1 advanced nodes and advanced methodologies release.
(IC6.1.8 Only)	Features supported only in mature node releases.

Licensing Requirements

The license number required for the Virtuoso® Schematic Editor L is 95100.

The license number required for the Virtuoso® Schematic Editor XL is 95115.

For information on licensing in the Virtuoso design environment, see *Virtuoso Software Licensing and Configuration Guide*.

Related Documentation

What's New and KPNS

- *Virtuoso Schematic Editor What's New*
- *Virtuoso Schematic Editor Known Problems and Solutions*

Installation, Environment, and Infrastructure

- *Cadence Installation Guide*
- *Virtuoso Design Environment User Guide*
- *Virtuoso Design Environment SKILL Reference*
- *Cadence Application Infrastructure User Guide*

Technology Information

- *Virtuoso Technology Data User Guide*
- *Virtuoso Technology Data SKILL Reference*

Virtuoso Tools

IC6.1.8 Only

- *Virtuoso Layout Suite L User Guide*
- *Virtuoso Layout Suite XL User Guide*

ICADVM18.1 Only

- *Virtuoso Layout Suite XL: Basic Editing User Guide*
- *Virtuoso Layout Suite XL: Connectivity Driven Editing*
- *Spectre Circuit Simulator RF Analysis Library Reference*

IC6.1.8 and ICADVM18.1

- *Virtuoso Schematic Editor SKILL Reference*

Additional Learning Resources

Video Library

The Video Library on the Cadence Online Support website provides a comprehensive list of videos on various Cadence products.

To view a list of videos related to a specific product, you can use the *Filter Results* feature available in the pane on the left. For example, click the *Virtuoso Layout Suite* product link to view a list of videos available for the product.

You can also save your product preferences in the Product Selection form, which opens when you click the *Edit* icon located next to *My Products*.

Virtuoso Videos Book

You can access certain videos directly from Cadence Help. To learn more about this feature and to access the list of available videos, see Virtuoso Videos.

Rapid Adoption Kits

Cadence provides a number of Rapid Adoption Kits that demonstrate how to use Virtuoso applications in your design flows. These kits contain design databases and instructions on how to run the design flow.

In addition, Cadence offers the following training courses on Virtuoso Schematic Editor:

- Virtuoso Schematic Editor
- Virtuoso Analog Design Environment
- Using Virtuoso Constraints Effectively
- Virtuoso Spectre Circuit Simulator
- Spectre Simulations Using Virtuoso ADE
- Virtuoso Electrically-Aware Design with Layout Dependent Effects

Cadence also offers the following training courses on the SKILL programming language, which you can use to customize, extend, and automate your design environment:

- SKILL Language Programming Introduction
- SKILL Language Programming
- Advanced SKILL Language Programming

To explore the full range of training courses provided by Cadence in your region, visit Cadence Training or write to training_enroll@cadence.com.

Note: The links in this section open in a separate web browser window when clicked in Cadence Help.

Help and Support Facilities

Virtuoso offers several built-in features to let you access help and support directly from the software.

- The Virtuoso *Help* menu provides consistent help system access across Virtuoso tools and applications. The standard Virtuoso *Help* menu lets you access the most useful help and support resources from the Cadence support and corporate websites directly from the CIW or any Virtuoso application.
- The Virtuoso Welcome Page is a self-help launch pad offering access to a host of useful knowledge resources, including quick links to content available within the Virtuoso installation as well as to other popular online content.

  The Welcome Page is displayed by default when you open Cadence Help in standalone mode from a Virtuoso installation. You can also access it at any time by selecting *Help – Virtuoso Documentation Library* from any application window, or by clicking the *Home* button on the Cadence Help toolbar (provided you have not set a custom home page).

For more information, see Getting Help in *Virtuoso Design Environment User Guide.*

Customer Support

For assistance with Cadence products:

- Contact Cadence Customer Support

Cadence is committed to keeping your design teams productive by providing answers to technical questions and to any queries about the latest software updates and training needs. For more information, visit https://www.cadence.com/support.

- Log on to Cadence Online Support

  Customers with a maintenance contract with Cadence can obtain the latest information about various tools at https://support.cadence.com.

Feedback about Documentation

You can contact Cadence Customer Support to open a service request if you:

- Find erroneous information in a product manual
- Cannot find in a product manual the information you are looking for
- Face an issue while accessing documentation by using Cadence Help

You can also submit feedback by using the following methods:

- In the Cadence Help window, click the *Feedback* button and follow instructions.
- On the Cadence Online Support Product Manuals page, select the required product and submit your feedback by using the *Provide Feedback* box.

1

Basic Library Categories

The symbols in the `basic` library are categorized as follows:

- Misc
- Obsolete
- Pins
- Supplies
- VHDLPins

You can customize the values of symbols by amending the definition of the associated SKILL variables in the file `schConfig.il`. The file `schconfig.il` (located in `install_dir/tools/dfII/samples/local`) specifies various schematic editor configurations, including a configuration of the symbols within this library.

Misc

The *Misc* category (miscellaneous) lists those symbols that do not fall within the *Supplies* and *Pins* categories. It includes the following symbols:

- cds_alias
- cds_thru
- cds_thrualias
- dummy
- nlpglobals
- noConn
- onPageConn
- patch

cds_alias

Symbol View

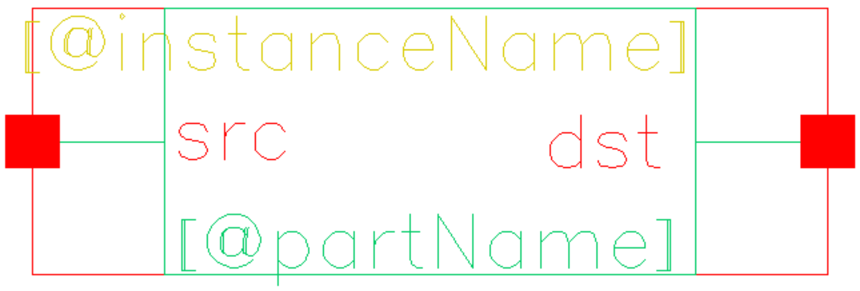
cds_alias does not have a symbol view.

Description

A simple cell used by Verilog netlisters to short two nets.

cds_thru

Symbol View



Description

This symbol is used as a component to create a connection that may otherwise be reported as a short. For example, when you need a short between two terminals or between a terminal and a global signal, use `cds_thru` as it prevents connectivity extraction errors in Virtuoso Schematic Editor due to shorted terminals.

The component is designed to netlist for the following tools:

- cdl/auCdl: a small value resistor that can be ignored by using `*.RESI` commands
- spectre: an iprobe, which is a zero-voltage source
- Verilog: a module describing the short. Terminal-to-terminal assigns are legal in Verilog.

In addition to a symbol view, this component cell has a functional view and other analog views, such as auLvs, hspice, hspiceD, spectre, and so on. The type of simulation determines whether a functional or analog view is used. To use an analog netlister to netlist a schematic, you can use analog views in the Hierarchy Editor (HED). To use the Verilog netlister, a functional view must be used in HED.

If you want to use another library and cell for shorting terminals, refer to *Virtuoso VHDL Toolbox User Guide*.

When an instance is bound to the function view of `cds_thru` in HED, cds_thrualias also display as a sub-instance of `cds_thru` in the hierarchy. AMS Unified Netlister (AMS UNL) generates an instance with `cds_thru` and passes the files, including the definitions of `cds_thru` and `cds_thrualias`, to the simulator. For example:

