



Routing the Design

Product Version 23.1
September 2023

© 2024 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		12
Layout Editor: Overview of the Routing Process		12
2		15
Component Fanout		15
Related Topics		16
Controlling Fanouts		16
Defining Via Structure		18
Overriding Line Width		18
Pin - Via Space		19
Specifying Vias and Orientation		20
Creating Via Structures		27
Copying Fanouts		27
3		30
PCB Editor: Developing Interconnect Flows		30
Flow Designer		30
GRE Feasibility		30
GRE		31
Advanced Routing Functionality		31
Enabling Allegro Advanced Routing Options		32
4		34
APD: Connections		34
Prerequisites		34
Defining Connectivity Automatically		35
Defining Connectivity Manually		44
Managing Net Assignments for Multi-Die Packages		45
5		48
APD: Physical Interconnection Creation		48
Estimating Layers for Flip-Chip Designs		49
Using the Flip-Chip Die Escape Generator		51
Creating Power and Ground Plane Distribution		53
Understanding Negative Planes		53

Understanding Positive Planes	54
Prerequisites to Defining Planes	54
Creating a Negative Plane	54
Creating a Positive Plane	55
Creating Power and Ground Rings	56
Creating a Shorting Scheme	58
Prerequisites to Defining a Shorting Scheme	58
Defining a Shorting Scheme	58
Adding Wire Bonds	59
Interconnecting Wire Bonds	59
Generating Offset Vias	61
6	63
APD: Introduction to the Wire Bonding Toolset	63
How the Wire Bond Tools Work	64
Menu Available before Element Selection	64
Wire Bond Heads-Up Display	65
Using the Wire Bond – Select Command	66
Pause/Resume Commands	71
Set Default Action Command	71
Other Wire Bond Commands	72
Managing Connections for a Wire Bond Die	74
NO_WIREBOND Property	74
Setting Up a Wire Bond Design	76
Groupless Wire Bonds	78
How You Use the Groupless Wire Bonds Use Model	78
Auto Wire Bonding	79
Populating the Guide Path with Bond Fingers	79
Non-Wired Bond Fingers	81
Non-Standard Bond Wires	84
When to Use the wirebond add nonstandard Command	86
Wire Bond Tack Point	87
Routing Stubs	90
Highlighting Bond Fingers	93
Example	93
Creating and Modifying Wire Profiles	95
Who Uses the Profile Editor?	96

Accessing the Wire Profile Editor	97
How the Wire Profile Editor Works	98
Wire Profile Example	102
XML Description for Example	103
Wire Bond Via Estimation	106
How the Wire Bond Via Estimator Works	106
Running the Wire Bond Via Estimator	108
Wire Bond Die Escape Generator	110
How the Wire Bond Die Escape Generator Works	111
Advanced Wire Bonding	111
Wire Bonding Multiple Stacked Dies	112
Pushing and Shoving Same Net Bond Fingers	121
Correctly Staggering Die Pins to Avoid False DRC Violations	132
Wires Bond from Same Location Die Pins out to Same or Different Bond Fingers	137
Wire Bonding: Using the dxf in and dxf out Commands	143
Wire Bond Use Models	145
Single-Chip Wire Bond Use Model	145
Die-to-Die Wire Bond Use Model	148
Wire Bond Stacked-Die Use Model	149
7	152
Routing	152
Prerequisites to Routing	152
Generating Radial Routes	153
Using Custom Smoothing	154
Routing Automatically with the Allegro PCB Router	155
Routing Differential Pairs and Buses	156
8	159
Interactive Routing	159
General Routing Prerequisites	160
Dynamic Etch Editing	162
Setting Visibility during Interactive Routing	163
About Bubble Mode	166
Hug and Shove-Preferred Modes When Adding Clines	166
Handling DRC Errors	167
Obeying Line Angle Controls	167
Odd Angle Lines	167

45-Degree Lines	168
Pads	168
Arcs	169
Constraint Areas	169
Via Shoving	169
Using the slide Command in Bubble Mode	173
Using the add_connect Command in Bubble Mode	173
About Scribble Mode	175
About Snake Mode	179
Adding Connections to a Design Interactively	181
Routing with Enhanced Pad Entry	183
Optimizing Routes in Channels	187
Viewing Clearances in Channels	190
APD: Routing or Sliding in Super Smooth Mode	193
How to Access the Super Smooth Option	193
Examples of Super Smooth Mode	194
Interactive Routing with Layer Set Constraints	197
Defining Line Width	199
Cornering	201
Adding Vias	201
Adding Via Structures	203
Use Models for Adding Vias and Via Structures	205
Adding Vias Using the Working Layers Mode	206
Adding Jumpers	212
Editing Connections and Vias	216
Routing High-Speed Circuits	219
Routing Rat Ts	219
Displaying Timing Feedback	219
Displaying Timing Feedback With Constraint Manager	228
Displaying Etch Length	229
Delay Tuning	232
Elongation Styles	232
Differential Pairs	236
Phase Tuning	237
Editing or Deleting a Vertex	237
Deleting a Vertex	237
Changing the Width of Clines	238

How the Cline Change Width Command Works	238
Adding Teardrops Interactively	240
Adding Teardrops Automatically	240
Tapering Traces	241
Glossing a Design	244
Interactive Routing for Differential Pairs	245
Single Trace Mode	245
Diffpair Driver-Receiver Model Translation	248
Setting Up and Editing Differential Pairs Using the Etch Edit Tools	249
Line Spacing	249
Cornering	250
Grid Snapping	252
Route Necking	252
Single Trace Mode	252
Gathering and Splitting	253
Slide	256
Interactive Group Routing	258
Routing Spacing	259
Control Trace	260
Cornering	260
Snapping and Hugging	260
Routing in Single Trace Mode	261
Via Patterns during Group Routing	261
Types of Via Patterns	262
Interactive Freestyle Multi-Line Routing	267
Setting Multi-Line Route Parameters	268
Multi-Line Routing and Graphic Feedback	270
Leveraging Design Intent to Route a Bus	271
Using Contour to Route Rigid-Flex Designs	272
Setting Contour Route Parameters	273
Generating Reports on Interactive Routing	276
Setting Ratsnest Schedule for a Net	276
9	282
Prerequisites for Allegro PCB Router Automatic Routing	282
General Routing Prerequisites	282
Grids and Automatic Routing	283

Defining Routing Grids	284
Via Grids	284
Related Topics	284
Controlling How Vias Are Used during Routing	284
Defining Vias for Use during Routing	285
Controlling Via Staggering	287
Allowing Via Placement on Pads	288
Controlling Via Stackups	289
Controlling the Distance between Buried Vias	290
Controlling the Number of Vias on a Net	291
Scheduling Nets Interactively	292
Constraints That Affect Automatic Routing	293
Net Properties	293
Component Properties	296
Optimizing Tpoint Location	297
10	300
Post-Route Clean Up	300
Before Glossing	302
Excluding Nets from Glossing	302
Excluding Areas	302
Defining Areas to be Glossed	303
Glossing Applications	304
Line and Via Cleanup	304
Via Eliminate	304
Line Smoothing	305
Center Lines between Pads	305
Improving Line Entry into Pads	305
Line Fattening	305
Converting Corner to Arc	307
Pad and T Connection Fillet	308
Dielectric Generation	315
Glossing Error Reporting	316
Warning Messages for Pad and T Connection Filleting	318
Error Reporting for Dielectric Generation	318
11	320
Using the Allegro PCB Router Translator	320

Prerequisites to Running the Translator	321
Restrictions and Considerations	322
Analysis Mode Settings in Constraint Manager	323
Constraints	323
Allegro / Constraint Manager	326
Allegro PCB Router	327
Running the Pre-Route Checker	327
Listing of Pre-Route Design Checks	328
Running the Translator	329
Translation Procedures	330
Layout Editor to Allegro PCB Router	330
Allegro PCB Router to the Layout Editor	330
Related Topics	331
Mapping of Properties, Assignment Tables, Rule Sets, and Constraints	331
Layout Editor Properties to Allegro PCB Router	331
Layout Editor Rule Sets to Allegro PCB Router	345
HDI Objects Support	350
Layout Editor Constraints to Allegro PCB Router Rules	352
Troubleshooting Translation Problems	359
12	361
Automatic Routing with Allegro PCB Router	361
Modes of Operation	361
Related Topics	362
Autorouting File Generation	362
Autorouting Task Flows	363
Mainstream Flow	363
High-Speed Flow	364
High-Speed Power User Flow	365
Autorouting Parameters	367
Setting Parameters in the Mainstream Flow	367
Setting Parameters in the High-Speed or High-Speed Power User Flows	370
Autorouting File Examples	370
Sample Rules File	370
Sample Forget File	373
13	375
Allegro Integrated Analysis and Checking	375

Performing Virtual Prototyping	377
Using the Design Set Up Workflow	381
Performing Impedance Analysis	382
Impedance Vision	384
Impedance Table	385
Performing Coupling Analysis	386
Coupling Vision	388
Coupling Table	389
Impedance and Coupling Plots	391
Performing Crosstalk Analysis	392
Crosstalk Vision	394
Crosstalk Table	395
Performing Return Path Analysis	398
Return Path Vision	399
Return Path Table	400
	400
Return Path Vision	400
Performing Reflection Analysis	401
Reflection Vision	403
Reflection Table	404
Reflection Vision	405
Performing IR Drop Analysis	406
IR Drop Vision	408
IR Drop Table	409
IR Drop Analysis Parameter Setup Window	413
IR Drop Table	416
IR Drop Vision	419
Performing Power Inductance Analysis	421
Performing Topology Extraction	423
Performing Via Modeling using a Wizard	424
Performing Interconnect Model Extraction	425
Selecting Nets	425
Specifying Directed Groups	427
Setting Up Component Models	430
14	434
IDA Via Wizard Workflow	434

15		439
Using Vision Manager: Visual Results for Analysis and DRCs		439
Route Vision		441
Placement Vision		443
16		444
APD: Working with a Plating Bar		444
Prerequisites		444
Creating a Plating Bar Symbol		444
Alternative Plating Bar Creation Techniques		447
Related Topics		448
Verifying Plating Bar Errors		448
Deleting a Plating Bar		449
Generating a Plating Bar Report		450

Layout Editor: Overview of the Routing Process

The layout editor provides tools that help you perform the following when routing physical designs:

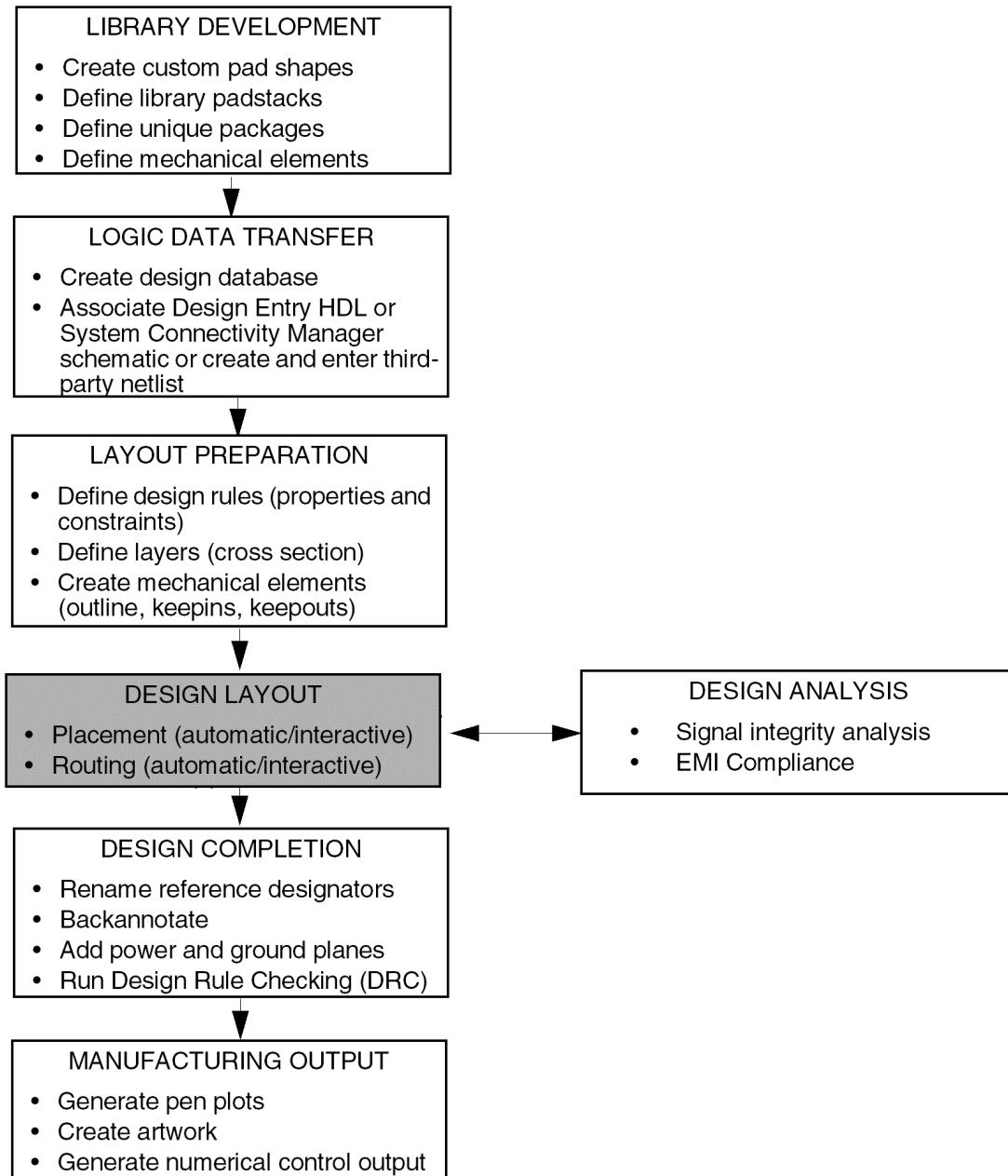
- Interactive routing
- Automatic routing with Allegro PCB Router
- Glossing to improve the appearance and manufacturability of a physical design

The following list describes a flowchart of the basic routing process. The basic routing flow—assuming automatic routing—is:

1. Prepare for routing:
 - Check layers to see that layer types and photo film types are correct.
 - Create internal shapes on planes.
 - Create blind and buried vias.
 - Set routing controls (routing areas, constraints and properties, grids, nets, and so on).
2. Manually route critical nets.
3. Define routing parameters in Allegro PCB Router to control how automatic routing functions.
4. Run Allegro PCB Router.
5. Review routing results.
6. Interactively finish or correct etch.
7. Gloss the design.
8. Optionally analyze the design for signal integrity or EMI.

Perform routing anytime after placement in a design flow as follows.

PCB Editor: Interactive Routing in a Physical Design Flow



Component Fanout

There are several interactive and automatic controls for component fanout, a process sometimes referred to as pin escaping. You can use the `fanout_by_pick` command to invoke the Allegro PCB Router and work with `.do` files, route interactively within the layout editor, build fanouts into your library symbols, or use the suite of interactive fanout commands located in the *Route* menu. The interactive suite includes four commands to create, copy, define via structures, and convert cline/via extensions to fanouts.

Fanouts created with the interactive suite are automatically associated to the symbol instance. This is beneficial when moving a component; fanouts not associated remain in place when a component is moved. The `mark fanout` command can be used to associate fanouts that may have originated from the Allegro PCB Router or ones routed manually. The `mark fanout` command can be used to convert escapes generated using various APD commands, such as the Flip-Chip Die Escape Generator or the Wirebond Die Escape Generator.

The `create_fanout` command is applied to a single package symbol. There are several control options available to customize the style and physical characteristics of the resulting fanout pattern. Unlike the `fanout_by_pick` command, the `create fanout` command is not DRC aware and may result in via to element conflicts. Running additional passes with parameter adjustments may be required to reach final intent.

Complex via structures are typically used on physical designs utilizing High Density Interconnect (HDI) rules. If required for fanout, they can be defined using the `redefine via structure` command and then applied to a symbol with the `create fanout` command. Via structures are uniquely named and are stored in the database.

The `copy fanout` command is used to replicate an existing fanout pattern to all other packages or devices of the same name on the same layer. There are no restrictions on the origin fanout; it can be auto generated or user defined.

Related Topics

- [fanout_by_pick](#)
- [mark fanout](#)
- [create_fanout](#)
- [fanout_by_pick](#)
- [create fanout](#)
- [redefine via structure](#)
- [copy fanout](#)

Controlling Fanouts

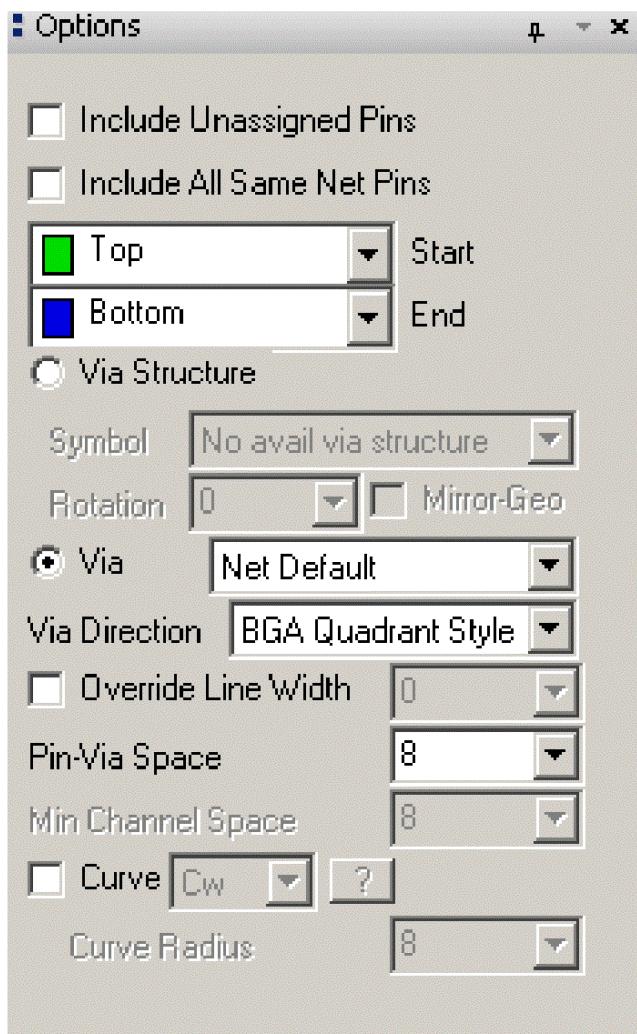
The [create_fanout](#) works with the *Options* tab of the Control Panel where several parameters exist to control the physical spacing and style of the fanout pattern. Available elements for selection are symbols and pins. Generating a fanout automatically replaces any existing fanout on the chosen elements. If pins connect to a different component, existing fanouts are preserved. The command does not create:

- Shared vias
- Multiple vias for voltage pins
- Fanouts for thru-hole pins
- Fanouts for pins whose padstack name contains FID, assumed to be fiducials

If multiple via padstacks are associated with a net, then fanout is created using the first via available in the list.

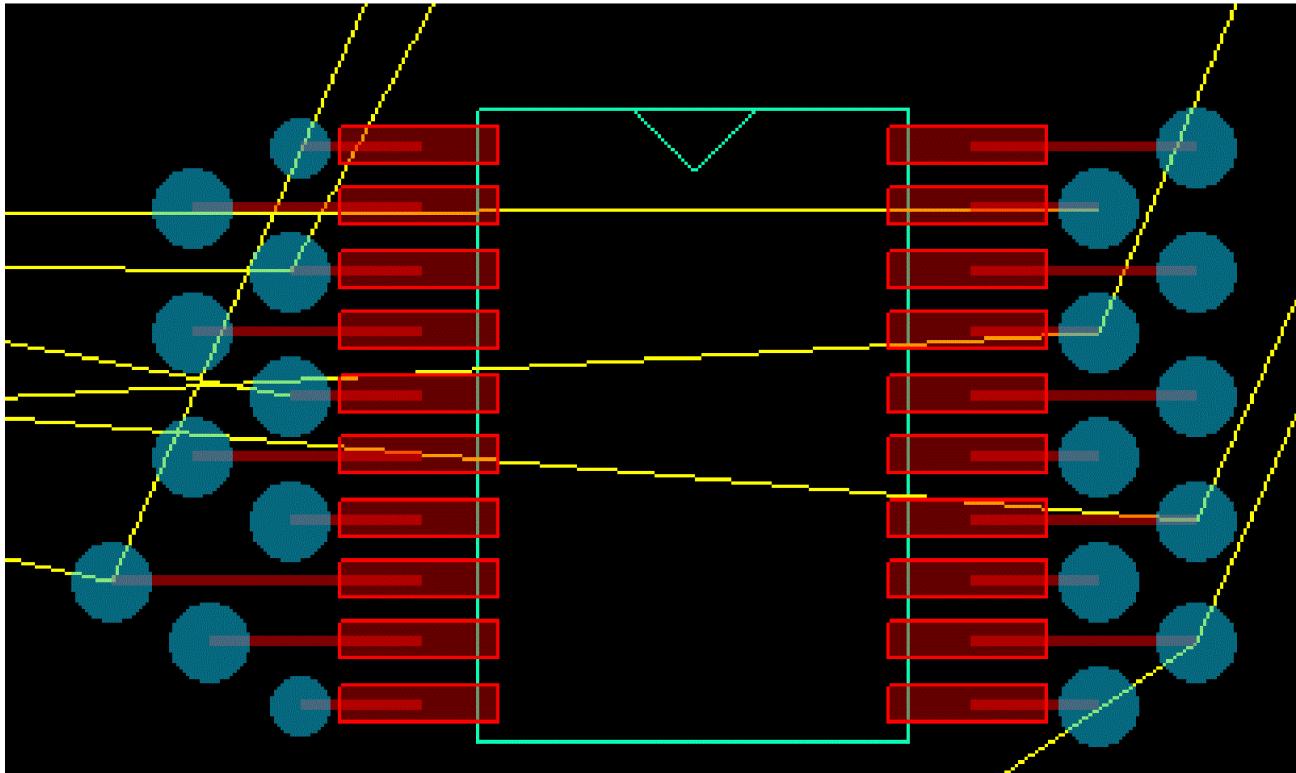
The [create_fanout](#) and the [copy_fanout](#) commands are not DRC aware. DRC errors may result after each operation. Parameter adjustments or interactive editing may be required to comply with DRC rules.

Options tab for `create_fanout`



The use of the *Unassigned Pins* option results in the fanout of all pins on a component. Pins not assigned a logical connection are considered unassigned, as shown in the following figure.

Fanout of Unassigned Pins



Defining Via Structure

You can add a via structure, which lets you capture complex fanout patterns that may include multiple vias or clines, to symbol or pins you choose.

Overriding Line Width

By default, the `create_fanout` command uses the line width specified in the respective net's cset. Use of the *Override Line Width* parameter overrides the inherited value and applies to all pins.

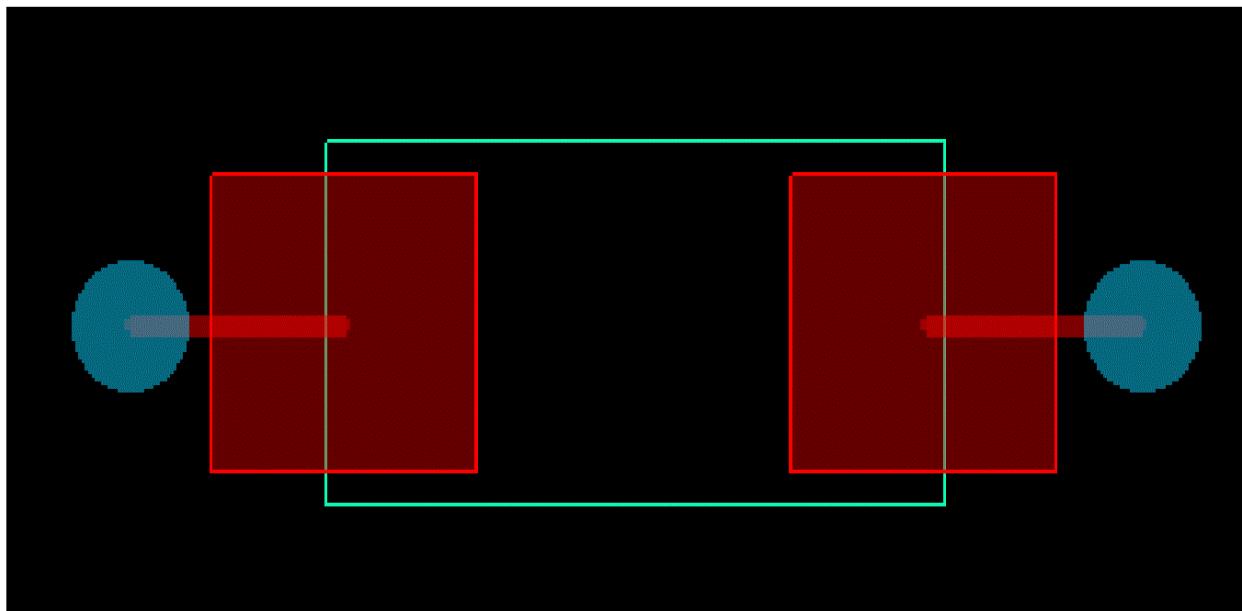
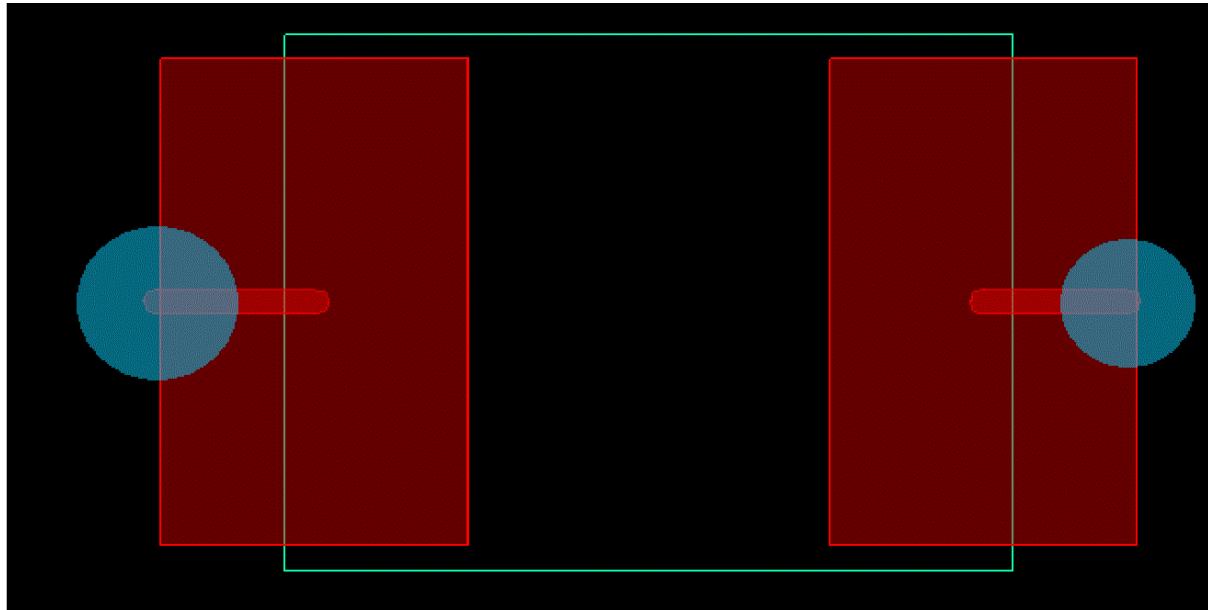
Selective override at the net level can be accomplished by enabling the *Include all Same Net Pins* option. For example, you may want all logical nets to be 6 mils, but DC nets to be 10 mils. After using the `create_fanout` command with override line width set to 6 mils, change the override line width to 10 mils, enable the same pin option, set the Find Filter to pins, then select a DC pin to fanout all pins on that net.

Pin - Via Space

Distance between the edge of the pin-pad and the edge of the via-pad may be regulated with the *Pin to Via Space* parameter. Zero or negative numbers are valid entries.

 The *Pin-Via Space* and *Min Channel Space* fields are copied from the constraints when the create fanout command is run for the first time. The Pin-Via Space and Min Channel Space fields are not updated if changes are made in Constraint Manager after the first run.

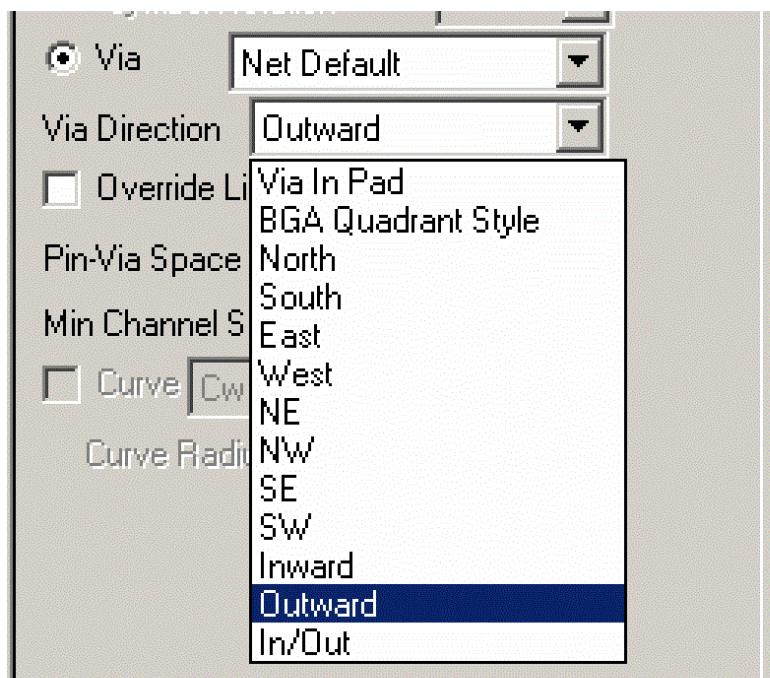
Pin to Via Spacing (top: negative clearance value; bottom: positive value)



Specifying Vias and Orientation

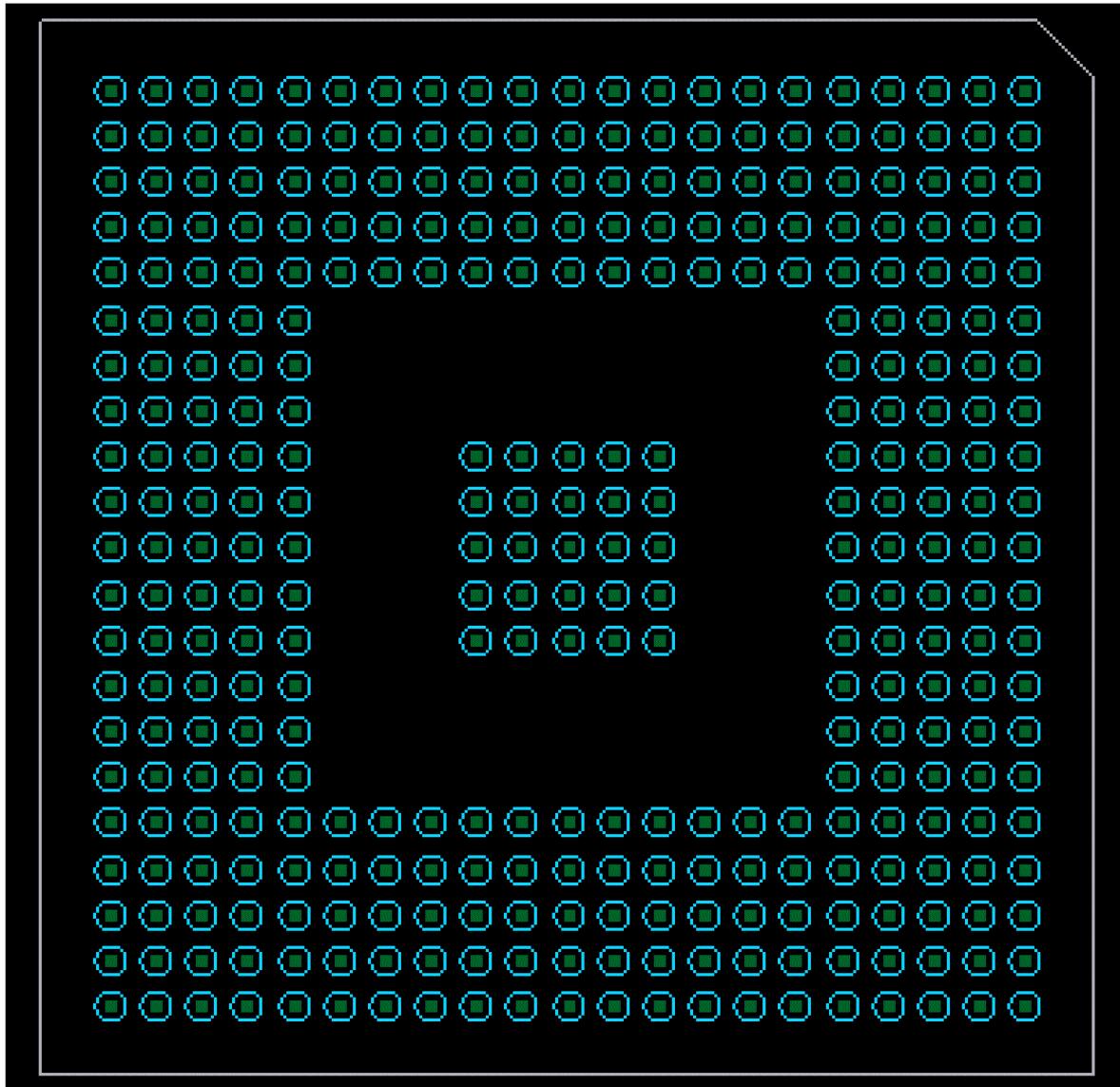
You can choose the type of via from a list of those stored in the database that span specified subclasses. The combination of the start-end layers produces the list of applicable via types for that range. There are 13 direction types to consider pending the type of package symbol with which you are working. The direction of the fanout via is relative to the pin location. Distance between the edge of the pin-pad and the edge of the via-pad may be regulated with the *Pin to Via Space* parameter. Zero or negative numbers are valid entries.

Via Direction Parameters



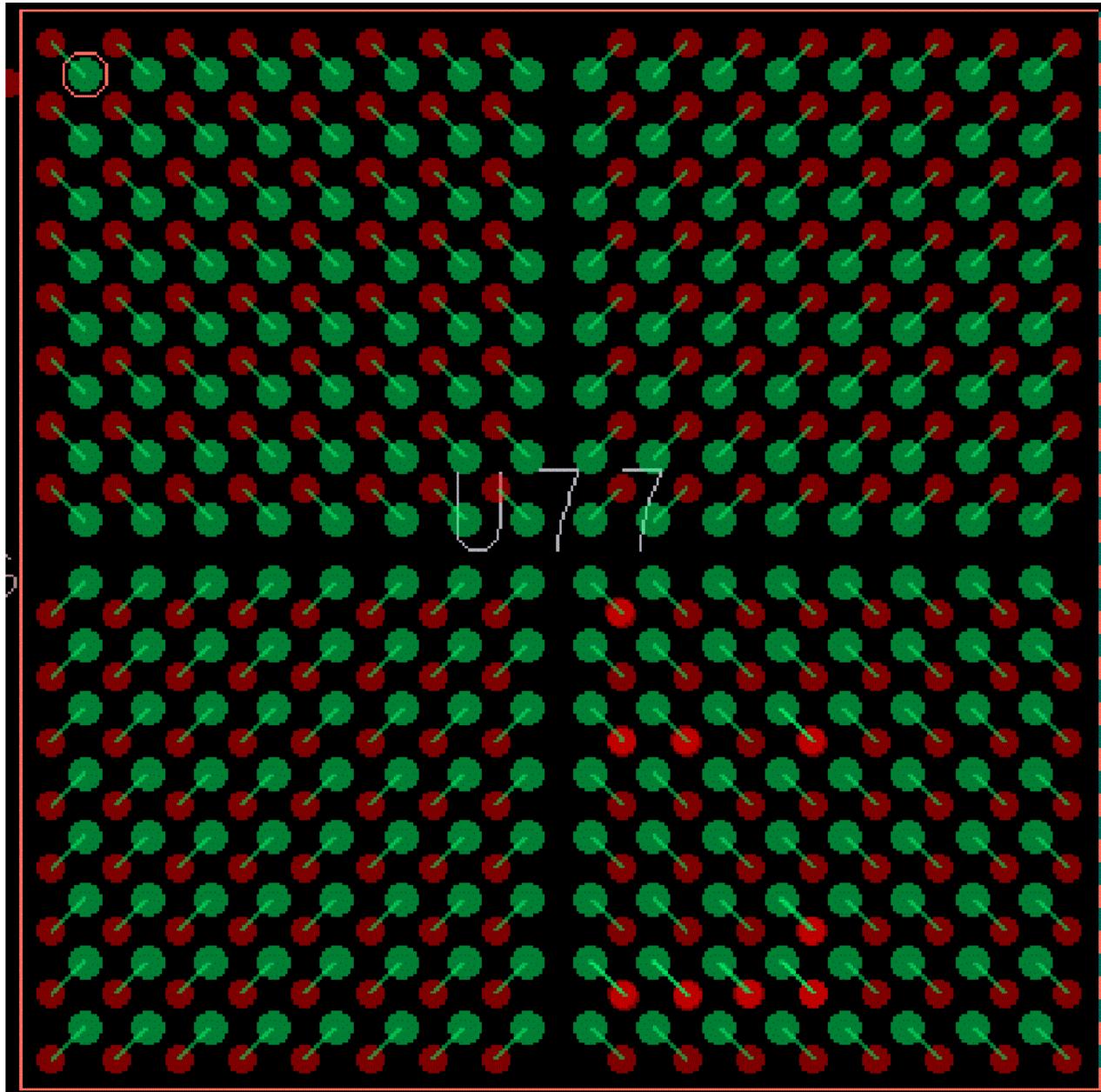
Via in Pad centers a via at the pin origin with no cline between the via and the pad.

