cadence®

Virtuoso Symbolic Placement of Devices User Guide

Product Version IC23.1
June 2023

© 2023 Cadence Design Systems, Inc.
Printed in the United States of America.

Cadence Design Systems, Inc. (Cadence), 2655 Seely Ave., San Jose, CA 95134, USA.

Open SystemC, Open SystemC Initiative, OSCI, SystemC, and SystemC Initiative are trademarks or registered trademarks of Open SystemC Initiative, Inc. in the United States and other countries and are used with permission.

Trademarks: Trademarks and service marks of Cadence Design Systems, Inc. contained in this document are attributed to Cadence with the appropriate symbol. For queries regarding Cadence's trademarks, contact the corporate legal department at the address shown above or call 800.862.4522.

All other trademarks are the property of their respective holders.

Restricted Permission: This publication is protected by copyright law and international treaties and contains trade secrets and proprietary information owned by Cadence. Unauthorized reproduction or distribution of this publication, or any portion of it, may result in civil and criminal penalties. Except as specified in this permission statement, this publication may not be copied, reproduced, modified, published, uploaded, posted, transmitted, or distributed in any way, without prior written permission from Cadence. Unless otherwise agreed to by Cadence in writing, this statement grants Cadence customers permission to print one (1) hard copy of this publication subject to the following conditions:

1. The publication may be used only in accordance with a written agreement between Cadence and its customer.
2. The publication may not be modified in any way.
3. Any authorized copy of the publication or portion thereof must include all original copyright, trademark, and other proprietary notices and this permission statement.
4. The information contained in this document cannot be used in the development of like products or software, whether for internal or external use, and shall not be used for the benefit of any other party, whether or not for consideration.

Disclaimer: Information in this publication is subject to change without notice and does not represent a commitment on the part of Cadence. Except as may be explicitly set forth in such agreement, Cadence does not make, and expressly disclaims, any representations or warranties as to the completeness, accuracy or usefulness of the information contained in this document. Cadence does not warrant that use of such information will not infringe any third party rights, nor does Cadence assume any liability for damages or costs of any kind that may result from use of such information.

Cadence is committed to using respectful language in our code and communications. We are also active in the removal and replacement of inappropriate language from existing content. This product documentation may however contain material that is no longer considered appropriate but still reflects long-standing industry terminology. Such content will be addressed at a time when the related software can be updated without end-user impact.

Restricted Rights: Use, duplication, or disclosure by the Government is subject to restrictions as set forth in FAR52.227-14 and DFAR252.227-7013 et seq. or its successor

Contents

© 2023 Cadence Design Systems, Inc.

© 2023 Cadence Design Systems, Inc.

© 2023 Cadence Design Systems, Inc.

© 2023 Cadence Design Systems, Inc.

© 2023 Cadence Design Systems, Inc.

© 2023 Cadence Design Systems, Inc.

1

Introduction to Symbolic Placement of Devices

Virtuoso® Symbolic Placement of Devices (SPD) is a symbolic row-based placer that displays only the necessary information required to perform device placement in the physical design environment.

The main focus of this tool is to improve productivity of the layout designers by facilitating quick and easy placement of PMOS and NMOS devices in medium-sized designs comprising around 100 to 200 devices. The tool makes it simple to edit device placement, preview the updates made, and then generate a layout.

This user guide is aimed at layout engineers who need to place devices with a fast turnaround time and assumes that you are familiar with:

- The Virtuoso Studio design environment and application infrastructure mechanisms supporting consistent operations between all Cadence tools.
- The applications for designing and developing integrated circuits in the Virtuoso Studio design environment, notably the Virtuoso Layout Suite L, Virtuoso Layout Suite XL, and layout editor.
- Virtuoso technology data.
- Component description format (CDF), which lets you create and describe your own components for use with Layout XL.

Licensing Requirements

- You can launch SPD from either the VLS EXL or VLS MXL cockpit.
- In Layout XL, SPD checks out a further two GXL tokens in addition to the licenses already checked out.

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

© 2023 Cadence Design Systems, Inc.

This section covers the following topics:

- Benefits of using SPD
- The SPD Flow
- Prerequisites for Running SPD
  - Source Schematic File
  - Configured Physical Hierarchy of Target Layout
  - Component Types
  - Parameter Name Mapping
  - Permute Rules
  - Layer Functions and Minimum Spacing Rules
  - Libraries Defined to Execute SPD
  - User-Defined Abutment Callback Functions
  - User-Defined Flow Callback Functions

© 2023 Cadence Design Systems, Inc.

Benefits of using SPD

SPD uses the intuitive ‘stick-diagram’ methodology that lets the layout designers visualize a lighter layout, which is easier to understand and work with.

The tool provides three main benefits:

- **Symbolic**: Symbolically-represented devices are easier to understand and manipulate than real ones.
- **Simplicity**: The simple use model of the SPD commands enables device placement in just a few mouse clicks.
- **Speed**: Editing in SPD is much faster than editing real devices on the main layout canvas.

  Due to the growing complexity of design rules for real devices, heavy computations are often involved starting from pcell evaluation to device placement. Several applications also perform complicated post-processing to ensure that every step is free of any DRC errors. SPD helps to avoid most of these tasks during editing; the complexity comes in only during preview when converting symbolic devices to real ones.

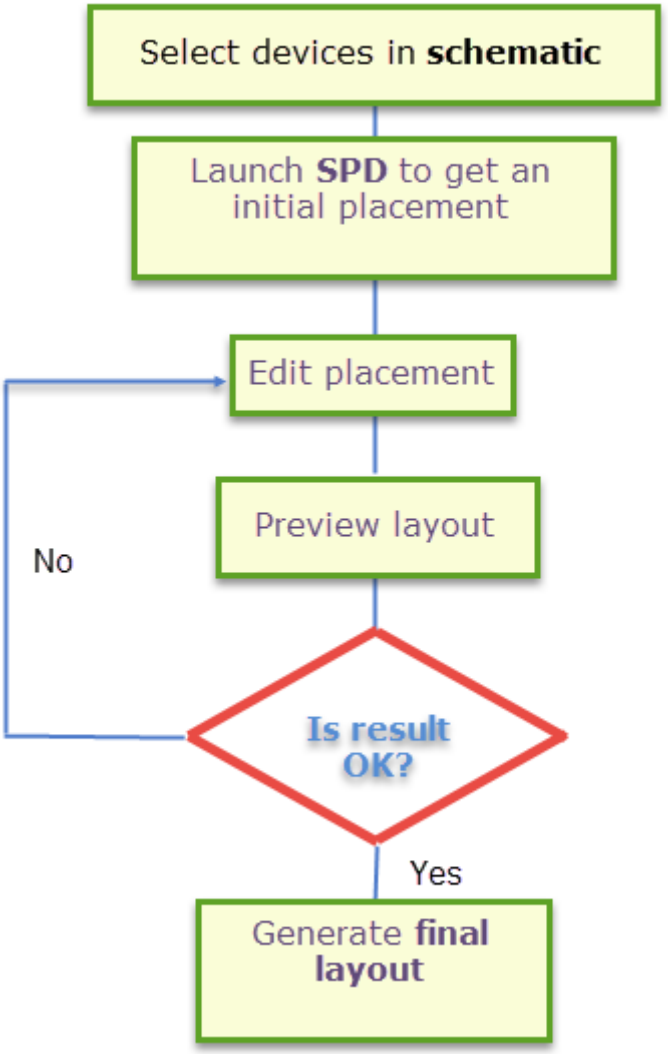
The SPD Flow

The basic SPD flow, involves following tasks:

1. Select the devices you want to edit in Schematic XL.
2. Launch SPD to view the symbolic representation of the selected devices.
3. Place the selected devices to meet your requirements by performing tasks such as, abutting, swapping, stacking, and flipping devices.
4. Preview the edited layout.

© 2023 Cadence Design Systems, Inc.

5. Make any required adjustments to the placement in the SPD window.
6. When you are satisfied with the results, generate the layout.



For a video on SPD flow, see Getting Started with Symbolic Placement of Devices on Cadence Online Support.

Prerequisites for Running SPD

SPD can run only on a Layout XL compliant PDK library that includes well-defined component types, parameter name mappings, permute rules, layer functions, and minimum spacing rules. This section discusses prerequisites for running SPD in detail.

© 2023 Cadence Design Systems, Inc.

Source Schematic File

You can launch SPD directly from an existing layout, however, the source schematic is always needed as a reference. If some instance parameters in the layout instances are different from those in the schematic, SPD will always take the values from the schematic.

Configured Physical Hierarchy of Target Layout

The SPD launch process refers to the physical configuration (CPH) of the target layout and carries it to the next session when the *Reuse Window for Faster Relaunch* option is enabled in Setup Options Pane.

If for any reason, the setting for target layout is modified in CPH, you must disable the *Reuse Window for Faster Relaunch* option and re-launch SPD for the new physical configuration take effect.

Component Types

SPD supports the following component types for cells. These component types must be set before running SPD.

- PMOS
- NMOS
- PFIN
- NFIN

To check component types, in the Layout XL window, choose *Edit – Component Types* to display the Configure Physical Hierarchy window in Component Types mode.

For more information on defining and assigning component types, see Configuring the Physical Hierarchy, in the *Virtuoso Layout Suite XL User Guide*.

Parameter Name Mapping

To run SPD, you need to:

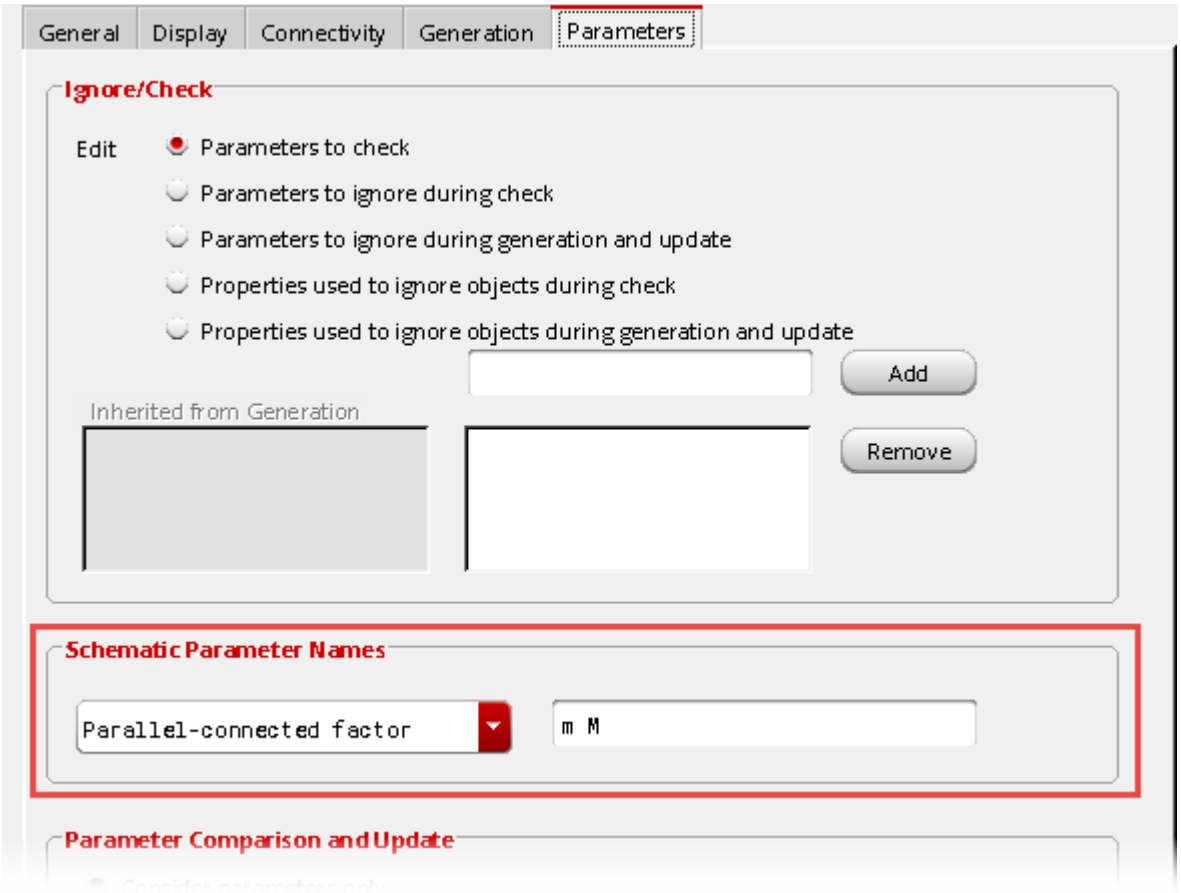
- Set width parameter name.
- Set finger parameter name.
- Set multiplication factor parameter name.

© 2023 Cadence Design Systems, Inc.

- Check the Evaluate CDF callbacks during generation and update option.

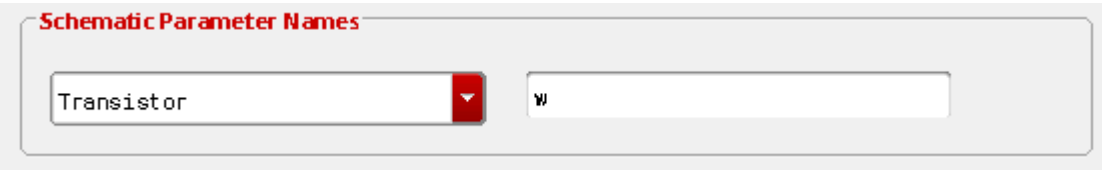
To check the these parameters, in the Layout XL window:

1. Choose *Options – Layout XL – Parameters.*



2. Set width parameter name.

   - In the *Schematic Parameter Names* group box, choose *Transistor* from the drop-down list and ensure that the correct `width` value is set.



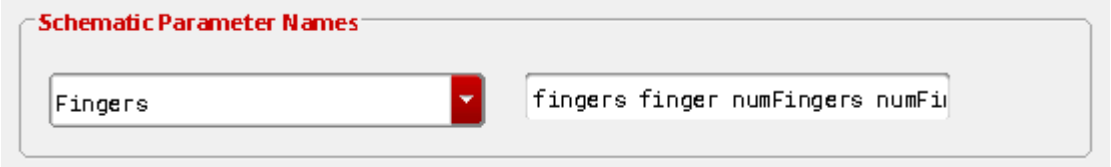
   - Use the following environment variable:

```
envSetVal("layoutXL" "transistorWidthParamNames" 'string "width_names")
```


© 2023 Cadence Design Systems, Inc.

3. Set finger parameter name.

   - In the *Schematic Parameter Names* group box, choose *Fingers* from the drop-down list and ensure that correct `Fingers Text` combination value is set.



   - Use the following environment variable:

```
envSetVal("layoutXL" "lxFingeringNames" 'string "num_of_fingers_names")
```

4. Set multiplication factor parameter name.

   a. In the *Schematic Parameter Names* group box, choose *Parallel-connected factor Text* from the drop-down list and ensure that correct `Parallel-connected factor Text` combination value is set.

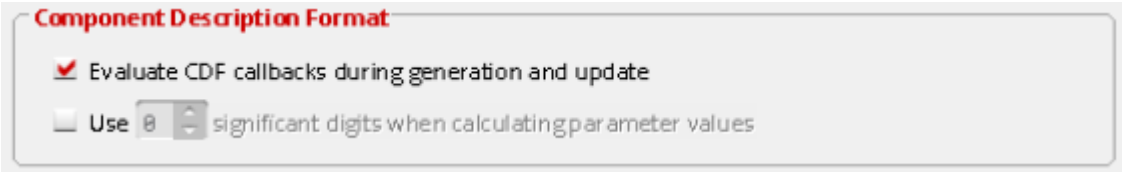
   - Use the following environment variable:

```
envSetVal("layoutXL" "mfactorNames" 'string "mfactor_names")
```

Note: Values set here should match the parameter names in the PDK libraries being used.

5. Check the Evaluate CDF callbacks during generation and update option.

   a. In the *Component Description Format* group box, check *Evaluate CDF callbacks* during generation and update option.



   - Use the following environment variable:

```
envSetVal("layoutXL" "lxEvalCDFCallbacks" 'boolean t)
```


© 2023 Cadence Design Systems, Inc.

Permute Rules

The permuteRule property must be defined for the instances in the design to make the instance pins or terminals permutable.

For information on how to check and add a permuteRule, see Checking XL Compliance and permuteRule in *Virtuoso Layout Suite XL User Guide*.

Layer Functions and Minimum Spacing Rules

To run SPD, certain material functions and their layers must be specified in the technology file. It is also important that minimum spacing rules are set correctly for these layers.

The following are mandatory functions that must be included in the technology file:

- `metal`
- `diff`
  - Layers of `diff` function must also be defined in the technology file.
  - If the minimum spacing rule of the layer is not available, you will have to adjust it manually in SPD by setting the value of the *X:OD* field in the Generation Options pane.

The following are optional functions that can be included in the technology file:

- `ndiff`
  - If the minimum spacing rule of the `ndiff` layer is not available, you can set it manually in SPD by setting the value of the *POD-NOD* field in the Generation Options pane.
- `pdiff`
  - If the minimum spacing rule of the `pdiff` layer is not available, you will have to adjust it manually in SPD by setting the value of the *POD-NOD* in the Generation Options pane.
- `poly`
  - If the minimum spacing rule of the `poly` layer is not available, you will have to adjust it manually in SPD by setting the value of the *Stack Spacing* field in the Stack Options pane.

© 2023 Cadence Design Systems, Inc.

- If the minimum spacing rule of the `poly` layer is not available, you may have incorrect placement results if you specify value of the *X: OD* option as `DummyPoly` in Generation Options pane.

- `nwell`
  - This value is required when the *Prevent well shorts* is set in the Generation Options pane.
- `pwell`
  - This value is required when the *Prevent well shorts* is set in the Generation Options pane.

For more information on defining layer functions, see Layer Rules in *Virtuoso Technology Data ASCII Files Reference*.

For more information on defining minimum spacing rules, see minSpacing in *Virtuoso Technology Data ASCII Files Reference*.

Libraries Defined to Execute SPD

SPD defines two libraries in cds.lib, `cdsSymbolicDevice` and `cdsSymbolicLib`.

cdsSymbolicDevice

The following entry is added to cds.lib:

```
DEFINE cdsSymbolicDevice $CDS_INST_DIR/tools/dfII/etc/cdslib/
cdsSymbolicDevice
```

Symbolic device is an instance that creates a relationship between schematic devices and the layout instance. This library is essential for SPD to work because it contains the symbolic devices used in SPD, which are compatible with the current tool version. This entry is not added to cds.lib by default. It is added automatically when SPD is launched.

cdsTmpSymbolicLib

The following entry is added to cds.lib:

```
DEFINE cdsTmpSymbolicLib ./.cadence/dfII/spd/cdsTmpSymbolicLib
```


© 2023 Cadence Design Systems, Inc.

This is the default temporary library required to process SPD. Information from the working schematic and layout cellview is copied to this library and then, used to build a relationship, such as in the CPH file.

This library gives user the write permission to save updates made in SPD or save recovery information in case of a crash.

In case, you don't want to use the default temporary library, you can specify a different library using the `spdDesignLibName` environment variable.

User-Defined Abutment Callback Functions

For PDK-specific abutments, such as dummy poly abutment, other than the regular oxide diffusion (OD) abutment, you will also need to define your own custom abutment callback functions. To enable user-defined abutment, you must load the callback functions required by the `spdRegUserAbutProc` SKILL function and register them before launching SPD. Otherwise, only oxide diffusion abutment will take place.

User-Defined Flow Callback Functions

SPD also supports user flow callbacks that further help in customizing the design in specific steps of the SPD flow. For more information, see User Flow Callback Functions in *Virtuoso Layout Suite SKILL Reference* and the environment variable launchRunOnce.

For a video on user-defined abutment and callback functions, see Using User-Defined Abutment and Callback Functions in SPD on Cadence Online Support.

© 2023 Cadence Design Systems, Inc.

2

Getting Started with SPD

This section covers the following topics:

© 2023 Cadence Design Systems, Inc.

- Exiting the SPD Editing Window

© 2023 Cadence Design Systems, Inc.

Generating a Symbolic Placement

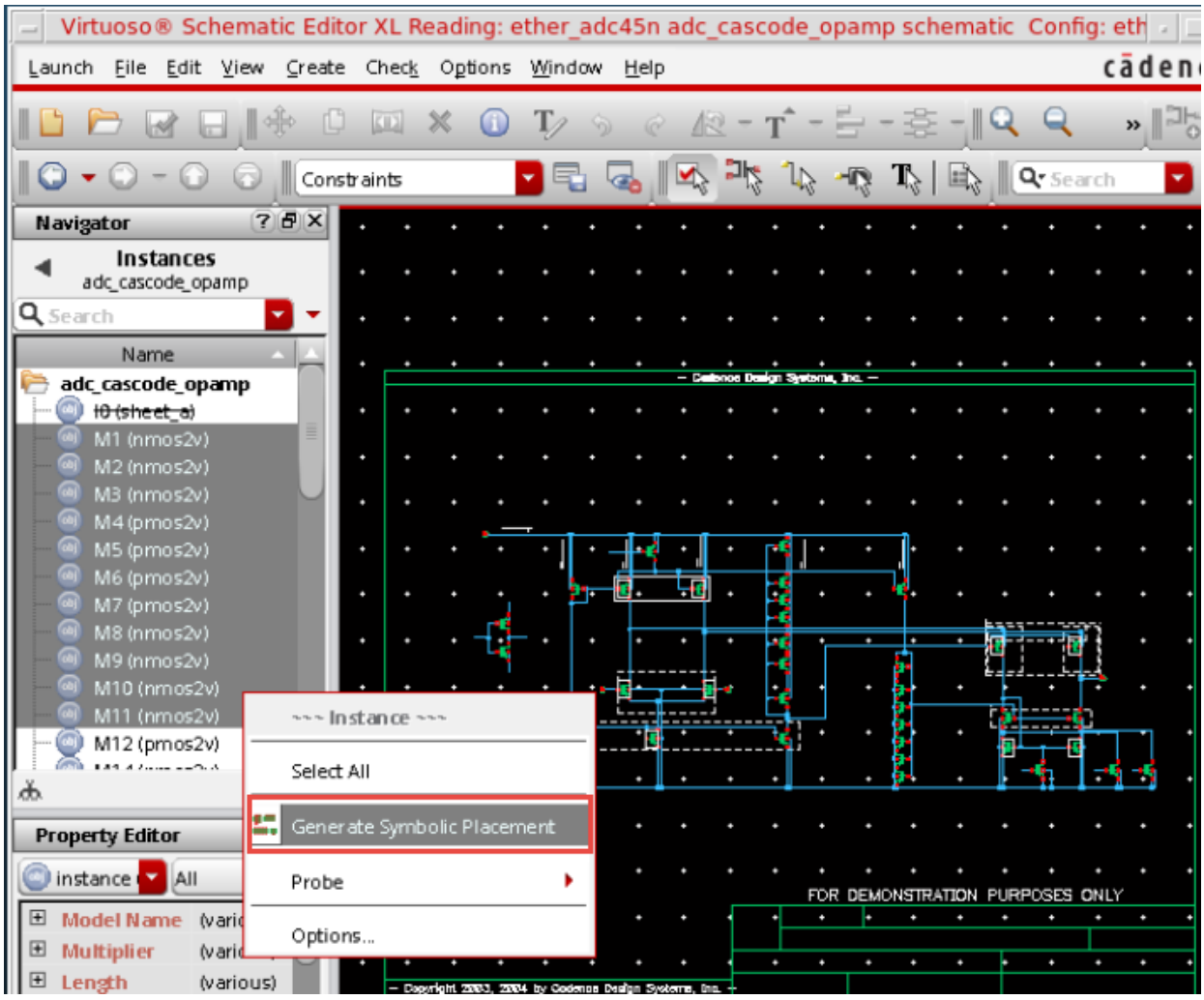
You can open a design in SPD directly from Schematics XL or from Layout XL:

Generating a Symbolic Placement from Schematics XL

a. Select instances of PMOS and NMOS devices in schematic for which you want to generate the symbolic design.

b. Right-click the selected devices in the Navigator assistant.

c. Choose *Generate Symbolic Placement* menu option.

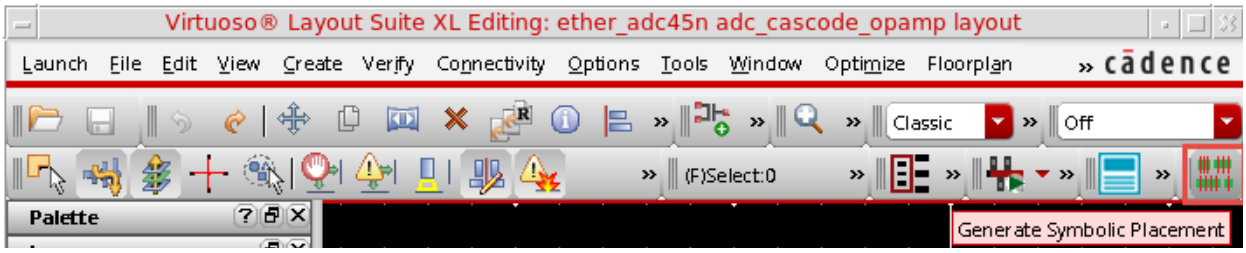


© 2023 Cadence Design Systems, Inc.

Generating a Symbolic Placement from Layout XL

a. Select instances of PMOS and NMOS devices in Layout XL for which you want to generate the symbolic design.

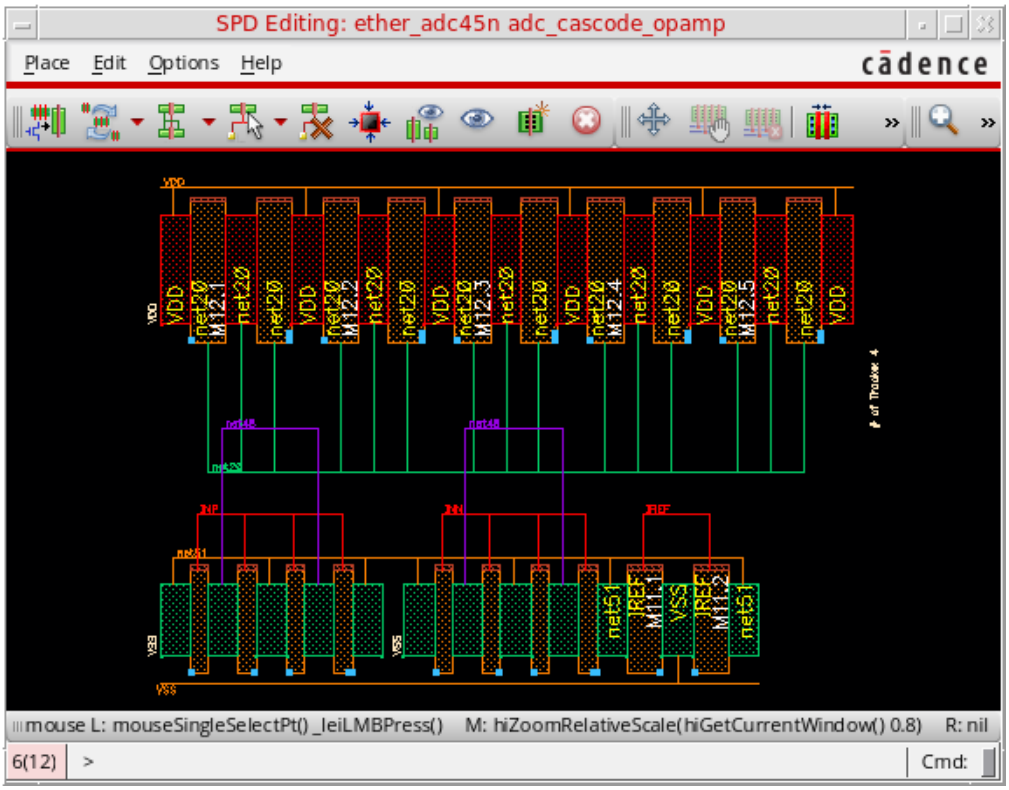
b. Click the *Generate Symbolic Placement* icon on the toolbar.



In case, this icon is not available on the toolbar, right-click in the toolbar area, and then select the *Symbolic Placement of Devices* menu option.

1. The SPD Editing window opens displaying the symbolic design of selected devices.

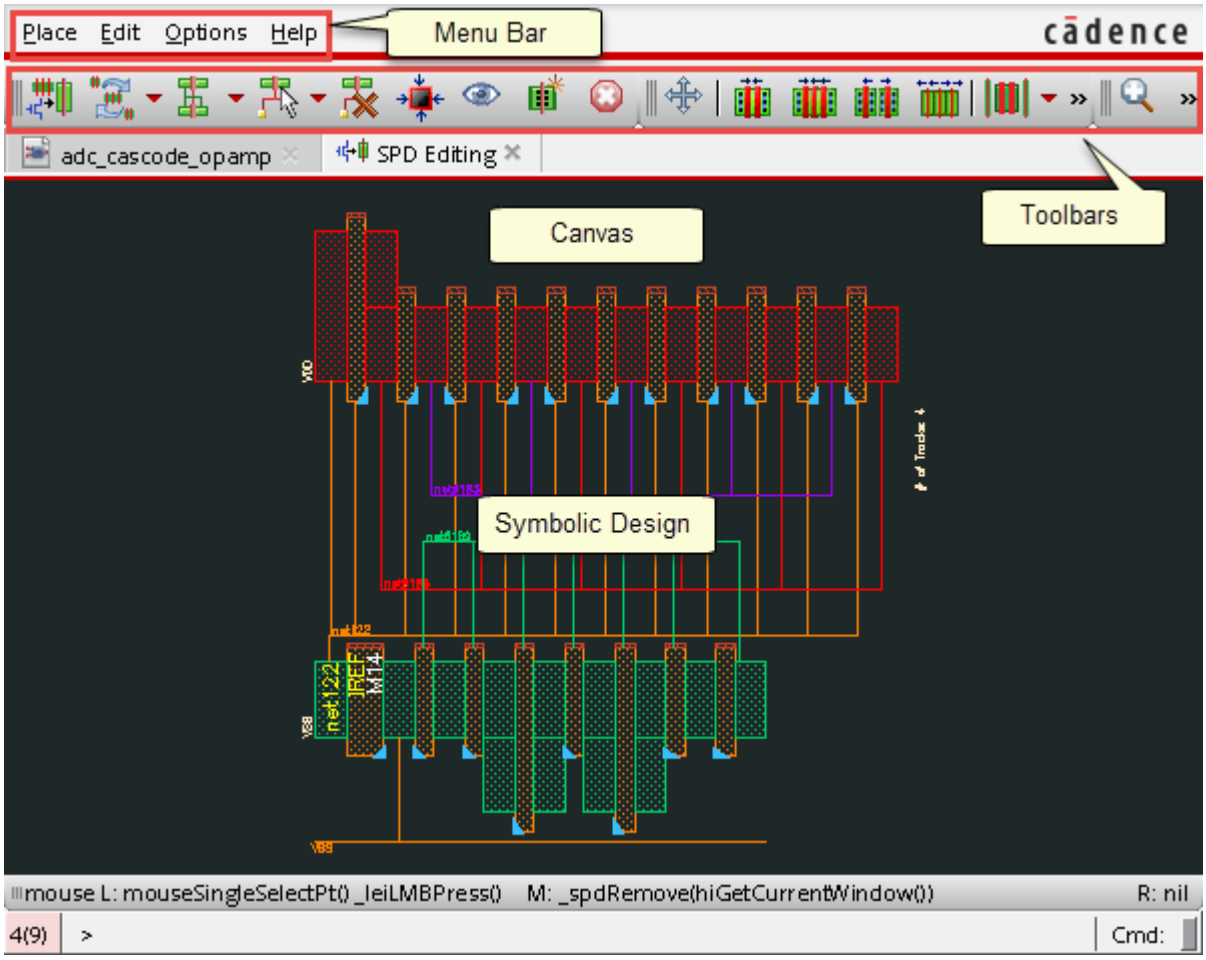
Note: If you did not select any device before launching SPD, the SPD Editing window is empty.



© 2023 Cadence Design Systems, Inc.

SPD Editing Window

The SPD Editing window comprises a menu bar, toolbars that let you use various SPD commands at a single-click and a canvas, in which you can display and manipulate your symbolic design graphically.



Menus

There are four menus in the SPD window. *Place*, *Edit*, and *Options* menus comprise various commands to enable you to perform various edit and placement tasks. The *Help* menu lets you access the help documentation.

© 2023 Cadence Design Systems, Inc.

Toolbars

SPD toolbars lets you access the main SPD commands directly without opening any menus. As with all toolbars, you can use the handle on the left hand side of the toolbar to reposition it anywhere within the layout window.

The Place Toolbar

The Place toolbar provides access to the following *Place* menu commands:

Icon	Command	Lets you...
	Append Selected from Source	Add new devices to the SPD design from the source schematic. See Appending Components from the Schematic for more information.
	Reset Placement *Reset Placement & Selection*	Discard any placement changes made to the symbolic design and revert to placement set in the SPD Options form. See Resetting Placement for more information.
	Flight Lines Both *Flight Lines Gate Only* *Flight Lines Active Only* *Flight Lines None*	Control how flight lines are displayed in the SPD canvas. See Displaying Flight Lines for more information.
	Probe Net *Probe & Align*	Select nets in the symbolic design to highlight the corresponding element in the schematic window. See Probing Selected Nets for more information.
	Clear All Probes	Clear all probes highlighted in the SPD canvas. See Probing Selected Nets for more information.

© 2023 Cadence Design Systems, Inc.

Icon	Command	Lets you...
	Pack	Remove empty spaces from the design. See Packing the Design for more information.
	Free Bucket Size *Off* *Free W* *Free L* *Gate Size Preview*	Toggle the free bucket size mode. See Displaying Free Bucket Size of Devices for more information.
	Preview Layout	Preview the physical layout of the symbolic design. See Previewing the Design for more information.
	Generate Layout	Generate the final layout and place it in the Layout XL canvas. See Generating the Layout for more information.
	Exit	Close the SPD Editing window. See Exiting the SPD Editing Window for more information.

The Edit Toolbar

The Edit toolbar provides access to the following *Edit* menu commands:

Icon	Command	Lets you...
	Move	Move symbolic devices from one location to another in the canvas. See Moving Symbolic Devices for more information.
	Reorder Trunks	Reorder trunks before generating the layout. See Reordering Trunks for more information.

© 2023 Cadence Design Systems, Inc.

Icon	Command	Lets you...
	*Reset Reorder Trunks*	Reset the modified trunk order. See Resetting Trunk Order for more information.
	*Abut*	Abut selected devices or device chains. See Abutting Devices for more information.
	*Abut All*	Abut all devices in the canvas. See Abutting All Devices for more information.
	*Unabut*	Unabut selected devices or device chains. See Unabutting Devices for more information.
	*Unabut All*	Unabut all abutted devices in the canvas. See Unabut All Devices for more information.
	*Add Dummy Both* Add Dummy Left Add Dummy Right	Add dummy devices to the design. See Adding Dummy Devices for more information.
	*Back Annotate All Dummy Instances*	Add the newly added dummy devices to schematic. See Backannotating Symbolic Devices for more information.
	*Undo*	Undo a command. See Undoing and Redoing Commands for more information.
	*Redo*	Reinstate a change that you canceled with *Undo*. See Undoing and Redoing Commands for more information.
	*Flip Horizontal*	Flip selected device or device chains horizontally. See Flipping Symbolic Devices for more information.

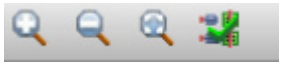
© 2023 Cadence Design Systems, Inc.

Icon	Command	Lets you...
	*Flip Vertical*	Flip selected device or device chains vertically. See Flipping Symbolic Devices for more information.
	*Permute*	Permutes the devices without changing the orientation. See Permuting Symbolic Devices for more information.
	*Swap*	Swap the positions of symbolic instances. See Swapping Symbolic Devices for more information.
	*Fold*	Divides symbolic instances into two or more layout instances such that their terminals are connected in parallel to the same nets. See Folding Symbolic Devices for more information.
	*Split Fingered Device*	Place each finger of a symbolic finger device as a separate device. See Splitting Fingered Devices for more information.
	*Remove*	Delete the selected device. See Removing Symbolic Devices for more information.
	*Align*	Align selected devices. See Aligning Symbolic Devices for more information.
	*Unalign*	Unalign selected devices. See Unaligning Devices for more information.
	*Check Align*	Identify misalignments in the symbolic design. See Checking Device Alignment for more information.

© 2023 Cadence Design Systems, Inc.

Icon	Command	Lets you...
	*Generate Chained Devices*	Create a device chain. See Generating Chained Devices for more information.
	*Stack*	Create a new stacked device. See Stacking a Device for more information.
	*Unstack*	Unstack a stacked device. See Unstacking a Stacked Device for more information.
	*Edit Stack*	Edit an existing stack. See Editing a Stacked Device for more information.

Other Toolbar Commands



Icon	Command	Lets you...
	*Zoom In*	Magnifies the image in the cellview window by a factor of 1.25. You can also scroll up the mouse scroll wheel to zoom in.
	*Zoom Out*	Reduces the image in the cellview window by a factor of 20.8. You can also scroll down the mouse scroll wheel to zoom out.
	*Zoom to Fit*	Redraws the window so that all objects in the cellview appear in the window. This saves time after you zoom or pan and want to see the entire cellview again
	*Check or Update*	Check the differences between the source and SPD layout, or update existing objects in the SPD window. See Checking Source or Updating Layout for more information.

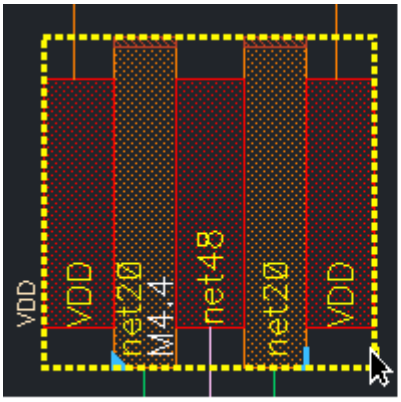
© 2023 Cadence Design Systems, Inc.

Canvas

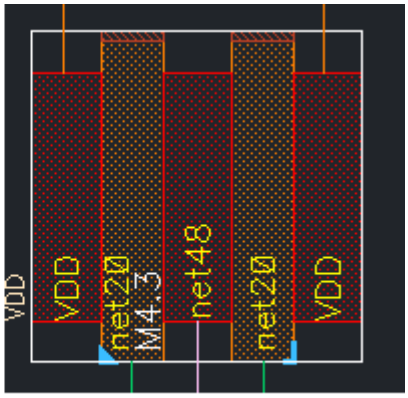
The canvas is always visible in the SPD editing window. It shows the symbolic design of the abstract devices arranged in rows. You can perform various device editing and placement tasks, such as abutting, folding, and stacking devices from here.

Highlighting and Selecting Data

Highlighting and selection of devices in SPD is the same as in Layout XL. Preselected devices are highlighted with dashed yellow line and selected devices are highlighted with white line.

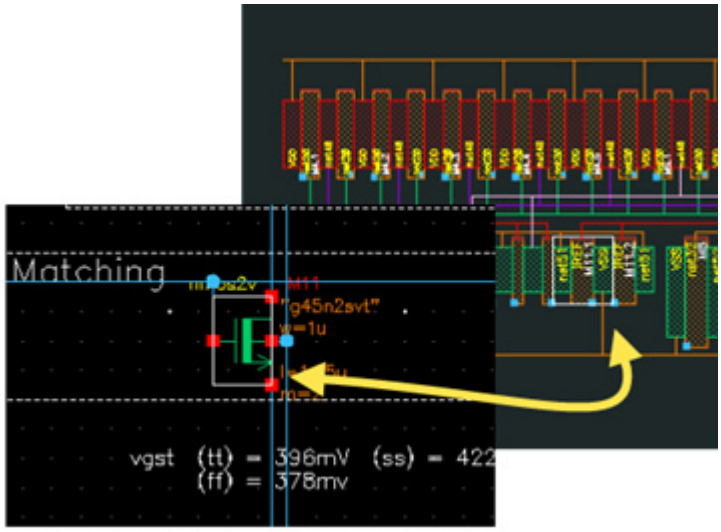


Preselected Device Highlight



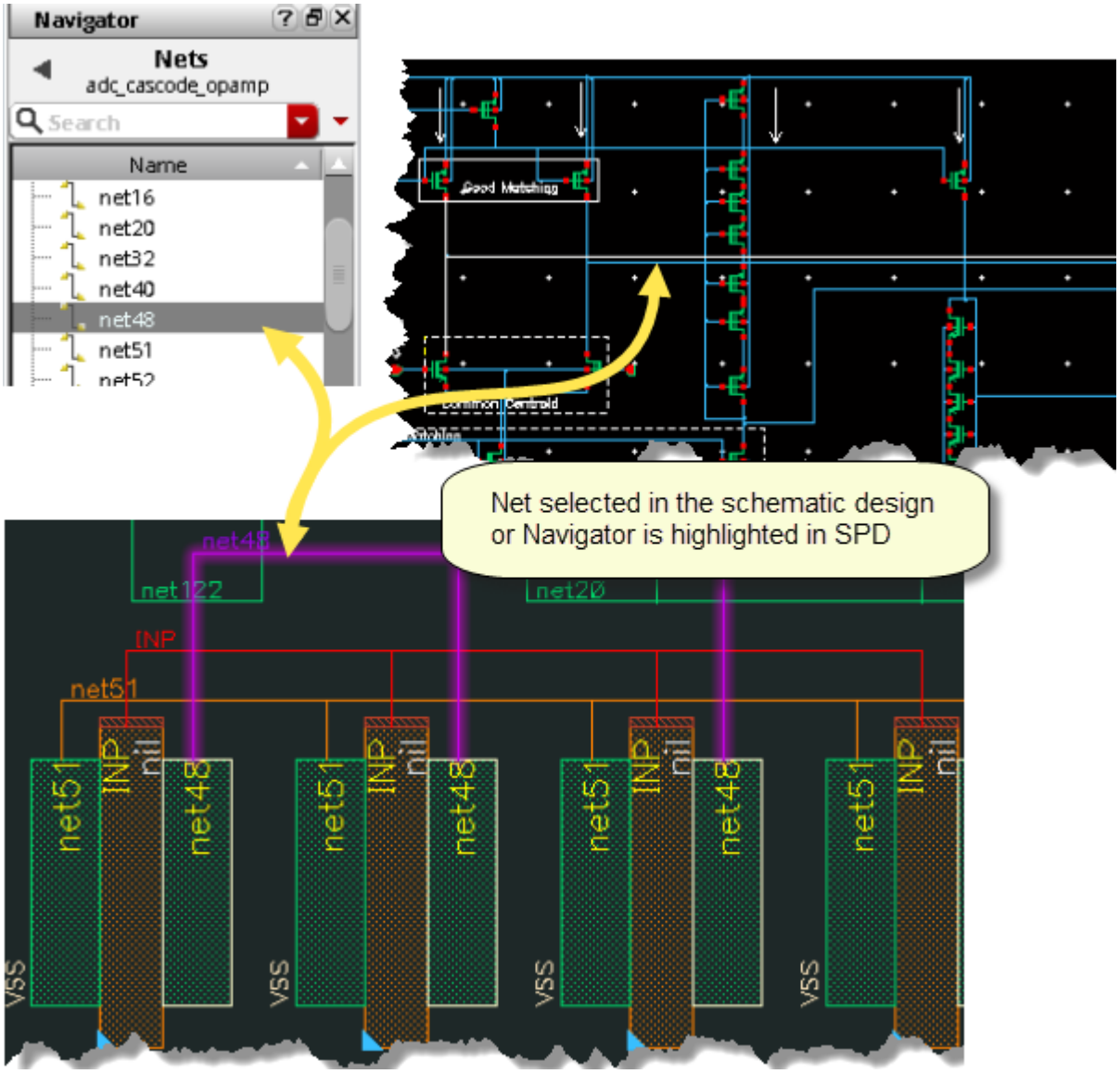
Selected Device Highlight

When you select a device in SPD, its corresponding device is automatically selected in the schematic:



© 2023 Cadence Design Systems, Inc.

When you select a net in the schematic design or Navigator, its corresponding net is automatically highlighted in the SPD Editor window:



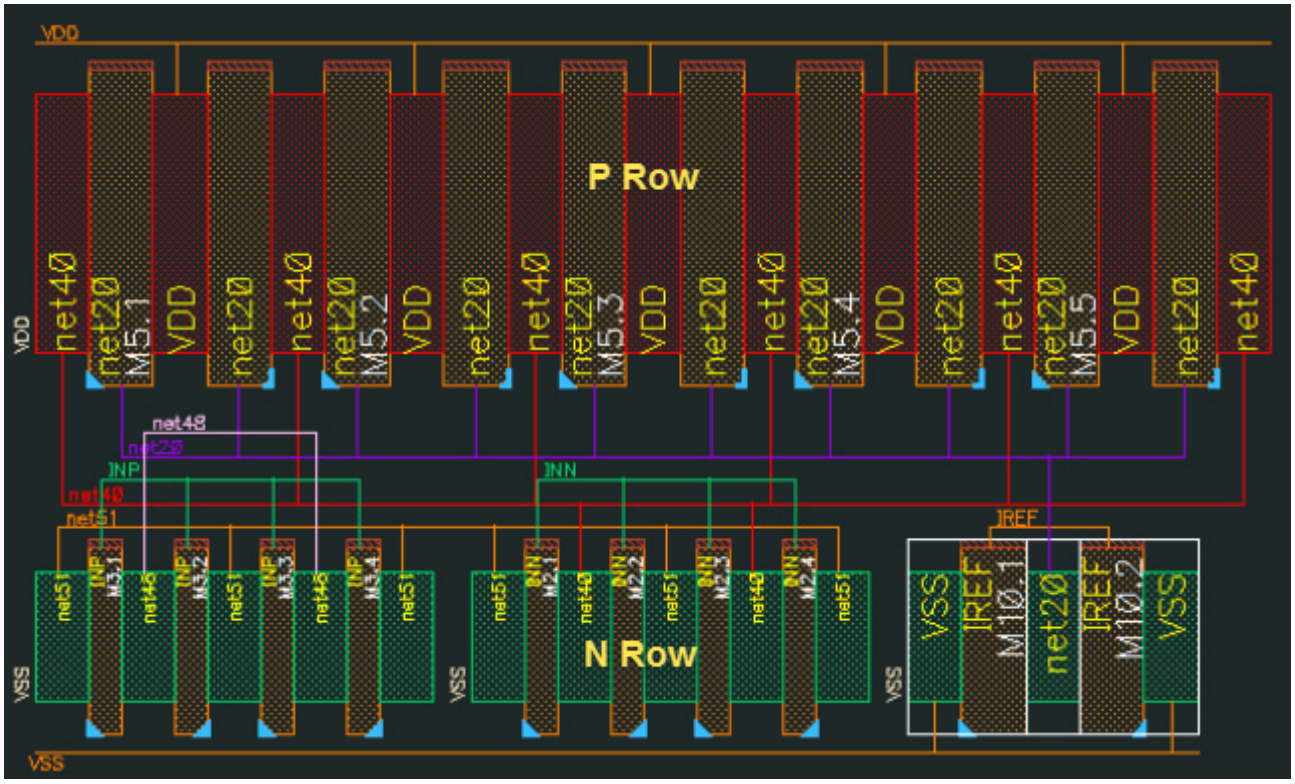
© 2023 Cadence Design Systems, Inc.

Understanding the Symbolic Design

The row-based, symbolic layout generated in SPD shows only the required information and does not show all the layers.

Row-based Placement

By default, when you generate the symbolic layout, devices are arranged in an NP row pattern. The top row (P row) of the design, represented in red, contains the PMOS devices and the bottom row (N row), represented in green, contains the NMOS devices.



Support for Multiple Rows

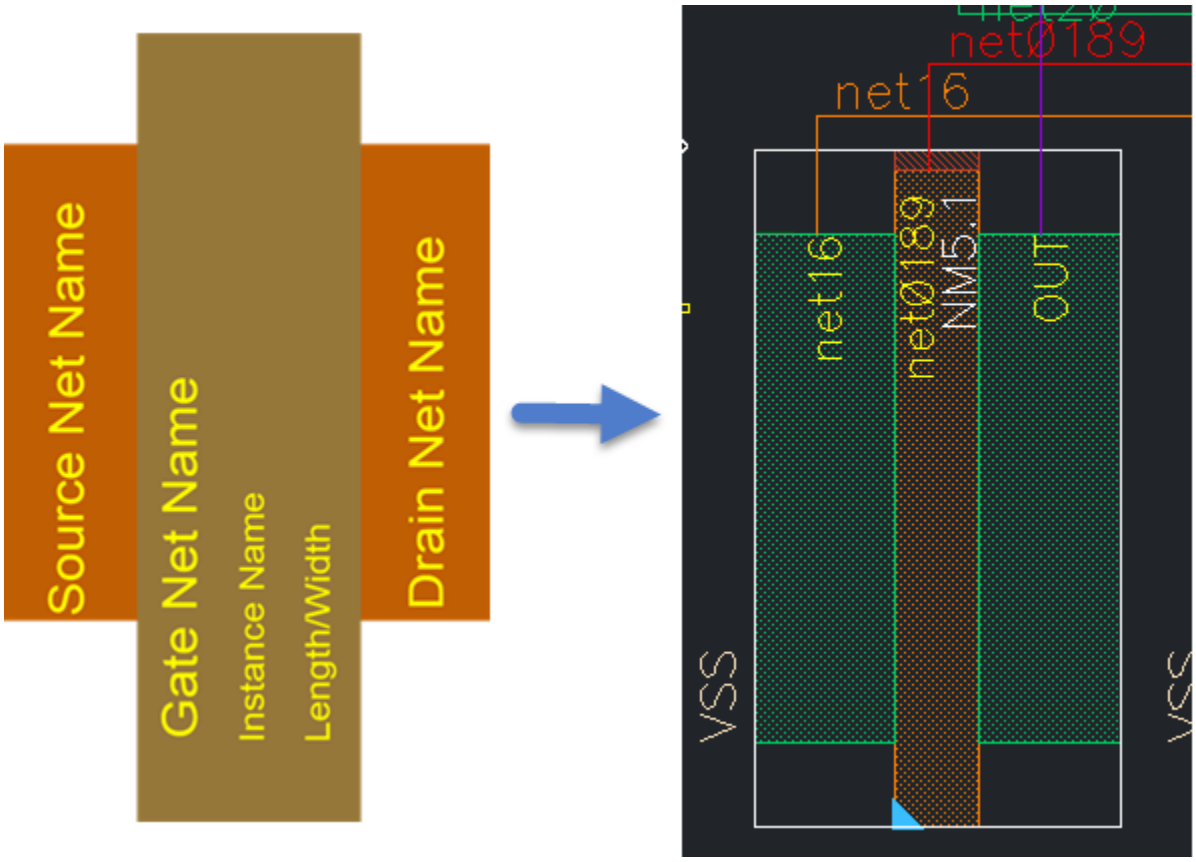
SPD enables you to add more number of rows and move devices to these rows based on your requirement. Some of the multi-row patterns supported by SPD include: NP, PN, NPPN, PNNP, NPNP, PNPN, NNPP, PPNN, NNNP, PNNN, NPPP, PPPN, NNNN, and PPPP. SPD can group instances with same bulk nets at the same row or at a near location during initializing SPD in the multiple row flow.

© 2023 Cadence Design Systems, Inc.

Device Representation

When a symbolic layout is generated from a schematic, all information that is not needed is hidden. What you see in the design are symbols that comprise the drain, source, and gate of each P-type or N-type transistor.

In the figure below, the drain of the symbolic device is displayed on the right side, the source on the left and the gate is located in between. Each component is labeled with the name of the net to which it is connected. The gate also shows name of the instance and you can choose to display the size of the device, which is not displayed by default.

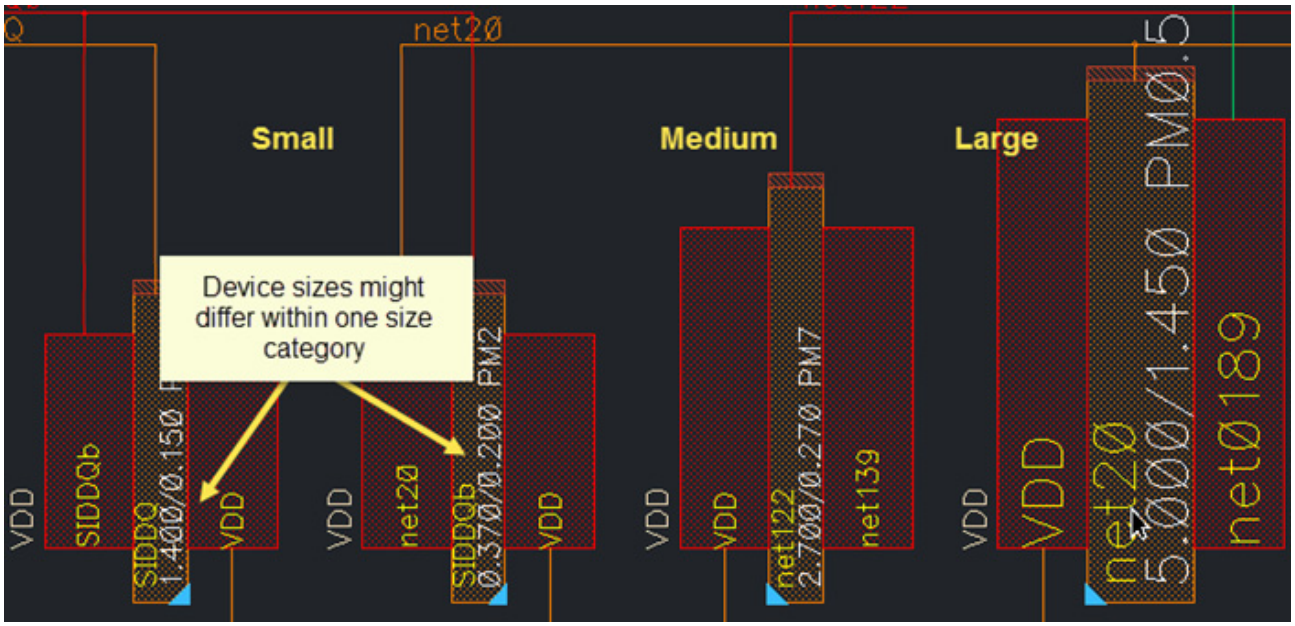


In the above image, you can see that the name of the instance is `NM5.1`, which is connected between `net16` and `OUT`.

© 2023 Cadence Design Systems, Inc.

Devices of Different Sizes

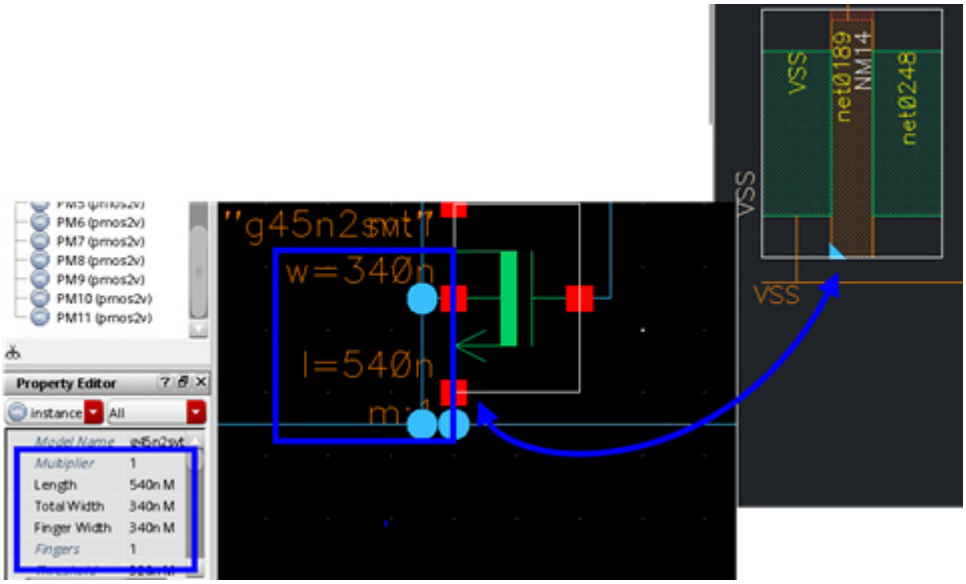
Instead of displaying the exact sizes of each abstract device, SPD shows their relative sizes by automatically categorizing the device sizes into small, medium, and large sizes based on their widths. Each category can include devices of different sizes as you can see in the screenshot below.



There are three ways in which you can find out the real size of the devices:

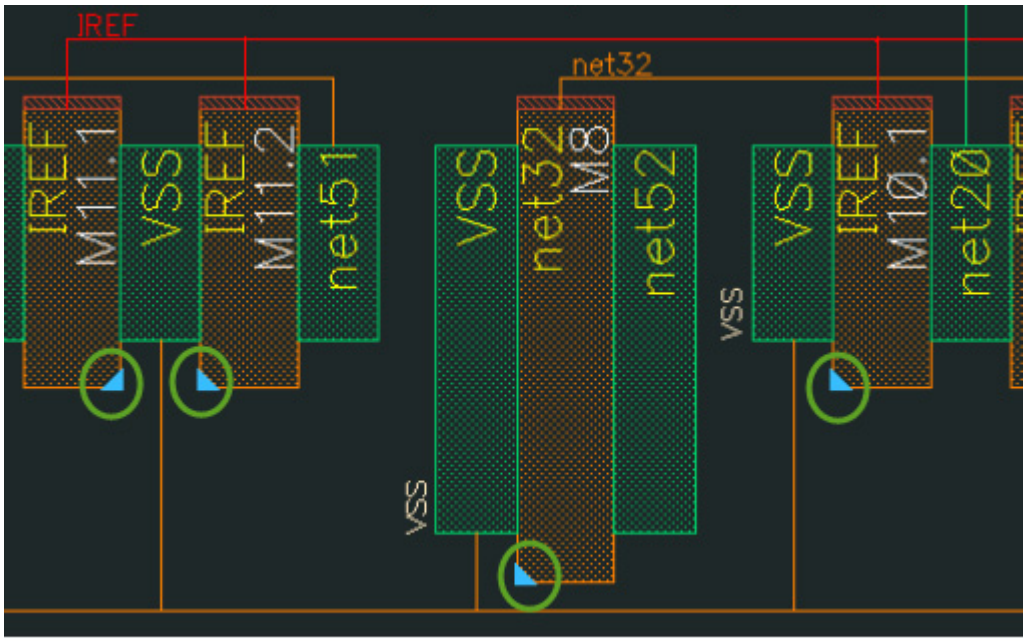
- View the size label for each device. These labels are not displayed by default. You can enable them from the Display Options pane.
- View the actual size in the schematic. When you select a device in the SPD window, the device is cross-selected in the schematic window is automatically selected, making it simpler to view its actual size.
- Enable the *Free Bucket Size* option to display the symbolic device in proportion to its real size. When you use this option, the design can approximately look like the actual layout. See Displaying Free Bucket Size of Devices for more information.

© 2023 Cadence Design Systems, Inc.



Device Orientation

Orientation of a symbolic device is indicated using a blue triangle. The size of these triangles is the same regardless of device sizes. The orientation of the symbolic device in the figure below is `R0`.

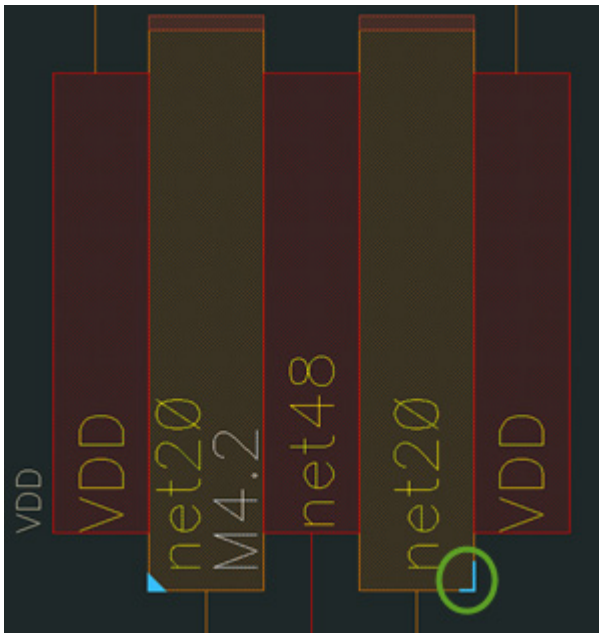


Orientation of symbolic devices represented by blue triangles

© 2023 Cadence Design Systems, Inc.

Device Fingers

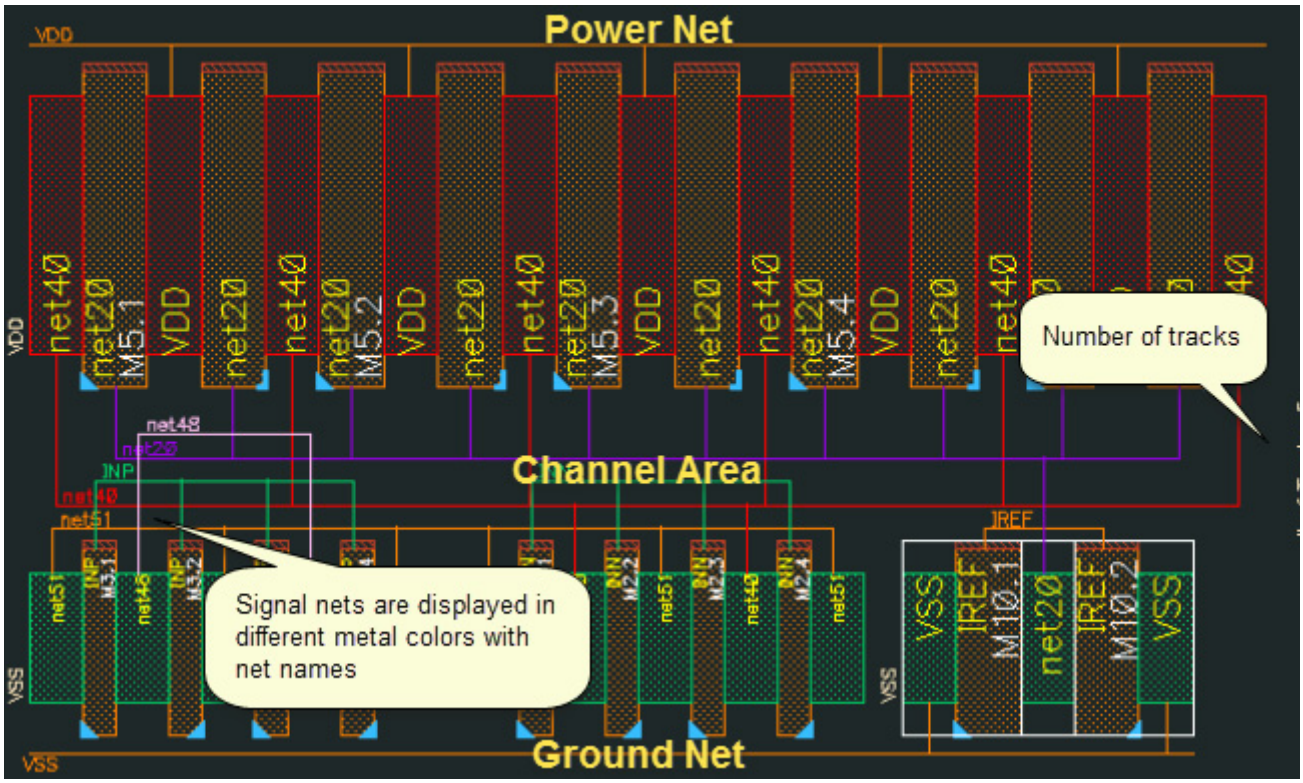
An instance that contains several fingers is identified by a blue L-shape. The first finger of the device is marked with a blue triangle and the last finger is identified by a blue L-shape. In the image below, you can see an instance with two fingers.



© 2023 Cadence Design Systems, Inc.

Displaying Flight Lines

The symbolic design displays flight lines in the channel area between the P and N rows. These flight lines are not real physical design objects and therefore they cannot be selected or modified. They depict logical connections between the drain, gate, and source pins of the devices.



Note the following:

- Power and ground nets are displayed by default in the design, in a single color.
- The number of tracks appears at the right of the channel area. The height of the channel area is also symbolic and remains fixed even if the number of tracks changes when the design is edited.
- Signal nets are displayed in the appropriate metal layer color in the channel area along with their net names.

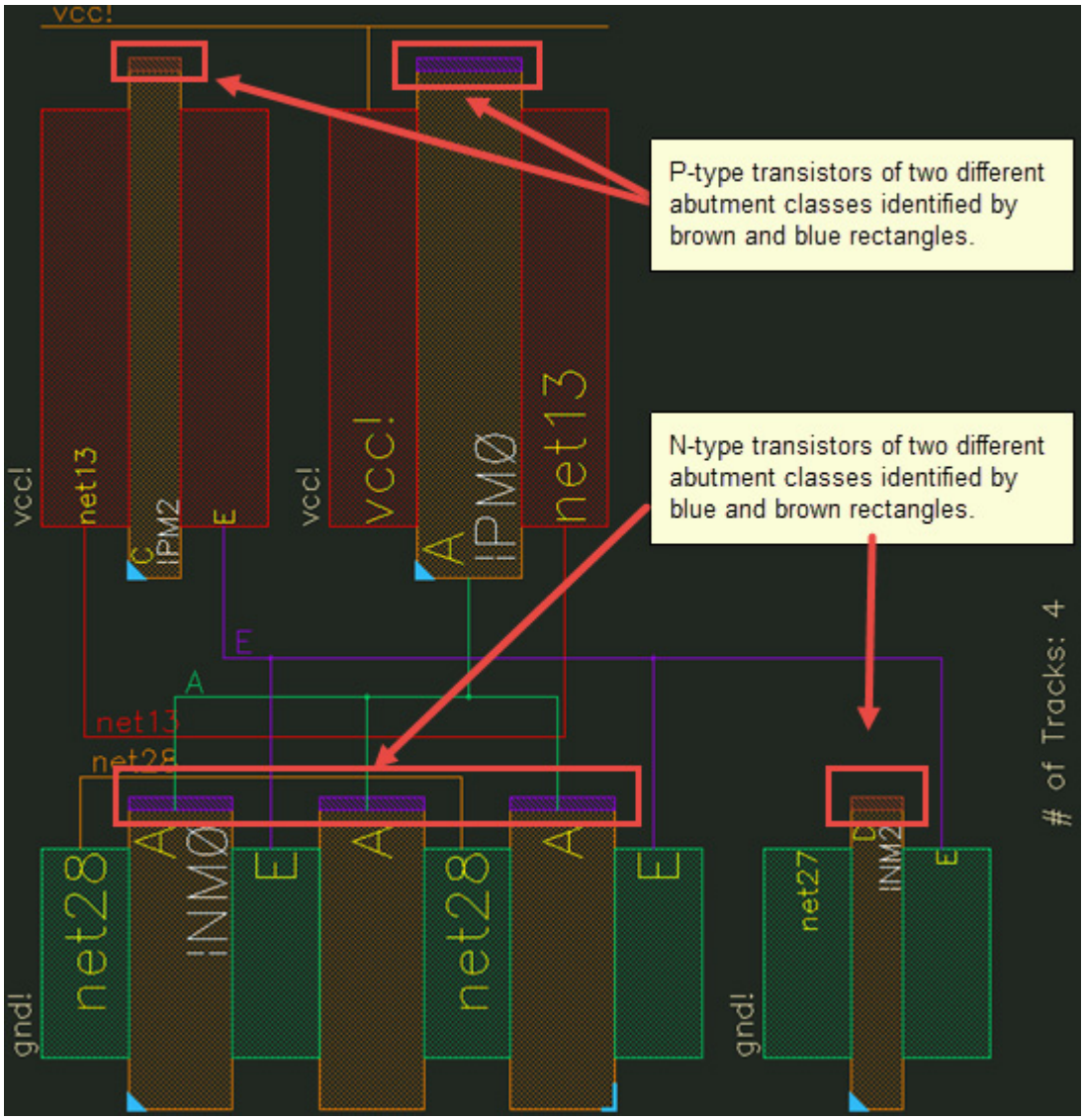
© 2023 Cadence Design Systems, Inc.

Abutment

Abutment Class of Devices

Symbolic devices of different masters or different abutment classes are identified by small rectangles drawn in different colors at the top of the gate. Instances of different abutment classes cannot be abutted.

In the image below, P-type transistors and N-type transistors of two different abutment classes, are identified by brown and blue rectangles.



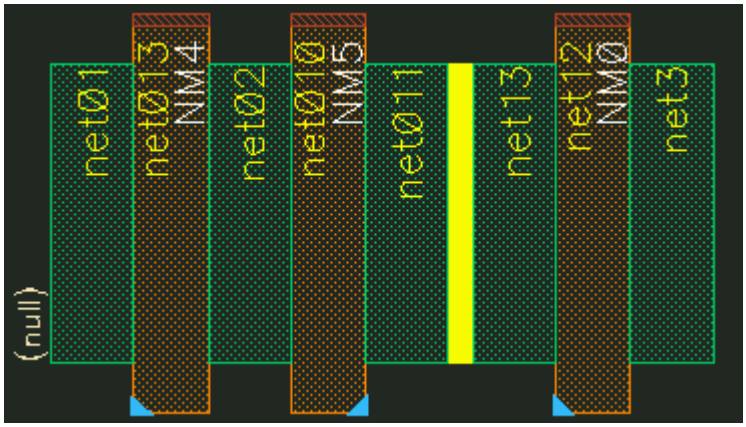
© 2023 Cadence Design Systems, Inc.

For more information on abutment class, refer abutClass in *Virtuoso Layout Suite XL User Guide*.

User Defined Abutments

You can define PDK-specific abutments that is different from the regular oxide diffusion (OD) abutment, for example the dummy poly abutment. A user-defined abutment is identified by a rectangle between the source and drain of two abutted instances. The color of the rectangle indicates the type of user defined abutments. Different colors denote different types of user-defined abutments.

In the image below, you can see a yellow rectangle between instance NM5 and NM0, which denotes that NM5 abuts NM0 by a user-defined abutment. While NM4 and NM5 is the regular oxide diffusion abutment.



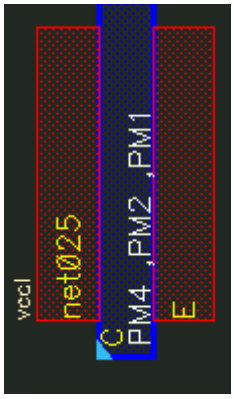
For more information, see Abutting and Unabutting Symbolic Devices.

