cadence®

Simulation Driven Interactive Routing Task Assistant

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. (Cadence) 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

1

Simulation Driven Interactive Routing Task Assistant

The simulation driven interactive routing flow extends the traditional electrically-aware design flow into a fully simulation-driven routing flow. Using this flow, you can capture current data in simulation datasets, combine it with parasitic extraction information (in the form of an ICT file) or High Current design intent information, and then use it to perform electromigration checks on the physical implementation of the design and update the layout to correct any violations."

SDR provides a powerful new way to satisfy current density constraints and improve the productivity and design reliability.

Related Topics

Getting Started

How do I use the SDR toolbar?

How do the visuals displayed on the layout canvas during interactive SDR help us?

What are the various checker modes available in interactive SDR?

What current estimation modes are available in interactive SDR?

What tasks can I perform using the SDR toolbar?

Getting Started

What is Simulation Driven Routing (SDR)?

What are the prerequisites for running simulation-driven interactive routing?

How does SDR fit into the layout design flow?

How do the visuals displayed on the layout canvas during interactive SDR help us?

What is Simulation Driven Routing (SDR)?

The Virtuoso simulation-driven routing (SDR) solution is a step towards correct-by-construction routing driven by electrical requirements. It provides an environment to consider current density and maximum resistance design rules and automatically resizes wires and vias during interactive routing.
The key features of Virtuoso simulation-driven routing let you:

- visualize the current distribution per net.
- control simulation-driven routing to calculate the current according to the net topology.
- automatically resize wires and vias according to the estimated current.
- automatically connect devices according to the estimated current.

Related Topics

What are the prerequisites for running simulation-driven interactive routing?

How does SDR fit into the layout design flow?

Getting Started

What are the prerequisites for running simulation-driven interactive routing?

Simulation-driven interactive routing is performed using the current of a net terminal coupled with EM reliability rules (ICT file).
The minimum set of files and data that you need to use SDR are as follows:

- ICT file with EM rules (same as Voltus-Fi)
  If only an ICT file is provided, SDR can run and display the maximum current passing through a wire. However, if the current of a terminal is also provided, SDR resizes the wire according to the simulation data. The terminal current extracted after simulation is generated using ADE.
- Simulation dataset
  The dataset can be generated from ADE Assembler with Spectre or a third-party simulator.
- .CSV file with terminal currents

The terminal currents can also be initialized manually in the EAD assistant tables or imported from a .csv file. Alternatively, you can specify the terminal currents directly in the EAD browser.

Related Topics

What is Simulation Driven Routing (SDR)?

How does SDR fit into the layout design flow?

Getting Started

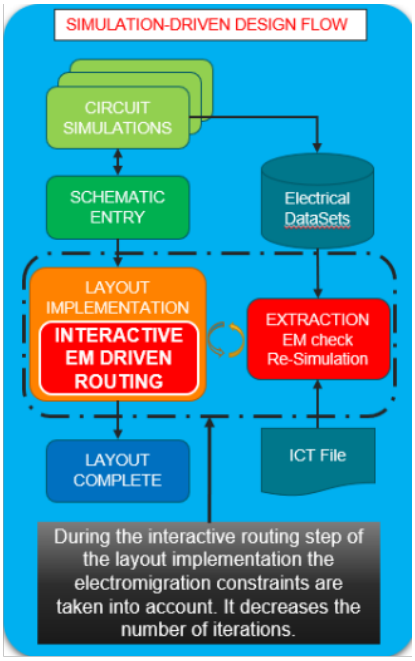
Editing Process Settings

Creating Datasets

Viewing Datasets

How does SDR fit into the layout design flow?

Before you start working with the SDR solution, ensure that EAD is initialized with an EAD setup file that defines the location of the ICT file and the EAD dataset and the analysis settings. When SDR is run using the setup information, it checks whether the specified EM data source files (EM data file and ICT file) exist and are readable.
The following diagram provides a snapshot of how SDR fits into the layout design flow.



In the SDR flow, the parasitic extraction and electromigration checks happen concurrently while the layout is implemented. In case of a classical flow (without SDR), the electrical impact is unknown until the layout is complete. In simulation-driven routing, during layout implementation, electromigration (EM) constraints are considered, which in turn, decreases the number of iterations.