Via in Pad



BGA Quadrant is the default style in which vias are created for each pin in the direction away from the symbol center. Two unused channels remain (one vertical, one horizontal) that pass through the symbol center. BGAs by default fan out using 45 degree angles.

BGA Quadrant Style



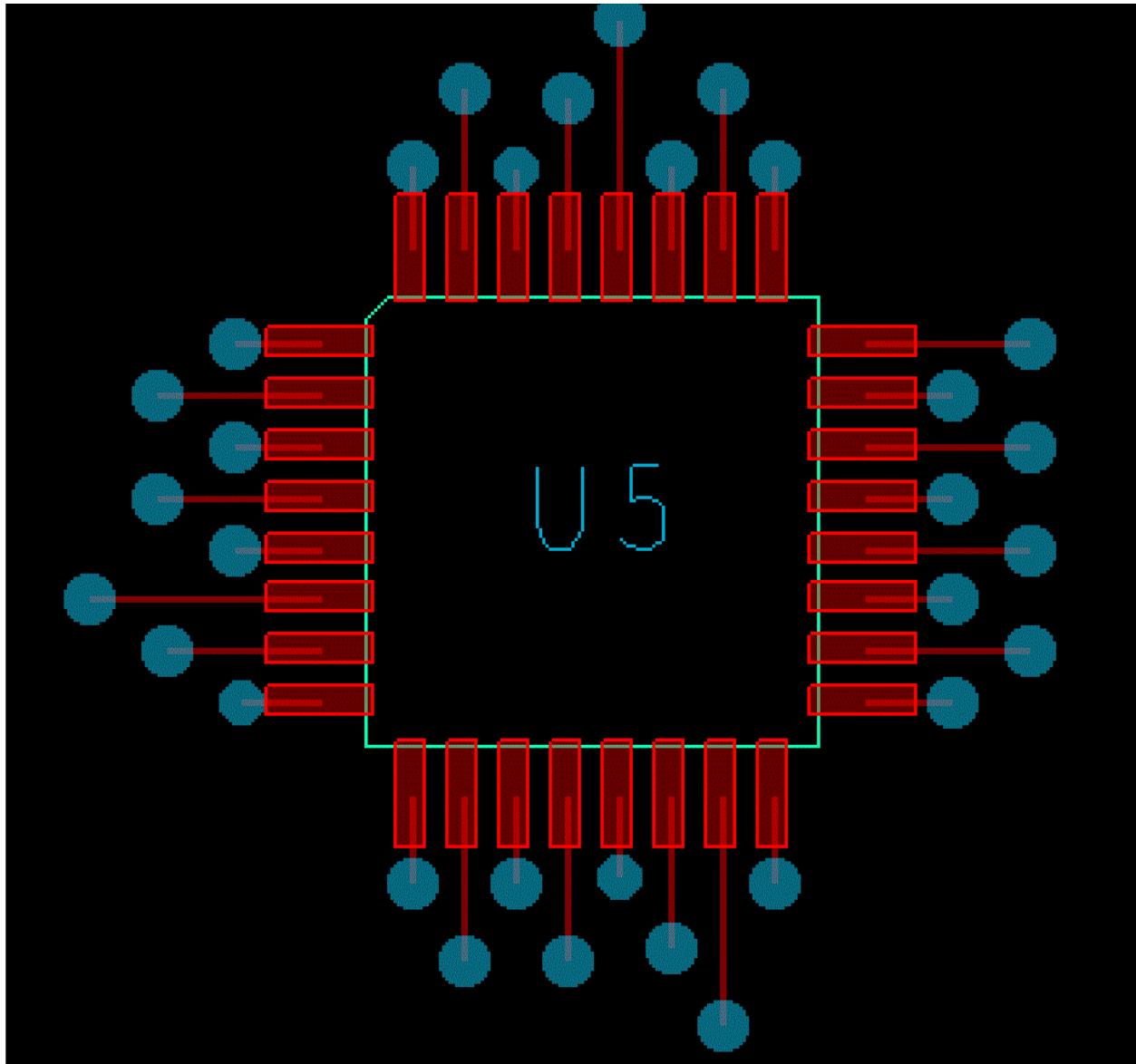
North, South, East, West specify compass-point directions.

- *NE* (northeast), *NW* (northwest), *SE*, (southeast), and *SW* (southwest) control 45 degree angles.
- *Inward*, *Outward*, and *In/Out* are useful for SOICs and other non-BGA components.

- *Inward* allows vias underneath the component.
- *Outward* allows vias outside the component.
- *In/Out* allows fanouts to alternate between *Inward* and *Outward* orientation.

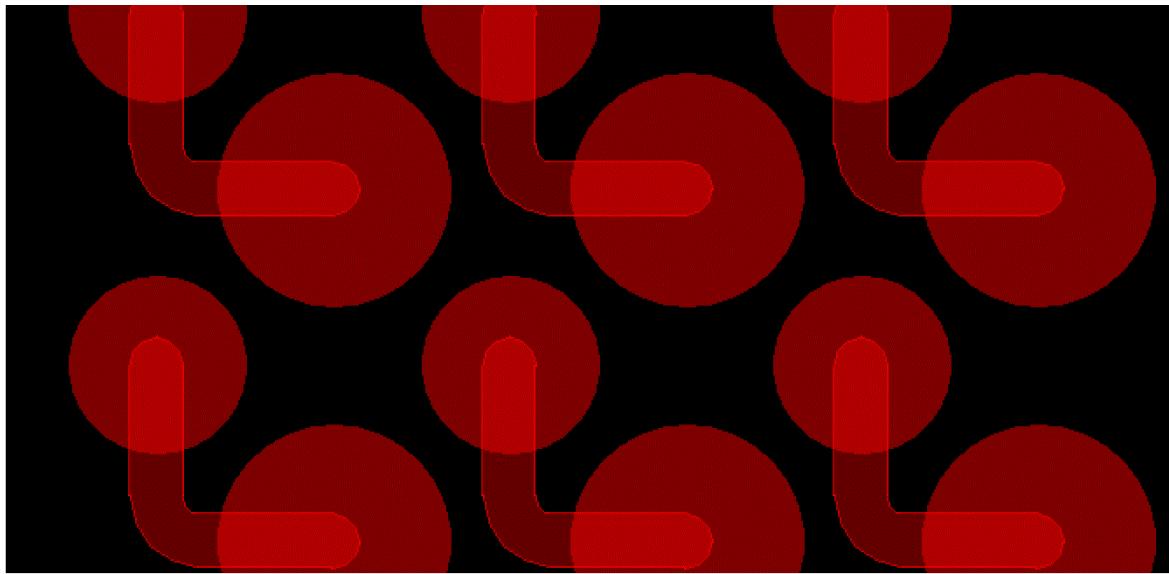
The *Min Channel Space* parameter is used to maintain a minimum space between adjacent fanout vias. The distance spans the edge of a via-pad to the edge of a via-pad on the diagonal, and as the distance increases, so does the stagger effect, as follows. The value defaults from the via-to-via space in the default constraint set and is available when you set *Via Direction* to *Inward*, *Outward*, or *In/Out*.

Minimum Channel Space between Vias



To create fanout clines with two segments and an arc either clockwise (cw) or counter-clockwise (ccw), a *Curve* option is available when used in conjunction with a *Via Direction* of *BGA Quadrant Style*, *NE*, *NW*, *SE*, or *SW*. A graphic depiction of the *Curve* option's effects is available by pressing the "?" on the *Options* tab.

Curved Cline Segments



Creating Via Structures

A via structure is a series of vias and clines used primarily on HDI designs to transition a signal from the surface into the HDI or core layers. The `create via structure` command can be used to define a single via structure that comprises a single via and connect line, or a multiple combination of these elements spanning many layers. Chosen vias and clines must all be connected to each other and belong to the same net. They may be connected to only one pin, whose location becomes the symbol origin. Duplicate symbol names are not created.

When more than one type of via structure is required for a component fanout, such as for power or ground fanouts that terminate on different end layers, you can define additional via structures, then use the `create fanout` command in combination with the *Include all Same Net Pins* option to disperse the new via structures across all same net pins across the symbol.

Related Topics

- [create via structure](#)
- [create fanout](#)

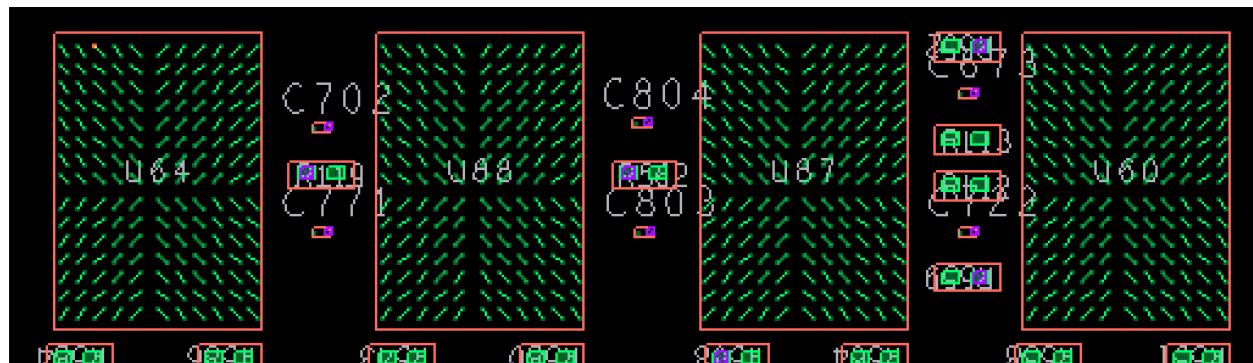
Copying Fanouts

The `copy_fanout` command replicates instantiated fanouts across all common package names or device types. Replication is limited to the same side of the board where the origin component is placed. For symbols on the opposite side, you can create a fanout for one instance using the `create fanout` command, and then copy it to the remaining symbols on that subclass.

Copying a fanout automatically replaces any existing fanout on the chosen component unless the **FIXED** property has been assigned, or the fanout is routed to a different component. A copy occurs even if you have modified a symbol pin's padstack on one instance, as long as the pin location remains unchanged.

Copied fanouts replicate the origin symbol attributes such as line width, via type, direction or via structure. DRC errors may occur after the command is completed; for example, the copied fanout via may conflict with an adjacent pad or may not meet minimum line width requirements.

Copied Fanouts



PCB Editor: Developing Interconnect Flows

Allegro provides advanced routing options that let you graphically develop, analyze, and even generate complete routing solutions for interconnect flows in your design. The following are licensed advanced routing options available with the Allegro PCB Design XL and Allegro PCB SI products.

- Flow Designer
- GRE Feasibility
- GRE (Global Route Environment)

Flow Designer

Flow Designer lets you work at an abstract level to develop interconnect bundles and flow paths in your design. It allows you to visually direct the flow of busses and critical nets without being concerned with traditional routing details such as shoving or moving adjacent trace.

This high-level input allows you to graphically communicate complex interconnect design requirements (design intent) that otherwise could only be expressed verbally or with crude sketches. In other words, your design intent is implied within the interconnect flows that you create and is captured as part of the design database. This enables other members of your design team to review your design intent and use it to drive manual routing of the design.

GRE Feasibility

GRE Feasibility is a superset of Flow Designer. In addition to developing interconnect flows, you can use features available with GRE Feasibility to instruct the GRE route engine to generate route plan lines in your design that show the spatial feasibility of your interconnect flows. This gives you the opportunity to perform route feasibility studies and make adjustments in the design to refine the flows that are used to guide manual routing.

GRE

Global Route Environment is a superset of all Allegro advanced routing options. It has two major components; the Interconnect Flow Planner and the GRE Route Engine. GRE not only lets you develop interconnect flows, but allows full access to its route engine that can interpret your design intent and automatically generate interconnect solutions for complex designs.

Advanced Routing Functionality

The following table lists differences in the feature sets between Allegro's advanced routing options.

Feature Differences between Advanced Routing Options

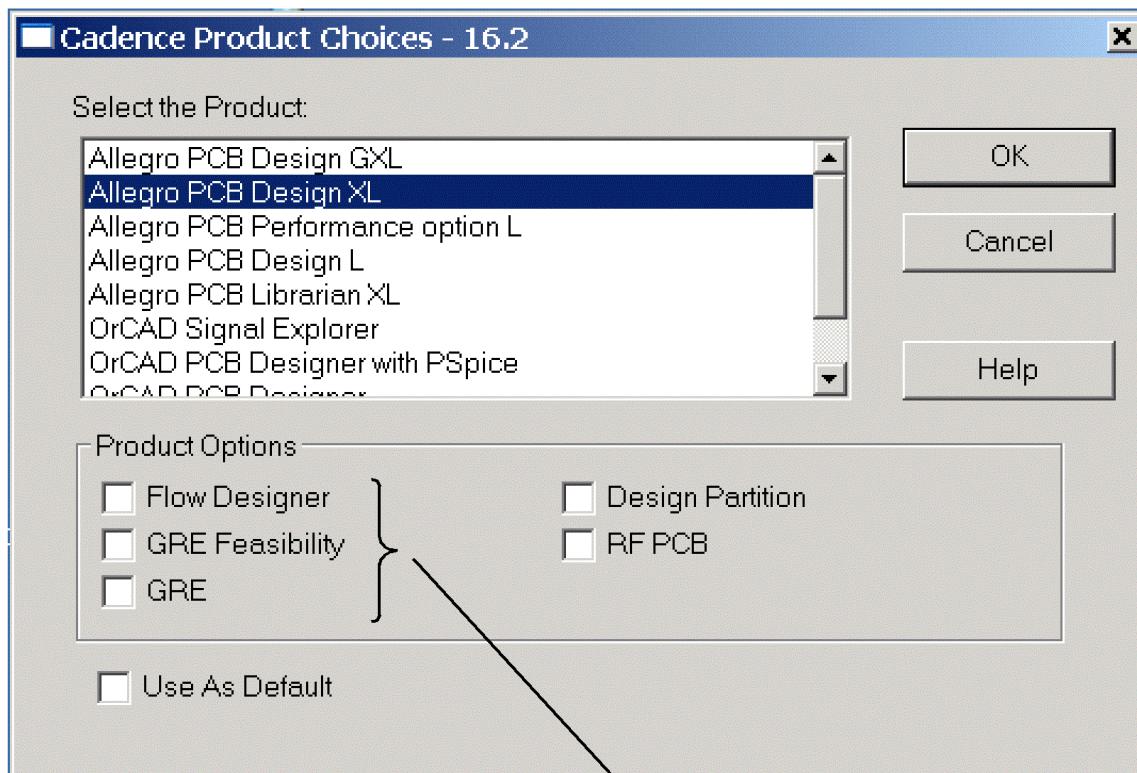
Feature	Flow Designer	GRE Feasibility	GRE
Design Parameters - General and Default - Auto Bundle - Layer, Plan, Route	x 1 x	x x x	x x x
Bundle Creation and Editing - Manual and Automatic - Edit, Split, Delete, etc. - Constraint Manager Bundling - Import Bundle	x x x 1 x	x x x x	x x x x
Flow Creation - Edit, Move, Slide, etc. - X/Y Guidance Control - Flow Vias	x x x	x x x	x x x
Bundle and Flow Properties - General - Bundle and Flow Layering - Routing Controls	x 1 x 1	x x x	x x x
Solution Planning - Spatial Feasibility - Topological - Accurate		x x	x x x
Etch Commands - Commit plan to etch - Convert etch to plan - Optimize		x 2	x x x
Graphic Display - Bundles and Flows - Plan data - Plan errors - Plan status	x x	x x x x	x x x x

1. Routing-related controls unavailable.
2. Conversion to Spatial only.

Enabling Allegro Advanced Routing Options

You enable the advanced routing options from the Cadence Product Choices dialog box as follows. This dialog box is displayed when you invoke Allegro or when you choose *File – Change Editor* from the Allegro menu bar.

Product Choices Dialog Box



Routing the Design

PCB Editor: Developing Interconnect Flows--Enabling Allegro Advanced Routing Options

APD: Connections

You logically connect pins in the physical design to create nets after you set the design parameters and generate the BGA and die symbols. Your layout editor provides both automatic and interactive commands to let you add, edit, and delete nets.

With the editor, you interactively assign and deassign I/O component and plating bar pins to nets in the design. You can also create new nets. Interactive net (and pin) assignment let you graphically assign component pins to nets for optimal routability.

You can perform pin assignment on individual pins by choosing an item on the net for assignment and then choosing the pin. You can do this simultaneously on all, or a subset of I/O or plating bar pins.

After you assign I/O and plating bar pins to nets, you can also attach various properties (attributes), such as delay, matched delay, and differential pair to critical nets. You can import properties through a technology file or you can attach properties interactively.

Prerequisites

Do the following before you assign pins:

1. Ensure that die pins have associated net names.
2. Place the die.
3. Ensure that I/O or plating bar pins, or both, exist in the design.
4. Assign the proper CLASS attribute to design elements:
 - die = IC
 - pkg = IO
 - platebar = PLATING_BAR

Defining Connectivity Automatically

To create and assign many nets for components, choose the *File – Import – Netlist-In Wizard* ([net list in](#)) command from the menu bar after you create a BGA component and die. If the netlist information is in a spreadsheet, you may organize it (netname; refdes; pin; refdes; pin...) to output a text file that the Netlist-In Wizard then uses. Otherwise, you must use a text editor to create the netlist.

The Netlist-In Wizard lets you:

- Generate pin assignments and connectivity by importing an ASCII spreadsheet of net information.
- Manipulate the spreadsheet information in the Netlist-In Wizard to modify individual net values.
- Place columns of data in a standard format.

The Netlist-In Wizard presents a series of dialog boxes to guide you through the process of importing net data.

This section describes:

- [Creating Nets](#)
- [Assigning Routing Layers](#)
- [Automatically Assigning Nets](#)
- [Displaying Rats by Layer](#)
- [Automatically Assigning Pin Use Codes](#)

Creating Nets

Creating an initial net assignment for a single component establishes a base from which you can automatically assign nets to other components in your design. You can create one or more net assignments by:

- Reading in a netlist file: *Generate – Netlist-In Wizard* ([net list in](#) command).
- Creating a net automatically: *Logic – Auto Create Net* ([auto create net](#) command).
- Creating a net manually: *Logic – Create Net* ([create net](#) command).

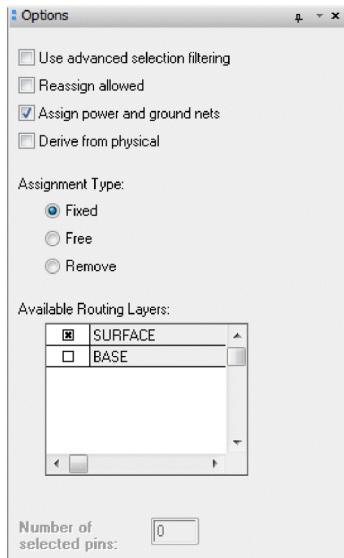
Assigning Routing Layers

Assigning routing layers for specific pins or nets or both ensures that the automatic assignment function adheres to your routing strategy. The importance of this process is proportionate to the complexity of your design so it is recommended for all but the simplest cases.

Assigning a routing layer to a pin or a net attaches the `ASSIGN_ROUTE_LAYER` property to it. The value of the property is the assignment type and layer you specify in the *Options* window pane of the Control Panel.

Based on the assignment type that you set in the *Options* window pane when the command is active, you can fix pins and nets on specific routing layers or free them to be routed on the layer most likely to ensure a successful connection when you route the design.

Options Window Pane for Assign Route Layer

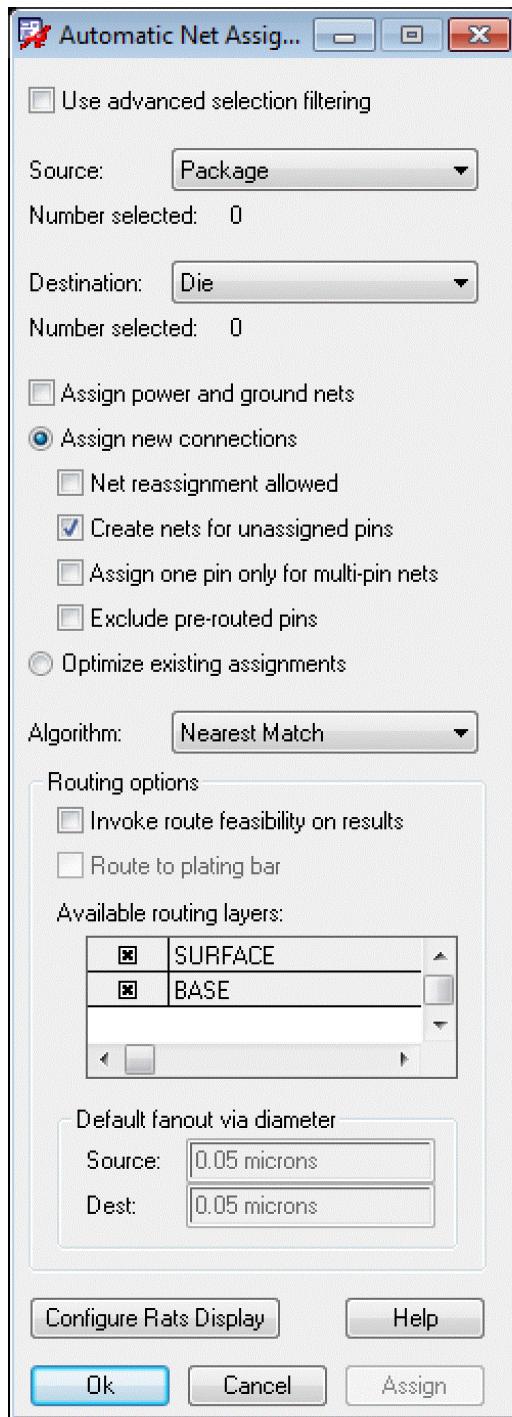


You can assign routing layers per pin or per net. Assigning by pin gives you greater latitude in routing your design since you can assign different pins in a single net to different routing layers. Assigning by net results in all the pins in a net routing to a single layer; however, you can override this for individual pins in a net.

Automatically Assigning Nets

To facilitate routing, you can create and assign routing conditions among your die, component, or plating bar. If the component I/O pins are not preassigned, then choose *Logic – Auto Assign Net* (*auto assign net* command) to use the net names from the die pins and assign them to the closest BGA pin. If some pins are already assigned, ensure that the *Net Reassignment Allowed* box in the Automatic Net Assignment dialog box is not checked so that the existing net assignment remains while unassigned pins receive net names. Prior to automatically assigning nets, you may want to connect wire bonds from the die pin to the bond finger. You can make these connection interactively or automatically.

Automatic Net Assignment Dialog Box



Automatic net assignment uses your design constraints, component layout design, and routing layer assignments to determine routing solutions among pins, nets, and components.

ⓘ If your design has a wire bond die, you must wire bond it before running the `auto assign net` command; otherwise the design is treated as a flip-chip die.

Optimizing Pin Assignments in a Co-Design Flow

When a source pin is assigned to a target pin, unless you use the existing *Optimize existing assignments* option, the `auto assign net` command sets the net on the target pin to the net on the source pin. When auto assigning source pins to the pins of a co-design die, the result is that the new net assignment on the target pin does not match the logical connectivity of the pin, which is set through the `VERILOG_PORT_NAME` property. The regular mode of `auto assign net` does not modify the `VERILOG_PORT_NAME` properties on the pins, even when net reassignment is allowed.

For example, the pins of a co-design die are randomly assigned to nets that are derived from the logic connections from the BGA in System Connectivity Manager (SCM). Suppose pin A1 is assigned to NET_A and the `VERILOG_PORT_NAME` property is `net_a`, while pin A2 is assigned to NET_B and the `VERILOG_PORT_NAME` property is `net_b`. If you run the `auto assign net` command to optimize the pin assignments, pin A1 is assigned to net NET_B, yet the `VERILOG_PORT_NAME` property remains `net_a`. The logical and physical net connections are not synchronized.

As part of the `auto assign net` command, however, you can optimize the logical-to-physical pin assignments to a component that has pin assignment freedom, such as a co-design object. The reassignment does not create a logical connectivity change, but rather reassigns the logic to different physical pin locations using swapping. Swapping keeps the physical net assignment synchronized with the logical port on the pin. Using this feature with the example above, if pin A1 is assigned to NET_B, the `VERILOG_PORT_NAME` property on the pin is also changed to `net_b`.

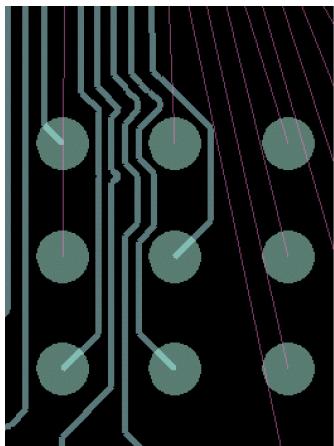
To optimize the pin assignments using the *Optimize existing assignments* option, you select pins belonging to one or multiple components for the source set, but you can only choose pins from one component for the destination set.

 Using Optimize existing assignments option preserves existing rat lines and rat bundles. Bundles are preserved only for nearest match.

The following figure shows an example before optimization.

 Components that are not co-design dies do not have the Verilog port name property.

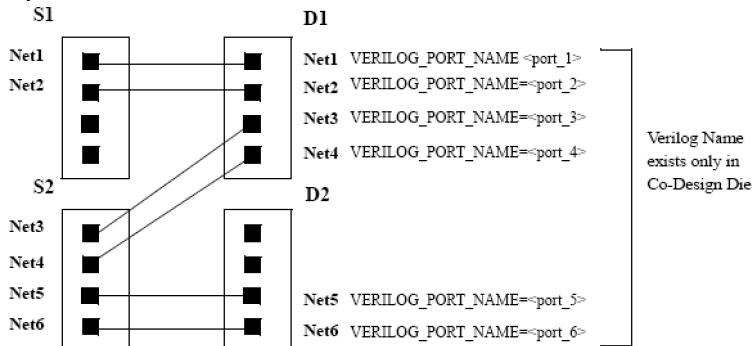
Before Optimization



In the following figure, the *Optimize existing assignments* option allows the selection of all the pins of source components S1 and S2 but allows only the selection of pins belonging to either components D1 or D2 for swapping in the destination set.

 Components that are not co-design dies do not have the Verilog port name property.

After Optimization



If you select pins belonging to both components D1 and D2 in the destination set, this error appears:

Optimize existing assignments can only be done when the destination pins belong to one component.

⚠️ If you use this feature on a fixed component, the tool does not move the logical pins to new pin locations; as a result, a logical connectivity change occurs.

Using Various Modes to Run the Command

You can choose various modes in which to run the `auto assign net` command. The mode available to you depends on the source-to-destination elements selected for assignment:

- **Nearest Match**

The *Nearest Match* mode (available for all element selections) is the default method for determining how net assignments are performed (except for Die-to-BGA and BGA-to-Die, which default to Router Based). *Nearest Match* creates connections between pairs of the nearest available pins without regard for constraints, layer cross-section, or other design considerations. After running the functionality in this mode, APD displays a list of all pins that cannot be assigned.

Nearest Match mode is unlikely to result in a solution for a design that can be routed, but may be useful as a starting point for routing analysis. If you cannot apply the router-based method to your design, the functionality reverts to nearest match.

- **Router Based**

The *Router-Based* mode is a self-contained algorithm that uses the parameters you previously established (constraint definitions, layer assignments, pre-routes, and so on) to calculate the best solution for routing your design. Note that actual routing does not take place; rather, the functionality determines the sequence and layers on which routing needs to occur to effect a successful result. In this mode, the `auto assign net` command:

- Creates a virtual fanout for your die.
- Adjusts pin ordering to compensate for any required layer swapping.
- Places pre-existing assignments and routes.
- Completes assignment of remaining free balls.

In cases where conflicts remain unresolved, you may need to reassign or add routing layers to effect a satisfactory conclusion.

- **Constraint Driven**

The *Constraint Driven* mode optimizes connectivity between die objects, discrete components, and the component substrate. Connectivity optimization provides for a more routable substrate using fewer layers. It also provides greater certainty that interconnect constraints (electrical and physical) will be achieved.

The goal is to ensure that any connectivity mapping or re-mapping meets impedance, timing, and net scheduling rules or constraints that are placed on particular nets. The *constraint driven* mode automates this process at the substrate floorplanning stage once the die and any other components are placed. (If any components remain unplaced, then connectivity to those components is temporarily ignored.) The *constraint driven* mode operates on single or multi-die packages and system-in-package designs.

For dense flip-chips, the fan-out pattern may control the connectivity assignment as much as crossovers do. The fan-out pattern can also control how many routing layers are needed in the component. This also applies to generating the bond finger pattern for a wire bond die before assignment.

The pins can have certain constraints, or swap codes, such as:

- Pins are fixed on either the pin or net level.

- Pins are part of a group that must move as a fixed group (hard macro, differential pair, and so on).
- Pins are part of a group that can be moved within the group (memory block, and so on).
- Pins at the net level can have an "unwillingness to move" factor: the higher the value, the less likely they will be moved. The ROUTE_PRIORITY property determines which pins need to be assigned first and should have less "willingness to move." Other pins with a lower value swap first to try to improve the connectivity pattern. If the pattern does not improve, then the pins with a higher value are moved.

Editing the Strategy File

You configure the *constraint driven* mode with a strategy file (`strategy.txt`). You can use the default settings, or you can change the parametric controls and rules in the strategy file to accommodate the requirements of your design.

Make edits to the strategy file in a text editor, based on the keyword descriptions shown in the tables below. Once you have made the changes, save the file and run the *auto assign net — constraint driven* command.

The default strategy file is located in the directory `...\\share\\pcb\\text\\tech`. The first strategy file that is found by searching your `$TECHPATH` setting is the one that is used.

- To use a specific strategy file for your current design, put that file in your current working directory.
- To make your new strategy file the default file used by the system, place it in the directory `...\\share\\local\\pcb\\tech`.

 Be sure to edit the `$TECHPATH` environment variable to reflect the correct location of your customized strategy file.

The default strategy file is shown below:

```
# Default Strategy File

# Auto Assign Net

# Comments are indicated with # signs.

# Keywords can contain no spaces. Each rule must be on a separate line, with the rule name followed by a list of values (depending on the rule) for it. Rule name and values are separated by spaces or tabs.

# Assignment Parametric Controls
HowHardToTry      Low          # Low | Middle | High
DiffPairOrientation Adjacent    # Adjacent | Tandem | Diagonal
AssignFirst        RingToRing  # Middle | Corners | RingToRing
LayerBalance       EvenDistribution
NetLengthBalance  Average     # Average | Minimize | Maximize
MultiNetScheduling Independent # Independent (pin to pin) | MST

# Weighted Rules (0 = unused; 1 = min; 99 = max penalty)
AssignAllNets      99
PhysicalViolation  95
DiffPair           90
MinLength          95
MaxLength          95
RelativeLength     85
MatchLength        80
PinCode            50
SwapCode           25

# End of File
```

The following tables describe the keywords that are supported in the strategy file. Keywords cannot contain spaces. Each rule must be on a separate line in the following format: `<rulename> <value> <value>`

To enter a comment line, precede the text with a number sign (#). For example:

```
# End of File
```

Assignment Parametric Controls

Keyword	Description
AssignFirst	Specifies the priority for which destination pins to use (fill the corners first, fill the middle first, assign ring to ring). Values = Middle Corners RingToRing Default = RingToRing
DiffPairOrientation	Specifies the preferred differential pair pin alignment (parallel, tandem, or diagonal). For example, if you pick <code>Adjacent</code> , then the tool uses this mode if possible, but may still pick <code>Tangent</code> or <code>Diagonal</code> if necessary to get a cleaner assignment. Values = Adjacent Tandem Diagonal Default = Adjacent
HowHardToTry	Specifies how hard the algorithm should attempt completion, either by returning the first solution (low), solving for the best solution (high), or accepting a solution in between (middle). Values = Low Middle High Default = Low
LayerBalance	Specifies a preference for using the minimum possible number of routing layers or for distributing the routing evenly across the available layers. <code>FillLayers</code> optimizes for cost (fewest layers). <code>EvenDistribution</code> optimizes for yield (most consistent metal density). Values = FillLayers EvenDistribution Default = EvenDistribution
MultiNetScheduling	Specifies how the routing should be scheduled if a net has more than one source and one destination (like a power/ground net or a die-die-component 3-pin net). <code>Independent</code> means pin-to-pin. For example, where you have two die pins, each gets assigned its own BGA ball. <code>MST</code> (Minimum Spanning Tree) means the set of connections which minimize the overall length of routing while ensuring that each pin is connected to every other pin on the same net. You only get one component pin using this option. Values = Independent (pin to pin) MST Default = Independent
NetLengthBalance	Specifies the relative relation of how to balance the lengths of nets in the assignment. <code>Average</code> means try to have all flight lines (on the same layer) be as close to the same length as possible. <code>Minimize</code> means keep each flight line (on the same layer) as short as possible. <code>Maximize</code> means keep each flight line (on the same layer) as long as possible. Values = Average Minimize Maximize Default = Average

Weighted Rules (0 = unused; 1 = min; 99 = max penalty for breaking the rule)

Keyword	Description
AssignAllNets	The cost of leaving a pin from the selected source pins as unassigned. A weight of 99 means that leaving the pin unassigned when finished is extremely undesirable. Default = 99
DiffPair	The cost of having the two pins of a diff pair assigned to pins that are not immediate neighbors in the destination set. Default = 95
MatchLength	The cost of making an assignment where the flight lines of the two nets constrained by the rule differ by more than the <code>MATCH_LENGTH</code> electrical constraint rule applied. Default = 80
MaxLength	The cost of making an assignment where the flight line of a net constrained by the rule differs by more than the <code>MAX_LENGTH</code> electrical constraint rule applied. Default = 99
MinLength	The cost of making an assignment where the flight line of a net constrained by the rule differs by more than the <code>MIN_LENGTH</code> electrical constraint rule applied. Default = 95
PhysicalViolation	The cost of derating physical constraints (space and trace size) in order to squeeze in additional nets in a given area. Default = 99
PinCode	The cost of making an assignment that violates the pin use code (power, ground, signal). Default = 50
RelativeLength	The cost of violating relative length to a specified reference net. Default = 85
SwapCode	The cost of violating device swap codes. This considers pin, function, and gate swap information from the symbol to determine legal destination pins for a given set of source pins. Default = 25

- Customized AXL-SKILL functions (available for all element selections)

You can specify customized routing modes for *Logic – Auto Assign Net* (`auto assign net` command) by creating AXL-SKILL functions which you can then load before running the `auto assign net` command. SKILL functions that you load appear in the *Algorithm* list in the Automatic Net Assignment dialog box.

⚠ These SKILL functions are not saved when you exit the tool. You must reload them on each startup or before using *Logic – Auto Assign Net* (`auto assign net` command).

The **axlAddAutoAssignNetAlgorithm** takes the following input parameters:

- algorithm

Case-sensitive text string for the actual name of the function. Takes two list parameters: source pin and destination pin. You cannot use the reserved keywords ROUTER-BASED and NEAREST-MATCH. Other existing algorithms that you pass with a new name overwrite the older name and display the more recent one.

- display name

Case-sensitive text string for the name that you want to associate with the algorithm in the Algorithm list in the Automatic Net Assignment dialog box. You cannot use the reserved keywords ROUTER-BASED and NEAREST-MATCH. Other existing algorithms that you pass with a new name overwrite the older name and display the more recent one.