© 2023 Cadence Design Systems, Inc.

Stacked Devices

SPD can stack multiple instances together. To indicate the bottom instance of a set of stacked devices, the gate of the bottom instance is drawn in dark blue and the names of stacked instances are shown on the gate.

In the image below, you can see instance PM4 is stacked by PM2 and PM1.



For more information, see Stacking Symbolic Devices.

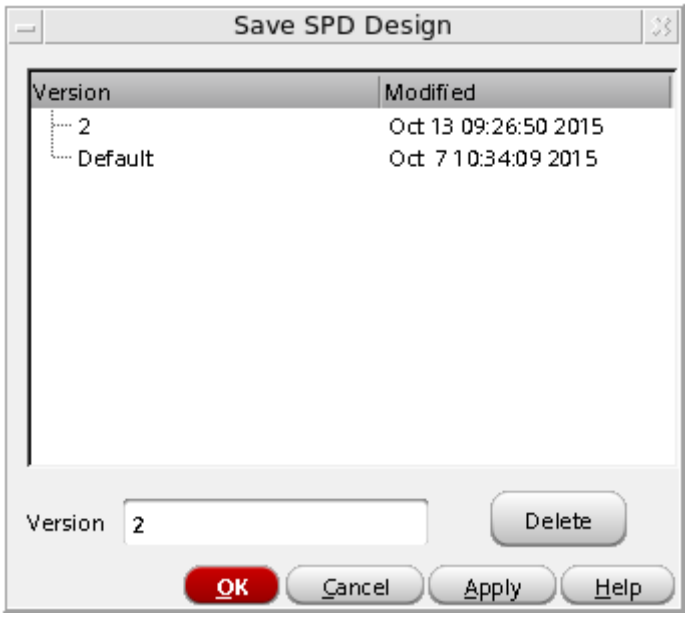
© 2023 Cadence Design Systems, Inc.

Saving an SPD Design

To save an SPD design:

1. Choose *Place – Save* [F2].

   The Save SPD Design form is displayed.



2. Specify the version number for the design. You can also select an existing version number from the *Version* list in the form.
3. Click *OK*.

Design with the specified version number is saved.

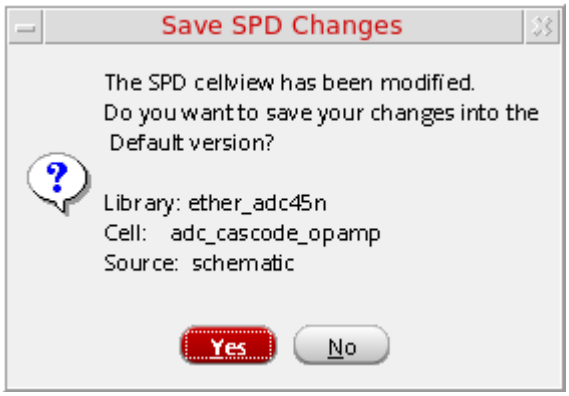
If there are several designs listed in the Save SPD Design form, you can perform the following tasks to sort the designs:

- Click the header of the Version column to sort designs by the version number.
- Click the header of the Modified column to sort the designs by the modification time.

Click any of the header columns two times to change the sorting direction.

© 2023 Cadence Design Systems, Inc.

The following dialog box will be displayed if you try to exit SPD without saving the design.



If you save a design when the *Options – SPD – Placement – Auto pack in X direction* check box is not selected, then any extra spacing in the design is also saved. This spacing is restored when you load the design for further editing.

Exiting the SPD Editing Window

Choose *Place – Exit* [`Ctrl+w`] or click the *Exit* button on the SPD toolbar to close the SPD Editing window.

© 2023 Cadence Design Systems, Inc.

© 2023 Cadence Design Systems, Inc.

3

Editing Symbolic Devices

After Generating a Symbolic Placement, you might want to edit the devices to optimize the design. SPD enables you to perform various device editing tasks such as moving, abutting, swapping, flipping, and stacking the devices. You can also split fingered devices, generate chained devices and add dummy devices to the design.

This chapter shows you how to edit symbolic devices in SPD. The chapter discusses the following topics:

- Loading a Design
- Undoing and Redoing Commands
- Moving Symbolic Devices
- Abutting and Unabutting Symbolic Devices
- Swapping Symbolic Devices
- Flipping Symbolic Devices
- Permuting Symbolic Devices
- Stacking Symbolic Devices
- Splitting Fingered Devices
- Folding Symbolic Devices
- Aligning Symbolic Devices
- Generating Chained Devices
- Adding Dummy Devices
- Backannotating Symbolic Devices

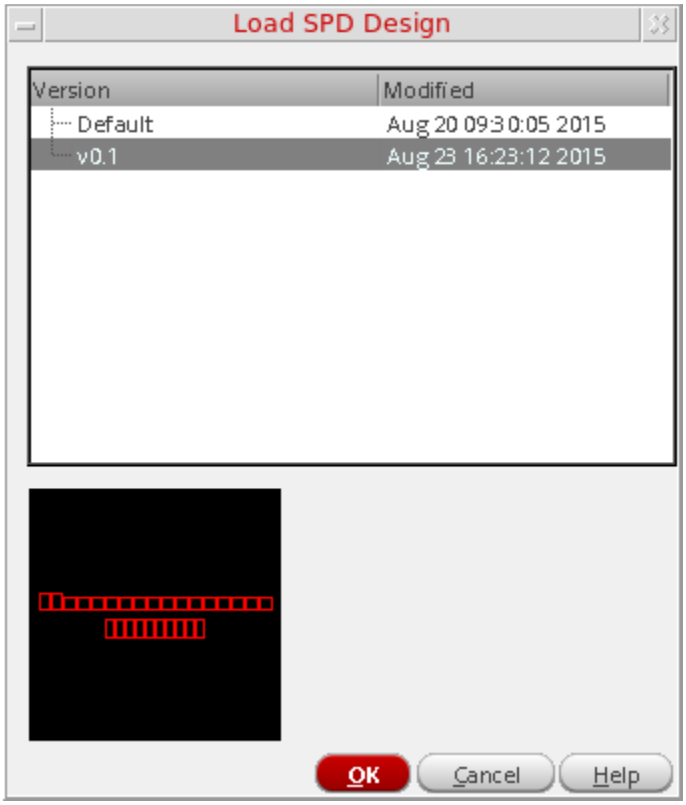
© 2023 Cadence Design Systems, Inc.

Loading a Design

SPD lets you load a previously saved SPD design and continue working in it. To load a design:

1. Choose *Place – Load* [F5].

   The Load SPD Design form is displayed.



2. Select the version of the saved design you want to open.

   A preview of the selected design is shown in the bottom section.

3. Click *OK*.

The selected design opens in the canvas.

If there are several designs listed in the Load SPD Design form, you can perform the following tasks to sort the designs:

- Click the header of the Version column to sort designs by the version number.
- Click the header of the Modified column to sort the designs by the modification time.
- Click any of the header columns two times to change the sorting direction.

© 2023 Cadence Design Systems, Inc.

Undoing and Redoing Commands

To undo a command, choose *Edit – Undo* or press `u`. Alternatively, choose the *Undo* button.

To reinstate a change that you canceled with *Undo*, choose *Edit – Redo* or press `Shift+u`. Alternatively, choose the *Redo* button.

For more information, see Undoing Commands and Redoing Commands.

Moving Symbolic Devices

The *Move* command lets you move symbolic devices from one location to another within the SPD canvas. You can move objects either to an empty space or insert them directly into an existing chain. When moved to another chain, the object is directly inserted into the target chain.

SPD provides the following moving options:

- Moving Devices
- Moving Multiple Instances
- Moving Devices Based on Whether Horizontal Packing is Enabled

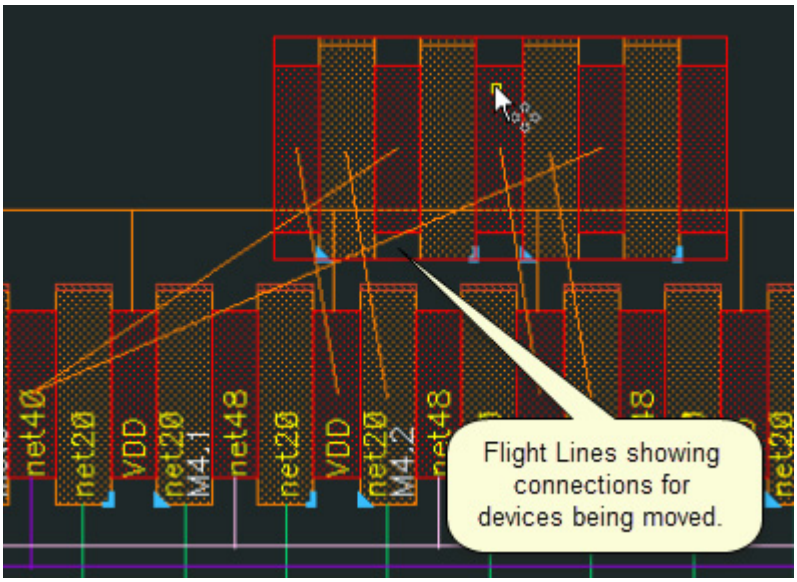
Moving Devices

To move devices using the *Move* Command:

1. Choose *Edit – Move* [`m`] or click the *Move* button on the SPD toolbar.
2. Select one or more symbolic objects in the SPD canvas.

© 2023 Cadence Design Systems, Inc.

3. Click the reference point for the move (the point from which the move starts).

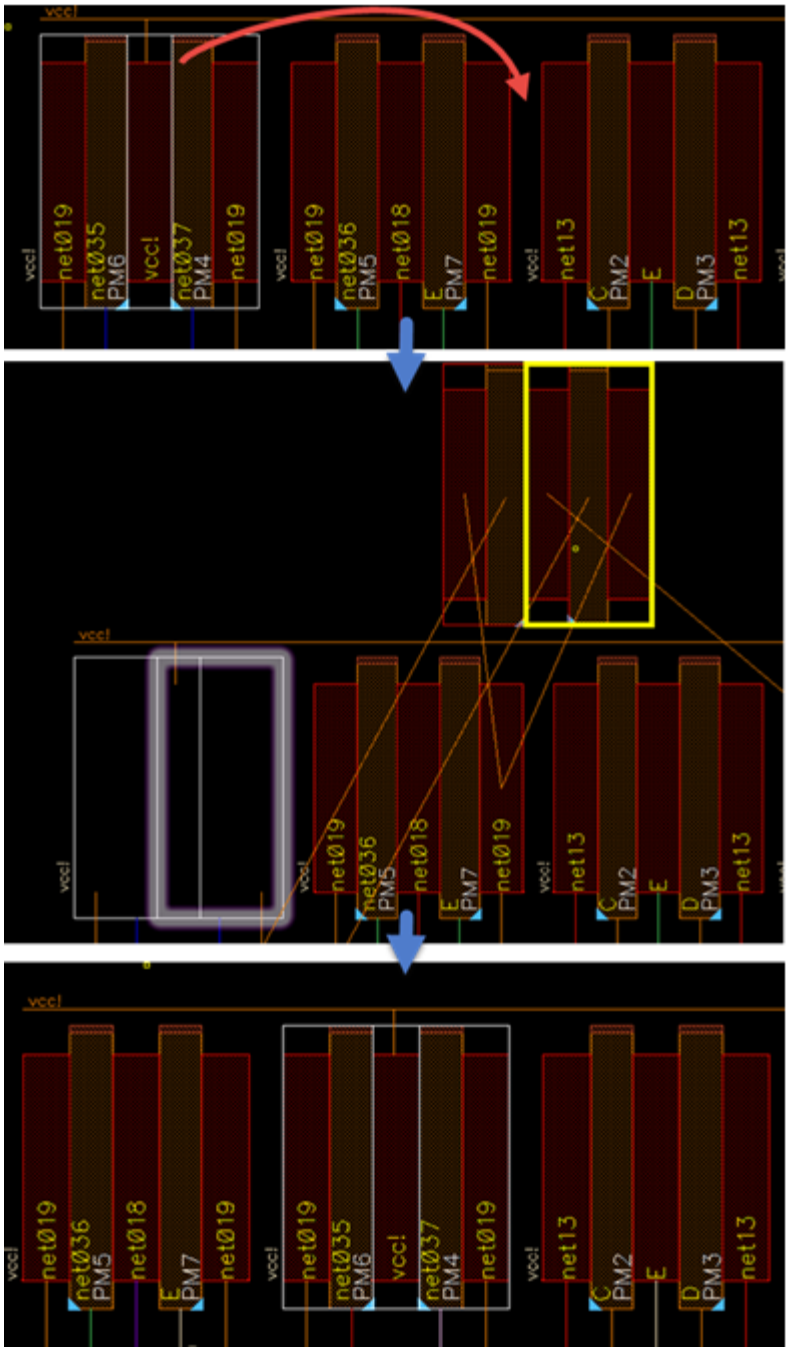


Flight lines show the connections for the devices being moved. When the objects are moved, the flight lines are dynamically updated.

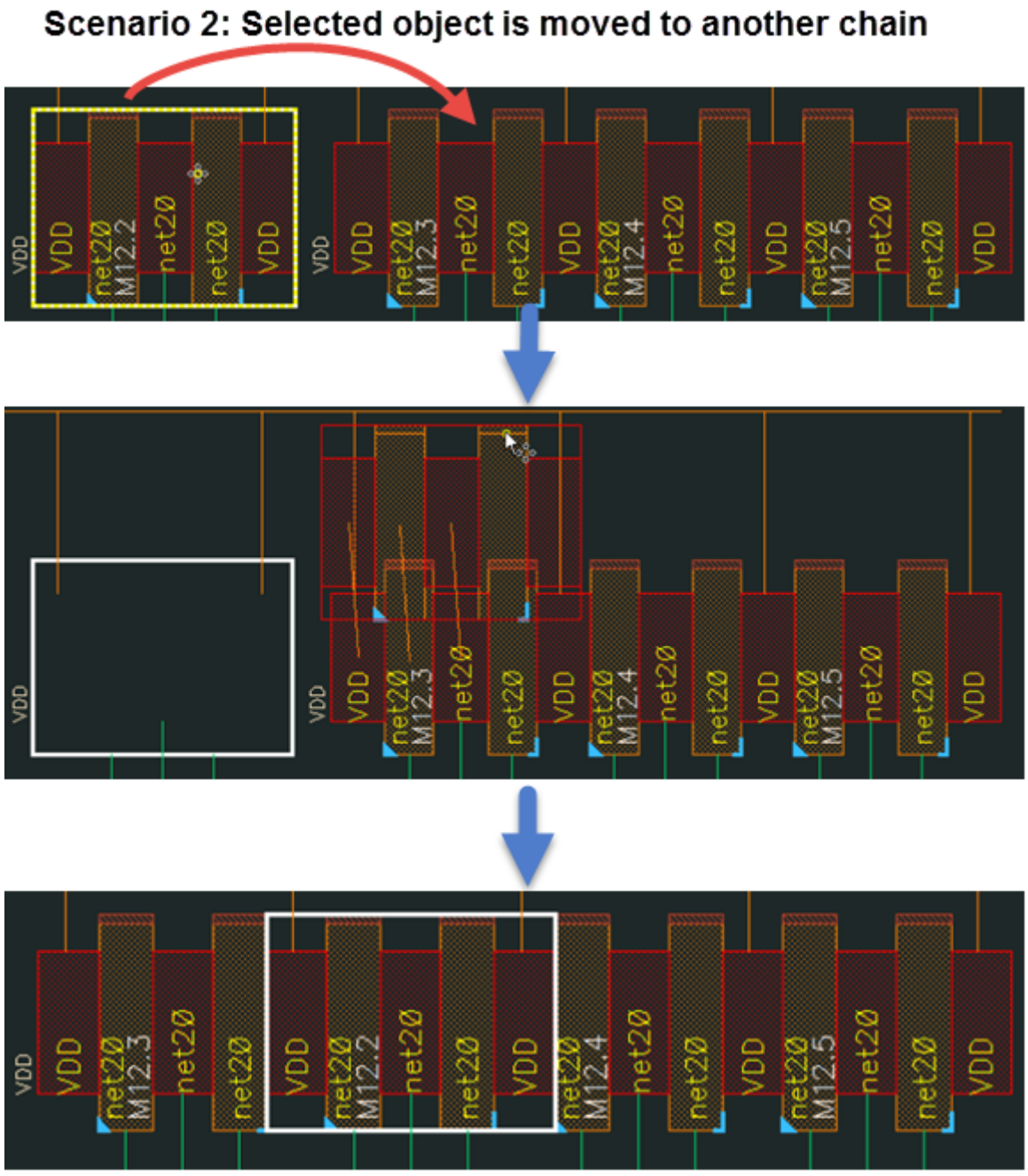
© 2023 Cadence Design Systems, Inc.

4. Click a target instance or the space into which you want to move the selected objects.

Scenario 1: Selected object is moved to empty space



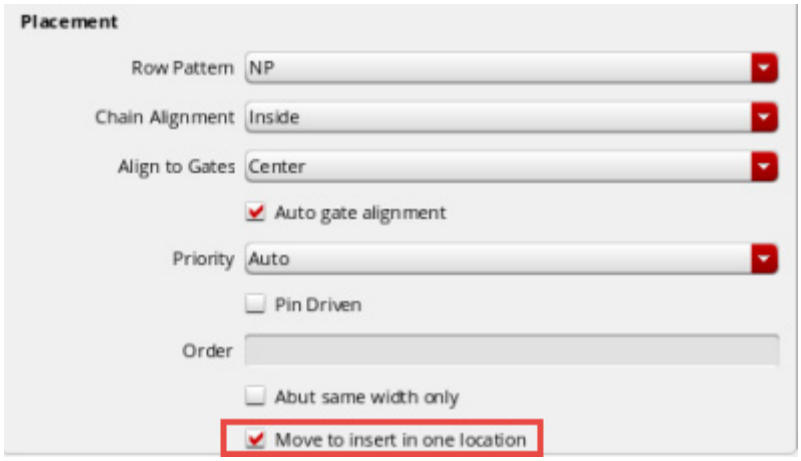
© 2023 Cadence Design Systems, Inc.



© 2023 Cadence Design Systems, Inc.

Moving Multiple Instances

Moving multiple instances depends on whether the *Move to insert in one location* check box on the Placement options form is selected.



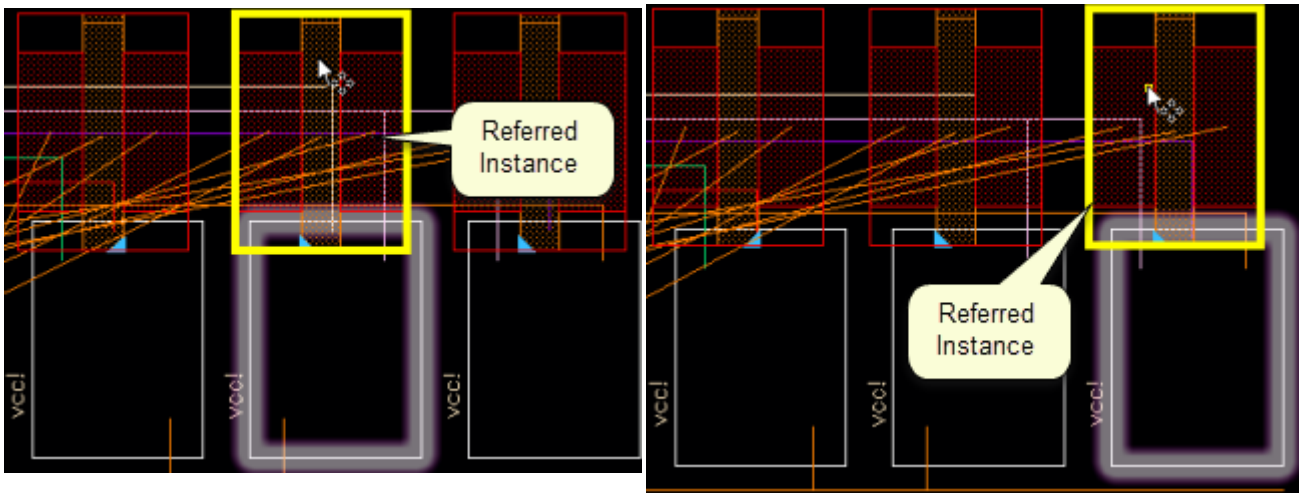
When this option is selected, SPD sets a referred instance, which is then used to move all selected instances to the specified location.

Referred Instances

Referred Instances are selected differently in the IC6.1.x and the advanced nodes version of the software.

- In IC6.1.x, when you select multiple instances to move, the left-most instance is considered as the referred instance.
- In advanced nodes, you can right-click to select the referred instance. The referred instance is highlighted in advanced nodes.

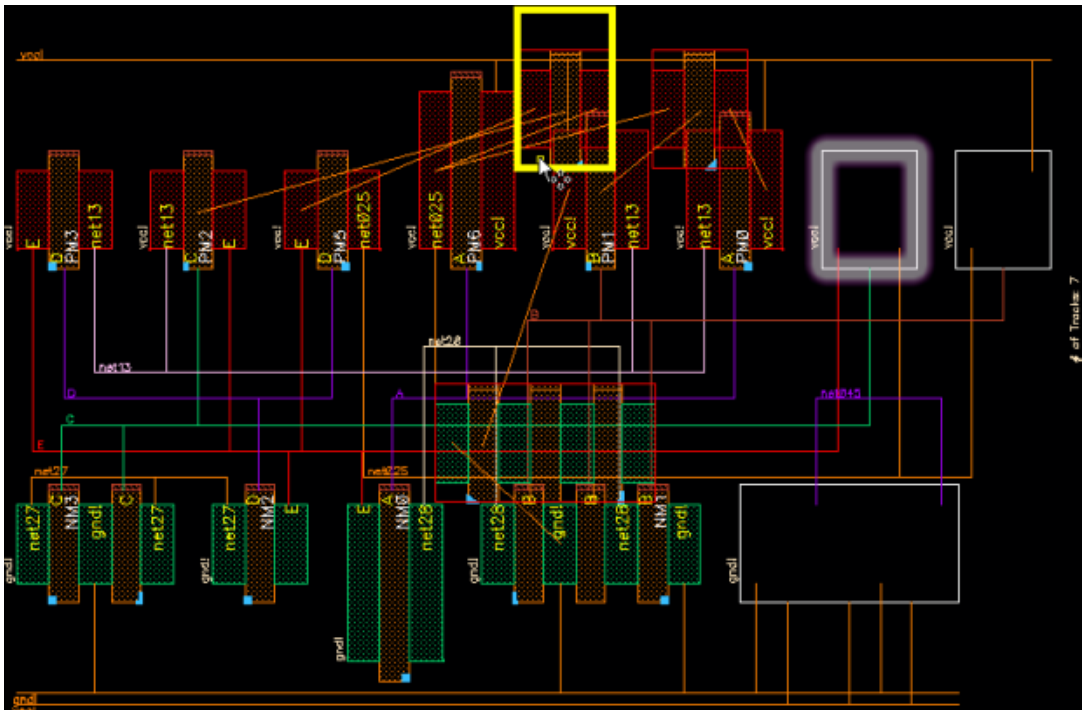
© 2023 Cadence Design Systems, Inc.



Right-click to Set the Referred Instance

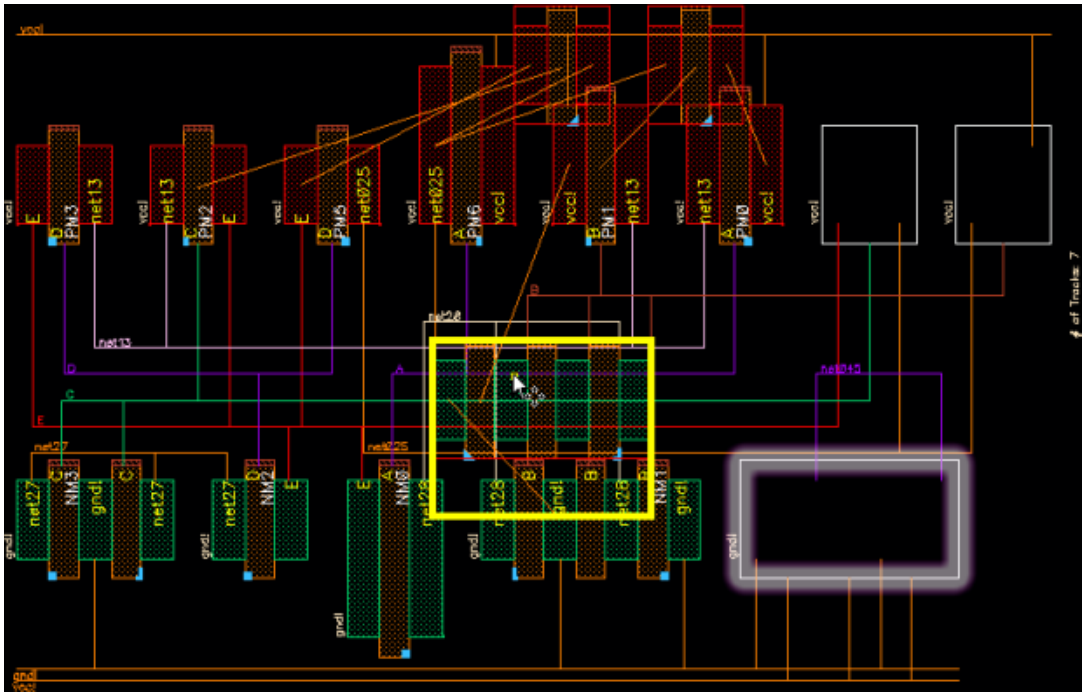
Move In Multiple Rows

When you select multiple chains to move, the chains are dropped to the X-coordinate of the referred instance.



PMOS as the Reference Object

© 2023 Cadence Design Systems, Inc.

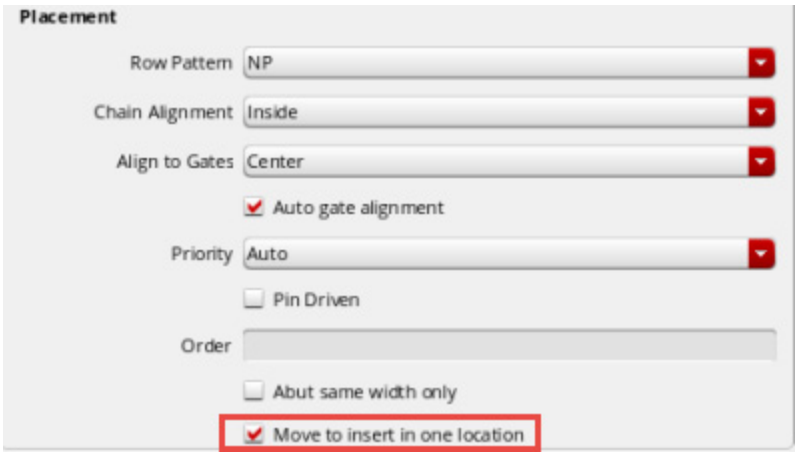


NMOS as the Reference Object

By default, when you move more than one instances in SPD, all moved instances are dropped at the specified location. The instances retain their original device order after move.

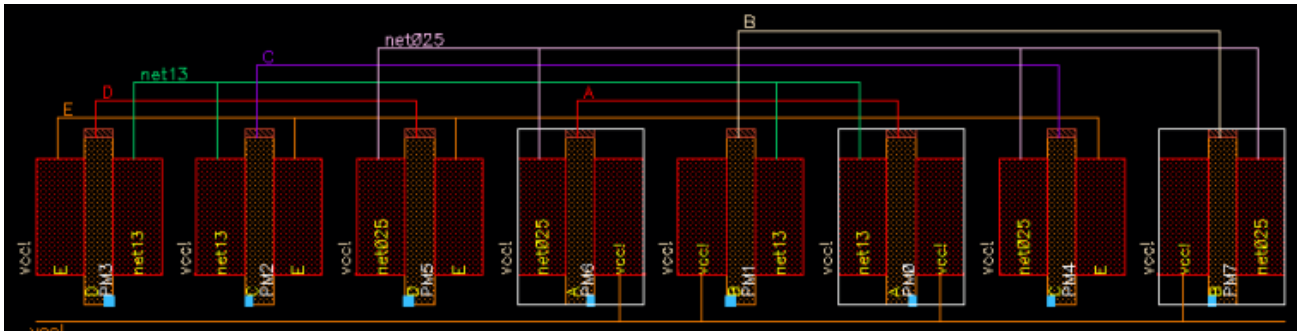
To move multiple instances to the same location:

1. Choose *Options – SPD – Placement.*
2. Ensure that the *Move to insert in one location* check box is selected.

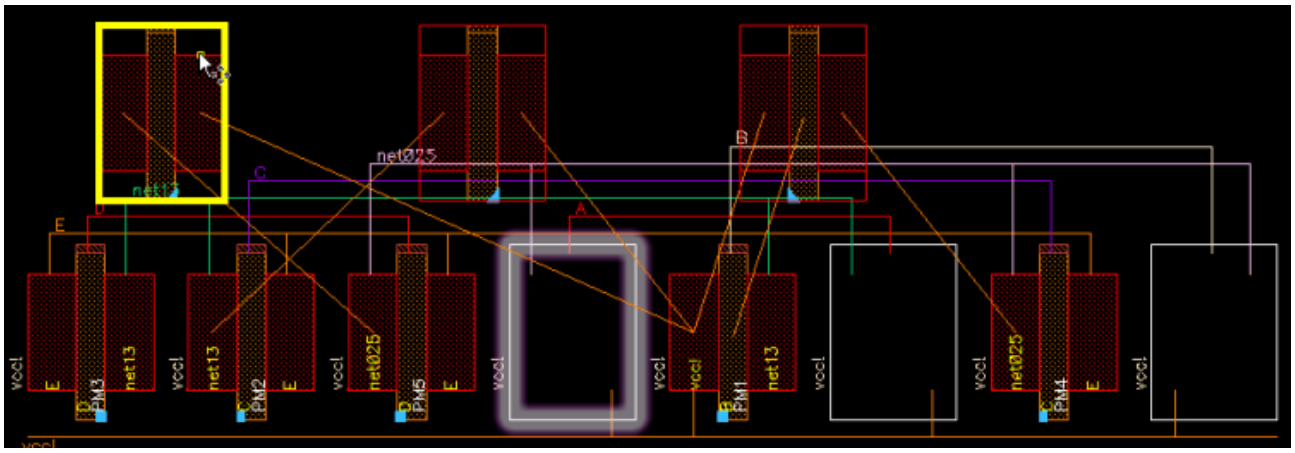


© 2023 Cadence Design Systems, Inc.

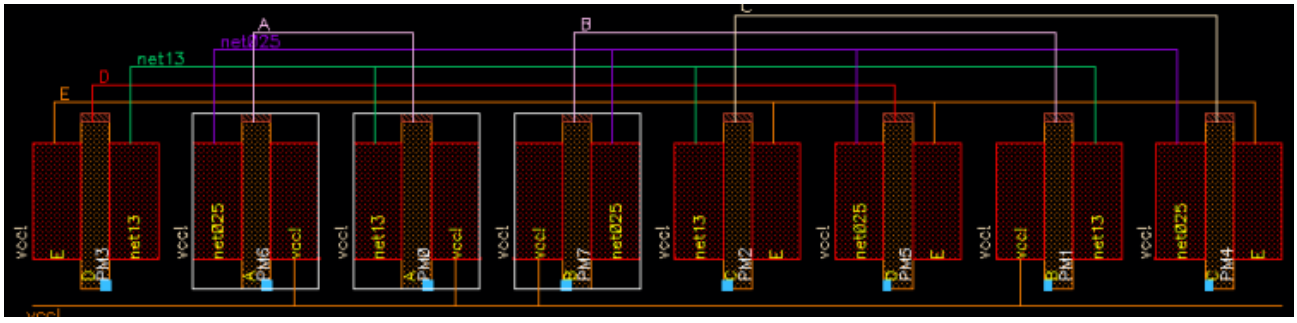
3. Select multiple instances to move.



4. Point to the location where you want to move the instances.



5. Selected instances are dropped at the specified location.

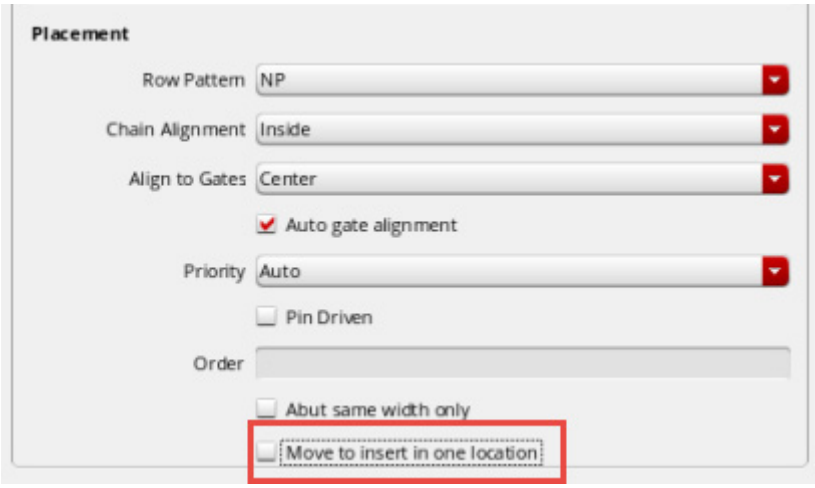


To move multiple instances to relative locations based on their x-coordinate:

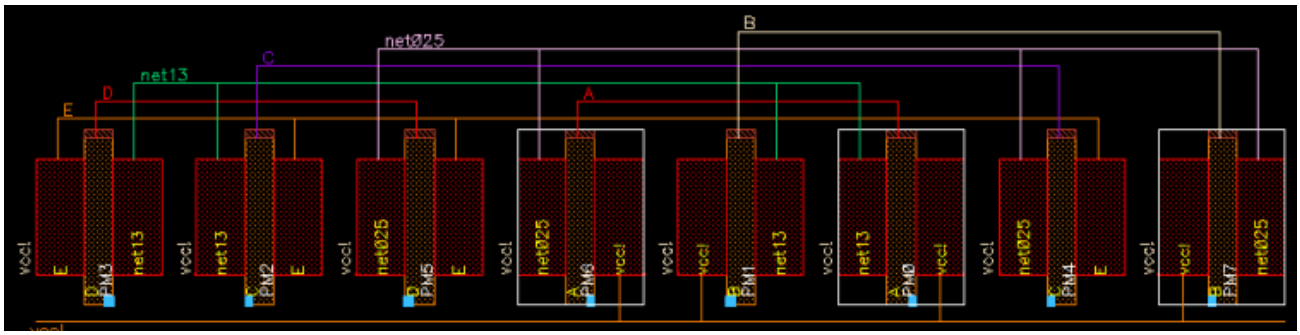
1. Choose *Options – SPD – Placement.*

© 2023 Cadence Design Systems, Inc.

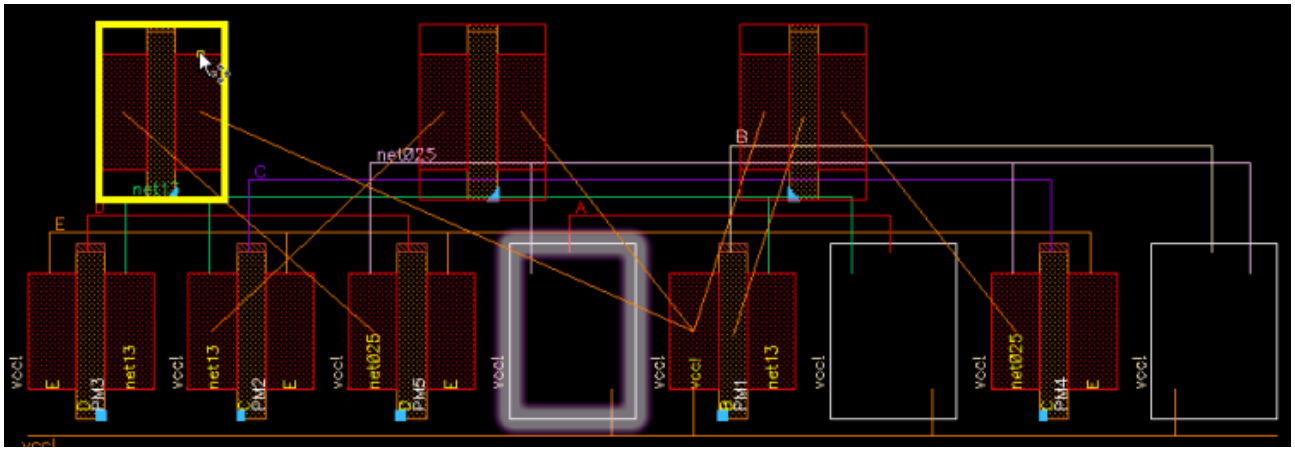
2. Deselect the *Move to insert in one location* check box.



3. Select instances to move.

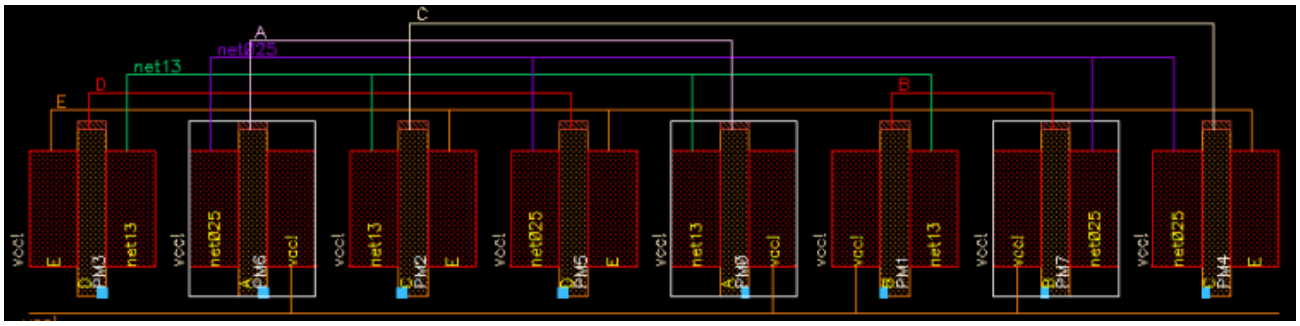


4. Point reference object to the location where you want to move them.

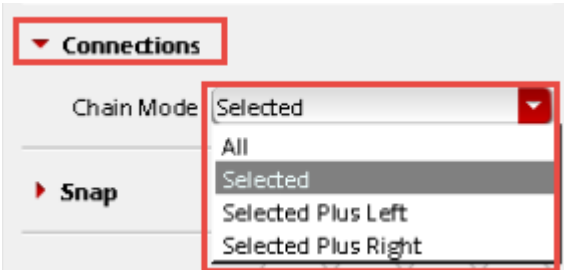


© 2023 Cadence Design Systems, Inc.

5. Instances are moved based on their relative positions.



With the *Move* command active, press `F3` to display the Move form. Use this form to customize the scope of move using the *Connections* option.



For more information on moving objects, see Moving Objects in the *Virtuoso Layout Suite XL User Guide*.

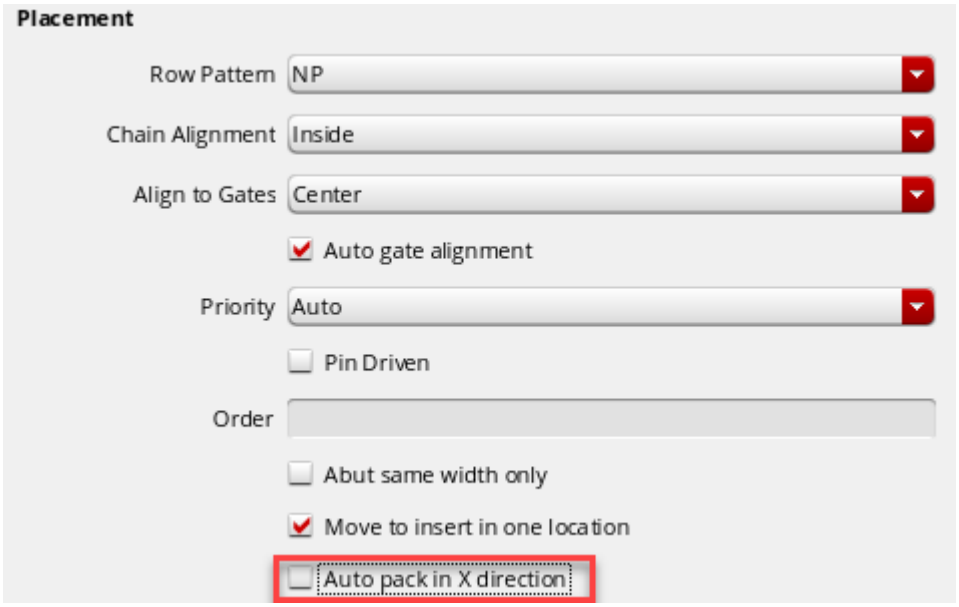
© 2023 Cadence Design Systems, Inc.

Moving Devices Based on Whether Horizontal Packing is Enabled

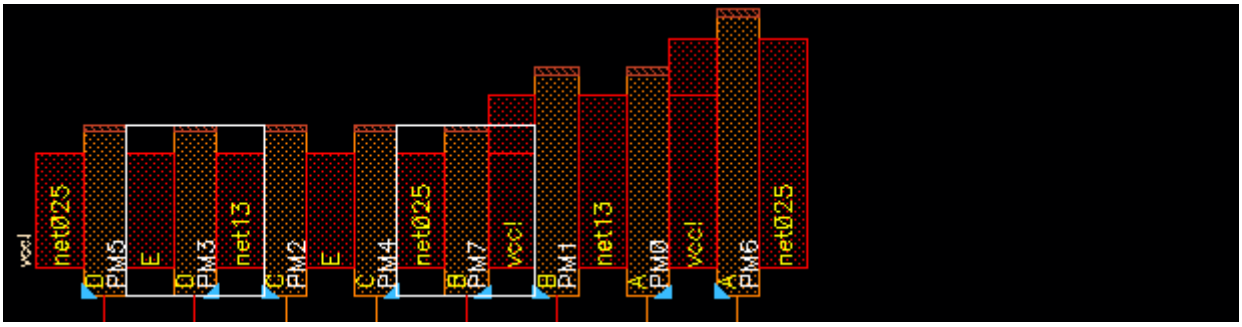
By default devices in an SPD design are packed when you move some of the devices. If needed, you can disable horizontal packing of devices after move. To do this, perform the following steps:

1. Choose *Options – SPD – Placement.*
2. Deselect the *Auto pack in X direction* check box.

.

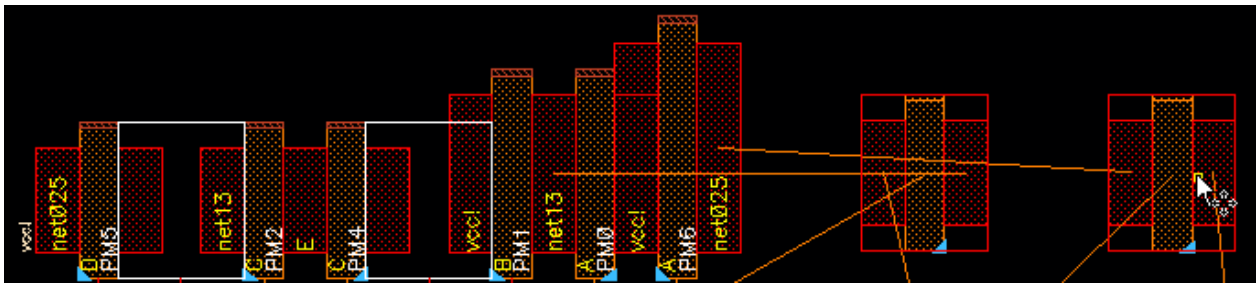


3. Select the instances to move.

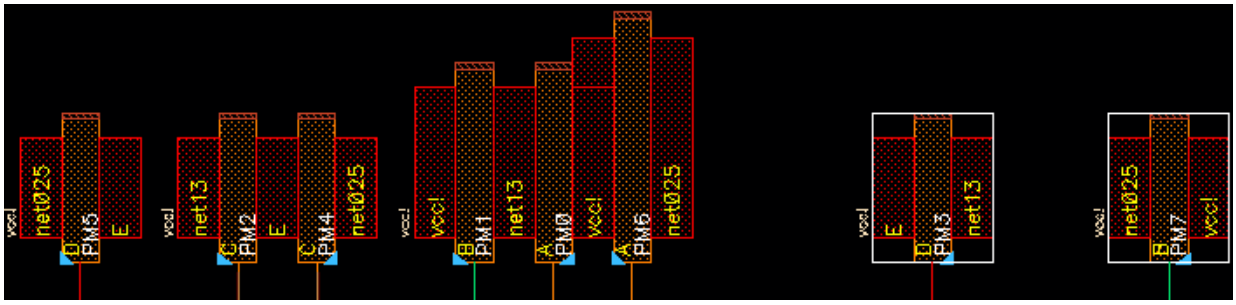


© 2023 Cadence Design Systems, Inc.

4. Point to the location where you want to move the instances.

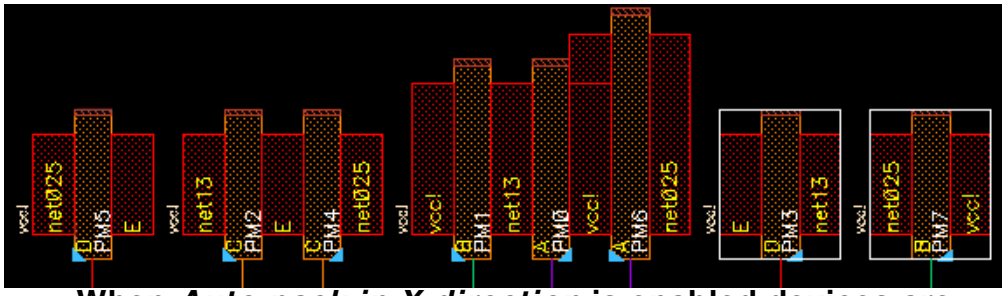


5. Instances are moved and placed at the exact location where you moved them.



When *Auto pack in X direction* is disabled devices are placed at the pointed location

When *Auto pack in X direction* option is selected, the instances are automatically packed after move, which is the default setting.



When *Auto pack in X direction* is enabled devices are packed after move

© 2023 Cadence Design Systems, Inc.

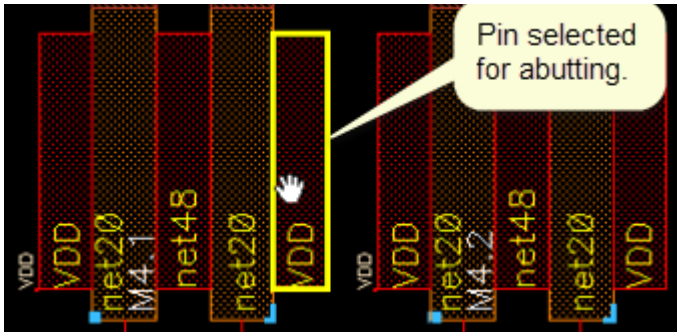
Abutting and Unabutting Symbolic Devices

Abutment allows symbolic devices to be automatically overlapped, aligned, and electrically connected without introducing a design rule violation or connectivity error. Abutment reduces the area occupied by a circuit and the length of the interconnect wiring. When two instances are abutted, the body contacts between the instances are deleted.

Abutment is a row-based operation. When you select multiple instances for abutment, the instances in each row are abutted separately.

Highlighting and Selection when the Abut Command is Active

When the *Abut* command is active, you can point at a source or drain of an instance, and select the respective pin for abutment. In this case, the selected instance is not highlighted in the source schematic.



Note: For information on how highlighting works when *Abut* command is not active, see Highlighting and Selecting Data.

The SPD Editor provides the following abutment and unabutment options:

- Abutting Devices
- Unabutting Devices
- Abutting Dummy Poly Devices

Abutting Devices

You can perform the following device abutment tasks from the SPD Editor:

- Abutting in Pre-select Mode

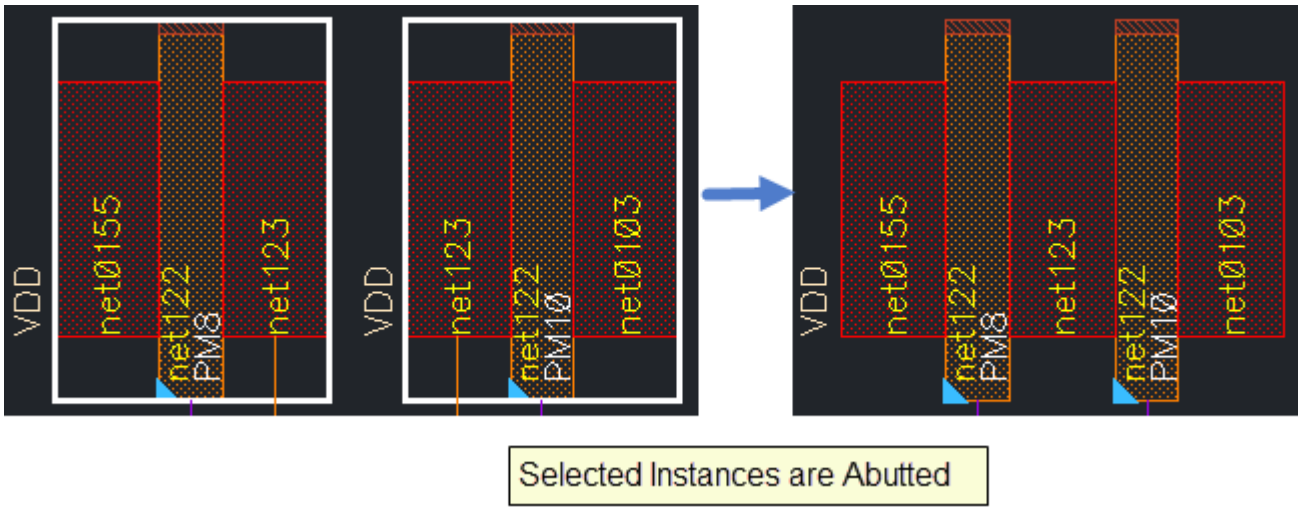
© 2023 Cadence Design Systems, Inc.

Abutting in Pre-select Mode

To abut symbolic devices in pre-select mode:

1. Select two or more devices to abut. Each selected instance includes a source, a gate, and a drain.
2. Choose *Edit – Abut* [b] or click the *Abut* button on the SPD toolbar.

   The selected instances are abutted.



Selected Instances are Abutted

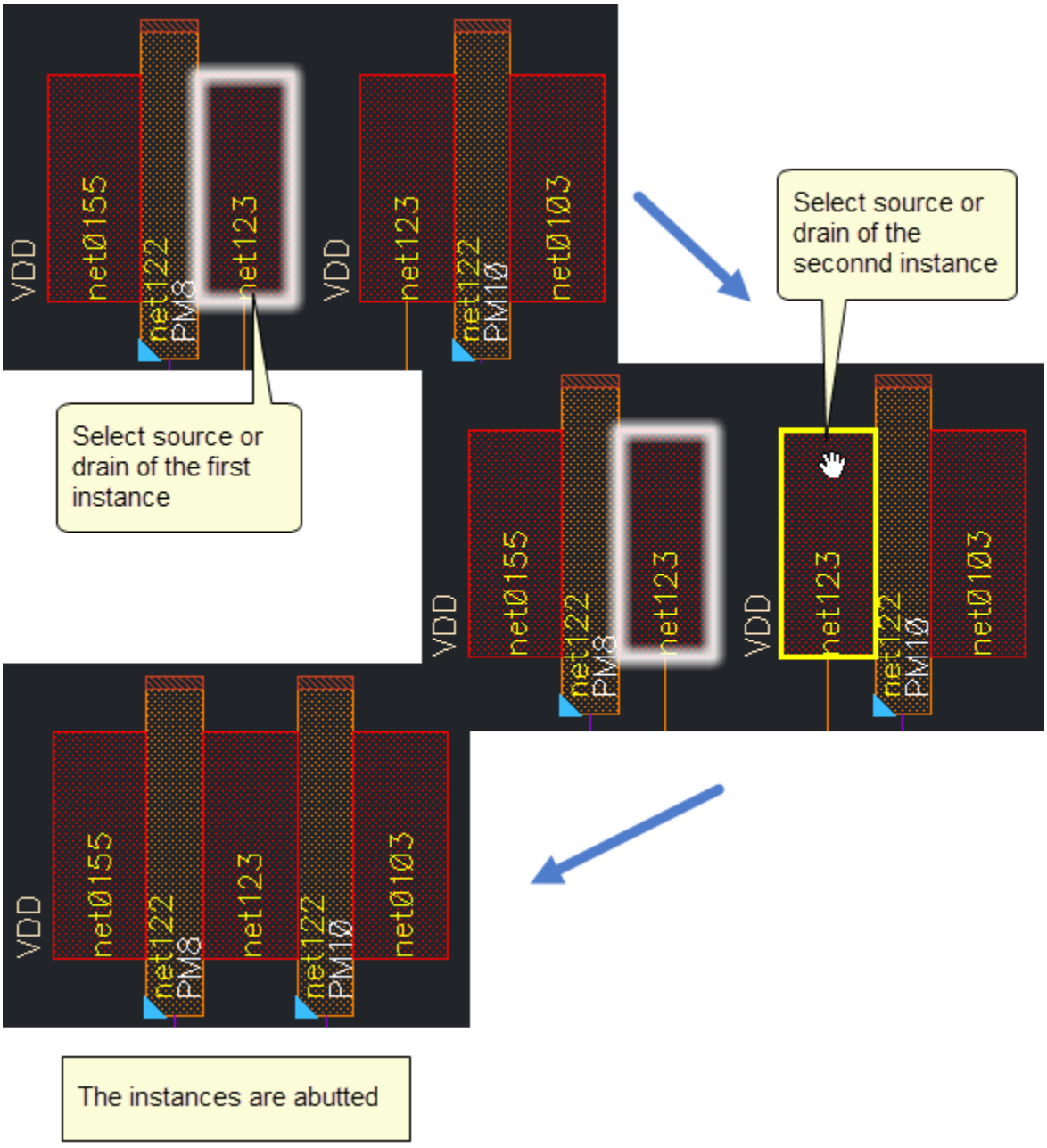
Abutting in Post-select Mode

To abut symbolic devices in post-select mode:

1. Choose *Edit – Abut* [b] or click the *Abut* button on the SPD toolbar.
2. Select source or drain of the first instance.
3. Select source or drain of the second instance. You will be able to select source and drain on same net as the first instance.

© 2023 Cadence Design Systems, Inc.

The selected instances are abutted.



Abutting Chains

To abut two chains in post-select mode:

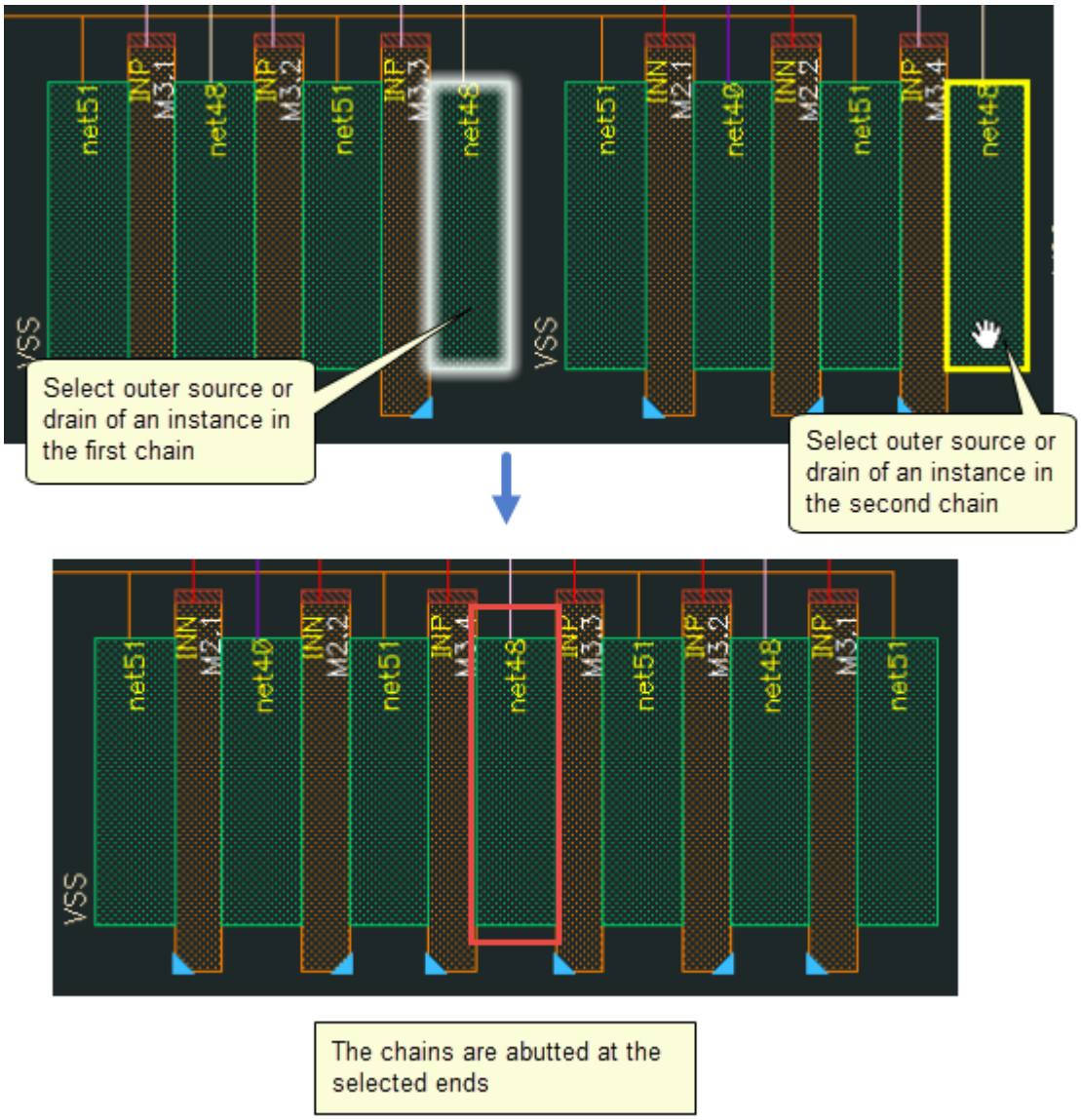
1. Choose *Edit – Abut* [b] or click the *Abut* button on the SPD toolbar.

© 2023 Cadence Design Systems, Inc.

2. Select the outer source or drain of the first chain.
3. Select the outer source or drain of the second chain.

   Note: You can select only the source or drain that is on the same net as the first instance.

   If these chains can be abutted at the selected ends, they are moved and abutted otherwise, a warning is displayed and abutment does not occur.



© 2023 Cadence Design Systems, Inc.

Change the Direction of Abutment

By default, the direction of abutment is set to left. You can change it to right using the Abut form. Changing the direction of abutment affects only when you are trying to abut in middle of a chain.

For more information, see Abutting in the Middle of a C hain.

To change the direction for abutment:

1. Choose *Edit – Abut* [b] or click the *Abut* button on the SPD toolbar.
2. Press `F3`.

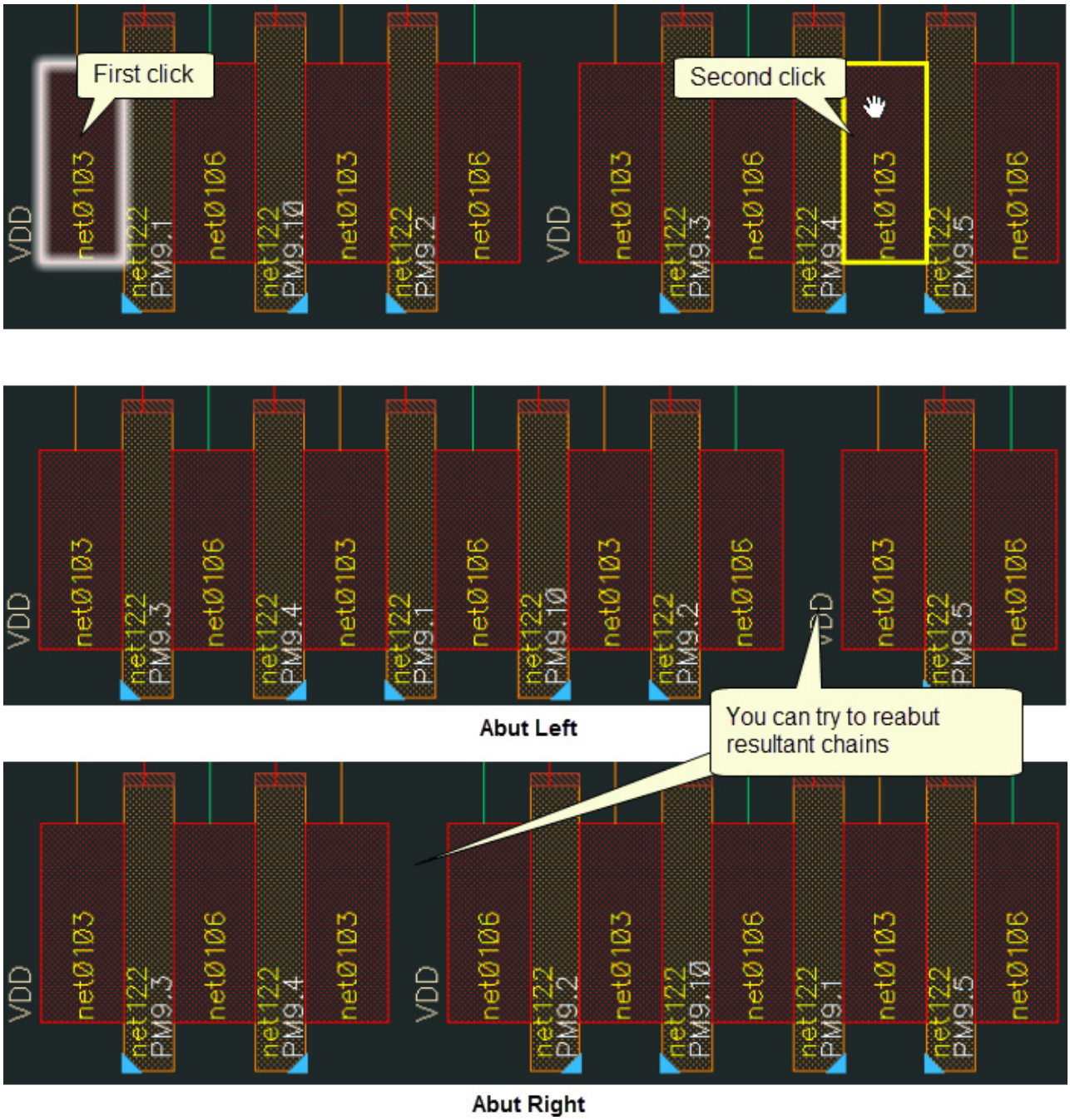
   The Abut form is displayed.

3. Change the direction for abutment and click *OK*.

© 2023 Cadence Design Systems, Inc.

Abutting in the Middle of a Chain

When you abut in the middle of a chain, the *Abut* command can move the chain and insert the first chain into another. Abutting in middle of chain works differently based on the direction for abutment. This is depicted in the figure below:



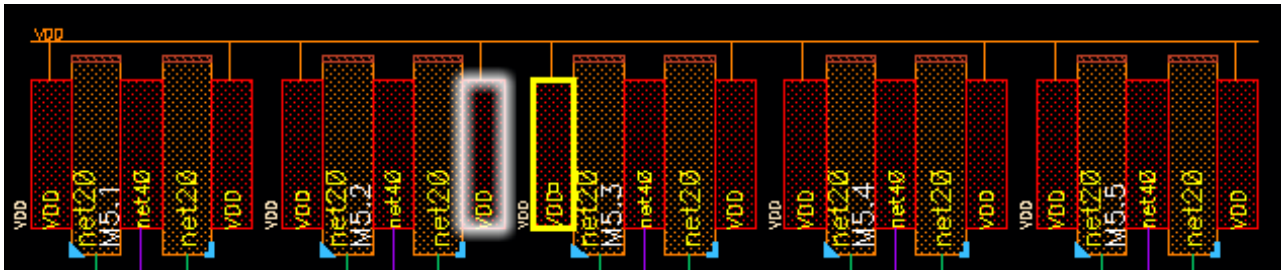
© 2023 Cadence Design Systems, Inc.

The abutment might break the chain. You can use the abut command again and try to abut the two chains.

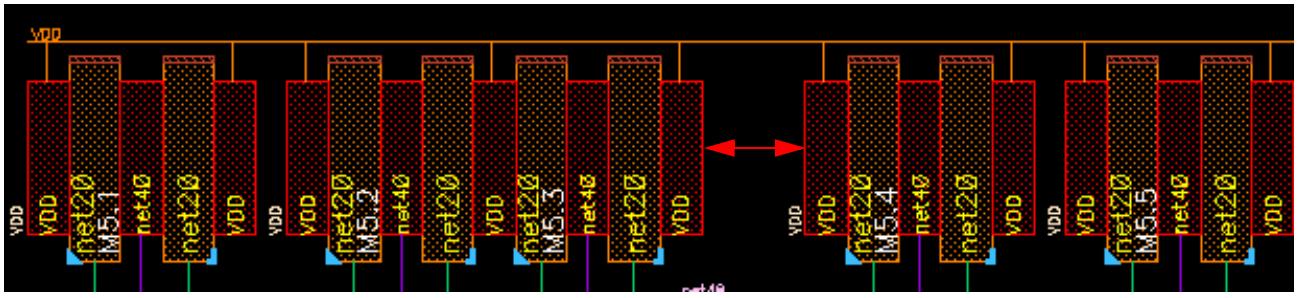
Abutting Devices Based on Whether Horizontal Packing is Enabled

When the *Options – SPD – Placement – Auto pack in X direction* check box is selected, devices are packed after abut. However, if this check box is deselected automatic packing of devices in the horizontal direction does not take place after abut. The following steps show this difference.

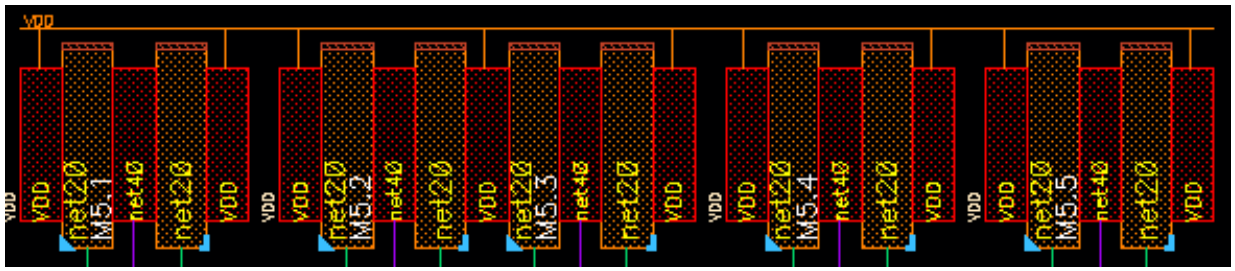
➡ Select devices to abut.



■ When the *Options – SPD – Placement – Auto pack in X direction* check box is not selected, devices are not packed after abut.



■ When the *Options – SPD – Placement – Auto pack in X direction* check box is selected, devices are automatically packed after abut.

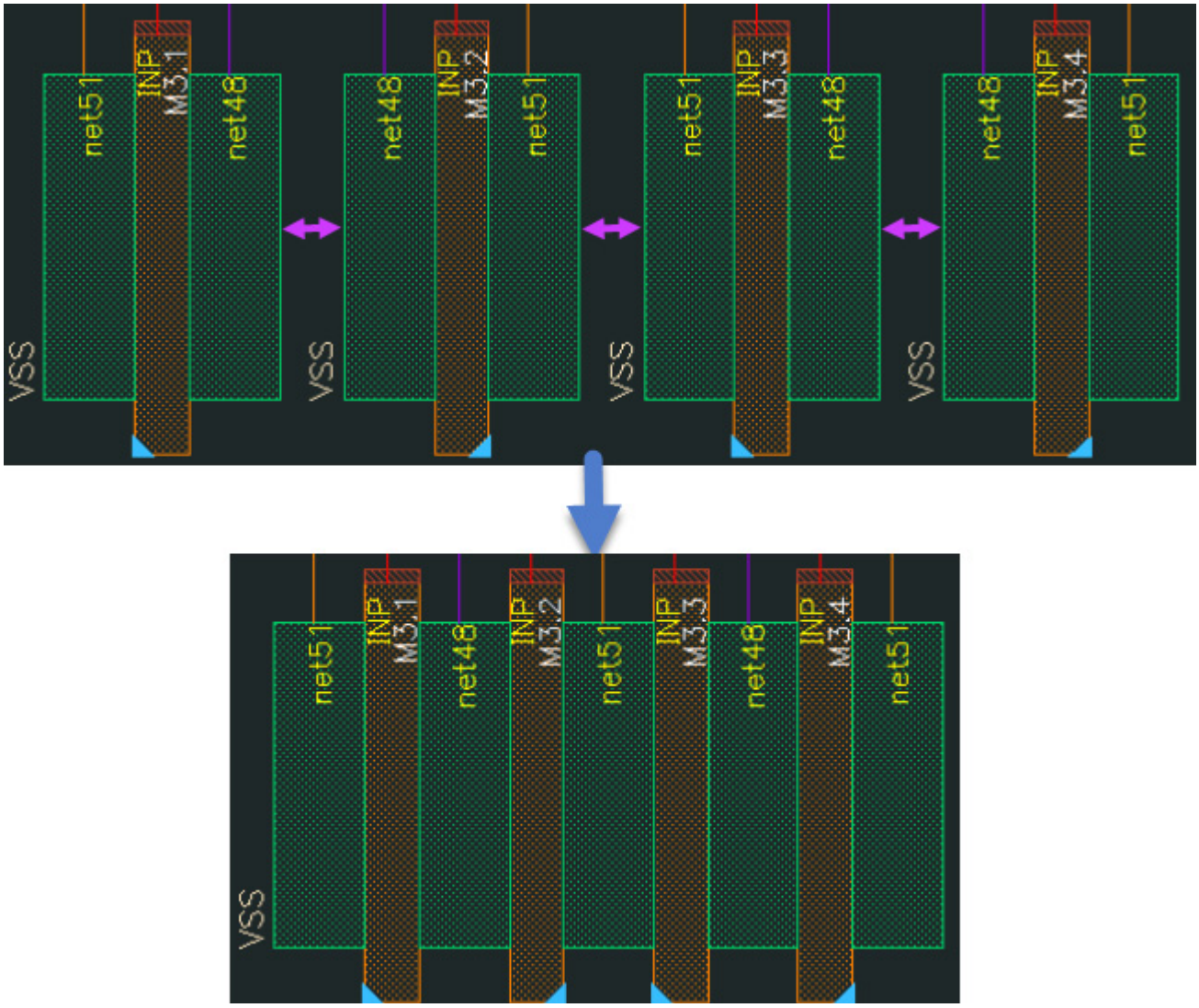


© 2023 Cadence Design Systems, Inc.