Related Topics

What is Simulation Driven Routing (SDR)?

What are the prerequisites for running simulation-driven interactive routing?

Getting Started

How do I use the SDR toolbar?

SDR is an effective way to find and display nets with accurate current and resistance estimation. The SDR toolbar is designed to help you with the various simulation driven interactive routing features and settings. The SDR toolbar is installed automatically when you launch Layout EXL from a layout or schematic view. If the toolbar is not available by default in Layout EXL, you can open it using *Windows – Toolbars – SDR Toolbar*. The icons on the SDR toolbar let you access the SDR features with a single click.

Related Topics

What tasks can I perform using the SDR toolbar?

SDR Toolbar 

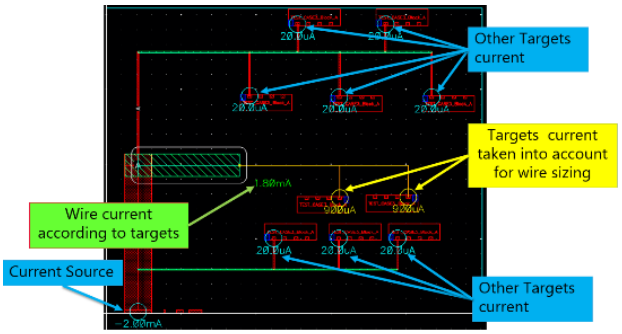
How do the visuals displayed on the layout canvas during interactive SDR help us?

When you start the *Create Wire* or the *Create Stranded Wire* command, the *Interactive Routing Bindkeys* info balloon is displayed at the top-left corner of the layout canvas. This info balloon shows specific bindkeys related to SDR and a legend to explain the visuals that are displayed in the layout canvas during simulation-driven routing.

The following visuals are shown on the layout canvas during interactive routing:

- Thermal map representation

- Yellow and blue halos around connected pins and current contributors
- Estimated current value close to the cursor
- Connection style
- Legend to show pin colors source and sink



You can minimize the *Interactive Routing Bindkeys* info balloon and then maximize it when required. However, if you close the info balloon, it does not appear again. To view the info balloon again, you must go to another layout tier and then reopen Layout EXL.

Related Topics

About SDR Bindkeys and Visuals

What are the various checker modes available in interactive SDR?

Specify a checker mode to be used for current and resistance estimation. The checker modes that are available in interactive SDR are as follows:

- **Enforce**
  Estimates the current in the edited wire and vias and automatically calculates the wire width to avoid EM or maximum resistance violations.
- **Notify**
  Estimates the current in the edited wire using a color coding based on EAD EM violations color settings.
- **Off**
  Does not update the wire width and there is no feedback on the estimated current or

resistance.

Related Topics

What are the differences in the Enforce checker mode when specified for current and resistance estimation?

What are the differences in the Notify checker mode when specified for current and resistance estimation?

What are the differences in the Enforce checker mode when specified for current and resistance estimation?

The Enforce checker mode behaves differently when specified for current estimation using the *Create Wire* and *Create Stranded Wire* commands and for resistance estimation.

- **Result of *Enforce* Mode Using the *Create Wire* Command**
  The width of the wire and vias is automatically adjusted to avoid EM violations. The adjusted width respects the *minWidth* constraint and the width defined in the Wire Assistant. When starting from a pin, if the wire width required to avoid EM violations is smaller than the pin width and if the *Use Width – Tap Shapes and Pins* Width option in the *Create Wire* context-sensitive menu is enabled, then the wire matches the pin width. However, if the *Tap Shapes and Pins Width* option is disabled, or the width required to avoid EM violations is larger than the pin width, then the estimated width is used.
  The estimated current of the wire is displayed using the color coding based on the color settings for EAD EM violations. The color is computed to represent the EM violation that will be reported by the EAD checker after routing. Text in green indicates that there is no EM violation and the current density check is satisfied.
- **Result of *Enforce* Mode Using the *Create Stranded Wire* Command**
  The number of stranded wires is automatically adjusted to avoid EM violations. The estimated current of the wire is displayed using the color coding based on the color settings for EAD EM violations. The color is computed to represent the EM violation that is reported by the EAD checker after routing. Text in green indicates that there is no EM violation and the current density check is satisfied.
  If you change the number of stranded wires, then the checker mode is automatically updated to *Notify*. Also, the label color of the estimated current of the stranded wires is changed. The label color of the estimated current indicates that the number of stranded wires is either more or less than required.