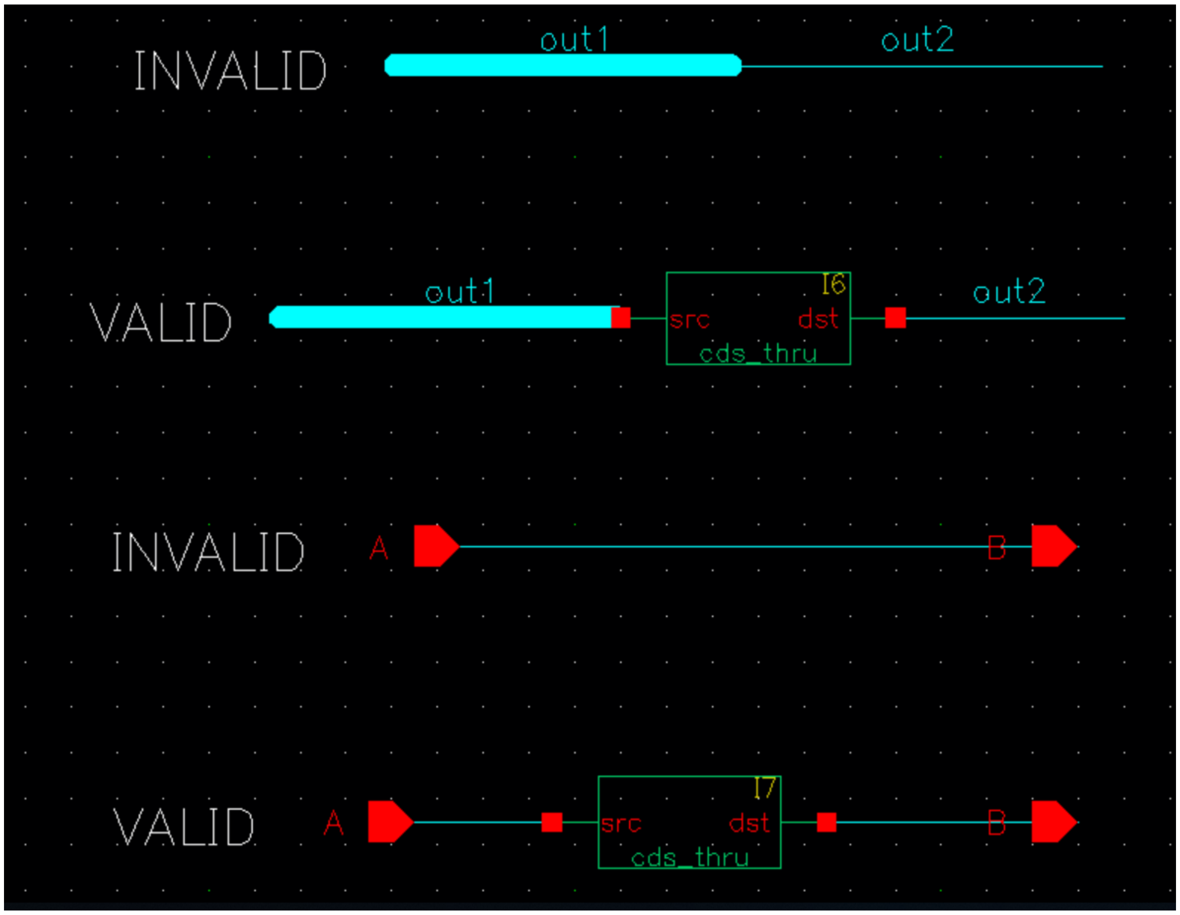
```
cds_thru I39 ( net012 ,  net09 );
```

Example

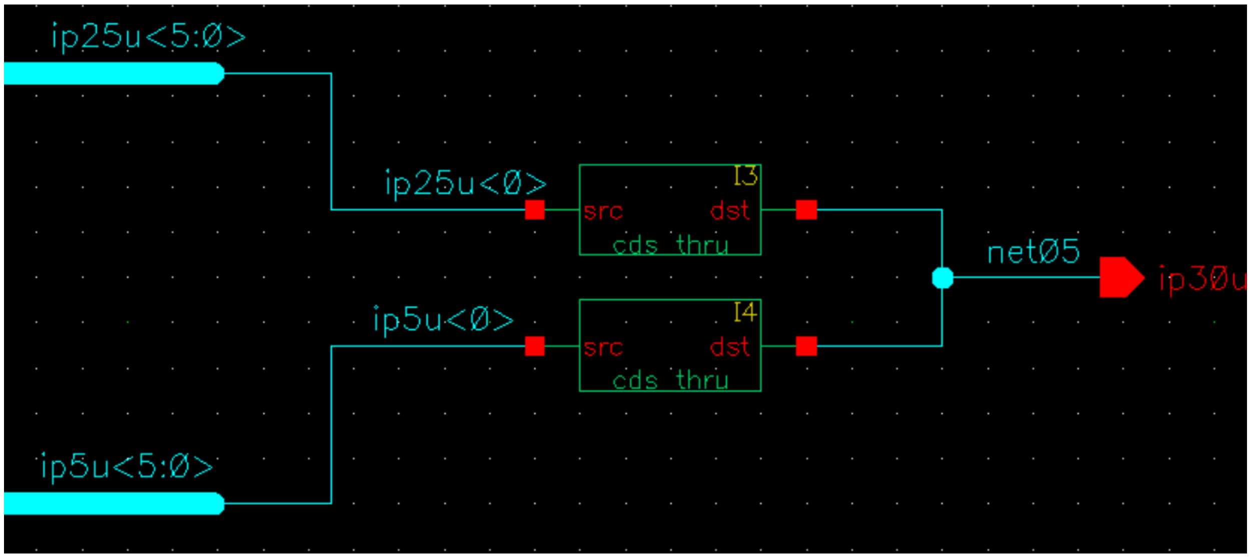
To short together two pins labeled `out1` and `out2`, you can place the `cds_thru` component between the two pins. It would netlist in Spectre as follows:

```
I0 (out1 out2) iprobe
```

The iprobe does not affect the simulation run. So you have shorted `out1` to `out2`. The following screenshot shows valid and invalid connections:

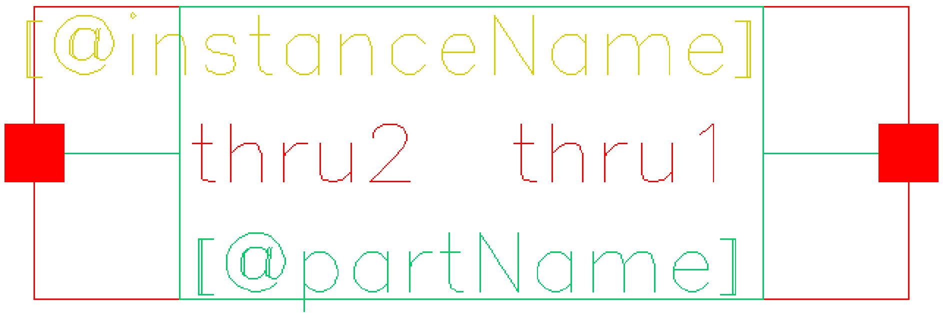


You can also combine two bus nets to form a single net using `cds_thru` as follows:



cds_thrualias

Symbol View



Description

When an instance is bound to the function view of cds_thru in the HED, `cds_thrualias` is also displayed as a sub-instance of `cds_thru` in the hierarchy.

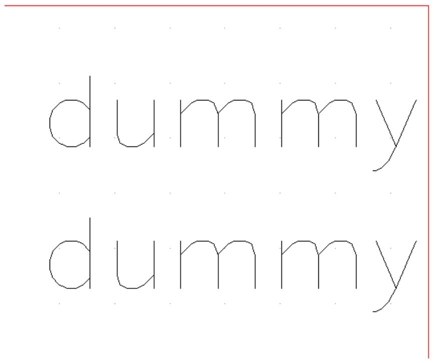
AMS UNL generates an instance with `cds_thru` and passes the files, including the definitions of `cds_thru` and `cds_thrualias`, to the simulator.

Example

```
cds_thrualias I40 ( net015 ,  net014 );
```

dummy

Symbol View

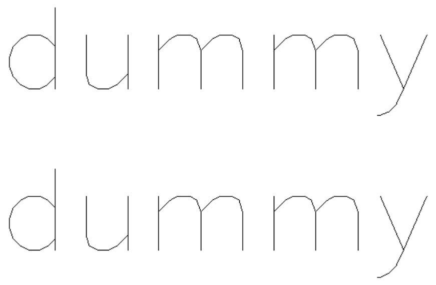


Description

A dummy symbol provided for nlpglobals.

nlpglobals

Symbol View



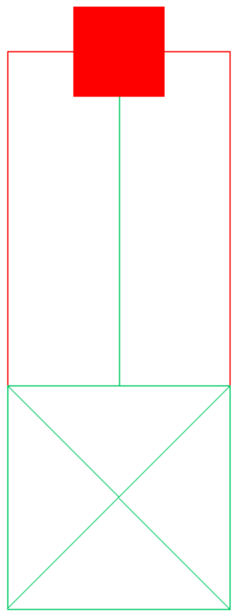
Description

A global block symbol containing the global netlist format property definitions used to indicate global nets. This is required when creating a global cellview for netlisting.

You must create an nlpglobals cell if using the flat netlister (FNL) to create a netlist. For more details on creating global cellviews, refer to the Open Simulation System Reference.

noConn

Symbol View

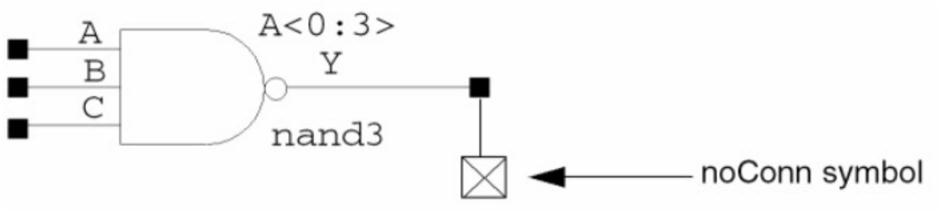


Description

This symbol can be attached to pins with signals that are unconnected to component output pins, schematic input pins, or any other schematic I/O pins. Adding this symbol to such pins ensures that they are not highlighted by the floating pin check.