Abutting All Devices

To abut all the devices in the design:

- Choose *Edit – Abut All* or click the *Abut All* button on the SPD toolbar to abut all instances in the current design.



If the *Abut All* command is run on a single device, device is permuted or flipped automatically to maximize the abutment while retaining the device order.

Note: The *Abut All* command is independent of the *Options – SPD – Placement – Auto pack in X direction* option. This means that all devices are packed automatically after the *Abut All* command is run, whether *Auto pack in X direction* is selected or not.

© 2023 Cadence Design Systems, Inc.

is run, devices are packed automatically *Options – SPD – Placement – Auto pack in X direction* option

Unabutting Devices

You can perform the following device unabutment tasks from the SPD Editor:

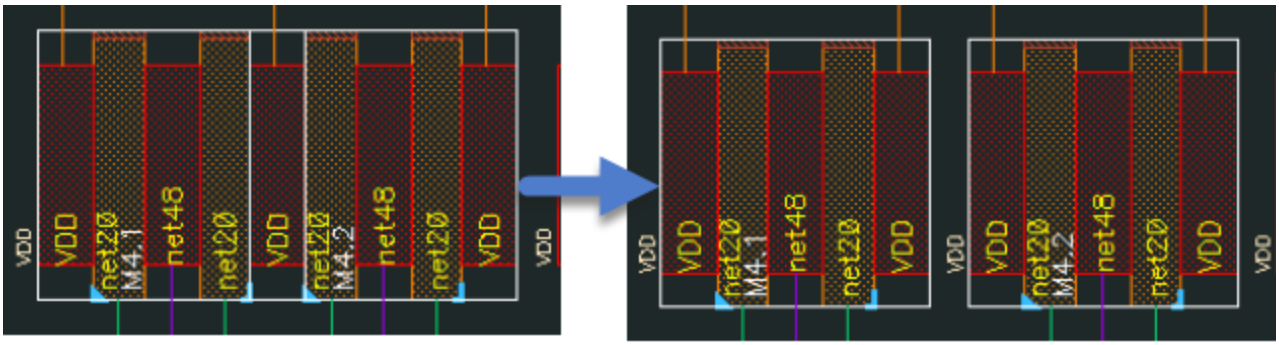
- Unabutting Devices in Pre-select Mode
- Unabutting Devices in Post-select Mode
- Unabutting Devices Based on Whether Horizontal Packing is Enabled
- Unabut All Devices

Unabutting Devices in Pre-select Mode

To unabut devices in pre-select mode:

1. Select two or more devices to unabut.
2. Choose *Edit – Unabut* [`Shift+b`] or click the *Unabut* button on the SPD toolbar to remove abutment from the selected devices.

   Notice that the unabut operation breaks the chain.



Unabutting Devices in Post-select Mode

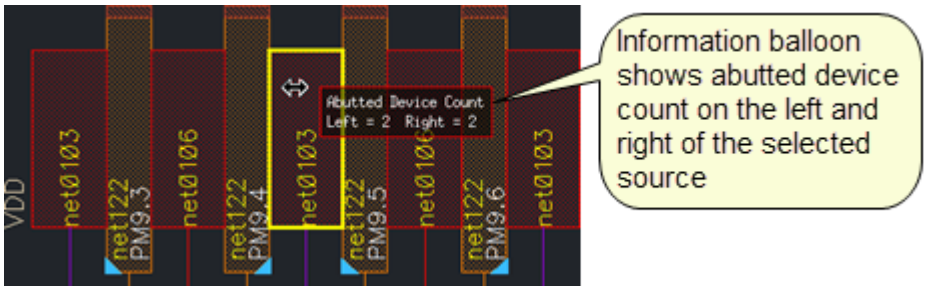
To unabut devices in post-select mode:

1. Choose *Edit – Unabut* or click the *Unabut* button on the SPD toolbar.

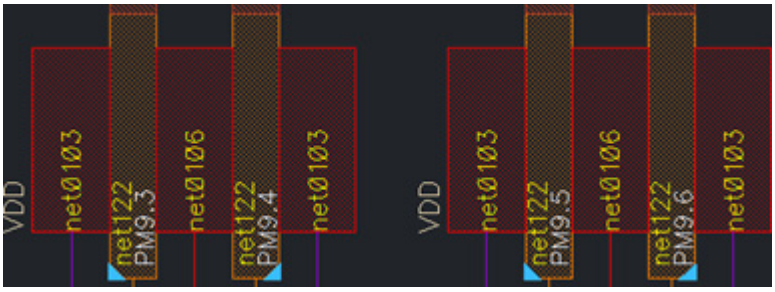
© 2023 Cadence Design Systems, Inc.

2. Choose source or drain to break the chain.

   When you pre-highlight a source or drain, an information balloon displays the number of abutted devices on the left and right of the selection, as shown in the figure below.



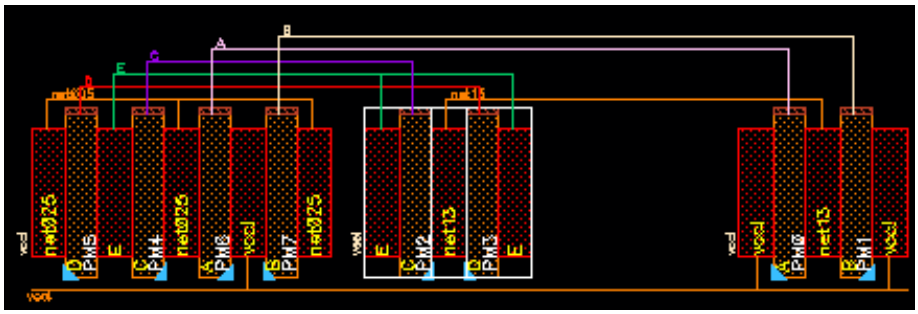
3. The devices are unabutted.



Unabutting Devices Based on Whether Horizontal Packing is Enabled

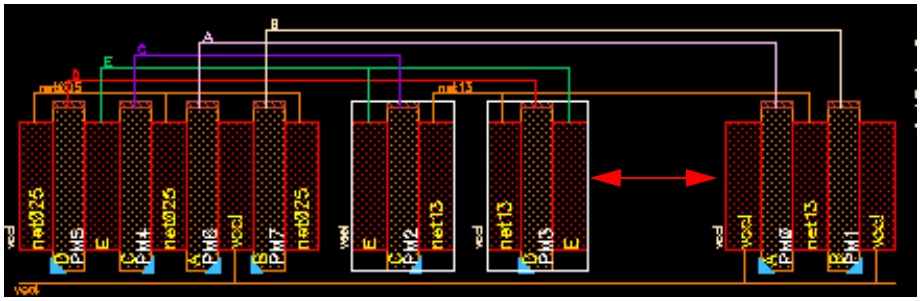
When the *Options – SPD – Placement – Auto pack in X direction* check box is selected, devices are packed after unabut. However, if this check box is deselected, automatic packing of devices in the horizontal direction does not take place after unabut. The following steps show this difference.

➡ Select devices to unabut.

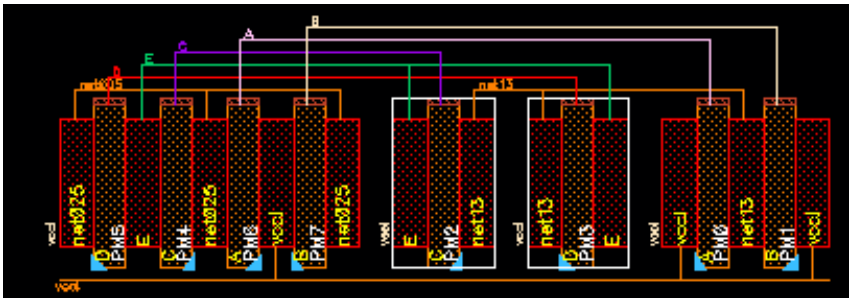


© 2023 Cadence Design Systems, Inc.

- When the *Options – SPD – Placement – Auto pack in X direction* check box is not selected, devices are not packed after unabut.



- When the *Options – SPD – Placement – Auto pack in X direction* check box is selected, devices are automatically packed after unabut.

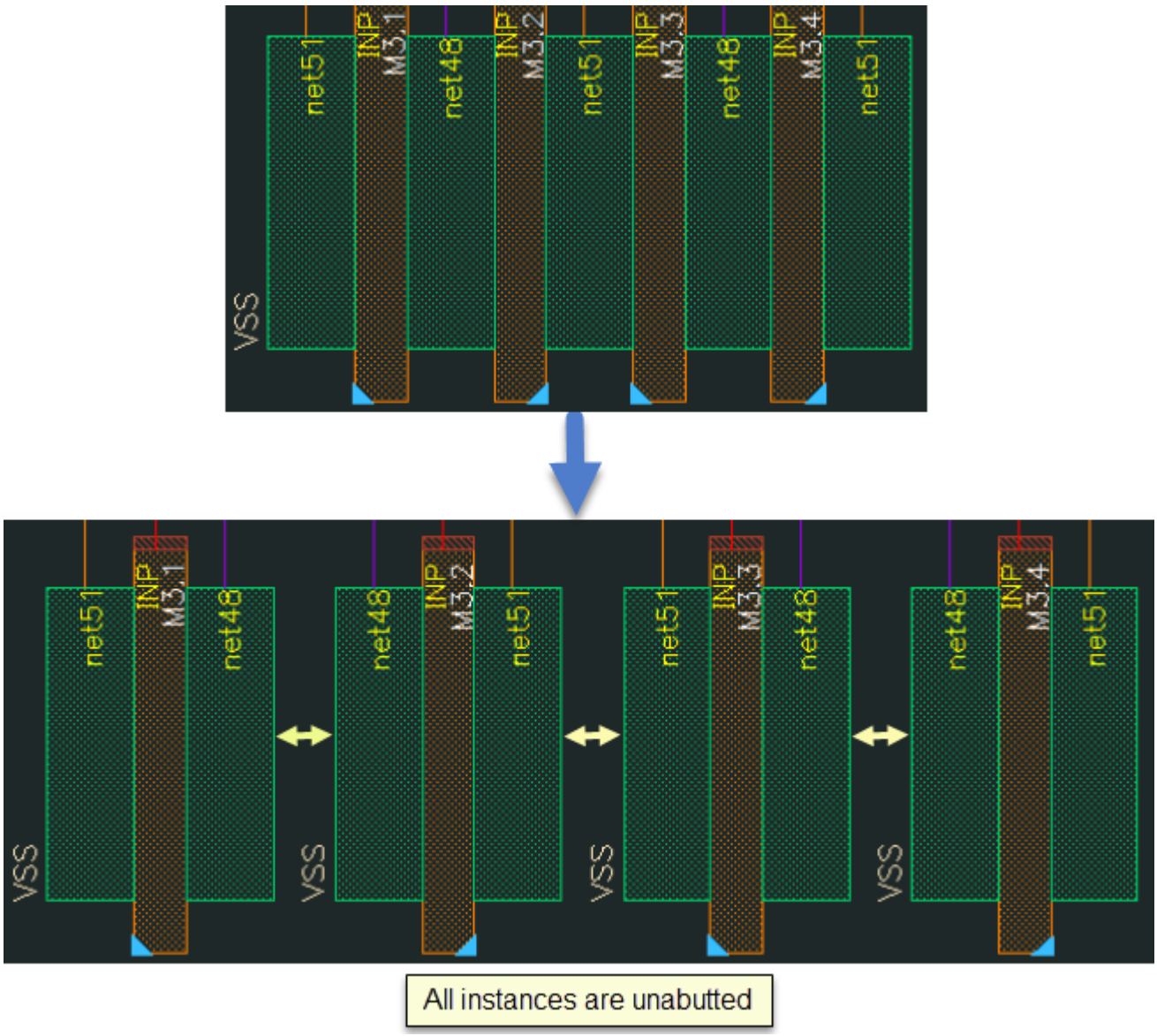


© 2023 Cadence Design Systems, Inc.

Unabut All Devices

To unabut all devices:

- Choose *Edit – Unabut All* or click the *Unabut All* button on the SPD toolbar to unabut all devices in the design.



Note: The *Unabut All* command is independent of the *Options – SPD – Placement – Auto pack in X direction* option. This means that all devices are packed automatically after the *Unabut All* command is run, whether *Auto pack in X direction* is selected or not.

© 2023 Cadence Design Systems, Inc.

Abutting Dummy Poly Devices

SPD supports dummy poly abutments only through user-defined abutment. You need to define and register callback functions in SKILL for dummy poly abutments.

Note: Reset Placement and some other SPD commands, such as Unabutting, Folding, and Permuting that trigger unabutment or modify CDF parameters or connectivity of instances, break existing user-defined abutments. In these cases, select the incorrectly abutted devices, unabut and reabut them using the correct user-defined abutment.

For more information on user-defined abutment SKILL functions, see Symbolic Placement of Devices Functions in *Virtuoso Layout Suite SKILL Reference*.

 Video

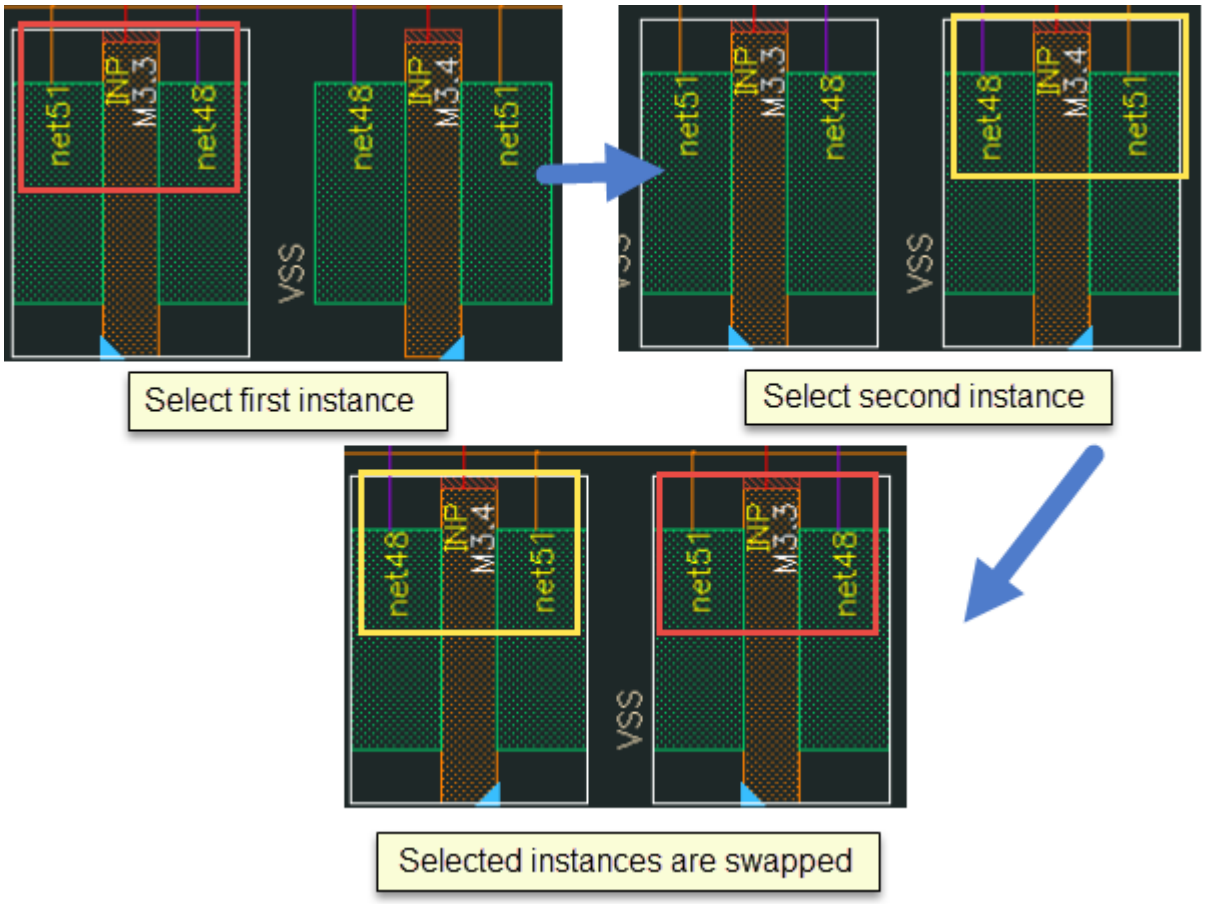
For a video on user-defined abutment and callback functions, see Using User-Defined Abutment and Callback Functions in SPD on Cadence Online Support.

© 2023 Cadence Design Systems, Inc.

Swapping Symbolic Devices

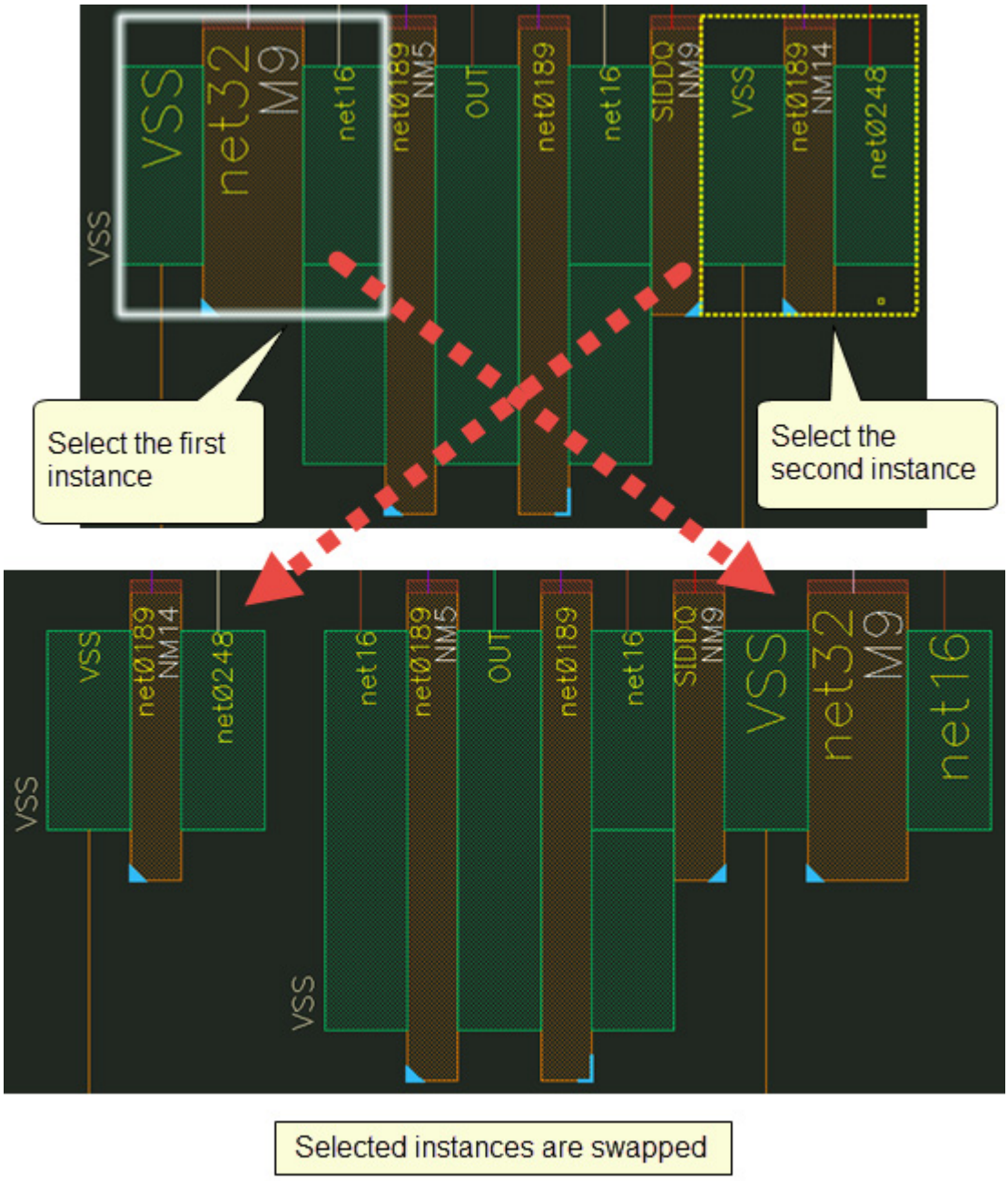
Use the *Edit – Swap* [S] command or click the *Swap* button on the SPD toolbar to swap the positions of symbolic instances in the SPD Editor. Flight lines are dynamically updated. The location of the symbolic instances is switched, but their orientation remains the same.

To swap two instances, select one of the above options and then click the instances that need to be swapped. In the following example, two adjacent instances in the same row are swapped:



© 2023 Cadence Design Systems, Inc.

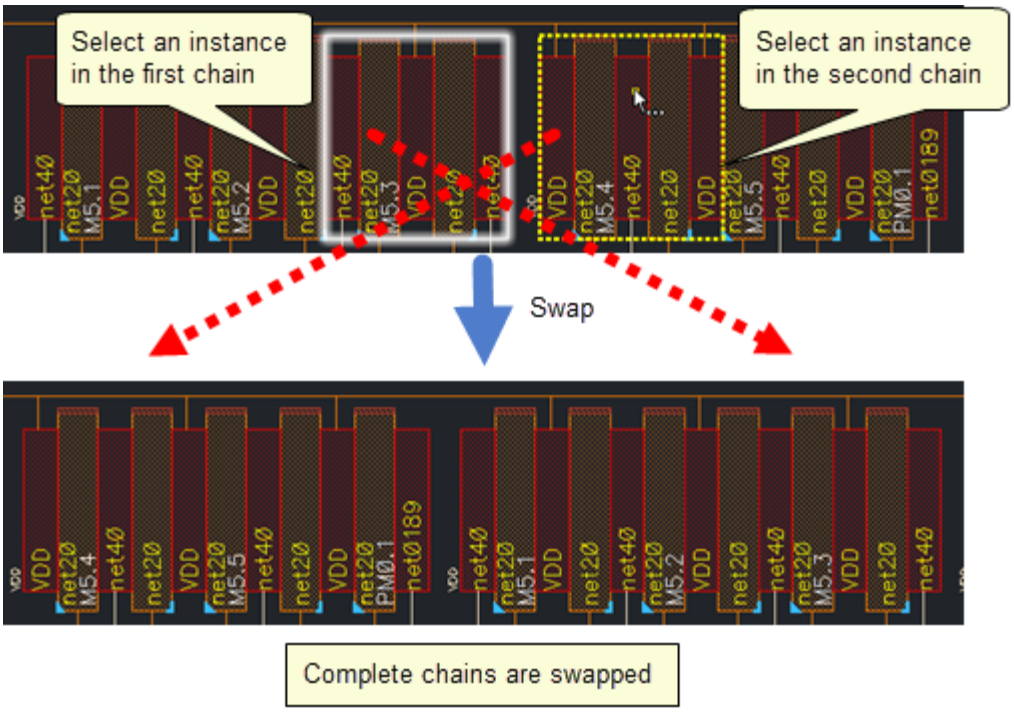
You can also swap non-adjacent instances in the same chain. In the following example, two non-adjacent instances in the same chain are swapped. The selected instances are swapped, but the original abutment is lost and the chain is broken.



Selected instances are swapped

© 2023 Cadence Design Systems, Inc.

The following example represents the swapping of instances in two different chains.

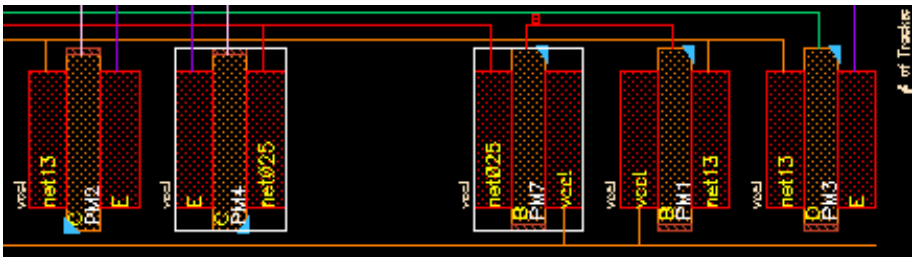


© 2023 Cadence Design Systems, Inc.

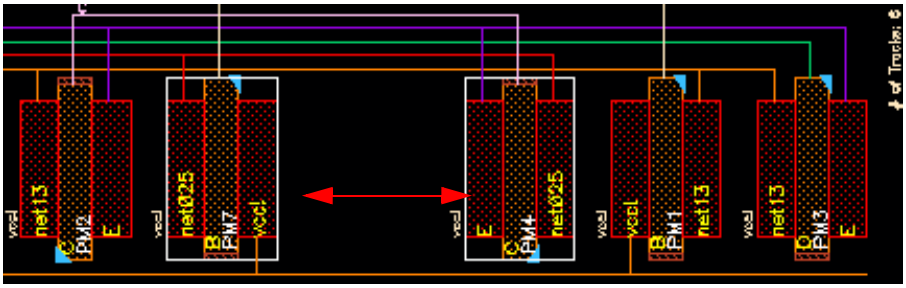
Swapping Devices Based on Whether Horizontal Packing is Enabled

When the *Options – SPD – Placement – Auto pack in X direction* check box is selected, devices are packed after you swap devices. However, if this check box is deselected, automatic packing of devices in the horizontal direction does not take place after swapping devices. The following steps show this difference.

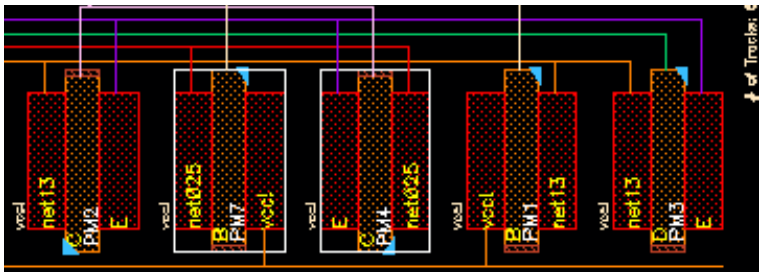
➡ Select the device you want to swap.



■ When the *Options – SPD – Placement – Auto pack in X direction* check box is not selected, devices are not packed after swap.



■ When the *Options – SPD – Placement – Auto pack in X direction* check box is selected, devices are automatically packed after swap.



© 2023 Cadence Design Systems, Inc.

Flipping Symbolic Devices

You may need to flip symbolic instances or chains horizontally or vertically to optimize placement. When you flip one instance, it is mirrored based on your selections. If you flip a chain, the chain order is reversed and all its instances are mirrored. Flight lines are dynamically updated.

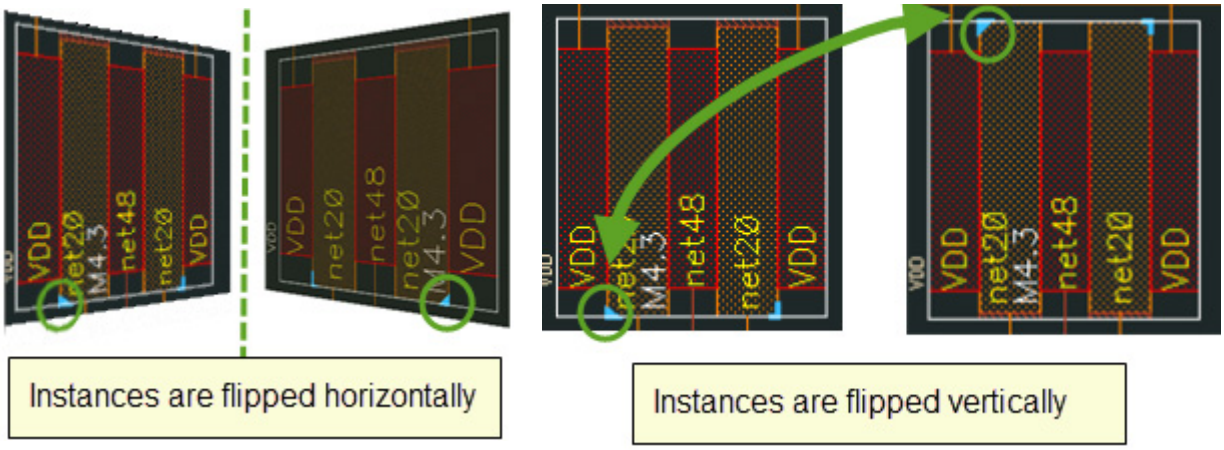
Flipping works differently depending on whether instances or chains are pre-selected or post-selected.

- Flipping in Pre-select Mode
- Flipping in Post-select Mode

Flipping in Pre-select Mode

To flip an instance or chain in pre-select mode:

1. Select one or more instances or chains to be flipped.
2. Select either *Edit – Flip Horizontal* [`Ctrl+j`] or *Edit – Flip Vertical* [`Shift+j`] or click the respective buttons on the SPD toolbar.
3. The selected instances or chains are flipped to their horizontal or vertical images.



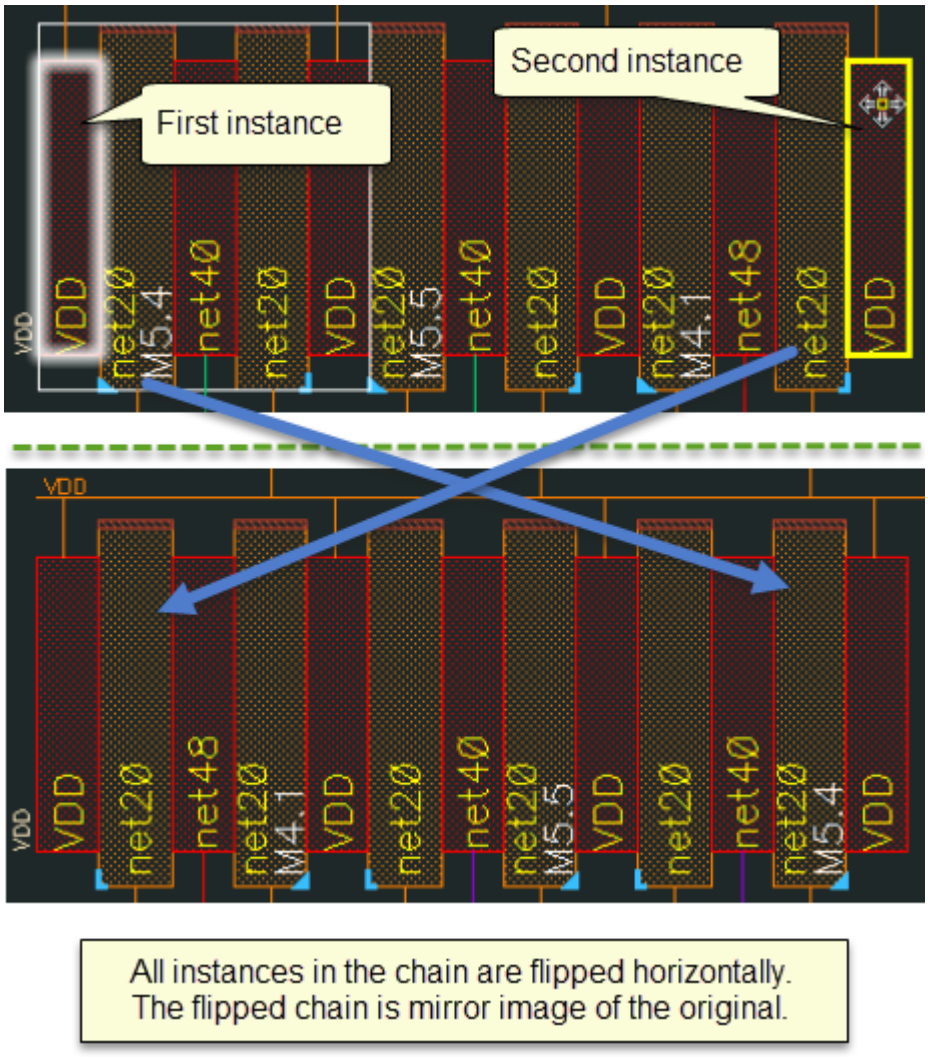
Flipping in Post-select Mode

To flip an instance or chain horizontally in post-select mode:

1. Select *Edit – Flip Horizontal* or click the *Flip Horizontal* button on the SPD toolbar.

© 2023 Cadence Design Systems, Inc.

2. Select the drain or source of the first instance.
3. Select the drain or source of the second instance.
4. All chains (and instances) that lie between the two selected instances are flipped.



All instances in the chain are flipped horizontally.
The flipped chain is mirror image of the original.

Note: The first and second instances can be same. In this case, only that instance is flipped.

To flip an instance or chain vertically in post-select mode:

1. Select *Edit – Flip Vertical* or click the *Flip Vertical* button on the SPD toolbar.
2. Select an instance.
3. The instance is flipped to its vertical mirror image.

© 2023 Cadence Design Systems, Inc.

Flipping Horizontal Using Permutation

You can use permutation if you need to flip symbolic instances or chains horizontally without changing their orientation.

- Flipping Instances Horizontally Using Permutation
- Flipping Chains Horizontally Using Permutation

Flipping Instances Horizontally Using Permutation

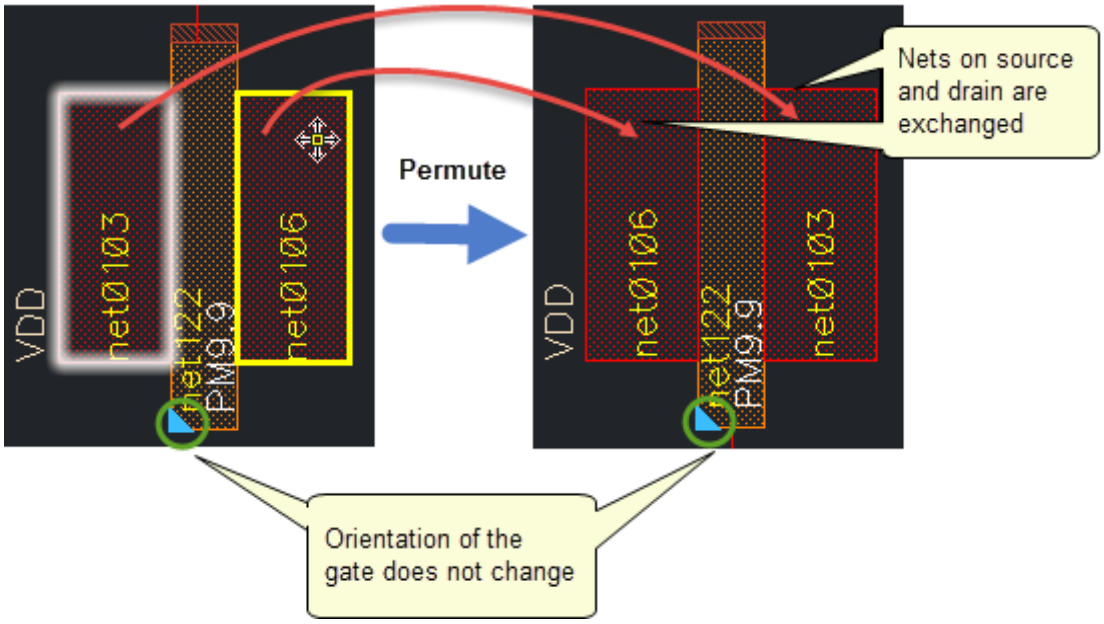
To flip instances horizontally using permutation:

1. Select either *Edit – Flip Horizontal* or click *Flip Horizontal* button on the SPD toolbar.
2. Press `F3` to display the Flip Horizontal form.

3. Select *Permute* and click *Hide*.
4. Select the drain and source of the instances to be permuted one by one.

   The instances are permuted. The nets of the source and drain are permuted, however, the orientation of the gate does not change.

© 2023 Cadence Design Systems, Inc.



Flipping Chains Horizontally Using Permutation

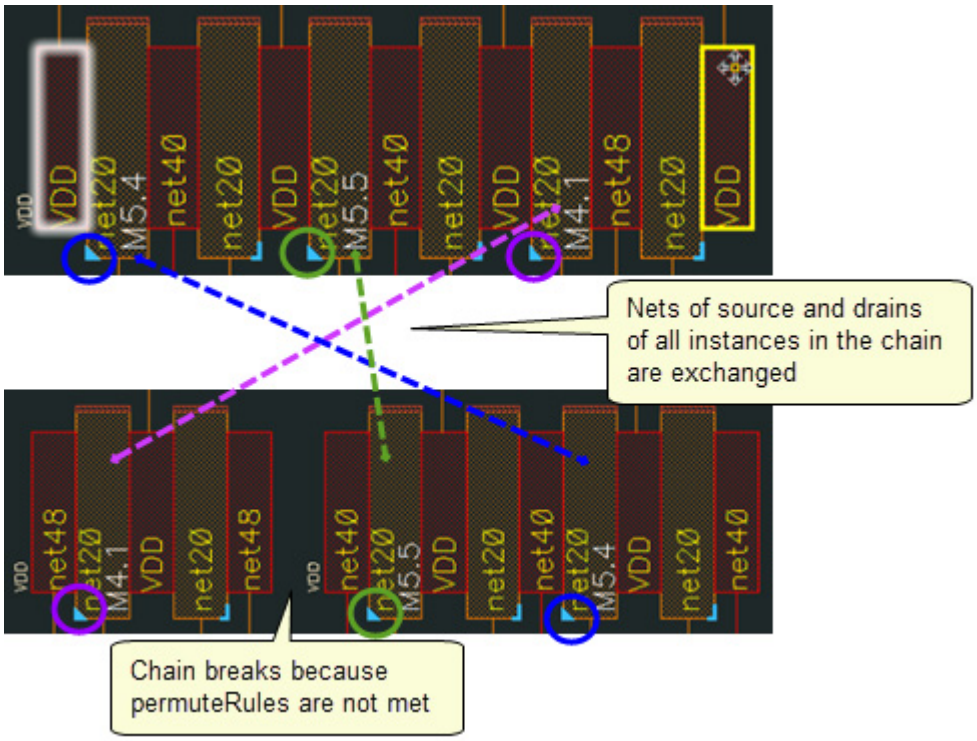
To flip chains horizontally using permutation:

1. Select either *Edit – Flip Horizontal* or click the *Flip Horizontal* button on the SPD toolbar.
2. Press `F3` to display the Flip Horizontal form.

3. Select *Permute* and click *Hide*.
4. Click the source or drain on the first instance to mark the start of the chain.
5. Click the source or drain of the last instance to mark the end of the chain.

© 2023 Cadence Design Systems, Inc.

Nets on the source and drain pins are exchanged for all instances in the chain, however, their original orientation is retained. Also, flight lines are dynamically updated and if needed, the chain is broken automatically to retain the original connections.



Important

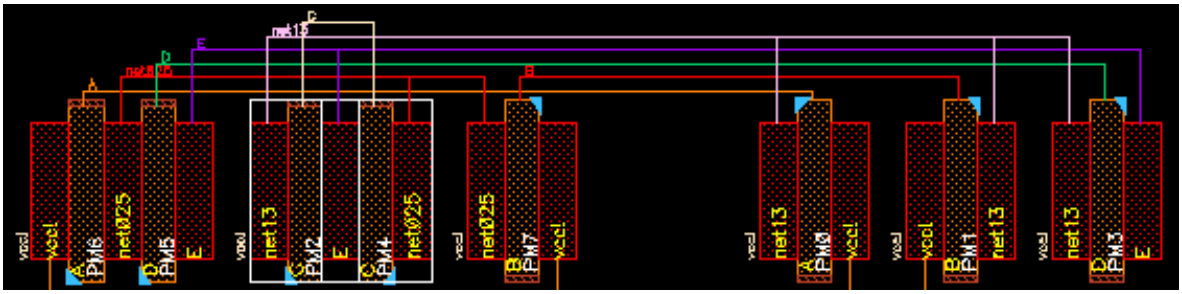
Pins can be permuted only if a `permuteRule` is appropriately defined. For more information, see `permuteRule` in the *Virtuoso Layout Suite XL User Guide*.

Horizontal Flipping Based on Whether Horizontal Packing is Enabled

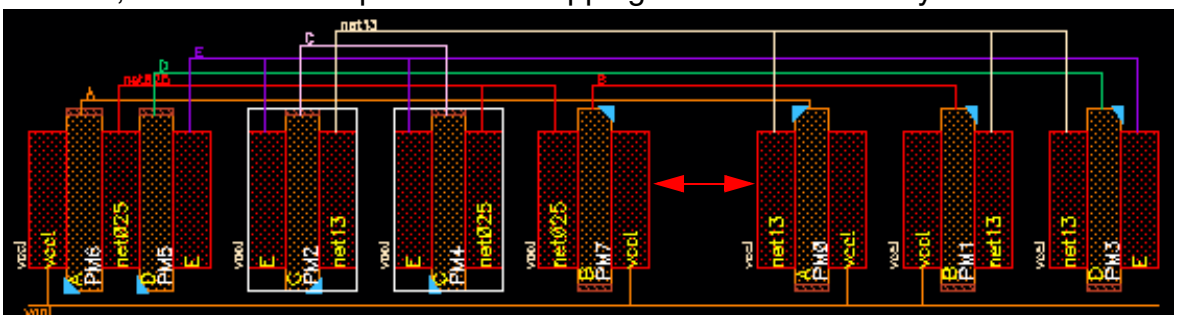
When the *Options – SPD – Placement – Auto pack in X direction* check box is selected, devices are packed after you flip devices horizontally. However, if this check box is deselected, automatic packing of devices in the horizontal direction does not take place after flipping devices horizontally. The following steps show this difference.

© 2023 Cadence Design Systems, Inc.

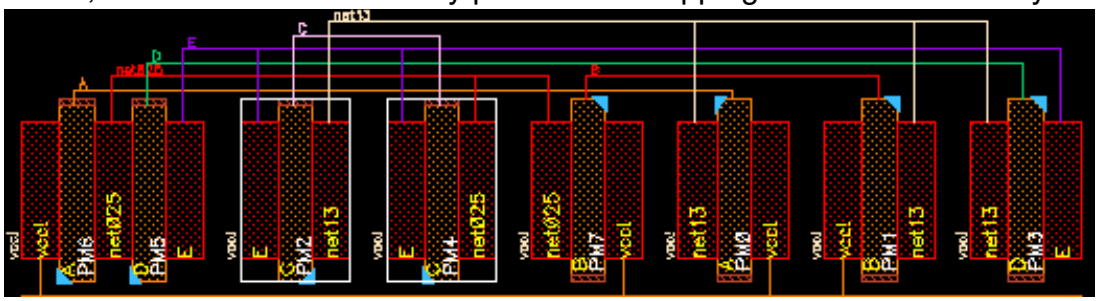
➡ Select the device you want to flip horizontally.



■ When the *Options – SPD – Placement – Auto pack in X direction* check box is not selected, devices are not packed after flipping devices horizontally.



■ When the *Options – SPD – Placement – Auto pack in X direction* check box is selected, devices are automatically packed after flipping devices horizontally.



© 2023 Cadence Design Systems, Inc.

Permuting Symbolic Devices

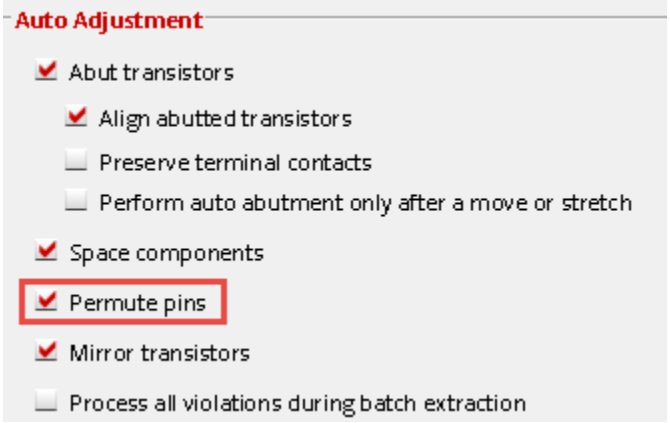
The *Permute* command permutes the devices without changing the orientation. The command works differently when the auto permutation is enabled or disabled in Layout XL.

To enable automatic pin permutation:

1. Choose *Options – Connectivity*.

   The Connectivity form is displayed.

2. On the *Generation* tab, select *Permute pins*.



3. Click *OK*.

For more information, see Layout XL form in *Virtuoso Layout Suite XL User Guide*.

Important

If you cannot permute an instance or chain in the real layout, you will not be able to permute the corresponding symbolic instance or chain in SPD. Additionally, pins can be permuted only if a `permuteRule` is appropriately defined. For more information, see `permuteRule` in the *Virtuoso Layout Suite XL User Guide*.

This section covers the following permutation tasks that you can perform from the SPD Editor:

- Permuting an Instance
- Permuting Multiple Instances
- Permuting Instance in a Chain
- Permuting Based on Whether Horizontal Packing is Enabled

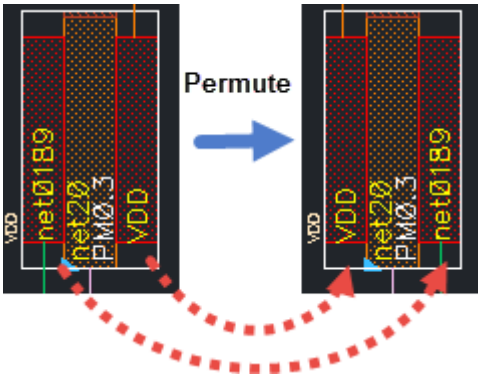
© 2023 Cadence Design Systems, Inc.

Permuting an Instance

To permute an instance:

1. Choose *Edit – Permute* [`Shift+p`] or click the *Permute* button on the SPD toolbar.
2. Select an instance to be permuted.

   The selected instance are permuted as show in the figure below.

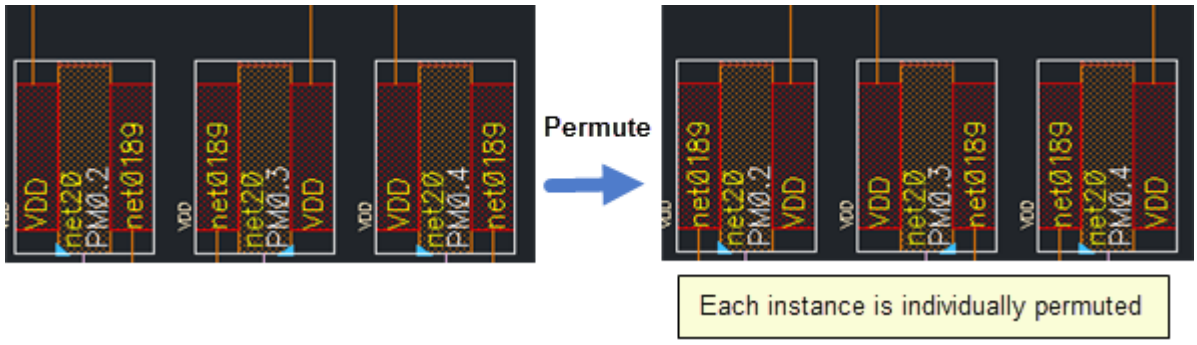


Permuting Multiple Instances

To permute multiple instances:

➡ Select all the instances you want to permute and then use the *Permute* command to permute them.

i



Permuting Instance in a Chain

To permute an instance in a chain:

1. Choose *Edit – Permute* [`Shift+p`] or click the *Permute* button on the SPD toolbar.

© 2023 Cadence Design Systems, Inc.

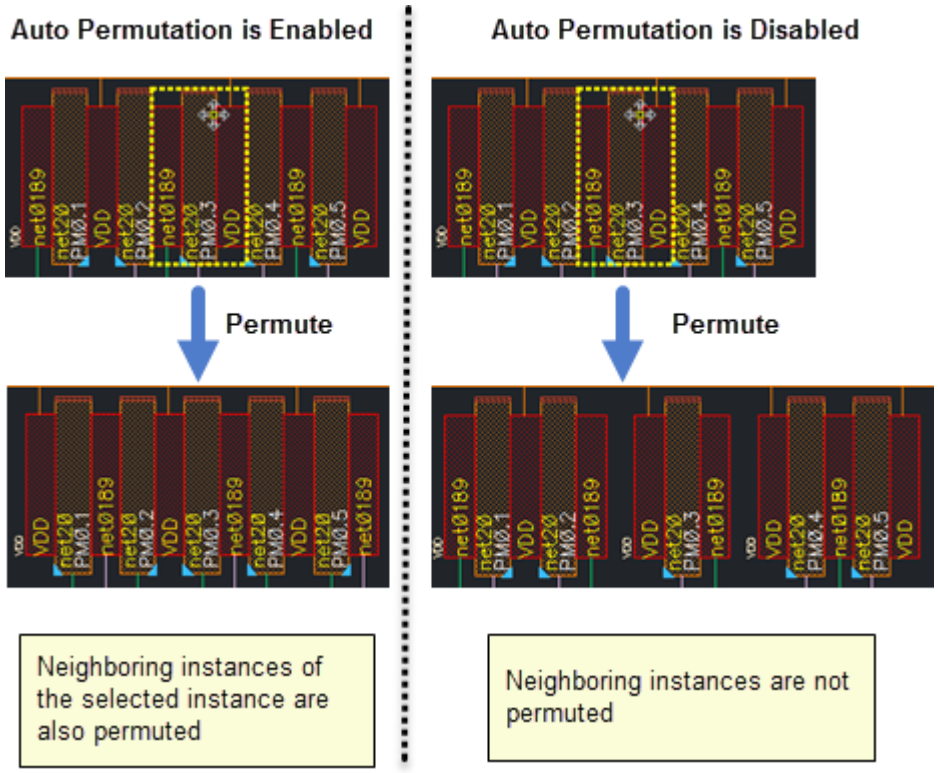
2. Select an instance in a chain to be permuted.

   The selected instance is permuted. Additionally, when auto permutation is enabled in Layout XL, the neighboring instances of the selected instance are also permuted recursively, if they can be abutted after permutation. Permutation stops until no more abutment is possible.

   When auto permutation is disabled in Layout XL, only the selected instance gets permuted.

   Note: The chain might break if two adjacent instances cannot abut after performing permutation.

   The following figure shows permutation of a chain based on the status of auto permutation in Layout XL.

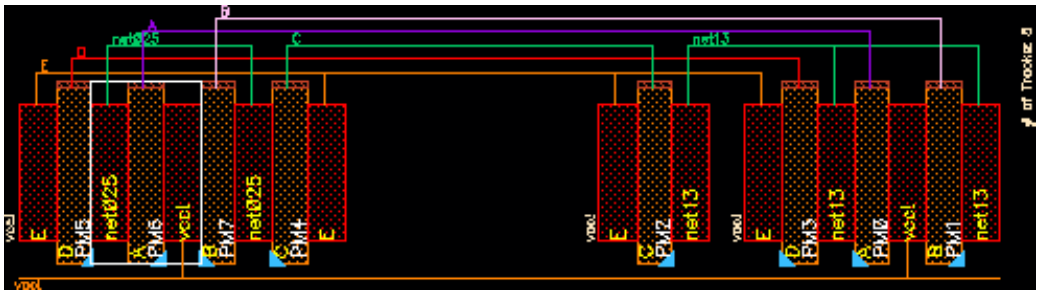


Permuting Based on Whether Horizontal Packing is Enabled

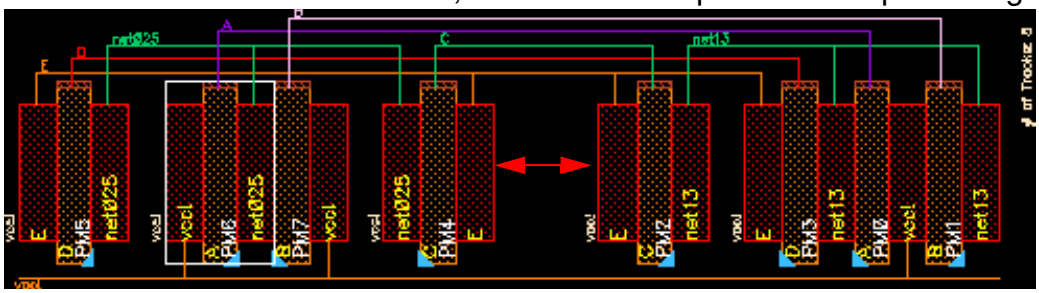
When the *Options – SPD – Placement – Auto pack in X direction* check box is selected, devices are packed after you permute devices. However, if this check box is deselected, automatic packing of devices in the horizontal direction does not take place after permuting devices. The following steps show this difference.

© 2023 Cadence Design Systems, Inc.

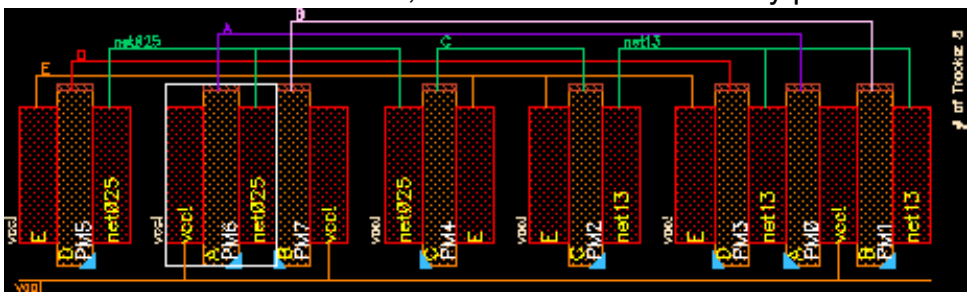
➡ Select the device you want to permute.



- When the *Options – SPD – Placement – Auto pack in X direction* check box is not selected, devices are not packed after permuting a device.



- When the *Options – SPD – Placement – Auto pack in X direction* check box is selected, devices are automatically packed after permuting a device.

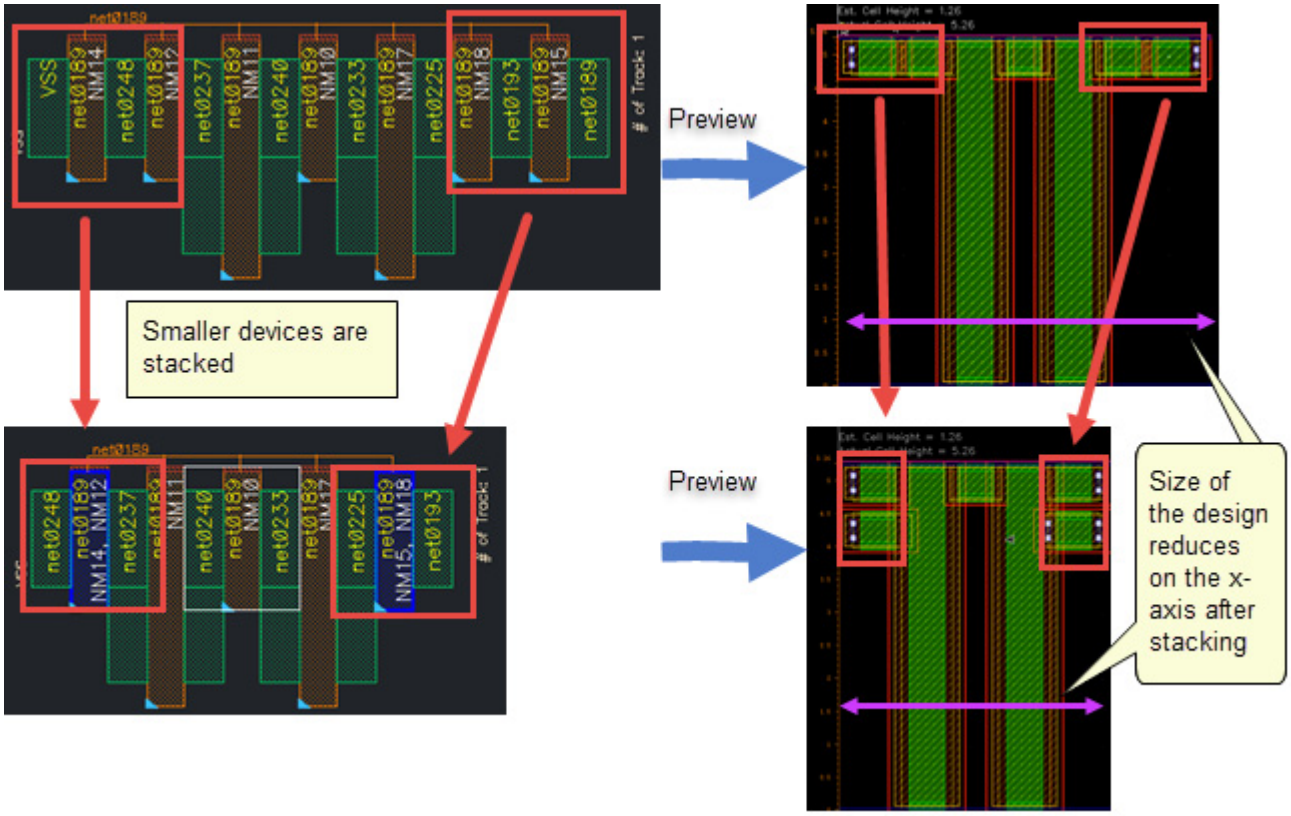


© 2023 Cadence Design Systems, Inc.

Stacking Symbolic Devices

Stacking helps in optimizing a design by helping you reduce the size of the design in the physical layout.

The following figure shows how the size of a design reduces when the devices are stacked in the symbolic design.



The SPD Editor provides the following stacking and unstacking options:

- Stacking a Device
- Editing a Stacked Device
- Stacking Based on Whether Horizontal Packing is Enabled
- Unstacking a Stacked Device
- Unstacking Based on Whether Horizontal Packing is Enabled
- Specifying Global Settings for Creating Stacked Devices
- Displaying List of Devices in the Stack

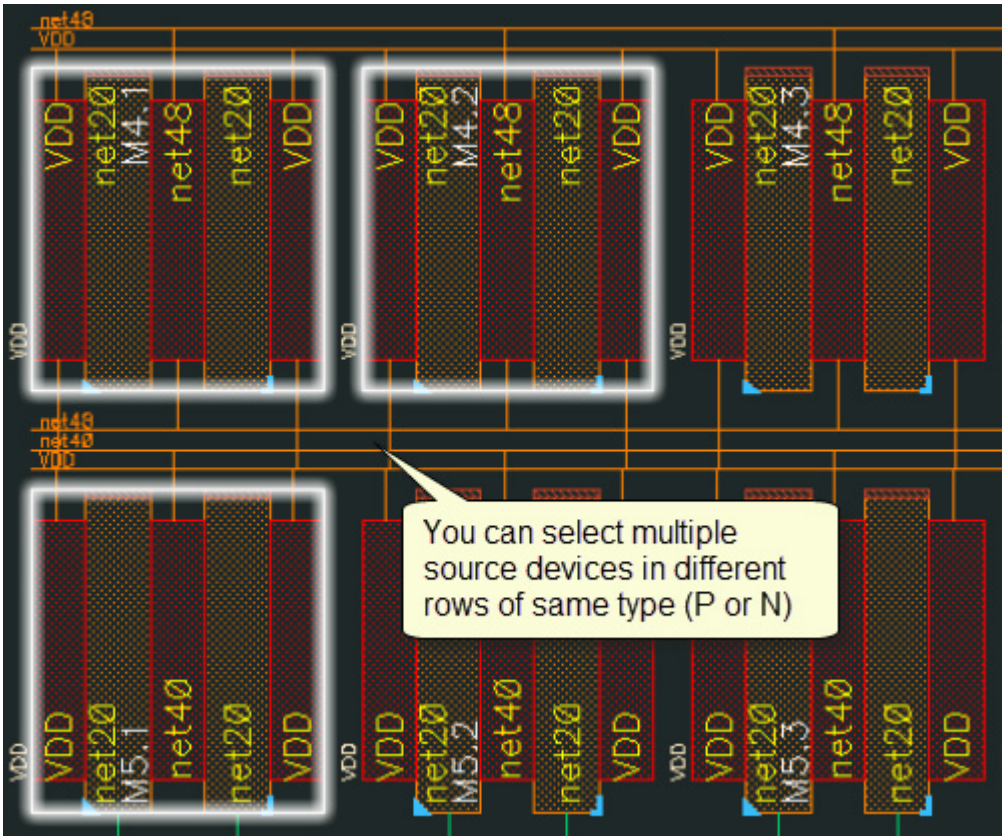
© 2023 Cadence Design Systems, Inc.

Stacking a Device

To create a stacked device:

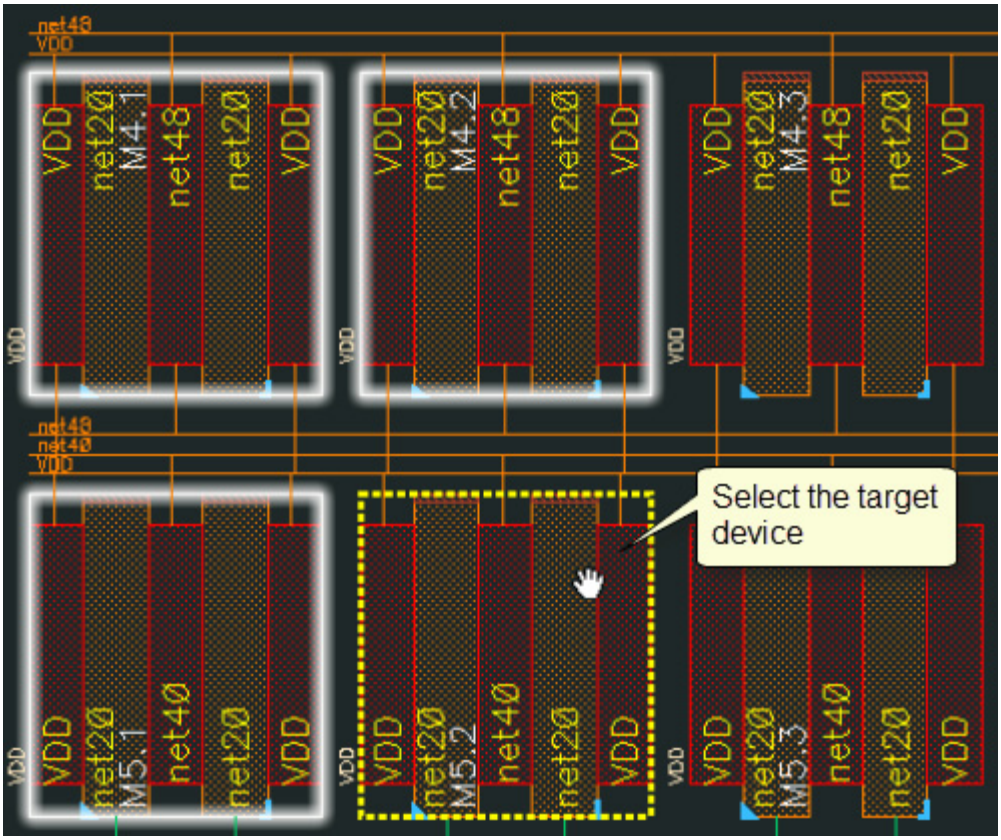
1. Select either *Edit– Stack* [`Ctrl+k`] or click the *Stack* button on the SPD toolbar.
2. Select one or more source devices for stacking.

   Note: Press [`Shift`] to select multiple source devices.



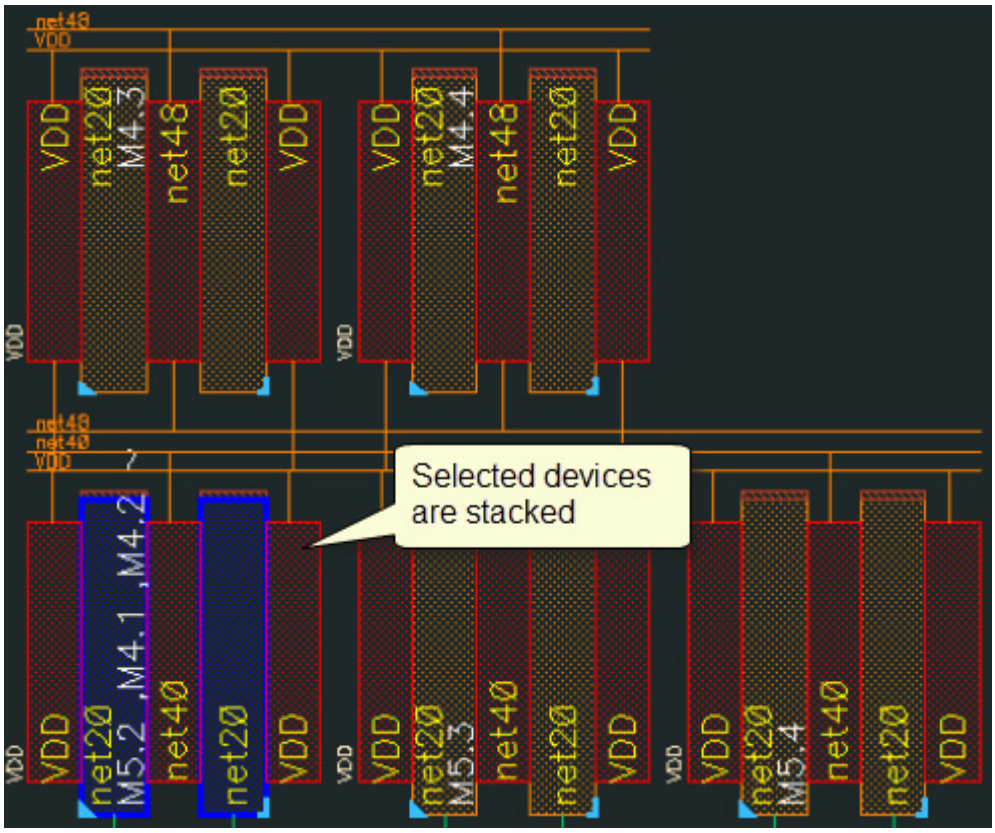
© 2023 Cadence Design Systems, Inc.

3. Select the target device.

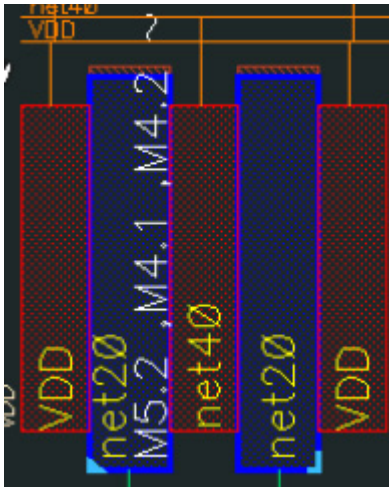


© 2023 Cadence Design Systems, Inc.

The devices are stacked as shown in the figure below.



The stacked device is highlighted in blue and it displays names of all devices that are in the stack.



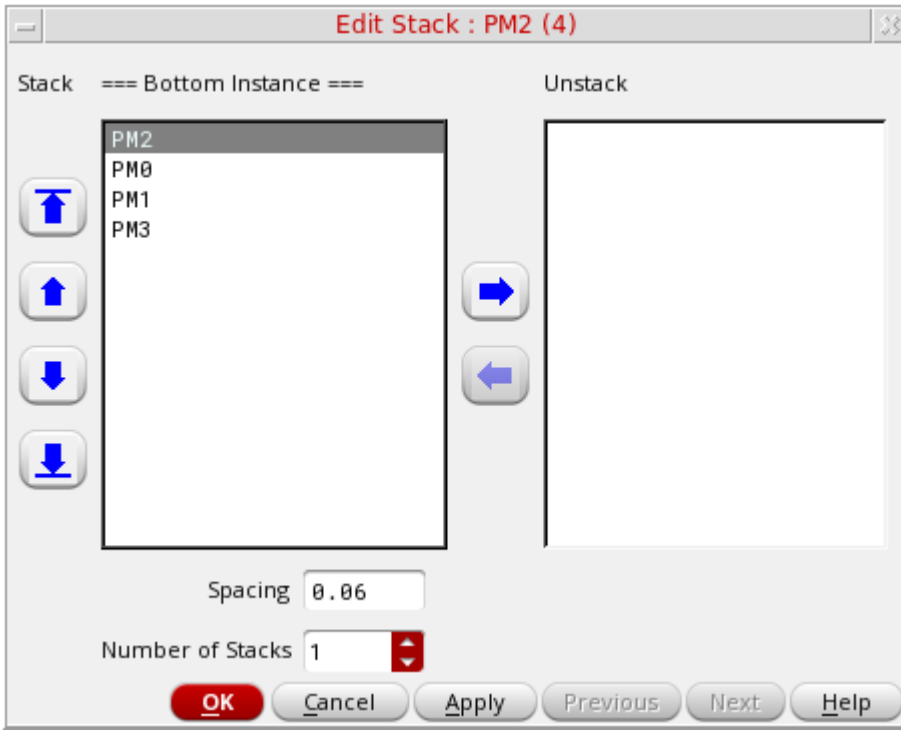
© 2023 Cadence Design Systems, Inc.

Editing a Stacked Device

To edit a stacked device:

1. Select either the *Edit – Edit Stack* [`Ctrl+e`] option or click the *Edit Stack* button on the SPD toolbar.

   The Edit Stack form is displayed.



   The top instance in the list is the bottom instance in the stack.

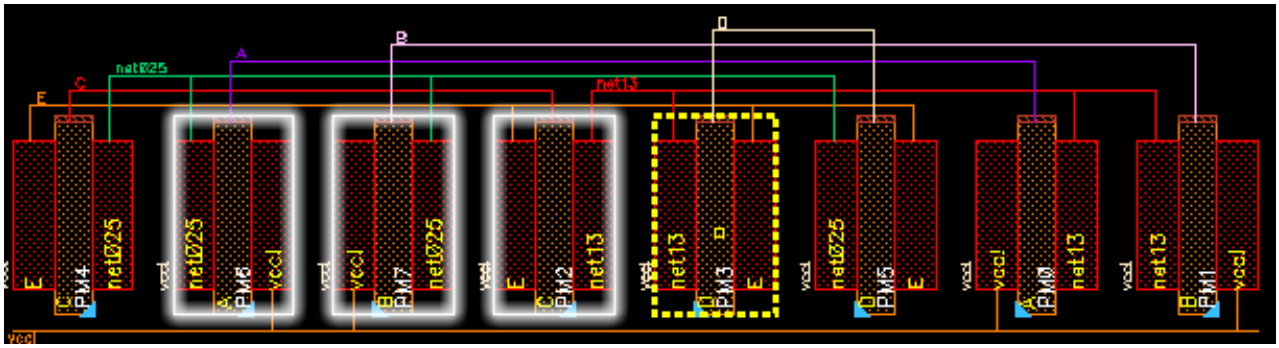
2. In the *Stack* box:
   - ❑ Use the up and down arrow keys to move a device up or down the stack.
   - ❑ Use the right arrow key to remove a device from the stack.
   - ❑ Use the left arrow key to add a device back to the stack.
3. Specify the spacing that should be applied between each device in the stack. The default values of P stacks and N stacks can be set using the Stack Options pane.
4. In the *Number of Stacks* box, specify the number of stacks you want to create. The default is `1`.
5. Click *OK* to save your changes and exit the Edit Stack form.