- **Result of *Enforce* Mode Using Maximum Resistance**
  Prevents *maxResistance* violations and automatically adjusts the wire and vias. The estimated resistance from the starting pin up to the mouse pointer is displayed using a color coding based on the color settings for EAD EM violations. The color is computed to represent the *maxResistance* violation that is reported by the EAD checker after routing. In the *Enforce* mode, the width of the wire and via is automatically updated to prevent the *maxResistance* violation, while enforcing the *minWidth* and *maxWidth* constraints. Text in green indicates that the *maxResistance* constraint is satisfied.

Related Topics

What are the differences in the Notify checker mode when specified for current and resistance estimation?

What are the various checker modes available in interactive SDR?

What are the differences in the Notify checker mode when specified for current and resistance estimation?

The Notify checker mode behaves differently when specified for current estimation and when specified for resistance estimation.

- **Result of *Notify* Mode Using the *Create Wire* Command**
  The default width of the wire is considered. You can also specify the width of the wire in the Create Wire form. The wire and vias provide feedback using the label color, which displays the estimated current in the edited wire. The width of the wire and via are not updated automatically and the EM violation can continue to exist. Text in red reports an EM violation.
- **Result of *Notify* Mode Using the *Create Stranded Wire* Command**
  The required number of stranded wires is displayed without changing the number of stranded wires. Because the number of stranded wires is not changed automatically, the EM violation continues to exist. You can specify the number of stranded wires in the Create Stranded Wire form. The stranded wires provide feedback using the label color, which displays the required number of stranded wires for the estimated current. Text in red reports an EM violation.
- **Result of *Notify* Mode Using in Maximum Resistance**
  When the Notify option is selected, you can choose the size of the wire and vias. The EM checker provides feedback using the label color, which displays the estimated resistance from the starting pin up to the mouse pointer. The color coding is based on the color settings for

EAD EM violations. The color is computed to represent the *maxResistance* violation that is reported by the EAD checker after routing. The width of the wire and via are not updated automatically, and the *maxResistance* violation continues to exist. Text in red reports a *maxResistance* constraint violation.

Related Topics

What are the differences in the Enforce checker mode when specified for current and resistance estimation?

What are the various checker modes available in interactive SDR?

What current estimation modes are available in interactive SDR?

You can control how the *Create Wire* command estimates the current in the edited pathSeg and via. The current estimation modes that are available in interactive SDR are as follows:

- ***Auto***
  Estimates the current automatically in the last section of the wire according to the connected objects and flightline targets.
- ***Sum Connected Pins Currents***
  Estimates the current after adding the current of all the connected pins.
- ***Maintain Constant Current***
  Estimates the current after inheriting the current from the previous wire.
- ***Nearest Island Current***
  Estimates the current based on all the pins connected by the flightline.

Related Topics

What are the differences in current estimation using the Auto current estimation mode?

What are the differences in current estimation using the Maintain Constant Current estimation mode?

What are the differences in current estimation using the Nearest Island Current estimation mode?

What are the differences in current estimation using the Sum Connected Pins Currents estimation

mode?

What are the differences in current estimation using the Auto current estimation mode?

The results of current estimation for the Auto current estimation mode are different when using the *Create Wire* and *Create Stranded Wire* commands.

- **Current Estimation Using the *Create Wire* Command**
  Calculates the current based on all the connected target pins identified in the channel separating two rows of instances or nmos and pmos devices.
  When entering a channel, the flightlines are displayed to show all the identified targets in the channel. The current estimation is based on the sum of all targets, for example, 706uA = 4 * 177uA, assuming the wire is routed only until the last target and then restarted from the bend. When passing through the last target of the channel, the current estimation is based on the sum of all connected pins, for example, 945uA = 4 * 177uA + 239uA. The wire transfers all the current attached to the wire to the right.
- **Current Estimation Using the *Create Stranded Wire* Command**
  Calculates the current based on all the connected target pins identified in the channel separating two rows of instances. When entering a channel, the flightlines are displayed to show all the identified targets in the channel. The current estimation is based on the sum of all targets, for example, 300uA = 2 * 150uA, assuming the wire is routed only until the last target and then restarted from the bend.