Example

When the noConn +symbol is placed on the end of a dangling wire, the schematic rules check does not highlight the wire as an error.

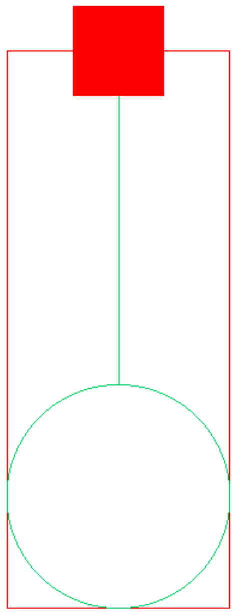


Related Topics

For more details, see Bypassing Floating Pin Checks in the *Virtuoso Schematic Editor User Guide*.

onPageConn

Symbol View

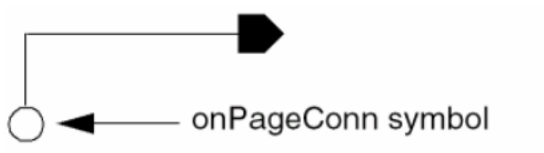


Description

Use this symbol on wires without end points that do not physically connect to schematic or component pins, labels, or other wire segments. Attaching this symbol bypasses the unconnected wire check.

Example

When the onPageConn symbol is placed on the end of a dangling wire, the schematic rules check does not highlight the wire as an error.

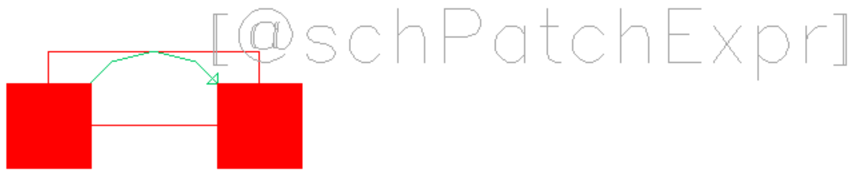


Related Topics

For more details, see Bypassing Unconnected Wire Checks in the *Virtuoso Schematic Editor User Guide*.

patch

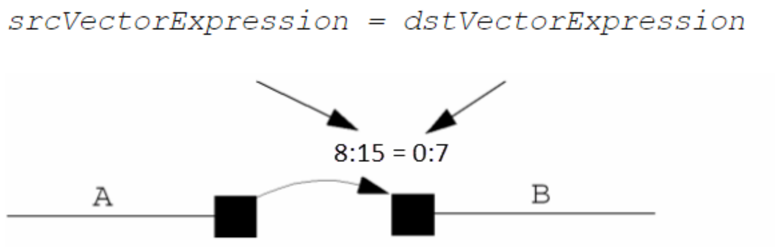
Symbol View



Description

Use this symbol to create a patchcord to establish aliases between the signals of two different nets.

Example



Nets `A` and `B` are aliased together. Logically, each object that `A` is connected to is also connected to `B`, and the patchcord connects the first bit named in `srcVectorExpression` to the first bit named in `dstVectorExpression`.

Related Topics

See the following topics in the *Virtuoso Schematic Editor User Guide:*

Patchcord Connections and Patchcord Naming Conventions

Adding Patchcords Using the Add Instances Form

Adding Patchcords Using the Create Patchcord Form

Obsolete

The symbols listed in this section are obsolete and should not be used for new designs. They are preserved for existing designs.

Connectors

FCON
MCON

MiscObsolete

simNameState
simState

Saber

idc	isin	vexp
iexp	vccs	vpulse
ipulse	vcvs	vpwl
ipwl	vdc	vsin

SuppliesObsolete

CGND	GND	VDD
DGND	PWR	
EGND	VCC	

TA

dst	iosc	src
gnd_net	oosc	ta_vdd

Pins

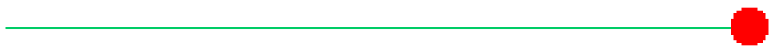
The *Pins* category lists the symbol pins. It includes the following symbols:

- actHiInOut / actHiInp / actHiOut
- blockiopin / blockipin / blockopin
- circle
- commActLoInOut / commActLoInp / commActLoOut
- ieeeActLoInOut / ieeeActLoInp / ieeeActLoOut
- in / io / out
- iopin / ipin / opin
- sympin
- tsgActLo
- tsgActLoClock
- tsgClock
- tsgIeeeActLoInp
- tsgIeeeActLoOut

For more details, see Adding Pins as Graphic Images in the *Virtuoso Schematic Editor User Guide*.

actHiInOut

Symbol View



Description

A symbol pin used when the pin type `actHi` is selected with a direction of `input-output` on the Create Pin — Symbol form. There are also `input` and `output` direction pins available for the pin type `actHi`.

The `schSymbolPinMasters` variable defines the symbol pins that are associated with the specific pin direction and type.

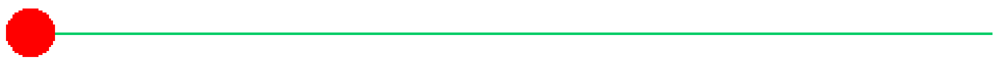
For more details, see Customizing Global Editor Variables for Form Fields in the *Virtuoso Schematic Editor User Guide*.

Example

```
schSymbolPinMasters = list(
    list("actHi"
        list("inputOutput" list("basic" "actHiInOut" "symbol")))
)
```

actHiInp

Symbol View



Description

A symbol pin used when the pin type `actHi` is selected with a direction of `input` on the Create Pin — Symbol form. There are also `output` and `inputOutput` direction pins available for the pin type `actHi`.

The `schSymbolPinMasters` variable defines the symbol pins that are associated with the specific pin direction and type.

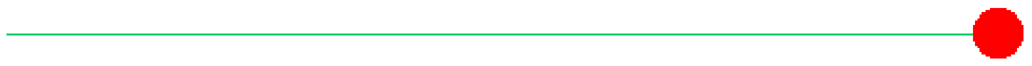
For more details, see Customizing Global Editor Variables for Form Fields in the *Virtuoso Schematic Editor User Guide*.

Example

```
schSymbolPinMasters = list(
    list("actHi"
        list("input" list("basic" "actHiInp"   "symbol")))
)
```

actHiOut

Symbol View



Description

A symbol pin used when the pin type `actHi` is selected with a direction of `output` on the Create Pin — Symbol form. There are also `input` and `inputOutput` direction pins available for the pin type `actHi`.

The `schSymbolPinMasters` variable defines the symbol pins that are associated with the specific pin direction and type.

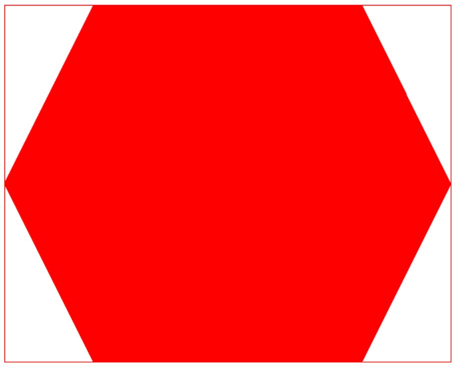
For more details, see Customizing Global Editor Variables for Form Fields in the *Virtuoso Schematic Editor User Guide*.

Example

```
schSymbolPinMasters = list(
    list("actHi"
        list("output" list("basic" "actHiOut"   "symbol")))
)
```

blockiopin

Symbol View



Description

A symbol pin used when the pin type `block` is selected with a direction of `inputOutput` on the Create Pin — Symbol form.

The `schSymbolPinMasters` variable defines the symbol pins that are associated with the specific pin direction and type. The `tsgConnectorMasters` variable defines the symbol of the pin connector.

For more details, see Customizing Global Editor Variables for Form Fields in the *Virtuoso Schematic Editor User Guide*.

Example

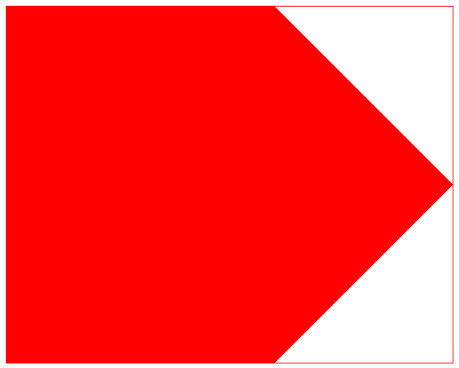
```
schSymbolPinMasters = list(
    list("block"
        list("inputOutput" list("basic" "blockiopin" "symbol")))
)
```

Related Topics

Adding Blocks in the *Virtuoso Schematic Editor User Guide*

blockipin

Symbol View



Description

A symbol pin used when the pin type `block` is selected with a direction of `input` on the Create Pin — Symbol form.