© 2023 Cadence Design Systems, Inc.

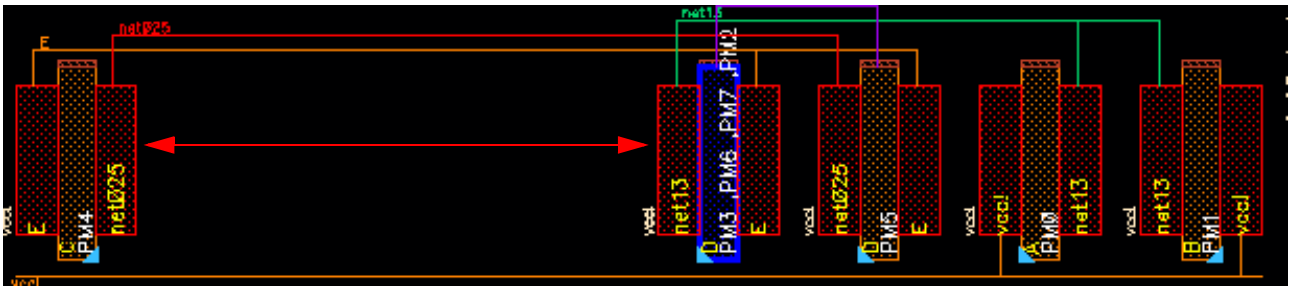
Stacking Based on Whether Horizontal Packing is Enabled

When the *Options – SPD – Placement – Auto pack in X direction* check box is selected, devices are packed after you stack devices. However, if this check box is deselected automatic packing of devices in the horizontal direction does not take place after stacking devices. The following steps show this difference.

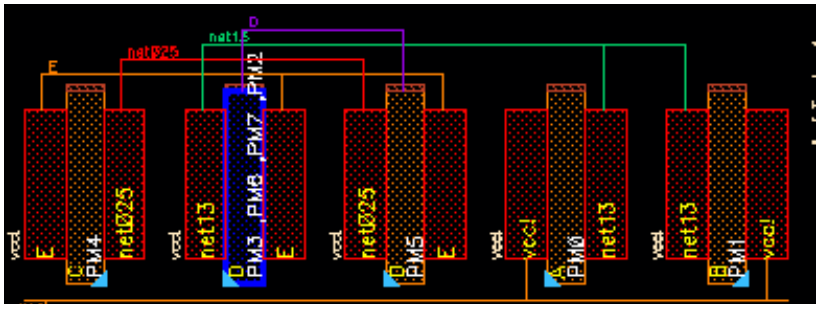
➡ Select the device you want to stack.



■ When the *Options – SPD – Placement – Auto pack in X direction* check box is not selected, devices are not packed after stacking.



■ When the *Options – SPD – Placement – Auto pack in X direction* check box is selected, devices are automatically packed after stacking.



Unstacking a Stacked Device

To unstack a stacked device:

© 2023 Cadence Design Systems, Inc.

1. Select the stacked device.
2. Select either the *Edit – Edit Unstack* [`Shift+k`] option or click the *Unstack* button on the SPD toolbar.

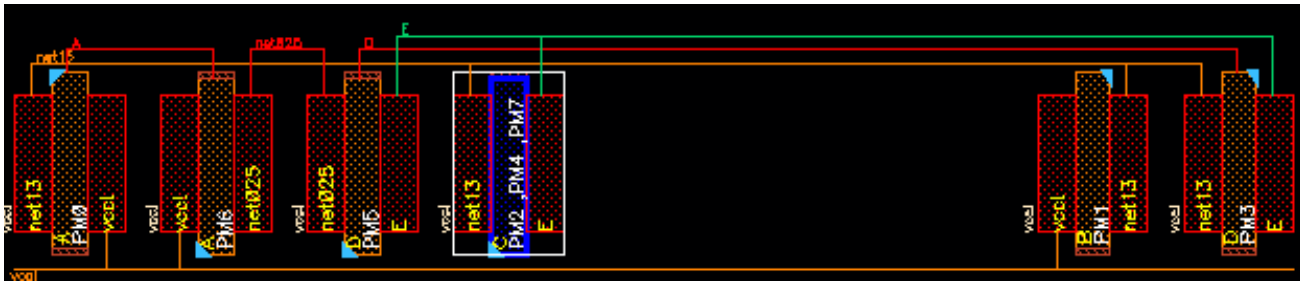
   The stacked device is unstacked and each instance is placed at the right of the bottom instance.

Note: You can also remove one or more individual devices from a stack using the Edit Stack form. See Editing a Stacked Device for more information.

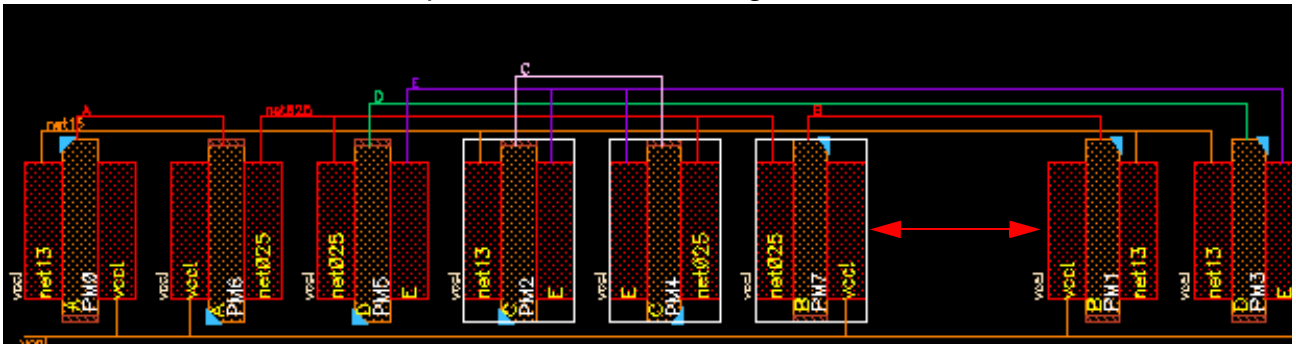
Unstacking Based on Whether Horizontal Packing is Enabled

When the *Options – SPD – Placement – Auto pack in X direction* check box is selected, devices are packed after you unstack devices. However, if this check box is deselected, automatic packing of devices in the horizontal direction does not take place after unstacking devices. The following steps show this difference.

➡ Select the device you want to stack.

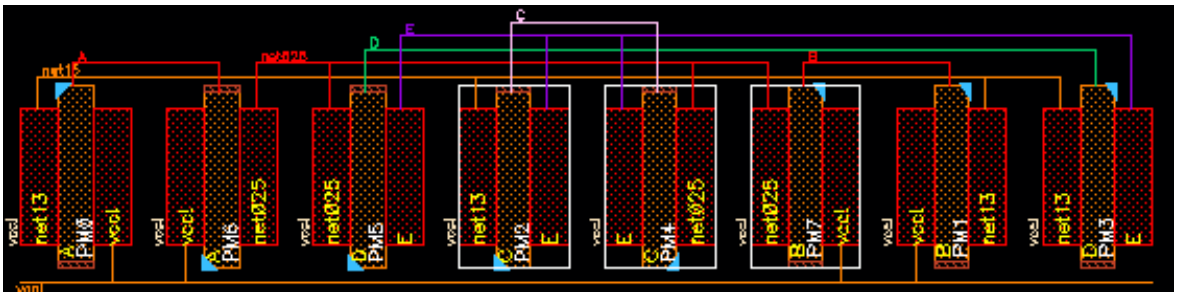


■ When the *Options – SPD – Placement – Auto pack in X direction* check box is not selected, devices are not packed after unstacking.



© 2023 Cadence Design Systems, Inc.

- When the *Options – SPD – Placement – Auto pack in X direction* check box is selected, devices are automatically packed after unstacking.



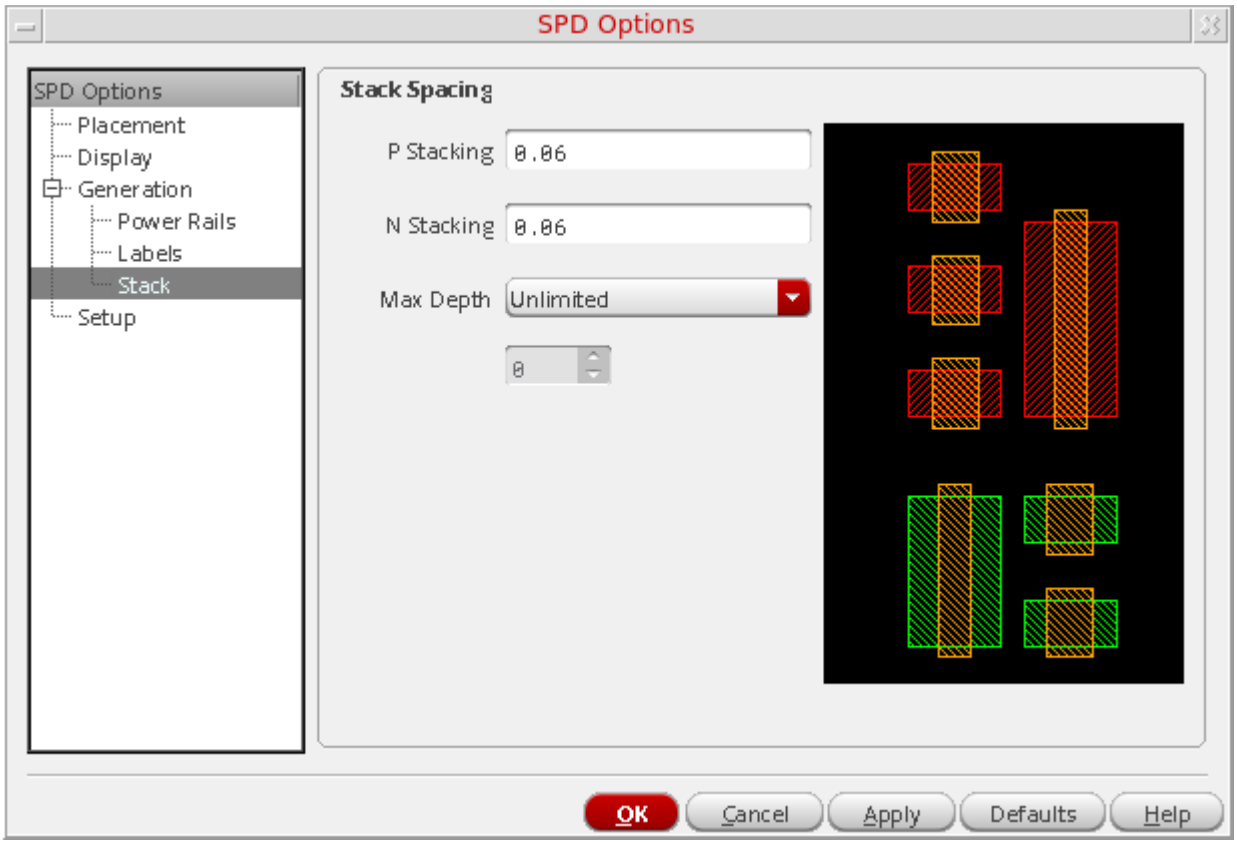
Specifying Global Settings for Creating Stacked Devices

You can also specify the general vertical spacing that should be maintained between stacked devices of P row and N row respectively.

1. Select *Options – SPD* to display the SPD Options form.
2. Select *Stack* from the *SPD Options* tree to display the Stack Options pane.

© 2023 Cadence Design Systems, Inc.

3. Here, specify *P Stacking*, *N Stacking*, and *Max Depth* values.



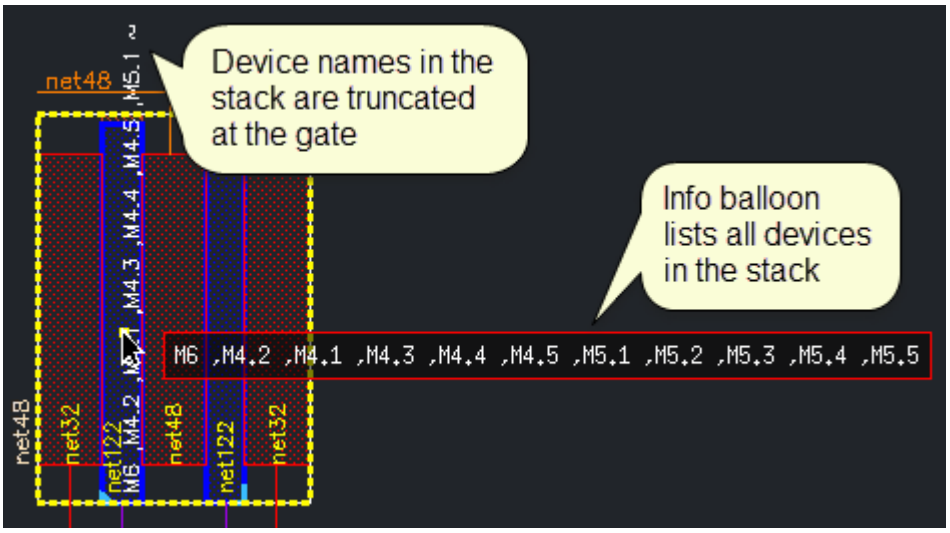
The settings made here apply globally to all stacked devices. However, you can override these setting for individual stacks, by specifying a different spacing in the Edit Stack form.

Displaying List of Devices in the Stack

SPD displays only the bottom device of the stack. Therefore, the names of all devices are displayed in the label of the gate of the stack. However, at times, the names of devices may not fit in the gate. In this case:

- Device names are truncated at the gate.
- Pointing at the gate displays an info balloon that lists all devices that are added to the stack.

© 2023 Cadence Design Systems, Inc.



Splitting Fingered Devices

The *Split Fingered Devices* option lets you place each finger of a symbolic finger device as a separate device.

For example, if you split an instance with three fingers, three separate instances, each with one finger, are generated after splitting.

To split a fingered device:

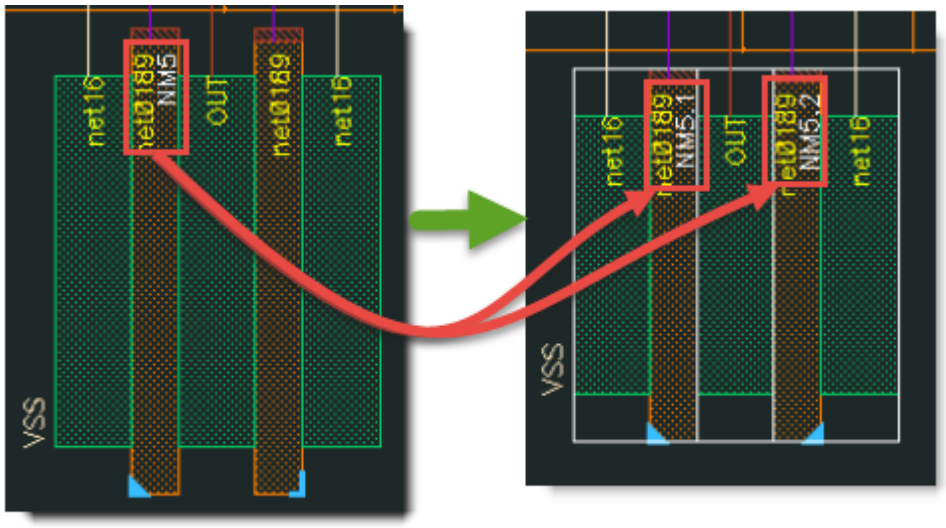
1. Select a fingered device.

   Note: In SPD, a fingered device is represented by a blue L added to the last finger of the fingered device.

2. Select *Edit – Split Fingered Device* [`Shift+s`] or click the *Split Fingered Device* button on the SPD toolbar.

© 2023 Cadence Design Systems, Inc.

The fingers in the device are split into separate instances.



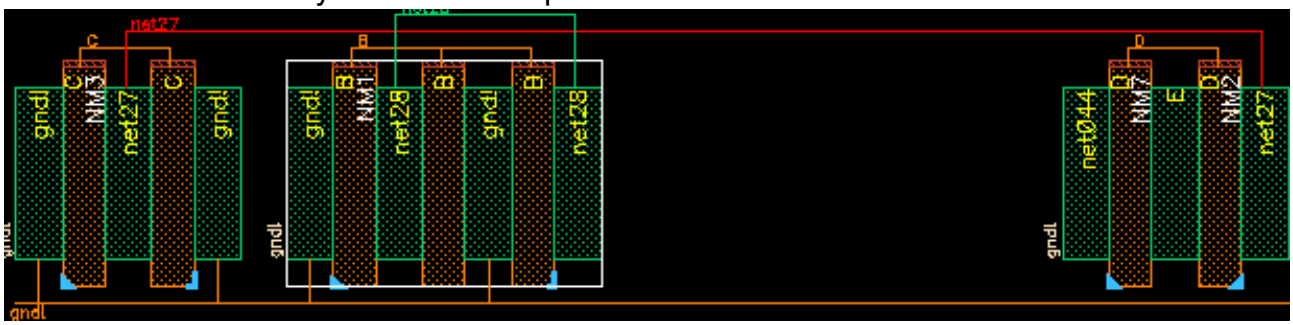
Instance NM5, comprising two fingers, is split into two instances NM 5.1 and NM 5.2

For more information, see Splitting Fingered Devices in *Virtuoso Layout Suite XL User Guide*.

Splitting Fingered Devices Based on Whether Horizontal Packing is Enabled

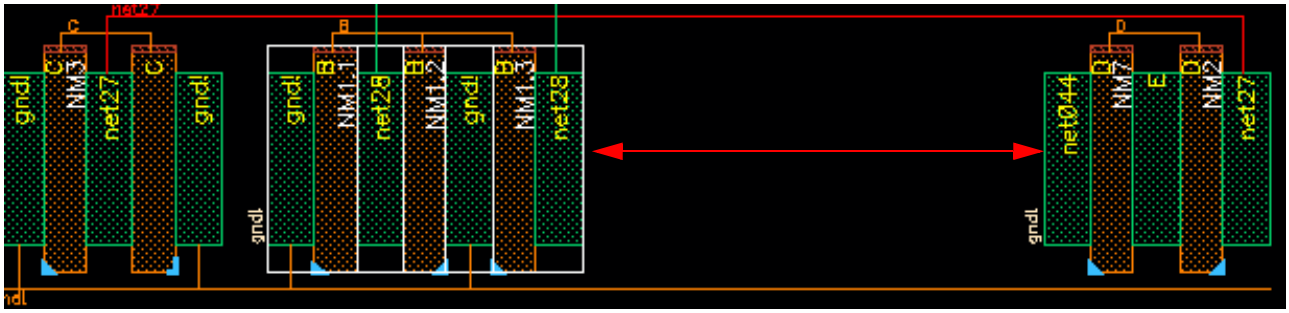
When the *Options – SPD – Placement – Auto pack in X direction* check box is selected, devices are packed after you split fingered devices. However, if this check box is deselected, automatic packing of devices in the horizontal direction does not take place after splitting fingered devices. The following steps show this difference.

➡ Select the device you want to swap.

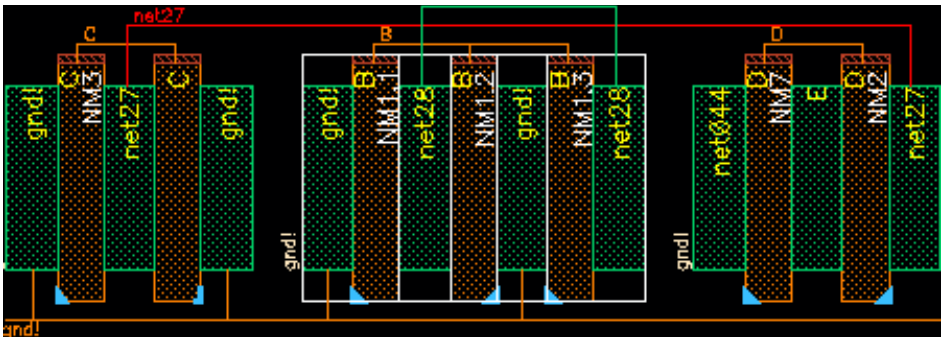


© 2023 Cadence Design Systems, Inc.

- When the *Options – SPD – Placement – Auto pack in X direction* check box is not selected, devices are not packed after splitting fingered devices.



- When the *Options – SPD – Placement – Auto pack in X direction* check box is selected, devices are automatically packed after splitting fingered devices.



© 2023 Cadence Design Systems, Inc.

Folding Symbolic Devices

Folding is the process of interactively dividing symbolic instances into two or more layout instances such that their terminals are connected in parallel to the same nets. You can fold the following types of symbolic instances:

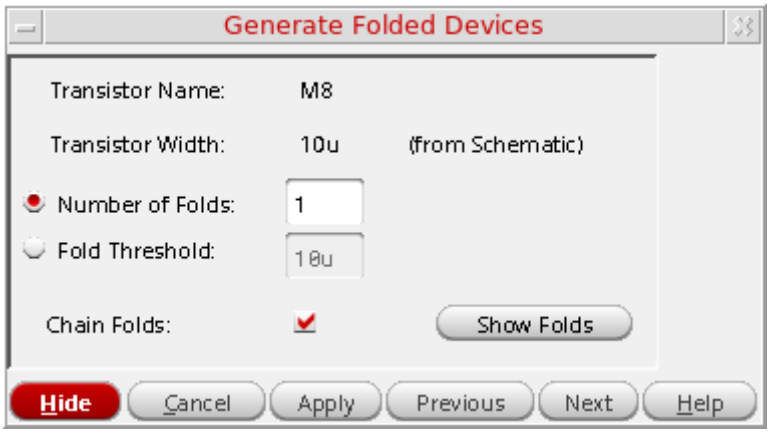
- An individual transistor or a chain of transistors
- A FinFET device or chain of FinFET devices

  Note: mfactored devices can be folded in SPD only if all devices are added to the SPD design.

To fold a transistor or a FinFET device:

1. Select objects to be folded.
2. Select *Edit – Fold* [○] or click the *Fold* button on the SPD toolbar.

   The Generate Folded Devices form is displayed.



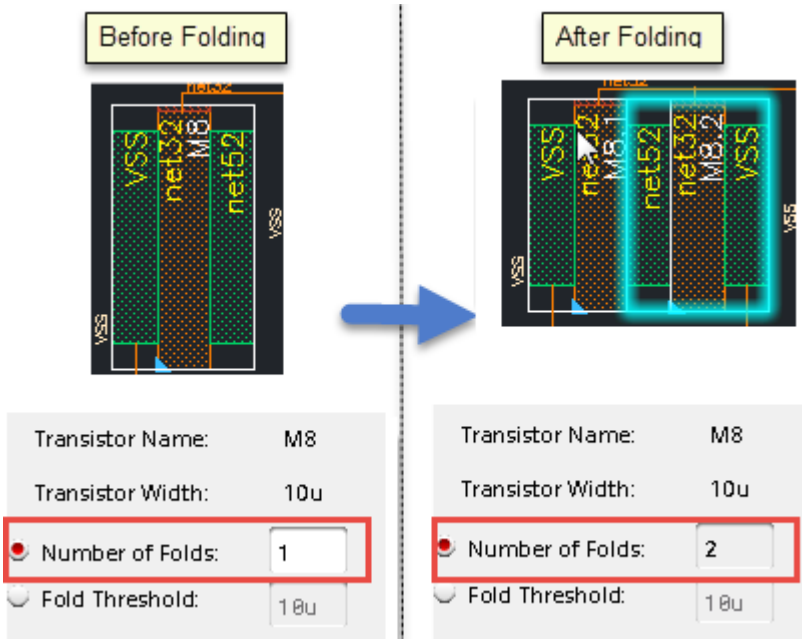
3. To fold devices, specify either the *Number of Folds* or click the *Folding Threshold* based on which the number of folds is calculated. The following sections cover the procedures for folding devices:
   - By specifying the number of folds
   - By Specifying the Number of Folds for a FinFET Device
   - By Specifying the Fold Threshold

By specifying the number of folds

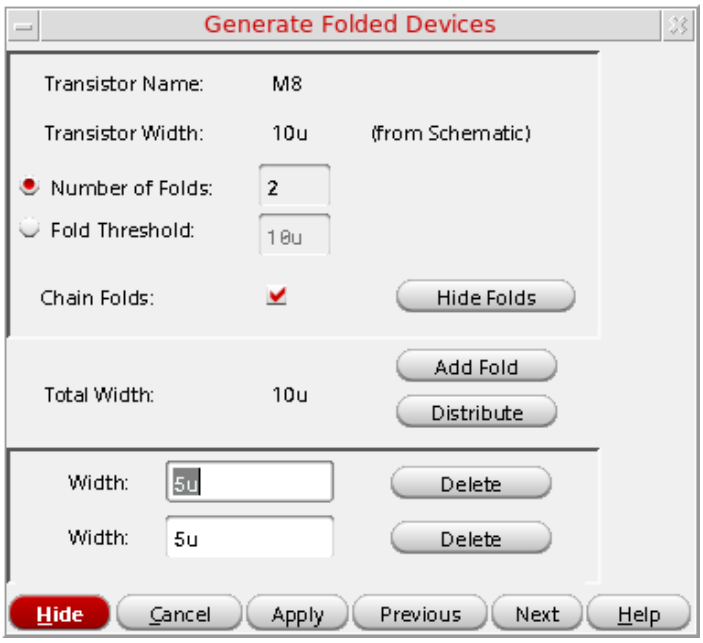
To fold a device by specifying number of folds:

© 2023 Cadence Design Systems, Inc.

1. Type the *Number of Folds* into which you want to divide the selected device. This value must be greater than the current value in this field.
2. Click *Apply*.



Click *Show Folds* or *Apply* to display the widths of the individual folds.



© 2023 Cadence Design Systems, Inc.

In the above example, the specified *Number of Folds* is 2. Therefore, the total transistor width is equally distributed between the two folds.

You can perform the following tasks:

- Click *Add Fold* to add new folds.

  Note: The *Number of Folds* field is updated dynamically when you add or delete folds.

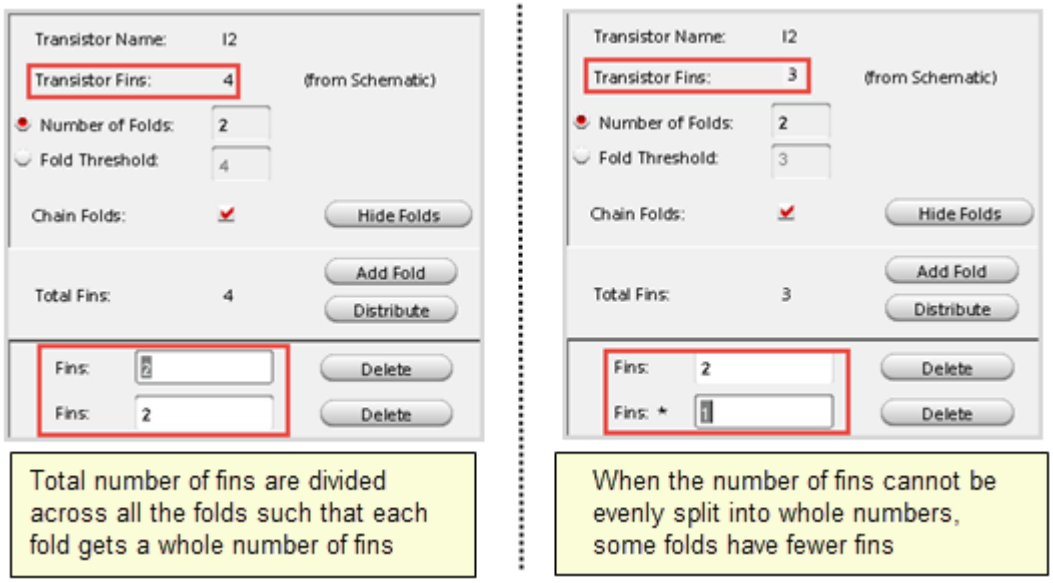
- Click *Distribute* to redistribute the available width across the folds.
- Click *Delete* to remove folds you no longer require.

If you specify multiple folds to be generated for the device, the width value displayed for each fold is controlled by the environment variable, `lxGetSignifDigits`. If the environment variable is not set, the width of each fold defaults to six significant digits.

For more information, see `lxGetSignifDigits` documentation in the *Virtuoso Layout Suite XL User Guide*.

By Specifying the Number of Folds for a FinFET Device

For FinFET devices that are driven for folding by the *Transistor Fins* value, clicking *Show Folds* divides the total number of fins across all the folds such that each fold gets a whole number of fins. If the number of fins cannot be evenly split into whole numbers, some folds have fewer fins.



© 2023 Cadence Design Systems, Inc.

The *Fins* field accepts only integer values for fins. If you type in a non-integer value, SPD automatically reverts to the last integer value.

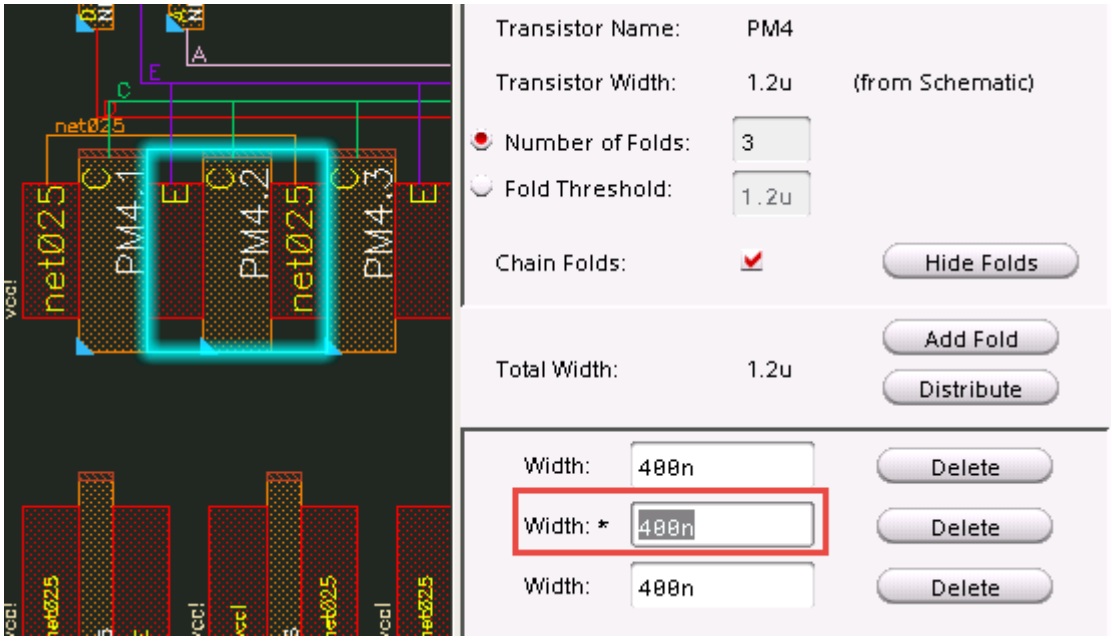
By Specifying the Fold Threshold

You can also fold a device by specifying a fold threshold, which defines the width for each fold.

For example, if the total device width is 120n, as displayed in the figure below, clicking *Show Folds* splits the width by the threshold value of 45n so that the widths of three folds are 45n, 45n and 30n.

© 2023 Cadence Design Systems, Inc.

When the selected instance is already folded, it is highlighted with a blue halo in the design. Also, the *Width* for the device is marked by an asterisk, as shown in the figure below.



If you only want to fold this device while keeping the others intact, you can add more folds by deducting the value from the selected *Width*.

Note: The mfactored devices are still folded together

Total Width of Folded Devices

As in the case of individual fold widths, the value displayed in the *Total Width* field is controlled by the `lxGetSignifDigits` environment variable.

Irrespective of the original width of the transistor, the *Total Width* field displays the sum of the width values of the various folds. For example, if the original width of the transistor is 4u and the sum of the fold widths is calculated at 3.999u, the *Total Width* field displays the width as 3.999u instead of rounding off the value to 4.

However, if the difference between the *Total Width* and the original transistor width exceeds the tolerance value set by using the `paramTolerance` environment variable, a warning message is displayed indicating that the sum of the folded widths is not equal to the original transistor width.

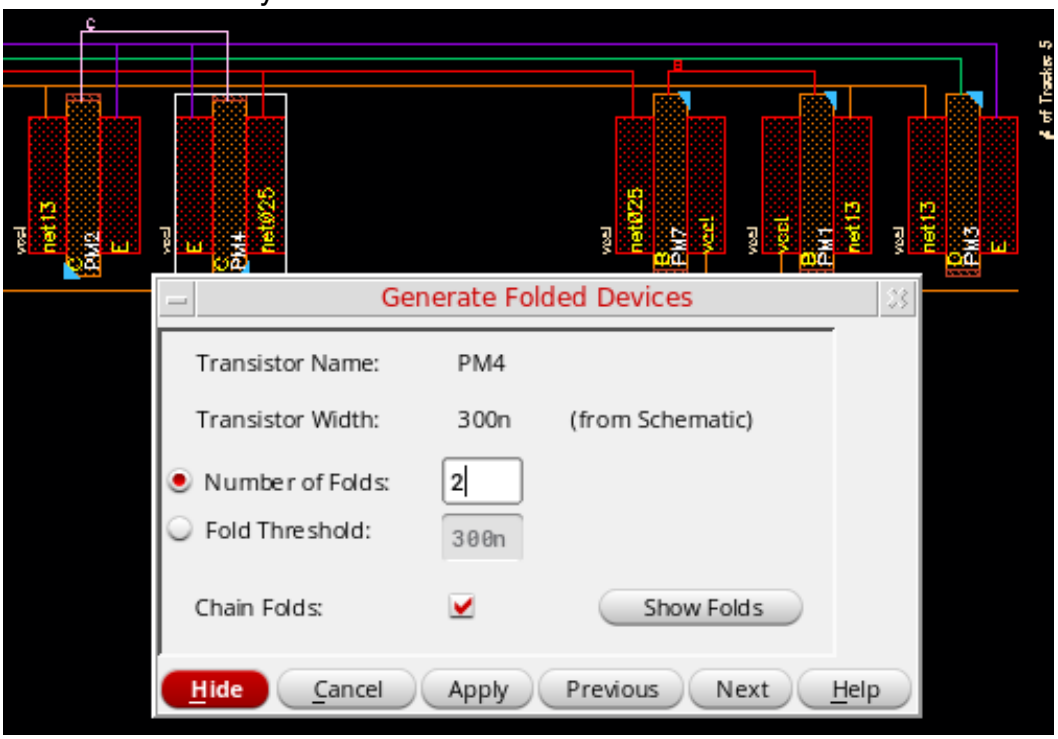
For more information, see `paramTolerance` in the *Virtuoso Layout Suite XL User Guide*.

© 2023 Cadence Design Systems, Inc.

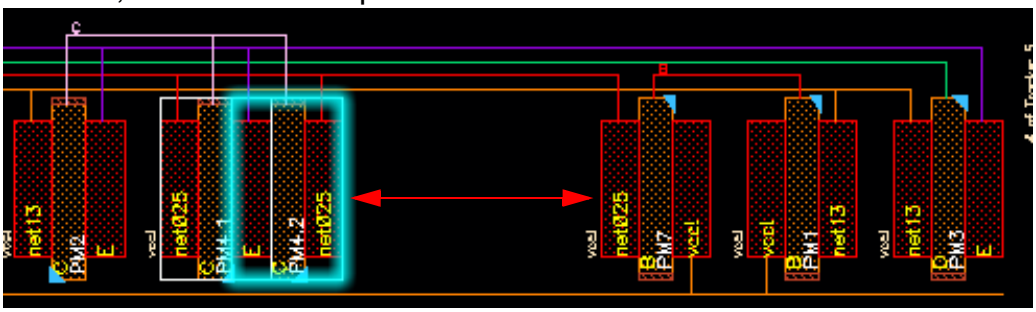
Folding Devices Based on Whether Horizontal Packing is Enabled

When the *Options – SPD – Placement – Auto pack in X direction* check box is selected, devices are packed after you fold devices. However, if this check box is deselected, automatic packing of devices in the horizontal direction does not take place after folding devices. The following steps show this difference.

➡ Select the device you want to fold.

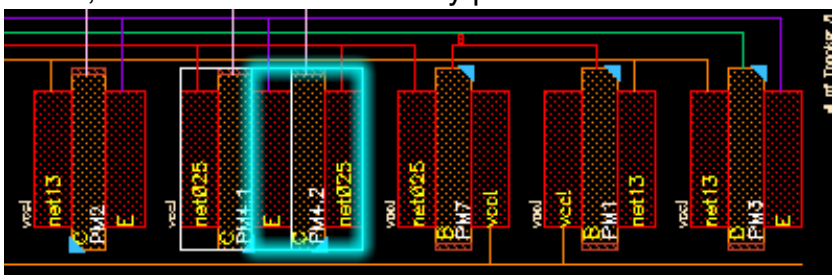


■ When the *Options – SPD – Placement – Auto pack in X direction* check box is not selected, devices are not packed after fold.



© 2023 Cadence Design Systems, Inc.

- When the *Options – SPD – Placement – Auto pack in X direction* check box is selected, devices are automatically packed after fold.



© 2023 Cadence Design Systems, Inc.

Aligning Symbolic Devices

You can align the gate of a symbolic device in one P or N type row with the gate of second device located in a different row located above or below the first row. When you align devices, the flight lines are dynamically updated and devices are moved in the two rows to align the selected devices. By default, the gate of the first device is aligned to the center of the second device.

If no alignment has been defined, the first gate of a chain is aligned with another gate when you generate the symbolic layout from schematic. To disable automatic alignment, deselect the *Auto gate alignment* check box on the Placement Options pane.

The SPD Editor provides the following alignment options:

- Aligning Devices
- Unaligning Devices
- Checking Device Alignment
- Aligning Devices Based on Whether Horizontal Packing is Enabled

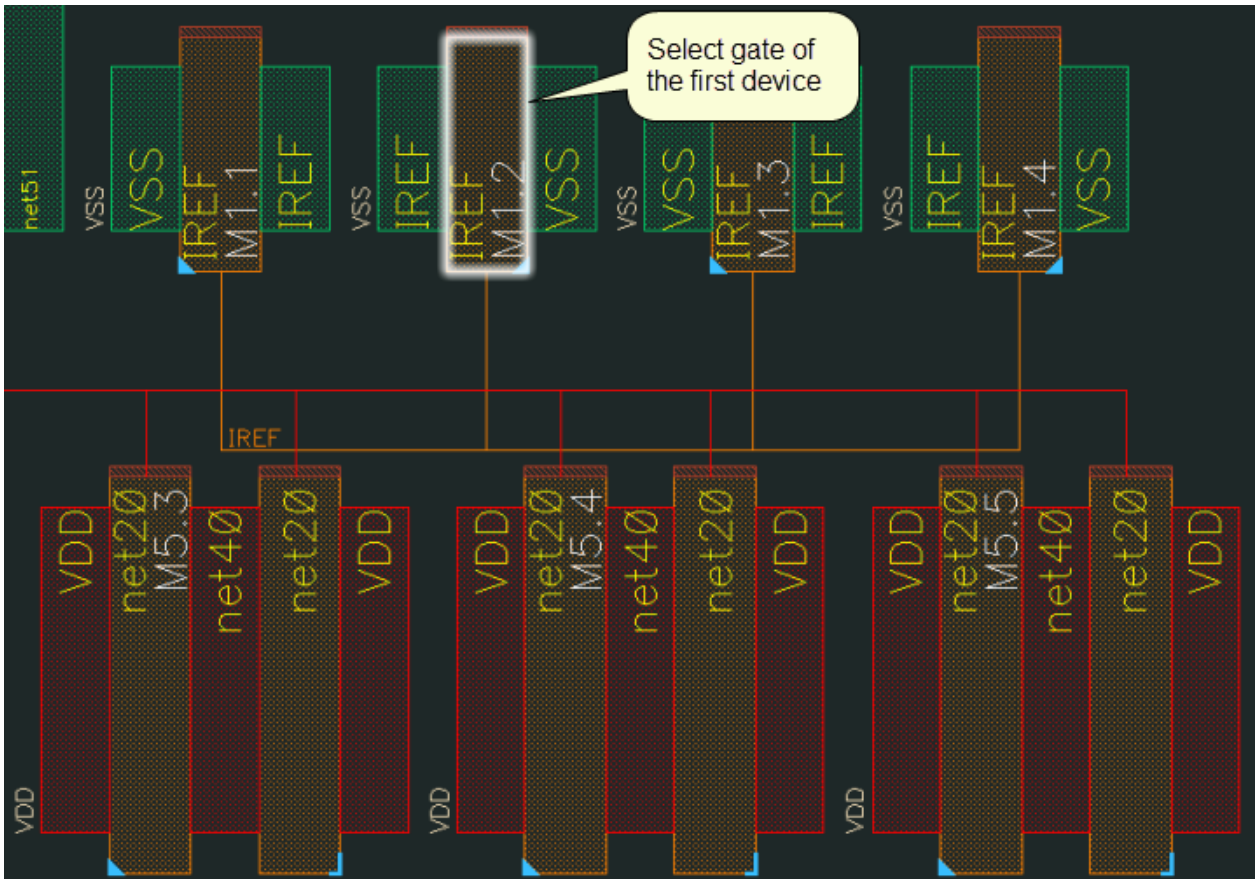
Aligning Devices

To align symbolic devices:

1. Select *Edit – Align* [a] or click the *Align* button on the SPD toolbar.

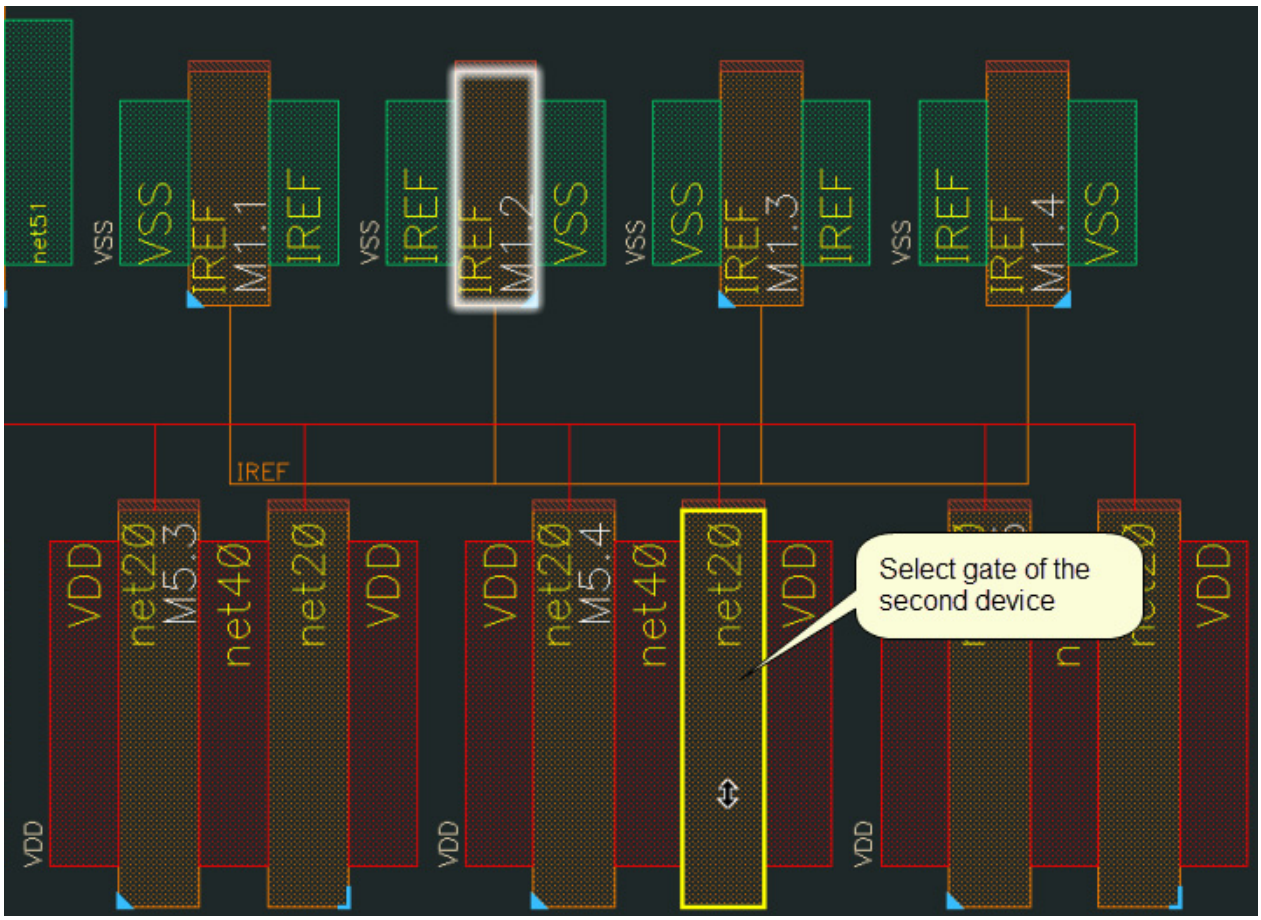
© 2023 Cadence Design Systems, Inc.

2. Select the gate of the first device.



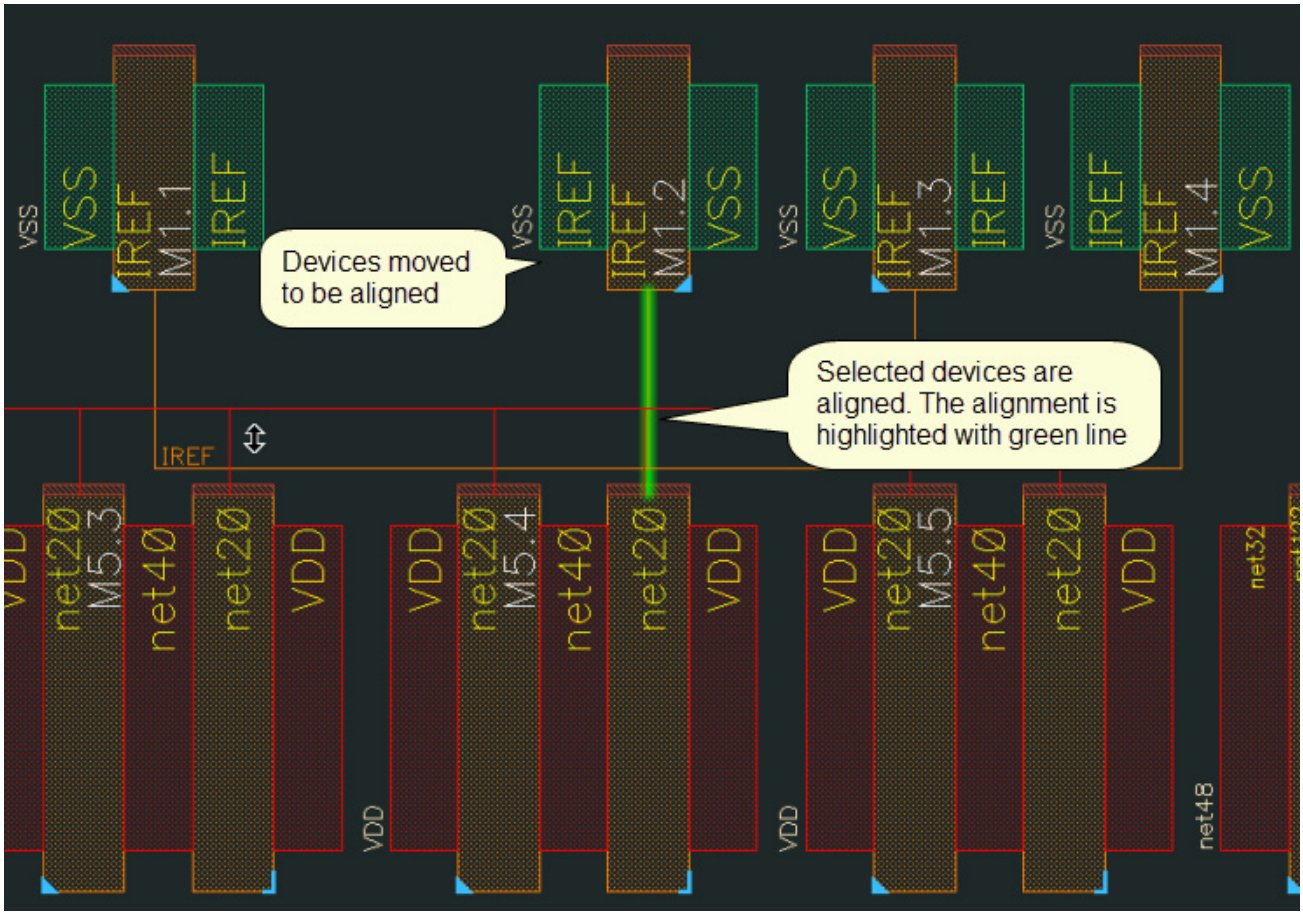
© 2023 Cadence Design Systems, Inc.

3. Select the gate of the second device located in either the row above or below.



© 2023 Cadence Design Systems, Inc.

If alignment between the selected devices is possible, they are aligned. Otherwise, an error message is displayed.



If you try to align a device across devices that are already aligned, an error message is be displayed informing you that cross alignment of devices is not possible.

Disabling Flight Lines During Alignment

By default, flight lines are displayed when you are aligning devices. To disable flight lines:

1. Select *Edit – Align* [a] or click the *Align* button on the SPD toolbar.
2. Press `F3`

   The Align form is displayed.
3. Deselect the *Show flight lines of gates* check box.

   The flight lines are removed from the canvas.

© 2023 Cadence Design Systems, Inc.

Customizing Gate Alignment

By default, the gate of the first device is aligned to the center of the second device. You can change this to align device to left or right of the second device. To do this:

1. Select *Options – SPD – Placement* [e].

   The Placement Options pane is displayed.

2. Use the *Align to Gates* drop-down list to change the alignment settings from *Center* to *Left* or *Right*.

Customizing Chain Alignment

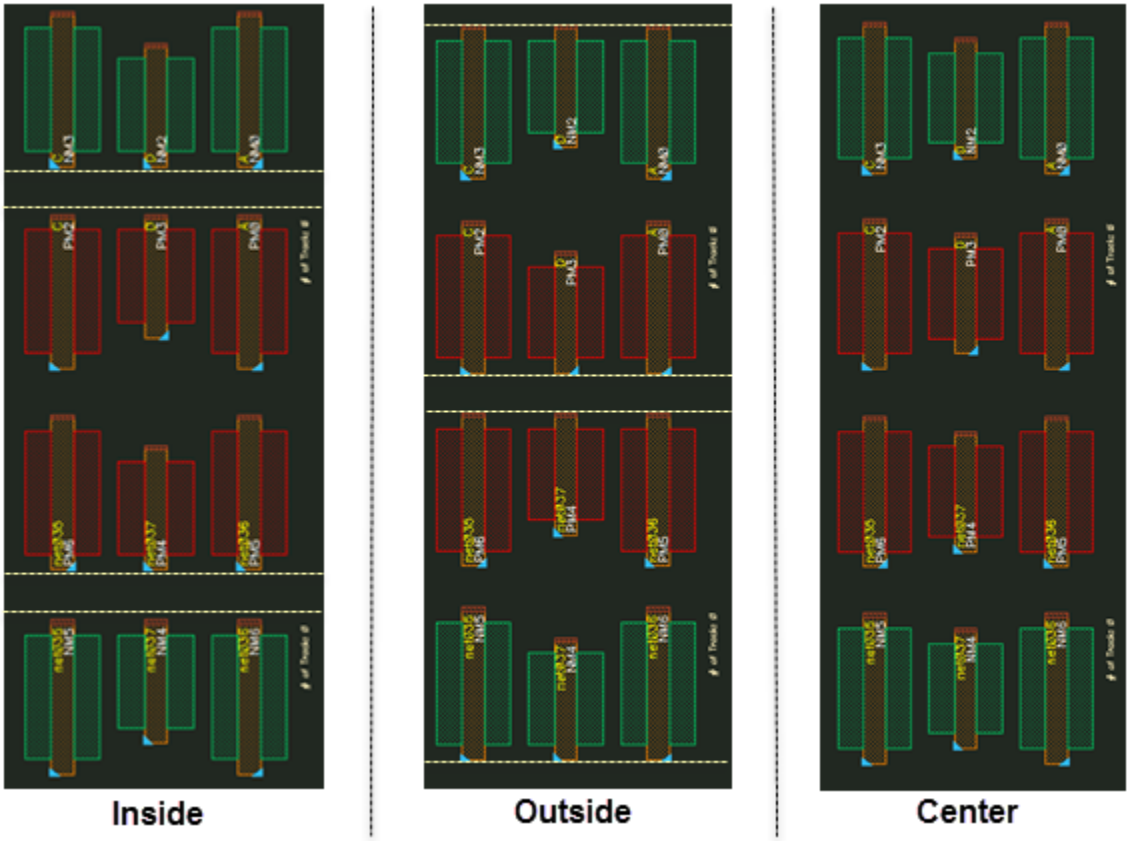
Instances in P and N rows, by default, are aligned on the inside, closer to channel area. You can change the alignment to outside or center. For this:

1. Select Options – *SPD – Placement.*

   The Placement Options pane is displayed.

© 2023 Cadence Design Systems, Inc.

2. Use the *Chain Alignment* drop-down list to change the alignment settings from *Inside* to *Outside*, or *Center*.



Unaligning Devices

To unalign symbolic devices:

1. Select *Edit – Unalign* [`Shift+a`] or click the *Align* button on the SPD toolbar.
2. Select the gate of any one device of the aligned pair.

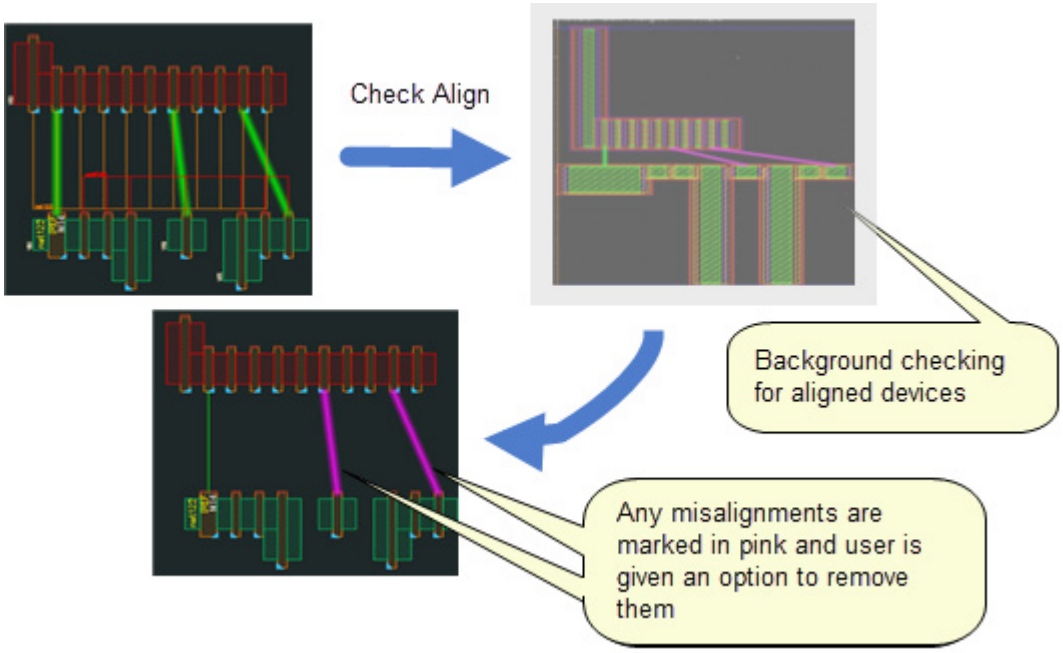
   Alignment between the devices is removed and the unaligned devices are moved back to their original locations. Flight lines are dynamically updated.

Checking Device Alignment

The *Check Align* option provides a way to identify misalignments in the symbolic design based on the real layout and remove them if needed.

© 2023 Cadence Design Systems, Inc.

To check the symbolic design for misalignments, select *Edit – Check Align* [`Ctrl+Shift+a`] or click the *Check Align* button on the SPD toolbar.

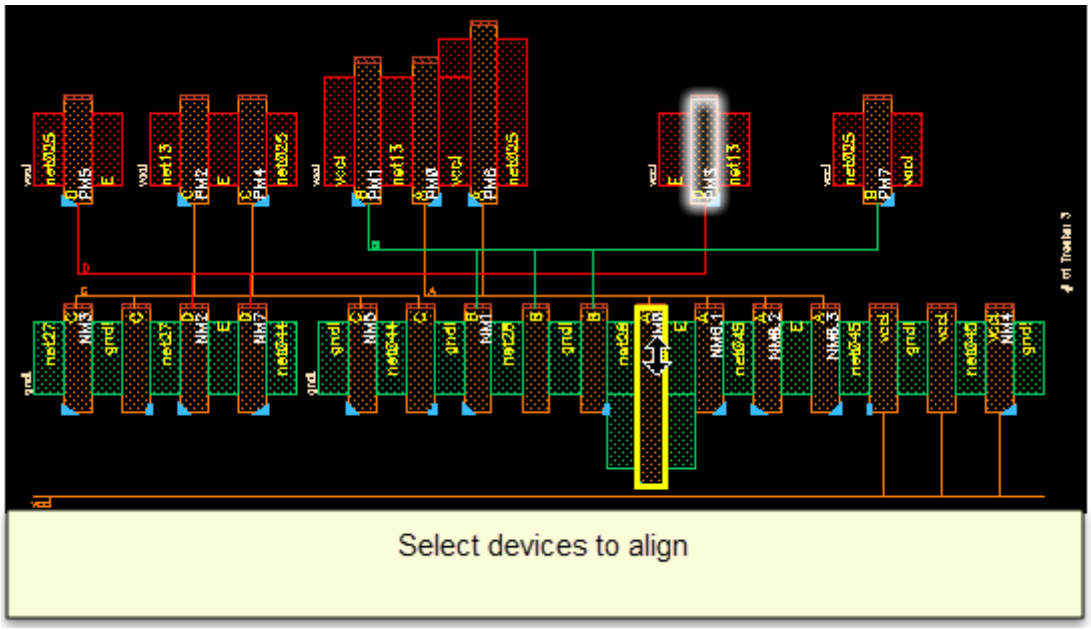


© 2023 Cadence Design Systems, Inc.

Aligning Devices Based on Whether Horizontal Packing is Enabled

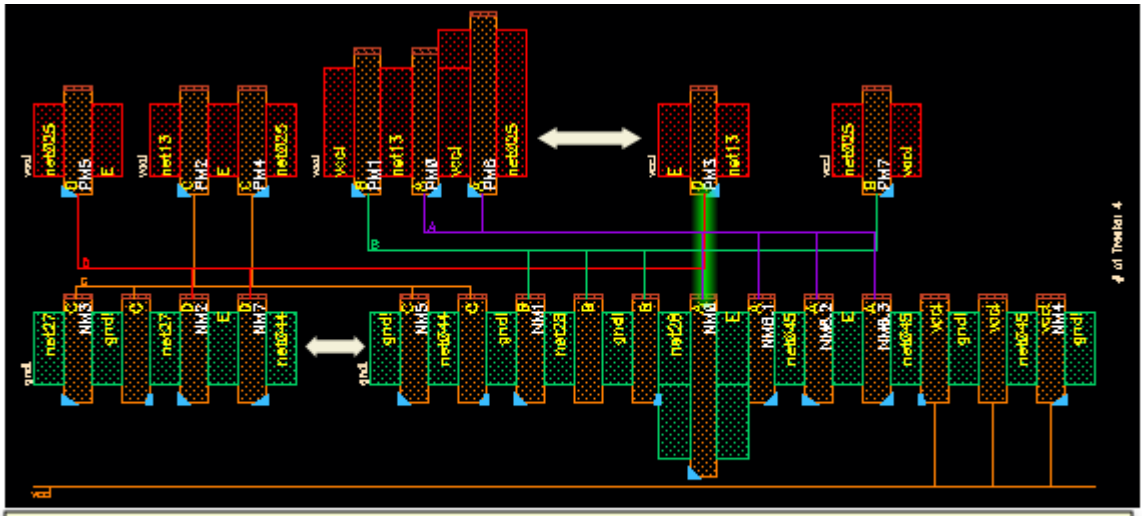
When the *Options – SPD – Placement – Auto pack in X direction* check box is selected, devices are packed after the alignment is complete. However, if this check box is deselected automatic packing of devices in the horizontal direction does not take place after alignment. The following steps show this difference.

➡ Select devices to align.



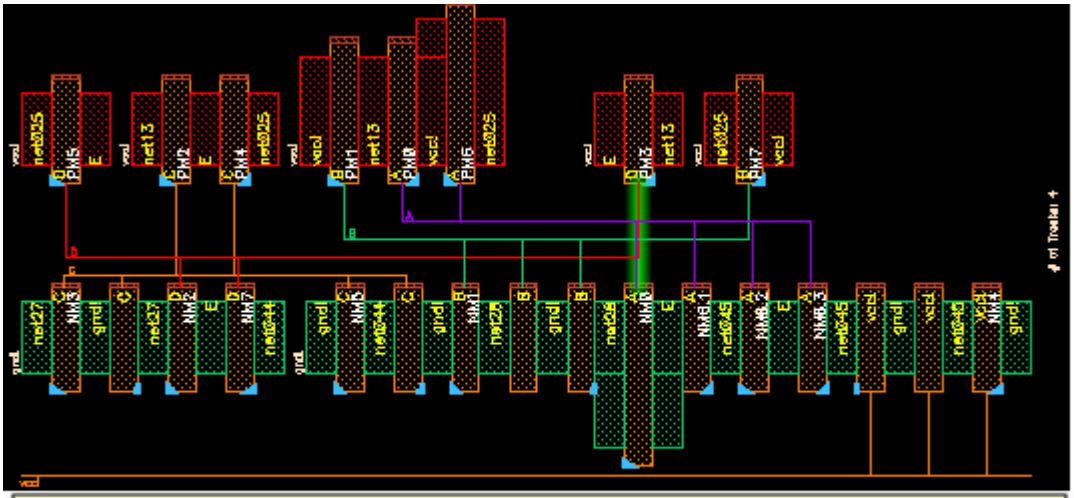
© 2023 Cadence Design Systems, Inc.

- When the *Options – SPD – Placement – Auto pack in X direction* check box is not selected, devices are not packed after alignment.



Devices are not packed after alignment when *Auto pack in X direction* option is not selected

- When the *Options – SPD – Placement – Auto pack in X direction* check box is selected, devices are automatically packed after alignment.



Devices are packed after alignment when *Auto pack in X direction* option is selected

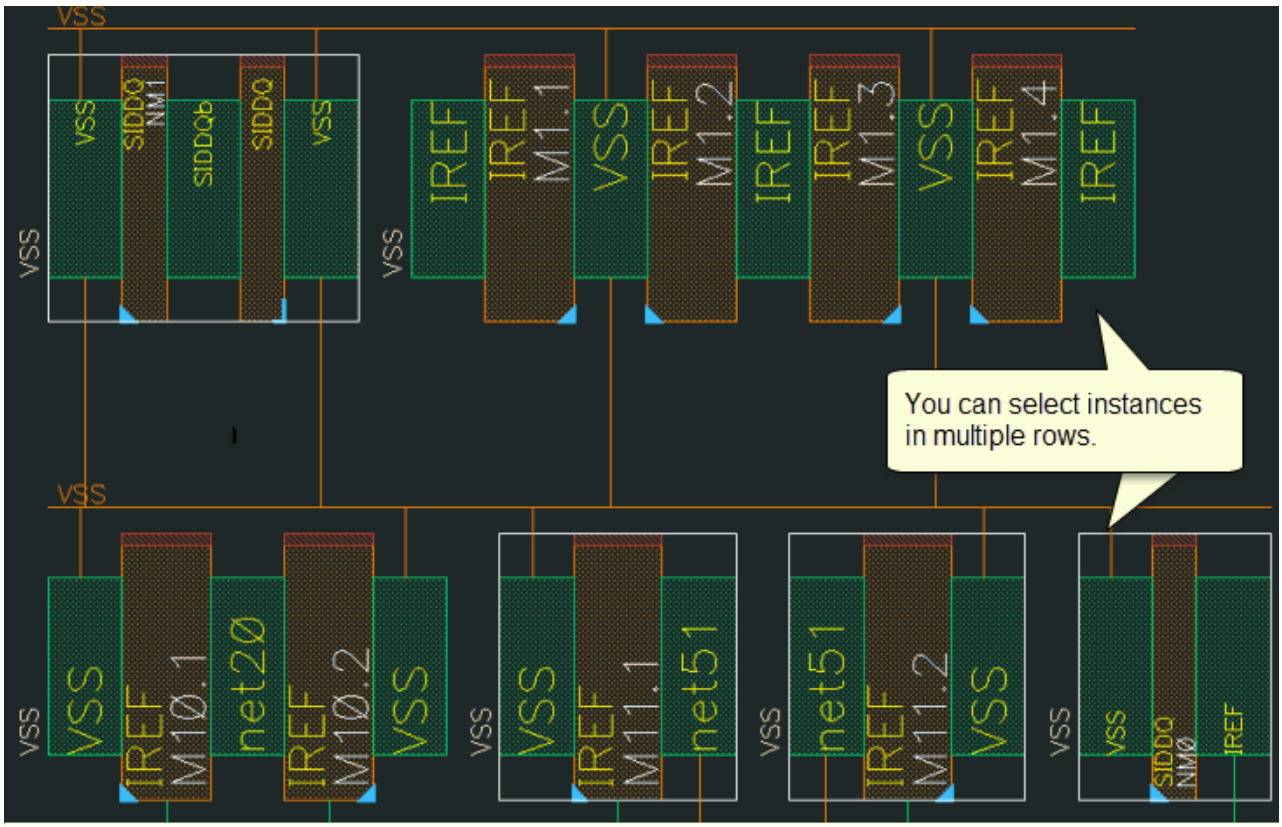
© 2023 Cadence Design Systems, Inc.

Generating Chained Devices

You can select multiple instances of same type in the symbolic design to create a device chain. The selected instances are chained and placed at the I/O pin-connected instance. If there are more than one I/O pin-connected instances, the Generate Chained Devices form is displayed to let you select the instance to place the device chain.

To chain symbolic devices:

1. Select the devices to add to be chained.



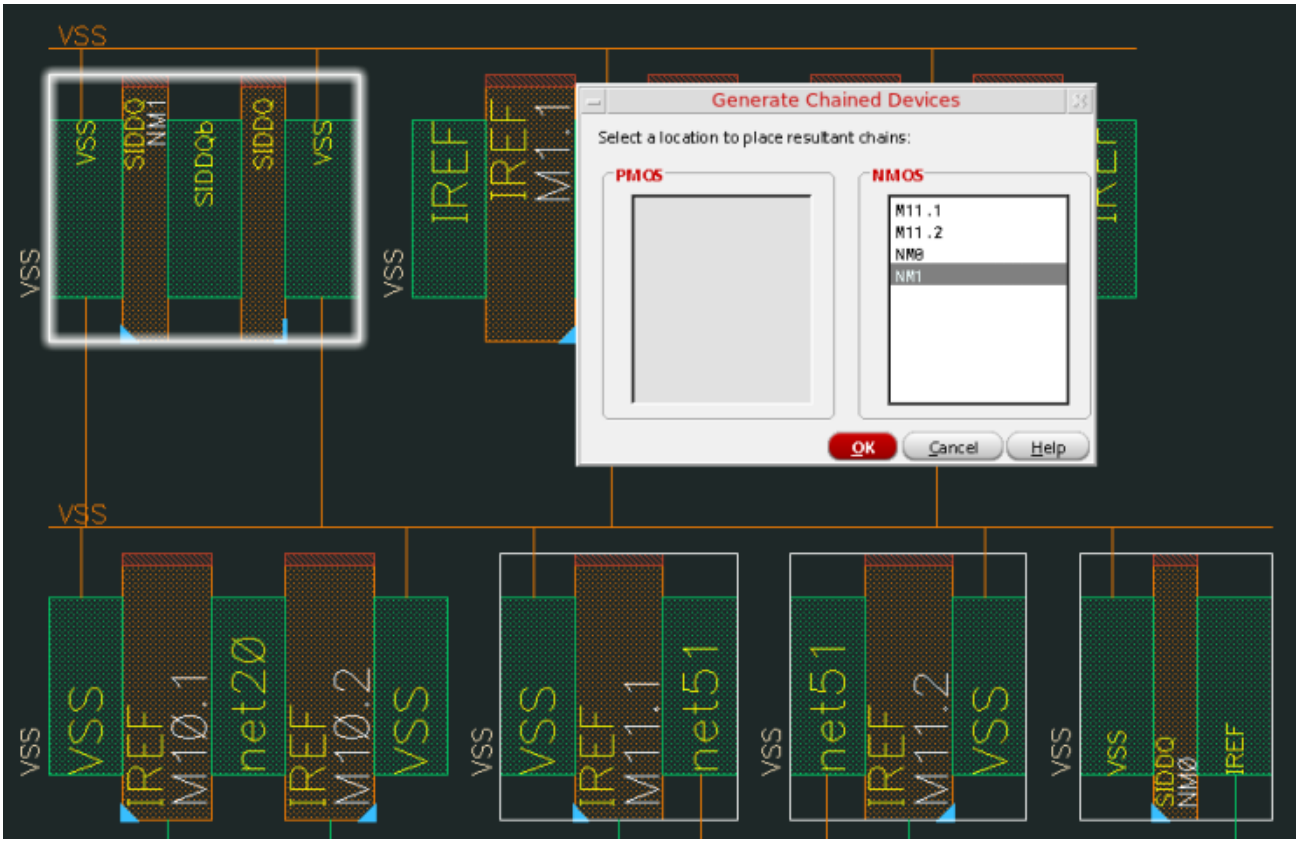
Select instances to add to the chain

2. Select *Edit – Generate Chained Devices* [C] or click the *Generate Chained Devices* button on the SPD toolbar.

   The Generate Chained Devices form is displayed.