Related Topics

What are the differences in current estimation using the Maintain Constant Current estimation mode?

What are the differences in current estimation using the Nearest Island Current estimation mode?

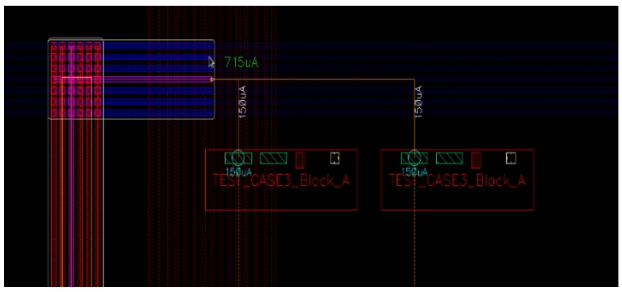
What are the differences in current estimation using the Sum Connected Pins Currents estimation mode?

What current estimation modes are available in interactive SDR?

What are the differences in current estimation using the Maintain Constant Current estimation mode?

The results of current estimation are different for the *Maintain Constant Current* estimation mode when using the *Create Wire* and *Create Stranded Wire* commands.

- **Current Estimation Using the *Create Wire* Command**
  Adjusts the wire width based on the current of the previous segment. The vertical pathSeg is created with the current of the connected pin. The horizontal pathSeg is also adjusted to maintain the same current as the vertical pathSeg.
- **Current Estimation Using the *Create Stranded Wire* Command**
  Adjusts the number of stranded wires based on the current of the previous segments. The number of stranded wires is adjusted when the current estimation mode is *Maintain Constant Current* and the checker mode is *Enforce*. The vertical stranded wires in red gets created for a current of 715uA. Thus, the stranded wires in blue are adjusted for the same amount of current, which is 715uA.



Related Topics

What are the differences in current estimation using the Auto current estimation mode?

What are the differences in current estimation using the Nearest Island Current estimation mode?

What are the differences in current estimation using the Sum Connected Pins Currents estimation mode?

What current estimation modes are available in interactive SDR?

What are the differences in current estimation using the Nearest Island Current estimation mode?

The result of current estimation for the Nearest Island Current estimation mode using the *Create Wire* and *Create Stranded Wire* commands.

- **Current Estimation Using the *Create Wire* Command**
  Current of the closest pin is considered to estimate the width of the wire.
- **Current Estimation Using the *Create Stranded Wire* Command**
  Calculates the number of stranded wires based on the current of the closest pin.

Related Topics

What are the differences in current estimation using the Auto current estimation mode?

What are the differences in current estimation using the Maintain Constant Current estimation mode?

What are the differences in current estimation using the Sum Connected Pins Currents estimation mode?

What current estimation modes are available in interactive SDR?

What are the differences in current estimation using the Sum Connected Pins Currents estimation mode?

The result of current estimation for the Sum Connected Pins Currents using the *Create Wire* and *Create Stranded Wire* commands.

- **Current Estimation Using the *Create Wire* Command**
  Keeps the wire width based on the estimated current of all the pins already connected to the wire. The current at the end of the last segment is the sum of the current for all connected pins. For example, 768uA = 239uA + 3 * 177uA.
- **Current Estimation Using the *Create Stranded Wire* Command**
  Keeps the number of stranded wires based on the estimated current of all the pins already connected to the wire. The current at the end of the last segment is the sum of current for all the connected pins. For example, 650uA, if there is only one pin producing 650uA of current connected to the wire.

Related Topics

What are the differences in current estimation using the Auto current estimation mode?

What are the differences in current estimation using the Maintain Constant Current estimation mode?

What are the differences in current estimation using the Nearest Island Current estimation mode?

What current estimation modes are available in interactive SDR?

What tasks can I perform using the SDR toolbar?

During interactive routing, you use SDR for two modes of routing checks, one based on the current density and the other on the maximum resistance. The following are the tasks for interactive SDR checks to avoid EM and maximum resistance violations in the design while performing simulation-driven interactive routing:

- Visualize the Current Distribution
  Identifies the net connections that either consume or generate the most current.
- Run the Current Density Check
  Estimates the current in the edited wire and lets you create a design with the appropriate width of wire based on the estimated EM value.
- Run the Maximum Resistance Check
  Connects the pins interactively and automatically estimates the final resistance between the two pins.
- Connect Twigs
  Connects multiple pins automatically with the appropriate wires and vias.
- Taper Wires
  Adjusts the width of each segment independently.