The `schSymbolPinMasters` variable defines the symbol pins that are associated with the specific pin direction and type. The `tsgConnectorMasters` variable defines the symbol of the pin connector.

For more details, see Customizing Global Editor Variables for Form Fields in the *Virtuoso Schematic Editor User Guide*.

Example

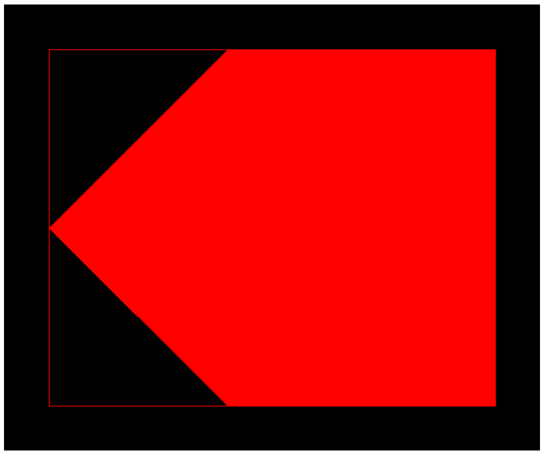
```
schSymbolPinMasters = list(
    list("block"
        list("input" list("basic" "blockipin"  "symbol")))
)
```

Related Topics

Adding Blocks in the *Virtuoso Schematic Editor User Guide*

blockopin

Symbol View



Description

A symbol pin used when the pin type `block` is selected with a direction of `output` on the Create Pin — Symbol form.

The `schSymbolPinMasters` variable defines the symbol pins that are associated with the specific pin direction and type. The `tsgConnectorMasters` variable defines the symbol of the pin connector.

For more details, see Customizing Global Editor Variables for Form Fields in the *Virtuoso Schematic Editor User Guide*.

Example

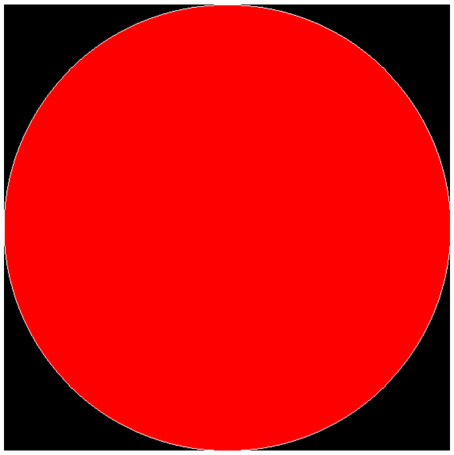
```
schSymbolPinMasters = list(
    list("block"
        list("output" list("basic" "blockopin"  "symbol")))
)
```

Related Topics

Adding Blocks in the *Virtuoso Schematic Editor User Guide*

circle

Symbol View



Description

A symbol pin used when the pin type `round` is selected on the Create Pin — Symbol form.

The `schSymbolPinMasters` variable defines the symbol pins that are associated with the specific pin direction and type. The `tsgConnectorMasters` variable defines the symbol of the pin connector.

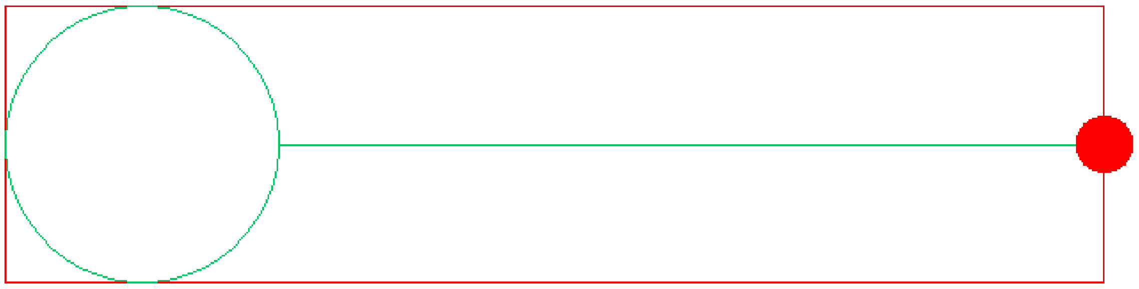
For more details, see Customizing Global Editor Variables for Form Fields in the *Virtuoso Schematic Editor User Guide*.

Example

```
schSymbolPinMasters = list(
    list("round"
        list("input" list("basic" "circle" "symbol"))
        list("output" list("basic" "circle" "symbol"))
        list("inputOutput" list("basic" "circle" "symbol"))
        list("switch" list("basic" "circle" "symbol"))
        list("jumper" list("basic" "circle" "symbol"))
        list("tristate" list("basic" "circle" "symbol"))
        list("unused" list("basic" "circle" "symbol")))
)
```

commActLoInOut

Symbol View



Description

A symbol pin used when the pin type `commActLo` is selected with a direction of `inputOutput` on the Create Pin — Symbol form. There are also `input` and `output` direction pins available for the pin type `commActLo`.

The `schSymbolPinMasters` variable defines the symbol pins that are associated with the specific pin types and directions.

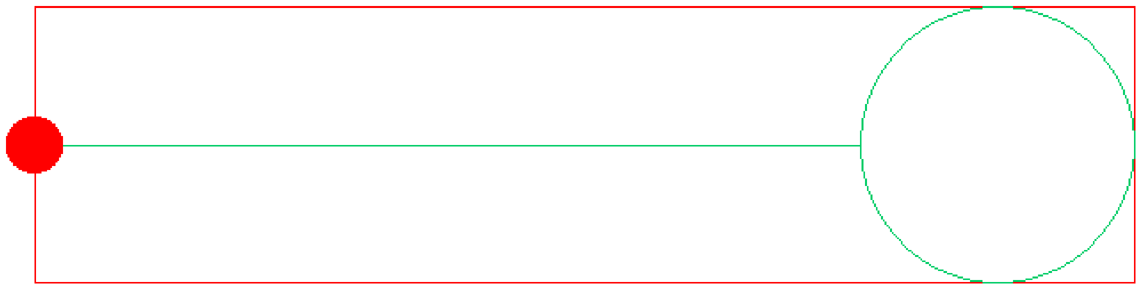
For more details, see Customizing Global Editor Variables for Form Fields in the *Virtuoso Schematic Editor User Guide*.

Example

```
schSymbolPinMasters = list(
    list("commActLo"
        list("inputOutput" list("basic" "commActLoInOut" "symbol")))
)
```

commActLoInp

Symbol View



Description

A symbol pin used when the pin type `commActLo` is selected with a direction of `input` on the Create Pin — Symbol form. There are also `output` and `inputOutput` direction pins available for the pin type `commActLo`.

The `schSymbolPinMasters` variable defines the symbol pins that are associated with the specific pin types and directions.

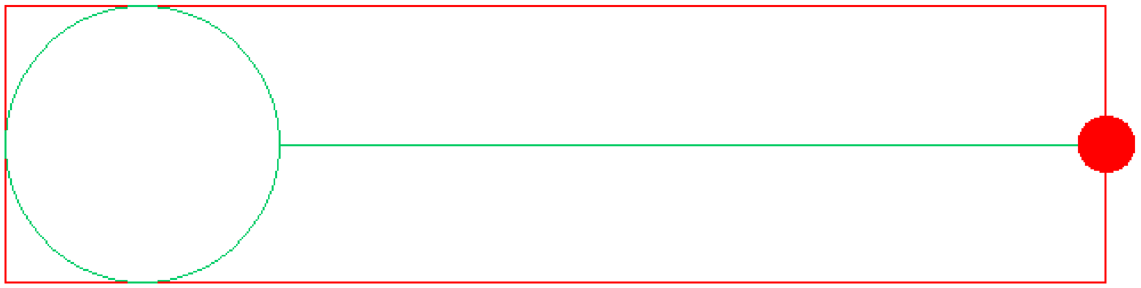
For more details, see Customizing Global Editor Variables for Form Fields in the *Virtuoso Schematic Editor User Guide*.

Example

```
schSymbolPinMasters = list(
    list("commActLo"
        list("input" list("basic" "commActLoInp"   "symbol")))
)
```

commActLoOut

Symbol View



Description

A symbol pin used when the pin type `commActLo` is selected with a direction of `output` on the Create Pin — Symbol form. There are also `input` and `inputOutput` direction pins available for the pin type `commActLo`.

The `schSymbolPinMasters` variable defines the symbol pins that are associated with the specific pin types and directions.