The following is a sample of a user-defined SKILL algorithm:

```
procedure( myAutoAssignAlgorithm( sourcePins destPins "ll" ) let( ( failedPins )

    ;// Assign a source pin to a destination pin.

    ;// Here, we just assign them in order.

    foreach( pin SourcePins )

        unless( axlDBAssignNet( car(destPins) pin->net )

            failedPins = cons( pin failedPins );

        );/unless

        destPins = cdr(destPins);

    );/foreach

    ;// If your algorithm fails,

    ;// this is what you should do to tell the system that.

    unless( failedPins

        failedPins = "FAIL";

    );/unless

    ;// Return the list of unassigned pins.

    failedPins

);)/_aagnetRunNearestMatch
```

To add this to the list, call:

```
axlAddAutoAssignAlgorithm("myAutoAssignAlgorithm" "Basic Assign"
```

In addition to the routing modes discussed above, the `auto assign net` command lets you filter out power and ground pins, change existing assignments, and create nets for unassigned source pins.

Upon completion of the auto assignment, the tool displays a preview of the approximate routing paths that you can expect the auto router to create when you route your design. The preview gives you an opportunity to visually inspect the routes for potential problem areas that you may want to address prior to routing.

The `auto assign net` command generates the `auto_assign_net.log` file in your current working directory when it completes a run.

Displaying Rats by Layer

As packages continue to increase in layer count and routing complexity, it becomes harder and harder to visualize the routability of the entire design by looking at the straight line "rats" display of the unrouted logical connectivity. These lines are generally enabled globally, by component, or by specific net. However, one of the most useful and informative ways to view them is actually based on the layer on which the net will be routed.

By looking at the nets that are to be routed on a single layer, you can make a quick visual inspection and gauge the level of congestion and probable routability of that layer. This is very helpful, even with a router tool providing accurate path estimations for routability analysis – the lines can become a jumble if not colored or displayed based on the connection's layer.

The Rats by Layer feature (`rats layer` command) provides you with the capability to turn the display of rat lines on or off depending on the net's primary routing layer. You can also permanently highlight nets based on their primary routing layer. These two toggles are independent. For example, you can color all the net connections on the top layer blue, yet keep their rat displays turned off so that you can route metal 1 cognizant of what nets will be going to the layer above.

! This command uses the permanent highlighting capabilities; any permanent net highlighting that you set previously is modified when you set the highlighting with Rats by Layer.

 For the Rats by Layer feature to work, the nets in your design must have the `ASSIGN_ROUTE_LAYER` property set, which defines the primary routing layer for each net. (For more information, see the description of the `assign route layer` command in Allegro PCB and Package Physical Layout Command Reference: A Commands.)

This Rats by Layer feature is designed as a visual aid to support the `auto assign net` command, and any other stage of design where you want to view the net assignments based on the layer where the primary connection will be created.

Automatically Assigning Pin Use Codes

In higher pin-count, multi-chip packages where there are many electrical constraints involved, it is becoming more important to properly set pin use codes. Many times, these are not properly set during the creation of a die, BGA, or other component particularly for components under design, as opposed to library parts. Some commands do not function properly without this information. With the Auto Pin Use Assignment feature, you can set your pin use codes for a component based on its netlist connections to other components in the same design.

To view a demonstration of this feature, see the Silicon-Package-Board Multimedia Library on Cadence Online Support.

Related Topics

- [assign route layer](#)
- [APD: Introduction to the Wire Bonding Toolset](#)
- [auto assign net](#)
- [auto assign pinuse](#)

Defining Connectivity Manually

When a complete netlist is unavailable or the net name in the file is incorrect, it may be more convenient to create a logical net in the database than to import a netlist file. To do so, choose *Logic – Create Net* ([create net](#) command).

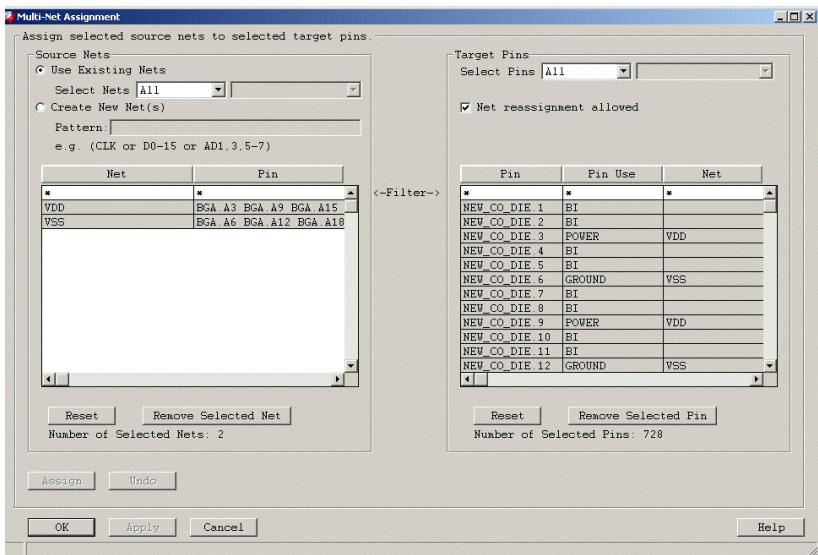
When the assignment was made incorrectly or a new project is based on a previous design where the main difference is the logic (netlist), *Logic – Deassign Net* ([deassign net](#) command) removes pins from an existing net, as described in the *Allegro PCB and Package Physical Layout Command Reference*.

Managing Net Assignments for Multi-Die Packages

APD supports easy net assignment in a multi-die component and the management of a multi-die netlist. With this feature, you can assign a list of nets to a list of pins on a die. If there are no appropriate existing nets to assign to the pins, you can select a list of pins and create a list of nets to assign to them.

Choose *Logic – Assign Multiple Nets* (*assign multi nets* command) from the menu bar to allow assignment of a list of nets to a list of pins (in the following figure). Or, choose *Logic – Auto Create Net* (*auto create net* command) to select only specific pins of a component for which new nets are to be automatically created. New nets are automatically created, based on pin number.

Multi-Net Assignment Dialog Box



The tool creates an `assign_nets.log` file when you use the `assign multi nets` command. It logs the details of all net assignments performed. It also logs any errors or warnings that are uncovered and reports them through the console window.

The following messages can appear in the message window or console log:

Message	Resolution/Description
Error: New net creation aborted due to invalid name list "<pattern>".	An illegal new net name creation pattern was typed into the <i>Pattern</i> field in the dialog box.
Error: New net creation aborted due to invalid name range "<range>".	A name range was typed into the new net name creation <i>Pattern</i> field in the dialog box, but the specification is invalid. Please correct it.
Error: New net creation aborted due to invalid name generation for <net>.	The new net name creation pattern specified in the <i>Pattern</i> field generated a name that is invalid. Please correct the pattern string.
Error: Invalid regular expression "<expr>" entered. Resetting to previous expression.	An invalid regular expression string was typed into one of the <i>Filter</i> fields of the tables or dialog boxes. Please correct it.
Error: Multi-Net Assignment unsuccessful. Info: Try changing your target or source selections.	The database was unable to make a requested net assignment. This message generally indicates a more detailed message.
Error: Must select valid source net and target pin lists before assigning.	Either no target pins were selected, or the number of source nets and target pins does not match, so no assignment can be made.
Error: Cannot create net <net>. Skipping assignment to pin <refdes>.<pin>...	The database is not able to create the requested net.
Error: Cannot assign net <net> to a mechanical pin <pin>. Skipping...	Illegal attempted net assignment. Mechanical pins are not allowed to have net assignments.

Error: Cannot reassign net of pin <refdes>.<pin> to <new_net>, already assigned to FIXED net <old_net>. Skipping...	An attempt was made to reassign a pin to a different net than its current net, but the current property set, so its pin assignments are not allowed to change. To make the reassignment property from the net. You must also enable the <i>Net reassignment allowed</i> check box in the <i>Net Properties</i> dialog.
Error: Reassign is not allowed and pin <refdes>.<pin> is already assigned to <net>. Skipping	An attempt was made to reassign a pin to a different net than its current net, even though it is allowed because the <i>Net reassignment allowed</i> check box is not enabled.
Error: Assignment of net <net> to <refdes>.<pin> failed!	The database is unable to or is not allowed to make the requested net assignment.
Warning: Assignment not possible. No source nets eligible for assignment (allow net creation?)	There are no nets in the <i>Source Nets</i> list, so no assignment can be made. If there are no nets in the list, you should click <i>Create New Nets</i> in the dialog box.
Warning: Assignment not possible. More source nets than eligible target pins.	There are more source nets for the assignment than target pins, so the assignment cannot be made. You must either add more target pins or remove some source nets.
Warning: Assignment not possible. More target pins than eligible source nets.	There are not enough source nets for the number of target pins in the table, so the assignment cannot be made. Either add more source nets or remove some target pins.
Warning: Not all target pins were successfully assigned source nets.	This is a follow-on warning message at the end of the assignment indicating that due to the number of target pins, not all target pins in the table received net assignments successfully.
Info: Filter Source Nets or Target Pins using the dialog box, or select target pins graphically.	A helpful introductory message to guide you as to what you should do after entering the command.
No target pins selected. You may need to adjust your Target Pin filters.	An attempt was made to select <i>Target Pins</i> in the graphical window (either by pick, window, or Group), but no pins that matched the filter were selected. Either try a different selection, change the filter settings and try again.

The following messages appear only in the .log file.

Warning: The following nets in the source list could not be mapped to any pin in the target list:	A list showing any nets from the <i>Source Nets</i> table that were not successfully assigned to any target pins.
Info: The following net assignments were made:	A catalog of all the net assignments made during this session.
Warning: Net name pattern "<pattern>" contains unescaped '-' with no start of range integer preceding it. Although not escaped with a backslash (\-), it will be treated as though it had been escaped.	The new net creation pattern is not allowed to have a hyphen character as a literal '-' that is part of the name. Instead, these have to be escaped using a backslash as "\-". This is because hyphen characters for defining lists of net names following a range pattern (for example, "ABC1-\") case it looks pretty clear that a name range was not intended. Therefore, we have treated the hyphen as a literal hyphen, and we warn you that we have done so.
Info: The following net assignments were made:	The creation of all new nets made by the command are logged to the log file.
Info: pin <refdes>.<pin> already assigned to net <net>. Skipping...	Any case where a request is made to assign a specific net to a pin, but that pin is already assigned to another net.
Info: Pin <refdes>.<pin> successfully assigned to net <net>.	All successful net assignments made by the command are logged to the log file.

APD: Physical Interconnection Creation

Creating physical interconnects involves converting the logical nets to physical conductors that connect the die to the component pins (and, when necessary, to the plating bar). Depending on your die attachment method, converting the nets might include creating and adjusting wire bonds. AllegroX Advanced Package Designer (APD) provides both automatic and interactive tools to route the nets.

Estimating Layers for Flip-Chip Designs

You can estimate the number of layers that you need to place and route your flip-chip design by choosing *Route – FlIP-Chip – Routing Layer Estimation* (*layer estimation command*). A dialog box appears with a report similar to the following:

```
Include Nets with Voltage Property: No
Include Unassigned Pins: No
Escape Distance From Die Outline: 0
Component = U1, # Pins = 1443, # Attempted Escapes = 1443
Subclass          # Escapes % Escapes Pad Size(s)
-----
TOP              292    20.2%   0.0450
LA02             256    17.7%   0.0450
LA03             114    7.9%    0.0450
LA04             108    7.5%    0.0450
LA05             194    13.4%   0.0450
BOTTOM           36     2.5%    0.0450
Unsuccessful     443    30.7%   
```

End of Layer Estimation Report.

On a flip-chip design, the number of routing layers is usually determined by the number of escape layers required for the most complex die in the component. To arrive at the minimum number of layers, the Layer Estimator attempts to escape as many pads as possible on the top layer. Next, all failed escapes are escaped through simulated vias to the next available layer; and so on, until successful completion or layers are exhausted. If you have a number of unsuccessful escapes, you need to add more layers to successfully escape all the pins.

For layer estimation:

- The flip-chip components should not have any clines, vias, or via structures attached to the pins. Other obstacles (such as route keepouts, or BGA vias, for example) are honored by the simulated Die Escape.
- The flip-chip must be mounted on the top layer of the component.
- The simulated Die Escapes are extended out to the die outline.

For the internal layers, via pads are simulated by using the same pad size, shape, and x/y location of the corresponding die-pin on the surface layer. You define the line width and line spacing rules for each layer with *Setup – Constraints*.

For multi-chip packages, the Layer Estimation report provides a layer analysis for each flip-chip component. Place components so that they do not overlap.

Nets with VOLTAGE Property and unassigned pins are not escaped during the simulated die escape.

Next Steps

Once you have the minimum number of layers needed to escape the die, you can perform the following tasks to semi-automate the routing solution.

1. Create the appropriate stack-ups and drive BGA pin and layer assignments. BGA pin and layer assignments then can give the layers that you use to escape each die pad.
2. Add via structures to escape the BGA to other layers. See *Route – Via Structure – Add*.
3. Use the Die Escape Generator to escape the pins. See *Route – Flip-Chip Die Escape Generator*.
4. Perform Bump/ball optimization when you have a better understanding about the direction and the order in which signals will be escaping the die and coming towards the BGA.



You can make successive passes with the Die Escape Generator and "Bump/Ball optimization" solution to get even better results.

5. Route to connect die escapes to BGA pins.

Using the Flip-Chip Die Escape Generator

Creating die escapes is one of the more time-consuming processes in the development of complex, flip-chip based packages. Manual solutions can require many hours of work to generate the desired pattern, with many iterations of placement and rip-up required before a final, usable solution is isolated.

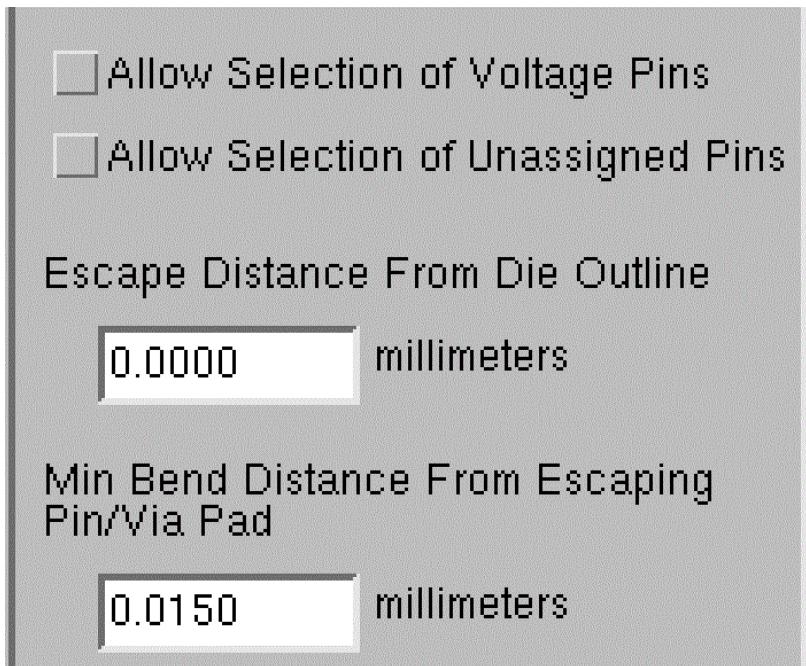
The Flip-Chip Die Escape Generator semi-automates die escaping for complex flip-chip dies, creating a significant reduction in the development time for high pin-count packages.

Flip-chips often require highly complex, multi-layer patterns consisting of often repetitive patterns of traces and vias. The Flip-Chip Die Escape Generator automatically escapes the pins on a single layer that can fit according to set constraints, and permits you to add vias so that you can escape other pins on another layer. You can make subsequent passes on non-escaped pins and added vias to help you quickly escape the pins that you require.

 The Die Escape Generator escapes differential pairs adjacent to each other. A "hug" feature keeps differential pairs as close as possible. Also, you cannot select the following pins:

- Pins with a FIX property
- Pins with nets that have the FIX property
- Pins with nets that have the NO_ROUTE property

To enter the die escape mode, choose *Route – Flip-Chip Die Escape Generator*. The *Options* window pane of the Control Panel appears as follows.



By default, the Die Escape generator does not escape nets with a VOLTAGE property or pins that are unassigned. However, you can generate die escapes for these by checking the boxes in the *Options* window pane, or by enabling these same options from the pop-up menu.

Pins are identified as "Power/Ground" if they belong to a net having the VOLTAGE property, or if their PINUSE property is set to *Power* or *Ground*.

⚠ If you leave the *Allow Selection of Voltage Pins* or the *Allow Selection of Unassigned Pins* unchecked, you cannot select those types of pins.

For related information, see [Wire Bond Die Escape Generator](#).

Creating Power and Ground Plane Distribution

You can add power and ground planes of various configurations, such as solid or crosshatched, to any layer of the design. You can then assign these planes to the appropriate power and ground nets. You can define and modify plane geometries at any point in the design process. Planes can also be "split" to allow multiple power or ground nets to reside on the same layer.

Creating power and ground planes is a combination of setting the cross-section with respect to the layer type for DRC errors and creating shapes (copper areas) on the appropriate *CONDUCTOR* subclass.

There are two types of copper planes: positive planes and negative planes. Positive planes show the locations where copper exists on the plane while negative planes show the locations where copper is removed from the plane.

Understanding Negative Planes

You can add negative planes earlier in the design process. Padstacks that pass through the plane use the padstack anti-pad definition if the pad does not connect to the plane. Padstacks use the padstack thermal-relief definition if the pad does connect to the plane. The tool does not include these definitions until it generates the manufacturing output, so you can move components and vias without having to edit the shape.

After you add a shape, you can edit the shape boundary or add manual voids to positive shapes. You must refill the shape after editing.

Advantages:

- The artwork file required to plot the copper area is much smaller because no data is required to fill the polygon.
- This type of copper area is more flexible; it can be created early in the design process and accommodates dynamic placement and routing changes.

Disadvantage:

- You must build flash symbols (`.fsm`) for all padstack flash names to see these features while viewing Gerber files in APD (otherwise, triangle symbols are substituted).

Understanding Positive Planes

Positive planes have voids that you can see in the tool, along with the connections to the positive plane. When you add the voids and connections to the shape, they are static. If you move a component or via, you must edit the shape to move the voids and connections. It is better to add positive planes after routing as the router cannot drop vias through a positive plane.

Advantage:

- APD displays WYSIWYG (that is, it displays the actual copper fill as well as the anti-pad and thermal relief features; no special flash symbols are required).

Disadvantages:

- For non-rasterized output, the artwork file is much larger. You also need to fix shape fill errors.
- For placement or routing changes, you must regenerate the shape to void the new or changed objects.

After you add planes to your design, you can quickly and easily connect the appropriate die and I/O pins to each plane; APD maintains the correct antipad clearances for drills that go through, but do not make contact with, each plane. You can define split and antipad clearances along with thermal pad geometries. You can also use a parameter dialog box to modify the size, direction, border width, and angle of crosshatched planes.

Prerequisites to Defining Planes

Before you define planes:

1. Identify the plane layers.
2. Verify that pads and vias have the appropriate definitions for plane layers to get the proper thermal connection or antipad clearance.
3. Identify the nets that connect to planes.
4. Plan the line size, angle, direction, and spacing for crosshatched planes.

Creating a Negative Plane

Although the tool displays the copper area as a filled polygon, this image is reversed, and the polygon appears as a clear area on the artwork. The tool evaluates the connectivity of all pins and vias within the polygon based on the padstack definition. These pins and vias become black circles (no connects) or thermal relief patterns (connects) on the artwork.

If you use negative planes in your design, it is usually easier to add these copper areas before routing. When you create the artwork for your design, this layer is plotted as a negative image and the plot shows where copper is etched away.

1. Choose *Shape – Polygon* ([shape add](#) command) or *Shape – Rectangle* ([shape add rect](#) command).
2. In the *Options* window pane of the Control Panel:
 - a. Toggle the *Class* field to *CONDUCTOR* and the *Subclass* field to the layer on which you want to work.
 - b. Choose *Dynamic copper* or *Static solid* as the shape fill type.
 - c. Click *Browse* to display the Select Net dialog box.
 - d. Choose the correct name from the list of net names.
The net name is displayed in the *Options* window pane.
3. Draw a polygon within the design outline for the plane. You must keep this polygon inside the route keepin area. Zoom in so you can see the route keepin boundary.
If you do not see the shape as you digitize the points, check your *Options* window pane settings and your visibility settings.
4. To close the polygon, right-click, and choose *Done* from the pop-up menu.
The copper area fills solidly, and the thermal-relief and anti-pad features appear.

When you generate manufacturing output for this layer, any padstack passing through this plane uses its thermal-relief pad definition to attach to the plane or the anti-pad definition to create a clearance from the plane.

Creating a Positive Plane

If you use positive planes in your design, create them after routing. If you create them before routing, the router cannot drop vias through a positive plane.

1. In the *Visibility* tab of the Control Panel turn on the visibility for *Pin*, *Via*, and *Conductor* for the layer you are working on.
2. Choose *Shape – Polygon* ([shape add](#) command) or *Shape – Rectangle* ([shape add rect](#) command).
3. In the *Options* window pane of the Control Panel:
 - a. Toggle the *Class* field to CONDUCTOR and the *Subclass* field to the layer on which you want to work.

- b. Choose *Dynamic copper* or *Static solid* as the shape fill type.
 - c. Click *Browse* to display the Select Net dialog box.
 - d. Choose the correct name from the list of net names.
The net name appears in the *Options* window pane.
4. Draw a polygon within the design outline for the plane. You must keep this polygon inside the route keepin area. Zoom in so you can see the route keepin boundary.
If you do not see the shape as you digitize the points, check your *Options* settings and your visibility settings.
 5. To close the polygon, right click and choose Done from the pop-up menu.
 6. Choose *Shape – Global Dynamic Params* ([shape global param](#) command).
The Shape Parameters dialog box appears, which you can use to control the voids or clearances that the tool places around objects on this layer.
 7. Set any shape parameters, and click OK to close the dialog box.
 8. Run the appropriate *Shape – Manual Void* command.
This command can take a few seconds to execute. A message states that the tool is creating voids, performing autovoiding, and connecting thermal-relief pads. Also the Shape log dialog box appears.
 9. Exit from the log dialog box by clicking *Close*.
 10. In the Design Window, right click and choose Done.
 11. Zoom in to see the thermal-relief pads and anti-pads.

Creating Power and Ground Rings

The **Power/Ground Ring Generator** lets you define and place one or more shapes in the form of a ring around the die. The PWR/GND Wizard creates up to 1000 rings (shapes) at a time. If you require more rings, you can run the PWR/GND Wizard as many times as needed. This command displays a Wizard in which you can specify:

- Number of rings to be generated
- Placement of the rings from a specified origin, distance from the edge of the die or from the nearest die pin
- The distance between rings
- The width of each ring

- Corner types on each ring (arc, chamfer, and right-angle)
- An assigned netname for each ring
- A label for each ring

The rings are basic in nature. For other shape geometries or split rings, choose *Shape – Polygon* or *Shape – Compose/Decompose Shape* from the menu in the Design Window.

Depending on the options selected, the Power/Ground Ring Wizard dialog boxes change, representing how the rings will be created. Verify the dialog box settings to ensure that the rings are created as intended. For details on using the Power/Ground Ring Wizard, see [Ring wizard](#) in the *Allegro PCB and Package Physical Layout Command Reference*.

Creating a Shorting Scheme

APD can accommodate complex power and ground distribution schemes that are sometimes required to control simultaneous switching noise (SSN). Based on those requirements, you may need to connect power and ground pins, attached to the same logic, to different supply planes.

Prerequisites to Defining a Shorting Scheme

Before you can define a shorting scheme interactively, your design must have the following:

- A netlist
Your netlist must use one net name for all the common power nets, such as VCC, and another net name for all the common ground nets, such as VSS. Your netlist can contain subnets.
- Power and ground planes
Use the appropriate *Shape* commands from the menu to create power and ground planes, then label them as planes in the Layout Cross Section dialog box.
The SUBNET_NAME is defined based on association, not name.
- Generic blind or buried via padstack
You create a generic via padstack using the Padstack Editor. Define pads on the layers that connect the blind or buried via. The via padstack should include padstack definitions for every layer.
- Via template
Define a via template in the database that defines all layers. Use the template to remove unnecessary pads and shorting vias.

Defining a Shorting Scheme

A shorting scheme is used only on power and ground nets. To define a shorting scheme, attach the SHORTING_SCHEME property to pins and vias in the nets or subnets connected to power or ground planes. The SHORTING_SCHEME value must match either the net name or subnet name of the power or ground planes. See the *Allegro PCB and Package Physical Layout Command Reference* for details on [creating a short](#).

Adding Wire Bonds

Wire bonds connect the die pins to the bond fingers. Prior to automatically assigning nets, you may want to connect wire bonds from the die pin to the bond finger. You can make these connections interactively or automatically.

To add wire bonds, the die pin must be on a *DIESTACK* layer type. Bond fingers are single-layer padstacks that you define on the top conductor layer only. The bond wire object connects the two items on the two different layers. The wires are categorized by their wire profile assignment.

Interconnecting Wire Bonds

To add bond wires and bond fingers for selected die pins based on the bond pattern that you specify (prior to auto assigning pins to a net), choose *Route – Wire Bond – Select* ([wirebond select](#) command) to automatically add wire bonds (along with the bond fingers) for a selected set of die pins. You can choose to place the wires orthogonally or radially.

The bond wires and bond fingers that the tool creates have these characteristics:

- Wire width determines the diameter of the bond wire.
- Automatic routers cannot move or delete wire bonds.
- The tool attaches the [BOND_PAD](#) property to bond fingers.
- The tool attaches the [ALIGNED](#) property to bond fingers if you select the *Align Pads With Wires* option.

 If the [ALIGNED](#) property is not attached, an orthogonal bond finger remains orthogonal if its bond wire is moved from orthogonal to angled, which may cause a DRC error with the adjacent bond finger. If a bond finger is not maintaining alignment, add the [ALIGNED](#) property to the bond finger by running the `property edit` command.

Auto Wire Bonding for Power and Ground Rings

APD assesses the area between the die edge and the region where the bond fingers will be placed for any power or ground rings. Die pins assigned to the same net as the rings are wired to the ring. A shape with the [VOLTAGE](#) property (or zero for ground; otherwise, it is a power net) defines a power or ground ring.

The tool uses the [WB_TACKPOINT](#) padstack if a suitable existing padstack does not exist. For bonding to rings, the tool always places the bond finger along the shape's centerline.

- ⚠** The WB_TACKPOINT padstack or any suitable padstack, if one exists, will be created with the smallest diameter possible as the pad does not exist in the final design. To change the diameter of the pad created, set the value of the *wirebond_auto_ring_pad_diameter* variable under *Wirebond* in the *IC_packaging* category to the desired diameter, including the design units if the design units are not to be used.

To determine the bond finger locations, the tool starts with the middle die pin unless there is an even number of bond fingers. Depending on whether it is a power, or ground, or IO net, a location and padstack are determined accordingly. For each subsequent die pin, the location of the bond finger is chosen based on the specified spacing values.

The Wire Bonding Toolset comprises a number of wire bonding functions. See Chapter 8.

Generating Offset Vias

The Offset Via Generator lets you create vias for one, some, or all I/O pins in a component. You also have the following options from which to choose before generating the offset vias:

- Choose the angle of the offset vias to be in alignment with the component origin or at a 45-degree angle to the origin.
- Specify that vias are on the inside (between the pin and the origin) or on the outside (with the pin between via and origin).
- Limit the creation of offset vias to pins that have been assigned to a net. You can also create offset vias for selected pins using the *Find* tab.
- Specify the distance between the center of the pin and the center of the via. You can also place vias directly at the pin location, extending its penetration through the layers.
- Automatically generate a simple two-pad padstack (and save to disk), search through a drawing database for a desired padstack, or load a padstack from disk.
- Automatically create fillets between the pins and the vias.
- Set line widths for non-filletted lines.

If component pins are too close to allow wire escape routes, or if component pins and die pins are on opposite sides of the substrate, you can automatically generate wire escape routes through vias that are offset from the component pins.

As shown in the drawing, an offset via is a via with a short piece of trace which is connected to a component pin. Pitch refers to the distance between the center of the pin and the center of the via.



This drawing also shows fillet lines, which is an optional feature you can use when you create offset vias.

You create offset vias by choosing *Generate – Offset Via Generator* ([offset via gen](#) command).

APD: Introduction to the Wire Bonding Toolset

Wire bonding is one of two common methods currently used to electrically connect an integrated circuit to a component. It is the lower-cost alternative which is most frequently used today, while flip-chip mounting is used for high-speed, increased density designs.

The arrangement of the bond fingers and bond wires is a complicated and time-consuming process involving the combination of many rules, placement parameters, and strategies. To generate the final pattern, you need to rely on automated tools for managing detailed spacing, while also using interactive commands for fine tuning results to meet specific electrical, physical, and design rules.

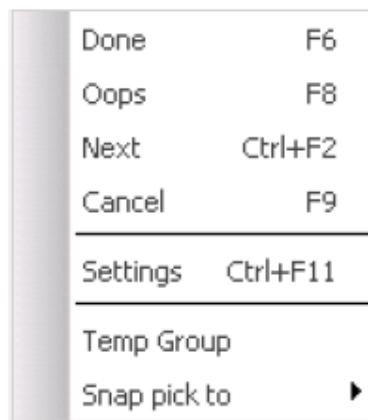
- [How the Wire Bond Tools Work](#)
- [Setting Up a Wire Bond Design](#)
- [Auto Wire Bonding](#)
- [Populating the Guide Path with Bond Fingers](#)
- [Non-Wired Bond Fingers](#)
- [Wire Bond Tack Point](#)
- [Routing Stubs](#)
- [Creating and Modifying Wire Profiles](#)
- [Wire Bond Via Estimation](#)
- [Wire Bond Die Escape Generator](#)
- [Wire Bond Use Models](#)

How the Wire Bond Tools Work

- [Menu Available before Element Selection](#)
- [Wire Bond Heads-up Display](#)
- [Using the Wire Bond - Select Command](#)
- [Pause/Resume Commands](#)
- [Set Default Action Command](#)
- [Other Wire Bond Commands](#)
- [Managing Connections for a Wire Bond Die](#)
- [NO_WIREBOND Property](#)

Menu Available before Element Selection

The following table describes the menu items available when you run any wire bond command (for example, *Route – Wire Bond – Select*), but have not selected any elements in the Design Window. This menu appears when you right-click to display the pop-up menu.



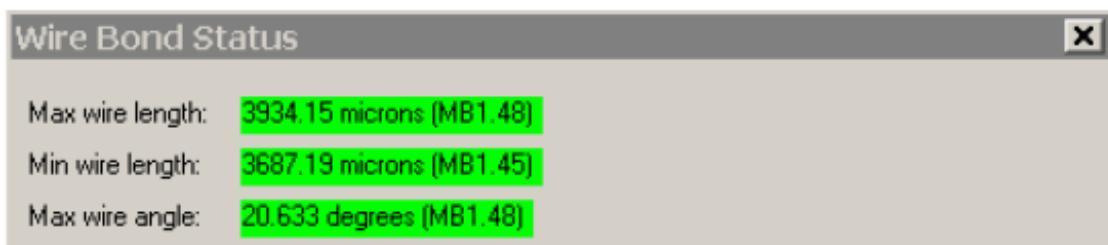
<i>Done</i>	Exits the wire bond session, saving all changes made.
<i>Oops</i>	Rolls back the last operation performed, staying inside the wire bond tool.

<i>Next</i>	Saves all changes so they are not affected by <i>Oops</i> or <i>Cancel</i> operations. This also sets a new undo/redo point for use with those commands after you exit the wire bond tool.
<i>Cancel</i>	Exits the wire bond environment, undoing any changes since the last <i>Next</i> event or since the command started, whichever is most current.
<i>Settings</i>	Invokes the wire bond settings command to view and edit the settings for all wire bond applications.
<i>Temp Group</i>	Allows access to the standard temp group routines for purposes of making complex, multi-item selections in the Design Window.
<i>Snap pick to</i>	Lets you choose a snap mode from a list of options found on the right mouse button pop-up submenu when in an interactive editing command.

Wire Bond Heads-Up Display

The heads-up display appears whenever you modify the bond finger pattern. It shows key values for bond wires (minimum and maximum wire lengths and maximum wire angle), violations, and lets you evaluate the proposed placement.

To hide the heads-up display, choose *Setup – User Preferences* (`enved` command) to access the User Preferences Editor. Check the `WIREBOND_NO_SHOW_HUD` environment variable box, located in the *Wirebond* category under *Ic_packaging* in the User Preferences Editor.



When you use the *Pause* command, the heads-up display appears as follows:



<i>Max wire length</i>	Specifies the maximum length of any bond wire in the currently active set of wire bonds. Also lists the die pad number to which this wire bond is connected.
<i>Min wire length</i>	Specifies the minimum length of any bond wire in the currently active set of wire bonds. Also lists the die pad number to which this bond wire is connected.
<i>Max wire angle</i>	Specifies the maximum angle (relative to the die side) of any bond wire in the currently active set of bond wires. Also lists the die pad number to which this bond wire is connected. Normally, this is the last pin on one of the two sides of the selected items.

Using the Wire Bond – Select Command

The *Route – Wire Bond – Select* (`wirebond select`) command is the main access to the wire bond tools. You can access all capabilities except for configuration import and export through this command. The results are based on the items you select in the Design Window and the item you choose from the context-sensitive mouse pop-up menu. Using the `wirebond select` command means that you can stay in the command and perform multiple tasks.

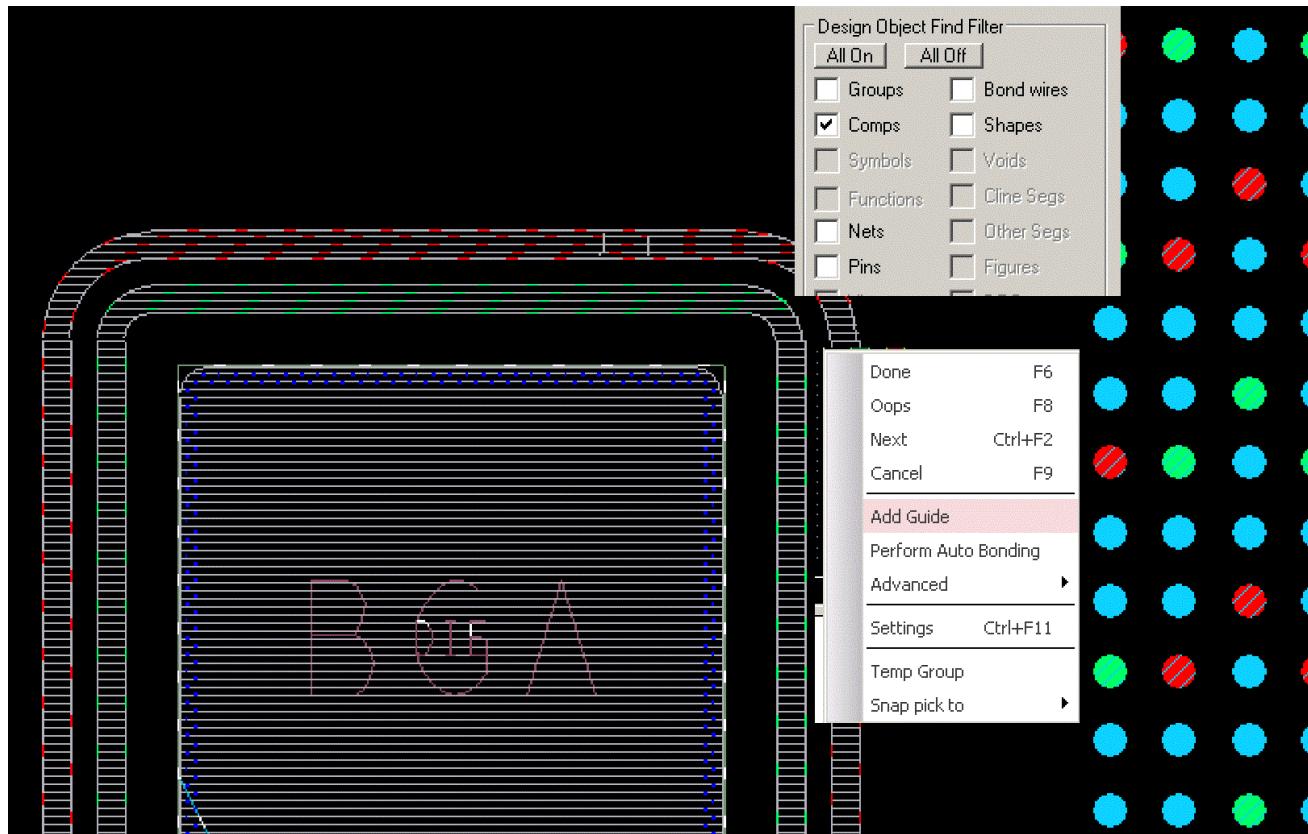
⚠ You can perform wire bonding tasks from die pins, discrete component pins, or from die-stack interposer pads.