Related Topics

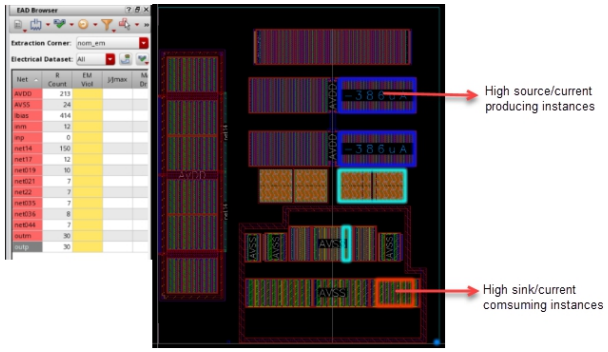
Does simulation driven interactive routing supports over-device routing?

How do I identify the net connections that either consume or generate the most current?

Before starting interactive SDR routing, it is important to identify the net connections that either consume or generate the most current. Identifying the main current consumers and producers helps you to estimate the wire widths and vias to create an EM-compliant topology.
To visualize the pins of a net according to their current distribution:

1. Select a net from the Navigator assistant or EAD Browser.
2. Click *Hide Sources and Sinks Map* on SDR Toolbar.

   The layout canvas displays the group of abutted instances of the selected net. The colored halo around the grouped instances indicates whether it is a current consumer or a producer. The pins or instance pins producing the current are shown with a blue halo, and the sinks, pins, or instance pins consuming the current are shown with a red halo. The dark blue and dark red indicate higher current; lighter colors indicate a lower current.



   Sometimes, you are unable to view the current value of the cluster. In this case, you can zoom in to the cluster and view the current value.

Related Topics

How do I perform a current density check in interactive SDR?

How do I perform resistance-driven interactive routing?

How do I connect twigs in interactive SDR?

How do I optimize each segment using SDR?

Does simulation driven interactive routing supports over-device routing?

What tasks can I perform using the SDR toolbar?

Sources and Sinks Maps

How do I perform a current density check in interactive SDR?

Based on the simulation results, *Create Wire* estimates the current in the edited wire and vias according to the EAD settings, such as dataset, temperature, and current scaling. The current density check lets you create a design with the appropriate width of the wire based on the estimated EM value.
To run the current density check:

1. Select a net in the Navigator assistant or the EAD browser.
2. Select an active layer in the Palette.
3. Click the drop-down arrow next to the Interactive SDR Mode button on SDR Toolbar.
4. Select the *Interactive SDR Current Density Check* option from the Interactive SDR Mode drop-down list.
5. To control the wire size and the number of strands compared to the estimated width required to avoid EM violations, specify a value in the *Width Multiplier* text box on the SDR toolbar. For example, if you specify the width multiplier as *2.0*, it means that the stranded width is multiplied by 2 compared to the estimated width to reach 100% of EM. Either the width of each strand is increased or the number of strands are increased.
6. Start to create a wire or a stranded wire from the selected pin or instance.
    a. Choose *Create – Wiring – Wire* or press *P*.
    b. Choose *Create – Wiring – Create Stranded Wire* or press *Ctrl + Shift + S*.
    When you start the *Create Wire* or the *Create Stranded Wire* command, the wires and vias interactively connect the pins and the software automatically estimates the current between the two pins.
    Based on the estimated current, the wire width and number of via cuts are automatically resized so that the estimated current of the wire is below the value of the final current of the wire.
7. Select a checker mode from the *Checker Mode* drop-down list. You can select one of the following checker modes: *Checker Mode: Enforce*, *Checker Mode: Notify*, or *Checker Mode: Off*.
8. To control the current estimation of wire, select a current estimation mode. You can select one of the following current estimation modes: *Auto*, *Sum Connected Pins Current*, *Maintain Constant Current*, and *Nearest Island Current*.

9. To complete interactive routing, press *Enter*.
10. Run EM checks as follows.
    a. Click *Extract Parasitics* on EAD Toolbar.
    b. Click *Run EM Checks* on EAD Toolbar.
11. To display the EM violations, click *Highlight EM violations* on the *EM* tab of the *Details* pane of the EAD Browser.
    This completes simulation-driven routing and EM checking for the design.