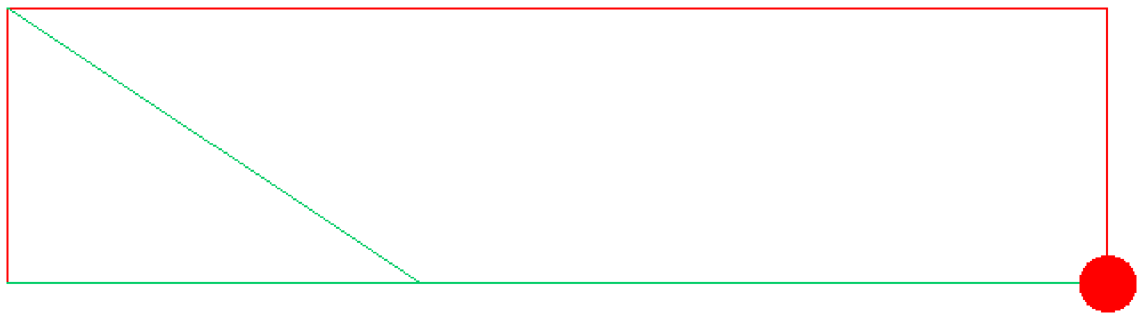
For more details, see Customizing Global Editor Variables for Form Fields in the *Virtuoso Schematic Editor User Guide*.

Example

```
schSymbolPinMasters = list(
    list("commActLo"
        list("output" list("basic" "commActLoOut"   "symbol")))
)
```

ieeeActLoInOut

Symbol View



Description

A symbol pin used when the pin type `ieeActLo` is selected with a direction of `inputOutput` on the Create Pin — Symbol form. There are also `input` and `output` direction pins available for the pin type `ieeActLo`.

The `schSymbolPinMasters` variable defines the symbol pins that are associated with the specific pin types and directions.

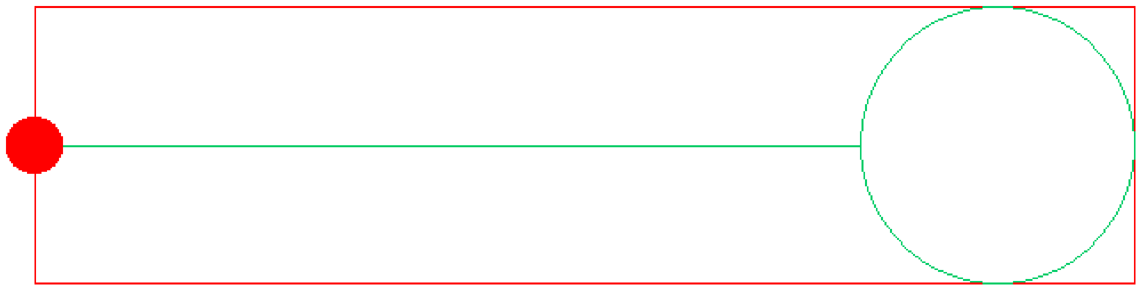
For more details, see Customizing Global Editor Variables for Form Fields in the *Virtuoso Schematic Editor User Guide*.

Example

```
schSymbolPinMasters = list(
    list("ieeeActLo"
        list("inputOutput" list("basic" "ieeeActLoInOut" "symbol")))
)
```

ieeeActLoInp

Symbol View



Description

A symbol pin used when the pin type `ieeActLo` is selected with a direction of `input` on the Create Pin — Symbol form. There are also `output` and `inputOutput` direction pins available for the pin type `ieeActLo`.

The `schSymbolPinMasters` variable defines the symbol pins that are associated with the specific pin types and directions.

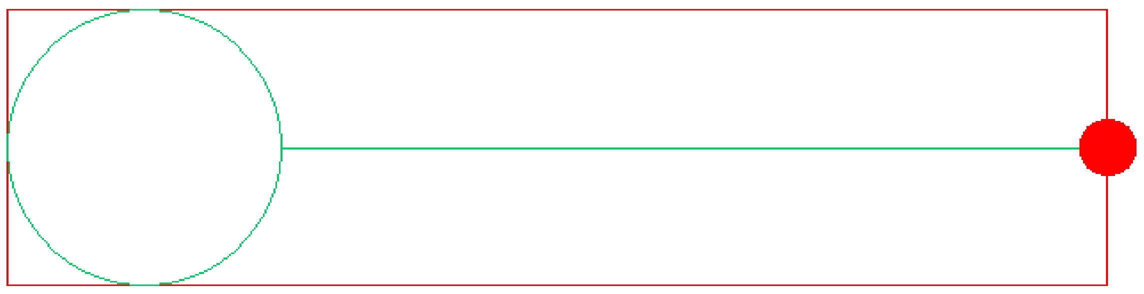
For more details, see Customizing Global Editor Variables for Form Fields in the *Virtuoso Schematic Editor User Guide*.

Example

```
schSymbolPinMasters = list(
    list("ieeeActLo"
        list("input" list("basic" "ieeeActLoInp" "symbol")))
)
```

ieeeActLoOut

Symbol View



Description

A symbol pin used when the pin type `ieeActLo` is selected with a direction of `output` on the Create Pin — Symbol form. There are also `input` and `inputOutput` direction pins available for the pin type `ieeActLo`.

The `schSymbolPinMasters` variable defines the symbol pins that are associated with the specific pin types and directions.

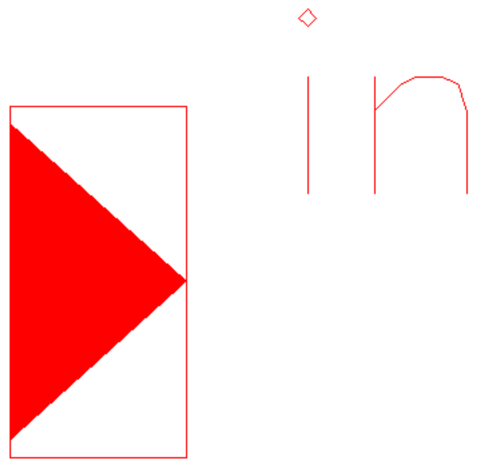
For more details, see Customizing Global Editor Variables for Form Fields in the *Virtuoso Schematic Editor User Guide*.

Example

```
schSymbolPinMasters = list(
    list("ieeeActLo"
        list("output" list("basic" "ieeeActLoOut" "symbol")))
)
```

in

Symbol View



Description

A symbol pin used to represent the `input` terminals to which a net connects in a schematic. It is one of the pin masters available when you choose the *Create – Pin* command in Virtuoso Schematic Editor. There are also `output` and `inputOutput` direction pins available.

Related Topics

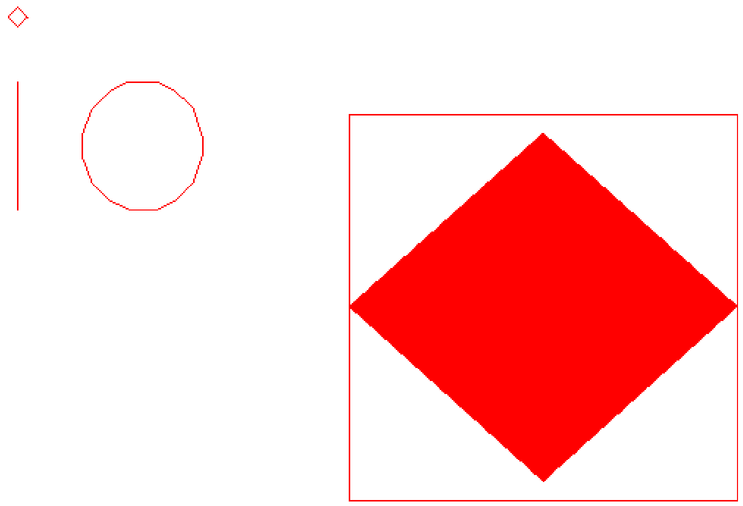
See the following topics in the *Virtuoso Schematic Editor User Guide*:

Adding Pins

Create Pin — Schematic

io

Symbol View



Description

A symbol pin used to represent the `inputOutput` terminals to which a net connects in a schematic. It is one of the pin masters available when you choose the *Create – Pin* command in Virtuoso Schematic Editor. There are also `input` and `output` direction pins available.

Related Topics

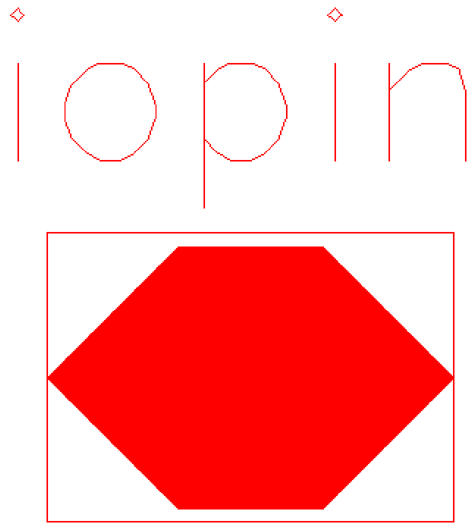
See the following topics in the *Virtuoso Schematic Editor User Guide:*

Adding Pins

Create Pin — Schematic

iopin

Symbol View



Description

A symbol pin used to represent the `inputOutput` direction. It is one of the symbol pin masters available when you choose the *Create — Pin* command in Virtuoso Schematic Editor. There are also `input` and `output` direction pins available.