Selecting a Component to Add Guide Paths to Your Design

You can add a bond finger guide path to your design when you choose *Route – Wire Bond – Select* (`wirebond select`) from the menu, highlight a component in the design, and right-click to choose the *Add Guide* menu item.

You can also select a component to perform auto bonding. For additional information on these procedures, see the `wirebond select` command.

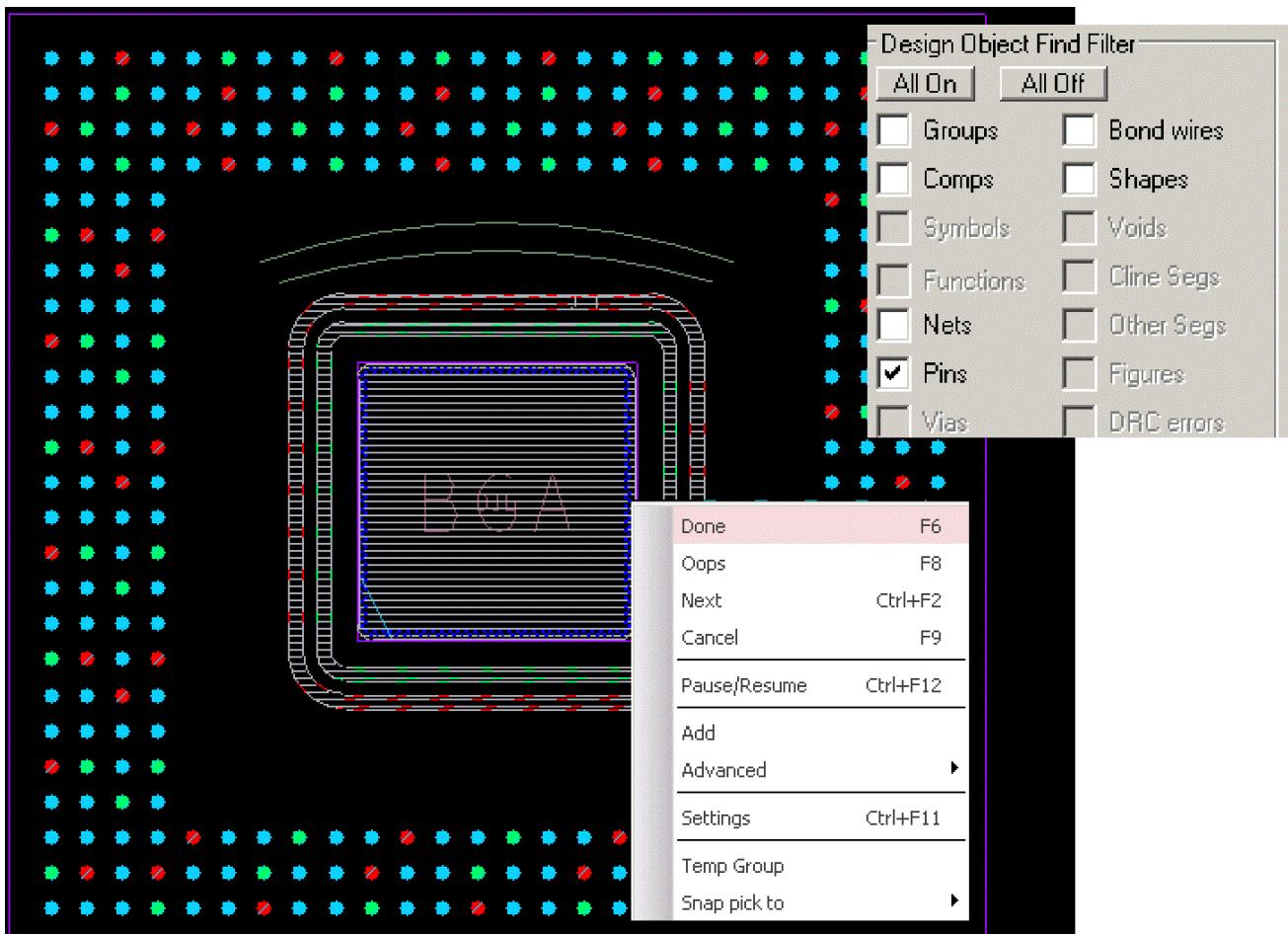
Selecting a Component to Add a Guide Paths to Your Design



Selecting Pins to Add Wire Bonds to Your Design

You can add wire bonds to your design when you choose *Route – Wire Bond – Select (wirebond select)* from the menu, highlight the pins in the design, and right-click to choose the *Add* menu item.

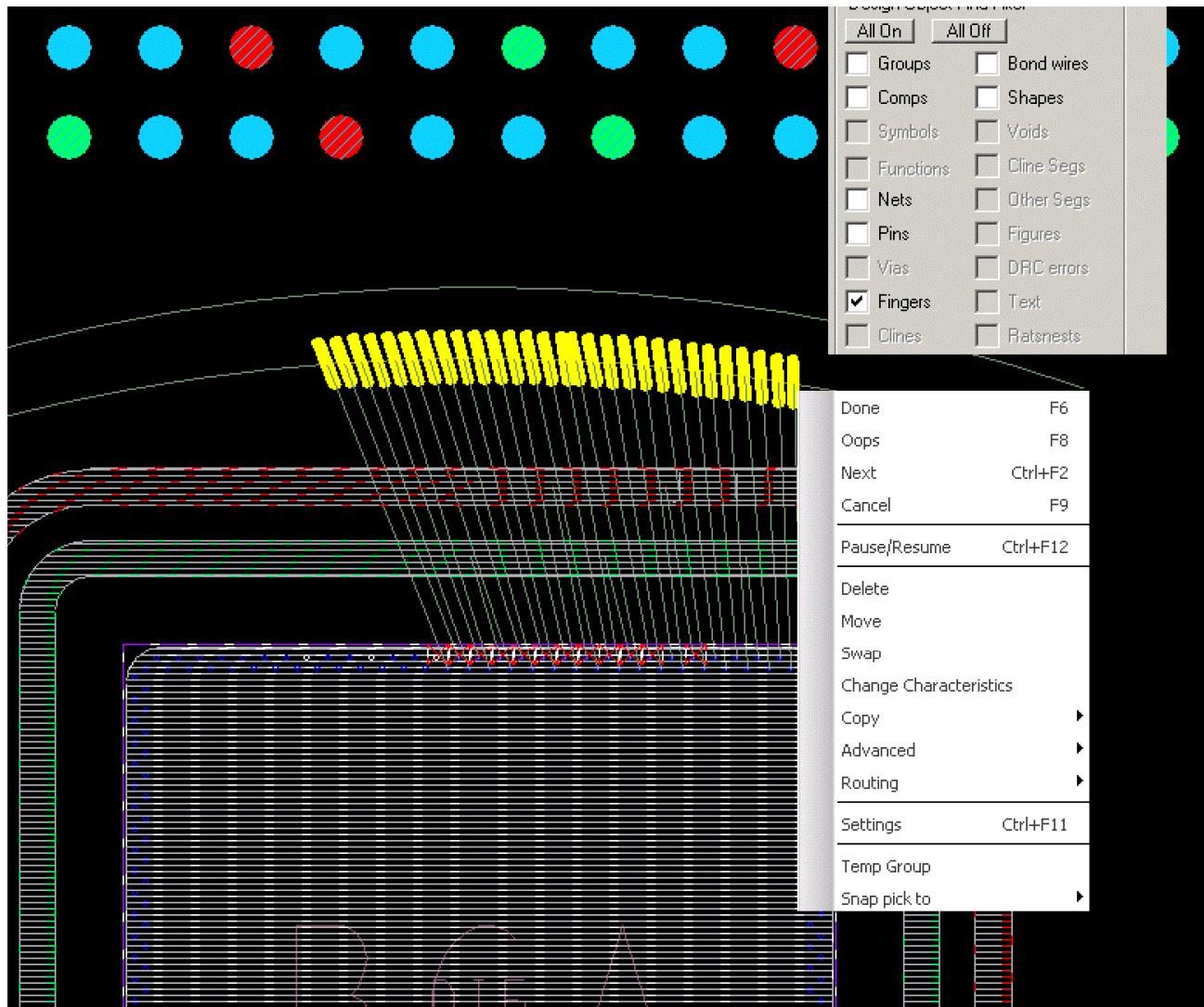
Selecting Pins to Add Wire Bonds to Your Design



Selecting Bond Fingers in Your Design to Perform Tasks

When you choose *Route – Wire Bond – Select* (wirebond select) from the menu, select bond fingers (*Fingers* in the Find Filter) in your design, and then right-click in the Design Window, the pop-up menu displays all the tasks that you can perform.

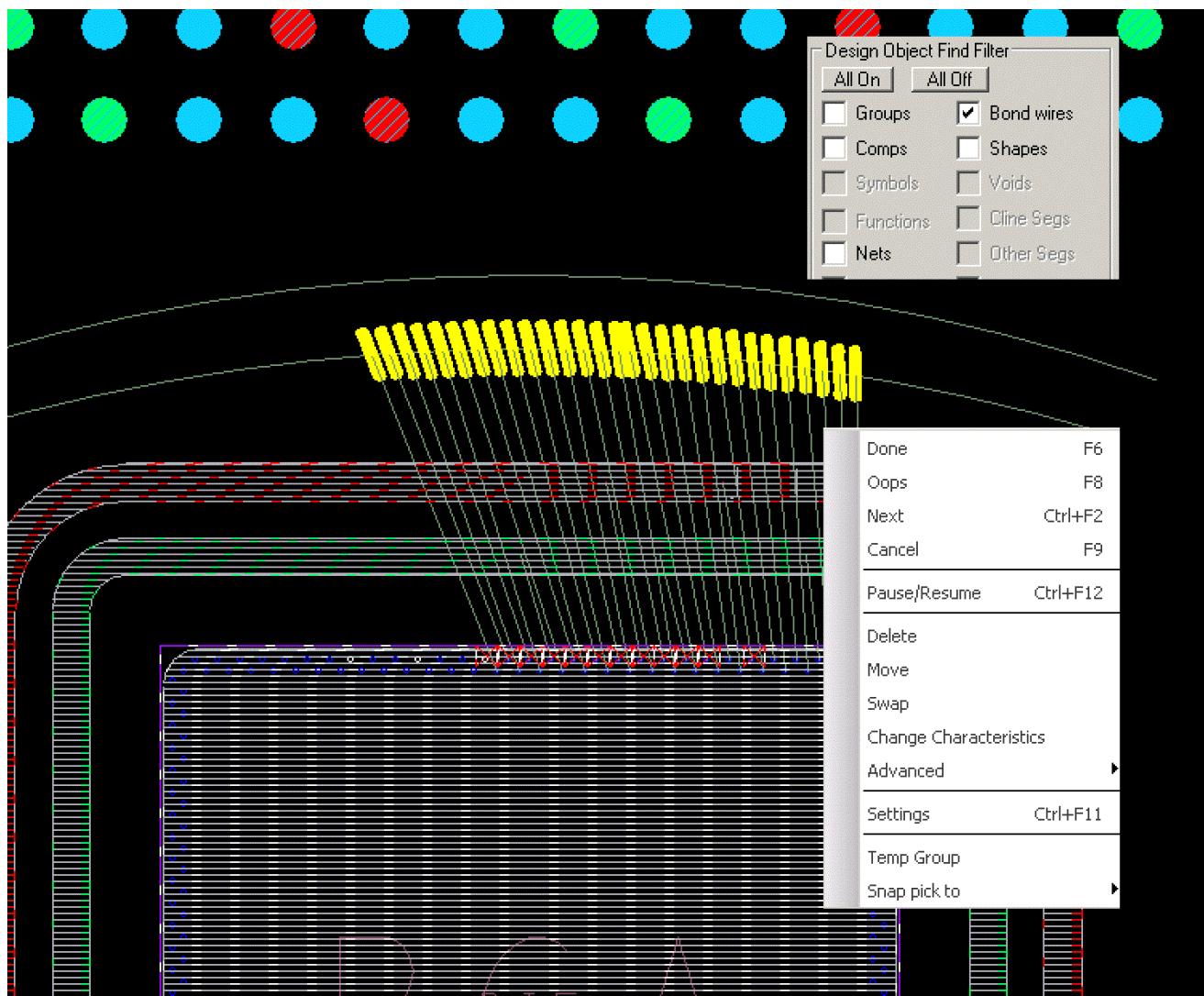
Selecting Bond Fingers to Perform Wire Bond Tasks



Selecting Bond Wires in Your Design to Perform Tasks

When you choose *Route – Wire Bond – Select* (wirebond select) from the menu, select bond wires (*Bond wires* in the Find Filter) in your design, and right-click in the Design Window, the pop-up menu displays all the tasks that you can perform.

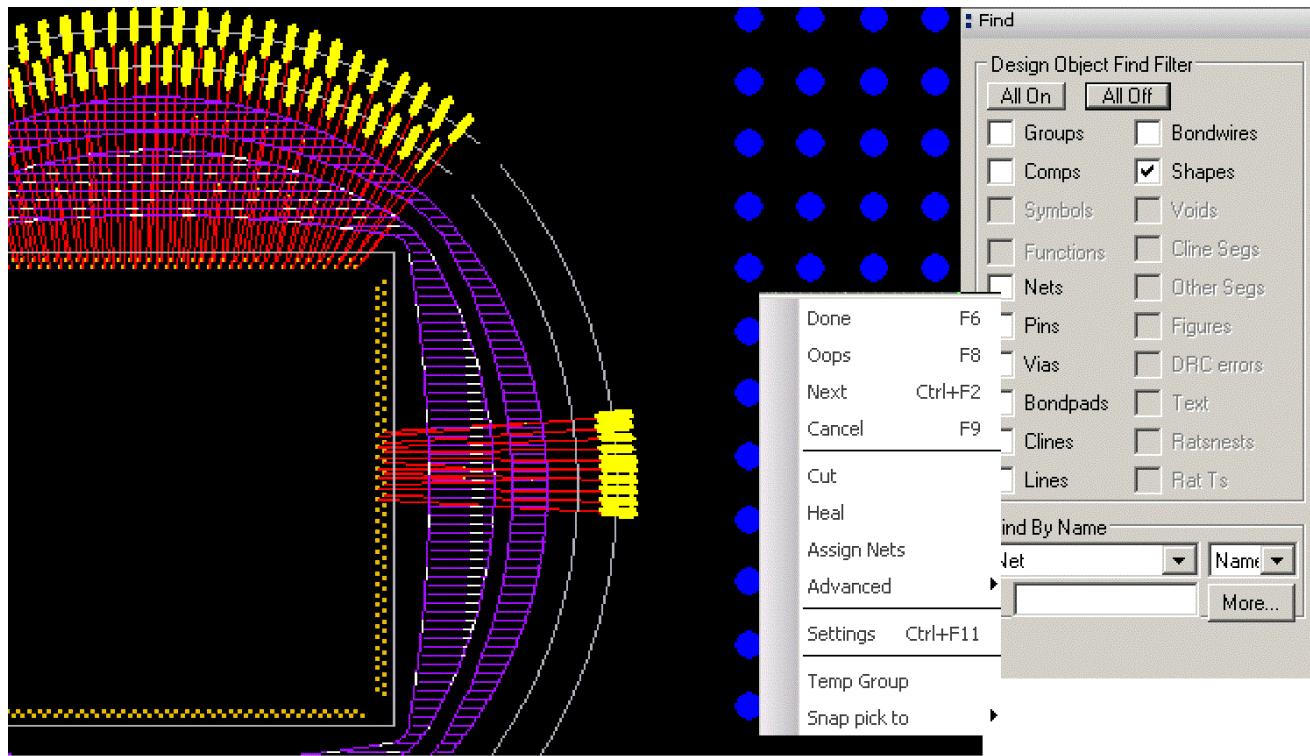
Selecting Bond Wires to Perform Wire Bond Tasks



Selecting Rings in Your Design to Perform Wire Bond Tasks

When you choose *Route – Wire Bond – Select (wirebond select)* from the menu, select rings (*Shapes* in the Find Filter) in your design, and right-click in the Design Window, the pop-up menu displays all the tasks that you can perform.

Selecting Rings to Perform Wire Bond Tasks



Pause/Resume Commands

The *Pause/Resume* menu item is available in all wire bond commands that require interactive placement such as adding or moving wire bonds. While paused, the tool does not process any cursor movements until the tool returns to the active state. Using this menu item alerts the tool that you are moving the cursor outside the Design Window to modify the *Options* window pane settings or to perform some other activity. This is useful to avoid having the elements on the cursor move when you move the cursor to the *Options* window pane.

When you use the *Pause* command, the heads-up display indicates that you are in a paused state.

Set Default Action Command

The *Set Default Action* menu item indicates that the tool, by default, should take the current action the next time you select items of this type. This applies to all item selections that have more than one operation associated with them.

For example, if you have a design with a bond finger guide path and some wire bonds, to use the *Set Default Action* command:

1. Run the `wirebond select` command.
2. Choose *Fingers* in the Find Filter and choose the bond fingers in the Design Window.
3. Right-click and choose an action such as *Delete*.
4. Click in the window to complete the command.
The bond fingers are deleted.
5. Right-click and choose *Set/Unset Default Action* from the pop-up menu.

The next time that you perform this task, when you choose bond fingers in the Design Window, the tool automatically deletes them.

This setting is stored in the `env` file and is saved from session to session until you unset this action.

Other Wire Bond Commands

In addition to using the `wirebond select` command, you can also use the commands listed in the following table to perform wire bond tasks.

Using Other Commands to Perform Wire Bond Tasks

To...	Use this Command...
<ul style="list-style-type: none"> • Add, edit, move, or copy a bond finger guide path • Populate the guide path with bond fingers 	<i>Route – Wire Bond – Add/Edit Guide Paths</i> (<code>wirebond manage guide paths</code>)
<ul style="list-style-type: none"> • Import the configuration of a wire bond group; optionally import the guide paths and power and ground ring shapes, plus the placement information for all associated fingers and wires. 	<i>Route – Wire Bond – Import</i> (<code>wirebond import</code>)
<ul style="list-style-type: none"> • Export the configuration of a wire bond group; optionally export the guide paths and power and ground ring shapes, plus the placement information for all associated fingers and wires. 	<i>Route – Wire Bond – Export</i> (<code>wirebond export</code>)

<ul style="list-style-type: none"> Change wire bond parameters such as setting the global wire bond constraints, specifying the orientation to use when placing bond fingers, configuring for and automatically updating bond finger labels, controlling the snapping of guide fingers to the guide path, and allowing DRC errors during wire bond manipulations. 	<i>Route – Wire Bond – Settings</i> (wirebond settings)
<ul style="list-style-type: none"> Define a bond finger padstack, add wire bonds (including fanout stubs), or add non-wired bond fingers. 	<i>Route – Wire Bond – Add</i> (wirebond add)
<ul style="list-style-type: none"> Delete, move, or swap bond fingers Space bond fingers evenly Adjust bond fingers to meet DRC minimum requirements Delete, move, swap, center, or adjust wire bonds Space wire bonds evenly Center wire bonds between fingers Add routing channels Delete routing channels Create a ring segment from a set of placed bond fingers Remove the merged finger shape that merges multiple bond fingers together 	<i>Route – Wire Bond – Edit</i> (wirebond edit)
<ul style="list-style-type: none"> Add non-standard wirebond fingers 	<i>Route – Wire Bond – Add/Edit Non-Standard</i> (wirebond add nonstandard)
<ul style="list-style-type: none"> Redistribute bond fingers across multiple guide paths 	<i>Route – Wire Bond – Add/Edit Guide Paths</i> (wirebond manage guide paths)

- Move the endpoint of a bond wire within the bond finger area

*Route – Wire Bond
– Tack Point
Move (wirebond
tack point)*

Managing Connections for a Wire Bond Die

If you modify a wire bond die (ECO), for example change a part or shrink the die, and want to replace the existing die with a new die, the tool automatically manages the connections when you use the following commands:

- Add – Standard Die – DEF (def in)*
- Add – Standard Die – Die Text-In Wizard (die text in)*
- Add – Standard Die – OA (oa in)*
- Add – Standard Die – Die Generator (die generator)*
- Edit – Die Generator (die editor)*

 If you have a wire bond co-design die and you are updating the die from the Cadence I/O Planner, the Wire Bond Die Replace dialog box appears. You must complete the settings.

NO_WIREBOND Property

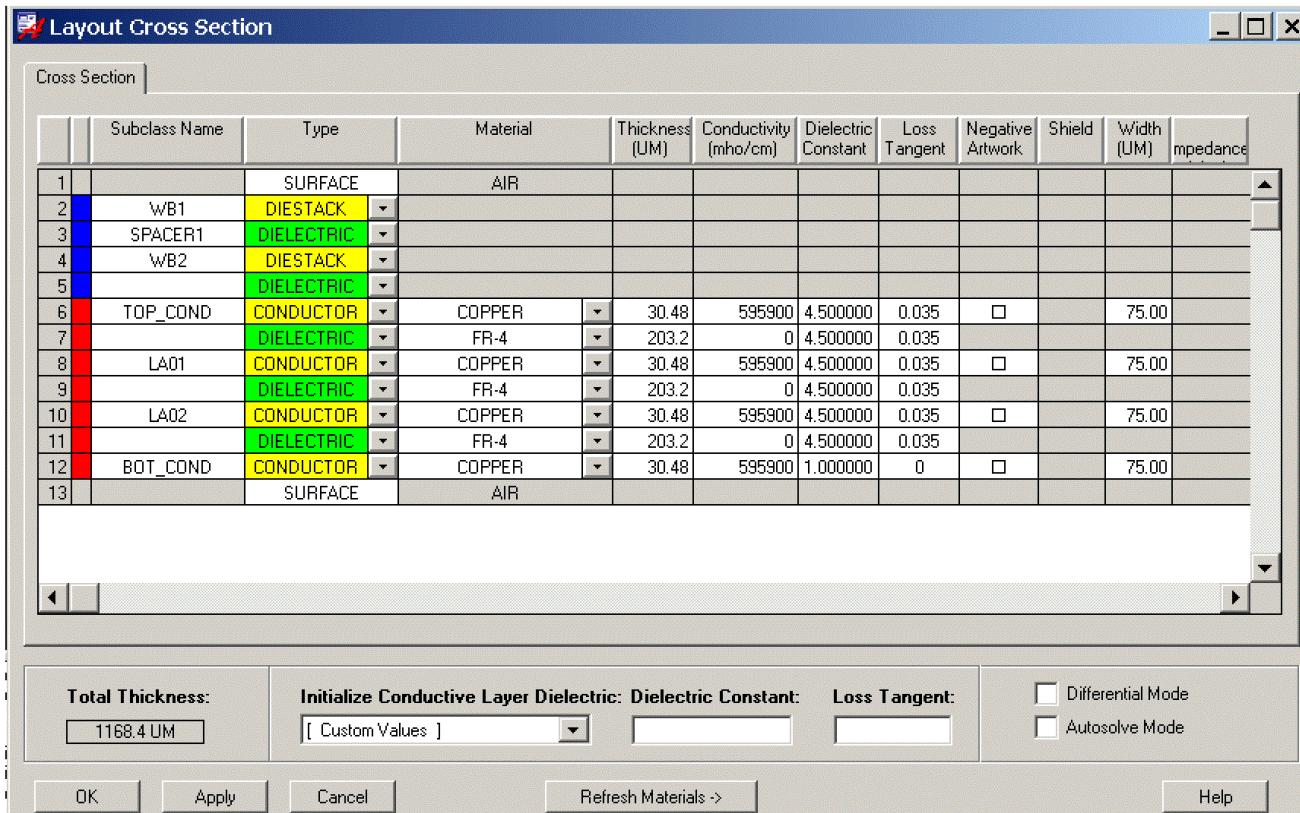
The NO_WIREBOND property, attached to nets or pins, causes the layout tool to automatically exclude these elements from wire bond add operations.

Related Topics

- [enved](#)
- [wirebond select](#)
- [wirebond manage guide paths](#)
- [wirebond import](#)
- [wirebond export](#)
- [wirebond settings](#)
- [wirebond add](#)
- [wirebond edit](#)
- [wirebond add nonstandard](#)
- [wirebond manage guide paths](#)
- [wirebond tack point](#)
- [def in](#)
- [die text in](#)
- [oa in](#)
- [die generator](#)
- [die editor](#)

Setting Up a Wire Bond Design

The following sections describe the recommended setup of bond fingers, die pads, and bond wires in a wire bond design. The example shown below is used in the description.



Bond Fingers

A bond wire automatically connects between objects on two different layers: the die pad on one layer and the bond finger on the CONDUCTOR layer. With this change, you can re-use the same bond finger padstack across multiple designs, regardless of the cross-section and number of die layers in each of the designs.

Build the padstack for a bond finger with the pad shape and size of the physical substrate pad on the top CONDUCTOR layer. Always define the oblong pad facing east, that is, with the width being greater than the height.

See *Defining Padstacks for Bond Fingers* in the `wirebond select` command.

Die Pads

Be sure that you define all the die pads on one DIESTACK layer so that the 3D Canvas and analysis tools can properly process the design. APD will use the layer ordering information to determine the die's position within the die stack.

Bond Wires

Bond wires are 3D entities that curve through space to connect a die pin to a package substrate finger or another die pin. The exact path that each wire in a design follows through space is unique. However, groups of similar wires use a similar path, called a wire profile.

With this release, bond wires are no longer tied to a physical layer in the design substrate because they are 3D connections between items on two different layers in the design.

Groupless Wire Bonds

The wire bond use model uses groupless wire bonds.

Previously, you grouped together sets of wire bonds with common characteristics. You added wire bonds and they automatically became members of the same group and acted the same when manipulated. You could change any characteristic of the group and it replicated to all members of the group. To change one or more characteristics in a sub-group of wire bonds, you created a new group, added these wire bonds to the new group, and changed the group characteristics.

In Release 16.3, wire bond groups have been removed. Instead, the attributes previously stored at the group level are now stored directly on the individual bond wires and fingers. This includes characteristics such as wire profile, finger placement style, and finger padstack. The tool stores default values for all these characteristics at the design level when you configure global wire bond settings using the `wirebond settings` command. With this new use model, you can create single-die designs efficiently, and lay out and refine more complex, stacked-die, custom designs where the wire bonds have multiple patterns, bond finger padstacks, and wire profiles.

Additionally, you can further customize the placement of bond fingers because each finger stores its own setting for its path, snap location, and rotational alignment. Previously, these settings were set globally for the entire design.

How You Use the Groupless Wire Bonds Use Model

Before you wire bond your die components, you should set up default wire bond characteristics including padstack, placement style, alignment, and the snap type for bond fingers using the `wirebond settings` command. You should also load the appropriate vendor-supplied (or custom-designed) wire profile definitions from your library.

Although these default characteristics are used during the bonding operation, you can change individual characteristics as you add or move wire bonds. During an *Add* operation, the default characteristics you defined using the `wirebond settings` command are used to bond the pattern, but you can change the settings on the *Options* window pane of the Control Panel or through the right-mouse button menu. On subsequent *Add* operations, the latest settings defined are used as the default settings. During a *Move* operation, you can also change the characteristics on the wire bonds. Once the wire bonds are placed, the new settings are saved on the bond finger and bond wire.

You can select a set of wire bonds and assign them common characteristics using the right-mouse button *Change Characteristics* menu command (see , described in the `wirebond select` command). Their characteristics do not have to be the same initially. Making changes may lead to slight placement changes for the bond fingers if the new setting is different than the current placement or finger alignment.

- [wirebond import](#)
- [wirebond export](#)
- [allegro_uprev](#)
- [downrev](#)
- [Changing the Characteristics of a Wire Bond](#)

Auto Wire Bonding

During the early stages of the wire bonding process, you can use the Auto Bond command to quickly define a set of parameters and constraints and then perform wire bonding. This process lets you assess the process of bonding a die, and provides a quick prototype of the general finger pattern style for the pin layout. You can modify the parameters multiple times until you produce a suitable pattern. Then you can use the interactive wire bond tool set to modify the wire bonding pattern in more detail.

Related Topics

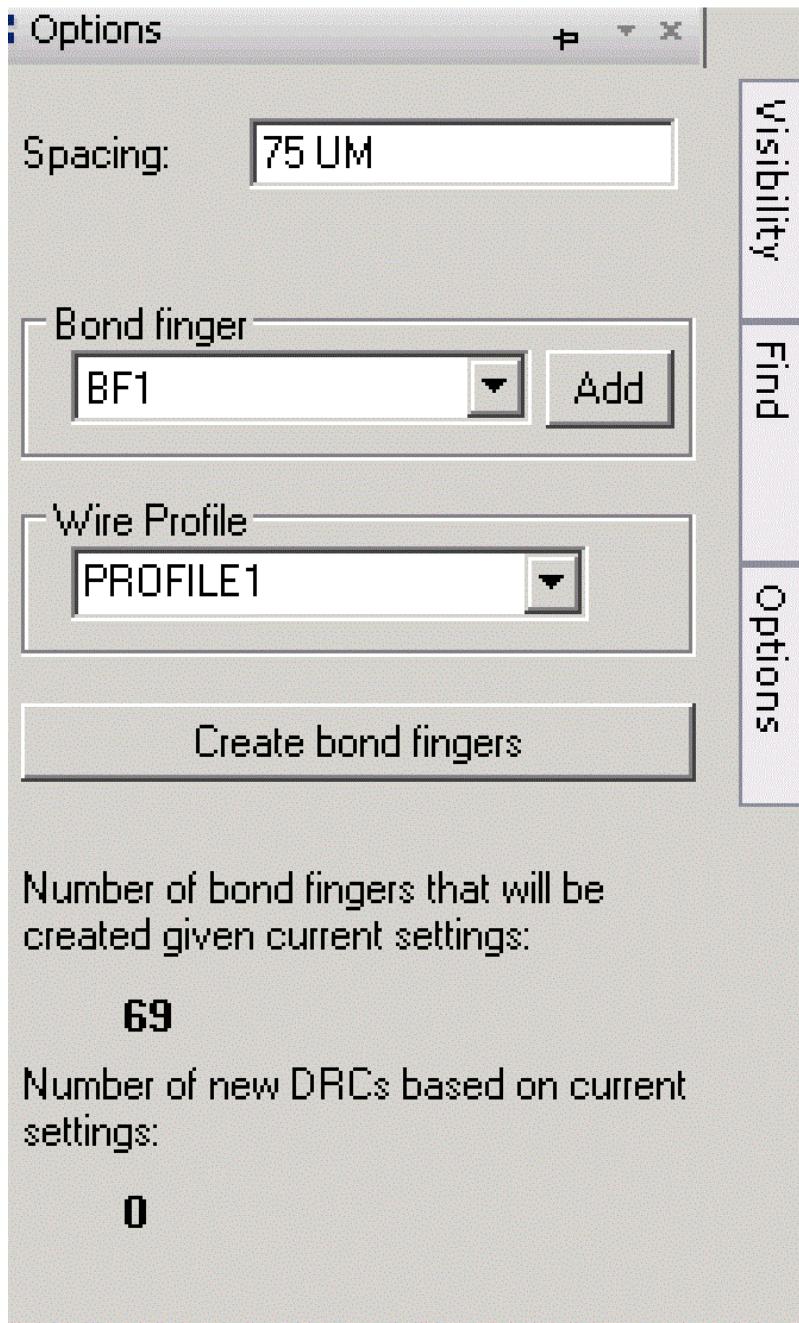
- [wirebond select](#)

Populating the Guide Path with Bond Fingers

During the planning stages of an IC package design, it is helpful to determine how many bond fingers can fit on a guide path under ideal situations. This lets you quickly make tradeoffs for spacing, narrow bond fingers, or multiple bond finger rows. Any decision that you can make early on saves design time later in the design process.

Before wire bonding the design, use the *Populate with Fingers* option, a feasibility tool used to select a guide path and populate it with as many unwired bond fingers as possible using a specified padstack and spacing. Because no bond wires are created by this command, the exact placement and spacing of bond fingers is based only on the supplied bond finger-to-bond finger spacing value. As a result, the tool places more fingers than can actually fit when you add the bond fingers and the tool is using other constraints.

Use the `wirebond manage guide paths` or `wirebond select` command to run this tool and complete the parameters shown in the *Options* window pane.



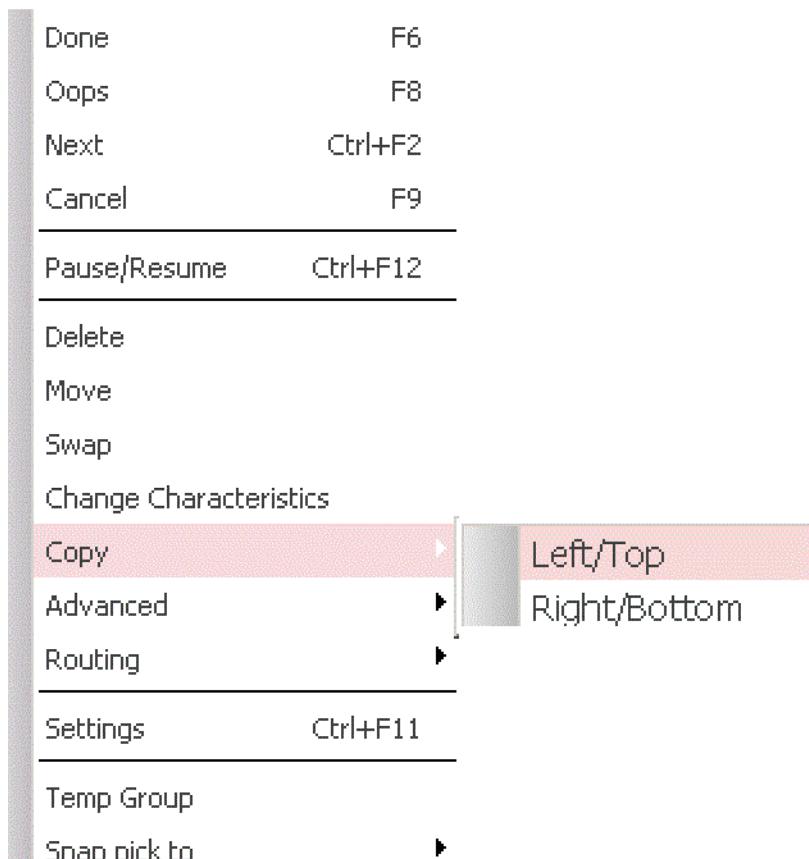
For additional information about using this tool, see the [wirebond select](#) command.

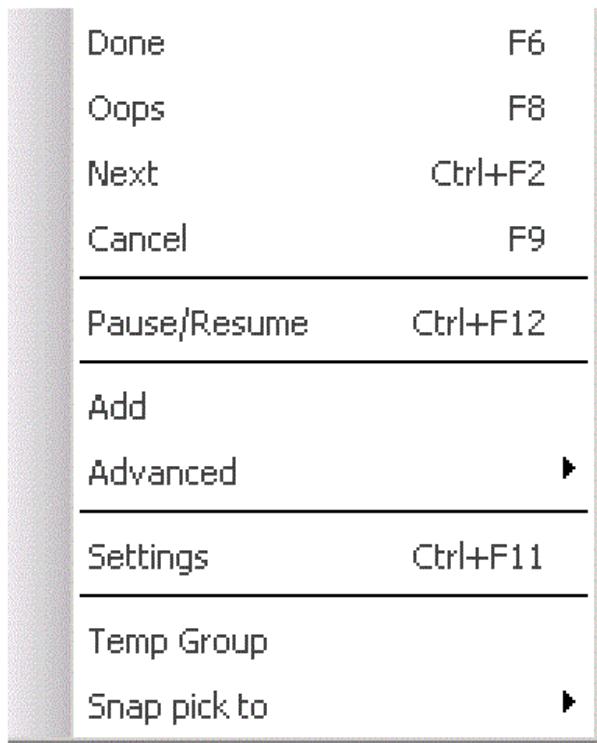
Non-Wired Bond Fingers

Once you have an existing bond finger pattern in place, you may want to insert one or more non-wired fingers within the existing pattern. For example, you may use these non-wired fingers as placeholders and later bond them to die variants or replacement dies; you may want to place them for anticipated new functionality at the time of manufacturing, or use them for different device bond-out options for feasibility studies and so on.