Related Topics

How do I identify the net connections that either consume or generate the most current?

How do I perform resistance-driven interactive routing?

How do I connect twigs in interactive SDR?

How do I optimize each segment using SDR?

Does simulation driven interactive routing supports over-device routing?

What tasks can I perform using the SDR toolbar?

How do I perform resistance-driven interactive routing?

When you start the *Create Wire* command, the wires and vias interactively connect the pins and automatically estimate the final resistance between the two pins. Based on the estimated resistance and the *maxResistance* constraint, the wire width and via num cuts of the wire and via being edited are automatically sized so that the final wire resistance is below the value of the *maxResistance* constraint.
The SDR Maximum Resistance Check mode is disabled for the Create Stranded Wire command. Even when the mode is set to maxResistance through the weSdrElectricalMode environment variable, the Create Stranded Wire command behaves as if the EM mode is enabled.
To run maximum resistance check:

1. Select a pin or an instance that you want to route.
2. Click the drop-down arrow next to *Interactive SDR Mode* on SDR Toolbar.
3. Select the *Interactive SDR Max Resistance Check* option from the drop-down list.
   Note: When the *Interactive SDR Max Resistance* option is selected, the icons on the SDR toolbar are updated with the icons for maximum resistance.

4. To control the scaling factor applied to the target resistance when sizing the wires or displaying the resistance information, specify a value in the *Max Resistance Target Percentage* text box on SDR Toolbar. For example, if you specify the maximum resistance percentage as 50%, it means that the wire is oversized to reach a resistance of 0.5*maxResistance.
5. Choose *Create – Wiring – Wire*.
6. Start creating a wire from the selected pin or the instance.
7. Click the drop-down arrow next to the *Checker Mode* icon on SDR Toolbar.
8. From the drop-down menu, select a checker mode. You can select one of the following checker modes: *Checker Mode: Enforce*, *Checker Mode: Notify*, or *Checker Mode: Off*.
9. To toggle between multiple *maxResistance* constraints, click the *Cycle Max Resistant Constraints* button on SDR Toolbar.
10. To complete interactive routing, press *Enter*.

Related Topics

How do I identify the net connections that either consume or generate the most current?

How do I perform a current density check in interactive SDR?

How do I connect twigs in interactive SDR?

How do I optimize each segment using SDR?

Does simulation driven interactive routing supports over-device routing?

What tasks can I perform using the SDR toolbar?

How do I connect twigs in interactive SDR?

In interactive SDR, you can identify the device pins (target pins identified by the flightlines) and automatically connect multiple pins with the appropriate wires and vias. The twig (wires from the device to the main wire) layer is automatically defined but can be changed. Also, the number of cuts in the vias and the width of the twigs are automatically adjusted.
The automatic twigs connection is independent of the current estimation mode and the checker modes.
To automatically connect twigs:

1. Select a net in the Navigator assistant or EAD Browser.

2. Select a source pin of the selected net in the layout design.
3. Choose *Create – Wiring – Wire*.
4. On SDR Toolbar, click the *Automatically Connect Twigs* button. The status of the button is modified to *Automatically Connect Twigs: ON*.
   When the mouse pointer moves close to an object on the same net, the twig connections are automatically created.
   By default, twigs are created on the layer above the layer on which the wire is created. However, you can change the twig layer at any time while creating the wire (See step 9).
5. Press *F3* to open the Create Wire form.
   By default, the *Snap to Pin Center* and *Cover Pin* options are deselected in the Create Wire form. In this case, the twigs are automatically created and snapped to the edge of the pin.
6. Select the *Snap to Pin Center* option in the Create Wire form. This automatically connects the twigs to the center of the pin of the created wire segment.
7. Select the *Cover Pin* option. This adjusts the twigs to automatically cover the source and target pins of the wire segment.
8. (Optional) Press the { key to automatically connect the twigs to the top-left or bottom-right targets of the wire. By default, the twigs are automatically connected to all targets. The following figure shows the automatic twig connection to the bottom targets.
9. Modify the twig layer by pressing the = key to move the twig layer up or the - key to move the twig layer down. When the twig layer is changed, the vias are automatically created on the wire being created.
10. To finish connecting the twigs automatically, press *Enter* or click the last twig connection.