Related Topics

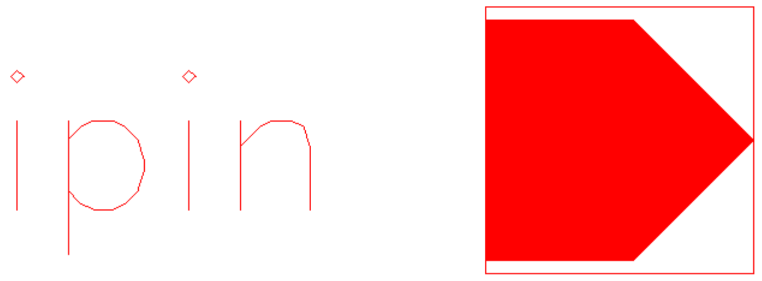
See the following topics in the *Virtuoso Schematic Editor User Guide:*

Adding Pins

Create Pin — Symbol

ipin

Symbol View



Description

A symbol pin used to represent the `input` direction. It is one of the symbol pin masters available when you choose the *Create — Pin* command in Virtuoso Schematic Editor. There are also `output` and `inputOutput` direction pins available.

Related Topics

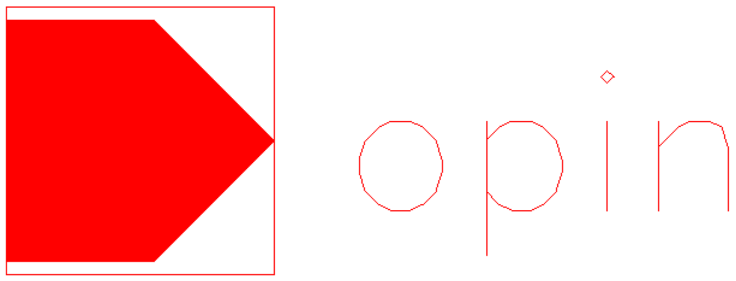
See the following topics in the *Virtuoso Schematic Editor User Guide:*

Adding Pins

Create Pin — Symbol

opin

Symbol View



Description

A symbol pin used to represent the `output` direction. It is one of the symbol pin masters available when you choose the *Create — Pin* command in Virtuoso Schematic Editor. There are also `input` and `inputOutput` direction pins available.

Related Topics

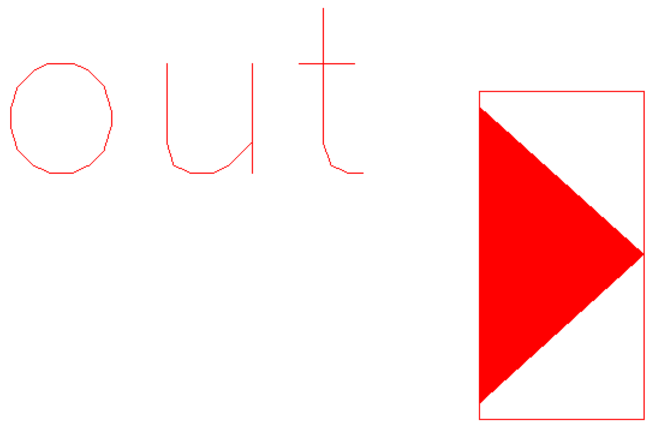
See the following topics in the *Virtuoso Schematic Editor User Guide:*

Adding Pins

Create Pin — Symbol

out

Symbol View



Description

This pin represents the `output` terminals to which a net connects in a schematic. It is one of the pin masters available when you choose the *Create – Pin* command. There are also `input` and `inputOutput` direction pins available.

Related Topics

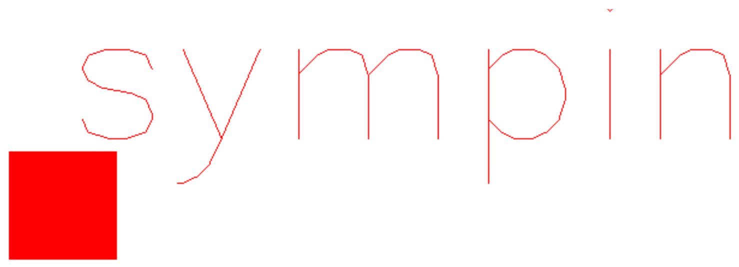
See the following topics in the *Virtuoso Schematic Editor User Guide:*

Adding Pins

Create Pin — Schematic

sympin

Symbol View



Description

This symbol pin represents the `square` pin type. It is one of the symbol pin masters available when you choose the *Create – Pin* command.

The `schSymbolPinMasters` variable defines the symbol pins that are associated with the specific pin direction and type. The `tsgConnectorMasters` variable defines the symbol of the pin connector.

For more details, see Customizing Global Editor Variables for Form Fields in the *Virtuoso Schematic Editor User Guide*.

Example

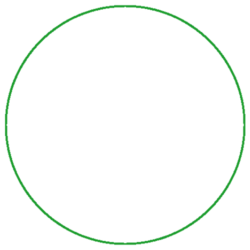
```
schSymbolPinMasters = list(
    list("square"
        list("input"       list("basic" "sympin" "symbolNN"))
        list("output"      list("basic" "sympin" "symbolNN"))
        list("inputOutput" list("basic" "sympin" "symbolNN"))
        list("switch"      list("basic" "sympin" "symbolNN"))
        list("jumper"      list("basic" "sympin" "symbolNN"))
        list("tristate"    list("basic" "sympin" "symbolNN"))
        list("unused"      list("basic" "sympin" "symbolNN")))
)
```

Related Topics

Create Pin — Symbol in the *Virtuoso Schematic Editor User Guide*

tsgActLo

Symbol View



Description

A pin graphic master for `actLo`. The pin graphic is selected using the appropriate *Attributes - List* option in the Symbol Generation Options form.

The `tsgActLo` pin graphic is available from the `tsgPinGraphicMasters` variable, which defines the characteristics of the graphic when generating symbols. This information is not used by any other tool.

The text-to-symbol generator (TSG) is a Cadence application program that automatically generates symbol cellviews for the Virtuoso Schematic Editor and subsequent simulation processes. TSG provides a quick way to generate a symbol from a list of pins in a TSG file.

Example

```
tsgPinGraphicMasters = list(
    list("actLo"
        list("input"        list("basic" "tsgActLo"   "symbol"))
        list("output"       list("basic" "tsgActLo"   "symbol"))
        list("inputOutput"  list("basic" "tsgActLo"   "symbol"))
        list("switch"       list("basic" "tsgActLo"   "symbol"))
        list("jumper"       list("basic" "tsgActLo"   "symbol"))
        list("tristate"     list("basic" "tsgActLo"   "symbol"))
        list("unused"       list("basic" "tsgActLo"   "symbol")))
)
```

Related Topics

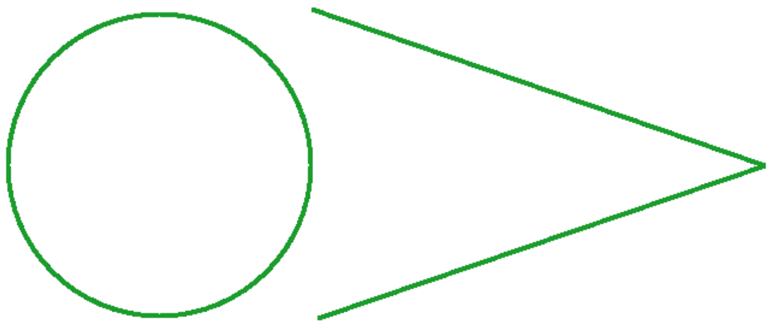
See the following topics in the *Virtuoso Schematic Editor User Guide:*

Text-to-Symbol Generator

Editing Symbol Generation Options

tsgActLoClock

Symbol View



Description

A pin graphic master for `actLoClock`. The pin graphic is selected using the appropriate *Attributes - List* option in the Symbol Generation Options form.

The `tsgActLoClock` pin graphic s available from the `tsgPinGraphicMasters` variable, which defines the characteristics of the graphic when generating symbols. It displays an additional pin shape on the symbol. This information is not used by any other tool.

The text-to-symbol generator (TSG) is a Cadence application program that automatically generates symbol cellviews for the Virtuoso Schematic Editor and subsequent simulation processes. TSG provides a quick way to generate a symbol from a list of pins in a TSG file.