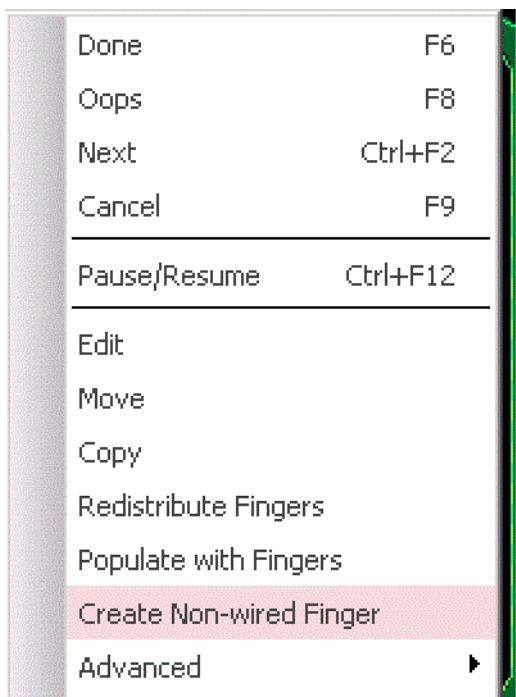
Using either the [wirebond select](#) or [wirebond add](#) commands, you can run the command using three different object types: bond finger, die pad, or guide path. Based on the object type that you select, The following shows the pop-up menus that appear when you run the command, select the object, and right-click.

Pop-Up Menus Based on Object Type





From Die Pads



From a Guide Path

For information on the command and procedures, see the `wirebond select` or `wirebond add` commands.

Non-Standard Bond Wires

Sometimes, you need to create non-standard bond wires or jumpers that connect two objects on the substrate. A non-standard bond wire connects any two substrate objects (bond finger, cline, and so on) on the same net together. A jumper is a specialized non-standard bond wire that connects two bond fingers. The bond wire may connect two substrate bond fingers to add length or delay to a net or it may connect directly to a filled shape in certain regions of a design.

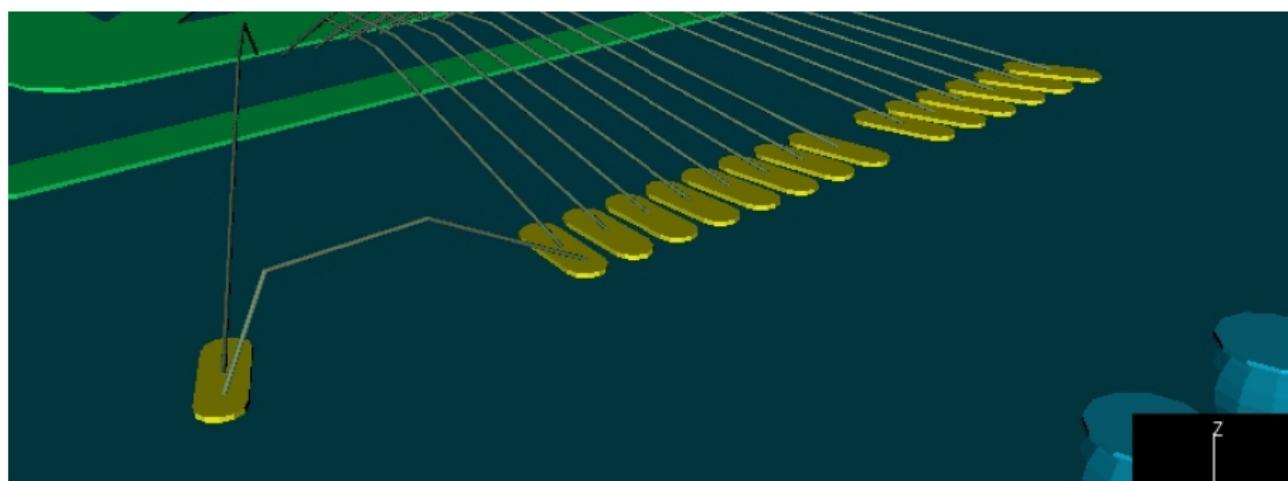
Using the *Non-standard Bond Wire* feature, you can connect any combination of two of the following types of objects:

- Pins (die pins, discrete pins, and so on)
- Bond fingers
- Filled conductor shapes
- Cline ends

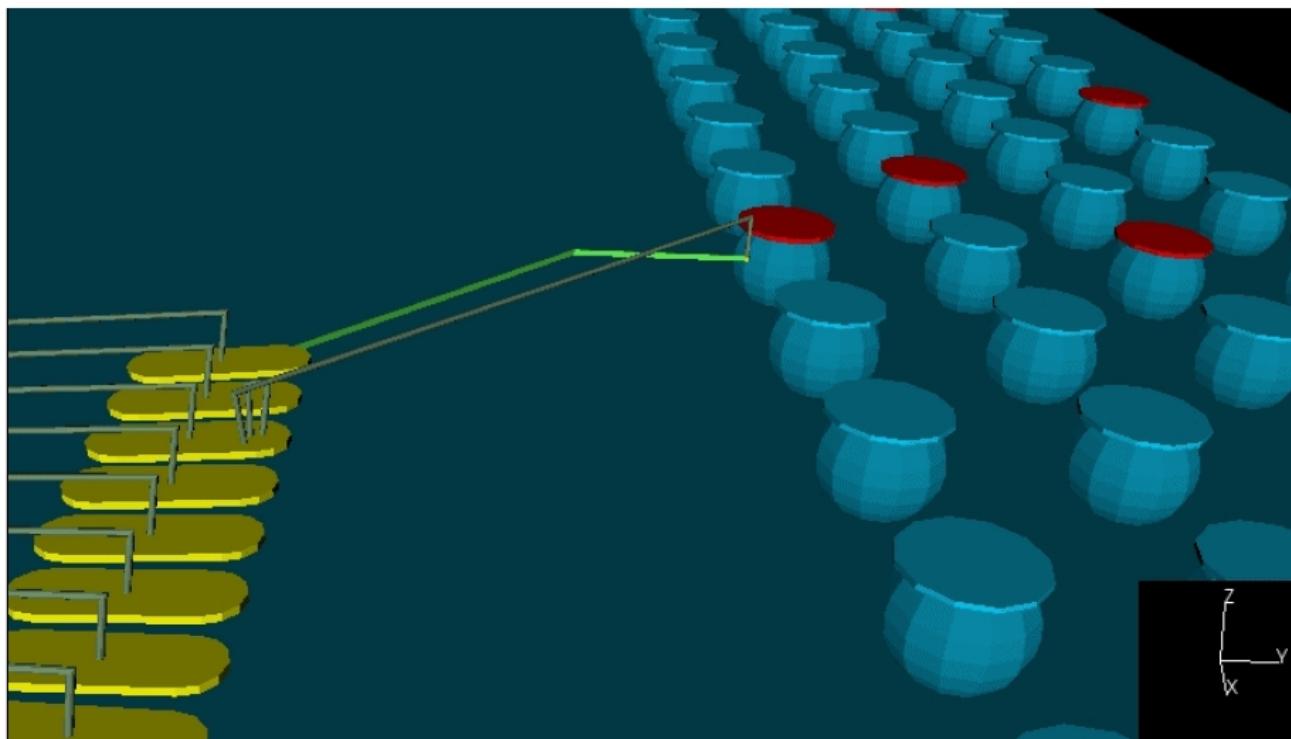
You cannot connect both ends of the bond wire to the *same* element in the design, for example, a bond wire that connects two points on the same filled shape on the same layer.

Both objects have to be on the same net. If one selected item is on a real net and the other is not on a net or is on a dummy net, the real net is automatically assigned to the other item in order to establish connectivity.

3D View of Non-Standard Bond Wire Connecting Two Bond Fingers



3D View of Non-Standard Bond Wire Connecting Bond Finger to Cline



How the Non-Standard Bond Wire Feature Works

To create a non-standard wire bond in a design, the source and destination objects must both be:

- Exposed for wire bonding
- On the same side of the substrate

You select the profile with which the bond wire is associated and then you perform two picks in the Design Window. The order in which you pick the end points is important. The first pick is the starting object for the bond wire. For *Forward* bonds, this is the start of the wire profile. For *Reverse* bonds, it is the end of the profile.

When editing an existing non-standard wire bond, your first pick selects the wire and the end point to be moved. The *Options window pane* updates to reflect the current profile for this wire.

Non-standard bond wires may not follow the standard rules for pushing and shoving, but they are still critical to the overall design correctness, in terms of signal integrity, delay, and 3D spacing of objects.

Online DRC checks wire-to-wire (same or different profile) and wire end-to-wire end on the same object.

When to Use the `wirebond add nonstandard` Command

You can use the `wirebond add nonstandard` command at any point in the design flow. However, the tool does not consider the non-standard wire bonds when pushing and shoving standard wire bonds through the wire bond command set. Ideally, you use this command after you create the standard wire bonds and when your pattern is as complete as possible. Additionally, it is recommended that you run the Cadence 3D Design Viewer before signing off on the design to verify that these wires have appropriate profiles and are not in conflict or cause shorts.

Wire Bond Tack Point

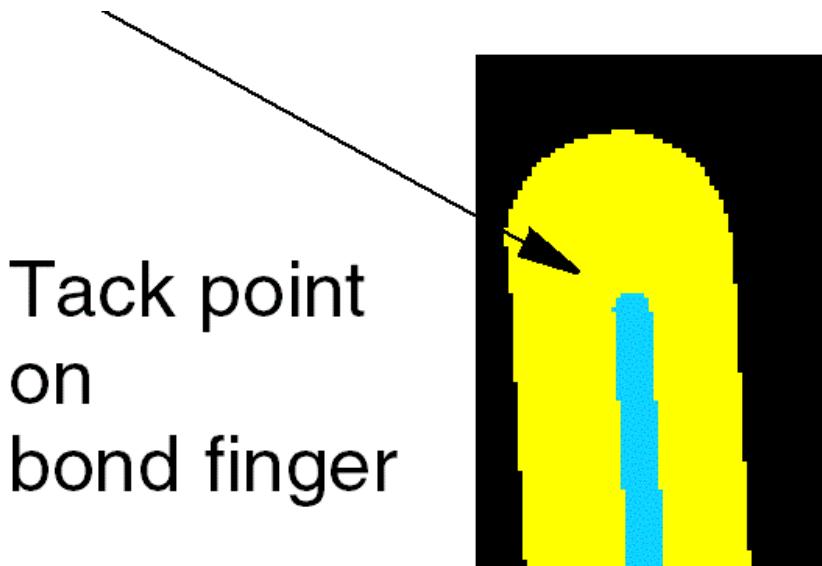
A wire bond assembly comprises three elements: a die pad or bond finger, a bond wire, and a second die pad or bond finger. In the database, the die pad is the pin of a die symbol, the bond wire is a two-point connect line that is part of the wire profile and the bond finger (a padstack-based object) has the BOND_PAD property attached. Typically, the two ends of a bond wire are connected to both the pad origin (connect point) at both the pin and the bond finger.

With the Wire Bond Tack Point feature, you can adjust the tack point, which is the contact point of the bond wire to the bond finger or the bond wire to the die pad. You can add wires that are not at the center of the pads, or even add multiple bond wires to the same die pin or bond finger if the exposed pad is large enough to accommodate multiple wires. You perform this task at the end of the design process to improve the process of manufacturing, more accurately model the actual design by separating multiple wires on the same pad, or adjusting the wire-to-bond finger clearance.

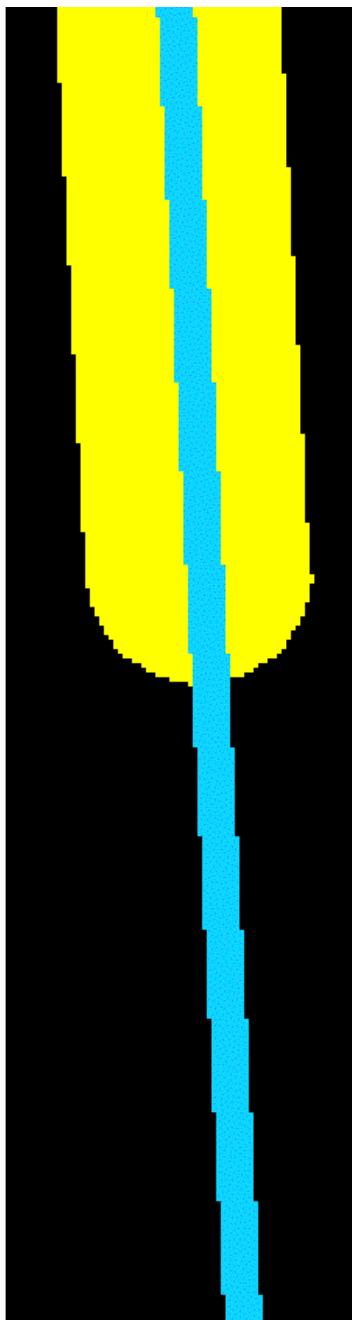
Because the tool, by default, does not let you add multiple wire bonds to a pin, set the `wirebond_multwire_pins` environment variable to override the setting and connect multiple wire bonds to a single pin. You can set this variable by choosing *Setup – Preferences* (`enved` command), then clicking the *IC_packaging* category and then *Wirebond*.

 If you change the padstack for a bond finger or pin and the adjusted wire end point lies outside the pad boundary, the tool disconnects the bond wire from the object.

Wire Bond Assembly with Tack Point



**Tack point
on
bond finger**

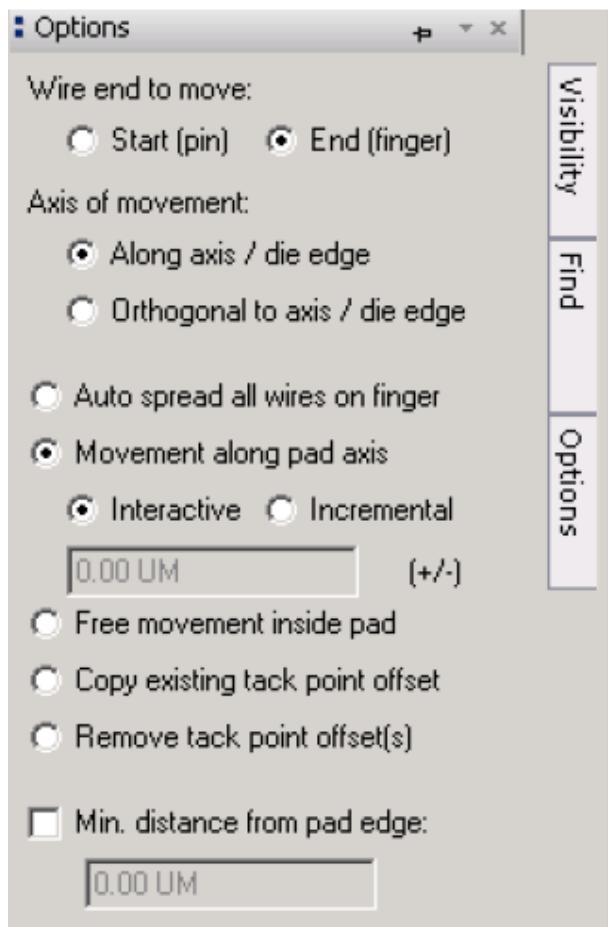


With the wire bond tack point feature, you can adjust the tack point without affecting the bond finger or die pin locations. You move the tack point along the center line of the bond finger or die pin, or freely anywhere on the pad, keeping the connection point at least a minimum distance from the edge of the finger or pin boundary to ensure that the bond wire remains in contact with the bond finger or pin.

Accessing the Wire Bond Tack Point Feature

To access the wire bond tack point feature, choose *Route – Wire Bond – Tack Point Move* ([wirebond tack point](#) command) from the menu. The *Options* window pane in the Control Panel appears as shown in the following figure.

Options Window Pane for the wirebond tack point Command



Routing Stubs

Routing stubs (referred to as *fanout stubs* in Release 16.2) are the first routing segments that extend from bond fingers. The wire bonding tool processes the routing stubs during push and shove to ensure that bond fingers meet minimum DRC clearance with nearby routing stubs.

With the enhancements in this release, the tool allows you more flexibility in your flow. Now, as you partially or completely route the design and change the wire bond pattern, the routing stubs automatically correct themselves to maintain spacing. By adding stubs when you add the fingers, you can see in real time where routing channels are too tight between bond fingers. You can then explore alternate solutions without having to move between the wire bond and routing tools.

Pushing and shoving of the pattern to maintain minimum clearance between the bond fingers and routing stubs of other clines allows you to easily achieve an ideally compressed pattern on each die side, thus improving efficiency.

You can add routing stubs using any one of these methods:

- During an add wire bond operation

Choose either the *Route – Wirebond – Add* ([wirebond add](#)) or *Route – Wirebond – Select* ([wirebond select](#)) menu command and choose *Pins* in the Find Filter.

- During a move wire bond operation

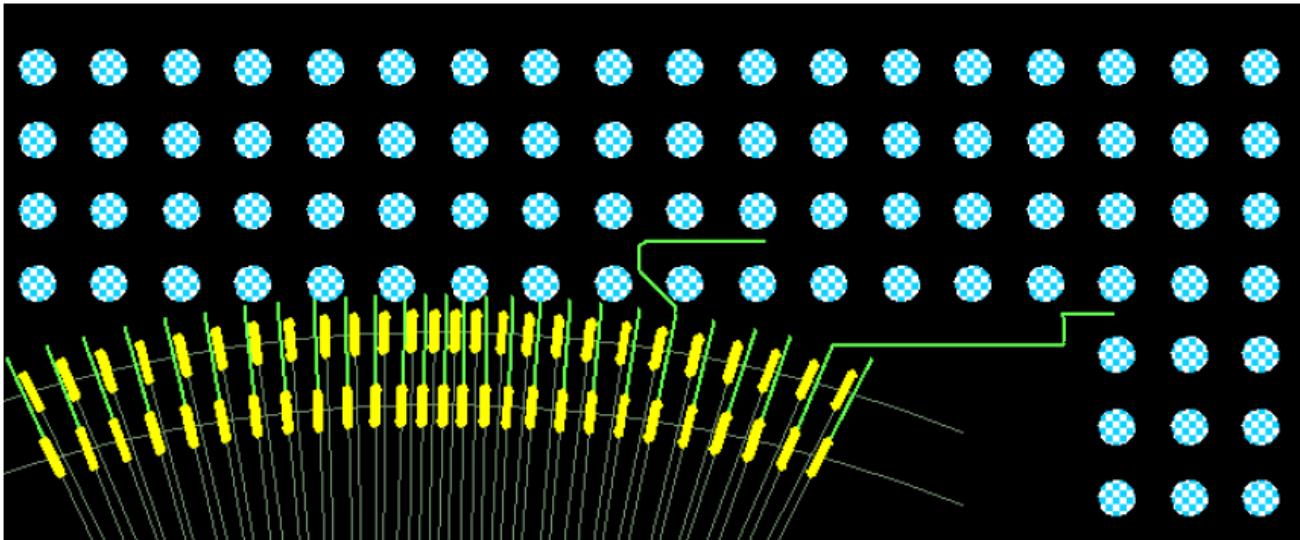
Choose either the *Route – Wirebond – Edit* ([wirebond edit command](#)) or *Route – Wirebond – Select* menu command; then choose either *Bond Wires* or *Fingers* in the Find Filter, and select those bond fingers that do not have stubs defined.

- During a stub modification

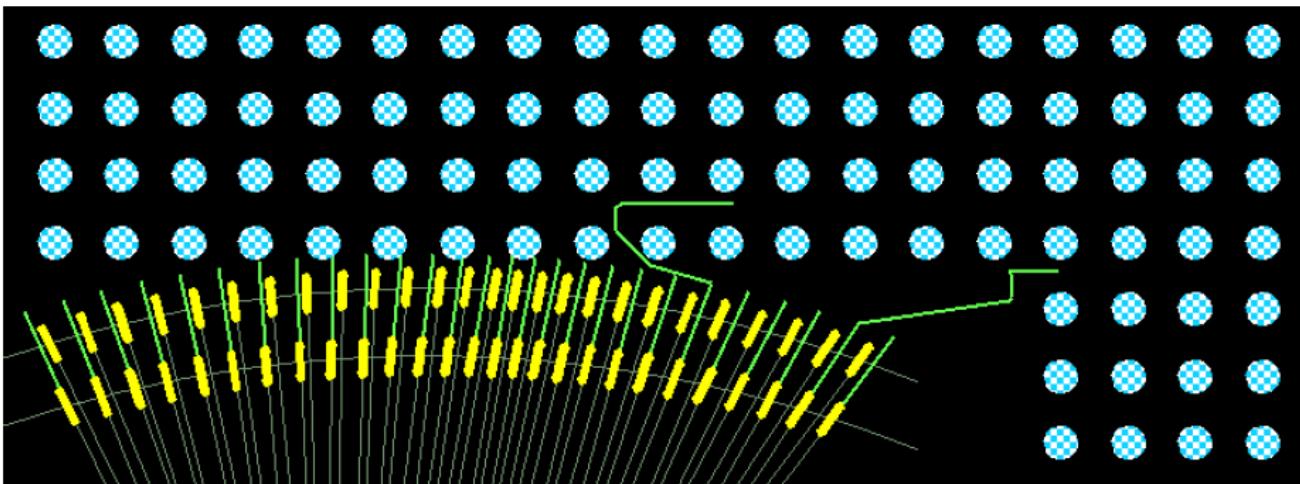
Choose either the *Route – Wirebond – Edit* or *Route – Wirebond – Select* menu command and choose *Fingers* in the Find Filter; then select the bond fingers in the Design Window, right-click, and choose *Routing – Edit Routing Stubs*.

You may also modify or delete the stubs using these commands at any point during the design process.

Wire Bonds with Routing Stubs and Clines



Pushing and Shoving



The tool does not create routing stubs for bond wires that go to the power and ground rings in your design. If a power or ground bond wire attaches to an I/O bond finger and not a ring, it receives the appropriate routing stub.

When you add or edit bond fingers, you can also choose to add routing stubs. You choose the length of the stub as well as the direction in which the stub should move. The wire bonding tool creates the routing stubs using the *Minimum Line Width* specified on the bond finger's layer. The first cline segment is the stub. If you change the stub's width or length, or delete it after it is created, the tool remembers the new value during future push and shove operations.

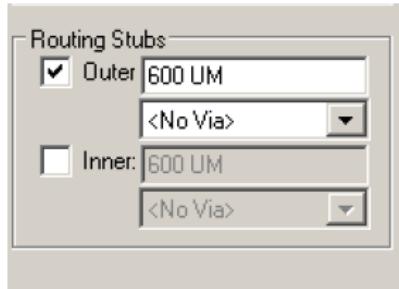
During the next wire bond operation, each stub extending from an existing bond finger is loaded individually based on its current state in the design, not on the values you originally specified when

you created the stub. So, if a stub is created independently of the wire bond system, it is still pushed, shoved, and spaced according to minimum DRC requirements.

The following figure shows the parameters in the *Options* window pane that you configure for this feature.

For additional information on descriptions in the *Options* window pane, see the [wirebond add](#) command.

Options Window Pane for the wirebond add Command



Highlighting Bond Fingers

The *Advanced Highlight* feature lets you temporarily or permanently highlight your bond fingers based on key attributes, such as padstack, net name, snap point, layout or placement style, and finger alignment. It makes it easier to find and isolate specified items for a given operation. You can perform this task after you create your wire bonds so that you can differentiate among objects during design selection, editing, and routing.

When using this feature, the visibility of highlighted colors in the Design Window is prioritized as follows:

- Temporary highlight
- Finger object
- Net assignment
- Symbol owner for the finger

For example, if you custom color a bond finger, its net, and the owner symbol, the bond finger highlight takes precedence.

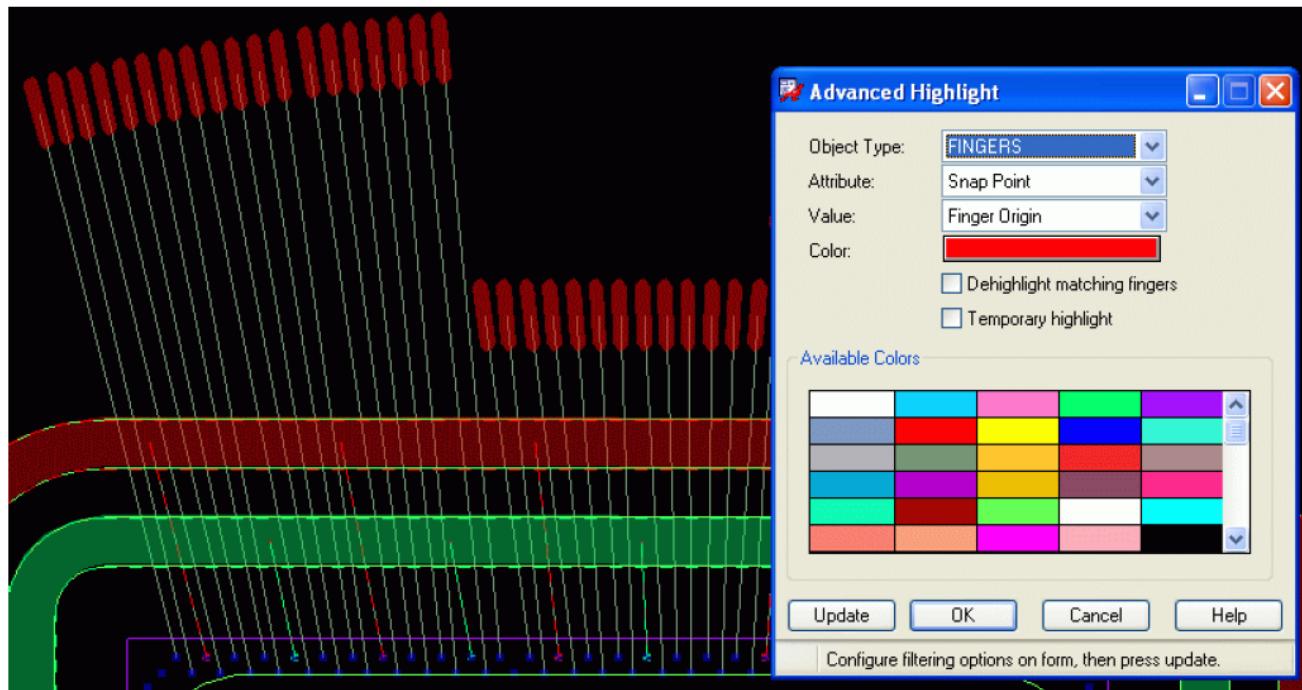
For information on using the command, see the [advanced highlight](#) command.

⚠ The `bond finger hilite` and `bond finger dehilite` commands have been removed from the tool, although the function is available in the new `advanced highlight` command using the temporary highlight option.

Example

The following figure shows bond fingers filtered in the Design Window by *Finger Origin* as a *Snap Point* attribute.

Highlighting the Snap Point Attribute



Creating and Modifying Wire Profiles

Bond wires are 3D entities that curve through space to connect a die pin to a package substrate finger or another die pin. The exact path that each wire in a design follows through space is unique. However, groups of similar wires use a similar path, called a wire profile. When wiring a die to the substrate, the wire bonding machine uses this profile information.

SPB tools now support the Wire Profile Editor for defining a wire profile, the model applied to bond wires. The Wire Profile Editor is sophisticated, but flexible enough to change the definition to a level of detail needed for a specified application.

Who Uses the Profile Editor?

Since different applications require different levels of accuracy for the 3D wire path, you need to tailor your wire profile definitions to meet the needs of your design tools. While the manufacturing group and tools may require a very detailed path, 3D analysis or optical checks may need only approximate information, as more detail provides little or no additional value.

Typically, a company may assign an individual or group to establish wire profile definitions before the design process begins, and store them in an XML-based Primary Definitions file. This ensures that specified profiles are available for the wire bonding process. Also, package companies might provide wire profile definitions to use for specific types of designs.

After establishing a wire bond pattern, you can implement your own definition for each wire profile name and load your own values when operating on the design. You can also use the Wire Profile Editor to adjust the wire bond profiles based on the Cadence 3D Design Viewer results, save them, and use the revised profiles when you launch the Cadence 3D Design Viewer again.

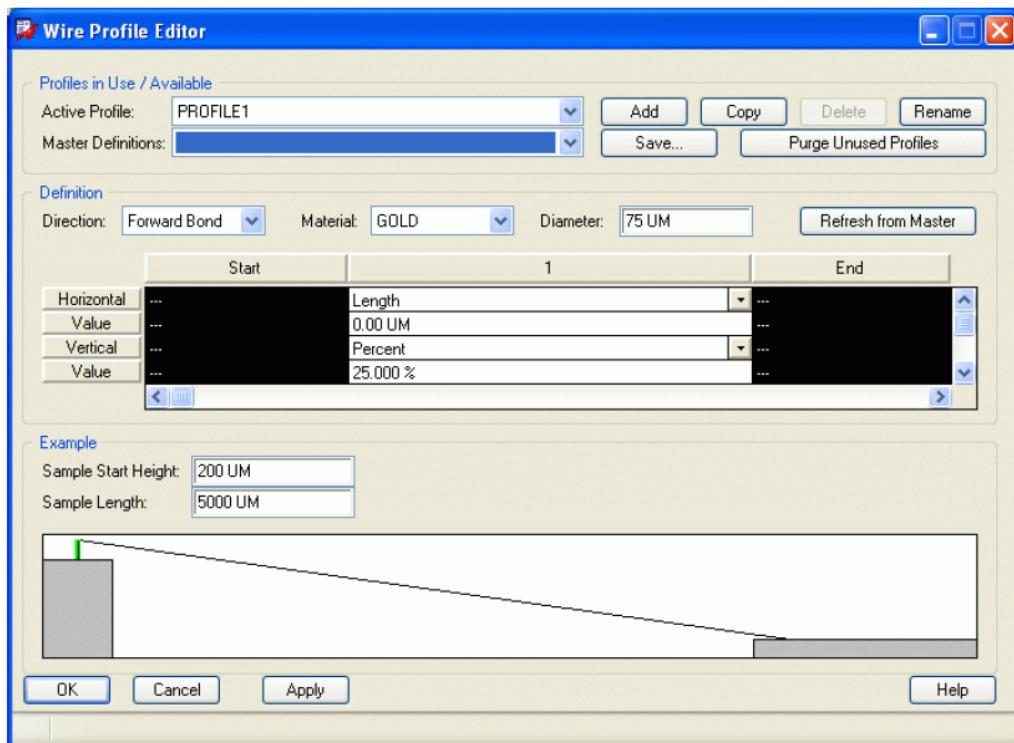
In addition, Cadence has been working with wire bonding machine manufacturers to supply default wire profiles known to be compatible with the manufacturers' wire bonding machines. This release provides the first set of profiles, located in the TECHPATH directories as configured for the tool.

Accessing the Wire Profile Editor

Choose *Route – Wire Bond – Settings* to access the Wire Profile Editor. In the Wire Bond Settings dialog box, click *View/Edit wire profiles*. The following figure shows the Wire Profile Editor.

For additional information, see the `wirebond` settings command.

Wire Profile Editor with Default Profile



How the Wire Profile Editor Works

The Wire Profile Editor displays one wire profile—the *Active Profile*—at a time. You can select the active profile from a list of those currently in the design or in the *Master Definitions* file. Once you select a profile, the screen refreshes to show data in that profile. You can also add and copy profiles, rename profiles or delete profiles that are not being used in the design.

In addition, the *Master Definitions* field lists available sets of pre-defined profile definitions as located in the TECHPATH variable. These are sets of profiles provided by your company, CAD team, or wire bond machine manufactures as "known good" profiles. Cadence recommends that you select a master definitions file early in your design process, to ensure that you have the desired profiles available when you begin the wire bond design process.

 If you rename a vendor-certified profile, it will be marked as uncertified, as the vendor will be unable to locate the profile in their list of definitions.

The Wire Profile Editor lets you create the wire profile by defining a series of steps that specify how the tool should advance from one point on the wire to the next. Each step consists of a horizontal and a vertical component. These components can be any of the following:

- Length – Specifies a fixed value.
- Percent – Provides the same movement but as a percentage of the wire's length. This value may differ for each wire that uses this profile as it is possible for multiple wires (*Orthogonal* setting) to have the same value.
- The tool computes a percentage of the sample height of the die, which is the height of the die pad in relation to the bond finger. For example, a die (250 UM thick) is placed on the substrate. The bond finger (0 UM) is also on the substrate. The difference in height is 250 UM. A vertical move of 50% results in 125 UM.
- Angle – Specifies a change in the angle of the given amount. Positive angles move up and away from the substrate; negative angles move down toward the substrate. Angles are measured from the horizontal plane.
- Switch – Indicates that the tool should move to the end of the wire to take the next point. This means that the remaining steps start from the other end of the wire. The tool starts from one end, then switches to the other end and proceeds to the segment where it left off. You can use only one *Switch* option per wire profile.

Wire profiles have an associated direction that you can change. *Forward* implies that the wire runs

(starts at X and finishes at Y) from the die pad to the bond finger; *Reverse* means that the wire runs from the bond finger to the die pad. For die-to-die wire bonds, the forward direction means that you are moving from the higher pad to the lower pad. The direction makes a difference when low clearance restrictions exist, for example, in a die stack, as the height of the perpendicular start bond is higher than the horizontal bond where the wire terminates.

As you add each step, you can view a graphical representation of the profile that you are creating.

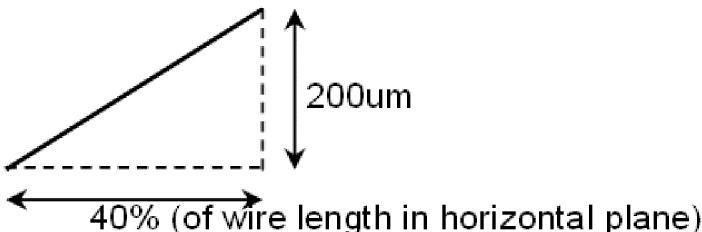
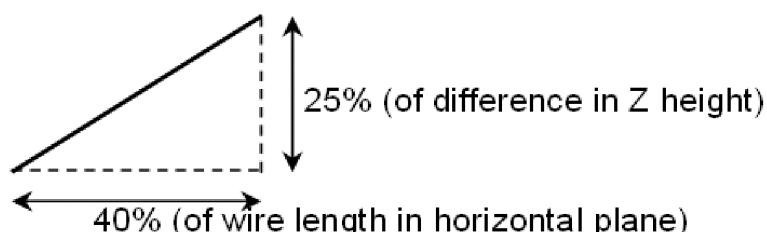
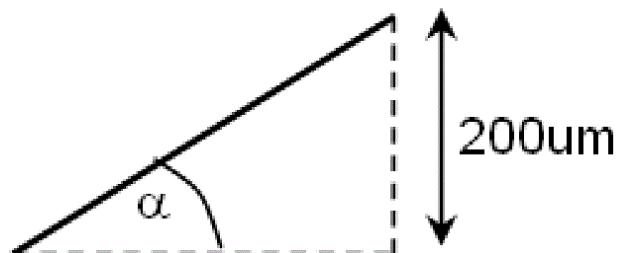
The following table shows the options for horizontal and vertical components.