    In addition, the *Create Wire* command identifies the *Must Connect All Pins* terminal instances and automatically makes twig connections for all the pins that must be connected.

Related Topics

How do I identify the net connections that either consume or generate the most current?

How do I perform a current density check in interactive SDR?

How do I perform resistance-driven interactive routing?

How do I optimize each segment using SDR?

Does simulation driven interactive routing supports over-device routing?

What tasks can I perform using the SDR toolbar?

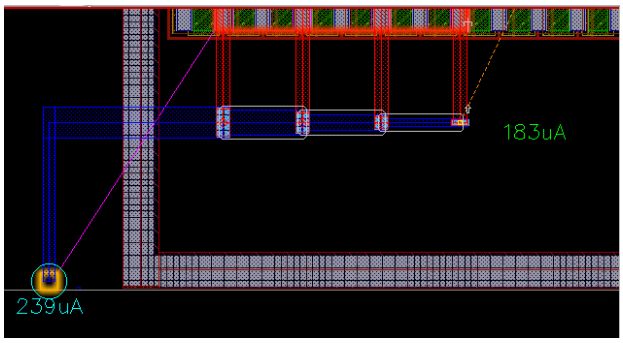
How do I optimize each segment using SDR?

Use the tapering feature to show how each segment can be optimized by SDR.

Note: Tapering is available only in *Enforce* checker mode for all current estimation modes.

To adjust the width and taper each segment independently:

1. Select a net from the Navigator Assistant or the EAD Browser.
2. Choose *Create – Wiring – Wire*.
   Note: The tapering is available only for the *Create Wire* command.
3. Select an instance from where you want to start creating the wire. By default, the taper mode is *noTaper*. This means that the wire width remains constant even after connecting the pins.
4. To enable tapering, press }.
   When tapering is enabled, the width of each segment is independently adjusted to match the estimated current value. The following figure shows how a wire width is adjusted when tapering is enabled.



When there are no additional connected shapes for the edited wire and you continue to move the mouse pointer in the same direction, then after crossing the last target, the tapering of the wire moves in the opposite direction.

Related Topics

How do I identify the net connections that either consume or generate the most current?

How do I perform a current density check in interactive SDR?

How do I perform resistance-driven interactive routing?

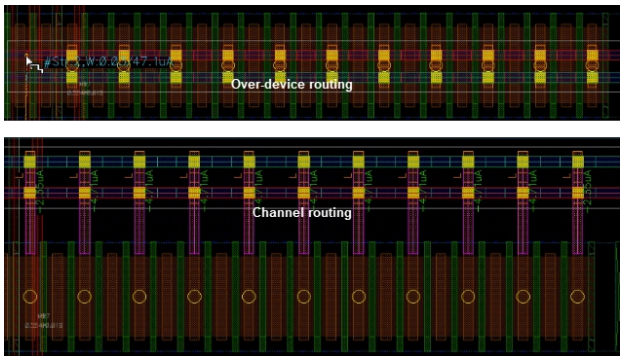
How do I connect twigs in interactive SDR?

Does simulation driven interactive routing supports over-device routing?

Yes, simulation driven interactive routing supports over-device routing.

With FinFET technology, the routing density is more important than other criteria. Devices (especially MOS) have a different structure than before and are less sensitive to over-device routes. Therefore, simulation-driven interactive routing supports over-device routing, which is automatically enabled when a trunk is over device pins.
To automatically route over devices:

1. Choose *Create – Wiring – Wire*.
2. Select a location in line with the instance to start creating a wire.
3. While creating a wire, route over the device. The trunk is automatically connected to the device pins by adding vias.
   If both sides of the channel are connected, the connections to the devices that are not overlapped are removed. In addition, the current from them is ignored.
   Similarly, when you run the *Create Stranded Wire* command, the trunk is automatically connected to the device pins by adding vias.
4. Hold the `Ctrl+Shift` keys and move the mouse pointer to a different current position. The wire is automatically switched between the channel routing and the over-device routing.



Similarly, you can switch between channel routing and over-device routing when the *Create Stranded Wire* command is used.

Related Topics

How do I identify the net connections that either consume or generate the most current?