© 2023 Cadence Design Systems, Inc.

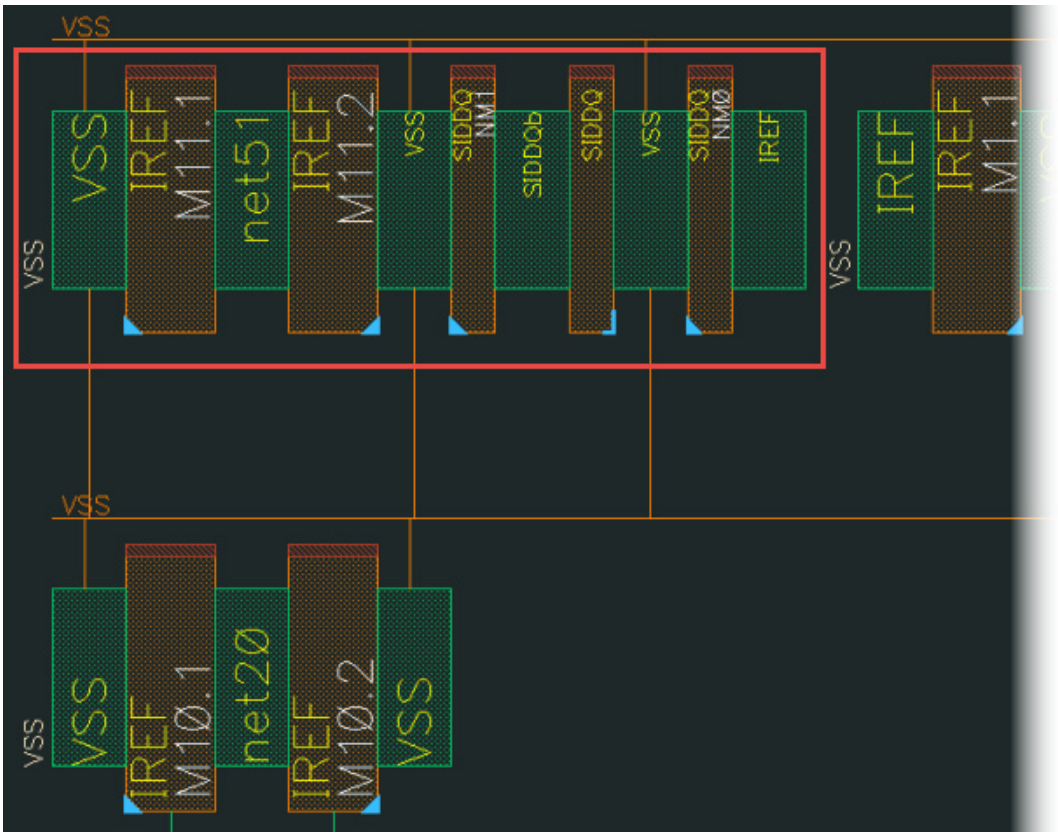
3. Select the instance where you want to insert the resultant chain.



4. Click *OK*.

© 2023 Cadence Design Systems, Inc.

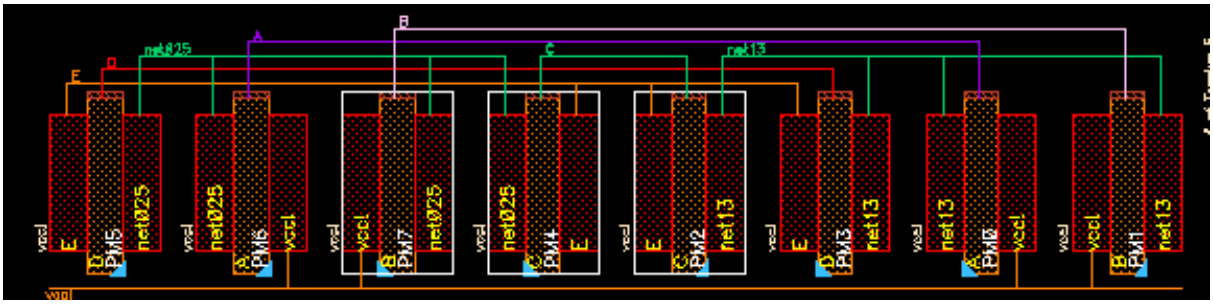
A device chain is created and inserted at the specified location.



Generating Chained Devices Based on Whether Horizontal Packing is Enabled

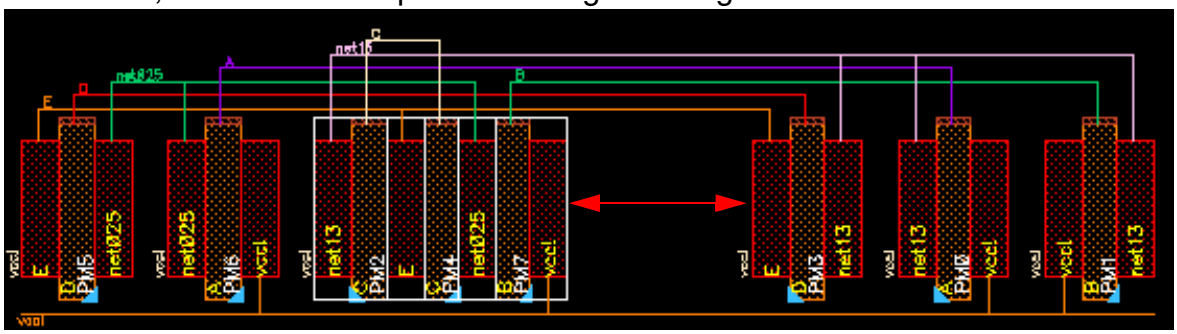
When the *Options – SPD – Placement – Auto pack in X direction* check box is selected, devices are packed after you generate chained devices. However, if this check box is deselected, automatic packing of devices in the horizontal direction does not take place after generating chained devices. The following steps show this difference.

➡ Select the devices to generate chained devices.

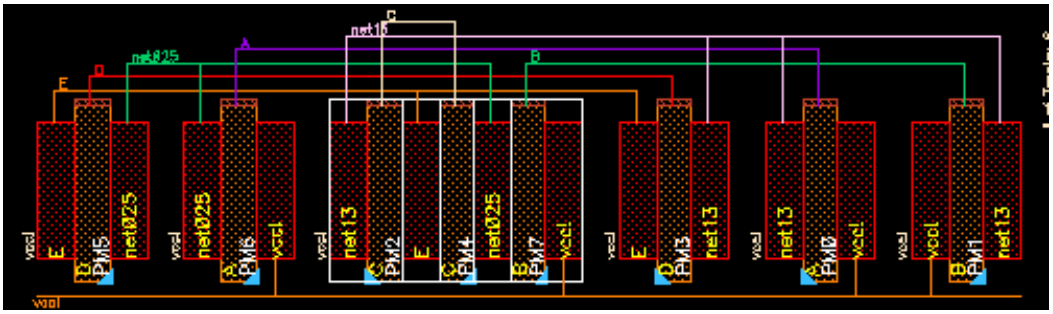


© 2023 Cadence Design Systems, Inc.

- When the *Options – SPD – Placement – Auto pack in X direction* check box is not selected, devices are not packed after generating chained devices.



- When the *Options – SPD – Placement – Auto pack in X direction* check box is selected, devices are automatically packed after generating chained devices.



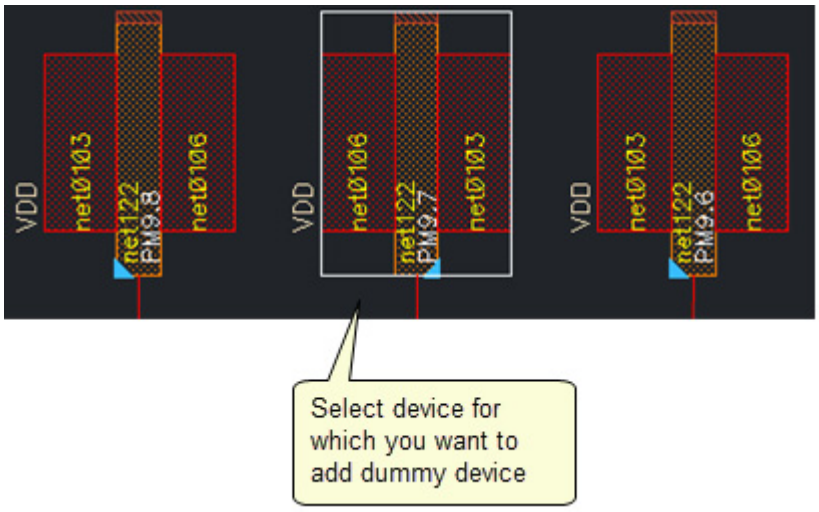
© 2023 Cadence Design Systems, Inc.

Adding Dummy Devices

Some electrical configurations might require you to add dummy devices in the design. You can add dummy devices to the left, right, or on both sides of the selected device.

To add a dummy:

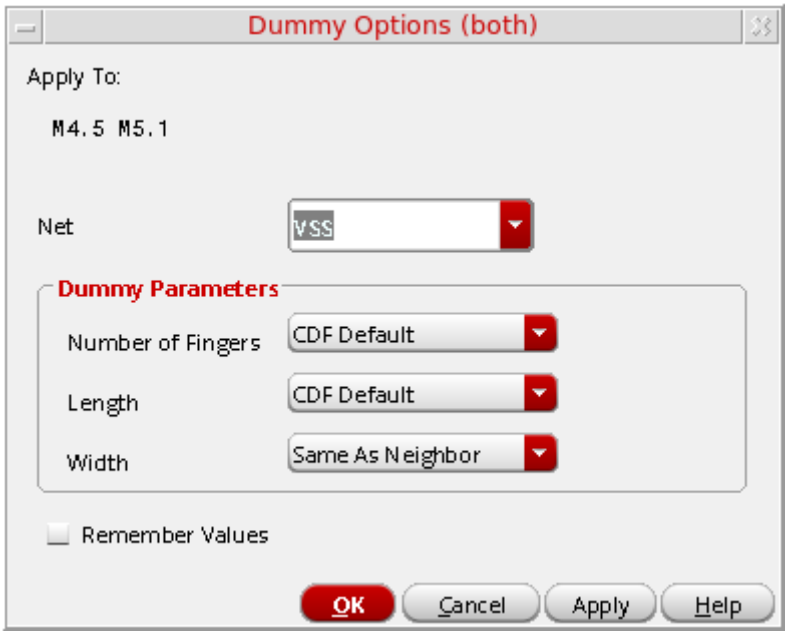
1. Select a device for which you want to add the dummy.



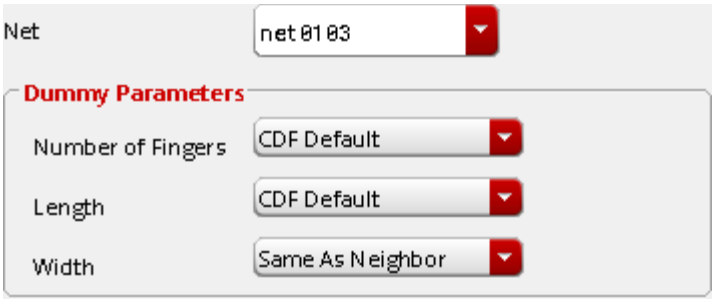
2. Select *Edit – Add Dummy – Left, Right or Both*, or select the respective option from the *Add Dummy* drop-down button on the SPD toolbar.

© 2023 Cadence Design Systems, Inc.

The Dummy Options form is displayed.

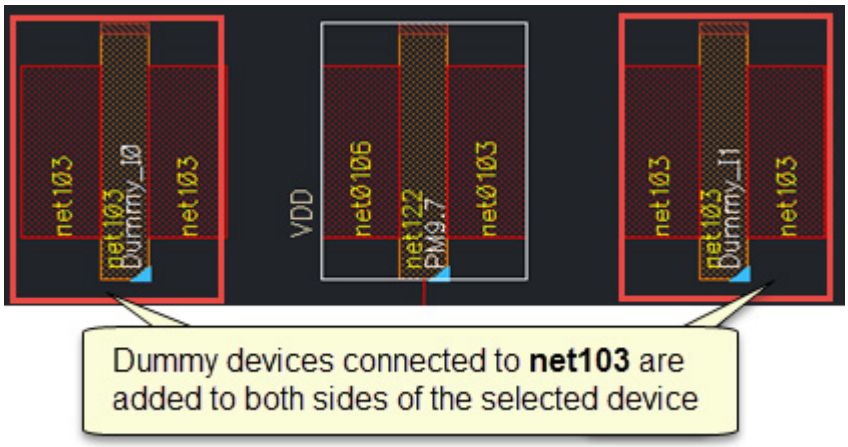


3. Specify the *Net* to which the dummy device will be connected.
4. Specify the following parameters and click *OK*:
   - *Number of Fingers*
   - *Length*
   - *Width*



© 2023 Cadence Design Systems, Inc.

Dummy devices are added to the design.



Abutment for Dummy Devices

The inserted dummy device is abutted to its neighbor, if the neighboring device is already abutted.

The inserted dummy device is not abutted:

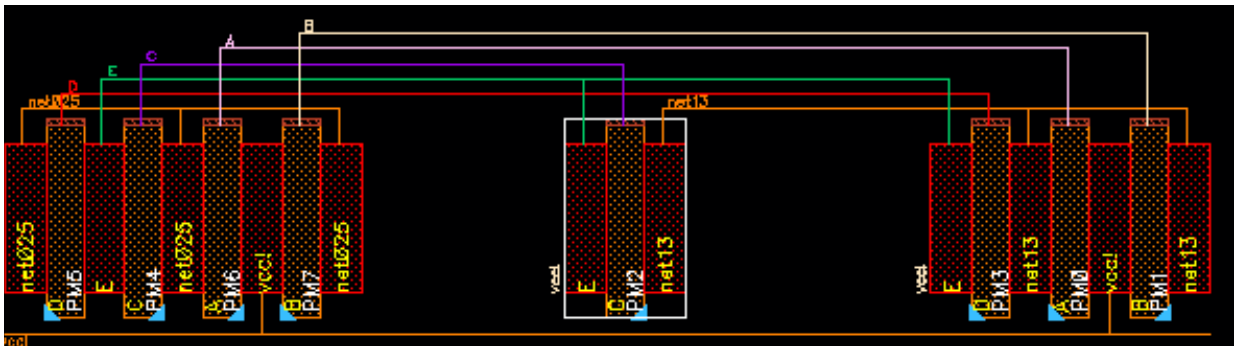
- If the neighboring device is not abutted.
- If the neighboring device has user abutment.

Adding Dummy Devices Based on Whether Horizontal Packing is Enabled

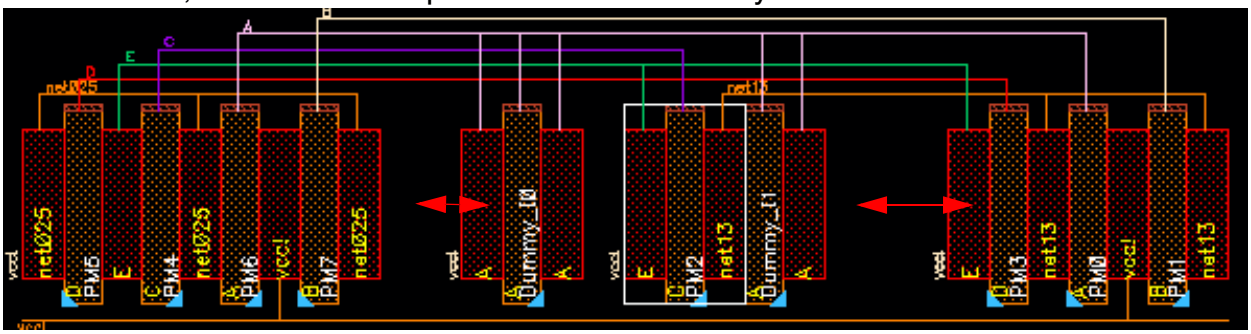
When the *Options – SPD – Placement – Auto pack in X direction* check box is selected, devices are packed after you add dummy devices. However, if this check box is deselected, automatic packing of devices in the horizontal direction does not take place after adding the dummy devices. The following steps show this difference.

© 2023 Cadence Design Systems, Inc.

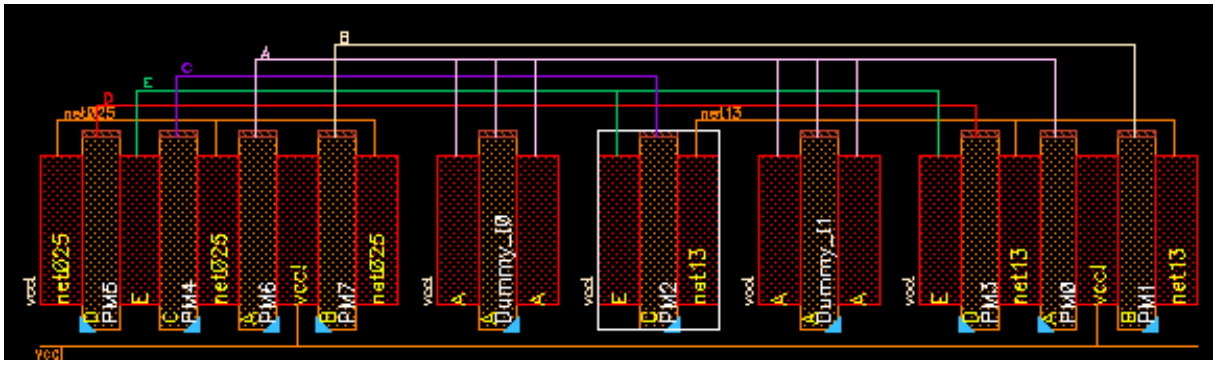
➡ Select the device for which you want to add the dummy devices.



■ When the *Options – SPD – Placement – Auto pack in X direction* check box is not selected, devices are not packed after the dummy devices are added.



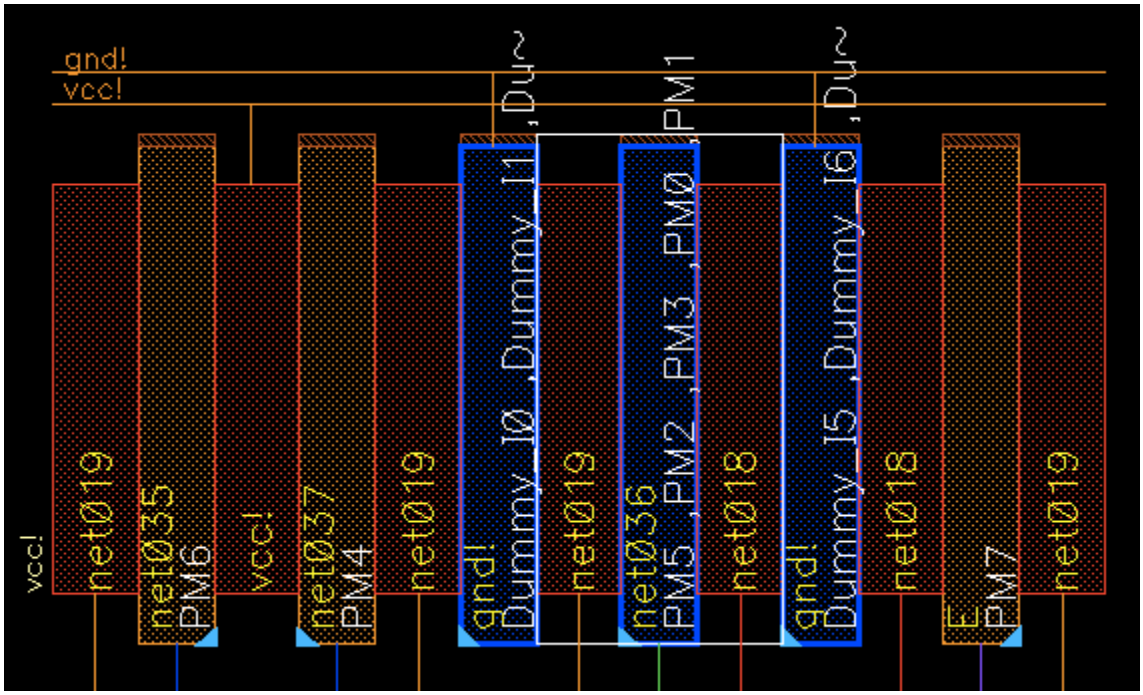
■ When the *Options – SPD – Placement – Auto pack in X direction* check box is selected, devices are automatically packed after the dummy devices are added.



© 2023 Cadence Design Systems, Inc.

Adding a Dummy Device to a Stacked Device

When a dummy device is added to a stacked device, a stacked dummy device is added on the specified side. The dummy device has same number instances as the stacked device.



© 2023 Cadence Design Systems, Inc.

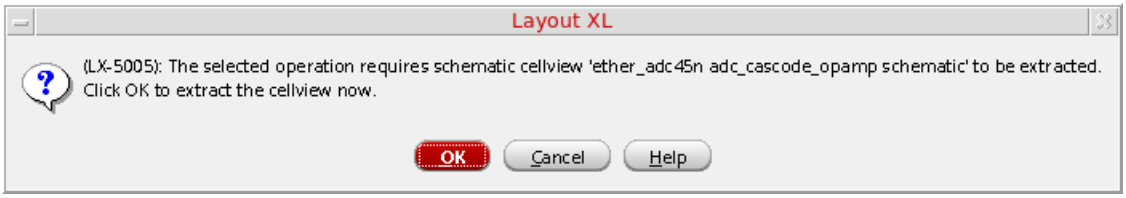
Backannotating Symbolic Devices

Backannotation lets you add the newly added dummy devices to schematic, the dummy devices may be renamed if there is a conflict in the name in schematic or layout.

To backannotate all dummy devices to schematic:

1. Select *Edit – Back Annotate – All Active Dummy Instances* [n] or click the *Back Annotate* button on the SPD toolbar.

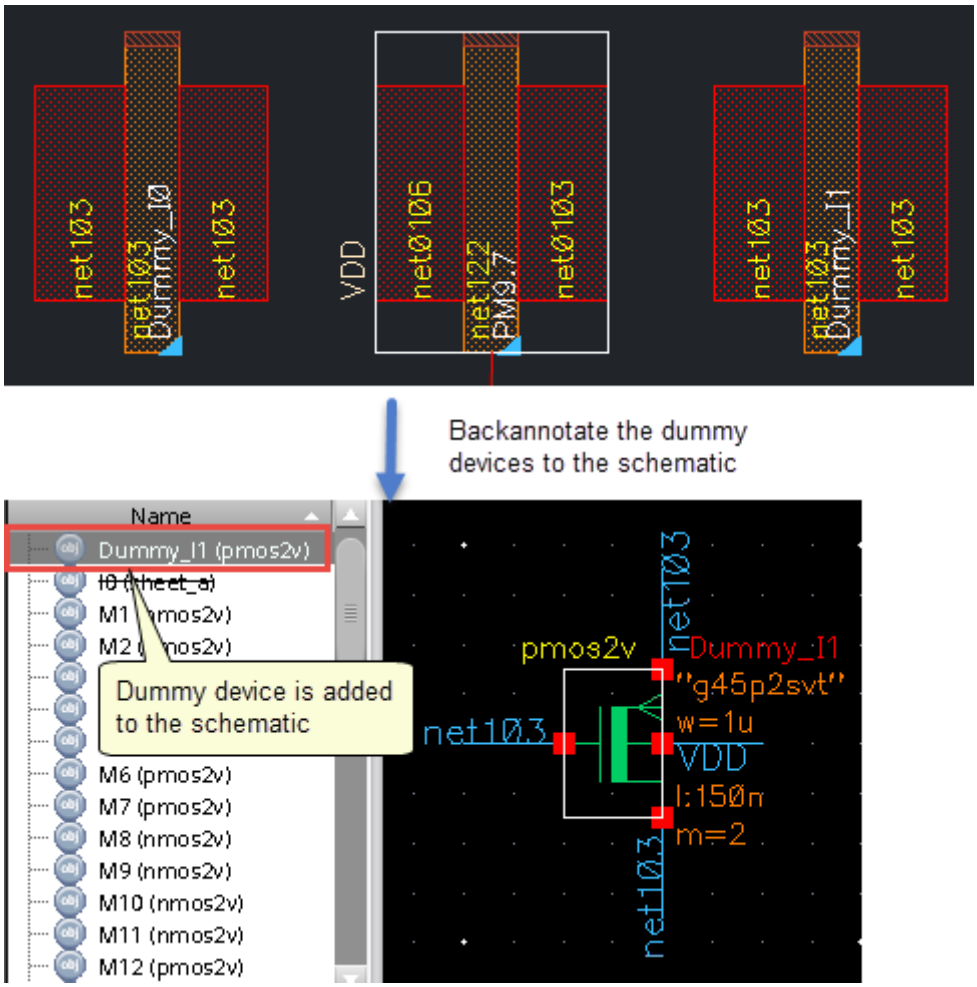
   A warning message informing you that the corresponding schematic cellview needs to be extracted is displayed.



2. Click *OK* to allow the schematic cellview to be extracted.

© 2023 Cadence Design Systems, Inc.

3. All dummy devices in the symbolic design are backannotated to the schematic.



To backannotate selected dummy devices:

1. Select the dummy devices to backannotate in the canvas.
2. Select *Edit – Back Annotate – Selected Dummy Instances* [`Shift+n`].
3. Click *OK* to allow the schematic cellview to be extracted.
4. The selected dummy devices in the symbolic design are backannotated to the schematic.

© 2023 Cadence Design Systems, Inc.

4

Optimizing the Placement and Generating the Layout

After you have created your symbolic placement, you might need to perform some final optimization tasks, such as removing any empty spaces created during editing, adding more devices from the source schematic, or resetting the placement of certain devices. You might also want to preview the edited design before going on to generate a final layout.

You can perform all these tasks using the options in the *Place* menu in the SPD Editing window. The chapter covers the following topics:

- Appending Components from the Schematic
- Appending Partial Components from Layout
- Removing Symbolic Devices
- Packing the Design
- Resetting Placement
- Displaying Flight Lines
- Reordering and Resetting Trunks
- Probing Selected Nets
- Checking Source or Updating Layout
- Displaying Free Bucket Size of Devices
- Previewing the Design
  - View Abstract Preview
  - Create Rulers
  - Automatic Shifting of Additional Tracks
  - Move and Pack Tracks

© 2023 Cadence Design Systems, Inc.

- Preview Based on Whether Horizontal Packing of Devices is Enabled in SPD Editor

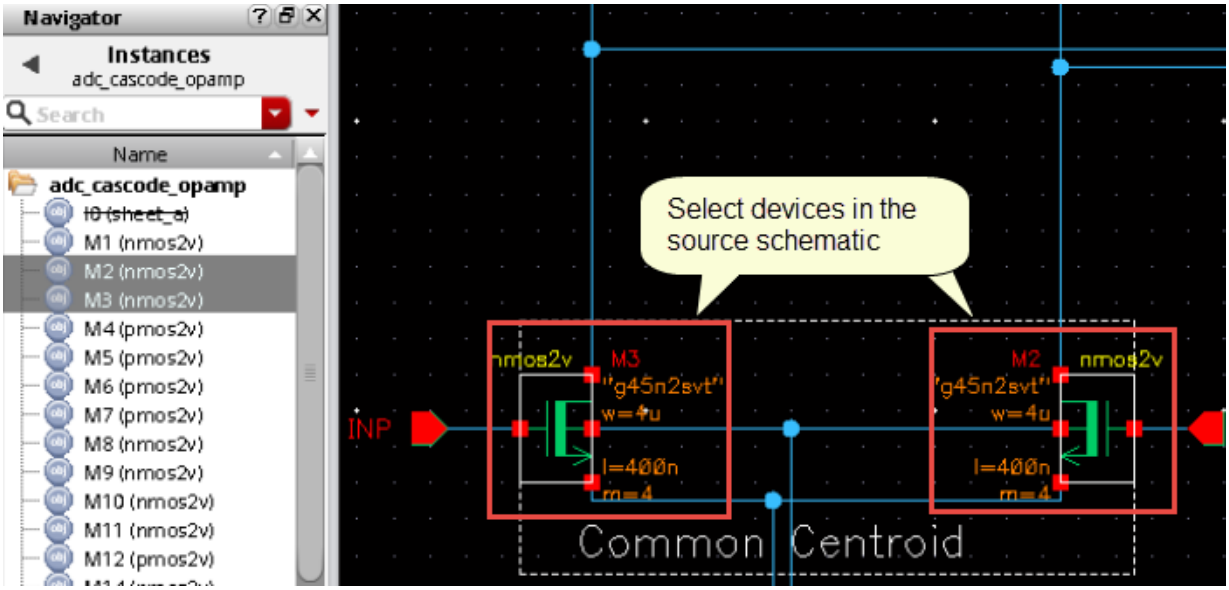
■ Generating the Layout

Appending Components from the Schematic

The *Append Selected From Source* command makes it possible to add new devices to the SPD design from the source schematic after you have generated a symbolic layout.

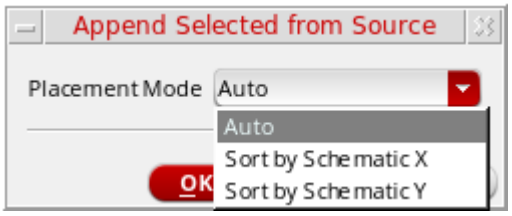
To append selected devices from the schematic:

1. Select the schematic instances you want to append.



2. Choose *Place – Append Selected From Source* or click the *Append Selected from Source* button on the SPD toolbar.

   You can press `F3` to display the Append Selected from Source form, where you can specify placement mode for adding devices to the SPD design.

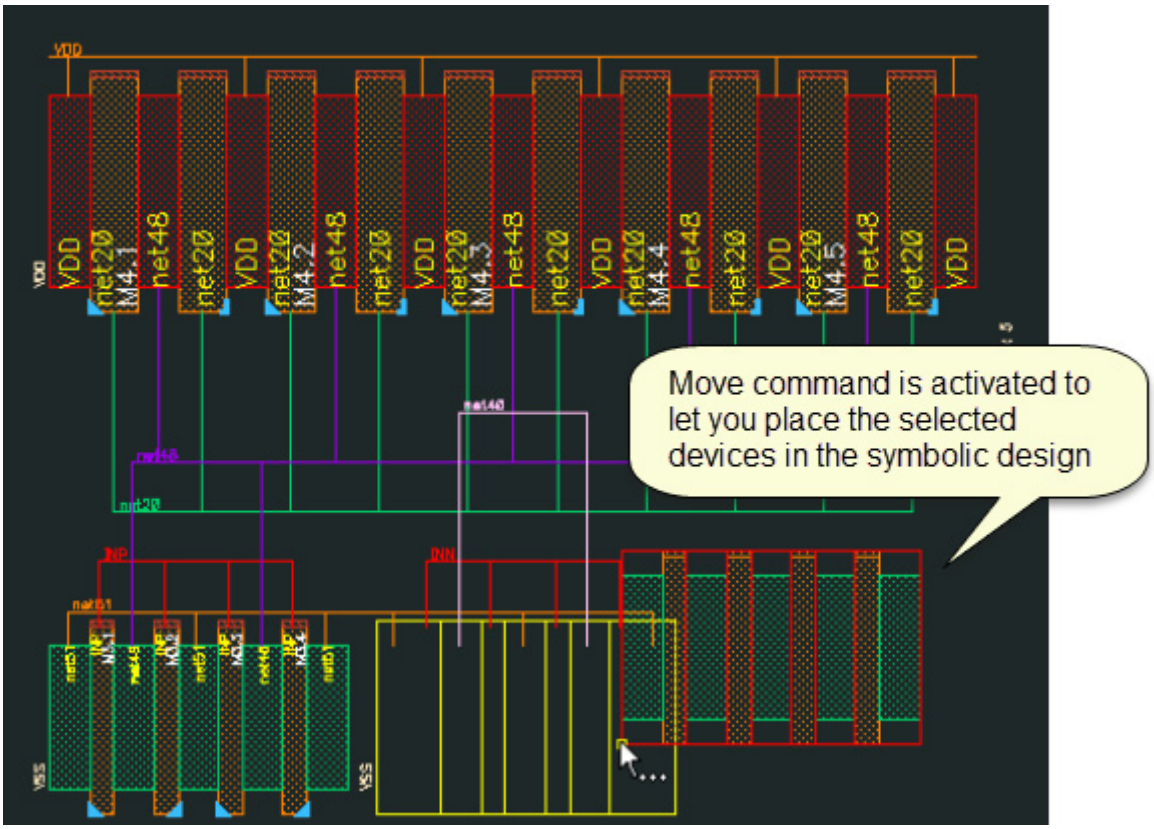


Note: If you import a device chain into SPD, then SPD will attempt to chain the devices with a common master after append. To disable automatic chaining of devices after

© 2023 Cadence Design Systems, Inc.

append, clear the *SPD Options - Placement - Chaining/Folding - Chain* check box. Next, append the selected devices and then use the *Sort by Schematic X* option while appending devices to get the correct order of devices in SPD.

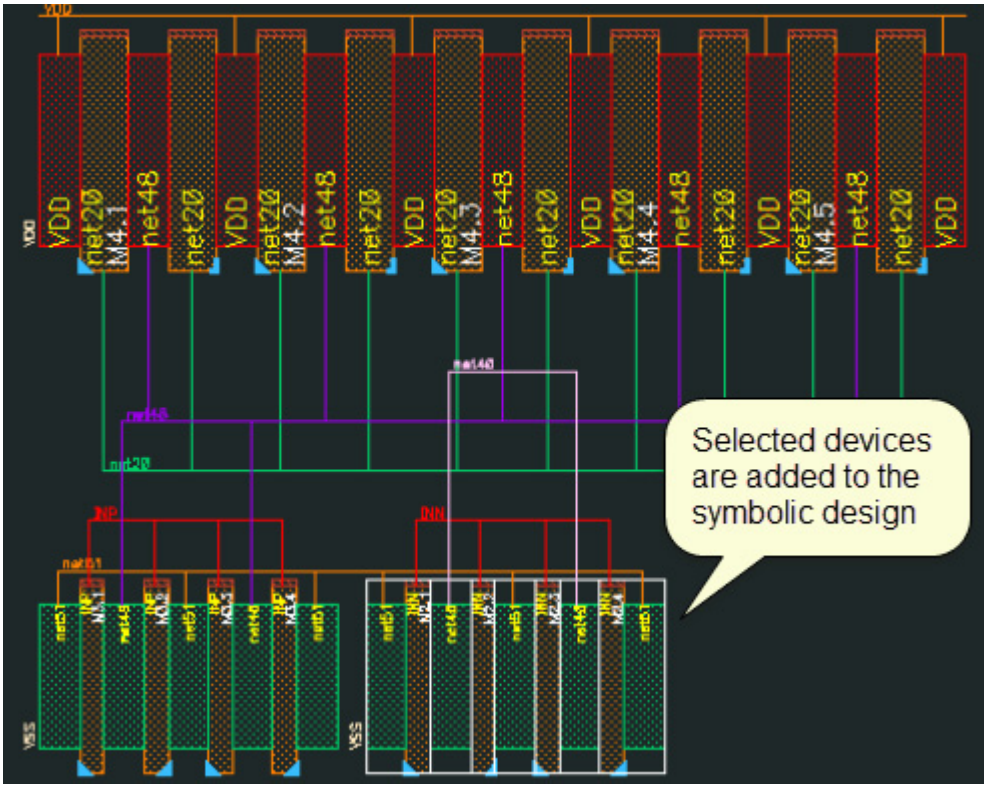
By default, SPD automatically activates the *Move* command to let you add the selected devices at the desired location in the symbolic design.



To disable the move after append feature, set the spdMoveAfterAppend environment variable to `nil`. If you disable move after append, instances are appended to the tails of P type and N type rows by default.

© 2023 Cadence Design Systems, Inc.

3. Move and append the P-type and N-type devices to respective rows. You can press `Esc` to append the selected devices to the end of rows.

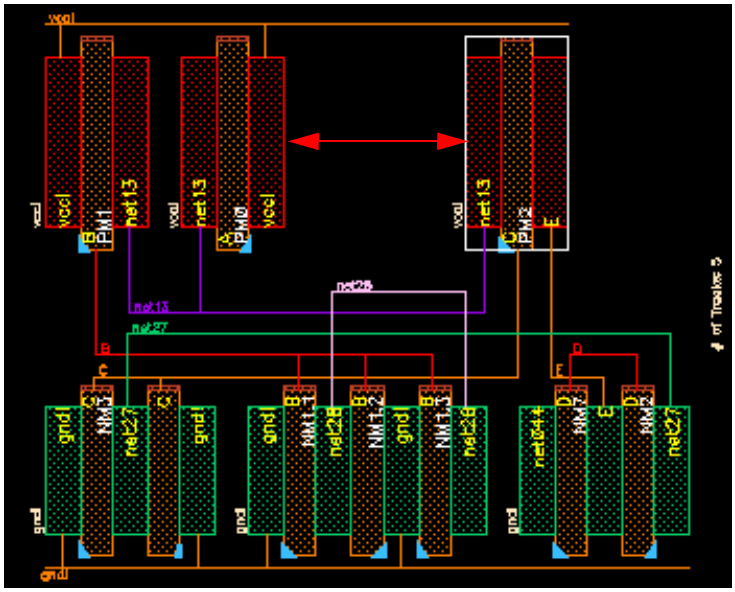


Appending Devices From Source Based on Whether Horizontal Packing is Enabled

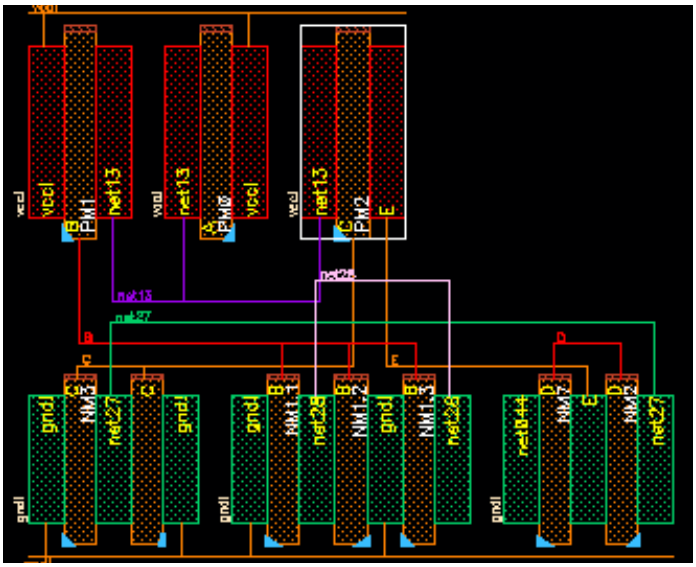
When the *Options – SPD – Placement – Auto pack in X direction* check box is selected, devices are packed after you append devices from source. However, if this check box is deselected. automatic packing of devices in the horizontal direction does not take place after appending from source.

© 2023 Cadence Design Systems, Inc.

- When the *Options – SPD – Placement – Auto pack in X direction* check box is not selected, devices are not packed after appending from source.



- When the *Options – SPD – Placement – Auto pack in X direction* check box is selected, devices are automatically packed after appending from source.



© 2023 Cadence Design Systems, Inc.

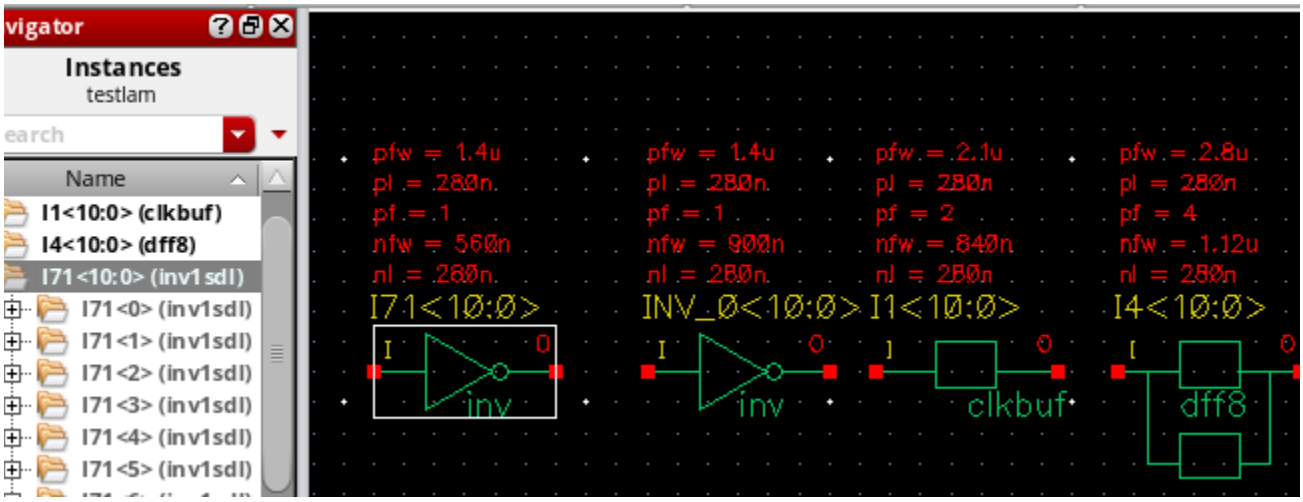
Appending Partial Components from Layout

SPD allows you to partially select hierarchal or hierarchal and vector components in Layout XL and generate an SPD layout for further editing.

The following example shows how you can achieve this.

1. Open your design in schematic.

   The following schematic instance is a vector and hierarchal instance. You can select one or more instances in the schematic and then open them in Layout XL. In this example, we select one instance.



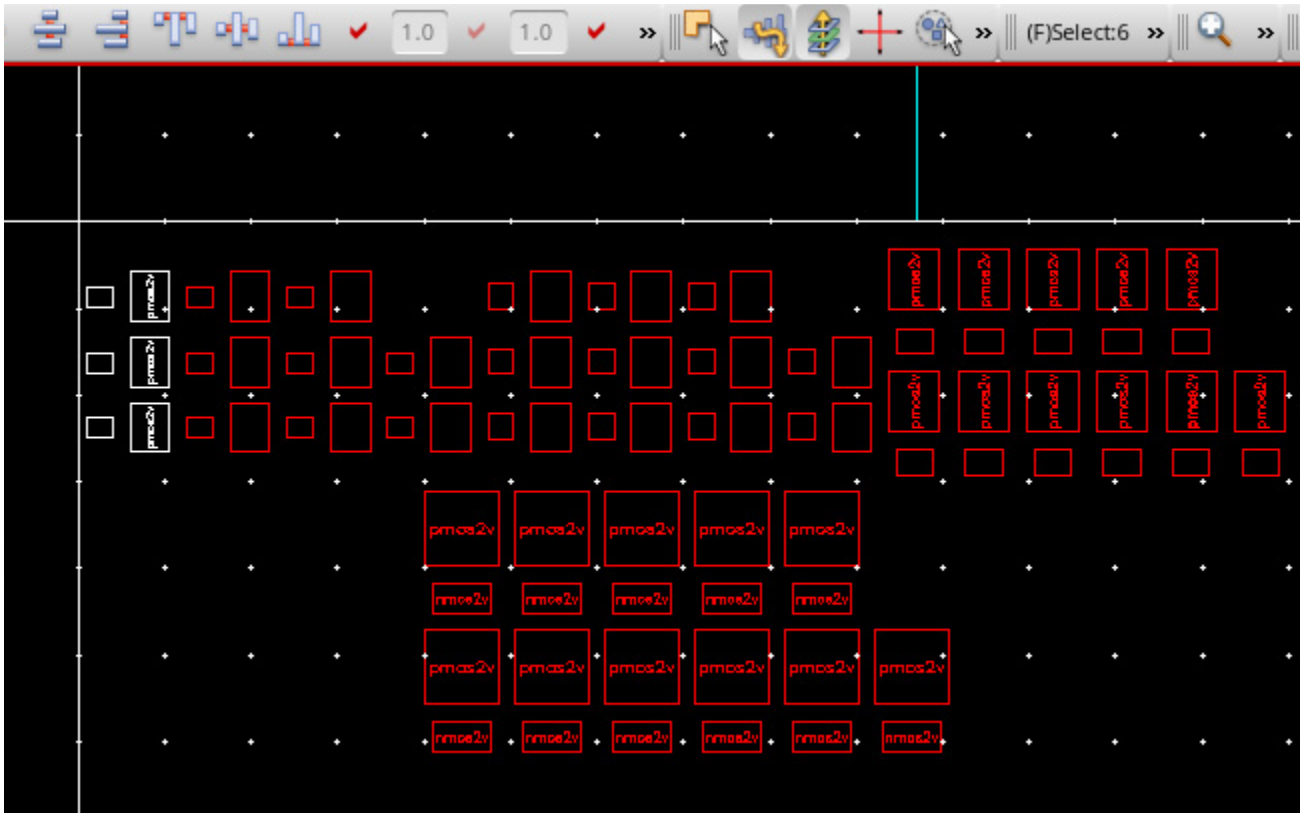
2. Open Layout XL.
3. In Layout XL, choose *Connectivity – Generate – All From Source.*

   The selected instance opens in the Layout XL canvas.

   Note: If you partially select a schematic instance that is folded or split into several layout instances, then all instances are brought into and opened in SPD.

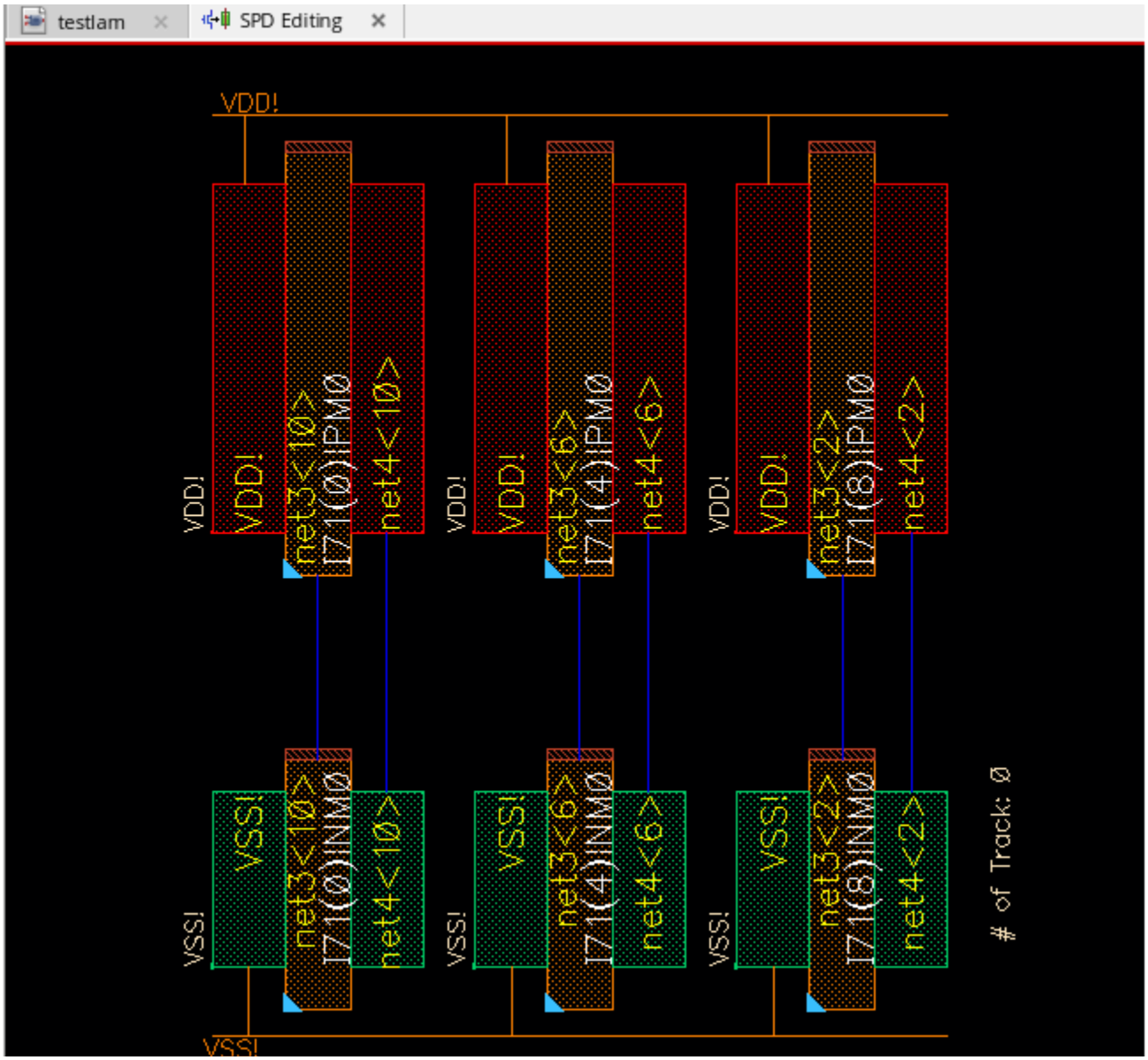
© 2023 Cadence Design Systems, Inc.

4. Select some devices in Layout XL and click the *Generate Symbolic Placement* button.



© 2023 Cadence Design Systems, Inc.

SPD layout is generated with the instances selected in Layout XL.



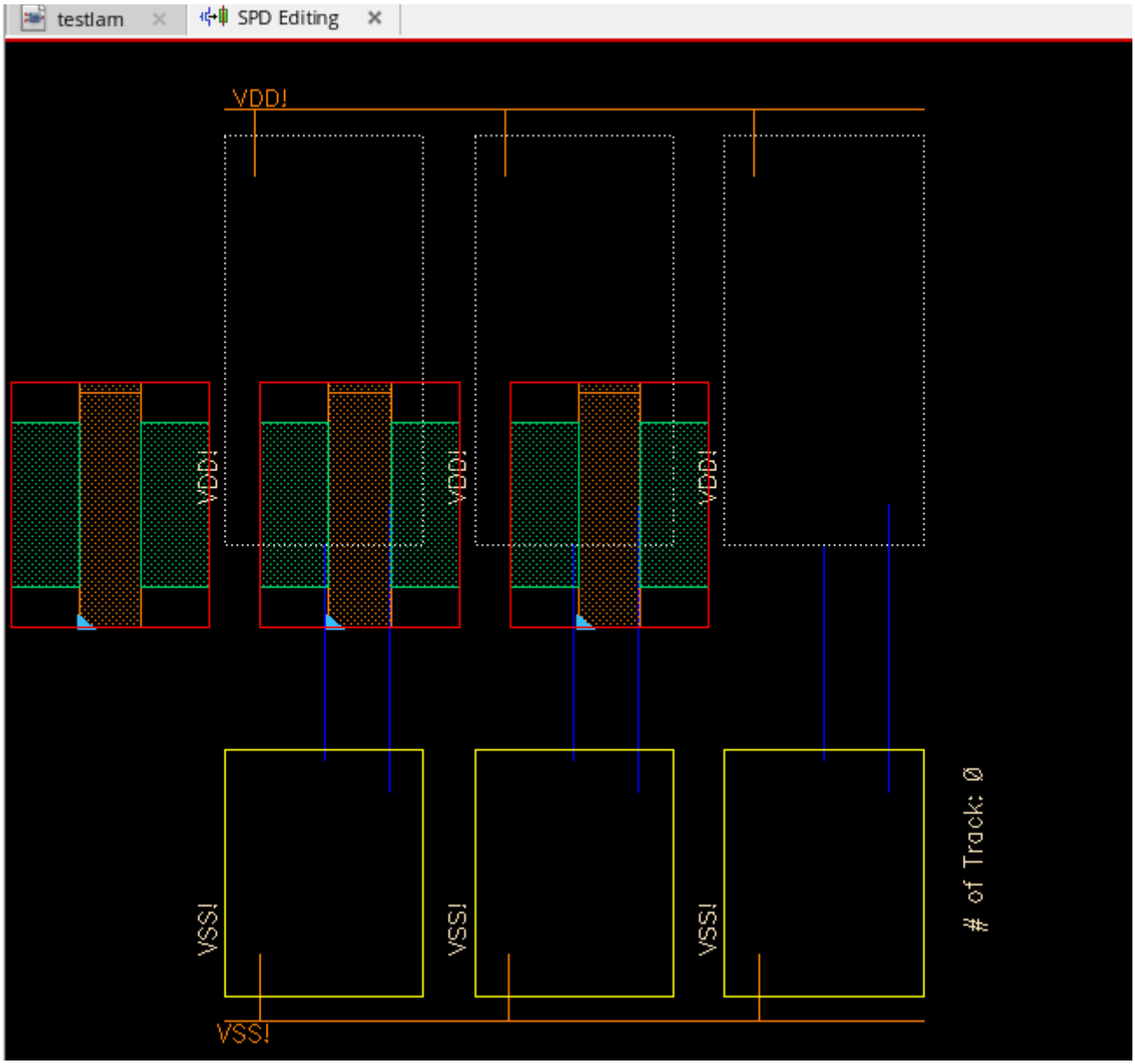
5. Exit SPD.
6. Deselect all devices in Layout XL and generate the SPD layout again.

   SPD opens with a blank canvas.
7. Open the Layout XL window, and partially select some instances.
8. Click the *Generate Symbolic Placement* button from the Layout XL toolbar.

© 2023 Cadence Design Systems, Inc.

The SPD command *Append Selected from Source* is triggered and SPD window is displayed automatically.

Partially-selected devices are added to the SPD layout.



Removing Symbolic Devices

To remove devices from the symbolic design:

1. Select one or more devices.

© 2023 Cadence Design Systems, Inc.

You can select both P type and N type devices.

2. Select the *Edit – Remove* [`Delete`] or click the *Remove* button on the SPD toolbar.

   The selected devices are removed from the design.

If the removed device is part of a chain, the chain might break when you remove the devices. Also, devices are not packed automatically after you remove the devices.

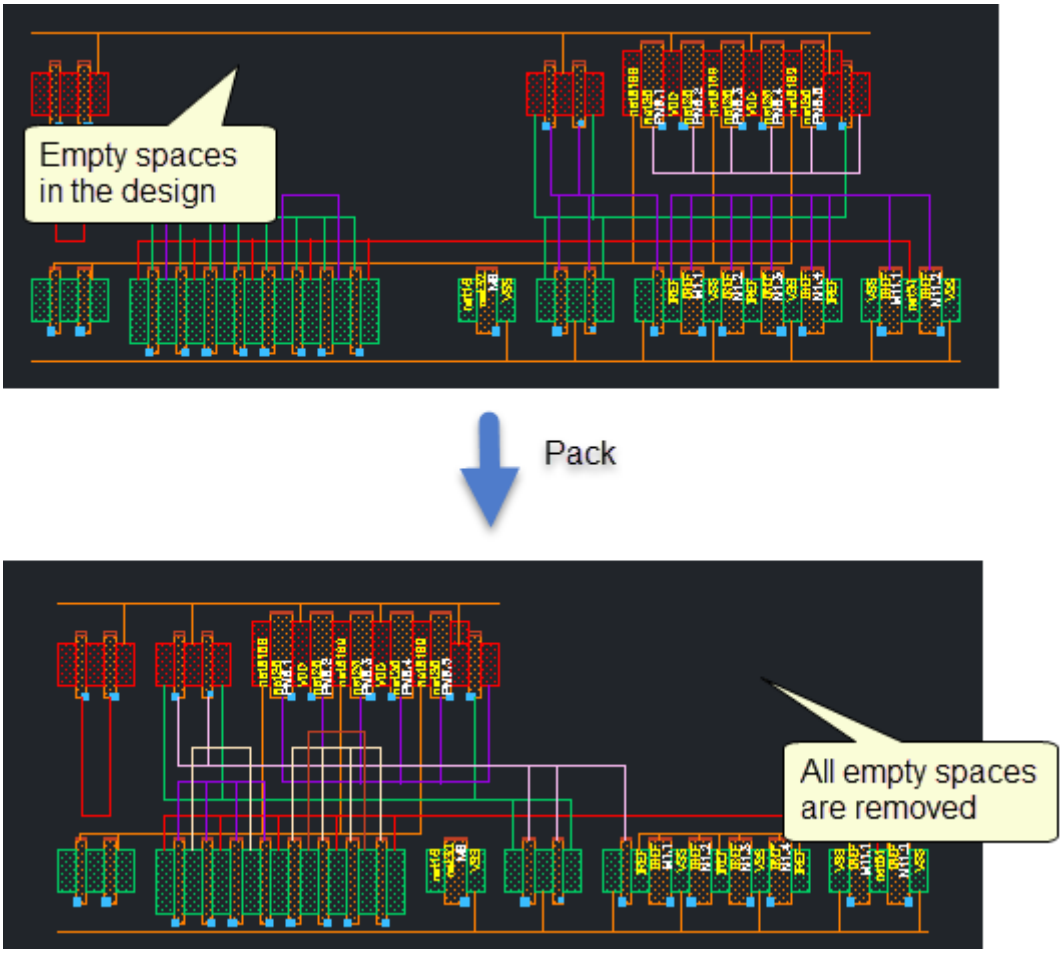
Packing the Design

After working on your design in SPD, you might find that you have created some empty spaces in the canvas, for example, after removing instances from your design. You can remove these spaces by packing the design. To do this:

➡ Choose *Place – Pack* [`p`] or click the *Pack* button on the SPD toolbar.

© 2023 Cadence Design Systems, Inc.

All empty spaces from the design are removed and the design is packed.



Resetting Placement

The *Reset Placement* option lets you discard any placement changes you made to the symbolic design and revert to the placement set in the SPD Options form.

You can reset placement in one of the following ways:

- To reset placement for all the instances currently in the canvas:

  Choose *Place – Reset Placement – Reset Placement* or click the *Reset Placement* button on the SPD toolbar.

- To reset placement of the selection set that you used to launch SPD and all devices you added using the *Append Selected From Source* Command:

© 2023 Cadence Design Systems, Inc.

Choose *Place – Reset Placement – Reset Placement & Selection* or select the *Reset Placement & Selection* option from the *Reset Placement* drop-down menu on the SPD toolbar.

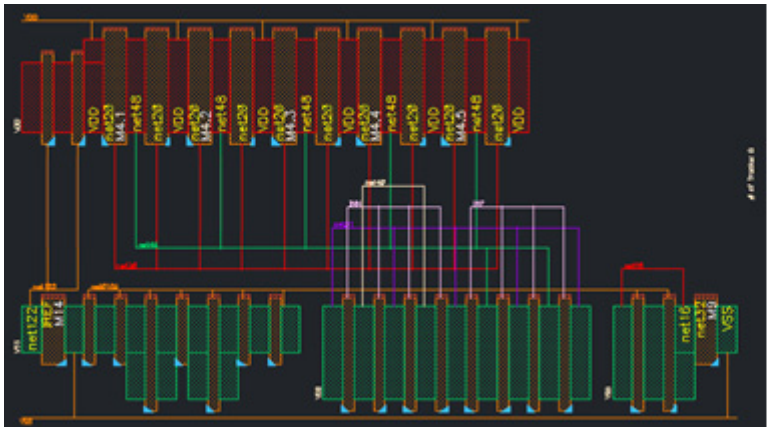
Displaying Flight Lines

SPD provides the following options for displaying flight lines in the canvas:

- Display flight lines for gate and active regions
- Display flight lines for gate nets only
- Display flight lines for active regions only
- Remove all flight lines

Display flight lines for gate and active regions

To display flight lines for both gate and active regions, choose *Place – Flight Lines – Both* or select the *Flight Lines Both* option from *Flight Lines* drop-down menu on the SPD toolbar.

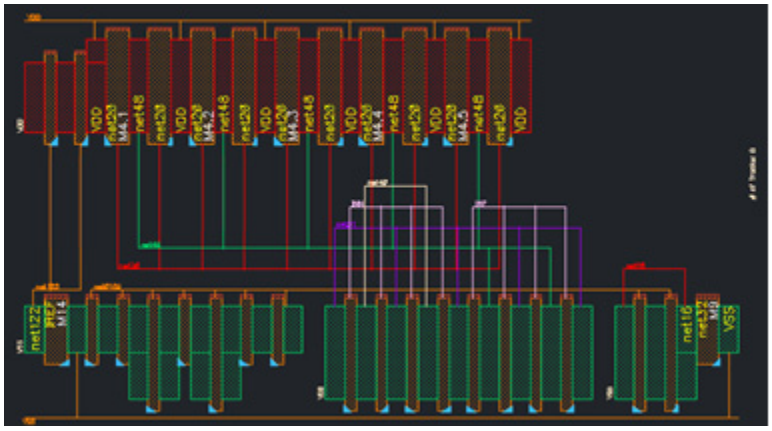


Flights lines for both gate nets and the active region nets are visible in the design.

© 2023 Cadence Design Systems, Inc.

Display flight lines for gate nets only

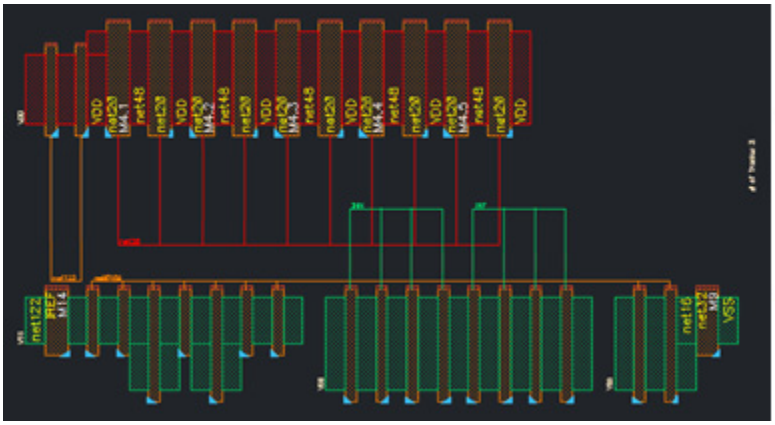
To display flight lines only for gate nets, choose *Place – Flight Lines – Gate Only* or select the *Flight Lines Gate Only* option from *Flight Lines* drop-down menu on the SPD toolbar.



Flights lines for both gate nets and the active region nets are visible in the design.

Display flight lines for active regions only

To display flight lines for only active regions, choose *Place – Flight Lines – Active Only* or select the *Flight Lines Active Only* option from *Flight Lines* drop-down menu on the SPD toolbar.

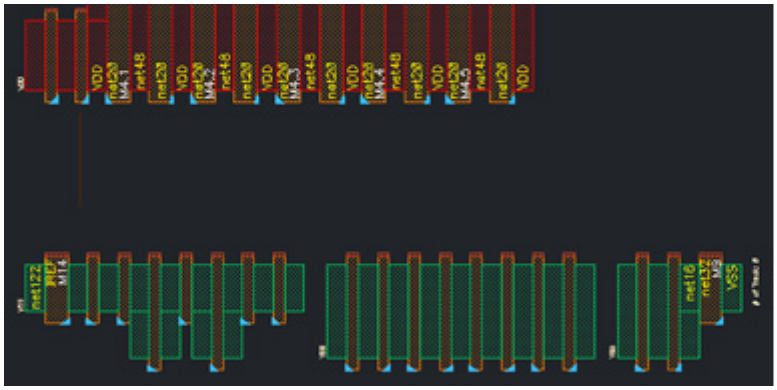


Flight lines for gate nets only are visible in the design

© 2023 Cadence Design Systems, Inc.

Remove all flight lines

To remove all flight lines, choose *Place – Flight Lines – None* or select the *Flight Lines None* option from the *Flight Lines* drop-down menu on the SPD toolbar.



Reordering and Resetting Trunks

SPD allows you to convert orthogonal flight lines into initial trunks in the SPD Editor window. You can reorder and set layer, width, and spacing information for these trunks in the SPD Editor. When you generate the layout, these trunks are shown as initial trunks in the layout that you can use to complete Pin-to-Trunk routing in Layout XL.

Video

For a video on signal trunks, see **Creating Signal Trunks in SPD for Pin-to-Trunk Routing** on Cadence Online Support.

You can use the SPD Editor to perform the following tasks

- Creating Signal Trunks
- Reordering Trunks
- Resetting Trunk Order

Creating Signal Trunks

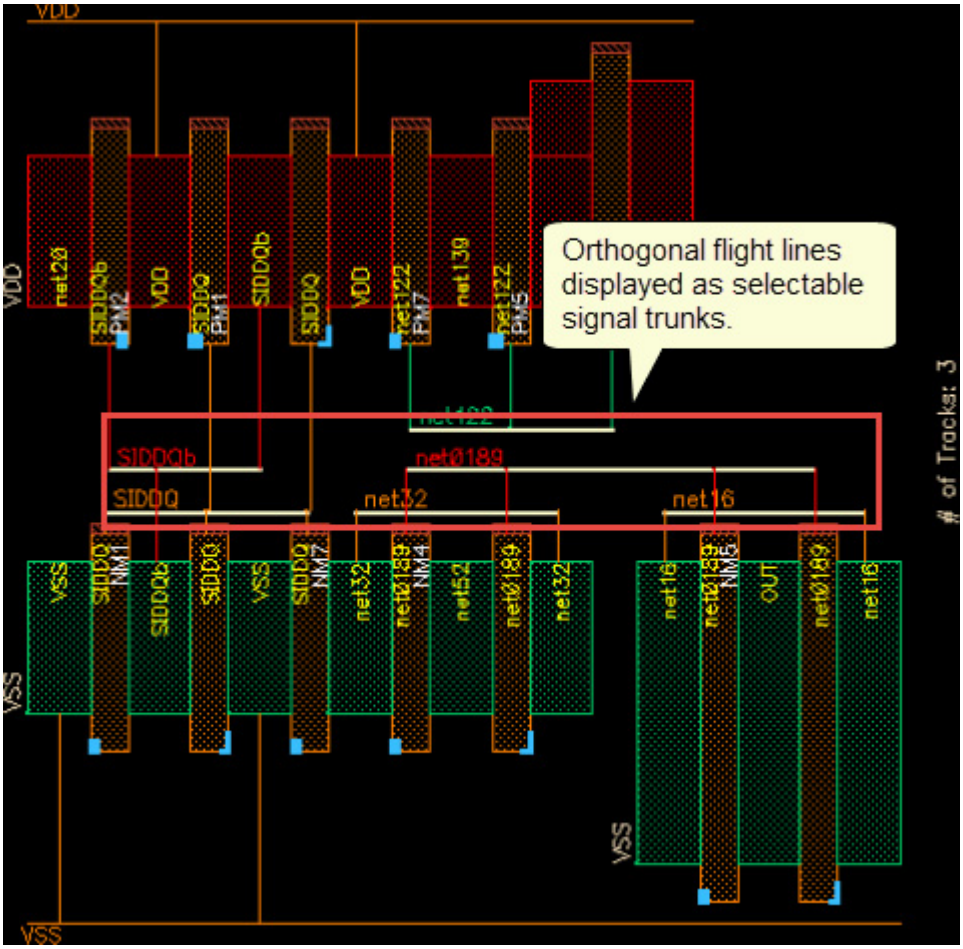
To create signal trunks in the symbolic design:

1. Choose *Options – Power Rails – Signal Trunks – Create.*
2. Specify layer purpose for creating the signal trunks.

© 2023 Cadence Design Systems, Inc.

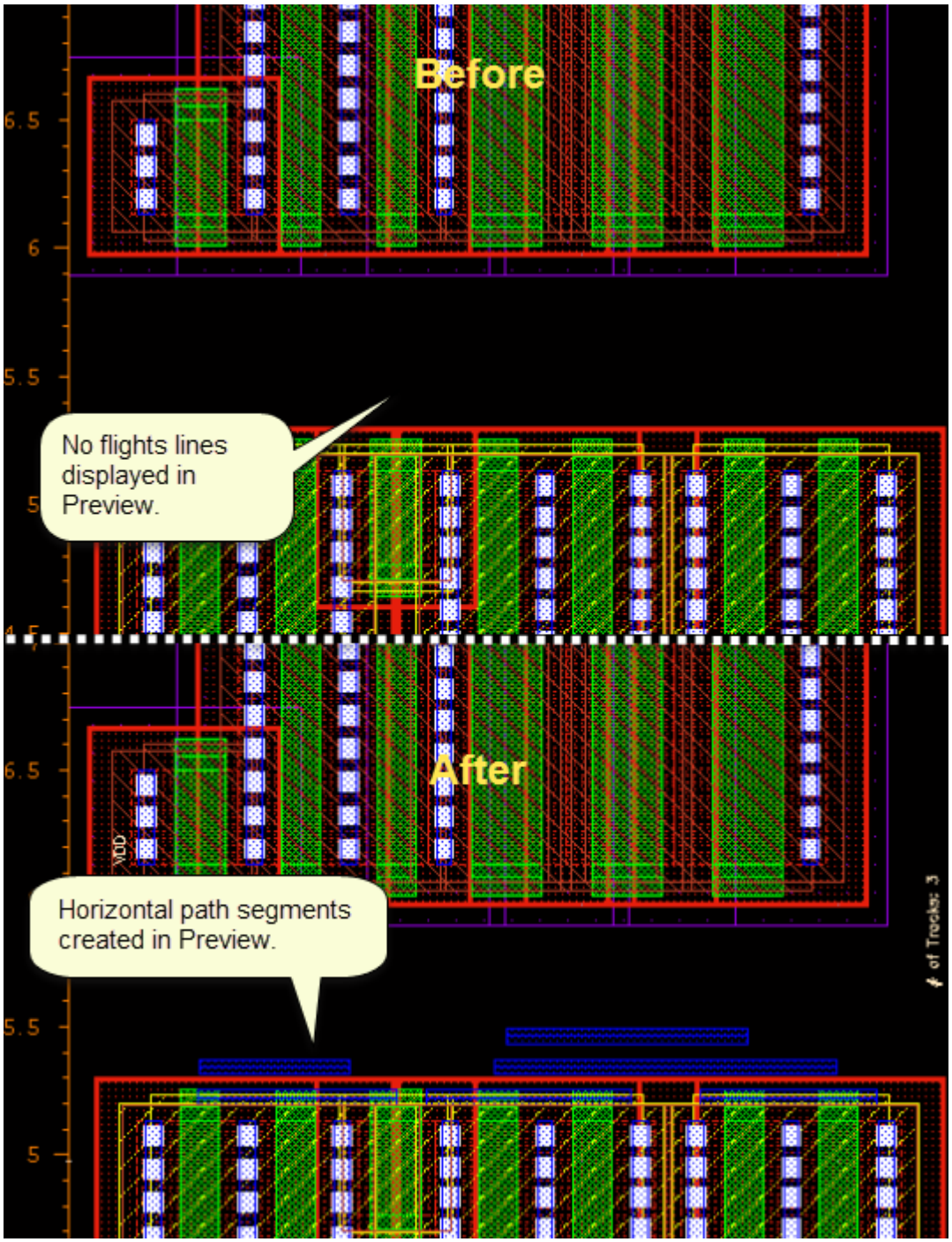
3. Specify the *Width*.
4. Specify *Trunk-Trunk* and *Trunk-OD* spacing.
5. Click *OK*.