Options for Horizontal and Vertical Components

Horizontal:	Length	
Vertical:	Length	
Horizontal:	Length	
Vertical:	Percent	

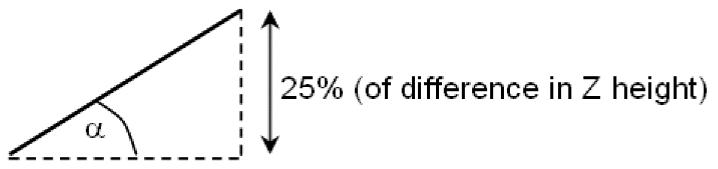
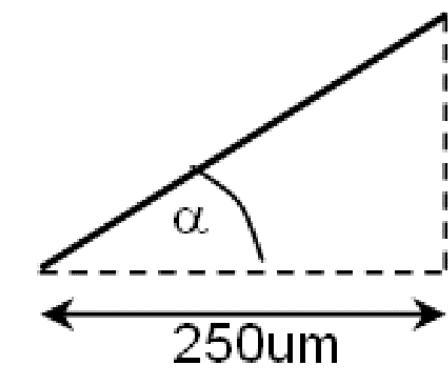
Routing the Design

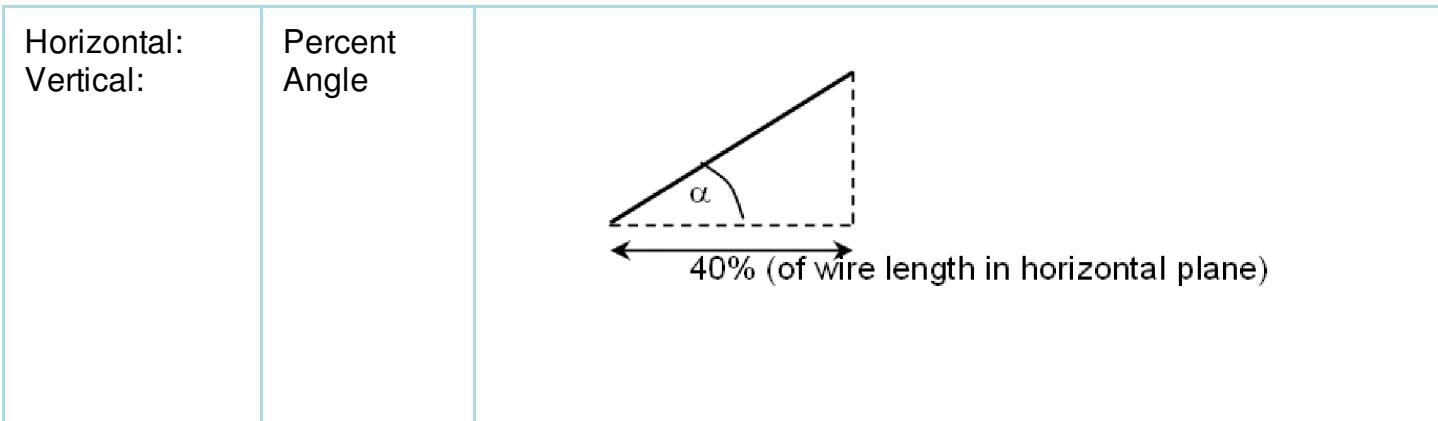
APD: Introduction to the Wire Bonding Toolset--How the Wire Profile Editor Works

Horizontal: Vertical:	Percent Length	 <p>A diagram showing a wire profile starting at a point and rising linearly to another point. A vertical dashed line segment connects the start and end points. A horizontal dashed line segment extends from the start point to the right. A double-headed arrow between these two dashed segments is labeled "200um". Below the horizontal dashed line is a double-headed arrow labeled "40% (of wire length in horizontal plane)".</p>
Horizontal: Vertical:	Percent Percent	 <p>A diagram showing a wire profile starting at a point and rising linearly to another point. A vertical dashed line segment connects the start and end points. A horizontal dashed line segment extends from the start point to the right. A double-headed arrow between these two dashed segments is labeled "25% (of difference in Z height)". Below the horizontal dashed line is a double-headed arrow labeled "40% (of wire length in horizontal plane)".</p>
Horizontal: Vertical:	Angle Length	 <p>A diagram showing a wire profile starting at a point and rising linearly to another point. A vertical dashed line segment connects the start and end points. A horizontal dashed line segment extends from the start point to the right. An angle α is shown between the wire profile and the horizontal dashed line. A double-headed arrow between the vertical dashed line and the horizontal dashed line is labeled "200um".</p>

Routing the Design

APD: Introduction to the Wire Bonding Toolset--How the Wire Profile Editor Works

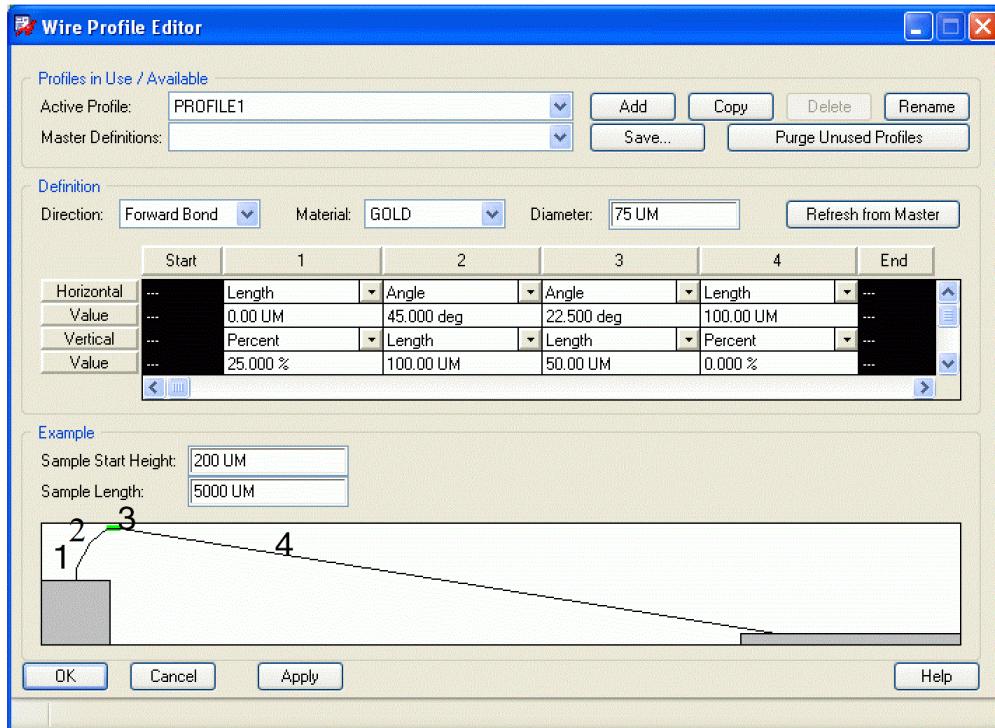
Horizontal: Vertical:	Angle Percent	
Horizontal: Vertical:	Length Angle	



Wire Profile Example

The following figure shows a completed profile. The steps to complete this profile are described below. The numbered steps appear only in this example, not in the software.

Wire Profile Editor



- Step 1

The first point has a horizontal length component of 0 UM and a vertical percentage component of 25%.

Because the horizontal length is 0, it means that the wire leaving the pin goes straight up, one-fourth the height of the die. The tool computes 25% of the *Sample Start Height* of the die (200 UM), which is the height of the die pad in relation to the bond finger. The result is 50 UM.

 You can change the default value for *Sample Start Height* and *Sample Length* in the Wirebond section of the IC Packaging category in the User Preferences dialog box.

- Step 2

The second point specifies a horizontal angle of 45 degrees and a vertical length of 100 UM. The horizontal angle means that the wire moves on a 45-degree angle up (away from the substrate) while at the same time it moves up 100 UM higher, illustrated by the second segment in the figure.

- Step 3

This point specifies a horizontal angle of 22.5 degrees and a vertical length of 50 UM. This means that the wire moves on an angle of 22.5 degrees to the right while at the same time moving 500 UM higher, illustrated by the third segment in the figure.

- Step 4

This point specifies a horizontal length of 100 UM and a vertical length of 0 UM. This means that the wire moves only to the right 100 UM, and does not go up any higher. The tool calculates this final segment based on the specifications and connects this last point to the bond finger.

The tool determines the remaining segment by how far away the bond finger is located on the substrate. This means that two bond wires with the same point definition, would actually have a different profile if the distance to the bond finger were different. The example graphics would look the same as the same profile definition.

XML Description for Example

The following shows the profile for the example above.

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE wire_profile_set SYSTEM "cdnwb.dtd">
<wire_profile_set>
  <wire_profile_def>
```

```
<profile_name>Profile 1</profile_name>
<profile_direction>FORWARD</profile_direction>
<profile_diameter>25.4 UM</profile_diameter>
<profile_material>GOLD</profile_material>
<profile_point>
  <horizontal_type>Length</horizontal_type>
  <horizontal_value>0 UM</horizontal_value>
  <vertical_type>Percent</vertical_type>
  <vertical_value>25.000 %</vertical_value>
</profile_point>
<profile_point>
  <horizontal_type>Angle</horizontal_type>
  <horizontal_value>45.000 deg</horizontal_value>
  <vertical_type>Length</vertical_type>
  <vertical_value>100 UM</vertical_value>
</profile_point>
<profile_point>
  <horizontal_type>Angle</horizontal_type>
  <horizontal_value>22.500 deg</horizontal_value>
  <vertical_type>Length</vertical_type>
  <vertical_value>50 UM</vertical_value>
</profile_point>
<profile_point>
  <horizontal_type>Length</horizontal_type>
  <horizontal_value>100 UM</horizontal_value>
  <vertical_type>Length</vertical_type>
  <vertical_value>0 UM</vertical_value>
</profile_point>
```