Example

```
tsgPinGraphicMasters = list(
    list("actLoClock"
        list("input"        list("basic" "tsgActLoClock"  "symbol"))
        list("output"       list("basic" "tsgActLoClock"  "symbol"))
        list("inputOutput"  list("basic" "tsgActLoClock"  "symbol")))
)
```

Related Topics

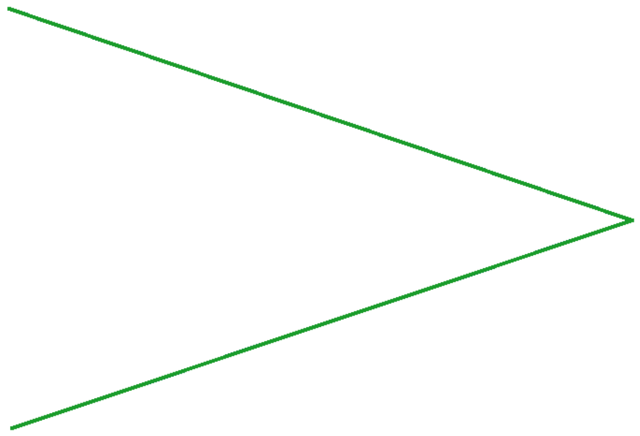
See the following topics in the *Virtuoso Schematic Editor User Guide:*

Text-to-Symbol Generator

Editing Symbol Generation Options

tsgClock

Symbol View



Description

A pin graphic master for `clock`. The pin graphic is selected using the appropriate *Attributes - List* option in the Symbol Generation Options form.

The `tsgClock` pin graphic is available from the `tsgPinGraphicMasters` variable, which defines the characteristics of the graphic when generating symbols. It displays an additional pin shape on the symbol. This information is not used by any other tool.

The text-to-symbol generator (TSG) is a Cadence application program that automatically generates symbol cellviews for the Virtuoso Schematic Editor and subsequent simulation processes. TSG provides a quick way to generate a symbol from a list of pins in a TSG file.

Example

```
tsgPinGraphicMasters = list(
    list("clock"
        list("input"        list("basic" "tsgClock"  "symbol"))
        list("output"       list("basic" "tsgClock"  "symbol"))
        list("inputOutput"  list("basic" "tsgClock" "symbol")))
)
```

Related Topics

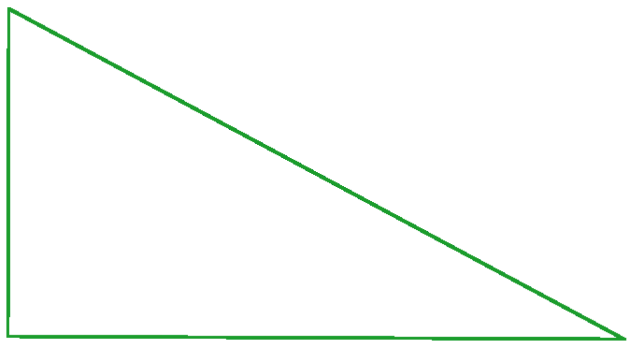
See the following topics in the *Virtuoso Schematic Editor User Guide:*

Text-to-Symbol Generator

Editing Symbol Generation Options

tsgIeeeActLoInp

Symbol View



Description

A pin graphic master for `ieeeActLoInp`. The pin graphic is selected using the appropriate *Attributes - List* option in the Symbol Generation Options form.

The `tsgIeeeActLoInp` pin graphic is available from the `tsgPinGraphicMasters` variable, which defines the characteristics of the graphic when generating symbols. It displays an additional pin shape on the symbol. This information is not used by any other tool.

The text-to-symbol generator (TSG) is a Cadence application program that automatically generates symbol cellviews for the Virtuoso Schematic Editor and subsequent simulation processes. TSG provides a quick way to generate a symbol from a list of pins in a TSG file.

Example

```
tsgPinGraphicMasters = list(
    list("ieeeActLo"
        list("input"        list("basic" "tsgIeeeActLoInp"  "symbol")))
)
```

Related Topics

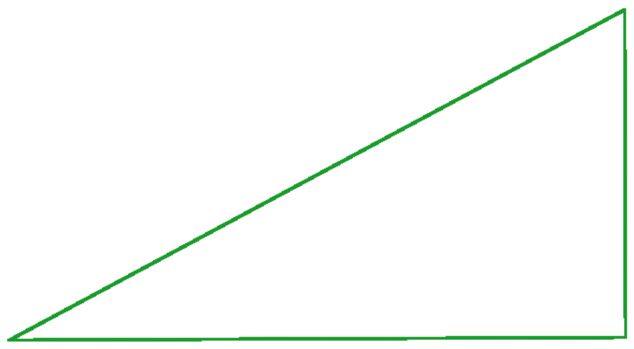
See the following topics in the *Virtuoso Schematic Editor User Guide:*

Text-to-Symbol Generator

Editing Symbol Generation Options

tsgIeeeActLoOut

Symbol View



Description

A pin graphic master for `ieeeActLoOut`. The pin graphic is selected using the appropriate *Attributes - List* option in the Symbol Generation Options form.

The `tsgIeeeActLoOut` pin graphic is available from the `tsgPinGraphicMasters` variable, which defines the characteristics of the graphic when generating symbols. It displays an additional pin shape on the symbol. This information is not used by any other tool.

The text-to-symbol generator (TSG) is a Cadence application program that automatically generates symbol cellviews for the Virtuoso Schematic Editor and subsequent simulation processes. TSG provides a quick way to generate a symbol from a list of pins in a TSG file.

Example

```
tsgPinGraphicMasters = list(
    list("ieeeActLo"
        list("output"       list("basic" "tsgIeeeActLoOut"  "symbol")))
)
```

Related Topics

See the following topics in the *Virtuoso Schematic Editor User Guide:*

Text-to-Symbol Generator

Editing Symbol Generation Options

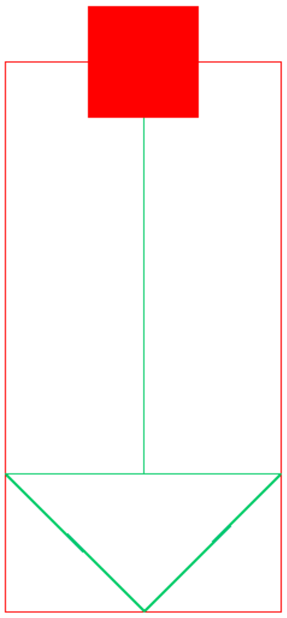
Supplies

The *Supplies* category lists the power and ground supply symbols. It includes the following symbols:

- gnd
- gnd_inherit
- vcc
- vcc_inherit
- vdd
- vdd_inherit
- vss
- vss_inherit

gnd

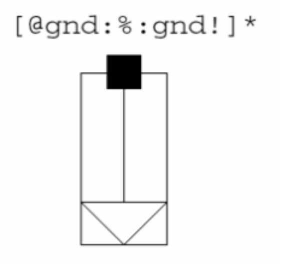
Symbol View



Description

A ground supply symbol used to indicate an explicit ground pin.

Example



Related Topics

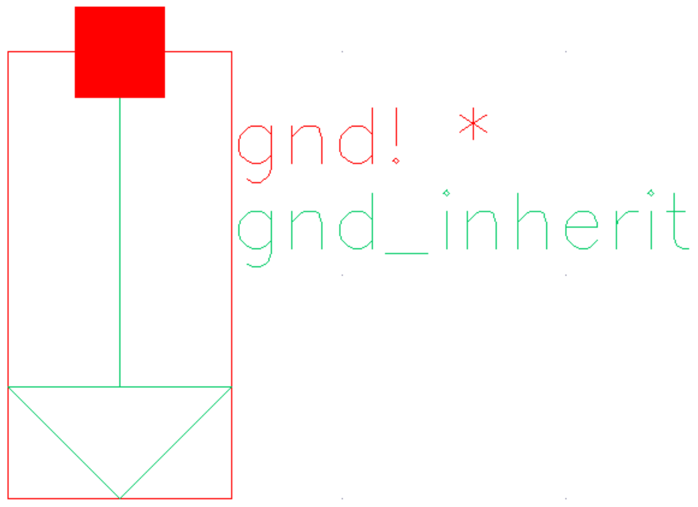
See the following topics in the *Virtuoso Schematic Editor User Guide:*

Global Net Name Connections

Inherited Connections

gnd_inherit

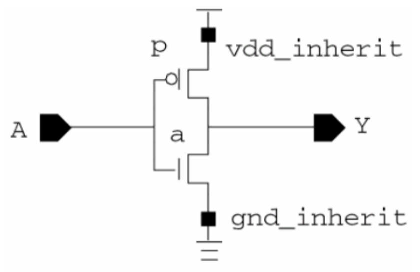
Symbol View



Description

A parameterized ground supply symbol with net expression property. It can be used to create global signals or override them across a hierarchy with inherited connectivity.