   Orthogonal flight lines are converted to and displayed as selectable signal trunks in the design.



© 2023 Cadence Design Systems, Inc.

6. Click the *Preview* button.



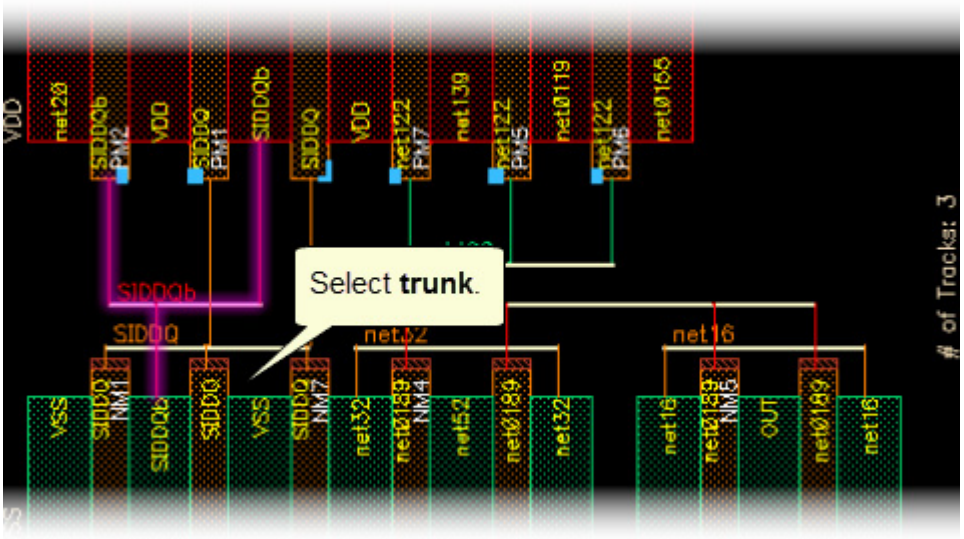
Note: Tracks in the Preview window might differ from that in SPD Editor because real size of devices is displayed in preview.

© 2023 Cadence Design Systems, Inc.

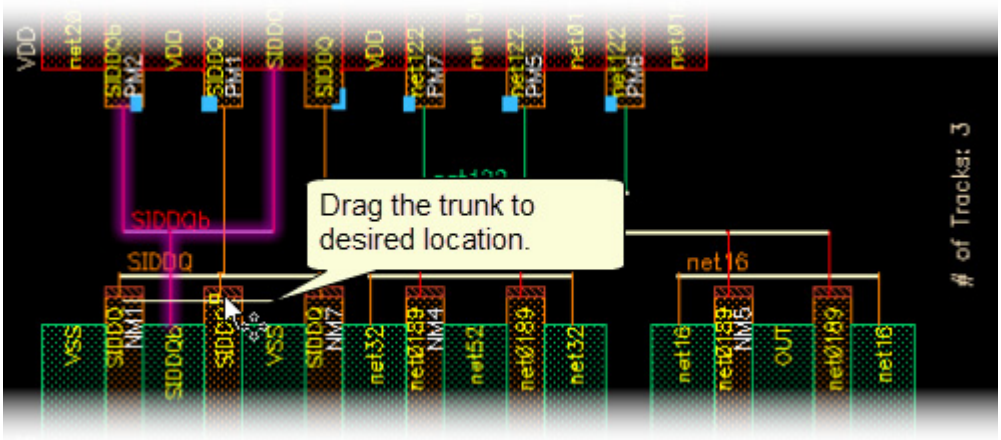
Reordering Trunks

To reorder trunks:

1. Select the trunk you want to move.



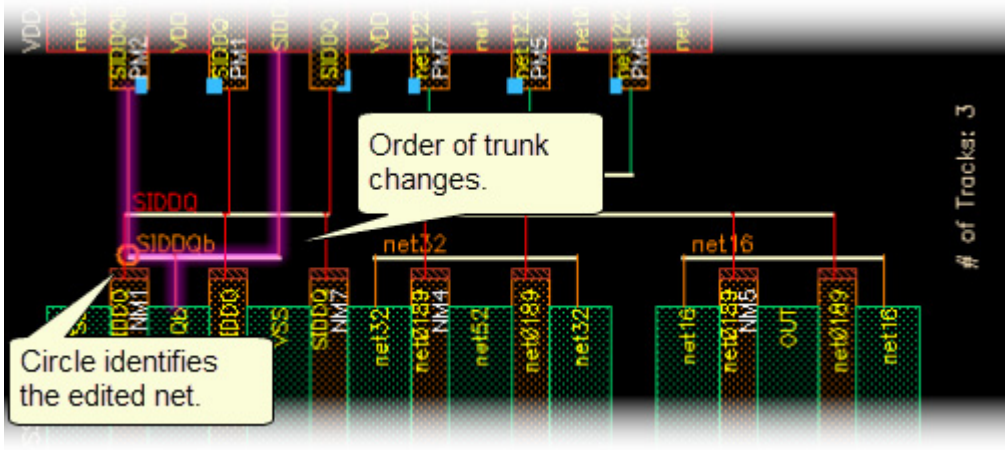
2. Click the *Reorder Trunks* button.
3. Drag it to the location where you want to place it.



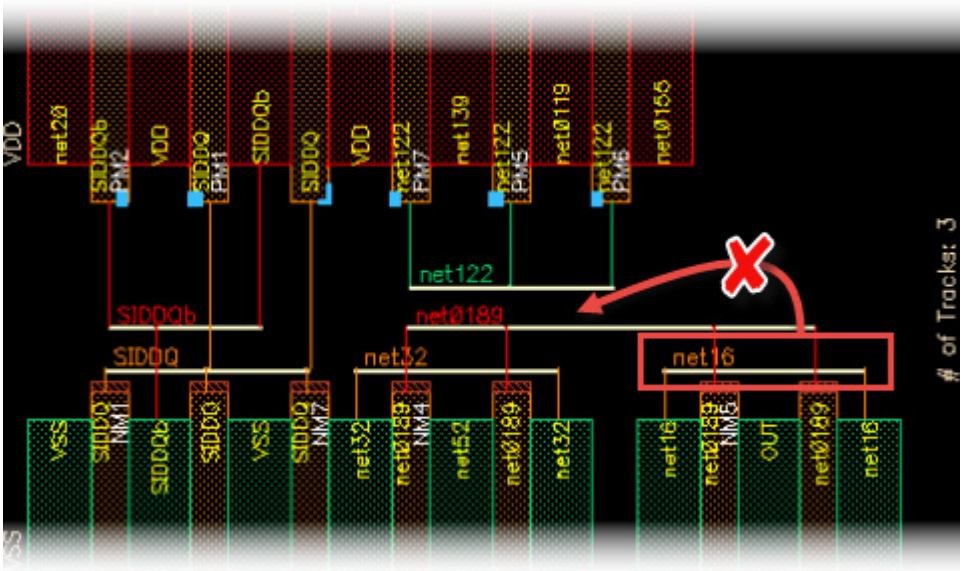
4. Drop the trunk at the desired location to change its order.

© 2023 Cadence Design Systems, Inc.

5. The order of trunk changes.



In SPD, trunks are compacted to minimize the number of tracks. Therefore, in the current example, if you try to move `net16` higher than `net0189`, `net16` will be still dropped into the empty track.



Resetting Trunk Order

➡ To reset the trunk order, click the *Reset Trunk Order* button.

© 2023 Cadence Design Systems, Inc.

Probing Selected Nets

Probing lets you select nets in the symbolic design to highlight the corresponding element in the schematic window. This helps you interactively explore the design and check that the selected net has a corresponding counterpart in the schematic window and where it is placed in the canvas. You can do the following:

- Probe Nets
- Probe Nets and Align Devices
- Clear All Probes

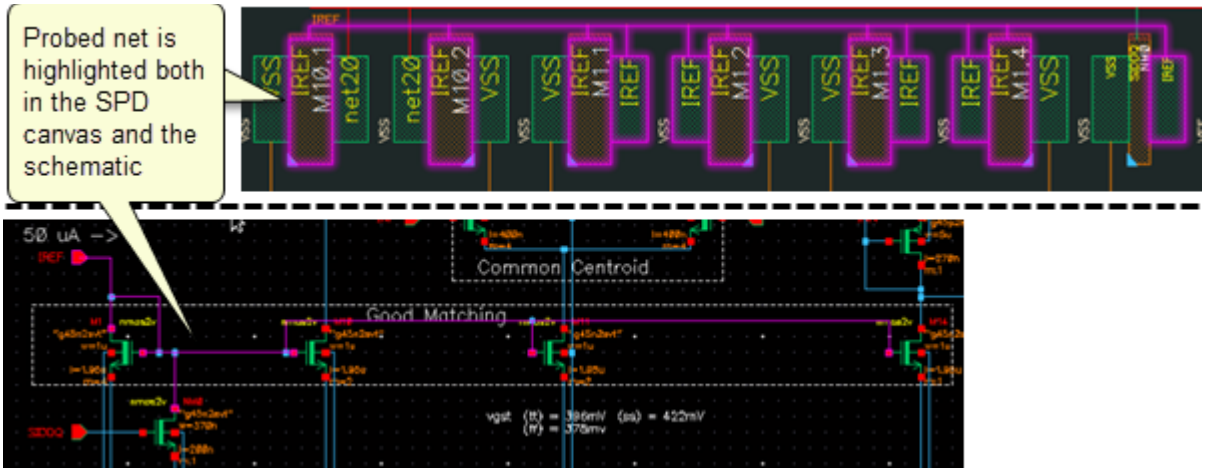
© 2023 Cadence Design Systems, Inc.

Probe Nets

To probe a net in the symbolic design:

1. Choose *Place – Probe Net – Probe Net* or click the *Probe Net* button on the SPD toolbar.
2. Select a gate, source or drain of a device in the symbolic design.

   The net connected to the selected gate, source or drain is highlighted both in the SPD canvas and the schematic.



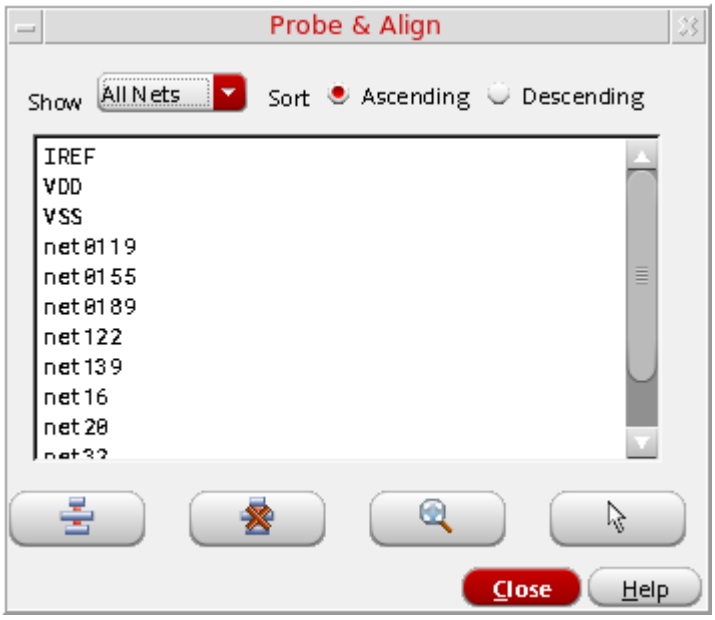
Probe Nets and Align Devices

To probe nets and then align the relevant devices:

1. Choose *Place – Probe Net – Probe & Align* or select the *Probe & Align* option from Probe Net drop-down menu on the SPD toolbar.

© 2023 Cadence Design Systems, Inc.

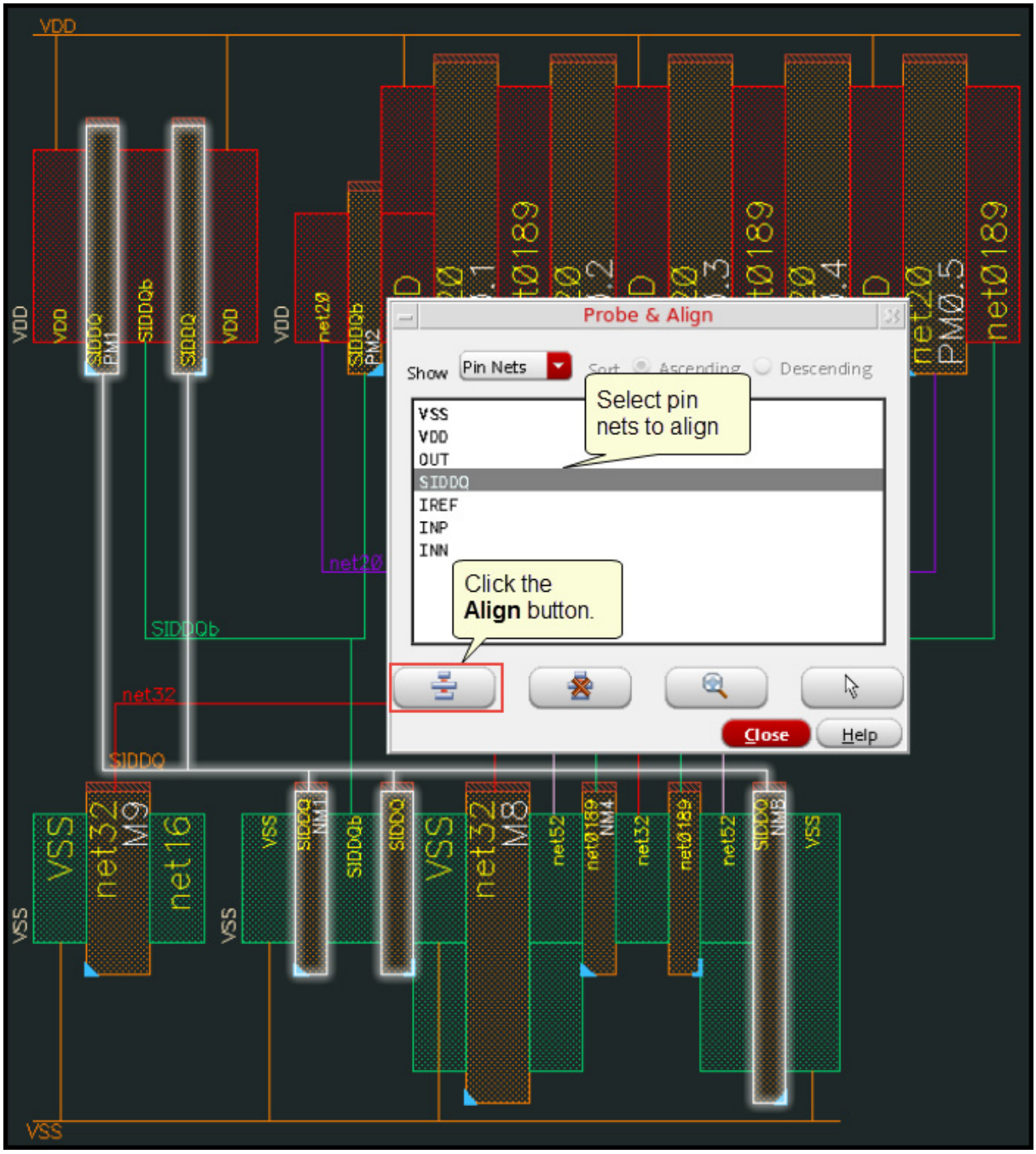
The Probe & Align form is displayed.



2. To align pin nets, select *Pin Nets* from the *Show* drop-down list.
3. Select the nets you want to align.

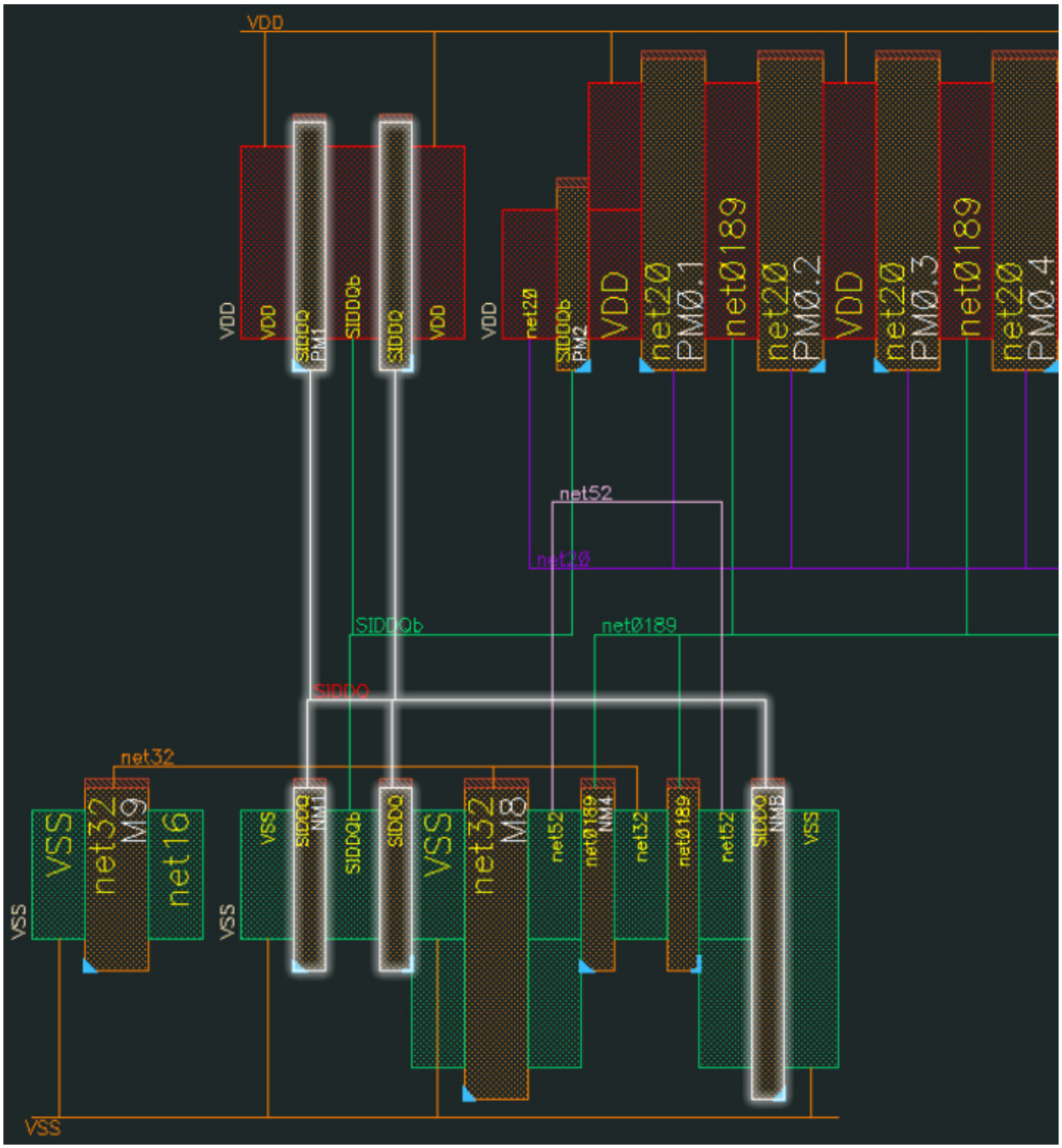
© 2023 Cadence Design Systems, Inc.

4. Click *Align*.



© 2023 Cadence Design Systems, Inc.

The first gate of the two left-most instances in the adjacent P-type and N-type rows of the selected pin net are aligned, as shown in the figure below.

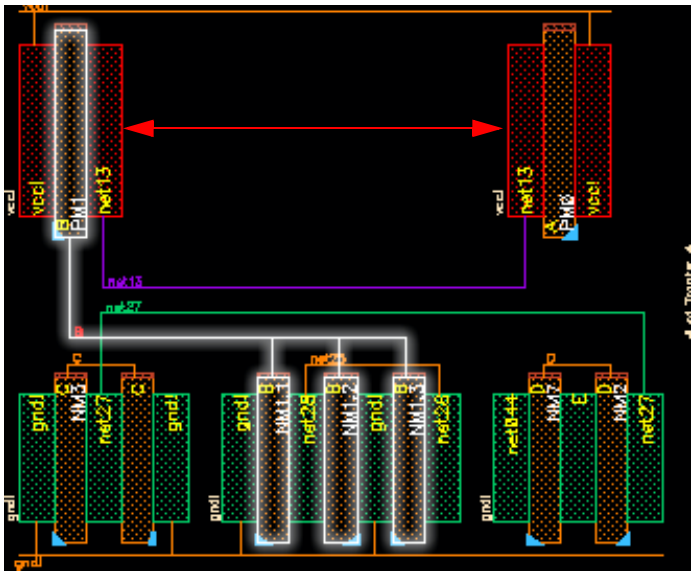


© 2023 Cadence Design Systems, Inc.

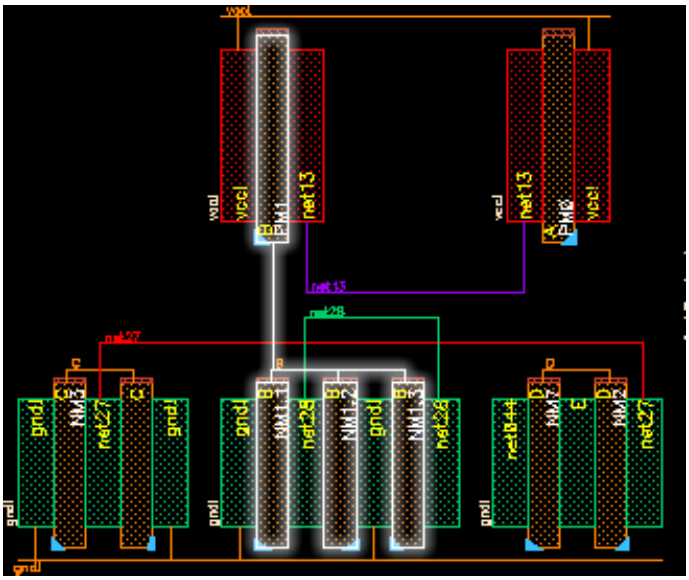
Probing and Aligning Devices Based on Whether Horizontal Packing is Enabled

When the *Options – SPD – Placement – Auto pack in X direction* check box is selected, devices are packed after you probe and align devices. However, if this check box is deselected, automatic packing of devices in the horizontal direction does not take place after devices are probed and aligned. The following steps show this difference.

➡ Select the net you want to probe and align.

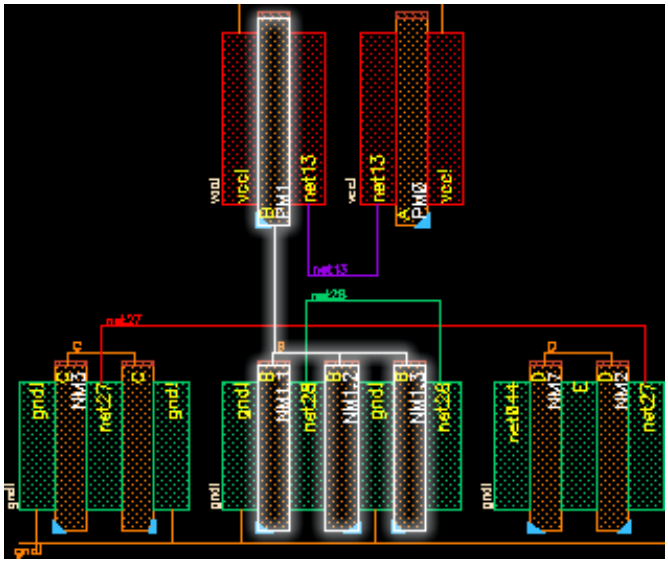


■ When the *Options – SPD – Placement – Auto pack in X direction* check box is not selected, devices are not packed after align.



© 2023 Cadence Design Systems, Inc.

- When the *Options – SPD – Placement – Auto pack in X direction* check box is selected, devices are automatically packed after align.



Clear All Probes

To clear all probes highlighted in the symbolic design canvas, choose *Place – Clear All Probes* or click the *Clear All Probes* button on the SPD toolbar.

For more information on probing, see Probing in the *Virtuoso Layout Suite XL User Guide*.

Checking Source or Updating Layout

The *Check or Update* [`Ctrl + u`] command lets you check the differences between the source design and the design open in the SPD Editing window or update the existing objects in the SPD Editing window from the source design.

After you have updated objects in the SPD Editing window, the edits made prior to the update cannot be undone. However, the design before making the update is saved automatically with a version number. Therefore, you can reverse the update, if needed. The version number of the automatically saved design is `SPD-COU-`*`nn`*`,` where value of *`nn`* is `01` - `99`. The number resets to `01` after `99`.

For information about checking against source and updating component and nets, see Check Against Source and Update Components and Nets and in the Virtuoso Layout Suite XL User Guide.

© 2023 Cadence Design Systems, Inc.

Displaying Free Bucket Size of Devices

By default, SPD displays relative sizes of devices by automatically categorizing them into three-buckets based on their widths– small, medium, and large. However, when device size variations are large, you may want to display the symbolic devices in proportion to their real sizes. You can use the *Free Bucket Size* option to achieve this. In the free bucket size mode, the design in the SPD Editing window looks approximately similar to the actual layout. This option allows you to enable free size based on either the width or length of the instances.

To enable the free bucket size mode:

- In the SPD Editing window, choose *Place – Free Bucket Size* [`Shift+ v`] or click the *Free Bucket Size* button on the SPD toolbar.

To enable the free bucket size based on length:

- In the SPD Editing window, choose *Place – Free Bucket Size – Free L* or select *Free L* from the *Free Bucket Size* drop-down menu on the SPD toolbar. By default, free bucket size is enabled for width of the device.

To revert to display free bucket size based on width:

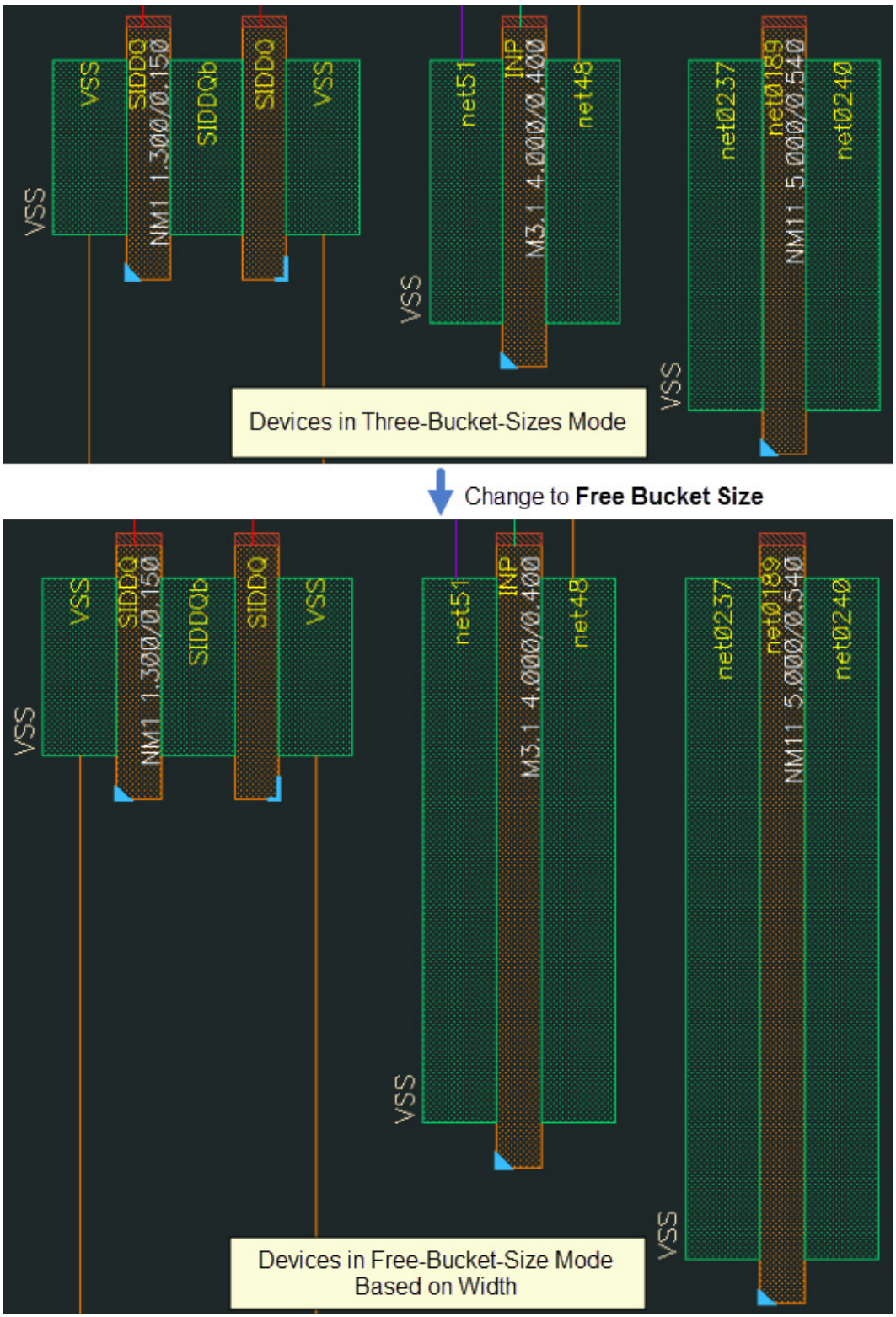
- In the SPD Editing window, choose *Place – Free Bucket Size – Free W* or select *Free W* from the *Free Bucket Size* drop-down menu on the SPD toolbar.

You can also preview gate size in real time. To do this:

- In the SPD Editing window, choose *Place – Free Bucket Size – Gate Size Preview* or select *Gate Size Preview* from the *Free Bucket Size* drop-down menu on the SPD toolbar.

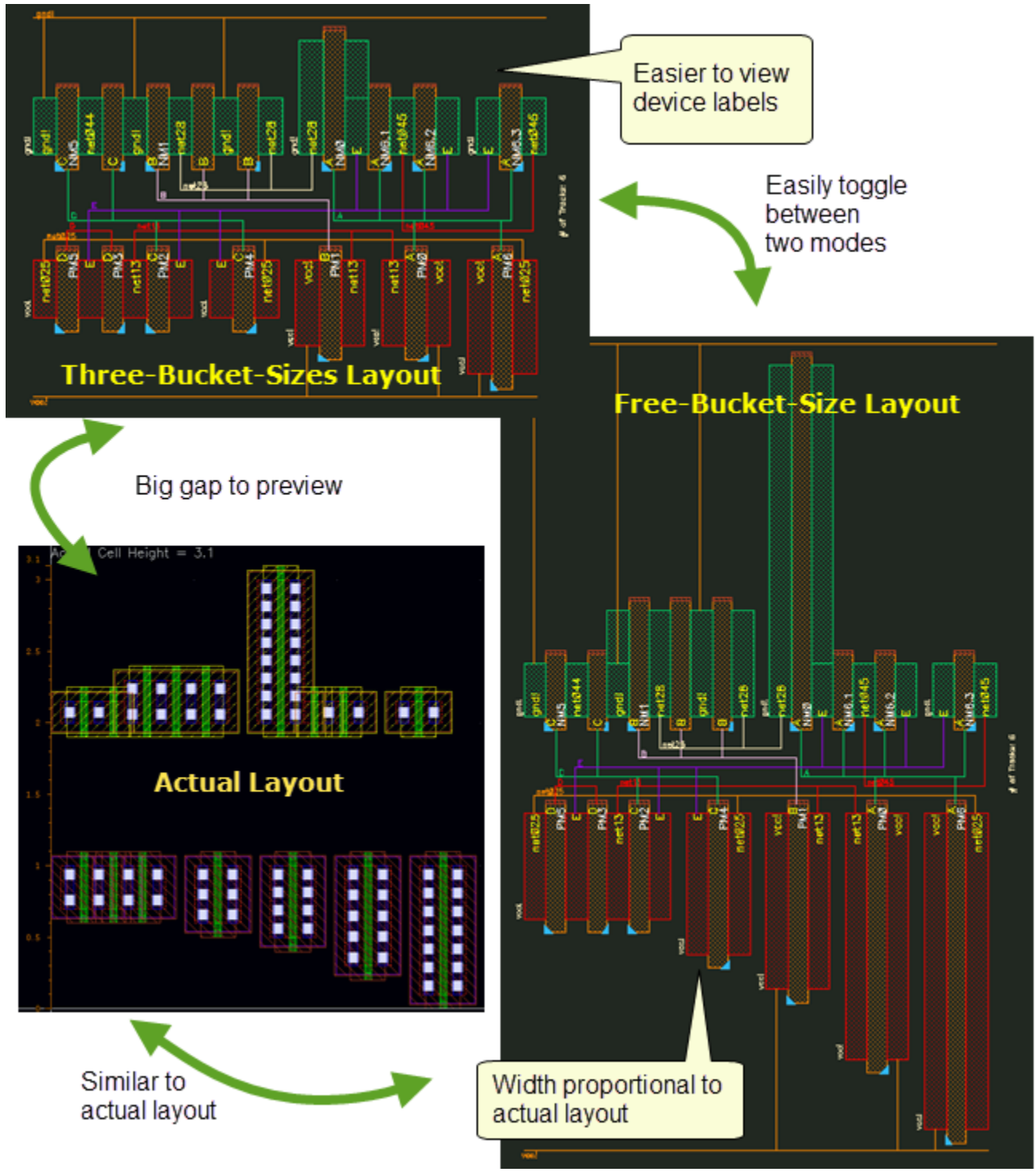
  When free bucket mode is enabled based on width, the smallest device in the design is mapped to the width of the current *S* or *Small* bucket. The rest of the devices are scaled proportionally.

© 2023 Cadence Design Systems, Inc.



© 2023 Cadence Design Systems, Inc.

The following figure compares the SPD layout in three-bucket-sizes layout and free-bucket-size layout to show how the latter looks much similar to the actual layout.



© 2023 Cadence Design Systems, Inc.

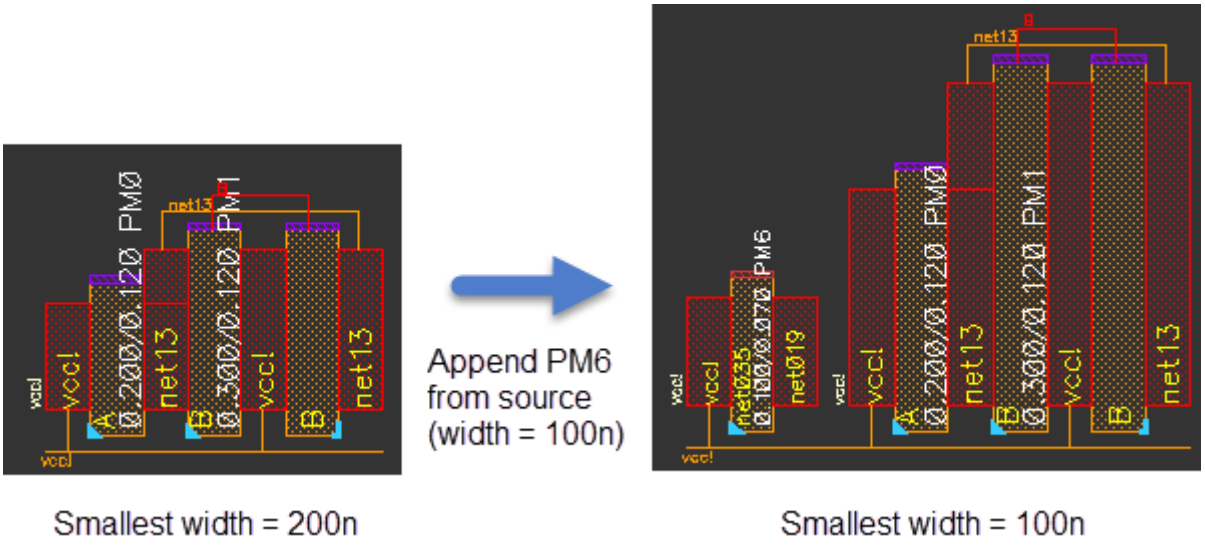
The following rules apply to the devices in the free bucket size mode:

- If the size of the smallest device reduces further, the bucket size is updated according to the new width of this device, and the design is rescaled accordingly.

  The following commands can change the size of the smallest device:

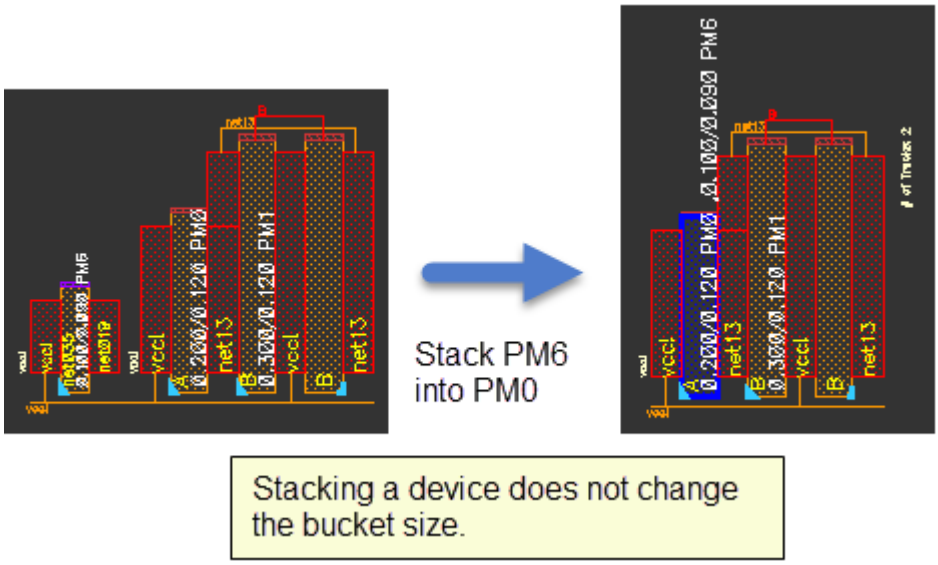
  - *Append selected from source*
  - *Fold*
  - *Add dummy*
  - Check or Update

  The following figure shows how appending a smaller device to the design changes the display of devices in the SPD Editing window.

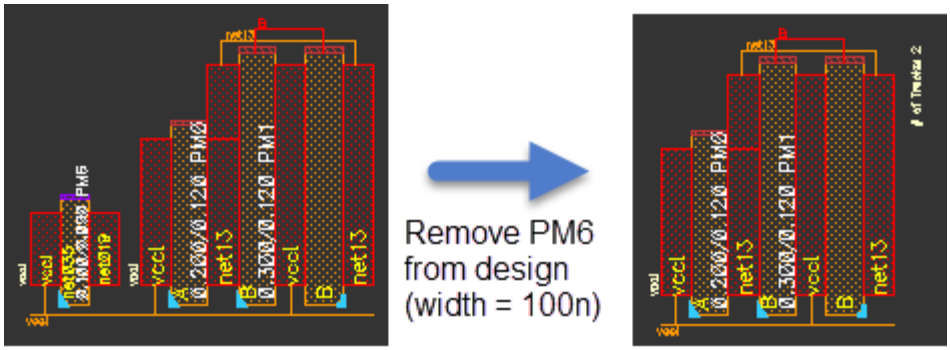


© 2023 Cadence Design Systems, Inc.

- If the smallest instance in the design is either stacked or removed, SPD does not update the bucket size.



Stacking a device does not change the bucket size.

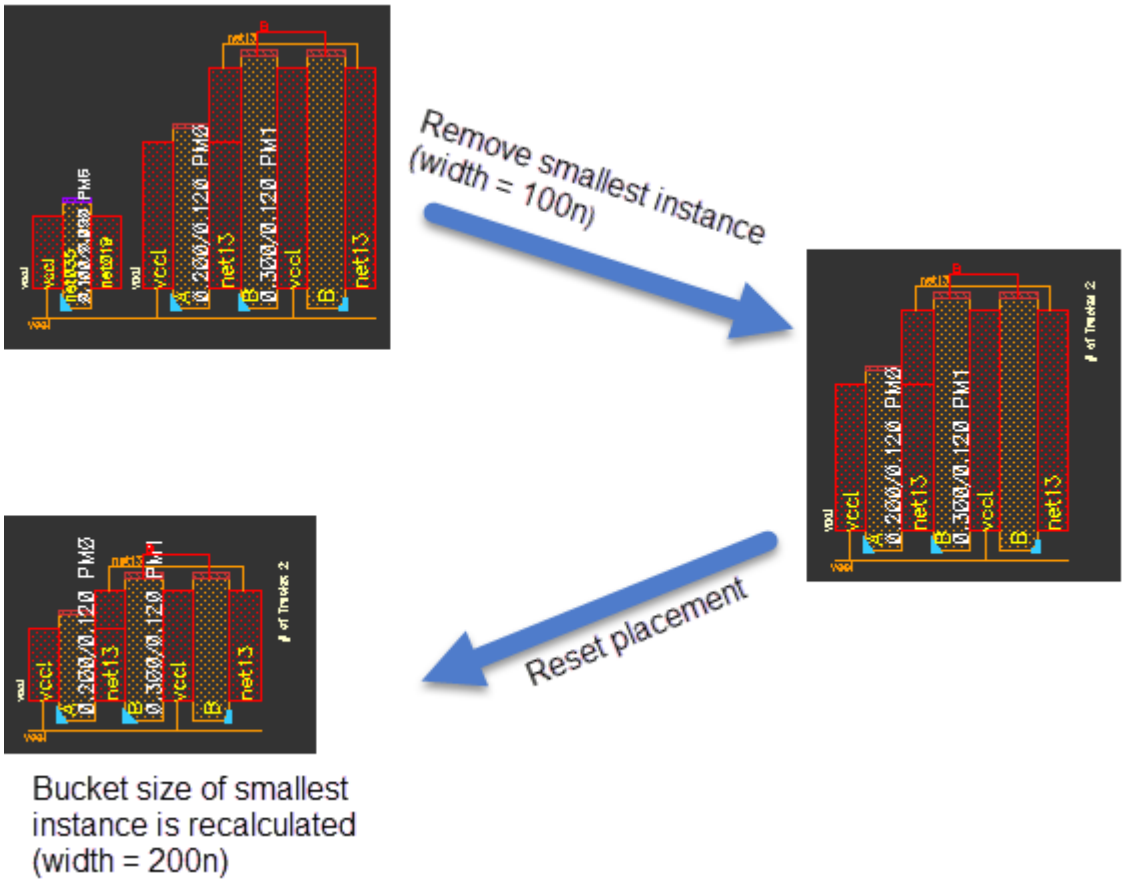


Removing a device does not change the bucket size.

- If the following commands are used, the bucket size is recalculated:
  - *Free Bucket Size* command is enabled or disabled
  - *Reset Placement*

© 2023 Cadence Design Systems, Inc.

- *Check or Update*

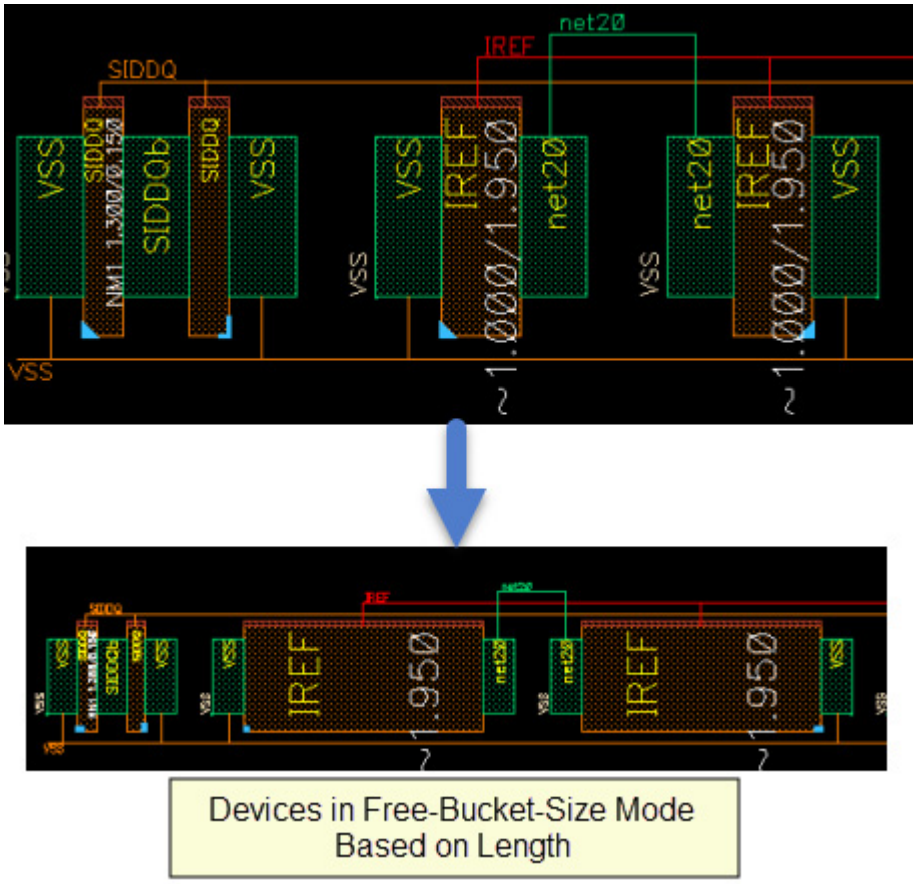


You can also set SPD to display free bucket size based on instance length. To view free bucket size based on instance length:

1. Choose *Options – Display.*
2. Select *Free for Length* in the *Size for Medium Symbol* section.

© 2023 Cadence Design Systems, Inc.

In this mode, the smallest device in the design is mapped to the length of the current *S* or *Small* bucket. The rest of the devices are scaled proportionally.



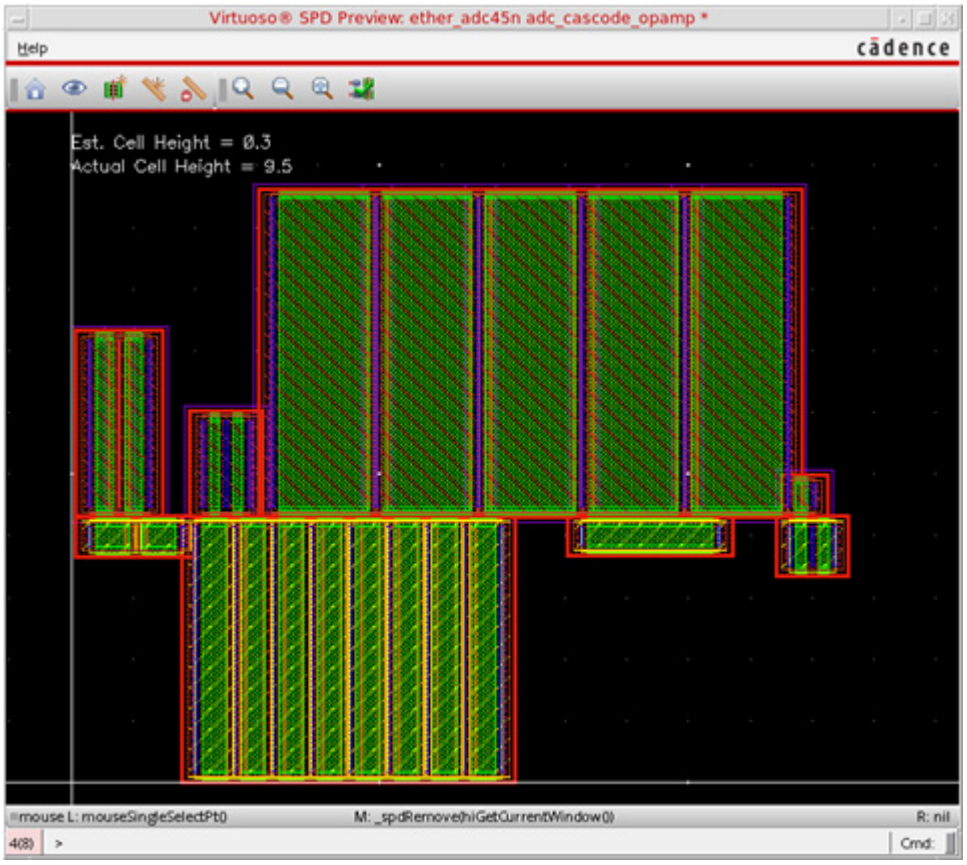
Previewing the Design

SPD lets you preview the physical layout of the symbolic design before you generate the final layout. You can preview the layout as often as you need to until you are satisfied with the design.

To preview the layout, choose *Place – Preview Layout* [v] or click the *Preview* button on the SPD toolbar.

© 2023 Cadence Design Systems, Inc.

The Virtuoso SPD Preview window is displayed.



The SPD Preview display can be controlled using the Display Options form in Layout Editor. To open this form in the Layout Editor window, press [e].

The Snap Pattern Display option on the Display Options form lets you configure whether you want to display the boundary, the tracks, the fins or periods for the snap patterns.

© 2023 Cadence Design Systems, Inc.

The SPD Preview toolbar provides access to the following commands:

Icon	Command	Lets you...
	*Close*	Close the SPD Preview window and return to the SPD Editing window.
	*Abstract*	Enable the Abstracted mode of Smart Display. See View Abstract Preview for more information.
	*Generate Layout*	Generate the final layout and place it in the Layout XL canvas. See Generating the Layout for more information.
	*Move Tracks*	Moves selected tracks vertically in the design preview. See Move Tracks for more information.
	*Pack Tracks*	Snaps tracks to correct location with respect to their relative Y positions. See Pack Tracks for more information
	*Create Ruler*	Create ruler objects to measure devices or the distance between devices. See Create Rulers for more information.
	*Clear Ruler*	Clear all rulers added to the preview. See Create Rulers for more information.

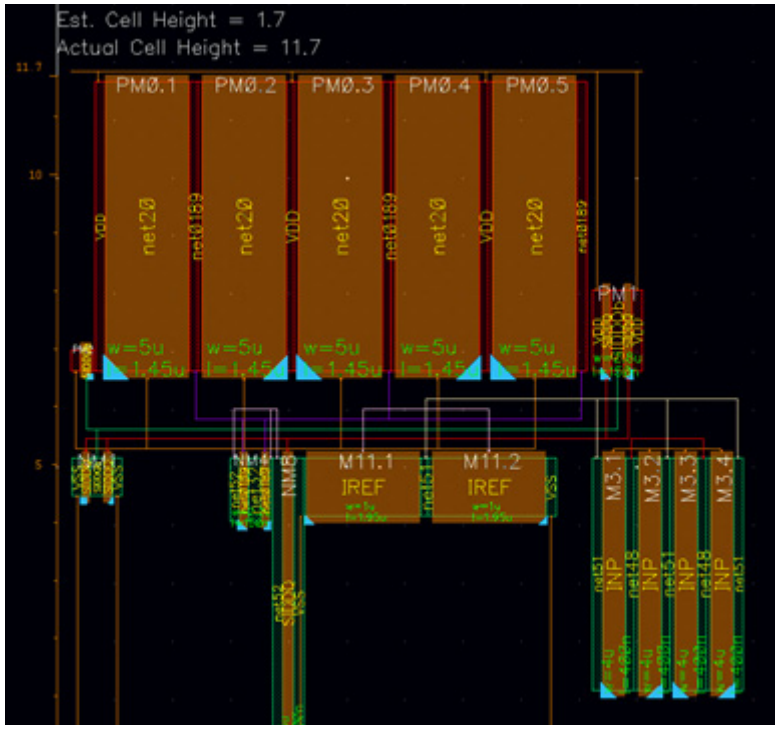
View Abstract Preview

The *Abstract* [a] command in the SPD Preview window enables the Abstracted mode of Smart Display and the connectivity information is displayed in orthogonal flight lines.

The abstract representation of devices makes it easy to identify instance names, instance terminal net names, orientation markers, device parameters and the connectivity that helps you understand the placement in the real layout.

© 2023 Cadence Design Systems, Inc.

For more information on Smart Display, see Using Smart Display in *Virtuoso Layout Suite XL User Guide*.



Create Rulers

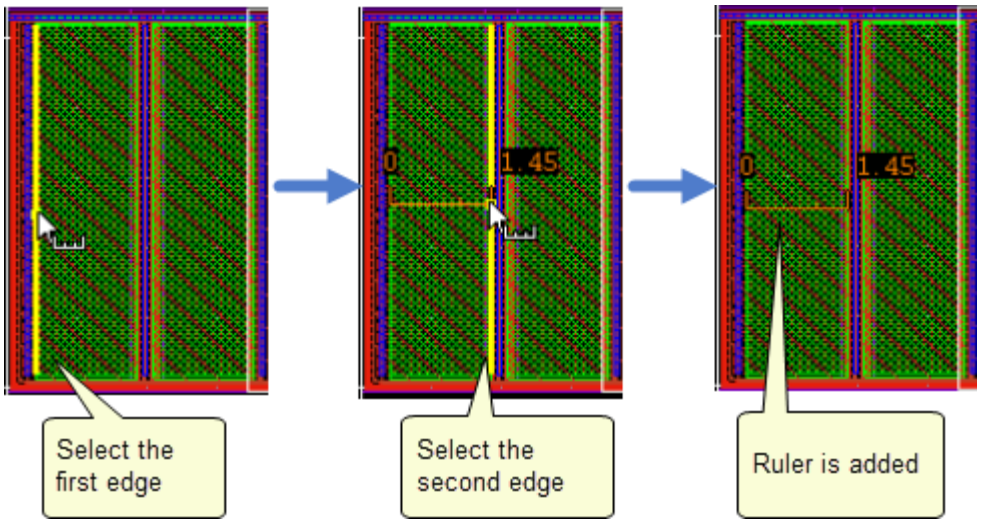
The SPD Preview window displays the estimated cell height and the actual cell height by default. You can also use the *Create Rulers* option to create a ruler object to measure devices or the distance between devices. You can also use the command to measure distances between two edges.

To create rulers:

1. Select the *Create Ruler* [k] button on the SPD Preview toolbar.
2. Select the first edge.
3. Select the second edge.

© 2023 Cadence Design Systems, Inc.

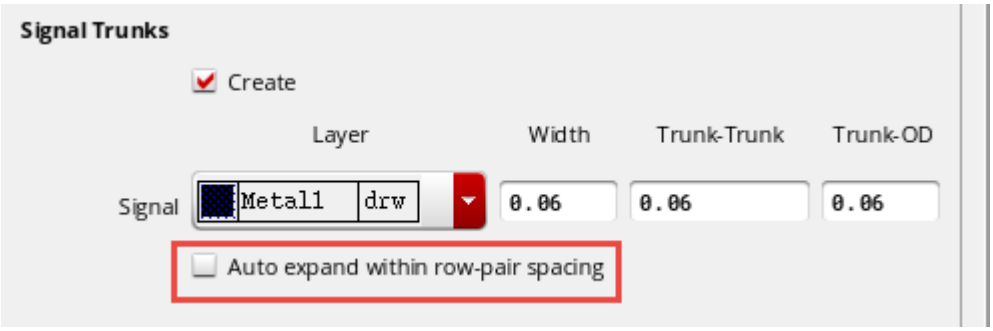
A ruler object displaying the distance between two edges is added to the preview.



To remove rulers added to the design, click the *Clear Ruler* [`Shift + k`] button on the SPD Preview toolbar.

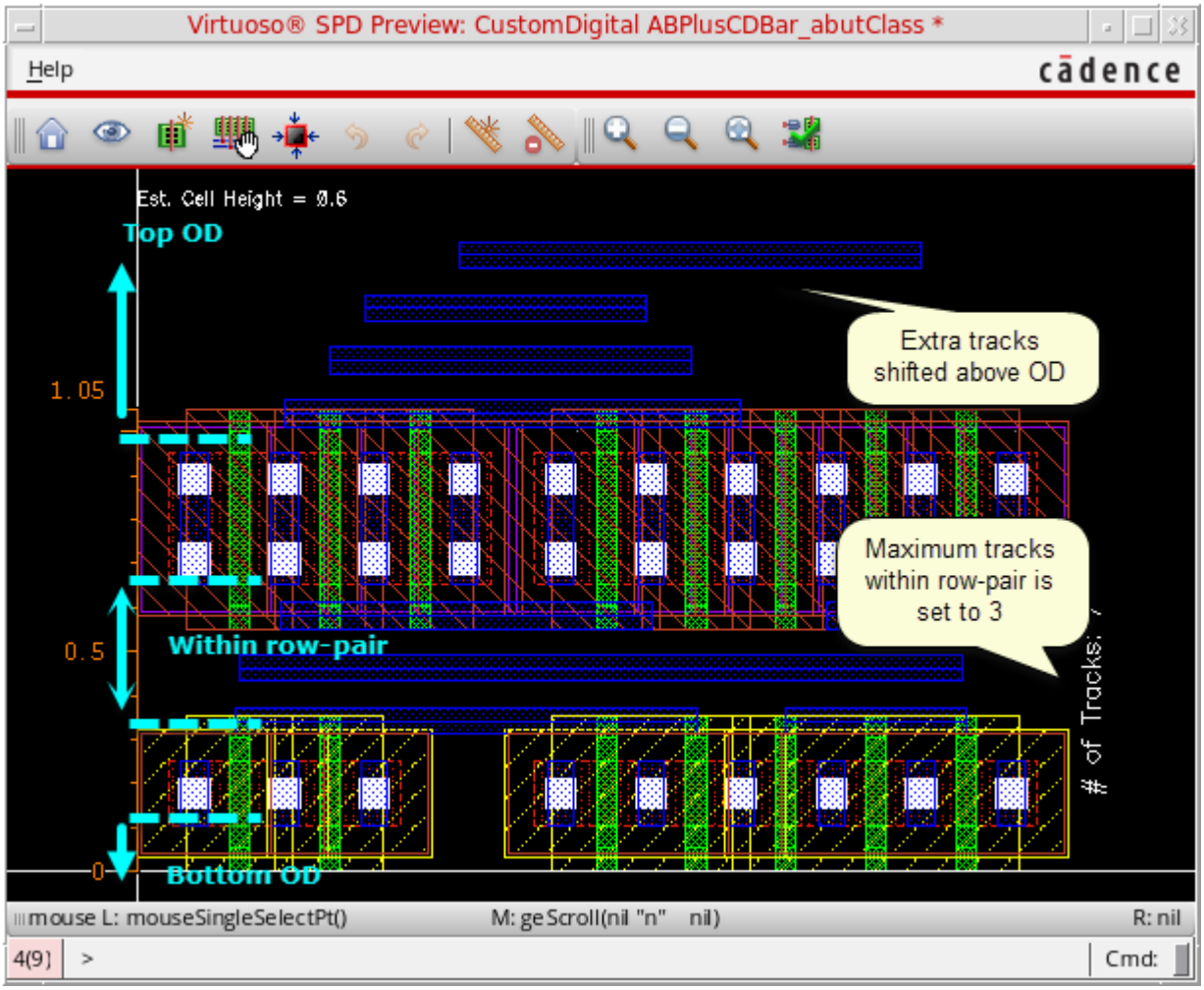
Automatic Shifting of Additional Tracks

The *Auto expand within row-pair spacing* option on the Power Rails options pane, enables you to control whether within row-pair spacing is increased to accommodate additional tracks.



© 2023 Cadence Design Systems, Inc.

When this option is disabled, tracks in the Preview window are shifted automatically over the top oxide diffusion (OD) layer, maintaining the Trunk-to-OD spacing and Trunk-to-Trunk spacing.

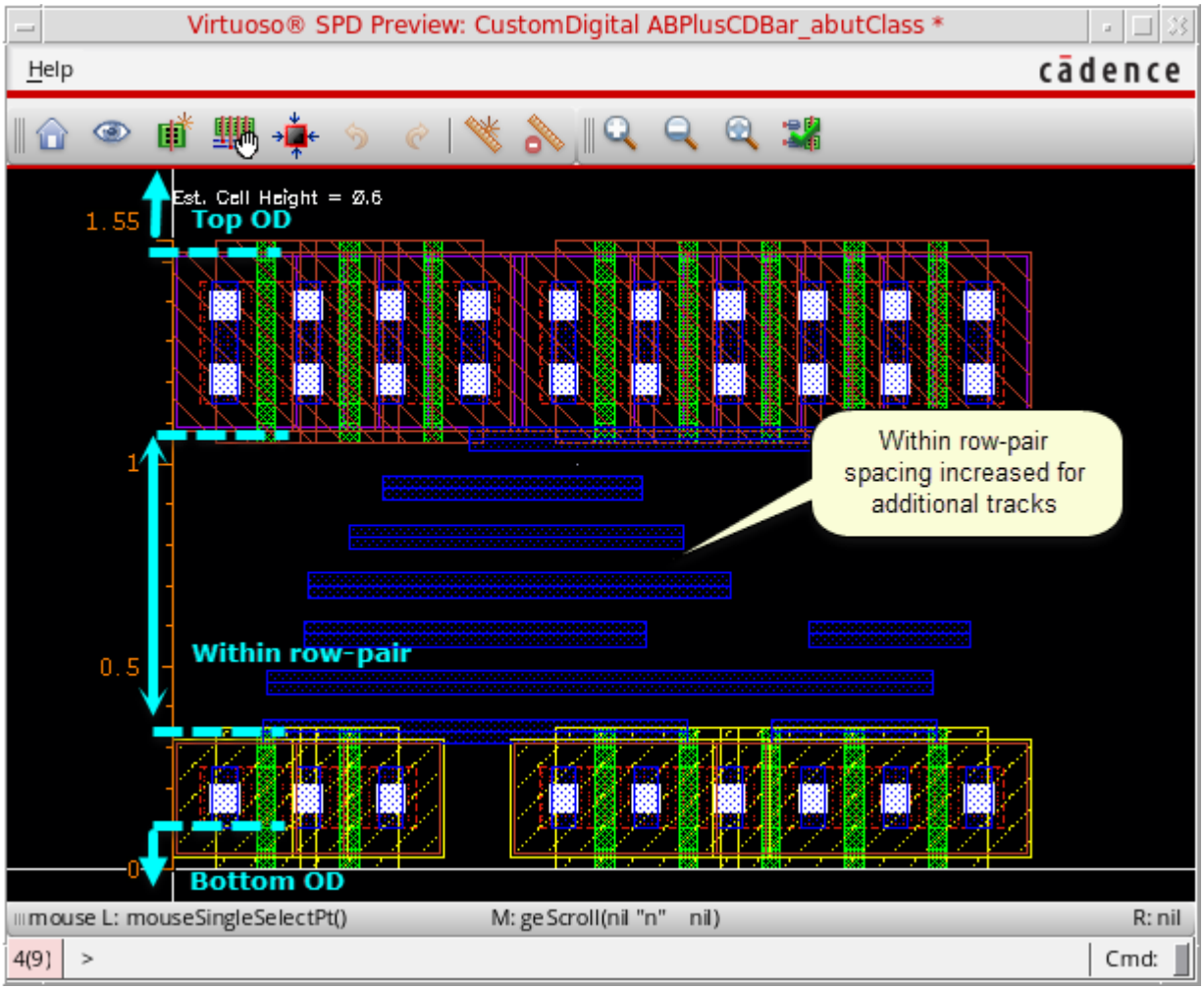


However, when this option is enabled, the maximum number of tracks is calculated using the following information:

- trunk width
- spacing information that includes within row-pair, trunk-to-OD, and trunk-to-trunk spacing values.

Within row-pair spacing increases automatically to accommodate the additional tracks.

© 2023 Cadence Design Systems, Inc.



Move and Pack Tracks

When you choose to create signal trunks on the Power Rails options pane, the following commands are enabled in the Preview window:

- Move Tracks
- Pack Tracks

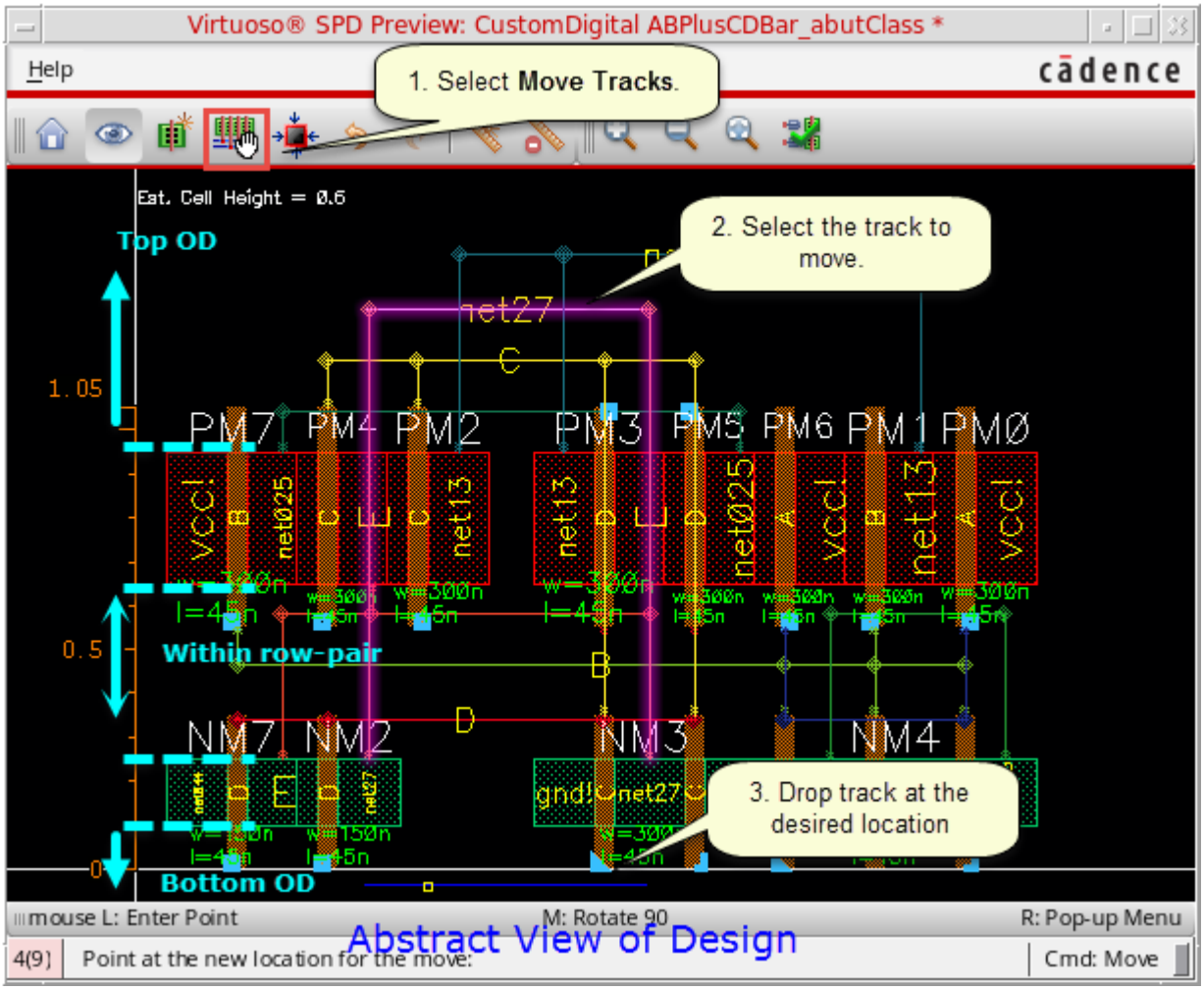
Move Tracks

The *Move Tracks* command enables you to move selected tracks vertically in the Preview window. The selection automatically extends to include or exclude other trunks of the same track so that they can be moved together.

© 2023 Cadence Design Systems, Inc.

To move a track:

1. Choose the *Move Tracks* command.
2. Select the track to move.
3. Drop the selected track at the desired location.



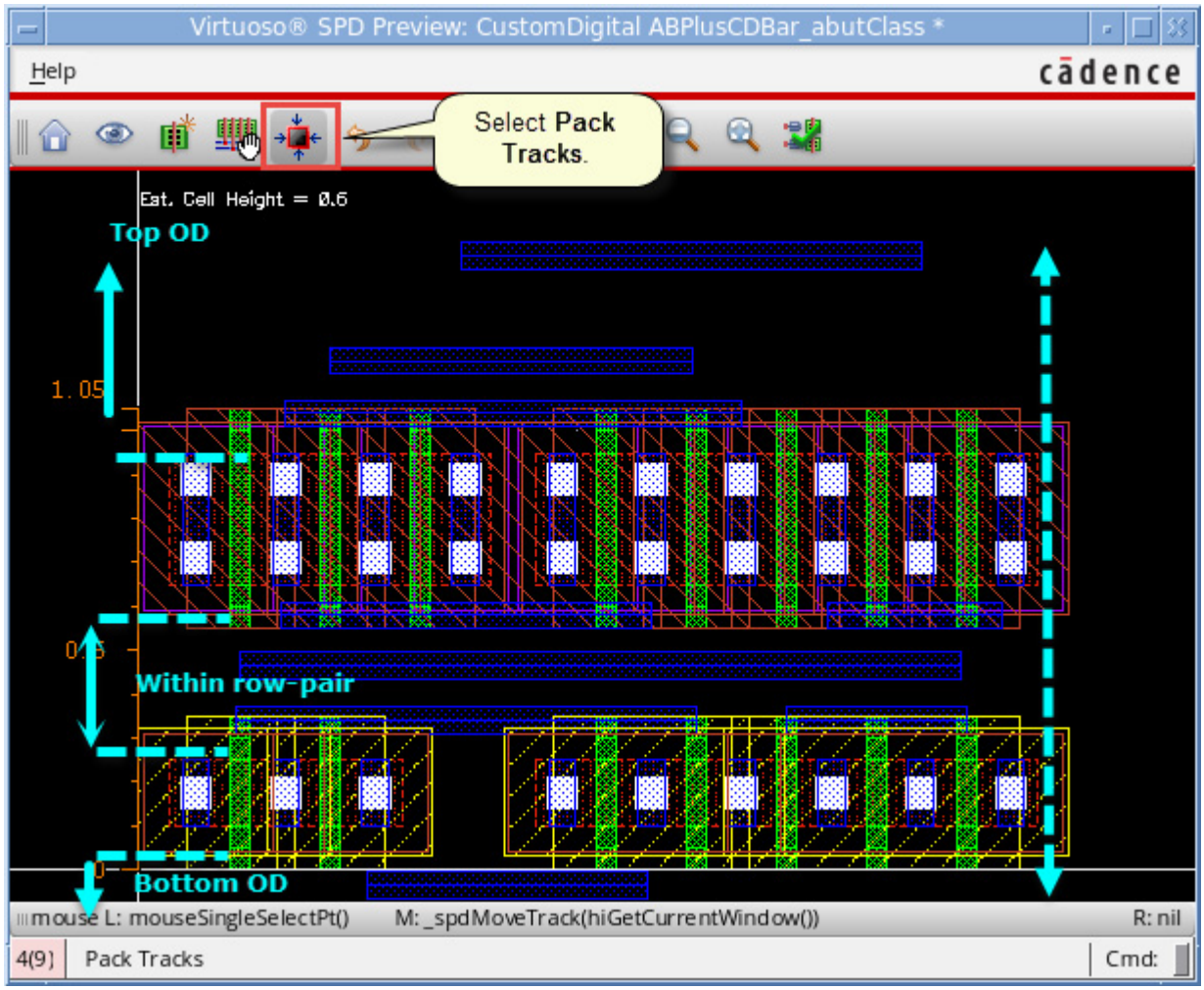
Pack Tracks

The *Pack Tracks* command snaps all tracks in the design to correct location with respect to their relative Y-axis positions. Tracks are packed into three regions separated by the top and bottom OD layers.