```
</wire_profile_def>
```

```
</wire_profile_set>
```

Wire Bond Via Estimation

After you create bond fingers in a wire bond package, you would like to know if there is enough room for vias before you begin breakout routing. The Wire Bond Via Estimator helps you determine if there is enough room, or if you should move the bond fingers to accommodate vias, thus eliminating re-work later.

How the Wire Bond Via Estimator Works

Select a die-stack, a via pad size, and line width and spacing values. The estimator then creates simulated vias, called via markers, which it places as densely as possible between the die and the outer row of bond fingers. A via marker is a plus sign (+) circumscribed by a circle. After placing all the via markers, the estimator evaluates the bond fingers to determine which ones require vias. Then it generates a report.

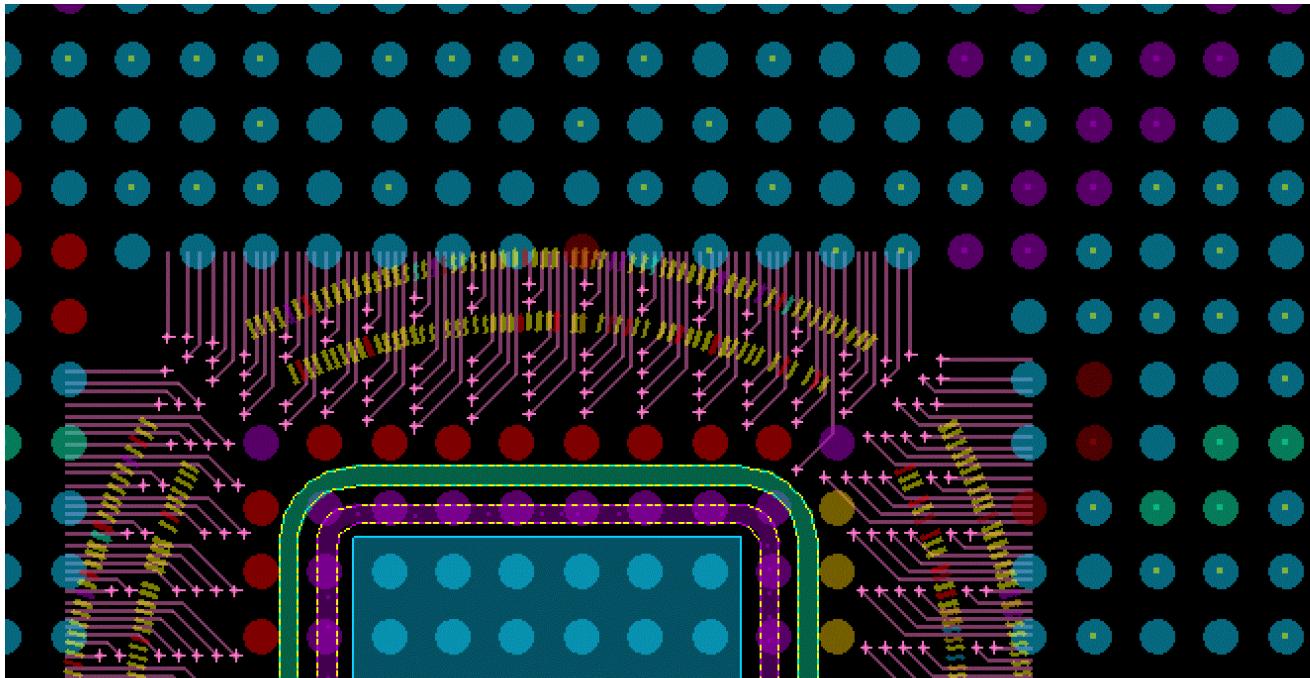
Placement of Via Markers

The estimator places via markers on the `Via_Estimation_Top` subclass on the `Substrate_Geometry` class, which it creates if the subclass does not already exist. It also displays the escape lines if you choose to display them. On the north and south sides of the die, the estimator places via markers in columns for every location that can accommodate a via. On the east and west sides, it places via markers in rows.

The estimator places via markers as close to the die outline as you specify, thereby adding as many via markers as possible (moving away from the die stack), and between tiers if possible, but it does not go beyond the outermost tier of bond fingers.

Once a column or row is full, the estimator creates another column or row. It determines spacing between columns or rows by the number of via markers in the previous column or row, along with other specified parameters. To maximize the number of via markers added, the starting location for the rows and columns is as far to one corner as possible.

Design with Via Markers



The estimator accommodates any number of bond finger tiers. It packs via markers as tightly as possible, allowing enough channel space for the anticipated routing wires on the escape layer to route through. When more than one layer is used for escape wires, the estimator assumes that the wires are distributed evenly across all escape layers. Adding more escape layers typically causes the via markers to be created closer to each other.

The estimator views via keepouts as obstacles. You can add via keepouts to the package in areas where you do not want the tool to place via markers, for example, in the corners of the package or between the rows of fingers.

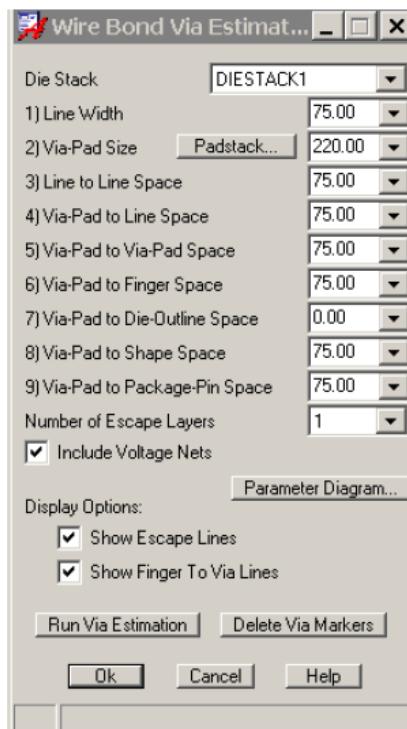
Bond Finger Evaluation

The estimator searches the bond fingers to determine how many of them require vias. For example, if a cline extending out from the bond finger will bump into another bond finger, the estimator determines that this cline requires a via. Using this method, the outer tier of bond fingers does not require vias and also some of the inner bond fingers may not require vias.

Running the Wire Bond Via Estimator

Use the [wire bond via estimation](#) command to run the Wire Bond Via Estimator. The following figure shows the parameters for this feature.

Wire Bond Via Estimation Dialog Box



Report

The estimator generates a report, summarizing the number of via markers created and the number of bond fingers needed. The *Status* column in the report displays *OK* if the number of via markers created is greater than or equal to the bond finger count, otherwise, *Not OK*.

This sample report shows that there is not have enough room for bond fingers that require vias on the East side of the die.

Wire Bond Via Estimation Report

Die Stack: U1

Via			
Die	Markers	Bond Fingers	
Side	Created	that require vias	status
North	124	89	ok
South	120	85	ok
East	81	116	not ok
West	112	77	ok

For information on using the estimator, see the [wire bond via estimation command](#).

Wire Bond Die Escape Generator

Escaping from a wire bond die involves time-consuming steps, such as via and escape cline creation and manual adjustment of the vias and clines. The Wire Bond Escape Generator lets you:

1. Automatically create escape clines, starting at the bond fingers and ending at the user-specified escape perimeter around the die.
2. Add vias to bond fingers for wire bond escape from another layer.
Individual vias dynamically track the cursor, with automatic routing between the bond finger and the via, and also from the via to the escape perimeter. With minimal key-clicks, you decide where the tool places individual vias.

How the Wire Bond Die Escape Generator Works

The Wire Bond Die Escape Generator starts the escape at a bond finger and automatically creates the first segment at the angle of the bond finger. Then it creates the other escape segments as horizontal, vertical, or at 45-degree angles until the wire reaches the escape perimeter. It determines the physical design rules at the escape start location and adheres to the rules, such as line width and spacing values defined as constraints. If an escape cline crosses a constraint area, however, the escape cline may not adhere to the rules within that constraint area.

If the generator cannot fully create the escape clines, it displays them as clines that do not fully reach the escape perimeter. You can add vias and then the tool generates the escape clines.

The Wire Bond Die Escape Generator also operates as follows:

- Although, you can add vias to the bond fingers on nets with the NO_ROUTE property, the tool does not generate the escape clines from the via to the escape perimeter.
- Because you cannot select vias or bond fingers with the FIXED property, the tool cannot modify them.
- You can add vias only to bond fingers that have no escape clines (excluding bond wires) attached.
- The generator does not push or shove pre-existing clines and vias.

Related Topics

- [wire bond escape](#)

Advanced Wire Bonding

This section describes some design challenges and solutions for advanced users of the Wire Bond tool set. Included are these topics:

- [Wire Bonding Multiple Stacked Dies](#)
- [Pushing and Shoving Same Net Bond Fingers](#)
- [Correctly Staggering Die Pins to Avoid False DRC Violations](#)
- [Wires Bond from Same Location Die Pins out to Same or Different Bond Fingers](#)

- [Wire Bonding: Using the dxf in and dxf out Commands](#)

Wire Bonding Multiple Stacked Dies

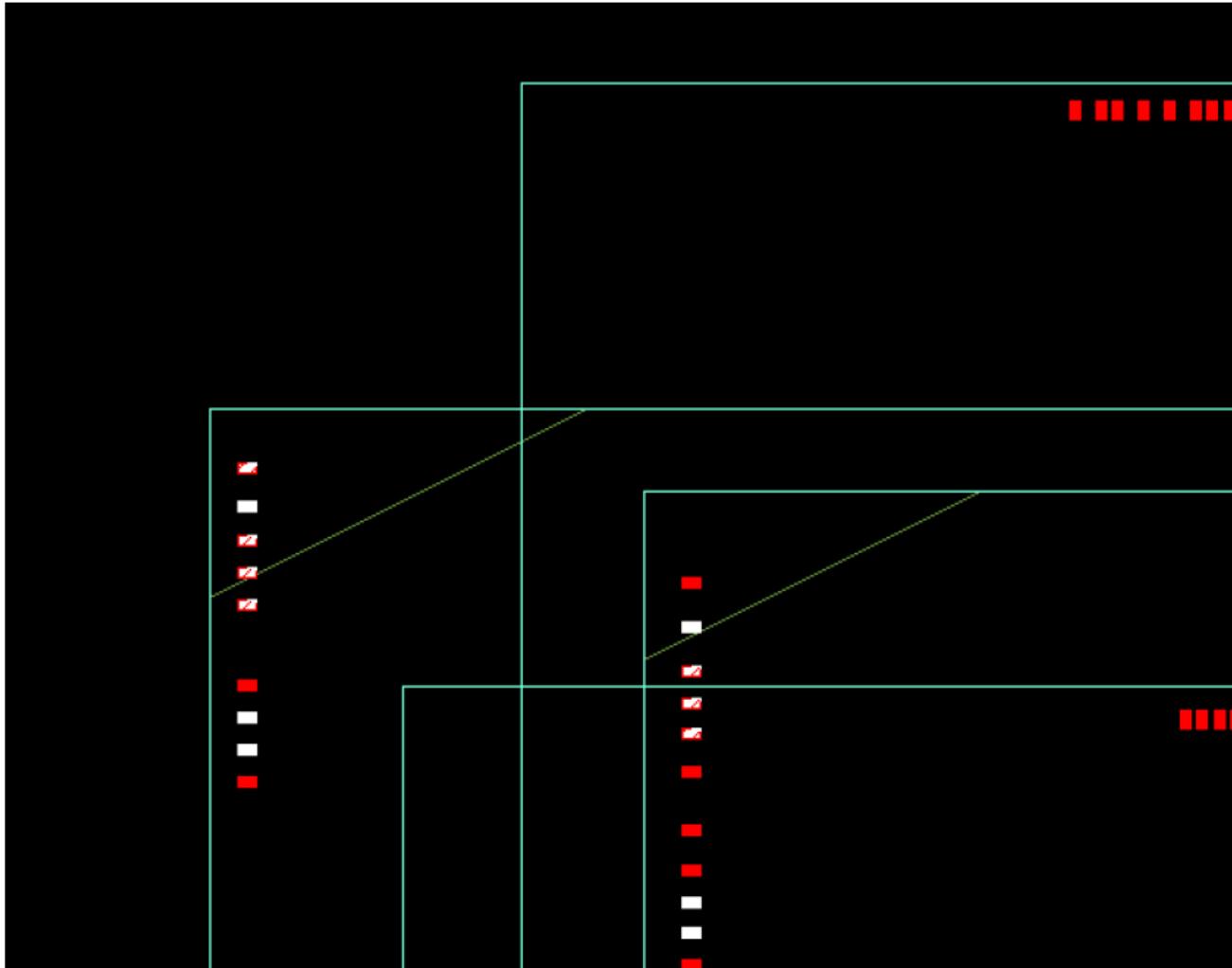
When wire bonding multiple stacked dies, you can encounter situations where the following topics apply:

- [Bonding Multiple Pins to Common Bond Fingers](#)
- [Bonding a Pin to Multiple Bond Fingers](#)
- [Splitting Multi-Wired Bond Fingers into Multiple Single-Wired Bond Fingers](#)

Bonding Multiple Pins to Common Bond Fingers

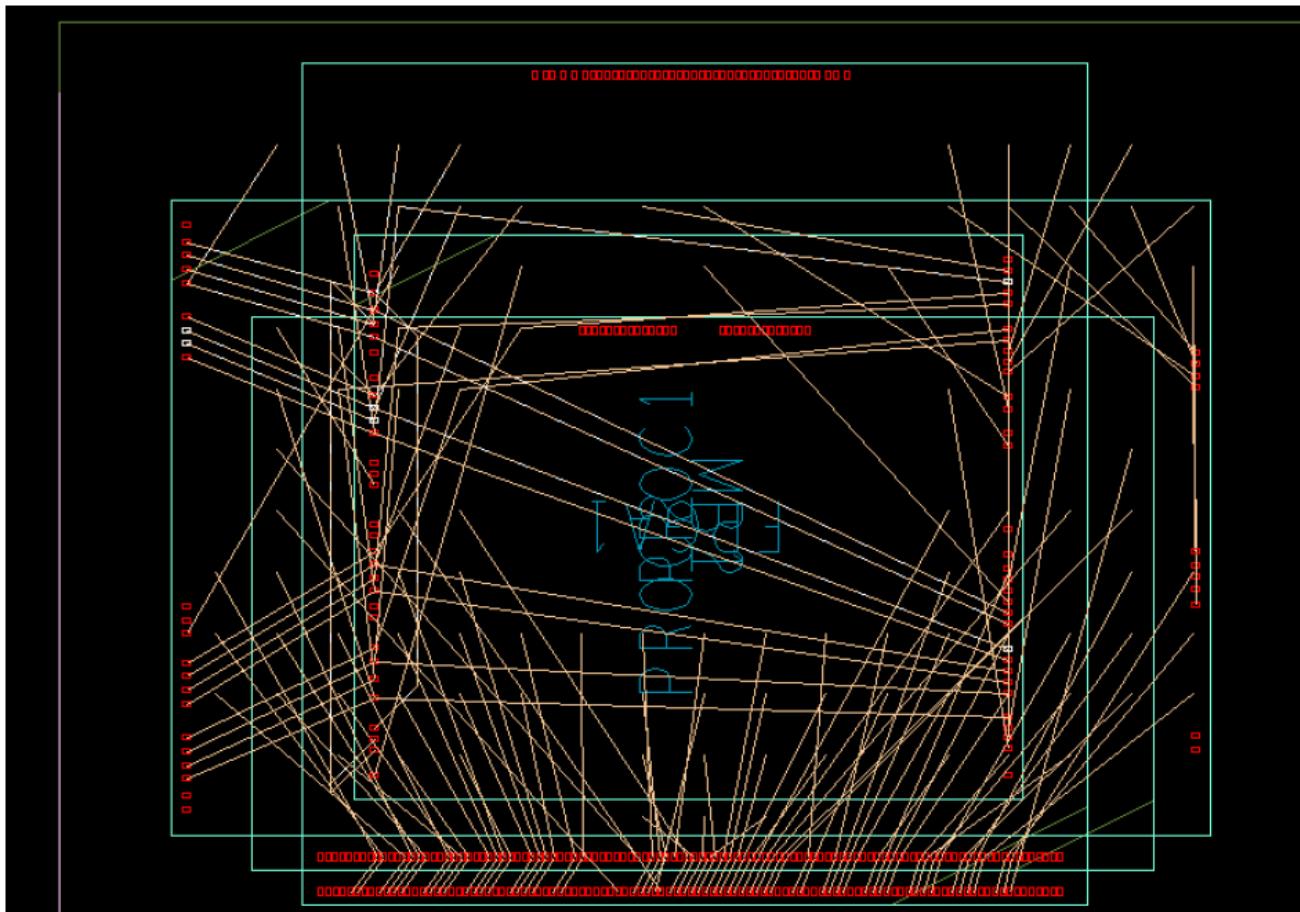
To describe this procedure, a design example containing multiple stacked dies is used (following figure).

Multiple Stacked-Die Design



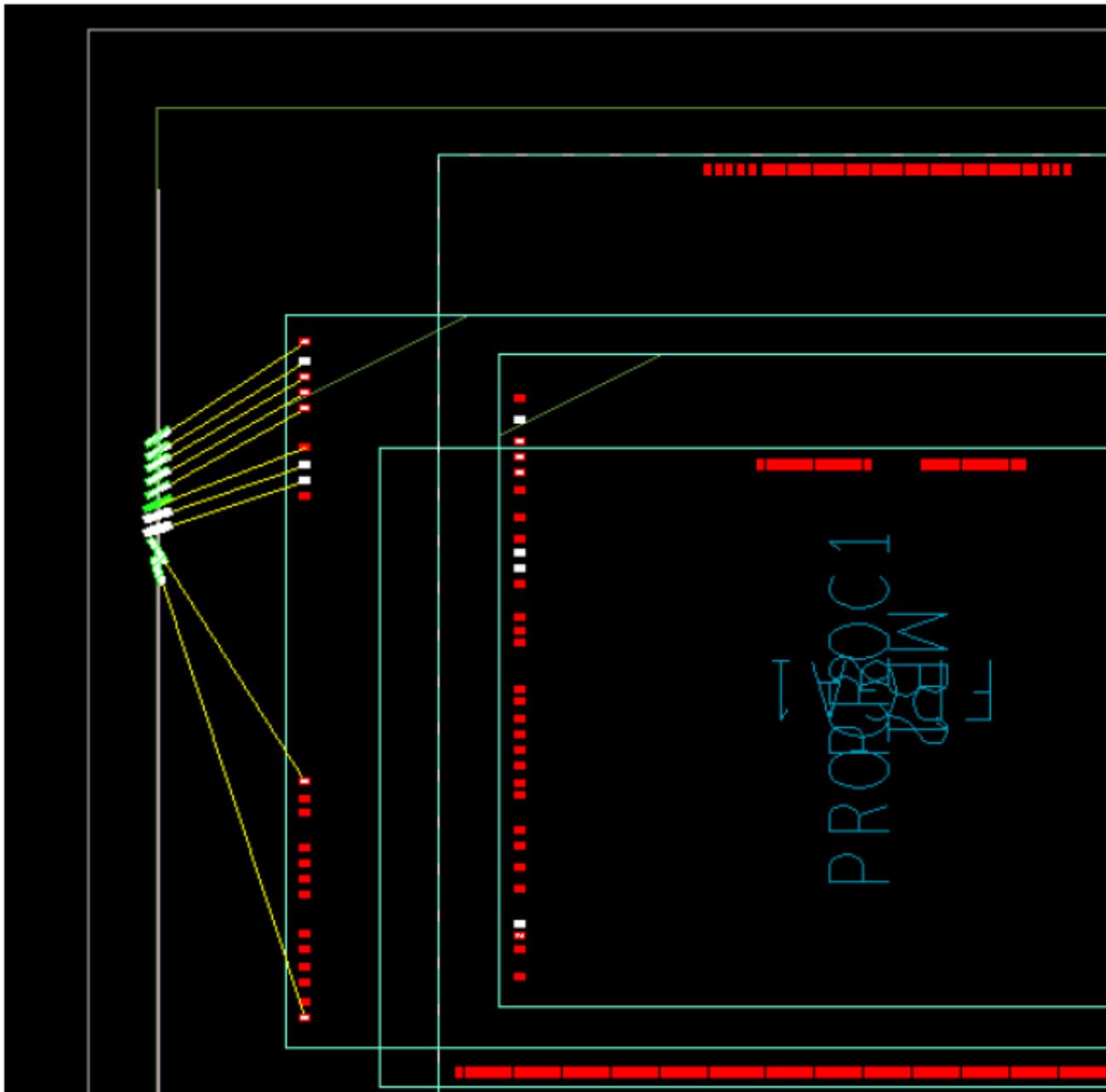
Several groups of pins from two of the memory die share common nets (see the following figure). You may want to bond these pins to the same bond finger, or merge these bond fingers with a shape.

Two Dies Share Common Nets



1. Choose *Wirebond – Select* (`wirebond select` command).
2. Choose *Pins* in the Find Filter.
3. Right-click and choose *Temp Group* from the pop-up menu. Click on the highlighted pins on the outer die, then right-click, and choose *Complete* from the pop-up menu.
4. Right-click again, and choose *Add* from the pop-up menu.
5. Modify parameters in the *Options* window pane if necessary.
6. Click to place the bond fingers.

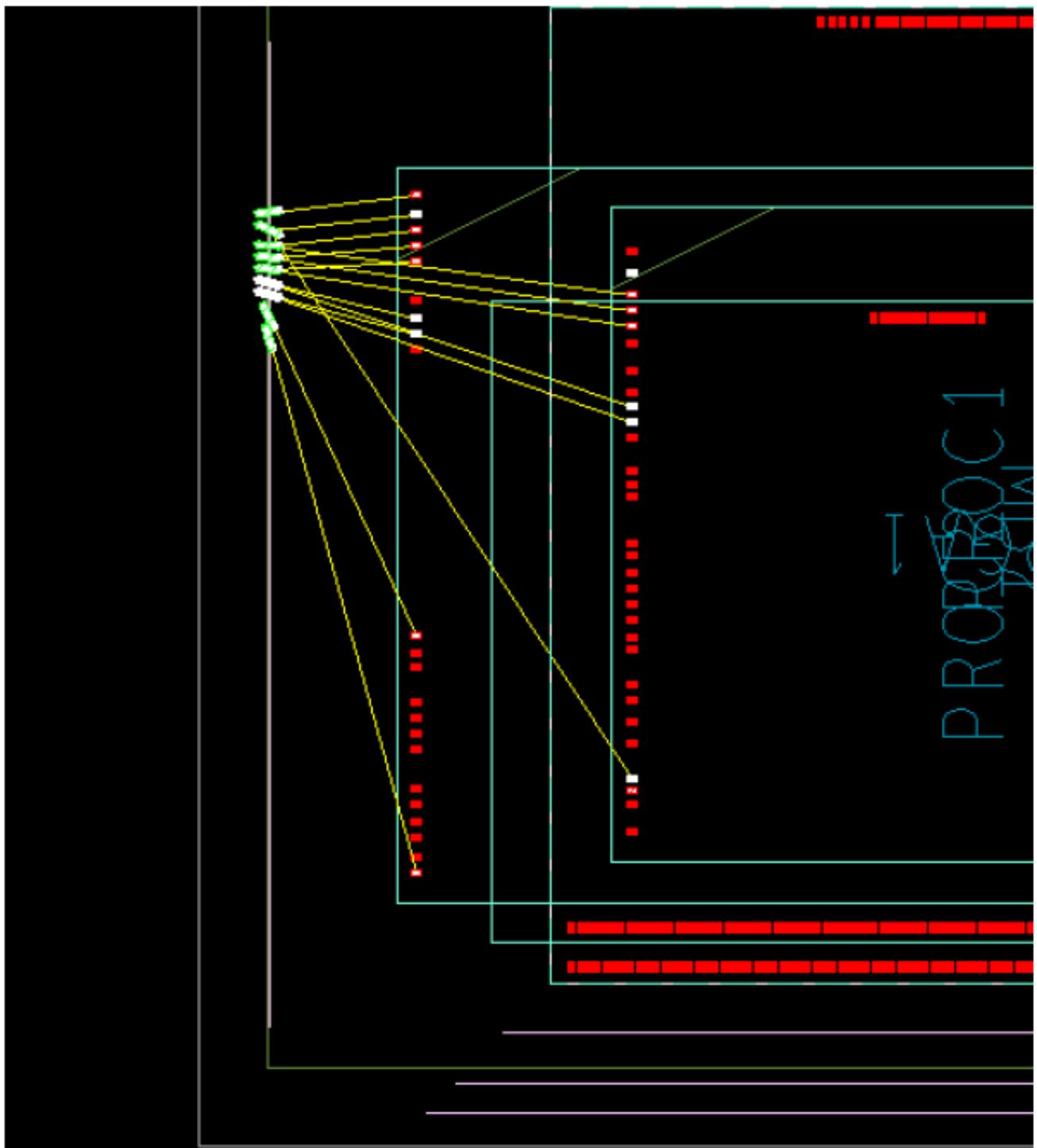
Bond Fingers on the Outer Die



7. Right-click and choose *Temp Group* from the pop-up menu. Click on the highlighted pins on the inner die, right-click, and choose *Complete* from the pop-up menu.
8. Right-click again, and choose *Add* from the pop-up menu.
When the *Options* window pane appears, uncheck the *Fingers* box. Then check the *Allow connection to wired fingers* box in the abbreviated *Options* window pane that appears.

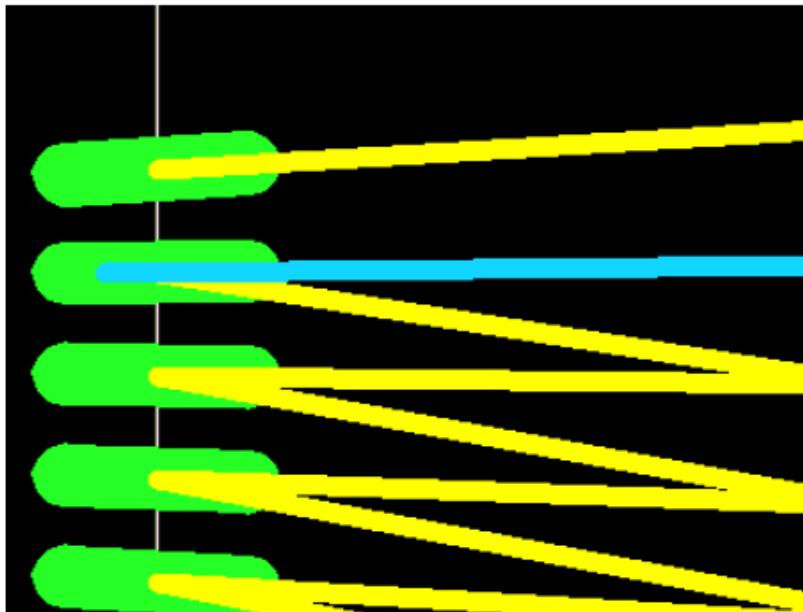
9. Place the bond wires on the existing bond fingers to which they should be connected and click.
10. Push and shove to better position the wire bonds.

Pushing and Shoving Bond Fingers



11. Adjust the tack point, which is the contact point of the bond wire to the bond finger or the bond wire to the die pad.

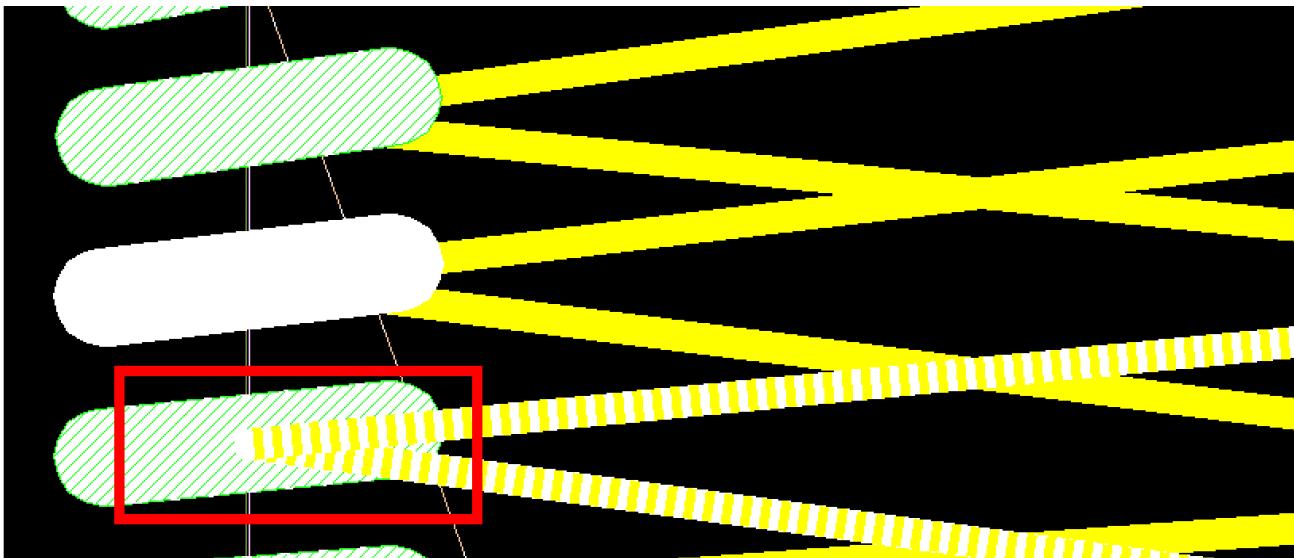
Using the tack point move Command



Splitting Multi-Wired Bond Fingers into Multiple Single-Wired Bond Fingers

1. Choose *Route – Wire Bond – Select* (`wirebond select` command).
2. Choose *Bond wires* in the Find Filter.
3. In the design where you have a multiple-wired bond finger, select all but one wire connecting to the bond finger, right-click, and choose *Delete* from the pop-up menu.

Splitting a Multi-Wired Bond Finger



4. Choose *Pins* in the Find Filter and add wire bonds to the unwired pins. Select the pins, right-click, and choose *Add* from the pop-up menu.
Make sure that you check both the *Fingers* and *Wires* boxes in the *Options* window pane.

Separate Bond Fingers



5. If you need to re-position the wire connect point on the original multi-wired bond finger, choose *Routing – Wire Bond – Tack Point Move* (`wirebond tack point` command).

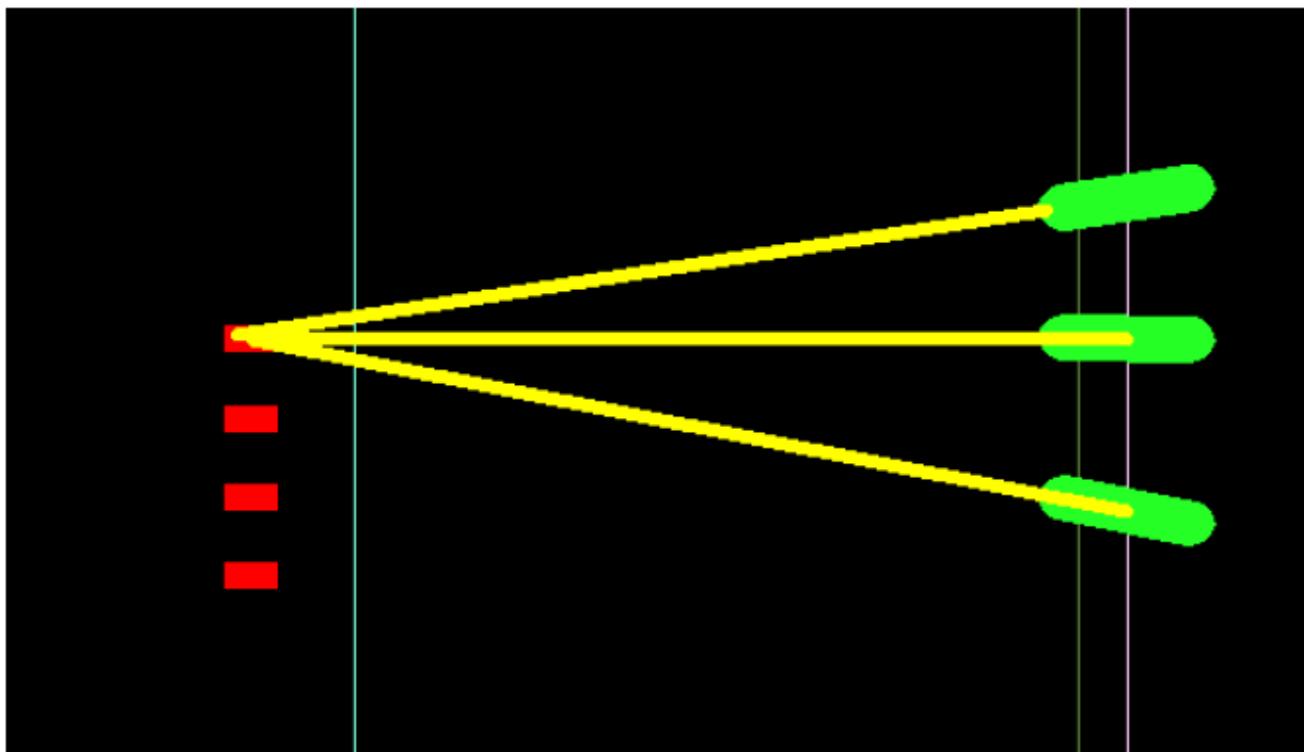
Bonding a Pin to Multiple Bond Fingers

1. Choose *Setup – Preferences* (`enved` command). From the *Ic_packaging* category, choose

Wirebonding and check the `wirebond_multewire_pins` box.

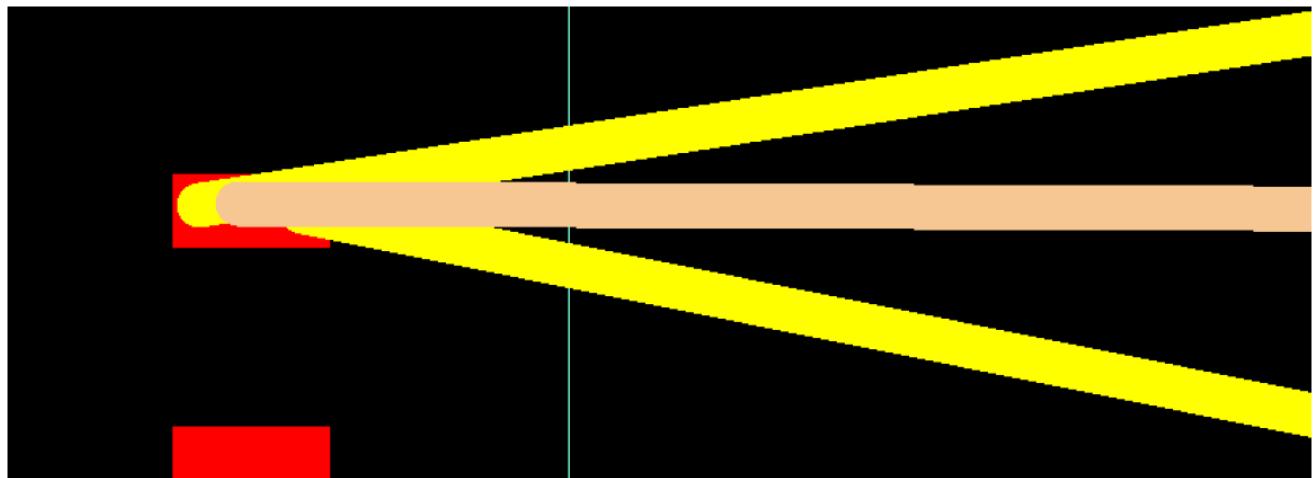
2. Choose *Route – Wire Bond – Select* (`wirebond select` command).
3. Choose *Pins* in the Find Filter, select a pin, right-click and choose *Add* from the pop-up menu.
4. Select the same pin and repeat step 3 to add additional wires until complete.

Single Pin Bonded to Multiple Fingers



5. Choose *Route – Wire Bond – Tack Point Move* (`wirebond tack point` command) to spread wire connect points on the die pad.

Using the tack point move Command



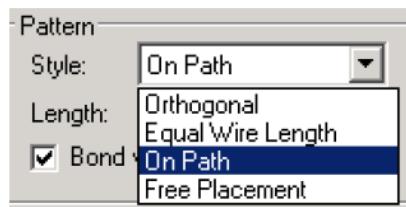
Pushing and Shoving Same Net Bond Fingers

This topic describes the two-phase process of pushing and shoving same net wire bonds, the environment variables that you set to control placement, and the use model to follow.

Push and Shove: Phase 1

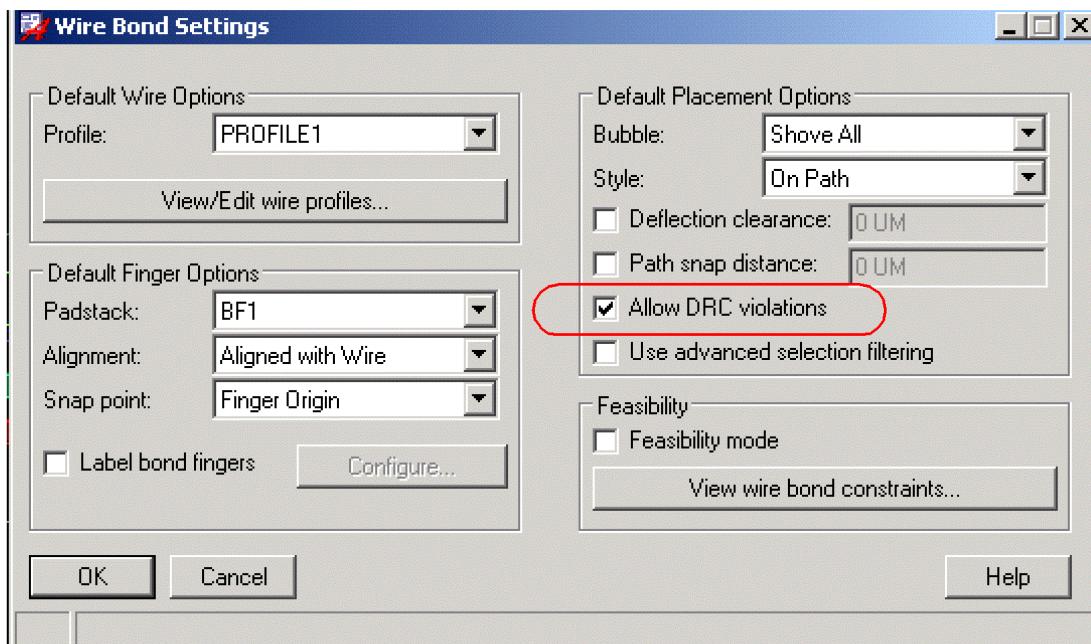
In Phase 1, once you pick a wire bond and choose the *Move* command, the referenced bond finger is on your cursor, and you move it along the guide path. The tool considers the current cursor position and the *Style* for the referenced wire bond. If you are in:

- *On Path* mode, the tool snaps the point to the nearest legal guide path (based on signal wire bonds or power and ground nets).
- *Equal Wire Length* mode, the tool snaps it to the circular path around the source pin for the wire bond.
- *Free Placement* mode, the tool places the bond finger where your cursor is positioned.



If you disable *Allow DRC violations* in the Wirebond Global Settings dialog box (`wirebond settings` command), the tool checks to see if DRC violations are created by placing the wire bond

in the specified location. If DRC violations are created, the tool moves the bond finger position back toward its original location until it finds a placement location that satisfies all the active constraints.



Push and Shove: Phase 2

Phase 2 is a recursive phase where the tool adjusts the wire bonds surrounding the referenced bond finger. The tool works in one direction clockwise from the referenced bond finger, and then in the opposite direction. It continues until all the wire bonds are adjusted.

Processing the Surrounding Wire Bonds

First, the tool finds the next wirebond that was affected by the previous wire bond's new placement. It also looks at the sorted order of the wire bonds, based on their original placement before the push and shove process started.

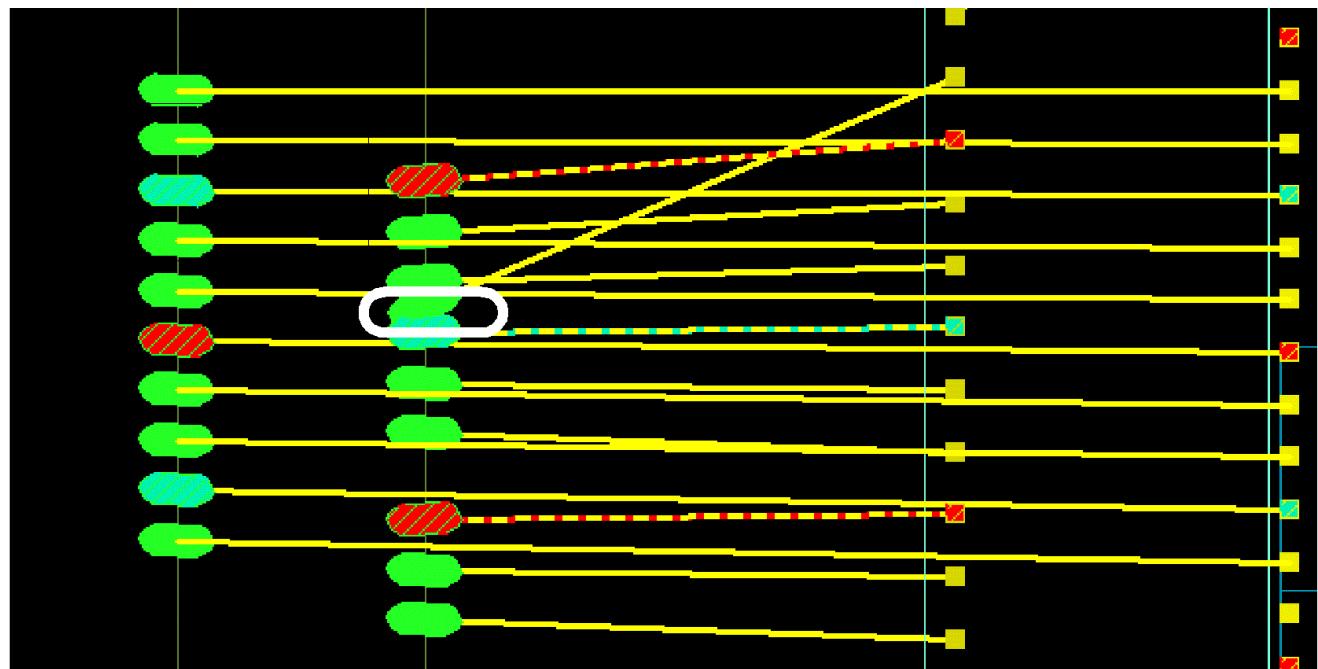
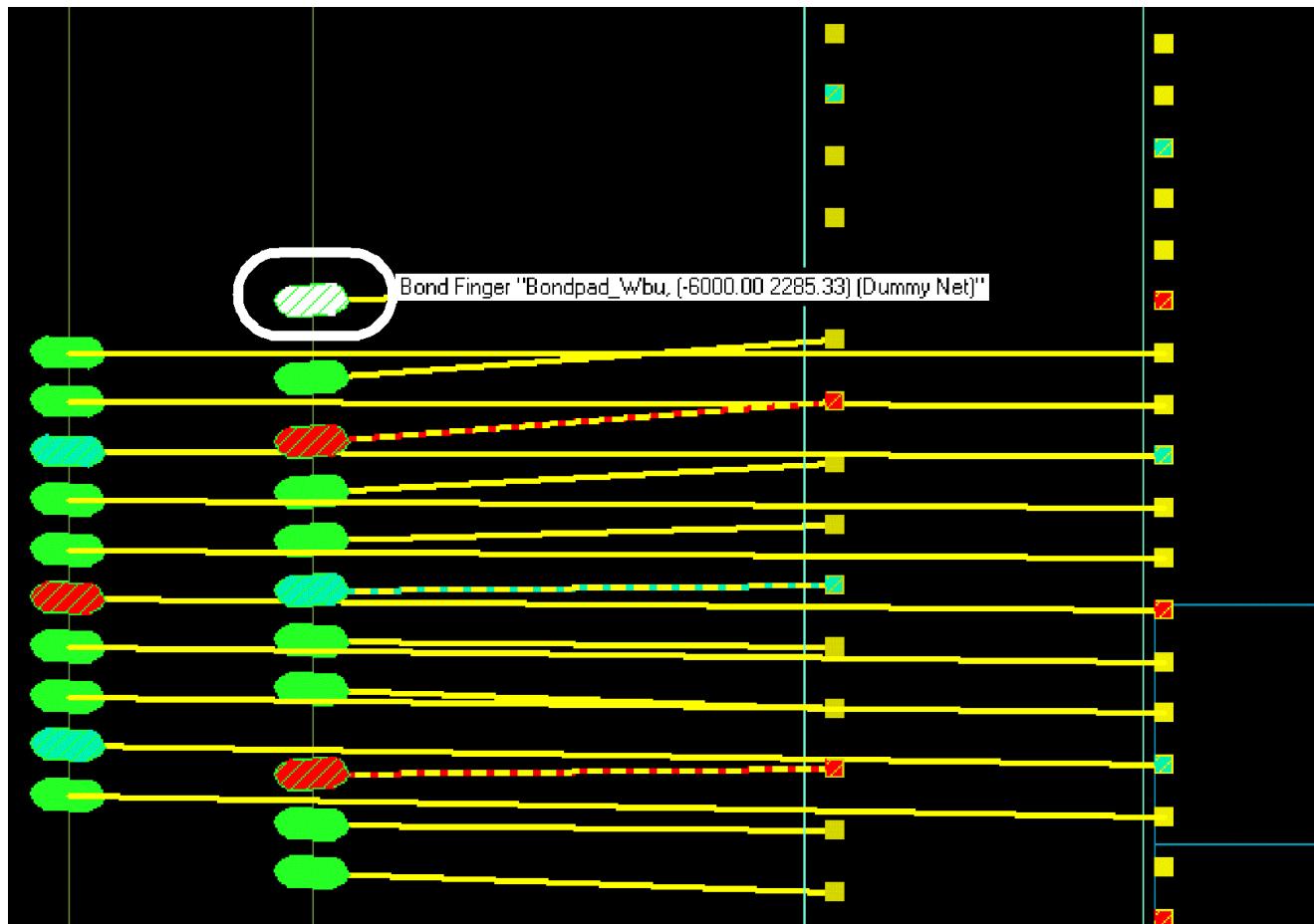
If you set *Bubble Mode* to:

- *Shove Off*, the tool moves only the referenced wire bond; it does not move any other wire bonds.

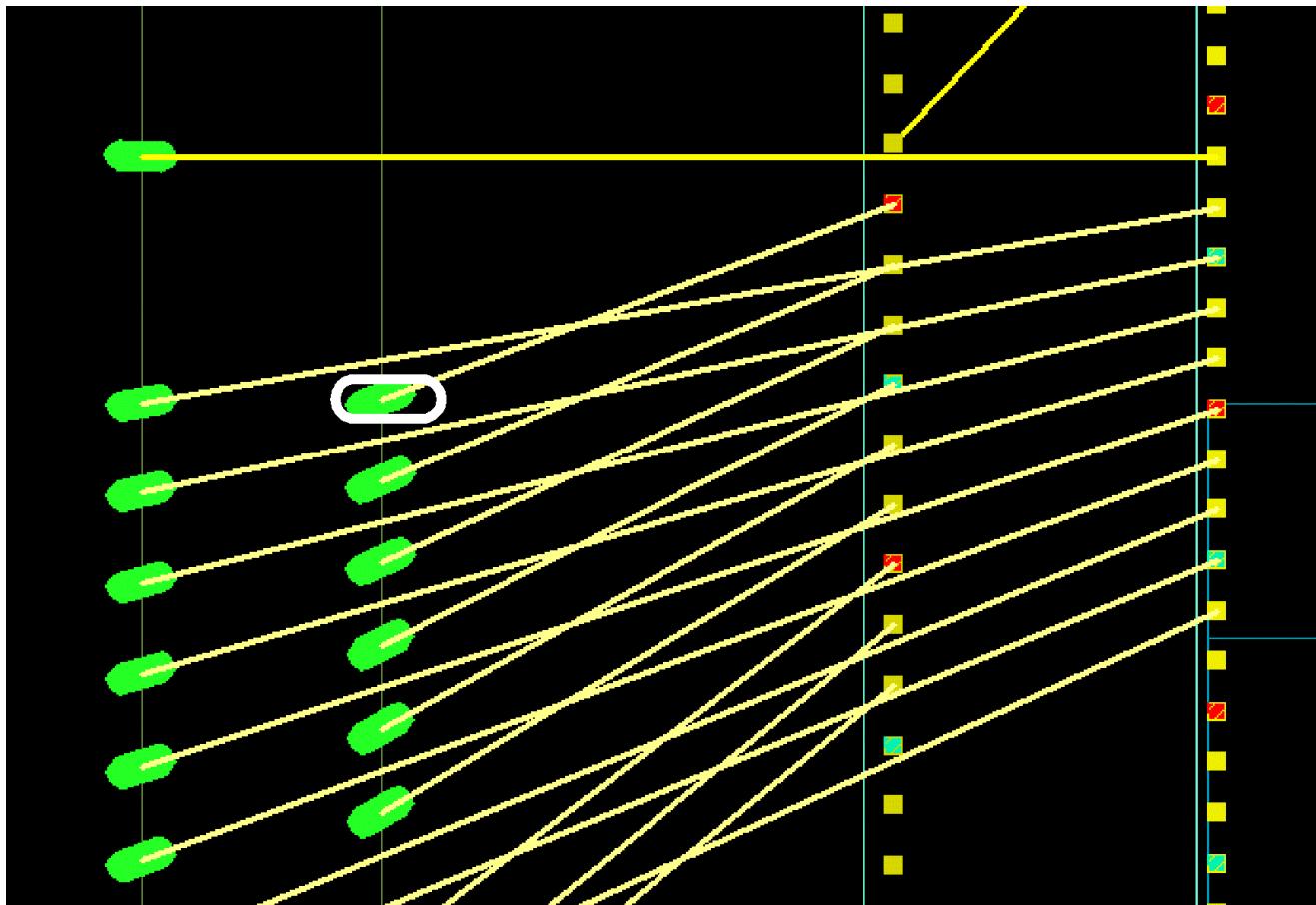
The referenced wire bond is circled in white in the following figures.

Routing the Design

APD: Introduction to the Wire Bonding Toolset--How the Wire Bond Die Escape Generator Works

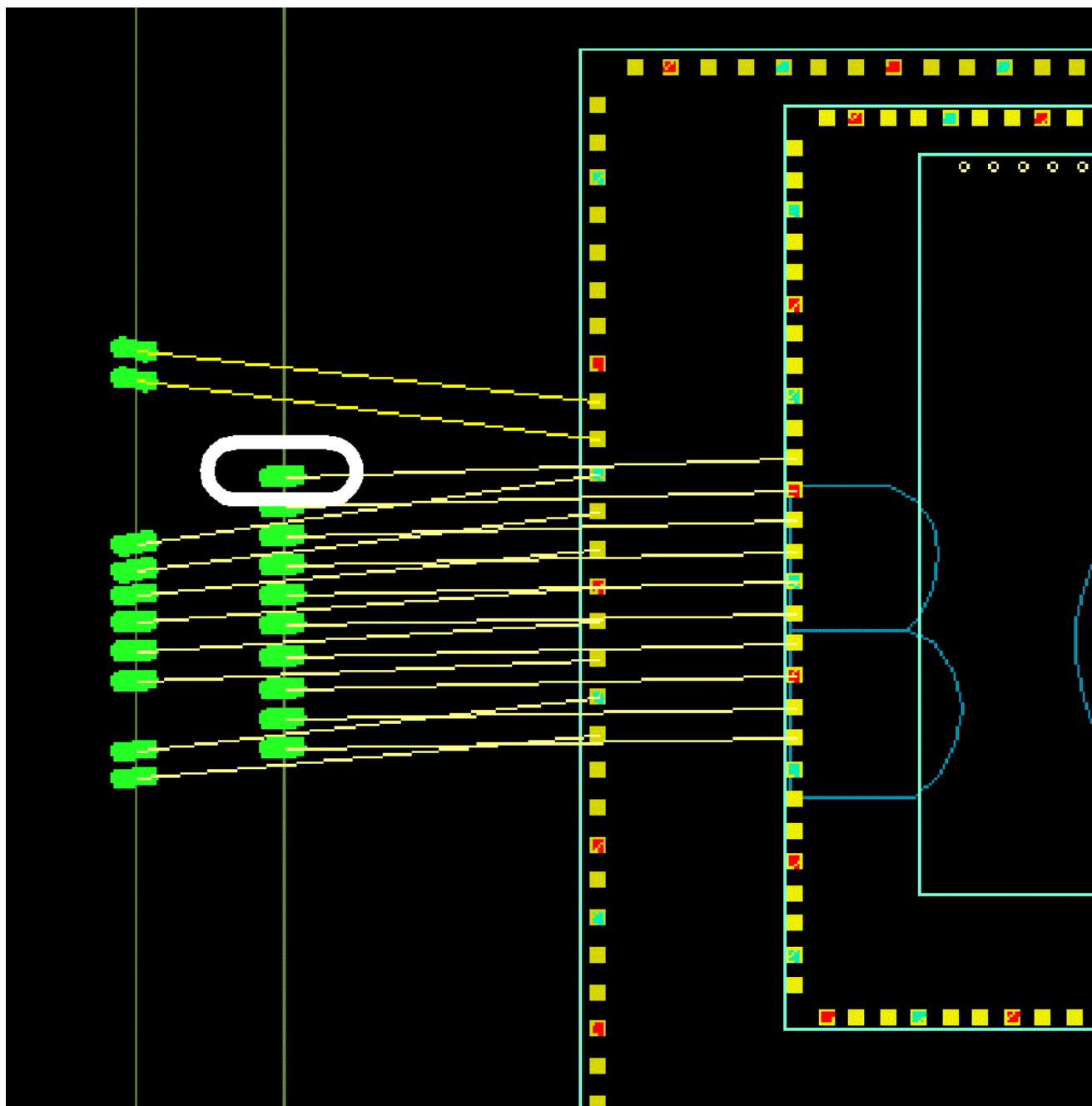


- *Shove Path*, the tool filters all wire bonds except for those on the same path as the referenced bond finger. It only moves the wire bonds on the same guide path.



Usually, you can perform placement faster, and more predictably, by using the *Shove Path* setting for complex, multi-profile designs. This works when you know that wires to fingers on different paths are on different profiles, which clear each other adequately in the Z-space.

For less complex patterns, or when you are not sure of the interaction between the wires of different profiles, you should use the *Shove All* mode.



If there are multiple guide paths with wires on different profiles, the tool checks if different profile wire-wire DRC violations means that it has to process any of these from their original sorted order. To visualize what the tool is doing, the order of bond fingers on the same guide path never changes. But the tool may want to process an inner bond before an outer bond, if the outer bond is not going to be impacted by the inner bond wire placement.

Establishing a Wire Bond Pattern

Once the tool identifies the next wire bond, it satisfies Design Rule Check (DRC) constraints for the specified wire bond relative to the wire bonds that are already updated. In this case, the tool ignores those wire bonds that come after the specified wire bond in the sort order, as they have not yet been updated to a new location.