Example



You can create an inherited connection in a schematic by placing an instance of a symbol where one of the symbol pins has a net expression label. When you run the checker program on the schematic, the net expression label from the symbol pin is propagated onto the net within the schematic.

Related Topics

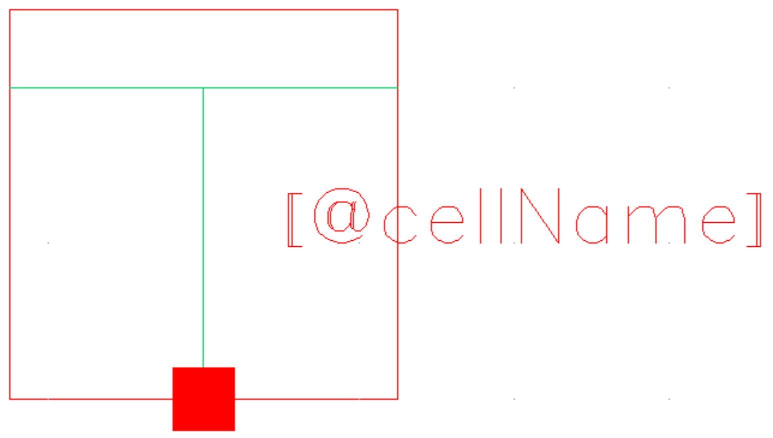
See the following topics in the *Virtuoso Schematic Editor User Guide:*

Global Net Name Connections

Inherited Connections

vcc

Symbol View



Description

A power supply symbol used to indicate an explicit power pin.

Related Topics

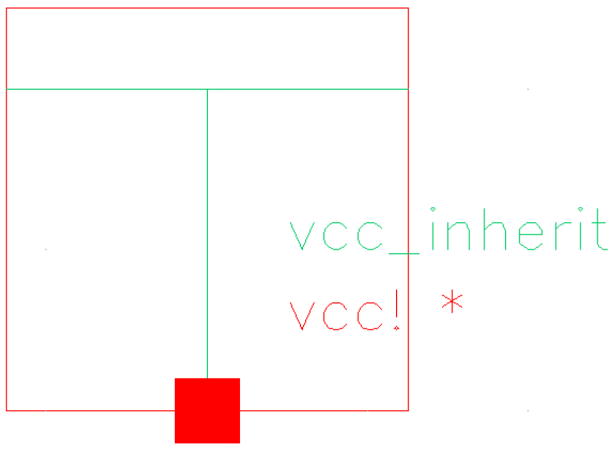
See the following topics in the *Virtuoso Schematic Editor User Guide:*

Global Net Name Connections

Inherited Connections

vcc_inherit

Symbol View



Description

Parameterized power supply symbol with net expression property. It can be used to create global signals or override them across a hierarchy with inherited connectivity.

Related Topics

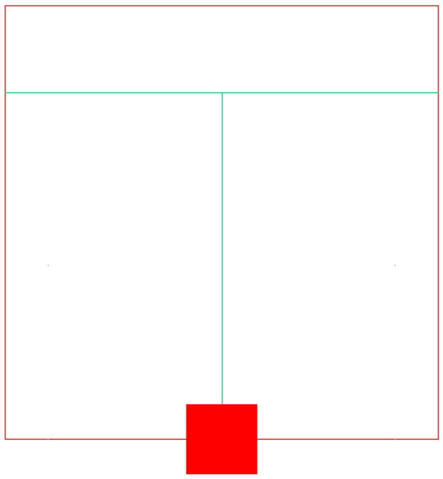
See the following topics in the *Virtuoso Schematic Editor User Guide:*

Global Net Name Connections

Inherited Connections

vdd

Symbol View



Description

A power supply symbol used to indicate an explicit power pin.

Related Topics

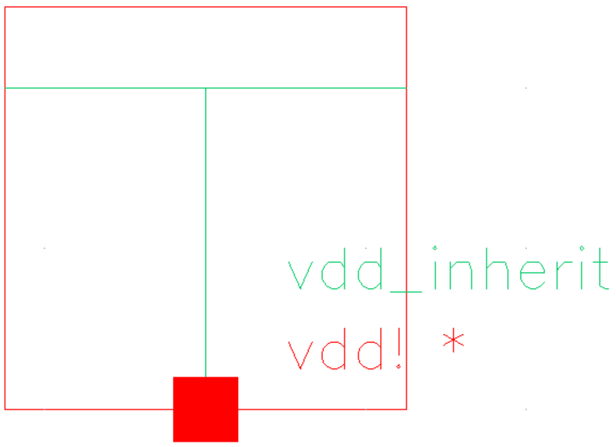
See the following topics in the *Virtuoso Schematic Editor User Guide:*

Global Net Name Connections

Inherited Connections

vdd_inherit

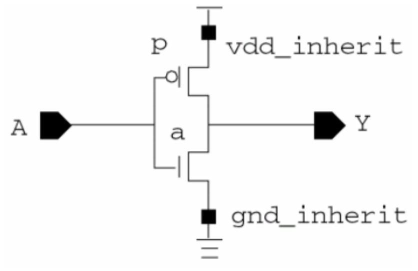
Symbol View



Description

A parameterized power supply symbol with net expression property. It can be used to create global signals or override them across a hierarchy with inherited connectivity.

Example



Related Topics

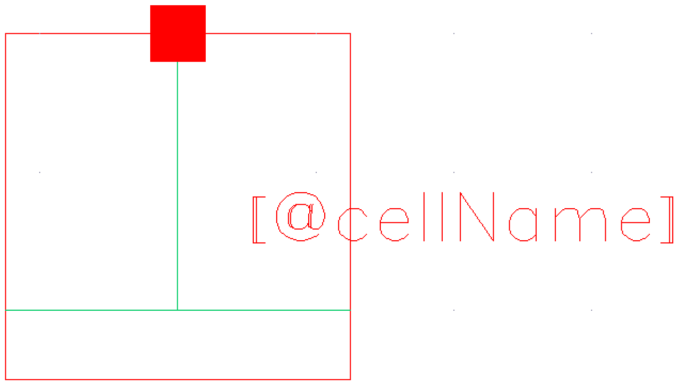
See the following topics in the *Virtuoso Schematic Editor User Guide:*

Global Net Name Connections

Inherited Connections

vss

Symbol View



Description

A ground supply symbol used to indicate an explicit ground pin.

Related Topics

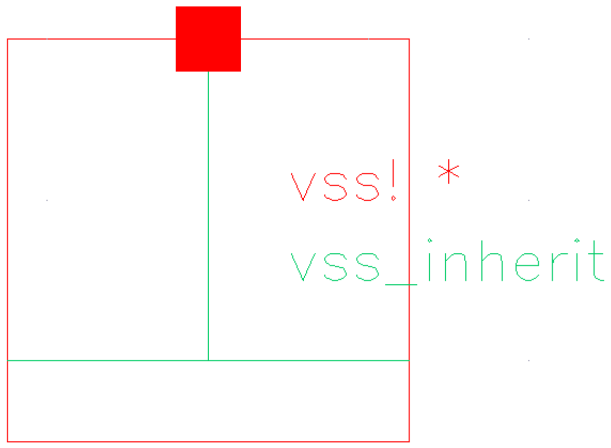
See the following topics in the *Virtuoso Schematic Editor User Guide:*

Global Net Name Connections

Inherited Connections

vss_inherit

Symbol View



Description

A parameterized ground supply symbol with net expression property. It can be used to create global signals or override them across a hierarchy with inherited connectivity.

Related Topics

See the following topics in the *Virtuoso Schematic Editor User Guide:*

Global Net Name Connections

Inherited Connections

VHDLPins

Within Virtuoso, you can convert a VHSIC Hardware Description Language (VHDL) structural or behavioral description into one of the following forms in OpenAccess database storage format:

- Schematic view
- Netlist view
- VHDL text views

For more details, refer to the VHDL In for Virtuoso Design Environment User Guide and Reference.

When generating schematics or netlists from VHDL, the following varieties of VHDL pins are available in the `basic` library:

vhdlActHiInOut
vhdlActHiInp
vhdlActHiOut
vhdlBlockIOPin
vhdlBlockIPin
vhdlBlockOpin
vhdlCircle
vhdlCommActLoInOut
vhdlCommActLoInp
vhdlCommActLoOut
vhdlIOPin
vhdlIPin
vhdlIeeeActLoInOut
vhdlIeeeActLoInp
vhdlIeeeAcLoOut
vhdlOPin
vhdlSymPin