To pack the tracks:

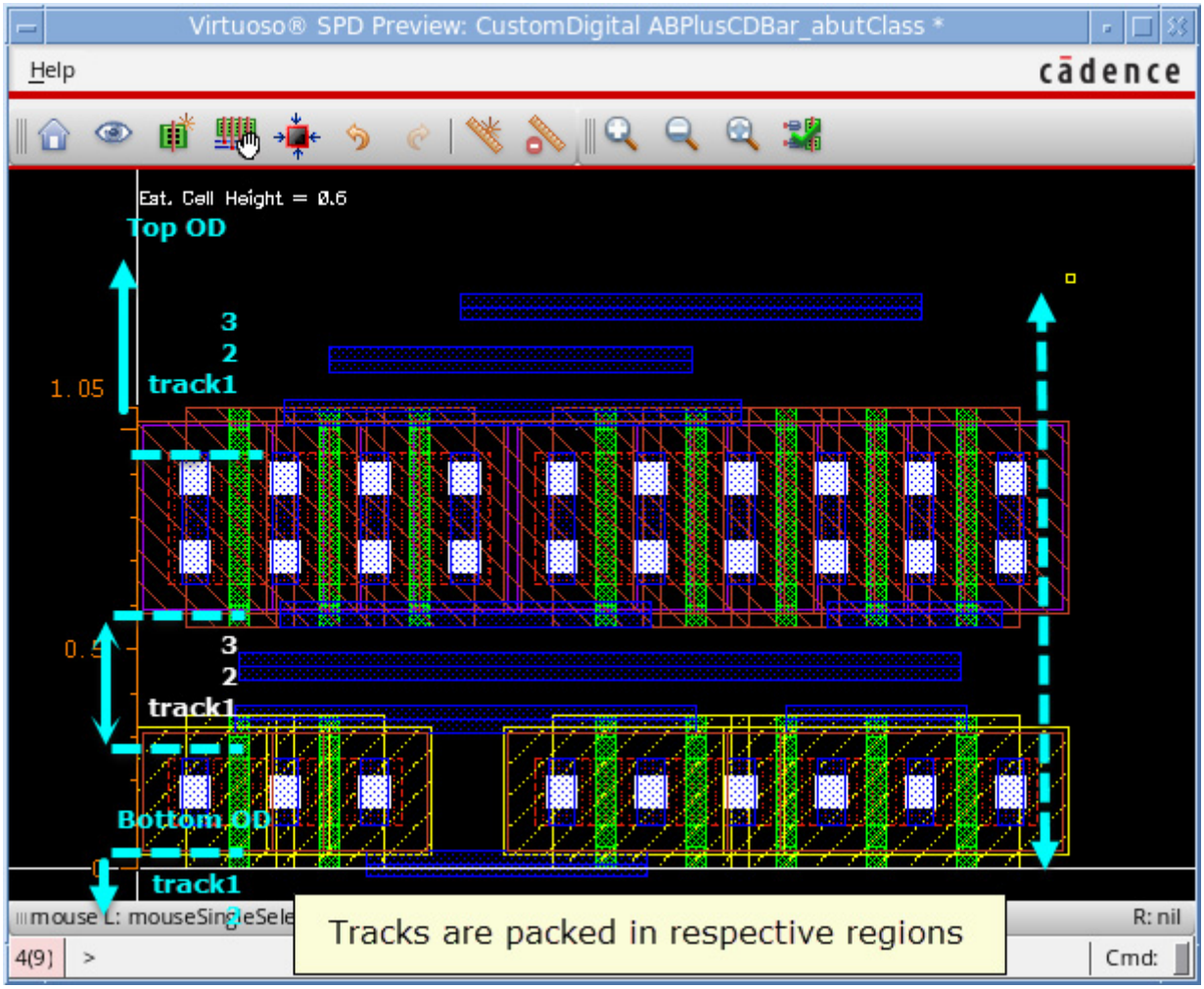
© 2023 Cadence Design Systems, Inc.

➡ Choose the *Pack Tracks* command.



© 2023 Cadence Design Systems, Inc.

Tracks in the design are packed within the three regions separated by top and bottom OD layers.



Preview Based on Whether Horizontal Packing of Devices is Enabled in SPD Editor

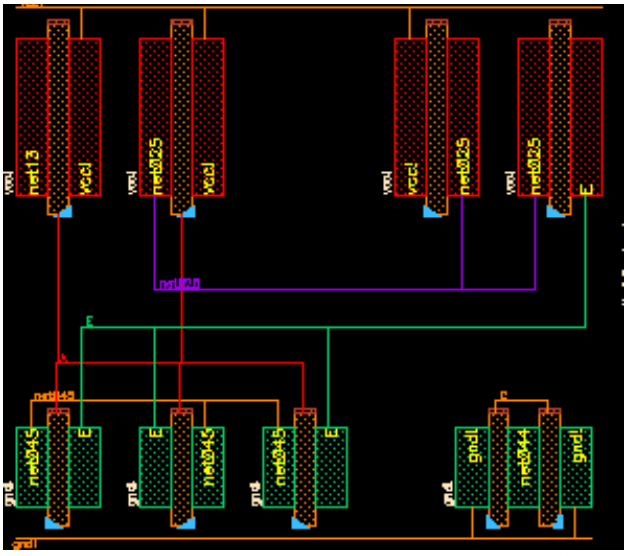
By default, devices in the SPD Preview window will be packed automatically. However, if both *Auto pack in X direction* and free bucket size for length are enabled, preview will honor the placement of devices in the SPD Editing window and the devices in the preview will not be packed.

The following steps show this difference.

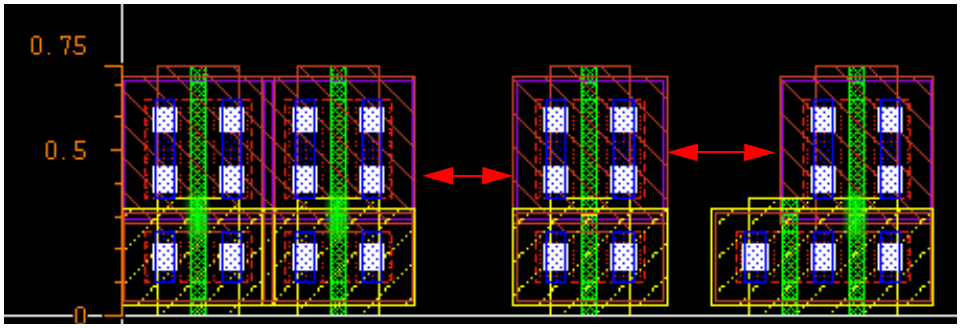
➡ To preview the layout, choose *Place – Preview Layout* [v] or click the *Preview* button on the SPD toolbar.

© 2023 Cadence Design Systems, Inc.

?



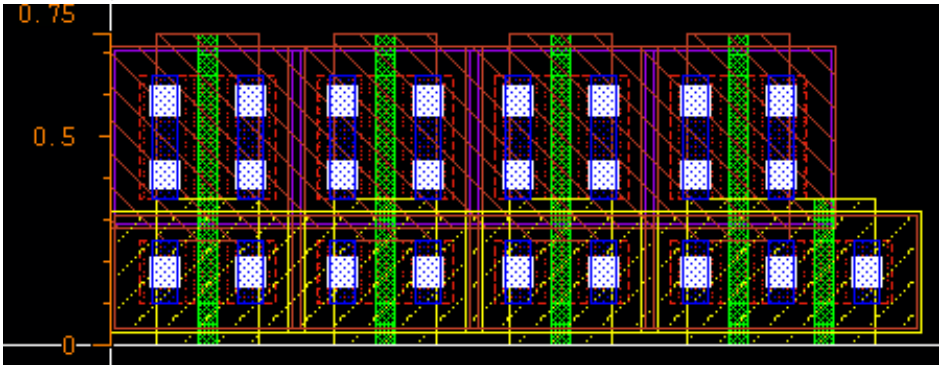
- Devices are not packed when both of the following conditions are true:
  - *Options – SPD – Placement – Auto pack in X* is not selected (automatic packing is disabled)
  - *Options – Display – Size for Medium Symbol – Length – Free* (free bucket size for length*)* is selected



- Devices in the Preview window are packed when either of the following conditions are true:
  - *Options – SPD – Placement* – Auto pack in X is selected (automatic packing is enabled)
  - *Options – SPD – Placement – Auto pack in X* check box is not selected, but free bucket size is not enabled for length

© 2023 Cadence Design Systems, Inc.

Generating the Layout



You can generate the layout from the symbolic design in two ways:

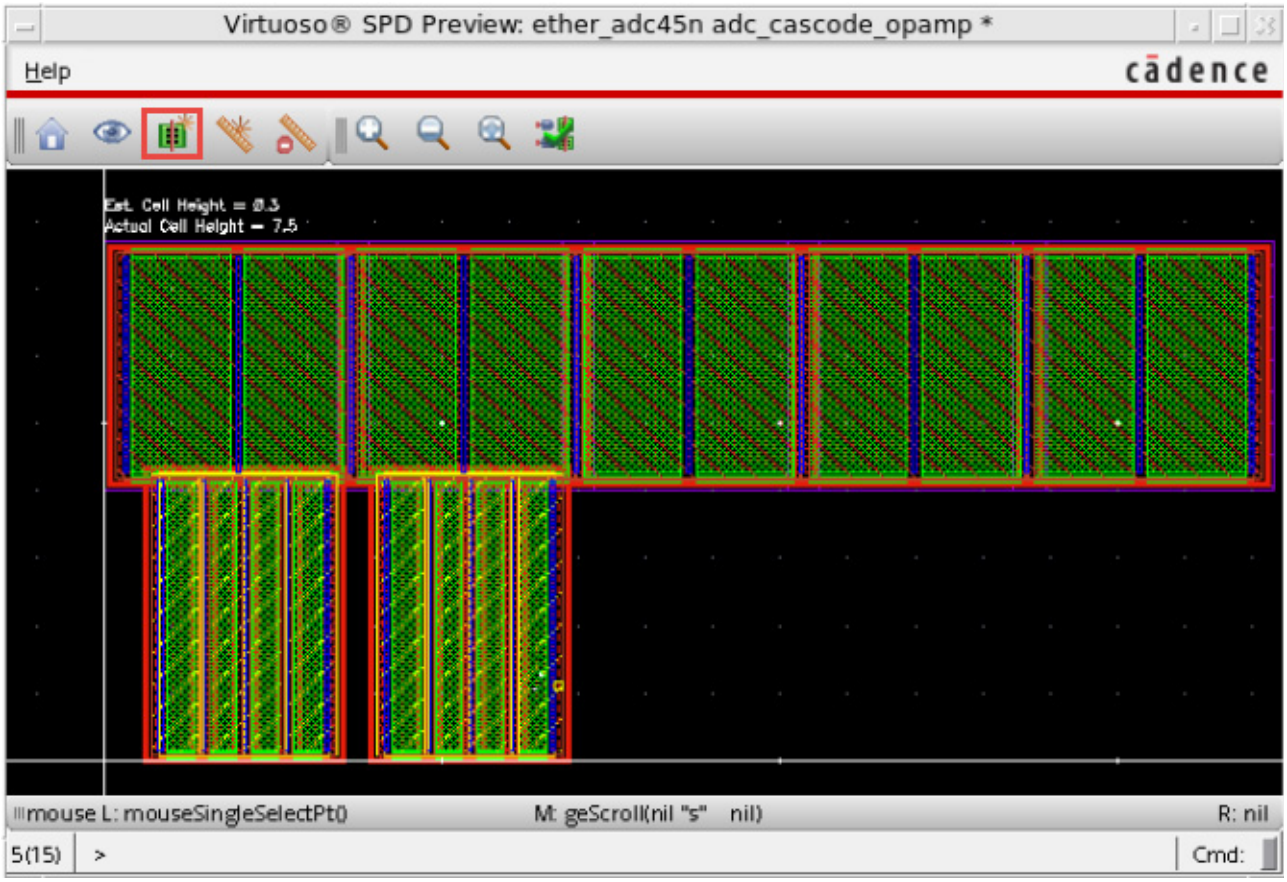
- Generate Layout from the Preview Window
- Generate Layout from the SPD Editing Window

Generate Layout from the Preview Window

To generate the layout from the preview window:

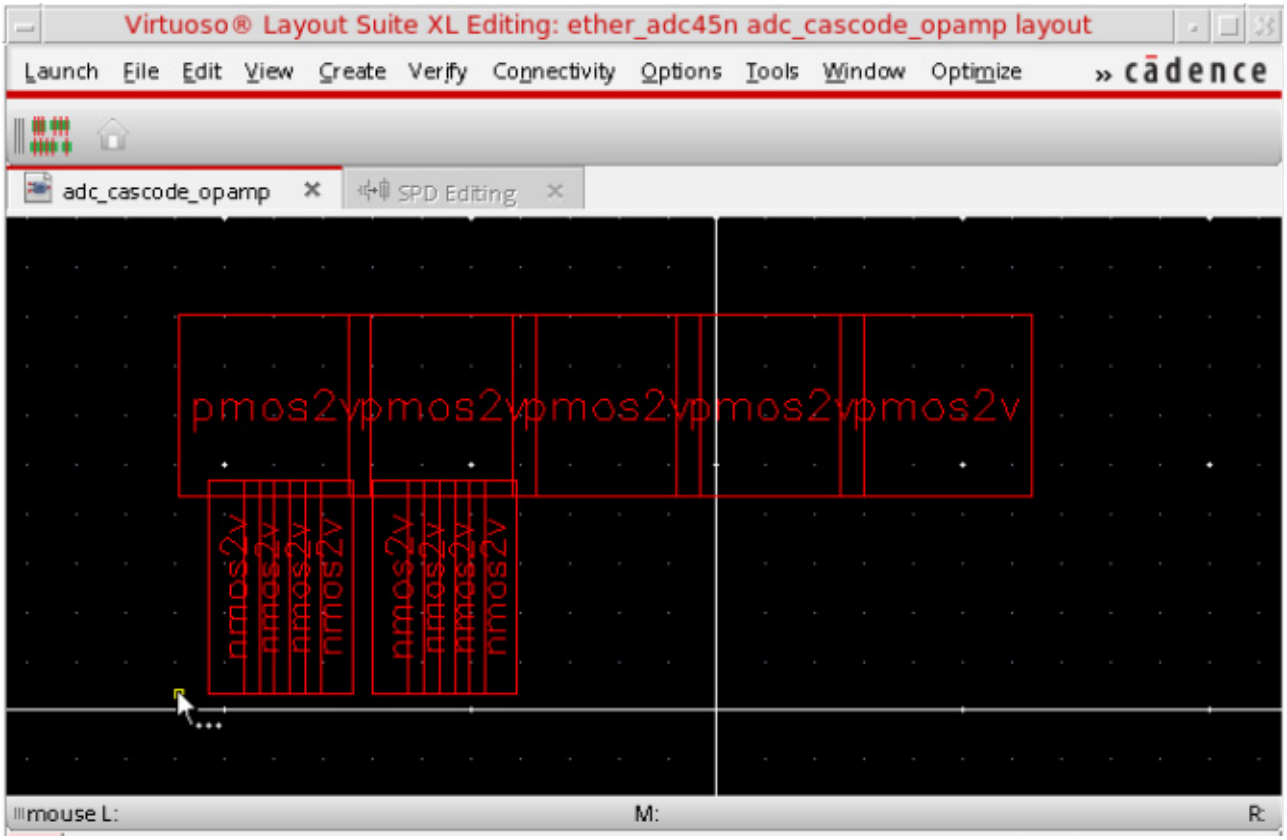
© 2023 Cadence Design Systems, Inc.

1. Select the *Generate Layout* [g] button on the SPD Preview toolbar.



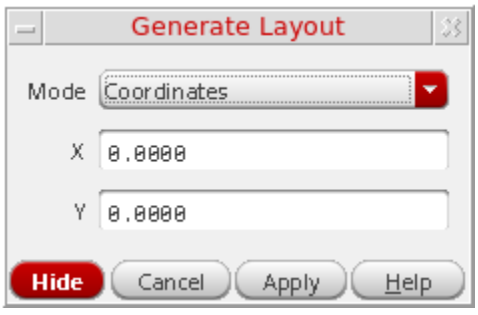
© 2023 Cadence Design Systems, Inc.

The focus changes to the Layout XL window.



2. Click the Layout XL window.

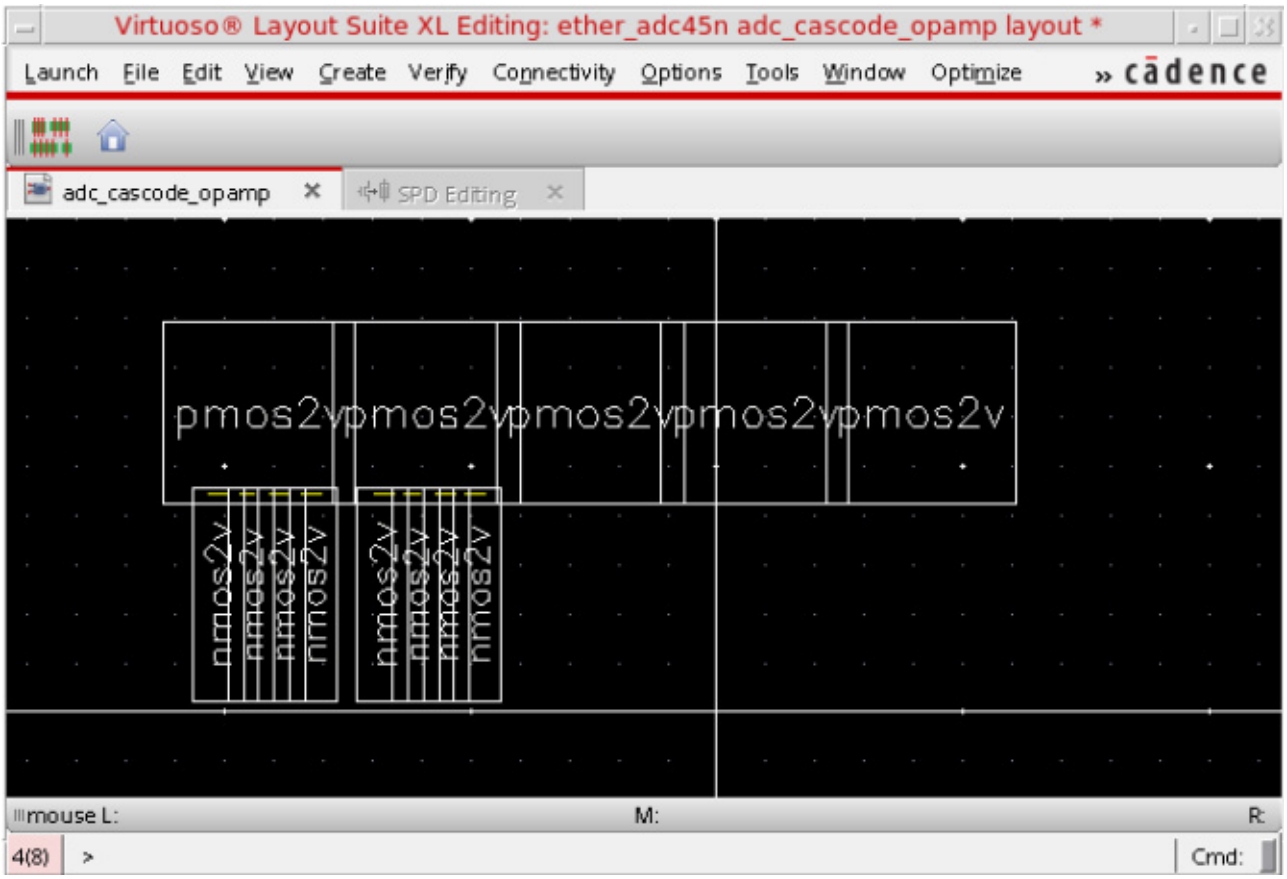
   You can press `F3` to display the Generate Layout form, where you can specify more precisely where the design should be placed in the Layout XL canvas.



3. Click in the canvas to point the location where you want to place the generated layout.

© 2023 Cadence Design Systems, Inc.

The generated layout is placed in the Layout XL canvas and the SPD Preview window is closed.



Generate Layout from the SPD Editing Window

To generate the layout from the SPD Editing window:

1. Choose *Place – Generate Layout* [g] or click the *Generate Layout* button on the SPD toolbar.

   The focus changes to the Layout XL window.

2. In the Layout XL window, point your mouse to the location where you want to place the generated layout.

   You can also press F3 to display the Generate Layout form, where you can specify more precisely where the design should be placed in the Layout XL canvas.

3. The design is placed at the specified location in the Layout XL window and the SPD window is closed.

© 2023 Cadence Design Systems, Inc.

Note: If you are not sure about your design, you can first preview and then generate the layout from the SPD Preview window. For more information, see Generating the Layout

© 2023 Cadence Design Systems, Inc.

© 2023 Cadence Design Systems, Inc.

A

Symbolic Placement of Devices Forms

This section lists the Virtuoso® Symbolic Placement of Devices (SPD) forms.

- Abut
- Align
- Append Selected from Source
- Check or Update
- Dummy Options
- Edit Stack
- Flip Horizontal
- Generate Chained Devices
- Generate Folded Devices
- Generate Layout
- Load SPD Design
- Move
- Probe & Align
- Save SPD Design
- SPD Options
  - Placement Options
  - Display Options
  - Generation Options
  - Power Rails Options
  - Labels Options

© 2023 Cadence Design Systems, Inc.

- Stack Options
- Setup Options

© 2023 Cadence Design Systems, Inc.

Abut

Use the **Abut** form to set the preferred direction for abutment when one chain is inserted into another chain.

Preferred direction lets you choose to abut the *left* or *right* half of the target chain by selecting the appropriate radio button.

Related Topics

Abutting and Unabutting Symbolic Devices

© 2023 Cadence Design Systems, Inc.

Align

Use the **Align** form to control the alignment of gates between two rows.

Show flight lines of gates specifies whether to display flight lines during alignment. This check box is selected by default.

Related Topics

Aligning Symbolic Devices

© 2023 Cadence Design Systems, Inc.

Append Selected from Source

Use the **Append Selected from Source** form to specify placement mode for the new devices being added to the SPD design from the source schematic.

Placement Mode specifies the mode for adding the devices.

Auto places the devices based on the current row pattern. This row pattern follows the pattern in layout and is detected when SPD is launched. You can change this pattern through SPD Options form.

Sort by Schematic X detects row pattern from schematic, and then sorts and places the devices from left to right.

Sort by Schematic Y detects row pattern from schematic, and then sorts and places the devices from bottom to top.

Related Topics

Appending Components from the Schematic

© 2023 Cadence Design Systems, Inc.

Check or Update

Use the **Check or Update** form to check the differences between the source and the SPD layout, or update existing objects in the SPD window.

Note: This form combines options on the Layout XL forms, Check Against Source and Update Components and Nets in one simplified form with only those options that are valid for SPD.

Check against source	checks the differences between the schematic and layout using the options available in the *Check* section. Use these options to choose what types of differences you want to see in the report and what format the report should take.
Update components and nets	automatically updates the SPD layout to take account of instances, pins, and connectivity you have changed in the schematic using the options available in the *Update* section.
Check	
Unbound instances	reports SPD instances that are not bound to schematic instances.
Connectivity	reports connectivity issues on top level pins and global nets; mismatched or missing terminals and instance terminals; and unbound nets in SPD.
Parameters	reports issues with CDF parameters. You can specify how the parameters are compared on the Parameters tab of the *Connectivity* form.
Display info window	

prints the report in a separate Info window, otherwise the report is printed in the CIW.

Update

Selected layout components only updates only the instances and pins currently selected in the layout window.

Layout parameters automatically updates the parameters and parameter values on SPD instances to match those on their schematic counterparts. Parameters that are set in SPD instances but are not present on their schematic counterparts are not removed.

© 2023 Cadence Design Systems, Inc.

Related Topics

Checking Source or Updating Layout

© 2023 Cadence Design Systems, Inc.

Dummy Options

Use the **Dummy Options** form to configure how dummy instances are added to the symbolic design.

Net specifies the net name to which you want the dummy device to be tied. All terminals of a dummy instance, except its bulk terminal, are connected to the tie net.
Environment variable: spdDummyNet

Dummy Parameters specifies the number of fingers and length and width of the dummy.

Number of Fingers sets the number of fingers for the dummy.
Environment variable: spdDummyNumFingersOptions

CDF Default sets the number of fingers to be the same as the default CDF parameter values. This is the default setting.

Same as Neighbor sets number of fingers to be the same as the neighboring instances.

Specify lets you specify the number of fingers manually in the text field that is displayed when you select this option.
Environment variable: spdDummyNumFingersValue

Length specifies the length for the dummy.
Environment variable: spdDummyLengthOptions

CDF Default sets the length to be the same as the default CDF parameter values. This is the default setting.

Same as Neighbor sets the length to be the same as the neighboring instance.

Specify lets you specify the length manually in the text field that is displayed when you select this option.
Environment variable: spdDummyLengthValue

Width specifies the width for the dummy.
Environment variable: spdDummyWidthOptions

CDF Default sets the width to be the same as the default CDF parameter values.

Same as Neighbor sets the width to be the same as the neighboring instance. This is the default setting.

Specify lets you specify the width manually in the text field that is displayed when you select this option.
Environment variable: spdDummyWidthValue

© 2023 Cadence Design Systems, Inc.

Remember Values specifies that the values set on the Dummy Options form are saved and remembered the next time the form is opened. This option is not selected by default.
Environment variable: spdRememberDummyVals

Related Topics

Adding Dummy Devices

© 2023 Cadence Design Systems, Inc.

Edit Stack

Changes the placement of stacked instances.

Field	Description
Stack	Lists the instances in the selected stack. In the list, instance at top is the bottom instance of the stack. You can move the instances up or down in the stack using the arrow buttons on the left. You can use the arrow buttons on the right to move instances between the *Stack* and *Unstack* boxes.
Unstack	Lists the instances that have been removed from the selected stack.
Spacing	Separates the instances in the stack by the distance specified in the text field when the layout is generated.
Number of Stacks	Specifies the number of stacks to create.

Related Topics

Stacking Symbolic Devices

© 2023 Cadence Design Systems, Inc.

Flip Horizontal

Use the **Flip Horizontal** form to specify whether to use pin permutation when flipping devices.

Permute lets you flip instances by pin permutation. By default, this check box is not selected.

For more information on how pin permutation works, see permuteRule in the *Virtuoso Layout Suite XL User Guide*.

Related Topics

Flipping Symbolic Devices

© 2023 Cadence Design Systems, Inc.

Generate Chained Devices

Use the **Generate Chained Devices** form to generate a device chain when more than one I/O pin-connected PMOS or NMOS devices are selected.

- The **PMOS** box lists the PMOS devices selected in the symbolic design to generate a device chain.
- The **NMOS** box lists the NMOS devices selected in the symbolic design to generate a device chain.

Select one of the devices in each list to specify the locations at which the respective device chains are to be placed.

Related Topics

Generating Chained Devices

© 2023 Cadence Design Systems, Inc.

Generate Folded Devices

Use the **Generate Folded Devices** form to control interactive folding, which lets you divide an individual transistor or a chain of transistors into two or more folds. Folding lets you change the aspect ratios of the devices in your design while retaining the original orientation. The devices to be folded need not be bound to a schematic instance.

> **Note:** The Generate Folded Devices form also lets you divide an individual FinFET device or a chain of devices into two or more folds.

Transistor Name shows the name of the transistor to be folded. If you have selected more than one transistor, use the *Next* and *Previous* buttons to move through the selected set.

Transistor Width shows the value of the width property of the selected device, indicating also whether the value is taken from the schematic or layout view. This option is available when the selected devices for folding are MOS transistors that have the *Width parameter* set.

Transistor Fins shows the value of the *Number of fins* parameter of the selected FinFET device, also indicating whether the value is taken from the schematic or the layout view.

> **Note:** The *Transistor Fins* field is displayed only for FinFET devices that have the *Number of fins* parameter set. For FinFET devices that have the *Width parameter* set, the form displays *Transistor Width*.

Number of Folds lets you type in the number of folds for the selected device. Enter a value greater than one in this field and then click the *Show Folds* button, which evenly distributes the transistor width across the available folds.

> **Note:** Such a distribution is also applicable for FinFET devices that are set for folding based on the *Width parameter.*

Fold Threshold specifies the maximum size of a folded transistor. Enter a value and then, click *Show Folds* to split the transistor width into folds according to the specified threshold.

Chain Folds tells SPD to chain the resultant folds together where possible.

Hide Folds and **Show Folds** buttons let you hide or display the section that shows how the width or fins have been split across all folds.

Total Width shows the total value of all the fold widths displayed in the *Width* fields at the bottom of the form.

> **Add Fold** adds an additional fold to the device and provides a field where you can specify the *Width* for the new fold.

© 2023 Cadence Design Systems, Inc.

Distribute evenly divides the total width of the transistor by the number of folds and sets the resultant *Width* as the value for each fold.

Delete removes the corresponding fold from the device.

The system automatically adds the *Transistor Width* unit identifier to any width value with no unit. If the *Transistor Width* has no unit identifier, no identifier is added to the *Width* fields. See Unit Identifiers, in the *Virtuoso Layout Suite XL User Guide* for a complete list of identifiers.

Related Topics

Folding Symbolic Devices

© 2023 Cadence Design Systems, Inc.

Generate Layout

Use the **Generate Layout** form to generate a physical layout of the symbolic design to a target Layout XL window.

Mode specifies the method used to generate the layout.

Coordinates places the origin of the layout in the SPD Preview window at the absolute coordinates of the target layout. You specify the absolute coordinates in the *X* and *Y* fields.

Delta to lower-left origin places the origin of the lower-left instance of the generated layout to the target Layout XL window at the origin where it was placed before it was updated in SPD. The origin is offset by values specified in the X and Y fields.

Delta to mouse lets you specify the origin of the lower-left instance of the generated layout by clicking in the canvas. The origin is offset by values specified in the X and Y fields.

© 2023 Cadence Design Systems, Inc.

Load SPD Design

Use the **Load SPD Design** form to load a previously saved SPD design. The selection box at the top of the form lets you select the version of the saved design you want to open. A preview of the selected design is shown in the bottom section.

Related Topics

Loading a Design

© 2023 Cadence Design Systems, Inc.

Move

Use the **Move** form to control how the move operation is performed, including the angles at which you can move the object, the presence of draglines on the object as you move it, and the orientation of the object you select.

For information about the Move options, see the Move Form documentation in the Virtuoso Layout Suite L User Guide.

The *Change To Layer* option, all snapping options, and the actions rotate left and rotate right are not supported by SPD.

Related Topics

Moving Symbolic Devices

© 2023 Cadence Design Systems, Inc.

Probe & Align

Use the **Probe & Align** form to find and highlight logical connectivity between selected nets.

The **Show** dropdown list lets you list *All Nets* or *Pin Nets* in the list box given below. You can choose one or more nets to highlight their logical connectivity in the design.

Sort specifies whether nets are listed in *Ascending* or *Descending* order in the list box.

Aligns the gates of the left most instances of the selected net in each row.

Removes all alignments from the symbolic design.

Zooms to the instances belonging to the selected nets.

Selects the instances belonging to the selected nets.

Related Topics

Probing Selected Nets

© 2023 Cadence Design Systems, Inc.

Save SPD Design

Use the **Save SPD Design** form to save different versions of your SPD design.

Version specifies the version number for the saved design. This version and the date on which it was modified is displayed in the information box located in the upper part of the form.

Delete removes the version specified in the *Version* field.

Related Topics

Saving an SPD Design

© 2023 Cadence Design Systems, Inc.

SPD Options

The SPD Options form comprises a tree that lets you access the following option panes to further customize the symbolic layout of devices:

- Placement Options
- Display Options
- Generation Options
- Power Rails Options
- Labels Options
- Stack Options
- Setup Options

Placement Options

Use the **Placement** pane to configure chaining and folding options for the selected instances, and to specify row patterns and alignment for these instances.

The **Chaining/Folding** section lets you optimize the layout by abutting selected MOS transistors into chains, folded devices, or folded chained devices.

Chain automatically abuts the selected set of MOS transistors to form a chain. The transistors to be abutted must be set up for abutment first. For more information, see Setting Up Pcells for Abutment in *Virtuoso Layout Suite XL User Guide*.

Environment variable: initDoChaining

Fold automatically divides MOS transistors into folds to optimize utilization.

Environment variable: initDoFolding

Chain Folds automatically chains the individual folds of a transistor.

Note: For the Chain Folds option to be available, only the Fold check box must be selected. If you select the Chain check box as well, Chain Folds is disabled.

Interdigitate Chains automatically identifies nodes that qualify as pseudoparallel connections and defines them accordingly during chaining and abutment. A pseudoparallel net connects nodes that are always the same voltage, so current does not pass through the net.

NMOS Fold Threshold is the maximum size of the folded NMOS transistor.

© 2023 Cadence Design Systems, Inc.

Note: The NMOS Fold Threshold option is available only when the Fold check box is selected.

Environment variable: foldingThresholdNType

PMOS Fold Threshold is the maximum size of the folded PMOS transistor.

Note: The PMOS Fold Threshold option is available only when the Fold check box is selected.

Environment variable: foldingThresholdPType

Placement section lets you specify how PMOS and NMOS devices are placed in the design.

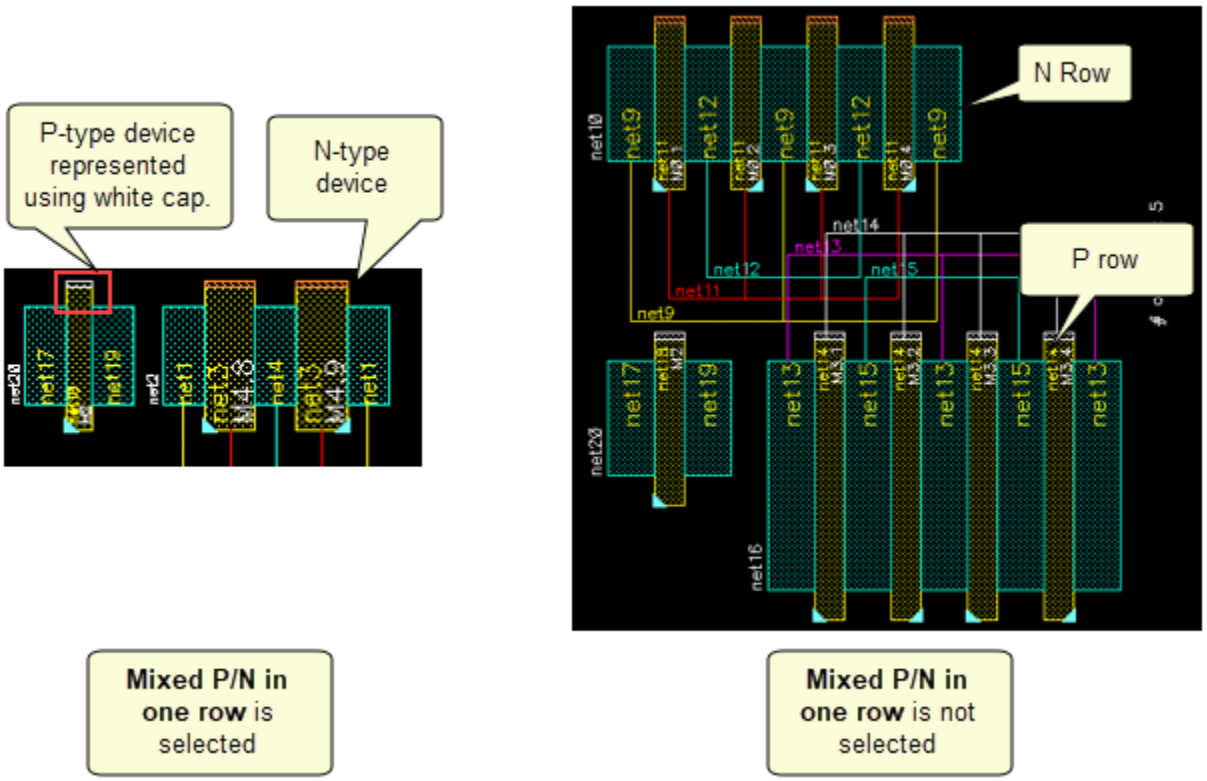
Row Pattern sets the row order for placement of devices. SPD supports the following two-row and multiple row device placement patterns: NP, PN, NPPN, PNNP, NPNP, PNPN, NNPP, PPNN, NNNP, PNNN, NPPP, PPPN, NNNN, PPPP, and (Arbitrary). The default is NP.

Row Number lets you specify the number of rows in the text field. This option is available when *Row Pattern* is set to *(Arbitrary)*. The default is `10`.

When this setting is used, PMOS devices are treated as NMOS devices and are represented with a white cap at the top.

© 2023 Cadence Design Systems, Inc.

Mix P//N in one row specifies whether both P and N type devices can be specified in a single row.



Environment variable: PNmixedRow

When you specify the row pattern as *(Arbitrary)* and specify four or more rows, the specified number of rows is used for the selected instances. However, if you have mixed P-type and N-type devices in a row and the *Mix P/N in one row* option is not selected then a dialog box is displayed that lets you either enable the *Mix P/N in one row* option or reset the placement.

Important

When you choose row pattern as *(Arbitrary)*, the configuration is applied after you restart SPD.

Chain Alignment specifies how the instances in the chain are aligned. Choose one of the following:

Inside aligns instances to that side of a P or N row that is close to the channel area.

Outside aligns instances to that side of a P or N row that is far from the channel area.

© 2023 Cadence Design Systems, Inc.

Center aligns instances to the center of a P or N row.

Align to Gates sets the side of the gate to which the selected instance are aligned. You choose between *Left*, *Center*, and *Right*. The default is *Center*.

Auto gate alignment specifies whether to try to align the first gate of a chain with another gate even if no alignment has been specified in the symbolic design. This check box is selected by default.
Environment variable: autoAlignGates

For more information, see Aligning Symbolic Devices.

Priority specifies the placement for the priority devices in the chain.
Environment variable: placementPriority

Maximize OD Sharing optimizes the generated chains by maximizing oxide diffusion sharing.

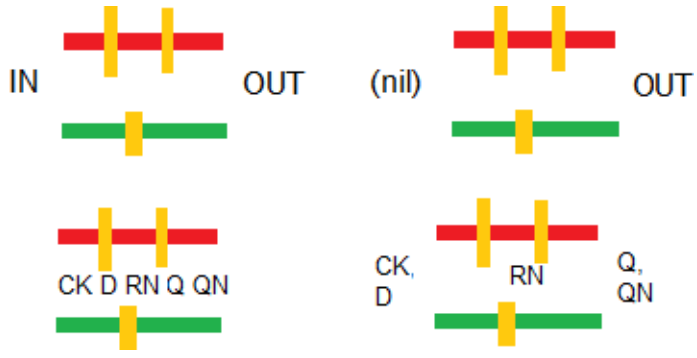
Minimize number of tracks optimizes the generated chains by minimizing the number of tracks in the chained devices.

Auto automatically sets the placement priority. This is the default.

Pin Driven enables pin driven placement and optimizes the device placement according to the pin order you specified. By default, this check box is not selected. Pin driven placement can be used only for NP and PN row patterns. The option is disabled when you choose any other row pattern.
Environment variable: pinDriven

Order lets you manually specify the pin order from left to right for the pin driven placement. The figure below shows different ways in which pin order can be specified.

- IN OUT
- / OUT
- CK D RN Q QN
- (CK D) RN (Q QN)



To specify pin order:

- ❑ Specify pin order from left to right.
- ❑ Separate pin names by spaces.

© 2023 Cadence Design Systems, Inc.

- The two ends of the pin order are assumed to be two edges of the placement. Therefore, the left most pin is assumed to be on the left edge of the placement and the right most pin is assumed to be on the right edge of the placement. In the above example:
  - For pin order `IN OUT`, `IN` is on the left edge and `OUT` is on the right edge.
  - For pin order `CK D RN Q QN`, `CK` is on the left edge and `QN` is on the right edge.
- `/` denotes an unspecified pin for one of the two ends. This works when there are only two pins in the pin order.
- `()` groups multiple pins. Pins in the same group are at the same position. In the example `(CK D) RN (Q QN)`, Pins `CK` and `D` and pins `Q` and `QN` are at the same position.

Note: Although it is not mandatory to perform pin driven placement with an empty pin order list, it is recommended that you always specify pin order when you enable pin driven placement to have a meaningful result.
Environment variable: pdPins

Abut same width only abuts devices of same width. When this option is set all commands that perform automatic abutment, such as *Reset Placement*, *Abut*, *Abut All*, and *Move*, will abut only those devices that have same width.

Auto flip chain in 'Abut All' command automatically flips the chain when you select the *Abut All* command.
Environment variable: abutAllAutoFlipChain

Move to insert in one location moves selected instances to the specified location in the design. By default, this check box is selected.
Environment variable: smartMoveInsertOnePos

Auto pack in X direction automatically packs devices in horizontal direction when commands Move, Undo, Redo, Align, Save, and Load are used to change the position of devices. When this option is deselected, automatic packing does not take place.
Environment variable: autoPackInX

Related Topics

Setting Up Pcells for Abutment

Aligning Symbolic Devices

© 2023 Cadence Design Systems, Inc.

Display Options

Use the **Display** options pane to specify how labels and flight lines are displayed in the design. You can also specify the dimensions considered to be medium-sized devices. The sizes for small and large devices are also derived from these values.

Label Display specifies what information is included in the labels for abstract devices. You can choose from:

- *All*
- *All except Device Size*
- *Gate*
- *Instance and Well Net Names*
- G*ate and Instance Net Names*
- *Gate Net Name Only*

The default value is *All except Device Size*.
Environment variable: labelDisplay

Show flight lines specifies that flight lines are displayed in the design. By default, this check box is selected.

Connect to sets the connectivity mode of flight lines. This option is available only when the *Show flight lines option* is selected.
Environment variable: flightLineMode

Gate displays flight lines for connections to gates.

Active displays flight lines for connections to the active region.

Both displays flight lines for connections to both gates and the active region. This is the default.

Connect between specifies the type of flight lines to display in the symbolic design. This option is available only when the *Show flight lines* option is selected and a multiple-row pattern, such as *NPPN* or *PNNP* is set for the *Row Pattern* option in the *Placement* options pane.
Environment variable: flightLineConn

1 row displays flight lines for connections within 1PN rows.

2 rows displays flight lines for connections between 2PN rows.

Both displays flight lines of both types. This is the default.

© 2023 Cadence Design Systems, Inc.

Show P/G Nets specifies how to display power and ground nets in the design.

None disables the display of power and ground flight lines.
Environment variable: showPGFlightLines

Separate displays power and ground (P/G) nets separate from signal nets at the bottom and the top of a PN row. When this option is selected, SPD still creates the P/G nets in channel. You need to specify the names of these nets in the *Connectivity* form or through Layout XL environment variables `lxGroundNetNames` and `lxSupplyNetNames`.

Mixed with signals mixes power and ground nets with signal nets in the channel area.
Environment variable: mixPGFlightLinesSignals

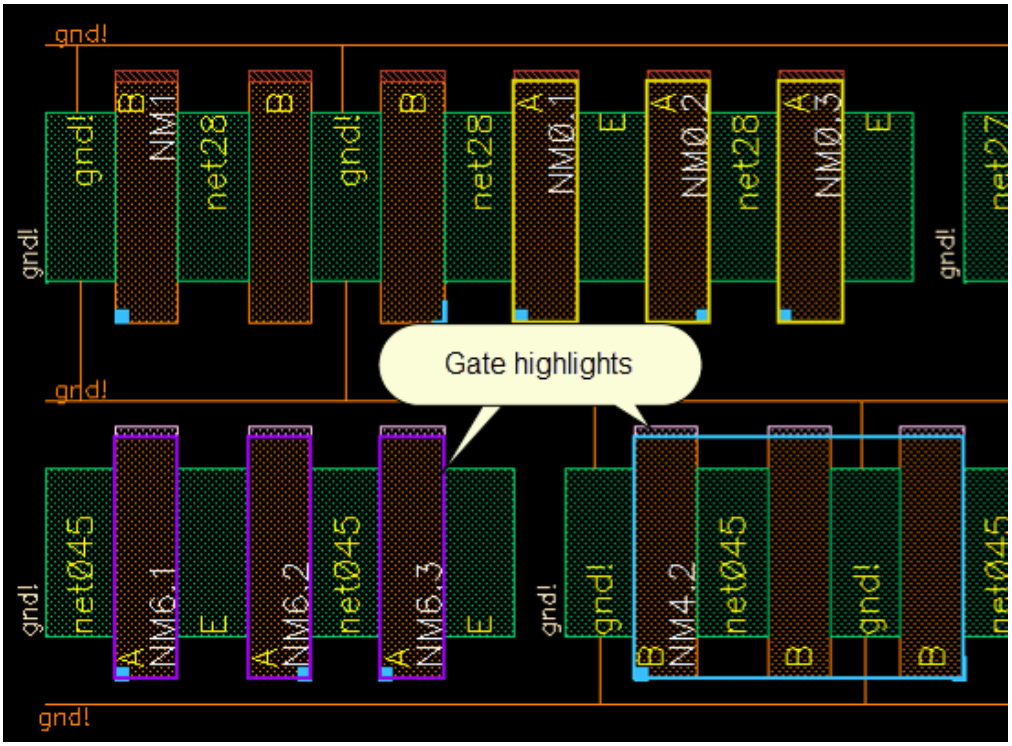
Note: SPD detects the P/G nets through Layout XL environment variables, the signal type of OA, and the nets of the bulk.

Highlight P/N device patterns on gates highlights different device patterns, such as m-factors, folds, and finger-splits in different colors in the design. This option can highlight seven patterns in the design. Gray color is reserved to highlight the indistinguishable patterns. The gate highlights are reanalyzed and updated in the canvas when:

- the *Highlight P/N device patterns on gates* option is enabled or disabled.
- new devices are added to the design using the *Append selected from source* option.
- design is reset using the *Reset placement* option**.**
- design is reopened using the *Load SPD design* option.
- edit commands– *Undo*, *Redo*, *Fold*, *Split Fingered Device*, *Remove*, *Check or Update*, *Stack*. *Unstack*, and *Edit Stack* are used.

© 2023 Cadence Design Systems, Inc.

- the *Free Bucket Size* option is enabled or disabled.



Environment variable: highlightGatePattern

Size for Medium Symbol sets the size for the medium symbol in the design. All other symbols, smaller and bigger, are calculated with respect to the size of the medium symbol set here.

Min Width sets the minimum width for the medium symbol.
Valid values: 0 to infinity
Environment variable: mediumSymbolMaxWidth

Max Width sets the maximum width for the medium symbol.
Valid values: 0 to infinity.
Environment variable: mediumSymbolMinWidth

Auto button for the *Width* option automatically calculates the minimum and maximum widths of the medium symbol.

Free specifies whether free bucket size mode should be used to display widths of symbolic devices.
Environment variable: freeSymbolWidth

© 2023 Cadence Design Systems, Inc.

Min Length sets the minimum length for the medium symbol.
Valid values: 0 to infinity
Environment variable: mediumSymbolMinLength

Max Length sets the maximum length for the medium symbol. Valid values: 0 to infinity.
Environment variable: mediumSymbolMaxLength

Auto button for the *Length* option automatically calculates the min and max lengths of the medium symbol.

Free specifies whether free bucket size mode should be used to display lengths of symbolic devices.
Environment variable: freeSymbolLength

If there are FinFET devices, driven by number of fins are included in the design, the *Size for Medium Symbol* section form displays *Fins* instead of *Width*. You can use this option to set minimum and maximum number of fins for the medium symbol.

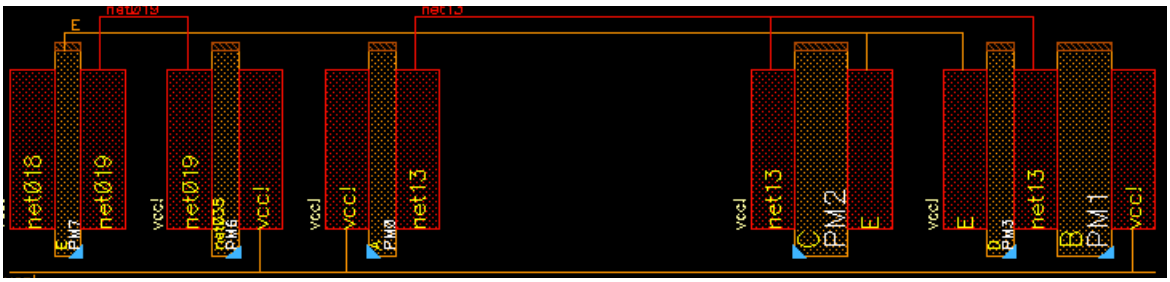
Environment variables: mediumSymbolMinFins and mediumSymbolMaxFins

Displaying Free Bucket Size of Devices Based on Whether Horizontal Packing is Enabled

When the *Options – SPD – Placement – Auto pack in X direction* check box is selected, devices are packed when you choose to display free bucket size of devices. However, if this check box is deselected automatic packing of devices in the horizontal direction does not take place when free bucket size is displayed. This is shown below.

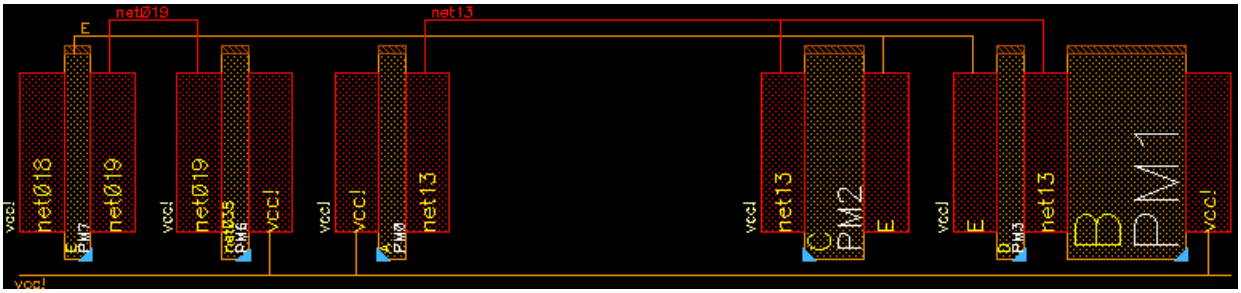
- Devices displayed based on relative sizes. Here, PM2 is 100nm and PM1 is 200nm.

.

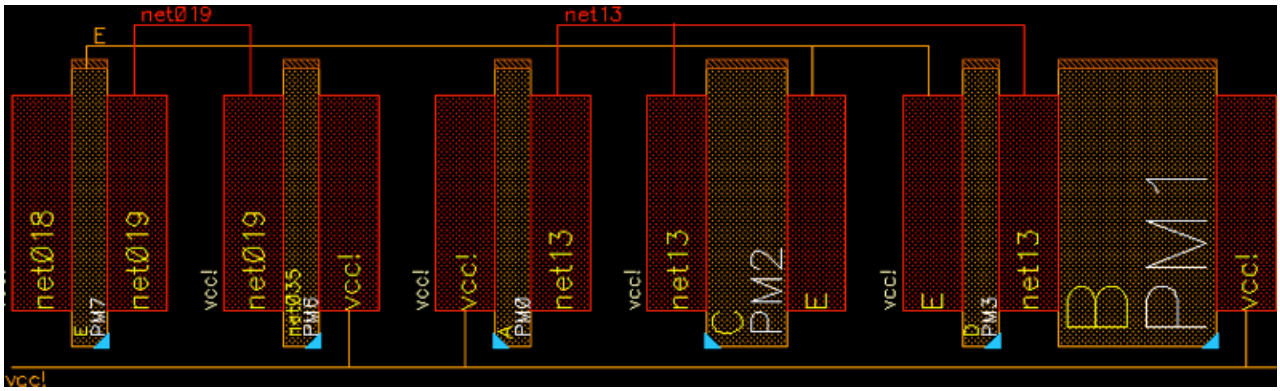


© 2023 Cadence Design Systems, Inc.

- Devices displayed when free bucket size is enabled, but automatic packing of devices is disabled. You can see that PM1 is bigger than PM2 and devices are not packed in free bucket size mode.



- Devices displayed when both free bucket size and automatic packing of devices are enabled. You can see that PM1 is bigger than PM2 and devices are packed in free bucket size mode.



Generation Options

Use the **Generation** options pane to configure spacing for a P-type oxide diffusion (POD) and an N-type oxide diffusion (NOD). You can also set the heights for PMOS and NMOS devices.

The **Spacing** section lets you set the spacing between power and ground rails with respect to POD and NOD and the spacing for between and within row-pairs. Counting from the bottom, the first and second rows are defined as a row-pair and so are the third and the fourth rows.

Power-POD specifies the spacing between a power rail and a POD.
Environment variable: spacingVddPod

© 2023 Cadence Design Systems, Inc.

Between Row-pair specifies spacing between two row pairs when generating the layout.
Environment variable: betweenRowPair

Within Row-pair specifies the spacing within a pair of rows.
Environment variable: spacingPodNod

Ground-NOD specifies the spacing between a ground rail and an NOD.
Environment variable: spacingVssNod

X: OD drop-down list specifies how oxide diffusion spacing is decided. The default value is *Fixed*.

Environment variable: spacingOdMode

- **Fixed** specifies the oxide diffusion spacing between chains. Type a floating number in the field under the *X:OD* drop-down list.
  Environment variable: spacingOd
- **DummyPoly** specifies that the software automatically calculates the required oxide diffusion spacing between two unabutted instances. The software considers the minimum oxide diffusion spacing and the minimum poly spacing defined in the technology file and then calculates a value that satisfies both. This becomes the default setting for this option when there are FinFET devices in the design.
  Environment variable: spacingOd
- **UserFunction** specifies a user callback SKILL function that calculates the required oxide diffusion spacing between two unabutted instances. Enter the SKILL function name in the field under the *X:OD* drop-down list.
  Environment variable: spacingOdUserFunc

For more information of dummy poly SKILL functions, see Symbolic Placement of Devices Functions in *Virtuoso Layout Suite SKILL Reference*.

Height sets the height of the PMOS and NMOS regions.

PMOS specifies the height for the PMOS region.
Environment variable: heightPmosRegion

PFIN specifies the height for the PFIN region. This option is available instead of *PMOS*, if there are FinFet devices in the design.

Environment variable: heightPmosRegion

NMOS specifies the height for the NMOS region.
Environment variable: heightNmosRegion

© 2023 Cadence Design Systems, Inc.

NFIN specifies the height for the NFIN region. This option is available instead of *NMOS*, if there are FinFet devices in the design.

Environment variable: heightNmosRegion

Prevent gate shorts specifies whether to prevent gate shorts across all rows when generating the layout.
Environment variable: preventGateShorts

Prevent well shorts specifies whether to prevent well shorts when generating the layout.
Environment variable: preventWellShorts

Cell (Est) displays the estimated cell height. This value is updated automatically when any of the above listed values are updated in this pane.

Click Check Fold Threshold to display the *Placement* pane where you can enable folding and make the fold threshold less than MOS height. This button is enabled only when either of the following is true:

- MOS height is greater than 0 but *Fold* is not enabled in the *Placement* pane
- Fold threshold specified in the *Placement* pane is larger than MOS Height.

 *Tip*

You can reset placement and split multi-finger devices that exceed MOS height.

The preview box on the right lets you see how spacing and height changes with each option you modify.

Power Rails Options

The **Power Rails** section helps you customize how power and ground rails are created in the design. Power rails can only be created for NP, PN, PNNP, and NPPN row patterns.

Create specifies whether power rails are created.
Environment variable: createPowerGroundRails

As trunk creates power rails as trunks. When selected, power rails are created as path segments and are converted to trunks. This option is available only when the Create option is selected.
Environment variable: createPGTrunk

Truncate creates power rails with no beginning and ending extension values.

© 2023 Cadence Design Systems, Inc.

Extend creates power rails with the default beginning and ending extension values.
Environment variable: pgTrunkBeginEndType

Note: These options are available only when the *Create* and *As trunk* options are selected.

Delete existing rails specifies whether existing power rails are to be deleted when generating the layout. By default power rails are not deleted when the layout is generated.
Environment variable: deleteExistingRails

Power sets the layer, width, and net name for the power rails.

Layer specifies the layer purpose used to create the power rail for the power net.
Environment variable: layerVdd

Width specifies the width for the power rail.
Environment variable: widthVdd

Net Name specifies the net name for the power rail.
Environment variable: vddNetName

Auto Detect automatically detects the net name. This check box is selected by default. In auto mode, power nets are obtained in the following order:

1. Power nets from the instances in the design.
2. The *Power net names* field in the Connectivity form or set using the `lxSupplyNetNames` environment variable.
3. Default name, which is `VDD`.

Ground sets the layer, width and net name for the ground rails.

Layer specifies the layer purpose used to create the power rail for the ground net.
Environment variable: layerVss

Width specifies width for the ground rail.
Environment variable: widthVss

Net Name specifies the net name for the ground rail.
Environment variable: vssNetName

Auto Detect automatically detects the net name. This check box is selected by default. In auto mode, ground nets are obtained in the following order:

1. Ground nets from the instances in the design.

© 2023 Cadence Design Systems, Inc.

2. The *Ground net names* field in the Connectivity form or set using the `lxSupplyNetNames` environment variable.

3. Default name, which is `VSS`.

The **Signal Trunks** section helps you specify whether signal trunks are created when the layout is generated. You can also specify the layer purpose, width and spacing for the signal trunks.

Create specifies whether signal trunks are created
Environment variable: createSigTrunk

Layer specifies the layer purpose used to create the signal nets when generating the layout.
Environment variable: sigTrunkLr

Width specifies width for the signal net when generating the layout.
Environment variable: sigTrunkWidth

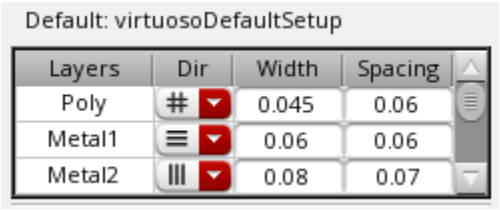
Trunk-Trunk specifies the trunk-to-trunk spacing when generating the layout.
Environment variable: sigTrunkSpacingT2T

Trunk-OD specifies the trunk-to-oxide-diffusion spacing when generating the layout.
Environment variable: sigTrunkSpacingT2O

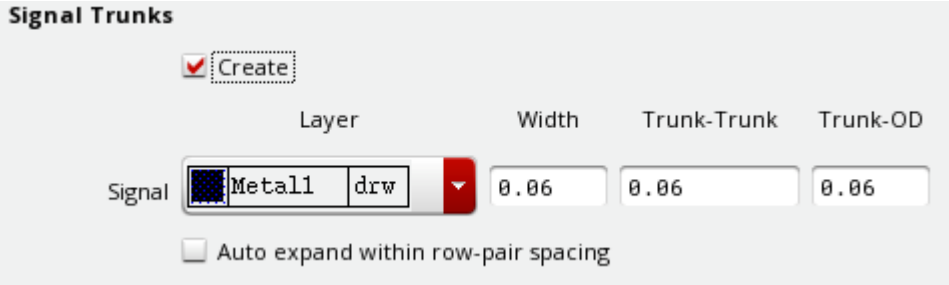
Auto expand within row pair spacing specifies whether the maximum number of tracks need to be calculated from trunk width and spacing information including within-row-pair spacing, trunk-to-OD spacing and trunk-to-trunk spacing.
Environment variable: expandSpacingForTrunks

© 2023 Cadence Design Systems, Inc.

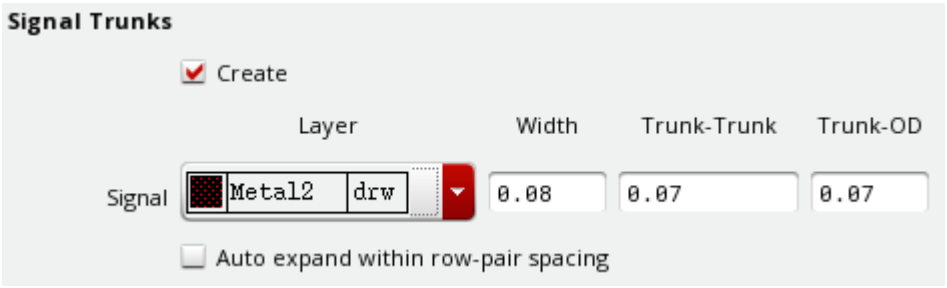
If you do not change the spacing and width values here, then these values are updated with the spacing and width of the corresponding layer, as shown below.



Default Layer Values



Values for layer `Metal1` reflected in SPD Options form



Values for layer `Metal2` reflected in SPD Options form

Labels Options

Use the **Labels** options pane to create and customize labels for the specified object type, instances, and power and ground rails.

Create labels lets you create labels in the design. By default, is not selected. All other options on this pane available only when you select the *Create labels* check box.
Environment variable: genLabel

Font specifies the font to be used for the labels. The default font is *roman*.
Environment variable: genLabelFont

© 2023 Cadence Design Systems, Inc.

Objects Type specifies the objects for which the labels need to be created. The options available in the drop-down list are *MOS*, *Power/Ground Rails*, and *Both*. The default value is *MOS*.
Environment variable: genLabelType

For Instance customizes instance labels. These options are available only if you specify the *Objects Type* as *MOS* or *Both*.

Layer specifies the layer-purpose pair for the instance labels.
Environment variable: genLabelLr

Orientation sets the orientation for the instance labels.
Environment variable: genLabelOri

Value	Meaning
Auto	Same orientation as the instance. This is the default value.
R0	No rotation
R90	Rotate 90 degrees clockwise
R180	Rotate 180 degrees clockwise
R270	Rotate 270 degrees clockwise
MY	Mirror about the Y axis
MYR90	Mirror about the Y axis and rotate 90 degrees clockwise
MX	Mirror about the X axis
MXR90	Mirror about the X axis and rotate 90 degrees clockwise

Justification sets the location of the label origin. The origin appears as a small square on the label when you place or select it. Choose from *Auto*, *lowerLeft*, *centerLeft*, *upperLeft*, *lowerCenter*, *centerCenter*, *upperCenter*, *lowerRight*, *centerRight*, and *upperRight*. By default this option is set to *Auto*, which sets justification as *centerCenter*.
Environment variable: genLabelJust

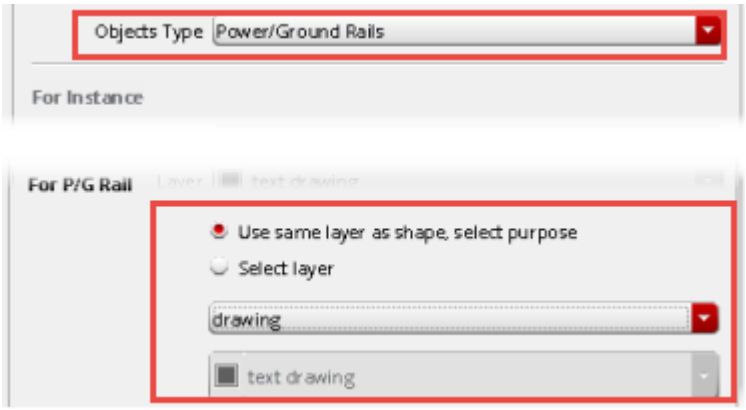
Height specifies the height of the instance label. You can choose *Auto* or *Fixed* from the drop-down list. By default, this option is set to *Auto*.
Environment variable: genLabelHeight

The entry field lets you manually specify the value of height for the instance labels. The default value is 0.1. This option is available only if you choose *Fixed* from the *Height*

© 2023 Cadence Design Systems, Inc.

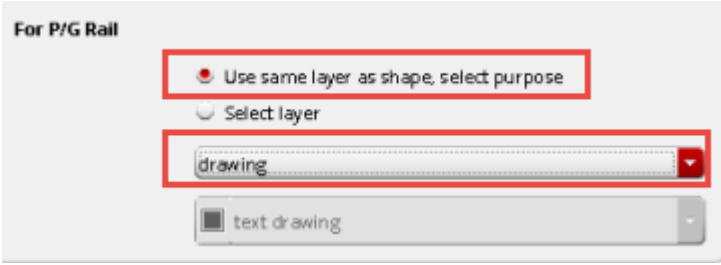
drop-down list.
Environment variable: genLabelHeightValue

For P/G Rail customizes labels for power and ground rails. These options are available only if you choose the *Objects Type* in the *Labels* section as *Power/Ground Rails* or *Both*.
Environment variable: genLabelPGLrRatio



If you select the **Use same layer as shape, select purpose** radio button, the layer for the label is same as the shape created as the power or ground rail, which can be specified in the Power Rails Options.

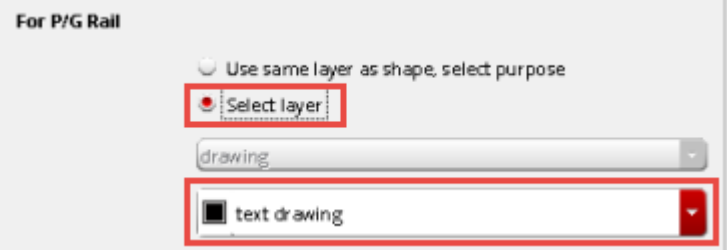
You must select the purpose from the first drop-down list included in the *For P/G Rail* section.
Environment variable: genLabelPGPur



If you choose the **Select layer** radio button, you must specify both layer and purpose from the second drop-down list included in the *For P/G Rail* section.

© 2023 Cadence Design Systems, Inc.

Use the layer purpose drop-down list to specify the layer used to create the power rail labels.
Environment variable: genLabelPGLr



Stack Options

Use the **Stack** options pane to specify how you want to pack stacked devices.

The **Stack Spacing** options let you specify the spacing between stacked devices.

P Stacking specifies the general vertical spacing in stacked devices of P rows.
Environment variable: spacingPStack

N Stacking specifies the general vertical spacing in stacked devices of N rows.
Environment variable: spacingNStack

Max Depth specifies the limit for stacking a device. You can choose between *Unlimited*, *No stacking*, and *Limited*. If you choose *Limited*, you can specify the limit manually in the cyclic field given below. You can specify a number between 1 and 9999.
Environment variable: maxStackDepth

The preview box on the right lets you preview how stacking changes with each stacking option you modify.

Related Topics

Specifying Global Settings for Creating Stacked Devices

Setup Options

Use the **Setup** options pane to control how SPD is invoked.

Launch SPD in controls whether SPD starts as a tabbed window in Virtuoso Layout Suite XL or if the same window should be reused next time SPD is launched.

© 2023 Cadence Design Systems, Inc.

Tabbed window opens SPD as tabbed window in Virtuoso Layout XL. If you clear this check box, SPD opens in a separate window. By default, this option is not selected. You must close and relaunch SPD for the changes to take effect.
Environment variable: tabbedWindow

Reuse window for faster relaunch reduces time needed to restart SPD by reusing the current SPD window when the tool is restarted. By default, this option is selected.

Note: The SPD window closes automatically when the target Layout XL window is closed.

You may want to clear this check box if:

- You want to release the license locked by the SPD window.
- You have made some changes that require you to launch SPD again. For example, the physical hierarchy configuration of the target layout has been modified, or the schematic view of the design has been modified.

When you change this option, you must close and reopen SPD for the changes to take effect.
Environment variable: reuseWindow

Load/Save specifies the file to which you can save or load all the settings in the SPD options form.

File specifies the name of the file to be loaded or saved. By default, this file is ~/.cdsenv.

Click **Save to** save the option settings in the specified file.

Environment variable: optionsFile

© 2023 Cadence Design Systems, Inc.

B

Symbolic Placement of Devices Environment Variables

This appendix provides information on the names, descriptions, and graphical user interface equivalents for the Virtuoso® Symbolic Placement of Devices (SPD) environment variables.

All environment variables related to the SPD Options form are used to set default values for the fields on the SPD Options form. You can set them either from CIW or load from cdsenv file through a SKILL function. For example, `envLoadFile("~/.cdsenv")`.

You must relaunch SPD Editor and click *Apply* on the corresponding form for changes in values to take effect. SPD retrieves the values of environment variables at startup as default values for forms, and does not refer to environment variables afterwards.

However, if you change values of environment variables through the SPD GUI, environment variables are updated and new values take effect immediately. You will not have to relaunch SPD. For example, you can change values on the SPD Options form and click *Apply*, or load `~./cdsenv` using the *Load from* option in the Setup Options form.

Note: Only the environment variables documented in this chapter are supported for public use. All other SPD environment variables, regardless of their name or prefix, and undocumented aspects of the environment variables described below, are private and are subject to change at any time.

© 2023 Cadence Design Systems, Inc.

List of SPD Environment Variables

© 2023 Cadence Design Systems, Inc.

- genLabelLr
- genLabelOri
- genLabelPGLr
- genLabelPGLrRatio
- genLabelPGPur
- genLabelType
- heightNmosRegion
- heightPmosRegion
- highlightGatePattern
- initDoChaining
- initDoFolding
- interdigitateChains
- labelDisplay
- launchRunOnce
- layerVdd
- layerVss
- maxStackDepth
- mediumSymbolMaxFins
- mediumSymbolMaxLength
- mediumSymbolMaxWidth
- mediumSymbolMinFins
- mediumSymbolMinLength
- mediumSymbolMinWidth
- mixPGFlightLinesSignals
- optionsFile
- pdPins
- pgTrunkBeginEndType

© 2023 Cadence Design Systems, Inc.

- pinDriven
- placementPriority
- preventGateShorts
- preventWellShorts
- reuseWindow
- rowPattern
- showPGFlightLines
- sigTrunkLr
- sigTrunkSpacingT2O
- sigTrunkSpacingT2T
- sigTrunkWidth
- spacingNStack
- spacingOd
- spacingOdMode
- spacingOdUserFunc
- spacingPodNod
- spacingPStack
- spacingVddPod
- spacingVssNod
- spdDesignLibName
- spdDummyLengthOptions
- spdDummyLengthValue
- spdDummyNet
- spdDummyNumFingersOptions
- spdDummyNumFingersValue
- spdDummyWidthOptions
- spdDummyWidthValue

© 2023 Cadence Design Systems, Inc.

- spdMoveAfterAppend
- spdRememberDummyVals
- tabbedWindow
- vddNetName
- vssNetName
- widthVdd
- widthVss

© 2023 Cadence Design Systems, Inc.

abutSameWidthOnly

```
layoutXL.symbolicPlacement abutSameWidthOnly boolean { t | nil }
```

Description

Specifies whether to only abut devices with same width. The default is `nil`.

GUI Equivalent

Command	*Options – SPD – Placement*
Field	*Abut same width only* (Placement Options)

Examples

```
envGetVal("layoutXL.symbolicPlacement" "abutSameWidthOnly")
envSetVal("layoutXL.symbolicPlacement" "abutSameWidthOnly" 'boolean t)
```

Related Topics

List of SPD Environment Variables

© 2023 Cadence Design Systems, Inc.

abutAllAutoFlipChain

```
layoutXL.symbolicPlacement abutAllAutoFlipChain boolean { t | nil }
```

Description

Automatically flips when the *Abut All* command is used. The default is `nil`.

GUI Equivalent

Command	*Options – SPD – Placement*
Field	*Auto flip chain in 'Abut All' command*

Examples

```
envGetVal("layoutXL.symbolicPlacement" "abutAllAutoFlipChain")
envSetVal("layoutXL.symbolicPlacement" "abutAllAutoFlipChain" 'boolean t)
```

Related Topics

Placement Options

Abutting All Devices

List of SPD Environment Variables

© 2023 Cadence Design Systems, Inc.

alignGates

```
layoutXL.symbolicPlacement alignGates boolean { t | nil }
```

Description

Specifies whether to consider device alignment when placing devices in the symbolic design. The default is `t`.

GUI Equivalent

Command	*Options – SPD – Placement*
Field	*Align to Gates* (Placement Options)

Note: When `alignGates` is set to `t`, the value in the *Align to Gates* field in GUI is not *None*.

Examples

```
envGetVal("layoutXL.symbolicPlacement" "alignGates")
envSetVal("layoutXL.symbolicPlacement" "alignGates" 'boolean nil)
```

Related Topics

List of SPD Environment Variables

© 2023 Cadence Design Systems, Inc.

alignGateStyle

```
layoutXL.symbolicPlacement alignGateStyle string { "left" | "center" | "right" }
```

Description

Specifies whether gates are aligned to the left, right, or center. The default is `center`.

This option is effective only when value of *Align to Gates* is not *None* or the alignGates environment variable is set to `t`.

GUI Equivalent

Command	*Options – SPD – Placement*
Field	*Align to Gates* (Placement Options)

Examples

```
envGetVal("layoutXL.symbolicPlacement" "alignGateStyle")
envSetVal("layoutXL.symbolicPlacement" "alignGateStyle" 'string "left")
envSetVal("layoutXL.symbolicPlacement" "alignGateStyle" 'string "center")
envSetVal("layoutXL.symbolicPlacement" "alignGateStyle" 'string "right")
```

Related Topics

List of SPD Environment Variables

© 2023 Cadence Design Systems, Inc.

alignMos

```
layoutXL.symbolicPlacement alignMos string { "inside" | "outside" | "center" }
```

Description

Specifies the chaining alignment of chained devices. The default is `inside`.

GUI Equivalent

Command	*Options – SPD – Placement*
Field	*Chain Alignment* (Placement Options)

Examples

```
envGetVal("layoutXL.symbolicPlacement" "alignMos")
envSetVal("layoutXL.symbolicPlacement" "alignMos" 'string "outside")
envSetVal("layoutXL.symbolicPlacement" "alignMos" 'string "center")
```

Related Topics

List of SPD Environment Variables

© 2023 Cadence Design Systems, Inc.

autoAlignGates

```
layoutXL.symbolicPlacement autoAlignGates boolean { t | nil }
```

Description

Aligns the first gate of a chain with another gate even if no alignment relationship has been assigned. The default is `t`.

This option is effective only when the value of *Align to Gates* is not *None or* the alignGates environment variable is set to `t`.

GUI Equivalent

Command	*Options – SPD – Placement*
Field	*Auto gate alignment* (Placement Options)

Examples

```
envGetVal("layoutXL.symbolicPlacement" "autoAlignGates")
envSetVal("layoutXL.symbolicPlacement" "autoAlignGates" 'boolean nil)
```

Related Topics

List of SPD Environment Variables

© 2023 Cadence Design Systems, Inc.

autoPackInX

```
layoutXL.symbolicPlacement autoPackInX boolean { t | nil }
```

Description

Automatically packs devices in horizontal direction when commands Move, Undo, Redo, Align, Save, and Load are used to change the position of devices. When set to `nil`, automatic packing does not take place. The default is `t`.

GUI Equivalent

Command	*Options – SPD – Placement*
Field	*Auto pack in X direction* (Placement Options)

Examples

```
envGetVal("layoutXL.symbolicPlacement" "autoPackInX")
envSetVal("layoutXL.symbolicPlacement" "autoPackInX" 'boolean nil)
```

Related Topics

List of SPD Environment Variables

Moving Devices Based on Whether Horizontal Packing is Enabled

Aligning Devices Based on Whether Horizontal Packing is Enabled

© 2023 Cadence Design Systems, Inc.

betweenRowPair

`layoutXL.symbolicPlacement betweenRowPair float` *`floating_point_number`*

Description

Specifies spacing between two row pairs when generating the layout. The default is `1.0`.

GUI Equivalent

Command	*Options – SPD – Generation*
Field	*Between Row-pair* (Generation Options)

Examples

```
envGetVal("layoutXL.symbolicPlacement" "betweenRowPair")
envSetVal("layoutXL.symbolicPlacement" "betweenRowPair" 'float 0.3)
```

Related Topics

List of SPD Environment Variables

© 2023 Cadence Design Systems, Inc.

chainFolds

```
layoutXL.symbolicPlacement chainFolds boolean { t | nil }
```

Description

Chains the resultant folds together where possible.

The default is `nil`.

GUI Equivalent

Command	*Options – SPD – Placement*
Field	*Chain Folds*

Examples

```
envGetVal("layoutXL.symbolicPlacement" "chainFolds")
envSetVal("layoutXL.symbolicPlacement" "chainFolds" 'boolean t)
```

Related Topics

Placement Options

List of SPD Environment Variables

© 2023 Cadence Design Systems, Inc.

createPGTrunk

```
layoutXL.symbolicPlacement createPGTrunk boolean { t | nil }
```

Description

Creates power rails as trunks. The default is `nil`.

This option is effective only when the *Options – SPD – Generation – Power Rails – Create* option is selected in GUI or the createPowerGroundRails environment variable is set to `t`.

GUI Equivalent

Command	*Options – SPD – Generation – Power Rails*
Field	*As trunk* (Power Rails Options)

Examples

```
envGetVal("layoutXL.symbolicPlacement" "createPGTrunk")
envSetVal("layoutXL.symbolicPlacement" "createPGTrunk" 'boolean t)
```

Related Topics

List of SPD Environment Variables

© 2023 Cadence Design Systems, Inc.

createPowerGroundRails

```
layoutXL.symbolicPlacement createPowerGroundRails boolean { t | nil }
```

Description

Creates power and ground rails. The default is `nil`.

GUI Equivalent

Command	*Options – SPD – Generation – Power Rails*
Field	*Create* (Power Rails Options)

Examples

```
envGetVal("layoutXL.symbolicPlacement" "createPowerGroundRails")
envSetVal("layoutXL.symbolicPlacement" "createPowerGroundRails" 'boolean t)
```

Related Topics

List of SPD Environment Variables

© 2023 Cadence Design Systems, Inc.

createSigTrunk

```
layoutXL.symbolicPlacement createSigTrunk boolean { t | nil }
```

Description

Specifies whether signal trunks are created when the layout is generated. The default is `nil`.

GUI Equivalent

Command	*Options – SPD – Generation – Power Rails– Signal Trunks*
Field	*Create* (Power Rails Options)

Examples

```
envGetVal("layoutXL.symbolicPlacement" "createSigTrunk")
envSetVal("layoutXL.symbolicPlacement" "createSigTrunk" 'boolean t)
```

Related Topics

List of SPD Environment Variables

Reordering and Resetting Trunks

© 2023 Cadence Design Systems, Inc.

deleteExistingRails

```
layoutXL.symbolicPlacement deleteExistingRails boolean { t | nil }
```

Description

Deletes any existing power rails when generating the layout. The default is `nil`, which means that power rails are not be deleted when the layout is generated.

GUI Equivalent

Command	*Options – SPD – Generation – Power Rails*
Field	*Delete existing rails* (Power Rails Options)

Examples

```
envGetVal("layoutXL.symbolicPlacement" "deleteExistingRails")
envSetVal("layoutXL.symbolicPlacement" "deleteExistingRails" 'boolean t)
```

Related Topics

List of SPD Environment Variables

© 2023 Cadence Design Systems, Inc.

expandSpacingForTrunks

```
layoutXL.symbolicPlacement expandSpacingForTrunks boolean { t | nil }
```

Description

Specifies whether the maximum number of tracks need to be calculated from trunk width and spacing information, which includes within row-pair spacing, trunk-to-OD spacing and trunk-to-trunk spacing. The default is `nil`.

GUI Equivalent

Command	*Options – SPD – Generation – Power Rails*
Field	*Auto expand within row-pair spacing* (Power Rails Options)

Examples

```
envGetVal("layoutXL.symbolicPlacement" "expandSpacingForTrunks")
envSetVal("layoutXL.symbolicPlacement" "expandSpacingForTrunks" 'boolean t)
```

Related Topics

List of SPD Environment Variables

Automatic Shifting of Additional Tracks

© 2023 Cadence Design Systems, Inc.

flightLineConn

```
layoutXL.symbolicPlacement flightLineConn cyclic { "both" | "1 row" | "2 rows" }
```

Description

Specifies the type of flight lines that are displayed in the symbolic design.

- `1 row` displays flight lines that show connections within 1 PN row.
- `2 row` displays flight lines that show connections between 2 PN rows.
- `both` displays flight lines of both types.

The default is `both`.

This option is available only when the *Options – SPD – Placement – Row Pattern* option or the rowPattern environment variable is set to a multirow pattern, such as `NPPN` or `PNNP`.

GUI Equivalent

Command	*Options – SPD – Display*
Field	*Connect between*

Examples

```
envGetVal("layoutXL.symbolicPlacement" "flightLineConn")
envSetVal("layoutXL.symbolicPlacement" "flightLineConn" 'cyclic "1 row")
envSetVal("layoutXL.symbolicPlacement" "flightLineConn" 'cyclic "2 rows")
```

Related Topics

Display Options

List of SPD Environment Variables

© 2023 Cadence Design Systems, Inc.

flightLineMode

```
layoutXL.symbolicPlacement flightLineMode cyclic { "both" | "gate" | "active" |
     "none" }
```

Description

Specifies the connectivity mode of flight lines. Connectivity modes are:

- `gate` displays flight lines that show connections to gates.
- `active` displays flight lines that show connections to the active region.
- `both` displays flight lines that show connections to both gates and the active region.
- `none` disables flight lines.

The default is `both`.

GUI Equivalent

Command	*Options – SPD – Display*
Field	*Connect to* (Display Options)

Examples

```
envGetVal("layoutXL.symbolicPlacement" "flightLineMode")
envSetVal("layoutXL.symbolicPlacement" "flightLineMode" 'cyclic "gate")
envSetVal("layoutXL.symbolicPlacement" "flightLineMode" 'cyclic "active")
```

Related Topics

List of SPD Environment Variables

© 2023 Cadence Design Systems, Inc.

foldingThresholdNType

`layoutXL.symbolicPlacement foldingThresholdNType float` *`floating_point_number`*

Description

Specifies the maximum size of a folded N-type transistor.

The default is `0`, which means that SPD automatically retrieves the value from the folding threshold parameter of cells belonging to the component class NMOS. In FinFET-based designs, SPD retrieves the value from the folding threshold fins parameter of cells belonging to component class NFIN.

GUI Equivalent

Command	*Options – SPD – Placement*
Field	*NMOS Fold Threshold*
	NFIN Fold Threshold (for FinFET-based designs)

Examples

```
envGetVal("layoutXL.symbolicPlacement" "foldingThresholdNType")
envSetVal("layoutXL.symbolicPlacement" "foldingThresholdNType" 'float 2)
```

Related Topics

Component Types

Placement Options

List of SPD Environment Variables

© 2023 Cadence Design Systems, Inc.

foldingThresholdPType

`layoutXL.symbolicPlacement foldingThresholdPType float` *`floating_point_number`*

Description

Specifies the maximum size of a folded P-type transistor.

The default is `0`, which means that SPD automatically retrieves the value from the folding threshold parameter of cells belonging to the component class PMOS. In FinFET-based designs, SPD retrieves the value from the folding threshold fins parameter of cells belonging to component class PFIN.

GUI Equivalent

Command	*Options – SPD – Placement*
Field	*PMOS Fold Threshold*
	PFIN Fold Threshold (for FinFET-based designs)

Examples

```
envGetVal("layoutXL.symbolicPlacement" "foldingThresholdPType")
envSetVal("layoutXL.symbolicPlacement" "foldingThresholdPType" 'float 2)
```

Related Topics

Component Types

Placement Options

List of SPD Environment Variables

© 2023 Cadence Design Systems, Inc.

freeSymbolLength

```
layoutXL.symbolicPlacement freeSymbolLength boolean { t | nil }
```

Description

Specifies whether free bucket size mode should be used to display lengths of symbolic devices. The default is `nil`.

Note: You can either display free bucket size for width or length. If both `freeSymbolWidth` and `freeSymbolLength` are set to `t`, SPD will enable free bucket size for width, which is the default.

GUI Equivalent

Command	*Options – Display – Size for Medium Symbol – Length*
Field	*Free* (Display Options)

Examples

```
envGetVal("layoutXL.symbolicPlacement" "freeSymbolLength")
envSetVal("layoutXL.symbolicPlacement" "freeSymbolLength" 'boolean t)
```

Related Topics

List of SPD Environment Variables

© 2023 Cadence Design Systems, Inc.

freeSymbolWidth

```
layoutXL.symbolicPlacement freeSymbolWidth boolean { t | nil }
```

Description

Specifies whether free bucket size mode should be used to display widths of symbolic devices. The default is `nil`.

GUI Equivalent

Command	*Options – Display – Size for Medium Symbol – Width*
Field	*Free* (Display Options)

Examples

```
envGetVal("layoutXL.symbolicPlacement" "freeSymbolWidth")
envSetVal("layoutXL.symbolicPlacement" "freeSymbolWidth" 'boolean t)
```

Related Topics

List of SPD Environment Variables

© 2023 Cadence Design Systems, Inc.

genLabel

```
layoutXL.symbolicPlacement genLabel boolean { t | nil }
```

Description

Creates labels when generating the layout. The default is `nil`.

GUI Equivalent

Command	*Options – SPD – Generation – Labels*
Field	*Create labels* (Labels Options)

Examples

```
envGetVal("layoutXL.symbolicPlacement" "genLabel")
envSetVal("layoutXL.symbolicPlacement" "genLabel" 'boolean t)
```

Related Topics

List of SPD Environment Variables

© 2023 Cadence Design Systems, Inc.

genLabelFont

```
layoutXL.symbolicPlacement genLabelFont cyclic { "euroStyle" | "gothic" | "math" |
     "roman" | "script" | "stick" | "fixed" | "swedish" | "milSpec" }
```

Description

Specifies the font for the labels. The default is `roman`.

GUI Equivalent

Command	*Options – SPD – Generation – Labels*
Field	*Font* (Labels Options)

Examples

```
envGetVal("layoutXL.symbolicPlacement" "genLabelFont")
envSetVal("layoutXL.symbolicPlacement" "genLabelFont" 'cyclic "euroStyle")
envSetVal("layoutXL.symbolicPlacement" "genLabelFont" 'cyclic "gothic")
envSetVal("layoutXL.symbolicPlacement" "genLabelFont" 'cyclic "math")
envSetVal("layoutXL.symbolicPlacement" "genLabelFont" 'cyclic "script")
envSetVal("layoutXL.symbolicPlacement" "genLabelFont" 'cyclic "stick")
envSetVal("layoutXL.symbolicPlacement" "genLabelFont" 'cyclic "fixed")
envSetVal("layoutXL.symbolicPlacement" "genLabelFont" 'cyclic "swedish")
envSetVal("layoutXL.symbolicPlacement" "genLabelFont" 'cyclic "milSpec")
```

Related Topics

List of SPD Environment Variables

© 2023 Cadence Design Systems, Inc.

genLabelHeight

```
layoutXL.symbolicPlacement genLabelHeight cyclic { Auto | Fixed }
```

Description

Specifies whether the height of instance labels is determined automatically or set to a fixed size. The default is `Auto`.

GUI Equivalent

Command	*Options – SPD – Generation – Labels*
Field	*Height* (Labels Options)

Examples

```
envGetVal("layoutXL.symbolicPlacement" "genLabelHeight")
envSetVal("layoutXL.symbolicPlacement" "genLabelHeight" 'cyclic "Fixed")
```

Related Topics

List of SPD Environment Variables

© 2023 Cadence Design Systems, Inc.

genLabelHeightValue

`layoutXL.symbolicPlacement genLabelHeightValue float` *`floating_point_number`*

Description

Specifies the value of height for the instance label. The default is `0.1`.

This option is effective only when the *Options – SPD – Generation – Labels – Height* option or the genLabelHeight environment variable is set to `Fixed`.

GUI Equivalent

Command	*Options – SPD – Generation – Labels*
Field	Field under *Height* (Labels Options)

Examples

```
envGetVal("layoutXL.symbolicPlacement" "genLabelHeightValue")
envSetVal("layoutXL.symbolicPlacement" "genLabelHeightValue" 'float 0.2)
```

Related Topics

List of SPD Environment Variables

© 2023 Cadence Design Systems, Inc.

genLabelJust

```
layoutXL.symbolicPlacement genLabelJust cyclic { "Auto" | "lowerLeft" |
     "centerLeft" | "upperLeft" | "lowerCenter" | "centerCenter" | "upperCenter" |
     "lowerRight" | "centerRight" | "upperRight" }
```

Description

Sets the justification or the alignment of the instance label. The default is `Auto`.

GUI Equivalent

Command	*Options – SPD – Generation – Labels*
Field	*Justification* (Labels Options)

Examples

```
envGetVal("layoutXL.symbolicPlacement" "genLabelJust")
envSetVal("layoutXL.symbolicPlacement" "genLabelJust" 'cyclic "lowerLeft")
envSetVal("layoutXL.symbolicPlacement" "genLabelJust" 'cyclic "centerLeft")
envSetVal("layoutXL.symbolicPlacement" "genLabelJust" 'cyclic "upperLeft")
envSetVal("layoutXL.symbolicPlacement" "genLabelJust" 'cyclic "lowerCenter")
envSetVal("layoutXL.symbolicPlacement" "genLabelJust" 'cyclic "centerCenter")
envSetVal("layoutXL.symbolicPlacement" "genLabelJust" 'cyclic "upperCenter")
envSetVal("layoutXL.symbolicPlacement" "genLabelJust" 'cyclic "lowerRight")
envSetVal("layoutXL.symbolicPlacement" "genLabelJust" 'cyclic "centerRight")
envSetVal("layoutXL.symbolicPlacement" "genLabelJust" 'cyclic "upperRight")
```

Related Topics

List of SPD Environment Variables

© 2023 Cadence Design Systems, Inc.

genLabelLr

```
layoutXL.symbolicPlacement genLabelLr string { "layerName purposeName" }
```

Description

Specifies the layer-purpose pair on which to create the instance labels. The default is `""`, which means that the software will automatically assign a value to the *Layer* field. The value will be `"text drawing"` if the technology file in use has it, otherwise the value will be the first metal layer.

GUI Equivalent

Command	*Options – SPD – Generation – Labels*
Field	*Layer* (Labels Options)

Examples

```
envGetVal("layoutXL.symbolicPlacement" "genLabelLr")
envSetVal("layoutXL.symbolicPlacement" "genLabelLr" 'string "text drawing")
```

Related Topics

List of SPD Environment Variables

© 2023 Cadence Design Systems, Inc.

genLabelOri

```
layoutXL.symbolicPlacement genLabelOri cyclic "Auto" { "Auto" | "R0" | "R90" |
     "R180" | "R270" | "MY" | "MYR90" | "MX" | "MXR90" }
```

Description

Sets the orientation of the instance label. The default is `Auto`.

Value	Meaning
Auto	Same orientation as the instance. This is the default value.
R0	No rotation
R90	Rotate 90 degrees clockwise
R180	Rotate 180 degrees clockwise
R270	Rotate 270 degrees clockwise
MY	Mirror over the Y axis
MYR90	Mirror over the Y axis and rotate 90 degrees clockwise
MX	Mirror over the X axis
MXR90	Mirror over the X axis and rotate 90 degrees clockwise

GUI Equivalent

Command	*Options – SPD – Generation – Label*
Field	*Orientation* (Labels Options)

Examples

```
envGetVal("layoutXL.symbolicPlacement" "genLabelOri")
envSetVal("layoutXL.symbolicPlacement" "genLabelOri" 'cyclic "R0")
envSetVal("layoutXL.symbolicPlacement" "genLabelOri" 'cyclic "R90")
envSetVal("layoutXL.symbolicPlacement" "genLabelOri" 'cyclic "R180")
envSetVal("layoutXL.symbolicPlacement" "genLabelOri" 'cyclic "R270")
envSetVal("layoutXL.symbolicPlacement" "genLabelOri" 'cyclic "MY")
envSetVal("layoutXL.symbolicPlacement" "genLabelOri" 'cyclic "MYR90")
envSetVal("layoutXL.symbolicPlacement" "genLabelOri" 'cyclic "MX")
envSetVal("layoutXL.symbolicPlacement" "genLabelOri" 'cyclic "MXR90")
```


© 2023 Cadence Design Systems, Inc.

Related Topics

List of SPD Environment Variables

© 2023 Cadence Design Systems, Inc.

genLabelPGLr

```
layoutXL.symbolicPlacement genLabelPGLr string { "layerName purposeName" }
```

Description

Specifies the layer-purpose pair on which to create the instance labels. The default is `""`, which means that the software will automatically assign a value to the second cyclic field in the *For P/G Rail* section. The value will be `"text drawing"` if the technology file in use has it, otherwise the value will be the first metal layer.

This option is available only when the following two conditions are met:

- The value of *Options – SPD – Generation – Labels – Objects Type* GUI option or the genLabelType environment variable is `Power/Ground Rails` or `Both`.
- The value of the *Options – SPD – Generation – Labels – Use same layer as shape, select purpose* GUI option or the genLabelPGLrRatio environment variable is set to `Select layer`.

GUI Equivalent

Command	*Options – SPD – Generation – Labels*
Field	The second cyclic field in the *For P/G Rail* section (Labels Options)

Examples

```
envGetVal("layoutXL.symbolicPlacement" "genLabelPGLr")
envSetVal("layoutXL.symbolicPlacement" "genLabelPGLr" 'string 'Metal1 drawing')
```

Related Topics

List of SPD Environment Variables

© 2023 Cadence Design Systems, Inc.

genLabelPGLrRatio

```
layoutXL.symbolicPlacement genLabelPGLrRatio cyclic { "Use same layer as shape,
     select purpose" | "Select layer" }
```

Description

Specifies how to select the layer-purpose pair on which to create power rail labels. The default is "`Use same layer as shape, select purpose`".

GUI Equivalent

Command	*Options – SPD – Generation – Labels*
Field	*Use same layer as shape, select purpose* (Labels Options)

Examples

```
envGetVal("layoutXL.symbolicPlacement" "genLabelPGLrRatio")
envSetVal("layoutXL.symbolicPlacement" "genLabelPGLrRatio" 'cyclic "Select layer")
```

Related Topics

List of SPD Environment Variables

© 2023 Cadence Design Systems, Inc.

genLabelPGPur

```
layoutXL.symbolicPlacement genLabelPGPur string { purposeName }
```

Description

Specifies the purpose used to create the instance labels. The default is `""`, which means that the software will automatically assign `"drawing"` to the first cyclic field in the *For P/G Rail* section.

This option is effective only when value of the genLabelPGLrRatio environment variable is set to `Use same layer as shape, select purpose`.

GUI Equivalent

Command	*Options – SPD – Generation – Labels*
Field	The first cyclic field in the *For P/G Rail* section (Labels Options)

Examples

```
envGetVal("layoutXL.symbolicPlacement" "genLabelPGPur")
envSetVal("layoutXL.symbolicPlacement" "genLabelPGPur" 'string "drawing")
```

Related Topics

List of SPD Environment Variables

© 2023 Cadence Design Systems, Inc.

genLabelType

```
layoutXL.symbolicPlacement genLabelType cyclic { "MOS" | "Power/Ground Rails" |
     "Both" }
```

Description

Specifies the objects for which the labels are to be created. The default is `MOS`.

GUI Equivalent

Command	*Options – SPD – Generation – Labels*
Field	*Objects Type* (Labels Options)

Examples

```
envGetVal("layoutXL.symbolicPlacement" "genLabelType")
envSetVal("layoutXL.symbolicPlacement" "genLabelType" 'cyclic "Power/Ground
Rails")
envSetVal("layoutXL.symbolicPlacement" "genLabelType" 'cyclic "Both")
```

Related Topics

List of SPD Environment Variables

© 2023 Cadence Design Systems, Inc.

heightNmosRegion

`layoutXL.symbolicPlacement heightNmosRegion float` *`floating_point_number`*

Description

Specifies the height for the NMOS region. The default is `0.0`.

GUI Equivalent

Command	*Options – SPD – Generation*
Field	*NMOS* (Generation Options)

Examples

```
envGetVal("layoutXL.symbolicPlacement" "heightNmosRegion")
envSetVal("layoutXL.symbolicPlacement" "heightNmosRegion" 'float 0.2)
```

Related Topics

List of SPD Environment Variables

© 2023 Cadence Design Systems, Inc.

heightPmosRegion

`layoutXL.symbolicPlacement heightPmosRegion float` *`floating_point_number`*

Description

Specifies the height for the PMOS region. The default is `0.0`.

GUI Equivalent

Command	*Options – SPD – Generation*
Field	*PMOS* (Generation Options)

Examples

```
envGetVal("layoutXL.symbolicPlacement" "heightPmosRegion")
envSetVal("layoutXL.symbolicPlacement" "heightPmosRegion" 'float 0.2)
```

Related Topics

List of SPD Environment Variables

© 2023 Cadence Design Systems, Inc.

highlightGatePattern

```
layoutXL.symbolicPlacement highlightGatePattern boolean { t | nil }
```

Description

Highlights different device patterns, such as m-factors, folds, and finger-splits in different colors in the design.The default is `nil`.

GUI Equivalent

Command	*Options – Display*
Field	*Highlight P/N device patterns on gates* (Display Options)

Examples

```
envGetVal("layoutXL.symbolicPlacement" "highlightGatePattern")
envSetVal("layoutXL.symbolicPlacement" "highlightGatePattern" 'boolean t)
```

Related Topics

List of SPD Environment Variables

© 2023 Cadence Design Systems, Inc.

initDoChaining

```
layoutXL.symbolicPlacement initDoChaining boolean { t | nil }
```

Description

Abuts the selected set of MOS transistors or FinFET devices automatically to form a chain. The transistors to be abutted must be set up for abutment first.

The default is `t`.

GUI Equivalent

Command	*Options – SPD – Placement*
Field	*Chain*

Examples

```
envGetVal("layoutXL.symbolicPlacement" "initDoChaining")
envSetVal("layoutXL.symbolicPlacement" "initDoChaining" 'boolean nil)
```

Related Topics

Placement Options

List of SPD Environment Variables

© 2023 Cadence Design Systems, Inc.

initDoFolding

```
layoutXL.symbolicPlacement initDoFolding boolean { t | nil }
```

Description

Divides MOS transistors or FinFET devices automatically into folds to optimize utilization.

The default is `nil`.

GUI Equivalent

Command	*Options – SPD – Placement*
Field	*Fold*

Examples

```
envGetVal("layoutXL.symbolicPlacement" "initDoFolding")
envSetVal("layoutXL.symbolicPlacement" "initDoFolding" 'boolean t)
```

Related Topics

Placement Options

List of SPD Environment Variables

© 2023 Cadence Design Systems, Inc.

interdigitateChains

```
layoutXL.symbolicPlacement interdigitateChains boolean { t | nil }
```

Description

Identifies the nodes that qualify as pseudoparallel connections automatically and defines them accordingly during chaining and abutment.

The default is `nil`.

GUI Equivalent

Command	*Options – SPD – Placement*
Field	*Interdigitate Chains*

Examples

```
envGetVal("layoutXL.symbolicPlacement" "interdigitateChains")
```

```
envSetVal("layoutXL.symbolicPlacement" "interdigitateChains" 'boolean t)
```

Related Topics

Placement Options

List of SPD Environment Variables

© 2023 Cadence Design Systems, Inc.

labelDisplay

```
layoutXL.symbolicPlacement labelDisplay string { "All" | "All except Device Size"
     | "Gate, Instance and Well Net Names" | "Gate and Instance Net Names" | "Gate
     Net Name Only" }
```

Description

Specifies how to display text label on abstract devices. The default is `All except Device Size`.

GUI Equivalent

Command	*Options – SPD – Display*
Field	*Label Display* (Display Options)

Examples

```
envGetVal("layoutXL.symbolicPlacement" "labelDisplay")
envSetVal("layoutXL.symbolicPlacement" "labelDisplay" 'string "All")
envSetVal("layoutXL.symbolicPlacement" "labelDisplay" 'string "Gate, Instance and
Well Net Names")
envSetVal("layoutXL.symbolicPlacement" "labelDisplay" 'string "Gate and Instance
Net Names")
envSetVal("layoutXL.symbolicPlacement" "labelDisplay" 'string "Gate Net Name
Only")
```

Related Topics

List of SPD Environment Variables

© 2023 Cadence Design Systems, Inc.

launchRunOnce

```
layoutXL.symbolicPlacement launchRunOnce string { "SKILL_Evaluation_String" }
```

Description

Specifies a string, which has a valid SKILL syntax, to be evaluated before SPD is launched for the first time. The default is `""`.

GUI Equivalent

None

Examples

```
envGetVal("layoutXL.symbolicPlacement" "launchRunOnce")
envSetVal("layoutXL.symbolicPlacement" "launchRunOnce" 'string "UserInitPDK1()
UserInitPDK2()")
; UserInitPDK1() and UserInitPDK2() will be called before SPD is launched for the
first time
```

Related Topics

User-Defined Flow Callback Functions

List of SPD Environment Variables

© 2023 Cadence Design Systems, Inc.

layerVdd

```
layoutXL.symbolicPlacement layerVdd string { layerName purposeName }
```

Description

Specifies the layer purpose used to create the power rail for the power net. This includes all valid layers that are assigned to metal or poly functions in the technology file. The default is `" "`, which means that the software will automatically assign the first metal layer defined in the technology file to *Power Layer* field.

GUI Equivalent

Command	*Options – SPD – Generation – Power Rails*
Field	*Power Layer* (Power Rails Options)

Examples

```
envGetVal("layoutXL.symbolicPlacement" "layerVdd")
envSetVal("layoutXL.symbolicPlacement" "layerVdd" 'string "Metal1 drawing")
```

Related Topics

List of SPD Environment Variables

© 2023 Cadence Design Systems, Inc.

layerVss

```
layoutXL.symbolicPlacement layerVss string { layerName purposeName }
```

Description

Specifies the layer purpose used to create the power rail for the ground net. This includes all valid layers that are assigned to metal or poly functions in the technology file. The default is `""`, which means that the software will automatically assign the first metal layer defined in the technology file to the *Ground Layer* field.

GUI Equivalent

Command	*Options – SPD – Generation – Power Rails*
Field	*Ground Layer* (Power Rails Options)

Examples

```
envGetVal("layoutXL.symbolicPlacement" "layerVss")
envSetVal("layoutXL.symbolicPlacement" "layerVss" 'string "Metal1 drawing")
```

Related Topics

List of SPD Environment Variables

© 2023 Cadence Design Systems, Inc.

maxStackDepth

```
layoutXL.symbolicPlacement maxStackDepth int { -1 to 9999 }
```

Description

Specifies the maximum depth for stacking devices.

- A positive integer denotes that *Max Depth* is *Limited* to this integer number.
- `0` denotes that *Max Depth* is set to *No stacking.*
- `-1` denotes that *Max Depth* is *Unlimited*.

The default is `-1`. This option is effective only when the value of *Max Depth* field is *Limited*.

GUI Equivalent

Command	*Options – SPD – Generation – Stack*
Field	The field under *Max Depth* (Labels Options)

Examples

```
envGetVal("layoutXL.symbolicPlacement" "maxStackDepth")
envSetVal("layoutXL.symbolicPlacement" "maxStackDepth" 'int -3)
```

Related Topics

List of SPD Environment Variables

© 2023 Cadence Design Systems, Inc.

mediumSymbolMaxFins

```
layoutXL.symbolicPlacement mediumSymbolMaxFins  int { 0 to 10000000 }
```

Description

Specifies the maximum number of fins for the medium symbol. The default is `0`, which means that *Fins Max* is computed by the software automatically. This option is available only when there are FinFET devices in the design.

GUI Equivalent

Command	*Options – SPD – Display*
Field	*Fins Max* (Display Options)

Examples

```
envGetVal("layoutXL.symbolicPlacement" "mediumSymbolMaxFins")
envSetVal("layoutXL.symbolicPlacement" "mediumSymbolMaxFins" 'int 6)
```

Related Topics

List of SPD Environment Variables

© 2023 Cadence Design Systems, Inc.

mediumSymbolMaxLength

`layoutXL.symbolicPlacement mediumSymbolMaxLength float` *`floating_point_number`*

Description

Sets the maximum length for the medium symbol. The default is `0.0`, which means that *Length Max* is computed by the software automatically.

GUI Equivalent

Command	*Options – SPD – Display*
Field	*Length Max* (Display Options)

Examples

```
envGetVal("layoutXL.symbolicPlacement" "mediumSymbolMaxLength")
envSetVal("layoutXL.symbolicPlacement" "mediumSymbolMaxLength" 'float 0.03)
```

Related Topics

List of SPD Environment Variables

© 2023 Cadence Design Systems, Inc.

mediumSymbolMaxWidth

`layoutXL.symbolicPlacement mediumSymbolMaxWidth` *`floating_point_number`*

Description

Sets the maximum width for the medium symbol. The default is `0.0`, which means that *Width Max* is computed by the software automatically.

GUI Equivalent

Command	*Options – SPD – Display*
Field	*Width Max* (Display Options)

Examples

```
envGetVal("layoutXL.symbolicPlacement" "mediumSymbolMaxWidth")
envSetVal("layoutXL.symbolicPlacement" "mediumSymbolMaxWidth" 'float 0.3)
```

Related Topics

List of SPD Environment Variables

© 2023 Cadence Design Systems, Inc.

mediumSymbolMinFins

```
layoutXL.symbolicPlacement mediumSymbolMinFins int { 0 through 10000000 }
```

Description

Specifies the minimum number of fins for the medium symbol. The default is `0`, which means that *Fins Min* is computed by the software automatically.

GUI Equivalent

Command	*Options – SPD – Display*
Field	*Fins Min* (Display Options)

Examples

```
envGetVal("layoutXL.symbolicPlacement" "mediumSymbolMinFins")
envSetVal("layoutXL.symbolicPlacement" "mediumSymbolMinFins" 'int 2)
```

Related Topics

List of SPD Environment Variables

© 2023 Cadence Design Systems, Inc.

mediumSymbolMinLength

`layoutXL.symbolicPlacement mediumSymbolMinLength float` *`floating_point_number`*

Description

Sets the minimum length for the medium symbol. The default is `0.0`, which means that *Length Min* is computed by the software automatically.

GUI Equivalent

Command	*Options – SPD – Display*
Field	*Length Min* (Display Options)

Examples

```
envGetVal("layoutXL.symbolicPlacement" "mediumSymbolMinLength")
envSetVal("layoutXL.symbolicPlacement" "mediumSymbolMnLength" 'float 0.01)
```

Related Topics

List of SPD Environment Variables

© 2023 Cadence Design Systems, Inc.

mediumSymbolMinWidth

`layoutXL.symbolicPlacement mediumSymbolMinWidth float` *`floating_point_number`*

Description

Sets the minimum width for the medium symbol. The default is `0.0`, which means that *Width Min* is computed by the software automatically.

GUI Equivalent

Command	*Options – SPD – Display*
Field	*Width Min* (Display Options)

Examples

```
envGetVal("layoutXL.symbolicPlacement" "mediumSymbolMinWidth")
envSetVal("layoutXL.symbolicPlacement" "mediumSymbolMinWidth" 'float 0.2)
```

Related Topics

List of SPD Environment Variables

© 2023 Cadence Design Systems, Inc.

mixPGFlightLinesSignals

```
layoutXL.symbolicPlacement mixPGFlightLinesSignals boolean { t | nil }
```

Description

Specifies whether to mix flight lines of power and ground nets with signal nets.The default is `nil`.

This option is effective only when *Options – SPD – Display – Show flight lines* option is selected in GUI and the showPGFlightLines environment variable is set to `t`.

GUI Equivalent

Command	*Options – SPD – Display*
Field	*Show P/G Nets* (Display Options)

Examples

```
envGetVal("layoutXL.symbolicPlacement" "mixPGFlightLinesSignals")
envSetVal("layoutXL.symbolicPlacement" "mixPGFlightLinesSignals" 'boolean t)
```

Related Topics

List of SPD Environment Variables

© 2023 Cadence Design Systems, Inc.

optionsFile

```
layoutXL.symbolicPlacement optionsFile string "optionsFileName"
```

Description

Specifies the path of the options file. You can use this file to save or load SPD environmental variables.

The default is `"~/.cdsenv"`.

GUI Equivalent

Command:	*Options – SPD – Setup*
Field:	*File* This environment variable applies to the initial and default values of the *File* field. However, a value change in this field is not set to this environmental variable.

Examples

```
envGetVal("layoutXL.symbolicPlacement" "optionsFile")
envSetVal("layoutXL.symbolicPlacement" "optionsFile" 'string "spd.options")
```

Related Topic

Setup Options

© 2023 Cadence Design Systems, Inc.

pdPins

```
layoutXL.symbolicPlacement pdPins string "list_of_pins"
```

Description

Specifies the pin order (left to right) for pin driven placement. List the pin names each separated by a space and enclose the list in double quotes. The default is `""`, which means no pin order is specified. The software will not assign any values to this field.

This option is effective only when *Options – SPD – Placement – Pin Driven* check box is selected in GUI, or the pinDriven environment variable is set to `t`.

GUI Equivalent

Command	*Options – SPD – Placement*
Field	*Order* (Placement Options)

Examples

```
envGetVal("layoutXL.symbolicPlacement" "pdPins")
envSetVal("layoutXL.symbolicPlacement" "pdPins" 'string "CK D RN Q QN")
```

Related Topics

List of SPD Environment Variables

© 2023 Cadence Design Systems, Inc.

pgTrunkBeginEndType

```
layoutXL.symbolicPlacement pgTrunkBeginEndType cyclic { "truncate" | "extend" }
```

Description

Specifies the begin and end styles for the power rails. The default is `truncate`.

This option is effective only when both createPowerGroundRails and createPGTrunk are set to `t`.

GUI Equivalent

Command	*Options – SPD – Generation – Power Rails*
Field	The dropdown list under *As trunk* (Power Rails Options)

Examples

```
envGetVal("layoutXL.symbolicPlacement" "pgTrunkBeginEndType")
envSetVal("layoutXL.symbolicPlacement" "pgTrunkBeginEndType" 'cyclic "extend")
```

Related Topics

List of SPD Environment Variables

© 2023 Cadence Design Systems, Inc.

pinDriven

```
layoutXL.symbolicPlacement pinDriven boolean { t | nil }
```

Description

Enables pin driven placement. The default is `nil`.

GUI Equivalent

Command	*Options – SPD – Placement*
Field	*Pin Driven* (Placement Options)

Examples

```
envGetVal("layoutXL.symbolicPlacement" "pinDriven")
envSetVal("layoutXL.symbolicPlacement" "pinDriven" 'boolean t)
```

Related Topics

List of SPD Environment Variables

© 2023 Cadence Design Systems, Inc.

placementPriority

```
layoutXL.symbolicPlacement placementPriority string { "auto" | "maxOdSharing" |
     "minNumTrack" }
```

Description

Specifies the placement priority.

- `auto` automatically sets the placement priority. This is the default value.
- `maxOdSharing` optimizes the generated chains by maximizing the oxide diffusion sharing.
- `minNumTrack` optimizes the generated chains by minimizing the number of tracks in the chained devices.

GUI Equivalent

Command	*Options – SPD – Placement*
Field	*Priority* (Placement Options)

Examples

```
envGetVal("layoutXL.symbolicPlacement" "placementPriority")
envSetVal("layoutXL.symbolicPlacement" "placementPriority" 'string
"maxOdSharing")
envSetVal("layoutXL.symbolicPlacement" "placementPriority" 'string "minNumTrack")
```

Related Topics

List of SPD Environment Variables

© 2023 Cadence Design Systems, Inc.

PNmixedRow

```
layoutXL.symbolicPlacement PMmixedRow boolean { t | nil }
```

Description

Specifies whether both P-type and N-type devices can be specified in a single row. The default is `nil`.

GUI Equivalent

Command	*Options – SPD – Placement – Row Pattern*
Field	*Mix P//N in one row* (Placement Options)

Examples

```
envGetVal("layoutXL.symbolicPlacement" "PMmixedRow")
envSetVal("layoutXL.symbolicPlacement" "PMmixedRow" 'boolean t)
```

Related Topics

List of SPD Environment Variables

© 2023 Cadence Design Systems, Inc.

preventGateShorts

```
layoutXL.symbolicPlacement preventGateShorts boolean { t | nil }
```

Description

Prevents gate shorts when generating the layout. The default is `t`.

GUI Equivalent

Command	*Options – SPD – Generation*
Field	*Prevent gate shorts* (Generation Options)

Examples

```
envGetVal("layoutXL.symbolicPlacement" "preventGateShorts")
envSetVal("layoutXL.symbolicPlacement" "preventGateShorts" 'boolean t)
```

Related Topics

List of SPD Environment Variables

© 2023 Cadence Design Systems, Inc.

preventWellShorts

```
layoutXL.symbolicPlacement preventWellShorts boolean { t | nil }
```

Description

Prevents well shorts when generating the layout. The default is `nil`.

GUI Equivalent

Command	*Options – SPD – Generation*
Field	*Prevent well shorts* (Generation Options)

Examples

```
envGetVal("layoutXL.symbolicPlacement" "preventWellShorts")
envSetVal("layoutXL.symbolicPlacement" "preventWellShorts" 'boolean t)
```

Related Topics

List of SPD Environment Variables

© 2023 Cadence Design Systems, Inc.

reuseWindow

```
layoutXL.symbolicPlacement reuseWindow boolean { t | nil }
```

Description

Re-uses the SPD editor window for faster invocation of the tool. The default is `t`.

GUI Equivalent

Command	*Options – SPD – Setup*
Field	*Reuse window for faster relaunch* (Setup Options)

Examples

```
envGetVal("layoutXL.symbolicPlacement" "reuseWindow")
envSetVal("layoutXL.symbolicPlacement" "reuseWindow" 'boolean nil)
```

Related Topics

List of SPD Environment Variables

© 2023 Cadence Design Systems, Inc.

rowPattern

```
layoutXL.symbolicPlacement rowPattern string { "NP" | "PN" | "NPPN" | "PNNP" |
     "NPNP" | "PNPN" | "NNPP" | "PPNN" | "NNNP" | "PNNN" | "NPPP" | "PPPN" | "NNNN"
     | "PPPP" }
```

Description

Controls the row order from bottom to top for placement. The default is `NP`.

GUI Equivalent

Command	*Options – SPD – Placement*
Field	*Row Pattern* (Placement Options)

Examples

```
envGetVal("layoutXL.symbolicPlacement" "rowPattern")
envSetVal("layoutXL.symbolicPlacement" "rowPattern" 'string "PN")
envSetVal("layoutXL.symbolicPlacement" "rowPattern" 'string "NPPN")
envSetVal("layoutXL.symbolicPlacement" "rowPattern" 'string "NPPN")
envSetVal("layoutXL.symbolicPlacement" "rowPattern" 'string "PNNP")
envSetVal("layoutXL.symbolicPlacement" "rowPattern" 'string "NPNP")
envSetVal("layoutXL.symbolicPlacement" "rowPattern" 'string "PNPN")
envSetVal("layoutXL.symbolicPlacement" "rowPattern" 'string "NNPP")
envSetVal("layoutXL.symbolicPlacement" "rowPattern" 'string "PPNN")
envSetVal("layoutXL.symbolicPlacement" "rowPattern" 'string "NNNP")
envSetVal("layoutXL.symbolicPlacement" "rowPattern" 'string "PNNN")
envSetVal("layoutXL.symbolicPlacement" "rowPattern" 'string "NPPP")
envSetVal("layoutXL.symbolicPlacement" "rowPattern" 'string "PPPN")
envSetVal("layoutXL.symbolicPlacement" "rowPattern" 'string "NNNN")
envSetVal("layoutXL.symbolicPlacement" "rowPattern" 'string "PPPP")
```

Related Topics

List of SPD Environment Variables

© 2023 Cadence Design Systems, Inc.

showPGFlightLines

```
layoutXL.symbolicPlacement showPGFlightLines boolean { t | nil }
```

Description

Specifies whether to display power and ground nets. The default is `t`.

GUI Equivalent

Command	*Options – SPD – Display*
Field	*Show P/G Nets* (Display Options)

Examples

```
envGetVal("layoutXL.symbolicPlacement" "showPGFlightLines")
envSetVal("layoutXL.symbolicPlacement" "showPGFlightLines" 'boolean nil)
```

Related Topics

List of SPD Environment Variables

© 2023 Cadence Design Systems, Inc.

sigTrunkLr

```
layoutXL.symbolicPlacement sigTrunkLr string { layerName purposeName }
```

Description

Specifies the layer purpose used to create the signal trunks when generating the layout.

GUI Equivalent

Command	*Options – SPD – Generation – Power Rails - Signal Trunks*
Field	*Signal Layer* (Power Rails Options)

Examples

```
envGetVal("layoutXL.symbolicPlacement" "sigTrunkLr")
envSetVal("layoutXL.symbolicPlacement" "sigTrunkLr" 'string "Metal1 drawing")
```

Related Topics

List of SPD Environment Variables

Reordering and Resetting Trunks

© 2023 Cadence Design Systems, Inc.

sigTrunkSpacingT2O

```
layoutXL.symbolicPlacement sigTrunkSpacingT2O float floating_point_number
```

Description

Specify the trunk-to-oxide-diffusion spacing when generating the layout.

GUI Equivalent

Command	*Options – SPD – Generation – Power Rails - Signal Trunks*
Field	*Signal Trunk-OD* (Power Rails Options)

Examples

```
envGetVal("layoutXL.symbolicPlacement" "sigTrunkSpacingT2O")
envSetVal("layoutXL.symbolicPlacement" "sigTrunkSpacingT2O" 'float 0.1)
```

Related Topics

List of SPD Environment Variables

Reordering and Resetting Trunks

© 2023 Cadence Design Systems, Inc.

sigTrunkSpacingT2T

`layoutXL.symbolicPlacement sigTrunkSpacingT2T float` *`floating_point_number`*

Description

Specify the trunk-to-trunk spacing when generating the layout.

GUI Equivalent

Command	*Options – SPD – Generation – Power Rails – Signal Trunks*
Field	*Signal Trunk-Trunk* (Power Rails Options)

Examples

```
envGetVal("layoutXL.symbolicPlacement" "sigTrunkSpacingT2T")
envSetVal("layoutXL.symbolicPlacement" "sigTrunkSpacingT2T" 'float 0.1)
```

Related Topics

List of SPD Environment Variables

Reordering and Resetting Trunks

© 2023 Cadence Design Systems, Inc.

sigTrunkWidth

`layoutXL.symbolicPlacement sigTrunkWidth float` *`floating_point_number`*

Description

Specifies the width of signal trunk when generating the layout.

GUI Equivalent

Command	*Options – SPD – Generation – Power Rails - Signal Trunks*
Field	*Signal Width* (Power Rails Options)

Examples

```
envGetVal("layoutXL.symbolicPlacement" "sigTrunkWidth")
envSetVal("layoutXL.symbolicPlacement" "sigTrunkWidth" 'float 0.1)
```

Related Topics

List of SPD Environment Variables

Reordering and Resetting Trunks

© 2023 Cadence Design Systems, Inc.

smartMoveInsertOnePos

```
layoutXL.symbolicPlacement smartMoveInsertOnePos boolean { t | nil }
```

Description

Allows moving selected instances to one location.

If `t`, SPD insert instances according to the X-coordinate of the reference object.

If `nil`, SPD inserts instances according to the X-coordinate of the position where the moved objects are dropped.

Default is `t`.

GUI Equivalent

Command	*Options – SPD – Placement*
Field	*Move to insert in one location* (Placement Options)

Examples

```
envGetVal("layoutXL.symbolicPlacement" "smartMoveInsertOnePos")
envSetVal("layoutXL.symbolicPlacement" "smartMoveInsertOnePos" 'boolean nil)
```

Related Topics

List of SPD Environment Variables

Moving Multiple Instances

© 2023 Cadence Design Systems, Inc.

smartMoveKeepChains

```
layoutXL.symbolicPlacement smartMoveKeepChainsboolean { t | nil }
```

Description

Retains the existing instance ordering when instances are moved. The default is `nil`, which means that SPD may flip or permute the instances to maximize the abutment.

GUI Equivalent

None

Examples

```
envGetVal("layoutXL.symbolicPlacement" "smartMoveKeepChains")
envSetVal("layoutXL.symbolicPlacement" "smartMoveKeepChains" 'boolean t)
```

Related Topics

List of SPD Environment Variables

© 2023 Cadence Design Systems, Inc.

spacingNStack

`layoutXL.symbolicPlacement spacingNStack float` *`floating_point_number`*

Description

Specifies the spacing for N stack devices. The default is `-0.1`, which means that the software assigns *N Stacking*, the minimum poly spacing specified in the technology file.

Note: A negative number other than `-0.1` will be adjusted to `0` automatically.

GUI Equivalent

Command	*Options – SPD – Generation – Stack*
Field	*N Stacking* (Labels Options)

Examples

```
envGetVal("layoutXL.symbolicPlacement" "spacingNStack")
envSetVal("layoutXL.symbolicPlacement" "spacingNStack" 'float 0.06)
```

Related Topics

List of SPD Environment Variables

© 2023 Cadence Design Systems, Inc.

spacingOd

```
layoutXL.symbolicPlacement spacingOd float floating_point_number
```

Description

Specifies the OD spacing between chains. The default is `0.0`, which means that the software uses the minimum oxide diffusion spacing from the technology file.

This option is effective only when the spacingOdMode environment variable is set to `Fixed`.

GUI Equivalent

Command	*Options – SPD – Generation*
Field	Field under *X: OD* (Generation Options)

Examples

```
envGetVal("layoutXL.symbolicPlacement" "spacingOd")
envSetVal("layoutXL.symbolicPlacement" "spacingOd" 'float 0.1)
```

Related Topics

List of SPD Environment Variables

© 2023 Cadence Design Systems, Inc.

spacingOdMode

```
layoutXL.symbolicPlacement spacingOdMode cyclic { "Fixed" | "DummyPoly" |
     "UserFunction" }
```

Description

Specifies how OD spacing is decided. The default is `Fixed`.

- `Fixed` lets you specify the oxide diffusion spacing between chains.
- `DummyPoly` lets the software automatically calculate the required oxide diffusion spacing between two unabutted instances.
- `UserFunction` specifies a user callback SKILL function that calculates the required oxide diffusion spacing between two unabutted instances.

GUI Equivalent

Command	*Options – SPD – Generation*
Field	*X: OD* (Generation Options)

Examples

```
envGetVal("layoutXL.symbolicPlacement" "spacingOdMode")
envSetVal("layoutXL.symbolicPlacement" "spacingOdMode" 'cyclic "DummyPoly")
envSetVal("layoutXL.symbolicPlacement" "spacingOdMode" 'cyclic "UserFunction")
```

Related Topics

List of SPD Environment Variables

© 2023 Cadence Design Systems, Inc.

spacingOdUserFunc

```
layoutXL.symbolicPlacement spacingOdUserFunc string { userFuncName }
```

Description

Specifies a user callback SKILL function that calculates the required oxide diffusion spacing between two unabutted instances.

Syntax for User Callback

```
userOdSpaceProc(
     d_lhsInst
     d_rhsInst
     )
     => n_odSpacing
```

Arguments

d_lhsInst	The left hand side instance.
d_rhsInst	The right hand side instance.

Value Returned

n_odSpacing	Returns the required spacing between the oxide diffusion of left-hand-side and right-hand-side instances.

This option is effective only when the value of *Options – SPD – Generation – X: OD* option in GUI or the spacingOdMode environment variable is set to `UserFunction`.

GUI Equivalent

Command	*Options – SPD – Generation*
Field	The field under *X: OD* (Generation Options)

Examples

```
envGetVal("layoutXL.symbolicPlacement" "spacingOdUserFunc")
envSetVal("layoutXL.symbolicPlacement" "spacingOdUserFunc" 'string MyOdSpcProc)
```


© 2023 Cadence Design Systems, Inc.

```
procedure(MyOdSpcProc(lInst rInst)
  let((odSpcWithPoly minOd minPoly)
    minOd = 0.038 ; minimum oxide diffusion spacing constraint
    minPoly = 0.068 ; minimum poly spacing constraint
    odSpcWithPoly = spdCalcOdSpacingWithPoly(lInst rInst minOd minPoly)
    info("spdCalcOdSpacingWithPoly: lInst %s rInst %s => %f\n" lInst~>name
rInst~>name odSpcWithPoly)
    odSpcWithPoly
  )
)
```

Related Topics

List of SPD Environment Variables

© 2023 Cadence Design Systems, Inc.

spacingPodNod

`layoutXL.symbolicPlacement spacingPodNod float` *`floating_point_number`*

Description

Specifies the spacing within a pair of rows. The default is `0`, which means that the software assigns *Within Row-pair* a fixed value of `1.0`.

GUI Equivalent

Command	*Options – SPD – Generation*
Field	*Within Row-pair* (Generation Options)

Examples

```
envGetVal("layoutXL.symbolicPlacement" "spacingPodNod")
envSetVal("layoutXL.symbolicPlacement" "spacingPodNod" 'float 0.3)
```

Related Topics

List of SPD Environment Variables

© 2023 Cadence Design Systems, Inc.

spacingPStack

`layoutXL.symbolicPlacement spacingPStack float` *`floating_point_number`*

Description

Specifies the spacing for P stack devices. The default is `0.1`, which means that the software assigns *P Stacking* the minimum poly spacing of the technology file.

Note: A negative number other than `-0.1` is adjusted to `0` automatically.

GUI Equivalent

Command	*Options – SPD – Generation – Stack*
Field	*P Stacking* (Labels Options)

Examples

```
envGetVal("layoutXL.symbolicPlacement" "spacingPStack")
envSetVal("layoutXL.symbolicPlacement" "spacingPStack" 'float 0.06)
```

Related Topics

List of SPD Environment Variables

© 2023 Cadence Design Systems, Inc.

spacingVddPod

```
layoutXL.symbolicPlacement spacingVddPod float floating_point_number
```

Description

Specifies the spacing between a power rail and a P-type oxide diffusion (POD). The default is `0.0`, which means that the software assigns *Power-POD* the minimum spacing for the layer used for creating power rails against the oxide diffusion layer.

GUI Equivalent

Command	*Options – SPD – Generation – Power*
Field	*Power-POD* (Generation Options)

Examples

```
envGetVal("layoutXL.symbolicPlacement" "spacingVddPod")
envSetVal("layoutXL.symbolicPlacement" "spacingVddPod" 'float 0.3)
```

Related Topics

List of SPD Environment Variables

© 2023 Cadence Design Systems, Inc.

spacingVssNod

`layoutXL.symbolicPlacement spacingVssNod float` *`floating_point_number`*

Description

Specifies the spacing between a power rail and an N-type oxide diffusion (NOD). The default is `0.0`, which means that the software assigns *Ground-NOD* the minimum spacing for the layer used for creating ground rails against the oxide diffusion layer.

GUI Equivalent

Command	*Options – SPD – Generation – Power*
Field	*Ground-NOD* (Generation Options)

Examples

```
envGetVal("layoutXL.symbolicPlacement" "spacingVddNod")
envSetVal("layoutXL.symbolicPlacement" "spacingVddNod" 'float 0.3)
```

Related Topics

List of SPD Environment Variables

© 2023 Cadence Design Systems, Inc.

spdDesignLibName

```
layoutXL.symbolicPlacement spdDesignLibName string { "cdsTmpSymbolicLib"}
```

Description

Specifies a name for the new temporary library to be used by SPD instead of `cdsTmpSymbolicLib`, which is created by default.

GUI Equivalent

None

Examples

```
envGetVal("layoutXL.symbolicPlacement" " spdDesignLibName ")
```

The following example shows how you can create a new library and then use `spdDeisignLibName` to specify this library to be used by SPD:

1. Create a new library named `mylib` using the Library Manager.

   Definition for this library is automatically added to cds.lib.

   Note: If the library already exists, for example `/user/mylib/`, then manually define it in cds.lib by adding the following line:

   ```
   DEFINE mylib  /user/mylib
   ```

2. Next, open Technology File Manager to attach `cdsSymbolicDevice` to `mylib`.
3. Set the environment variable `spdDesignLibName` to use `mylib`.

   ```
   envSetVal("layoutXL.symbolicPlacement" " spdDesignLibName " 'string "mylib")
   ```

Related Topics

List of SPD Environment Variables

Libraries Defined to Execute SPD

© 2023 Cadence Design Systems, Inc.

spdDummyLengthOptions

```
layoutXL.symbolicPlacement spdDummyLengthOptions cyclic { "" | "CDF Default" |
     "Same As Neighbor" | "Specify" }
```

Description

Specifies the default value for the length of a dummy. The default is `""`, which means that the software will assign `CDF Default` as the value for *Length*.

- `CDF Default` sets the length to be the same as the default CDF parameter values. This is the default setting.
- `Same as Neighbor` sets the length to be the same as the neighboring instance.
- `Specify` lets you specify the length manually.

GUI Equivalent

Command	*Add Dummy – Dummy Options*
Field	*Length* (Dummy Options)

Examples

```
envGetVal("layoutXL.symbolicPlacement" "spdDummyLengthOptions")
envSetVal("layoutXL.symbolicPlacement" "spdDummyLengthOptions" 'cyclic "CDF
Default")
envSetVal("layoutXL.symbolicPlacement" "spdDummyLengthOptions" 'cyclic "Same As
Neighbor")
envSetVal("layoutXL.symbolicPlacement" "spdDummyLengthOptions" 'cyclic "Specify")
```

Related Topics

List of SPD Environment Variables

© 2023 Cadence Design Systems, Inc.

spdDummyLengthValue

```
layoutXL.symbolicPlacement spdDummyLengthValue string { lengthValue }
```

Description

Sets a the default value for the length of a dummy. The default is `""`, which means that you must input a value in the GUI.

Note: Unit of the length also needs to be specified.

This option works when the value of *Add Dummy – Dummy Options – Length* option or the spdDummyLengthOptions environment variable is *Specify.*

GUI Equivalent

Command	*Add Dummy – Dummy Options*
Field	*Length – Specify* (Dummy Options)

Examples

```
envGetVal("layoutXL.symbolicPlacement" "spdDummyLengthValue")
envSetVal("layoutXL.symbolicPlacement" "spdDummyLengthValue" 'string "45n")
```

Related Topics

List of SPD Environment Variables

© 2023 Cadence Design Systems, Inc.

spdDummyNet

```
layoutXL.symbolicPlacement spdDummyNet string "net_name"
```

Description

Specifies the default net to use when creating dummy devices. The default is `""`, which means that the software automatically chooses a net from the existing nets in the design.

GUI Equivalent

Command	*Add Dummy – Dummy Options*
Field	*Net* (Dummy Options)

Examples

```
envGetVal("layoutXL.symbolicPlacement" "spdDummyNet")
envSetVal("layoutXL.symbolicPlacement" "spdDummyNet" 'string "VSS")
```

Related Topics

List of SPD Environment Variables

© 2023 Cadence Design Systems, Inc.

spdDummyNumFingersOptions

```
layoutXL.symbolicPlacement spdDummyNumFingersOptions cyclic ( "" | "CDF Default" |
     "Same As Neighbor" | "Specify" )
```

Description

Specifies default value for the number of fingers for a dummy. The default is `""`, which means that the software will assign `CDF Default` as the value for *Number of Fingers*.

- `CDF Default` sets the number of fingers to be the same as the default CDF parameter values.
- `Same as Neighbor` sets number of fingers to be the same as the neighboring instances.
- `Specify` lets you specify the number of fingers manually.

GUI Equivalent

Command	*Add Dummy – Dummy Options*
Field	*Number of Fingers* (Dummy Options)

Examples

```
envGetVal("layoutXL.symbolicPlacement" "spdDummyNumFingersOptions")
envSetVal("layoutXL.symbolicPlacement" "spdDummyNumFingersOptions" 'cyclic "CDF
Default")
envSetVal("layoutXL.symbolicPlacement" "spdDummyNumFingersOptions" 'cyclic "Same
As Neighbor")
envSetVal("layoutXL.symbolicPlacement" "spdDummyNumFingersOptions" 'cyclic
"Specify")
```

Related Topics

List of SPD Environment Variables

© 2023 Cadence Design Systems, Inc.

spdDummyNumFingersValue

```
layoutXL.symbolicPlacement spdDummyNumFingersValue string "numFingers"
```

Description

Specifies default value for the number of fingers for a dummy. The default is `" "`, which means that you must input a value in the GUI.

This option works when the value of *Add Dummy – Dummy Options – Number of Fingers* option or the spdDummyNumFingersOptions environment variable is *Specify.*

GUI Equivalent

Command	*Add Dummy – Dummy Options*
Field	*Number of Fingers* (Dummy Options)

Examples

```
envGetVal("layoutXL.symbolicPlacement" "spdDummyNumFingersValue")
envSetVal("layoutXL.symbolicPlacement" "spdDummyNumFingersValue" 'string "2")
```

Related Topics

List of SPD Environment Variables

© 2023 Cadence Design Systems, Inc.

spdDummyWidthOptions

```
layoutXL.symbolicPlacement spdDummyWidthOptions cyclic { "" | "CDF Default" | "Same
     As Neighbor" | "Specify" }
```

Description

Specifies the default value for the width of a dummy. The default is `""`, which means that the software will assign the value `Same As Neighbor` for *Width*.

- `CDF Default` sets the width to be the same as the default CDF parameter values.
- `Same as Neighbor` sets the width to be the same as the neighboring instance.
- `Specify` lets you specify the width manually.

GUI Equivalent

Command	*Add Dummy – Dummy Options*
Field	*Width* (Dummy Options)

Examples

```
envGetVal("layoutXL.symbolicPlacement" "spdDummyWidthOptions")
envSetVal("layoutXL.symbolicPlacement" "spdDummyWidthOptions" 'cyclic "CDF
Default")
envSetVal("layoutXL.symbolicPlacement" "spdDummyWidthOptions" 'cyclic "Same As
Neighbor")
envSetVal("layoutXL.symbolicPlacement" "spdDummyWidthOptions" 'cyclic "Specify")
```

Related Topics

List of SPD Environment Variables

© 2023 Cadence Design Systems, Inc.

spdDummyWidthValue

```
layoutXL.symbolicPlacement spdDummyWidthValue string { widthValue }
```

Description

Sets the default value for the width of a dummy. The default is `""`, which means that you must enter a value in the GUI.

Note: Unit of the width also needs to be specified.

This option works only when the value of *Add Dummy – Dummy Options – Width* option in GUI or the spdDummyWidthOptions environment variable is *Specify.*

GUI Equivalent

Command	*Add Dummy – Dummy Options*
Field	*Width – Specify* (Dummy Options)

Examples

```
envGetVal("layoutXL.symbolicPlacement" "spdDummyWidthValue")
envSetVal("layoutXL.symbolicPlacement" "spdDummyWidthValue" 'string "150n")
```

Related Topics

List of SPD Environment Variables

© 2023 Cadence Design Systems, Inc.

spdMoveAfterAppend

```
layoutXL.symbolicPlacement spdMoveAfterAppend boolean { t | nil }
```

Description

Invokes the *Move* command immediately after the *Append Selected From Source* command is complete. The default is `t`.

GUI Equivalent

None

Examples

```
envGetVal("layoutXL.symbolicPlacement" "spdMoveAfterAppend")
envSetVal("layoutXL.symbolicPlacement" "spdMoveAfterAppend" 'boolean nil)
```

Related Topics

List of SPD Environment Variables

© 2023 Cadence Design Systems, Inc.

spdRememberDummyVals

```
layoutXL.symbolicPlacement spdRememberDummyVals boolean { t | nil }
```

Description

Specifies that the values from the Dummy Options form should be saved. The default is `nil`, which means the values are not saved.

GUI Equivalent

Command	*Add Dummy* – Dummy Options form
Field	*Remember Values* (Dummy Options)

Examples

```
envGetVal("layoutXL.symbolicPlacement" "spdRememberDummyVals")
envSetVal("layoutXL.symbolicPlacement" "spdRememberDummyVals" 'boolean t)
```

Related Topics

List of SPD Environment Variables

© 2023 Cadence Design Systems, Inc.

tabbedWindow

```
layoutXL.symbolicPlacement tabbedWindow boolean { t | nil }
```

Description

Launches SPD in a tabbed window. The default is `nil`.

GUI Equivalent

Command	*Options – SPD – Setup*
Field	*Tabbed window* (Setup Options)

Examples

```
envGetVal("layoutXL.symbolicPlacement" "tabbedWindow")
envSetVal("layoutXL.symbolicPlacement" "tabbedWindow" 'boolean t)
```

Related Topics

List of SPD Environment Variables

© 2023 Cadence Design Systems, Inc.

vddNetName

```
layoutXL.symbolicPlacement vddNetName string "net_name"
```

Description

Specifies the net name for the power rail. The default is `""`, which means that the net name is automatically detected.

This option is effective only when the *Options – SPD – Generation – Power Rails – Create* check box is selected in GUI or the value of the createPowerGroundRails environment variable is set to `t`.

GUI Equivalent

Command	*Options – SPD – Generation – Power Rails*
Field	*Power Net Name* (Power Rails Options)

Examples

```
envGetVal("layoutXL.symbolicPlacement" "vddNetName")
envSetVal("layoutXL.symbolicPlacement" "vddNetName" 'string "VDD")
```

Related Topics

List of SPD Environment Variables

© 2023 Cadence Design Systems, Inc.

vssNetName

```
layoutXL.symbolicPlacement vssNetName string "net_name"
```

Description

Specifies the net name for the ground rail. The default is `""`, which means that the net name is automatically detected.

This option is effective only when the *Options – SPD – Generation – Power Rails – Create* check box is selected in GUI or the value of the createPowerGroundRails environment variable is set to `t`.

GUI Equivalent

Command	*Options – SPD – Generation – Power Rails*
Field	*Ground Net Name* (Power Rails Options)

Examples

```
envGetVal("layoutXL.symbolicPlacement" "vssNetName")
envSetVal("layoutXL.symbolicPlacement" "vssNetName" 'string "VSS")
```

Related Topics

List of SPD Environment Variables

© 2023 Cadence Design Systems, Inc.

widthVdd

`layoutXL.symbolicPlacement widthVdd float` *`floating_point_number`*

Description

Specifies width for the power rail. The default is `0.0`, which means that the software assigns the minimum width defined for the power layer.

This option is effective only when the *Options – SPD – Generation – Power Rails – Create* check box is selected in GUI or the value of the createPowerGroundRails environment variable is set to `t`.

GUI Equivalent

Command	*Options – SPD – Generation – Power Rails*
Field	*Power Width* (Power Rails Options)

Examples

```
envGetVal("layoutXL.symbolicPlacement" "widthVdd")
envSetVal("layoutXL.symbolicPlacement" "widthVdd" 'float 0.1)
```

Related Topics

List of SPD Environment Variables

© 2023 Cadence Design Systems, Inc.

widthVss

`layoutXL.symbolicPlacement widthVss float` *`floating_point_number`*

Description

Specifies width for the ground rail. The default is `0.0`, which means that the software assigns the minimum width defined for ground layer.

This option is effective only when the *Options – SPD – Generation – Power Rails – Create* check box is selected in GUI or the value of the createPowerGroundRails environment variable is set to `t`.

GUI Equivalent

Command	*Options – SPD – Generation – Power Rails*
Field	*Ground Width* (Power Rails Options)

Examples

```
envGetVal("layoutXL.symbolicPlacement" "widthVss")
envSetVal("layoutXL.symbolicPlacement" "widthVss" 'float 0.1)
```

Related Topics

List of SPD Environment Variables

© 2023 Cadence Design Systems, Inc.

C

Symbolic Placement of Devices Bindkeys

To view the default bindkeys defined for commands in SPD Editor and SPD Preview, see Bindkeys in Virtuoso Layout Suite.

© 2023 Cadence Design Systems, Inc.

© 2023 Cadence Design Systems, Inc.

D

Symbolic Placement of Devices Videos

This section lists videos related to SPD that you can access on Cadence Online Support.

- **Getting Started with Symbolic Placement of Devices**

  This video shows how to use Virtuoso Symbolic Placement of Devices to quickly edit selected PMOS and NMOS devices in a design, preview the updates made, and then generate a layout.

- **Performing Multirow Placement in SPD**

  This video shows how using multiple row patterns in SPD helps in optimizing a design.

- **Creating Signal Trunks in SPD for Pin-to-Trunk Routing**

  This video shows how to create signal trunks in SPD and then use them for Pin-to-Trunk routing in Layout XL.

- **Using User-Defined Abutment and Callback Functions in SPD**

  This video shows how user-defined abutment and callback functions can be used in SPD.

© 2023 Cadence Design Systems, Inc.