The tool always checks both wire-to-wire and finger-to-finger spacing, regardless of whether you enable or disable the `wirebond_ignore_disabled_cns` environment variable, to ensure that wire crossings are minimized and that no electrical shorts have been introduced on the substrate surface.

- **Wire-to-Wire Spacing**

The tool attempts to satisfy wire-to-wire spacing by checking for DRC violations between the wire bonds being placed and the ones already placed. It moves the current wire bond to meet wire-to-wire DRC constraints, and then moves on to satisfy the next constraint that is enabled. In addition to wire-to-wire and finger-to-finger checks, the tool also checks finger-to-wire, wire-to-finger, and finger-to-cline (if there are routing stubs), subject to the setting of the `ignore_disabled_constraints` environment variable.

- **Wire-to-Wire Spacing on Different Profiles**

When checking wire-to-wire spacing for different profile wires (the `wirebond_ignore_wire_profiles` environment variable is disabled), the tool computes the distance between the two straight-line wire segments. It also computes how far along each wire that the crossing point occurs. If the point is less than the DRC-system specified percentage of the wire length, then the tool does not consider this an error. The recommendation is that you use the Cadence 3D Design Viewer to check the exact 3D clearance clearances of bond wires before design signoff.

- **Finger-to-Finger Spacing**

When checking finger-to-finger spacing, the tool picks the appropriate rule based on the setting of the `wirebond_ignore_disabled_cns` variable, the *On* or *Off* status of the same net and differential pair finger spacing constraints, and the nets on which the wire bonds are located. For example, if the two objects being compared are on the same net, but the tool ignores disabled constraints and you disabled the same-net DRC, the tool uses the general finger-to-finger spacing requirement.

- **Wire-to-Wire Spacing and Multi-wired Bond Fingers**

Also, when checking for wire-to-wire spacing, if your design has multi-wired bond fingers, the tool considers them as a single object. If you have one bond finger with three wires and place

it relative to a bond finger with two wires of its own, the tool tests all three wires individually against both wires of the other bond finger. The tool does not treat the wires as a big *cone* area between the two outermost wires.

-  The tool does not check wire-to-pin spacing during wirebond placement because that is an online DRC violation. Also the tool cannot correct it, and often, it is not a problem once you consider that the wire is in 3D space.

Best Placement of Wire Bonds

If the tool is unable to satisfy all constraints for this bond finger, and you have turned off *Allow DRC violations* (not recommended for multi-wire, multi-profile designs), then it places the current bond finger at the best place possible. Then it adjusts the wire bonds from this one back to the reference wire bond so that there are no DRC violations. This allows the tool to bump against a fixed wire bond or to the end of a guide path.

-  The tool pushes and shoves only those objects that belong to the same die stack (die for APD), and which are on the same side of the die as the selected referenced bond finger.

Push and Shove Wire Bond Environment Variables

These environment variables found in the User Preferences Editor are related to the push and shove wire bond process.

-  Be sure that when you set the environment variables, you exit the wirebond command you are running and run the command again.

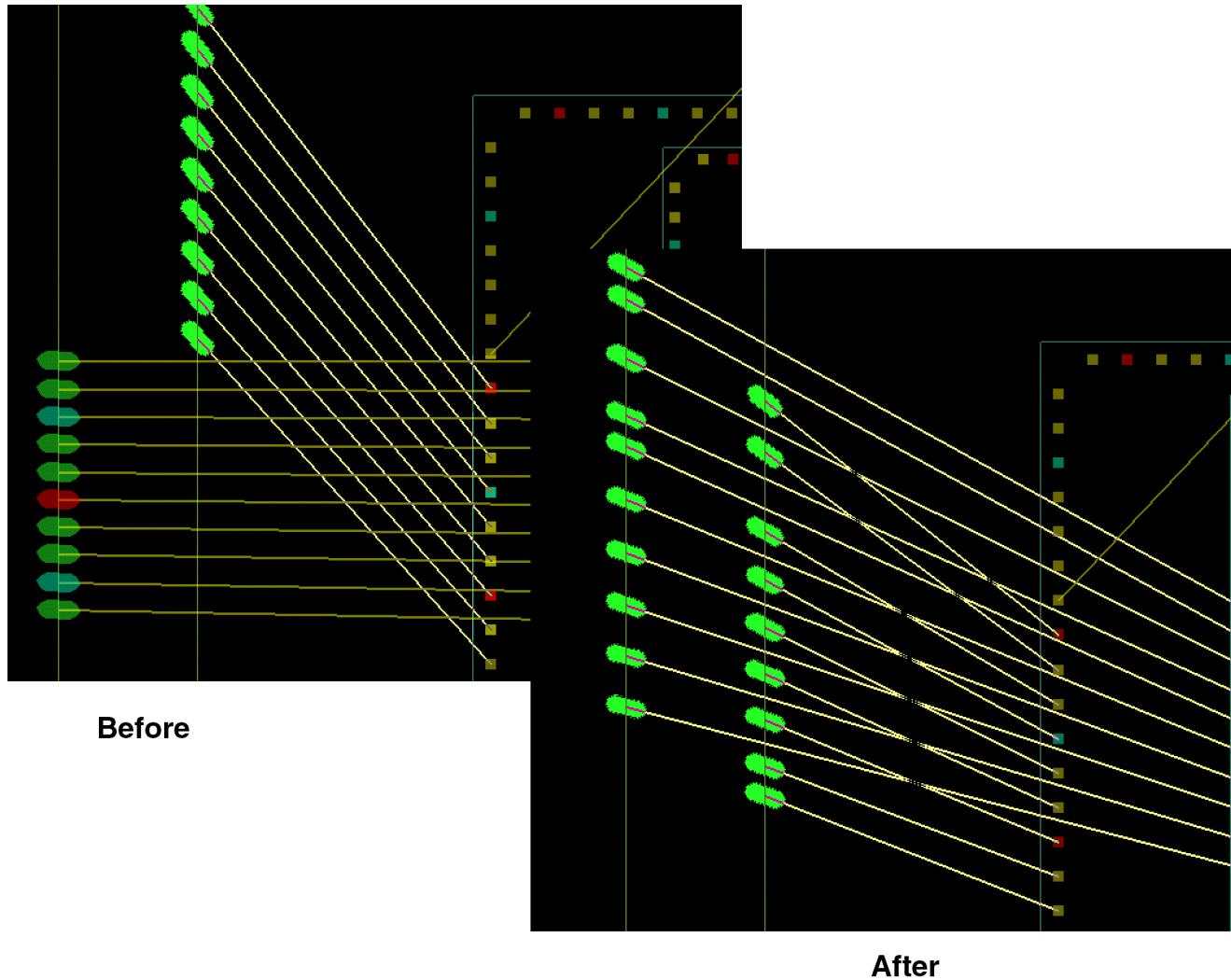
- `wirebond_ignore_wire_profiles`

By default, the tool respects wire profiles when computing legal wire bond placements. As a result if two wires use different profiles, they are allowed to cross within a specified distance from the die or pin. Turning this environment variable on causes the tool to treat all wires as if they were in the same profile for placement only, thus ensuring that the wire bonds do not cross.

For example, if you have four identical memory stacked dies bonding down to common pads on the substrate, the tool ignores wire-wire checks to the same bond finger, so it is not an issue. However, you would need to disable this variable if you have multiple stacked dies

bonding out to different fingers on different paths, since most likely the wire bonds would cross.

The following figures show how the tool works before you set the `wirebond_ignore_wire_profiles` environment variable and after you set the environment variable.



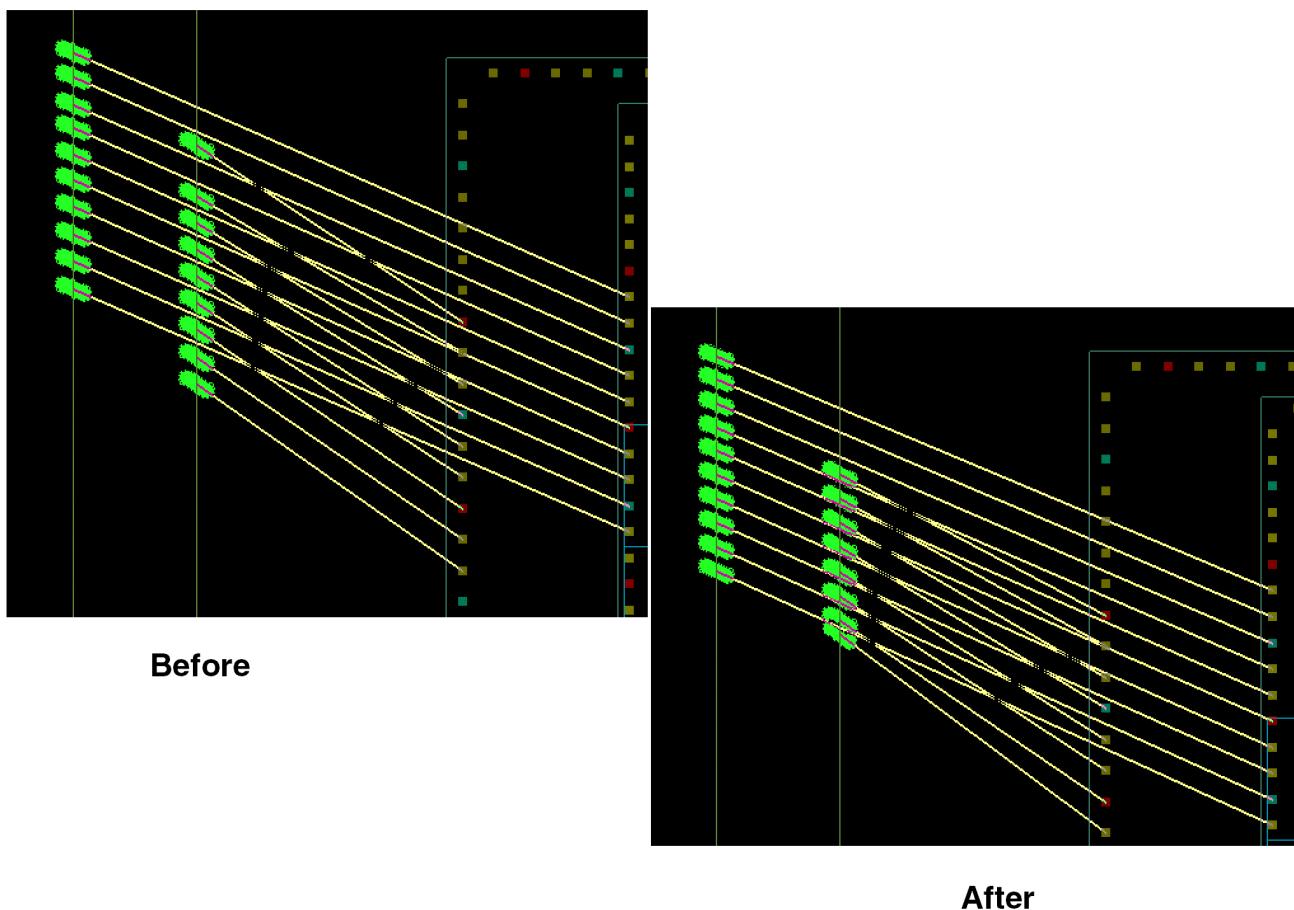
- `wirebond_ignore_disabled_cns`

By default, the tool attempts to satisfy wire-to-wire, finger-to-wire, and finger-to-finger DRC constraint values during placement, regardless of whether you have currently enabled them for online DRC. By enabling this variable, disabled DRC is not used to guide placement. Bond finger-to-wire is not checked if disabled. Same-net and differential pair finger-to-finger spacing use the general finger-to-finger spacing. Note that finger-to-finger and wire-to-wire spacing are always used to ensure that placement does not introduce unnecessary electrical

shorts.

Enable or disable this environment variable so that you can remove finger-to-wire spacing considerations during placement. For example, in a dense multi-wired design, the bond wires cross over other bond fingers on inner paths, especially as you advance outside the pattern.

The following figures show how the tool works before you set the `wirebond_ignore_disabled_cns` environment variable and after you set the environment variable.



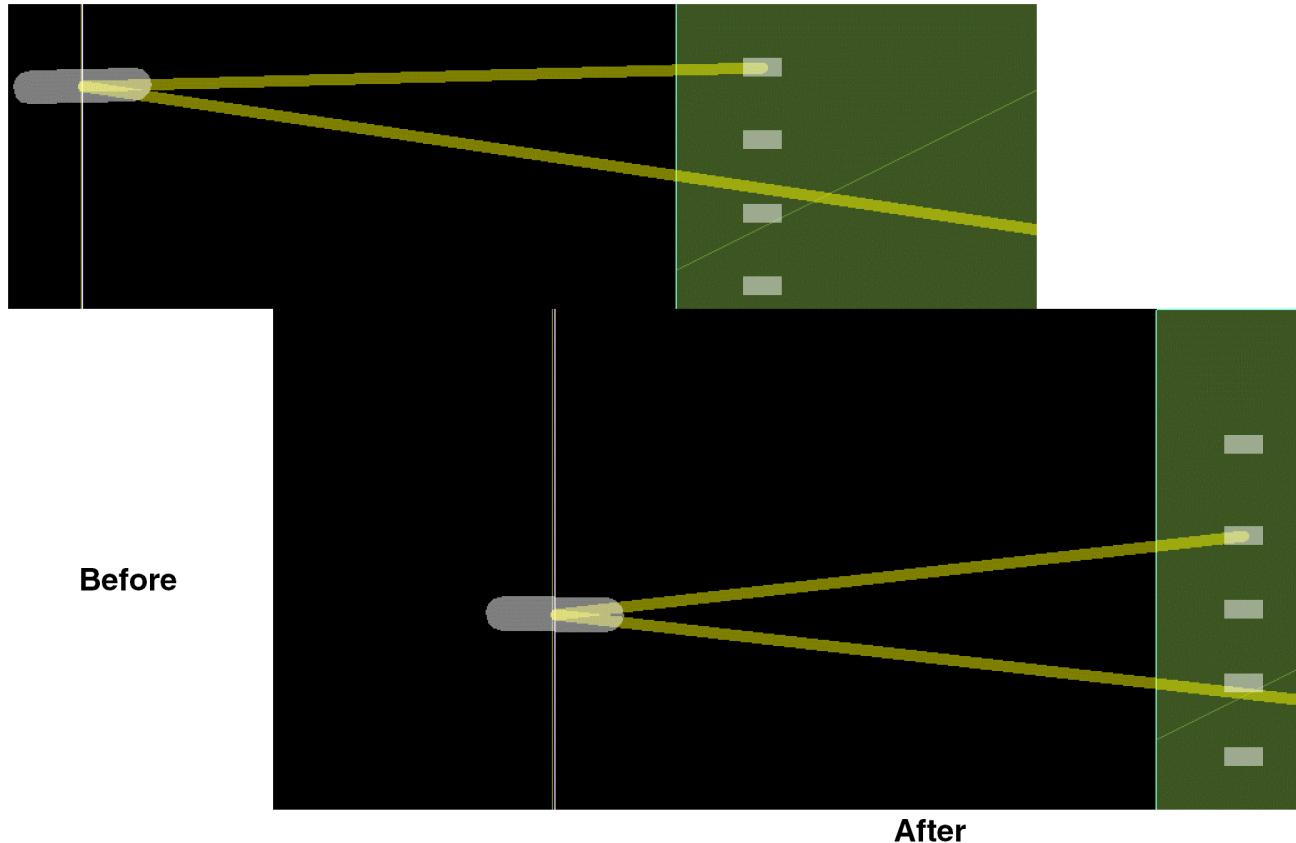
- `wirebond_align_average_wire_angle`

When a bond finger has only a single connected wire, the tool easily uses the *Align With Wire* setting. For multi-wired bond fingers, however, the tool either aligns to the primary (dominant) wire or else aligns to the average angle of all the wires entering the bond finger. By default the tool aligns to the dominant wire.

For example, if you are working on a variant design or a multi-die stack, you probably want the bond finger to be at the average angle so that no wire has a bad entry angle into the pad.

The following figures show how the tool works before you set the

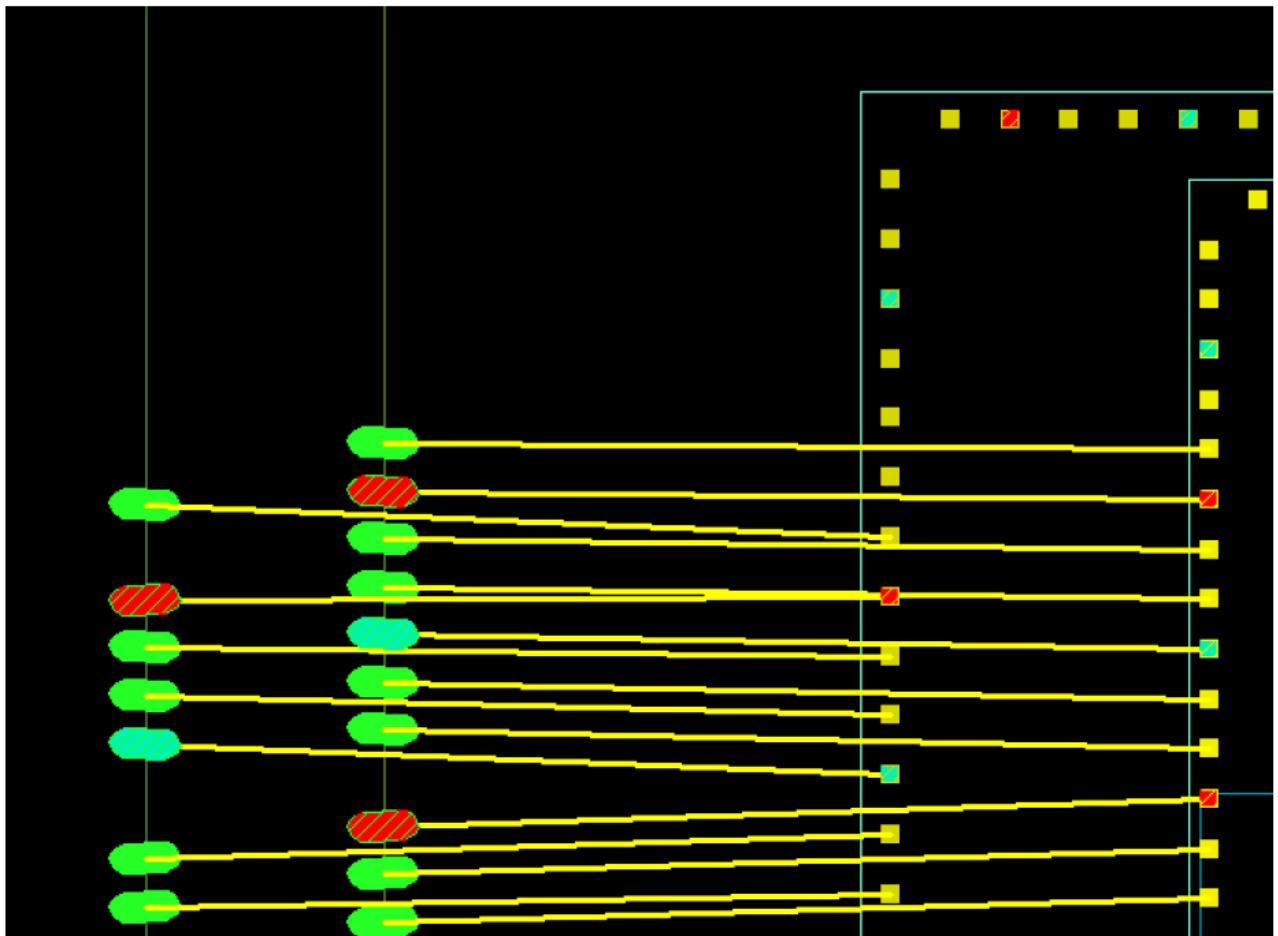
`wirebond_align_average_wire_angle` environment variable and after you set the environment variable.



Pushing and Shoving Wire Bonds

1. Set these environment variables described in the Push and Shove Wire Bond Environment Variables as follows:
 - a. Enable the `wirebond_ignore_disabled_cns` and `wirebond_align_average_wire_angle` environment variables.
 - b. Make sure that the `wirebond_ignore_wire_profiles` environment variable is disabled.
2. Set the global constraints in the Constraint Manager:
 - a. Disable *Finger to Wire*, and possibly *Finger to Route* (for performance, if the design is already routed).
 - b. Enable *Same net* and *Diff pair finger to finger*

3. Enable *Allow DRC violations*. in the Wire Bond Settings dialog box.
4. If you know that your profiles have adequate 3D clearance, set *Default bubble* to *Shove path* in the *Options* window pane.
5. Perform pushing and shoving as follows:
 - a. Identify the wire bond to use and select it. Select only one wire bond for reference. If you window select, the tool attempts to maintain relative spacing among all your selected items.
 - b. Once you start pushing and shoving, look at the pattern. If you are in *Shove All* mode, it may affect wire bonds on other guide paths. While these changes may be correct from a DRC perspective, you need to assess whether you want to correct the DRC violations through push and shove, or through changing the profile heights. If you change the profile heights, then switch to *Shove Path* before continuing.



- c. Move the referenced bond finger to the desired location, paying attention to all the

impacted bond fingers around it.

- d. Once they are placed, you may need to adjust wire end points on the fingers. If so, use the `wirebond tack point` command.

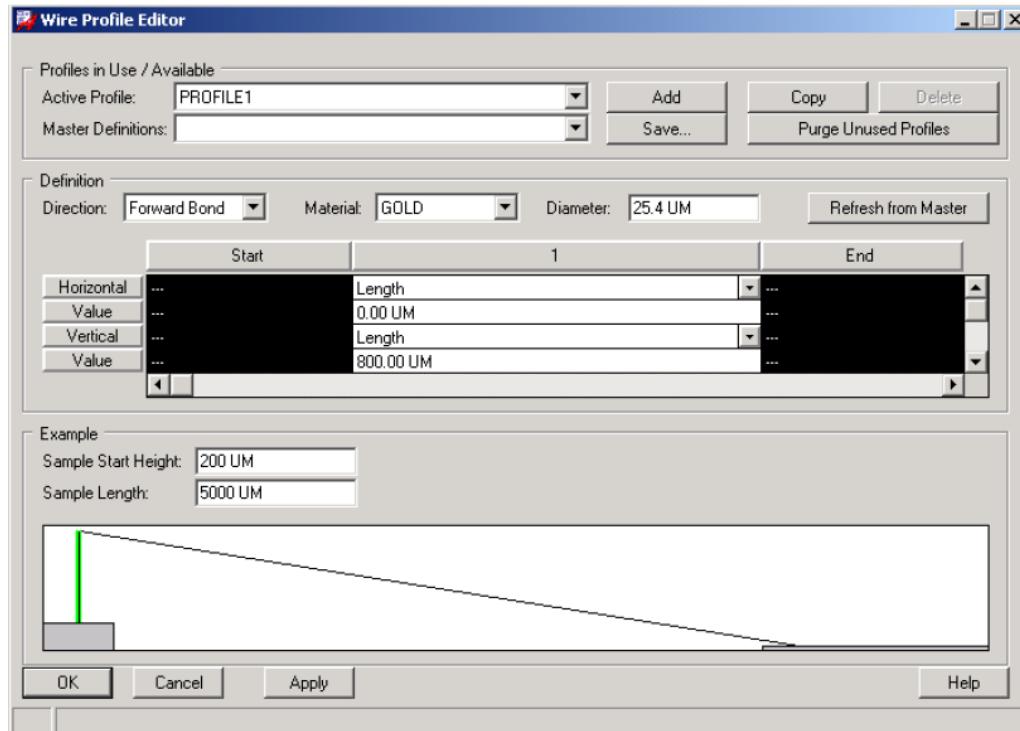
Correctly Staggering Die Pins to Avoid False DRC Violations

This procedure describes the wire bonding process and constraint setting when you have a die with multiple rows of pins or multiple stacked dies.

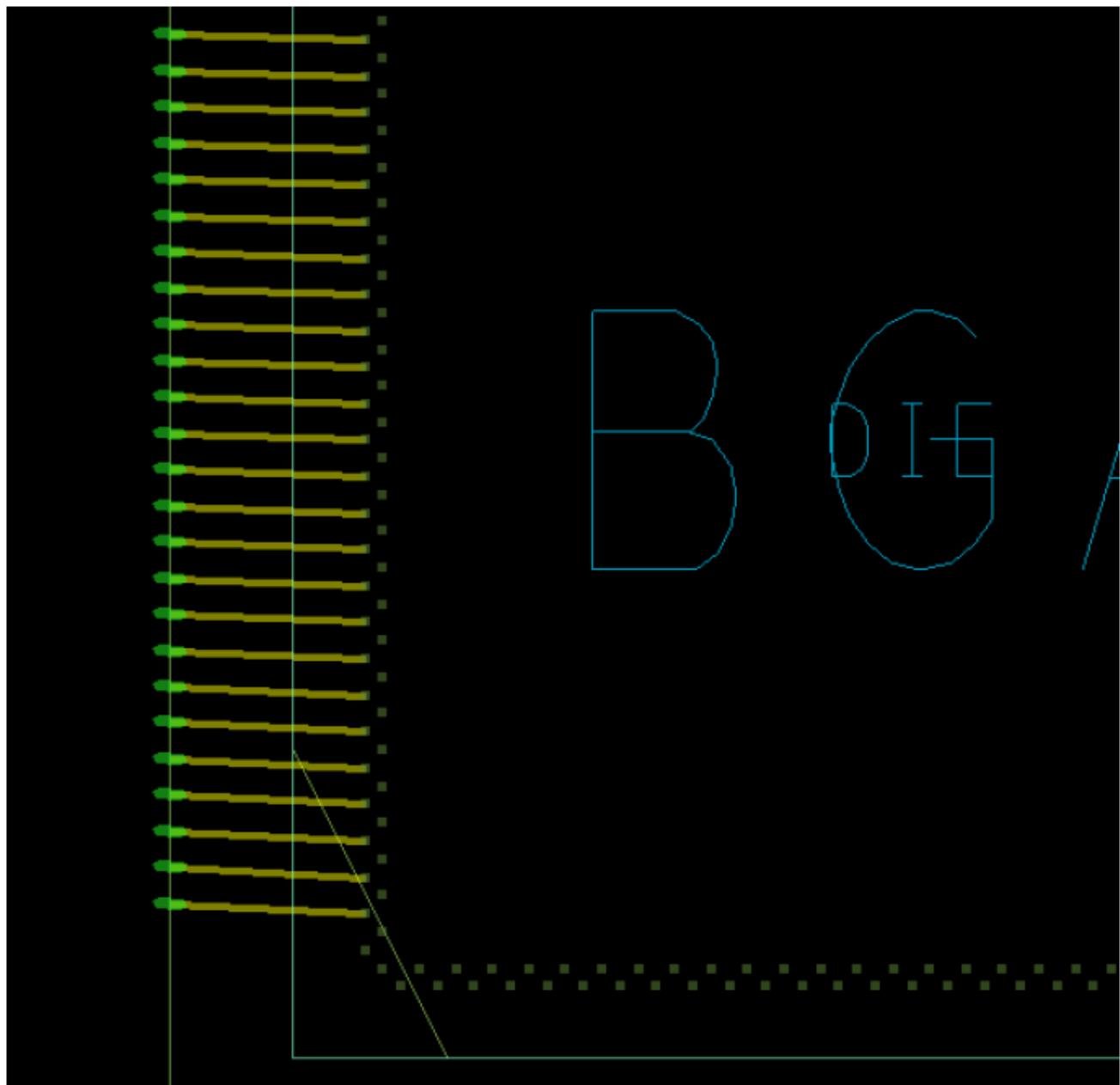
1. Turn off the *Wires to Pin* constraint in the Constraint Manager if you have a staggered pin pattern.
If you do not disable this constraint, wires from the inner ring of pins may cross over or be too close to the outer pins of the same or a lower component. Because this is a 2D DRC, you do not need this constraint enabled.
2. Make sure that the *Same profile Wire-to-wire* and *Diff profile Wire-to-wire* constraints are On and that you set the *Cross Length* constraint so that it clears the edge of the die or die stack. For dense designs, turn Off the *Finger-to-Wire* constraint. The wires are going to cross over the bond fingers of the inner rows of bond fingers. However, they clear the bond fingers adequately in the Z-axis. Since the *Finger-to-Wire* constraint is a strictly 2D check, it is not useful for this type of design.
3. Define or select your wire profiles in the Wire Profiles Editor using the `wirebond` settings command (Bond fingers on different guide paths are in different profiles).
Define your profiles so that the inner pins use a profile that climbs higher at the start of the profile than the outer pins' profile. This ensures that in the die region, the wires clear each other vertically, since they cannot avoid crosses in the 2D horizontal plane.
The first step in the profile should be vertical to a height that you know will clear the profile in front of or below it.
Cadence recommends that you use pre-defined certified wire profiles from your wirebond manufacturer (for example, K&S) to ensure that the profiles can be manufactured by the bonding machine.
If you are using predefined profiles (for example, K&S) and are not sure what the ideal combination of profiles is for the different rows of pins, bond out a few pins using different profiles and look at them with the Cadence 3D Design Viewer to see how they interact. You can use the Cadence 3D Design Viewer to change the profiles so that you have adequate clearance. Then you can go back to the design and bond out the full pattern using the ideal profiles.

Routing the Design

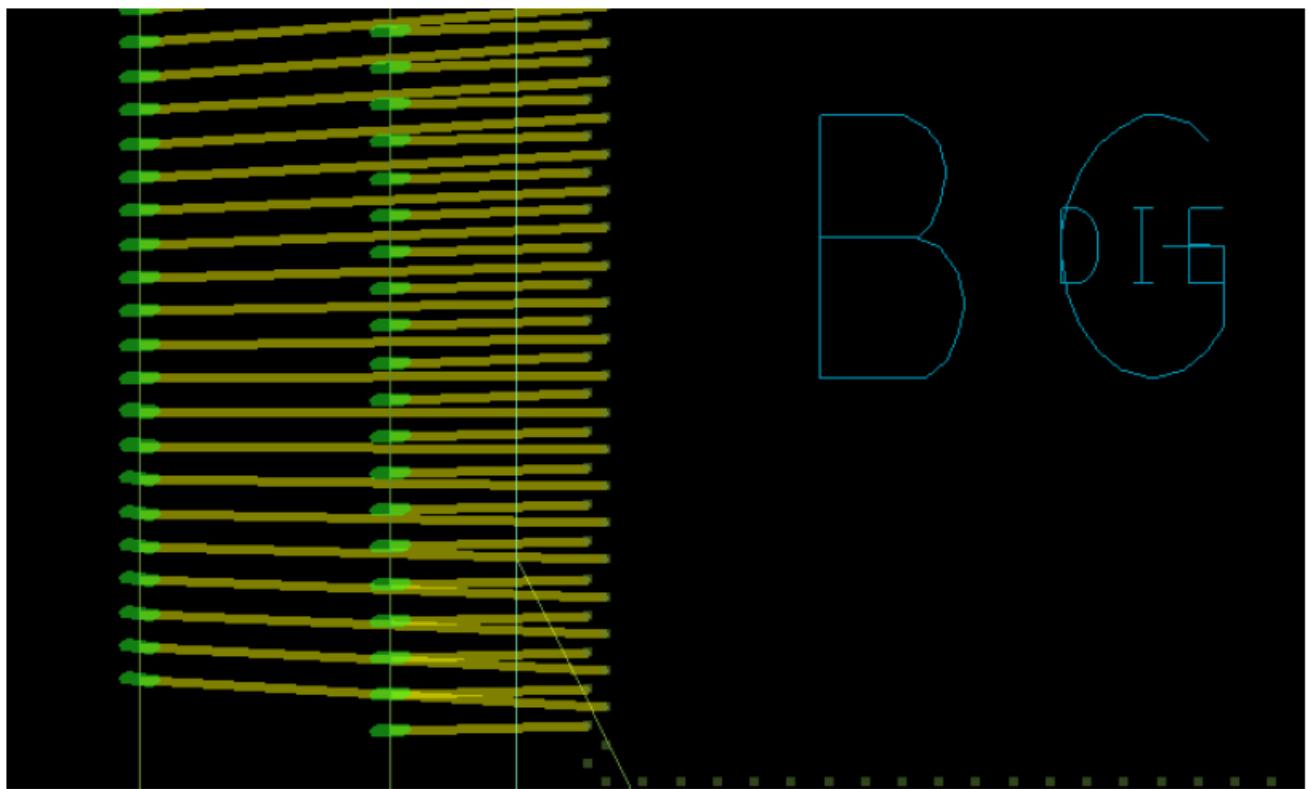
APD: Introduction to the Wire Bonding Toolset--How the Wire Bond Die Escape Generator Works



4. Add guide paths, placing them based on your finger-to-finger spacing, finger-to-shape (ring) spacing, and your minimum and maximum wire length requirements.
5. Bond the outer pins to bond fingers on the inner guide path.

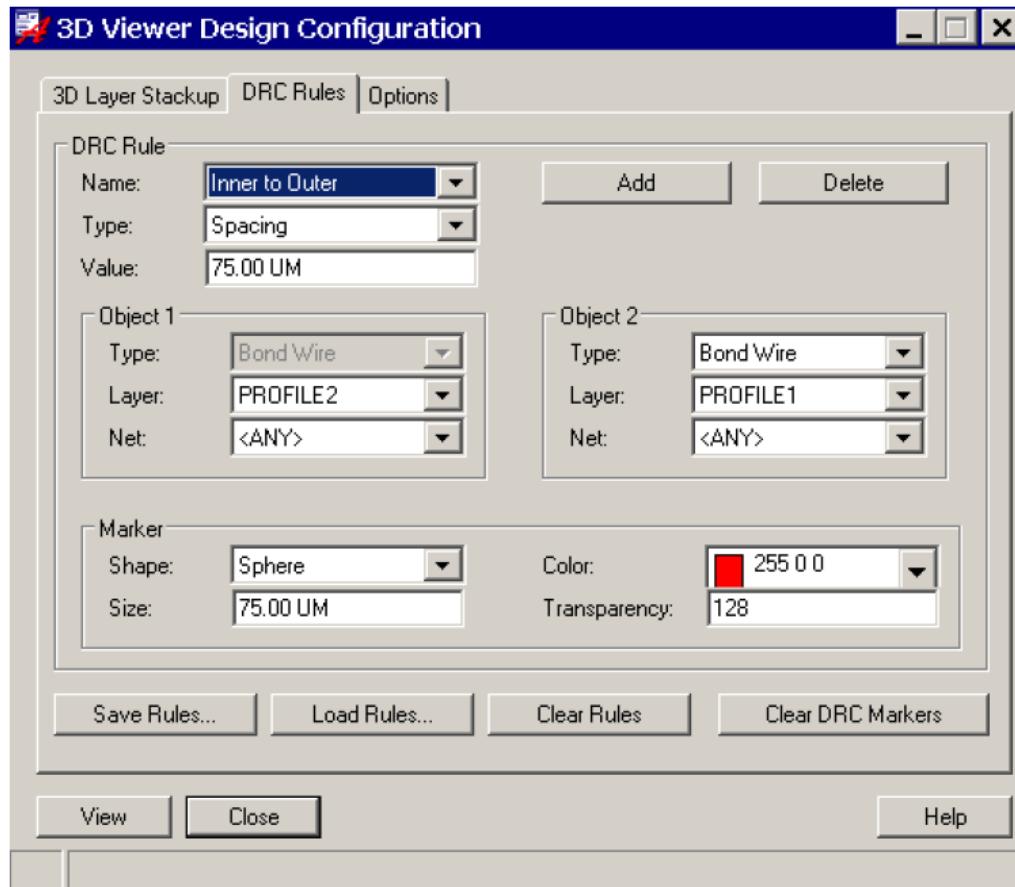


6. Bond the inner pins to bond fingers on the outer guide path.

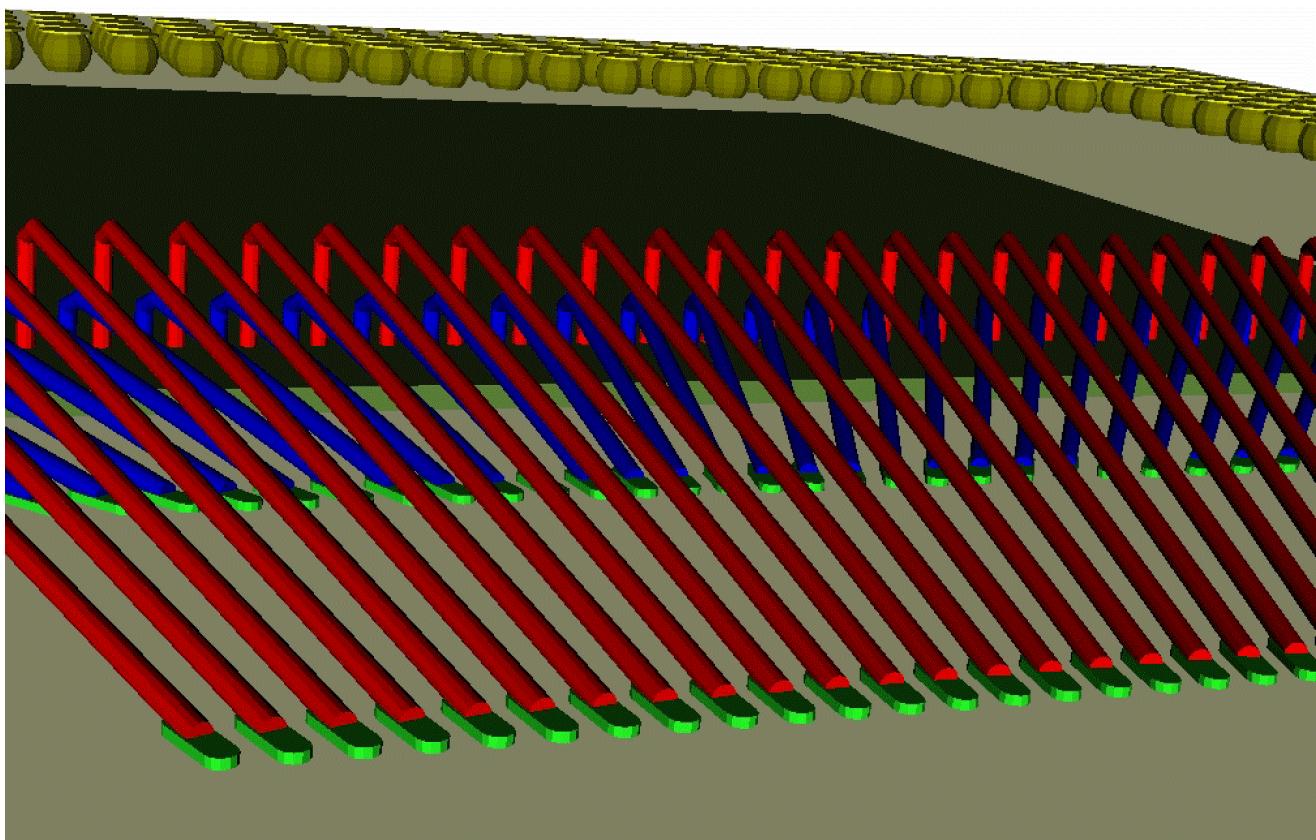


If performing single-layer substrate routing, make sure that you include routing stubs when adding the bond fingers to ensure that the appropriate routing channel space is left between fingers.

7. When finished preliminary bonding, load the design into the Cadence 3D Design Viewer with the actual 3D DRC rules specified by your team.



8. Check all wire clearances as necessary, and backannotate any 3D DRC violations into .mcm design using the *File – Update* menu item.



You can turn off (or hide) the online DRC errors for different profile spacing, since the 3D DRC errors are more accurate.

9. As you continue to refine the pattern based on substrate routing and logic changes, use the Cadence 3D Design Viewer to validate that the 3D wire spacing is acceptable.

Wires Bond from Same Location Die Pins out to Same or Different Bond Fingers

This procedure describes how to wire bond a design with multiple same-type dies. In this example, wire bonding is complex because the pin locations are on top of each other.

Included are the following topics:

- Selectively choosing pins for wire bonding
- Bonding the pins to the same bond finger
- Moving tack points
- Avoiding crossing bond wires

Routing the Design

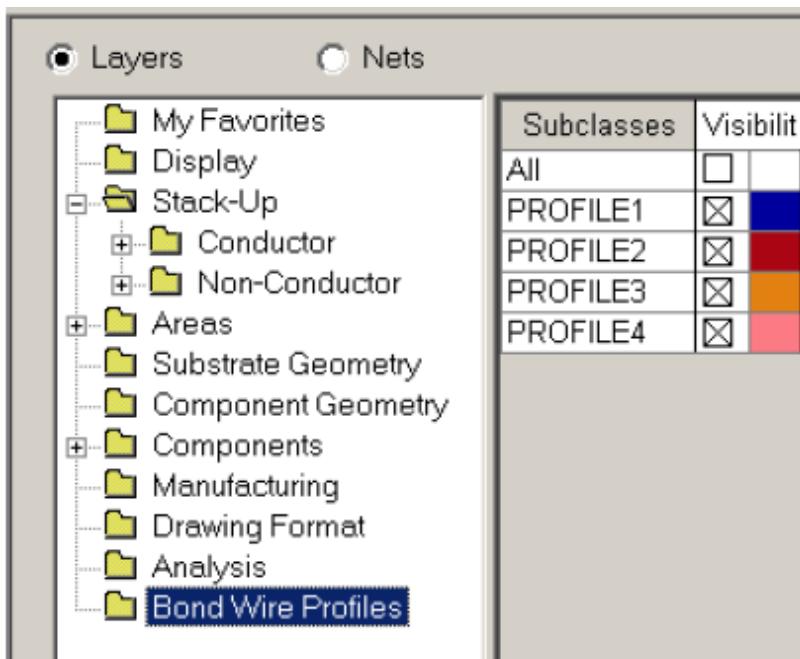
APD: Introduction to the Wire Bonding Toolset--How the Wire Bond Die Escape Generator Works

Follow the steps to wire bond a design:

1. To help you control the selection of pins for wire bonding, name the die stack layers in the cross section using the reference designator of the die component that occupies that layer.

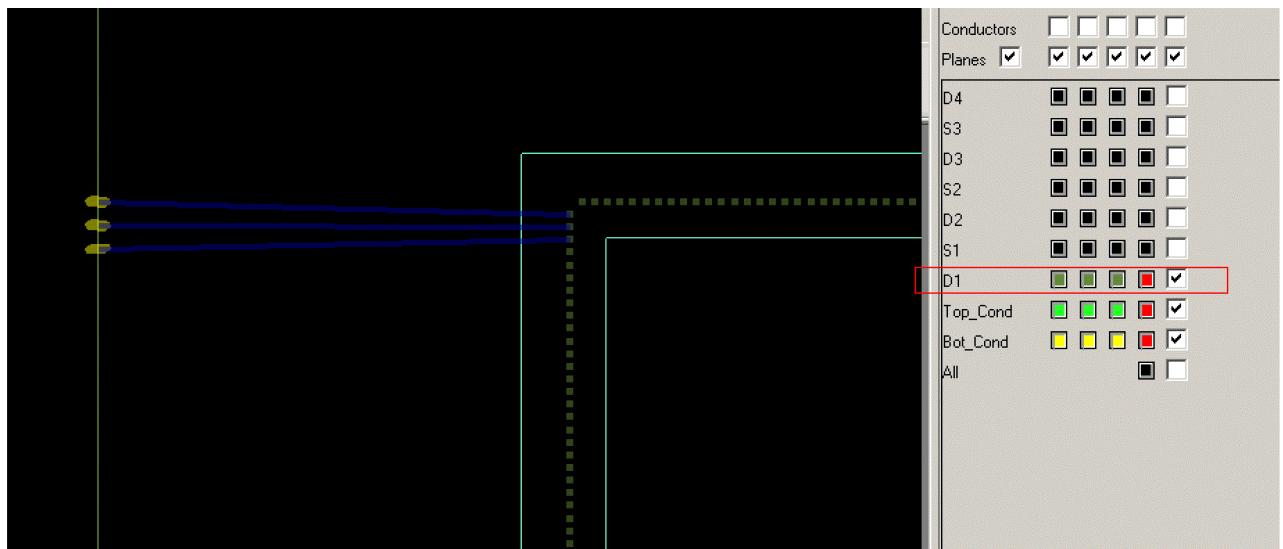
	Subclass Name	Type	Material	Thickness (µM)	Conductivity (mho/cm)	Dielectric Constant	Loss Tangent	Freq Dep File	Negative Artwork	Shield	Width (µM)	Etch Factor (degrees)
1		SURFACE	AIR			1	0					
2		DIELECTRIC										
3	D4	DIESTACK										
4	S3	DIELECTRIC										
5	D3	DIESTACK										
6	S2	DIELECTRIC										
7	D2	DIESTACK										
8	S1	DIELECTRIC										
9	D1	DIESTACK										
10		DIELECTRIC										
11	TOP_COND	CONDUCTOR	COPPER	30.48	595900	4.5	0		□		75.0	90
12		DIELECTRIC	FR-4	203.2	0	4.5	0.035					
13	BOT_COND	CONDUCTOR	COPPER	30.48	595900	4.5	0		□		75.0	90
14		SURFACE	AIR			1	0					

- a. Turn on the wire profiles (`color192` command) so that you can see the existing bond wires during wire bonding. Be sure that the bond fingers are visible.



- b. Turn off all die pad layers (*Visibility* tab of the Control Panel) except the layer of the die that you are currently bonding, and follow the procedure for adding wire bonds described in the `wirebond select` command.

D1 Bonding

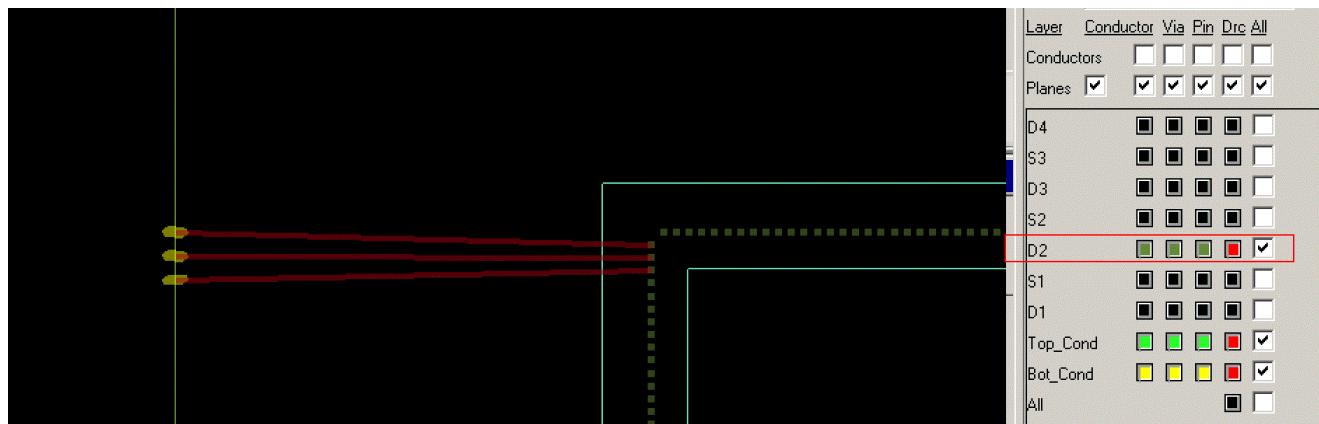


2. To bond pins to the same bond finger, add wires from other dies in the die stack to the existing bond fingers.

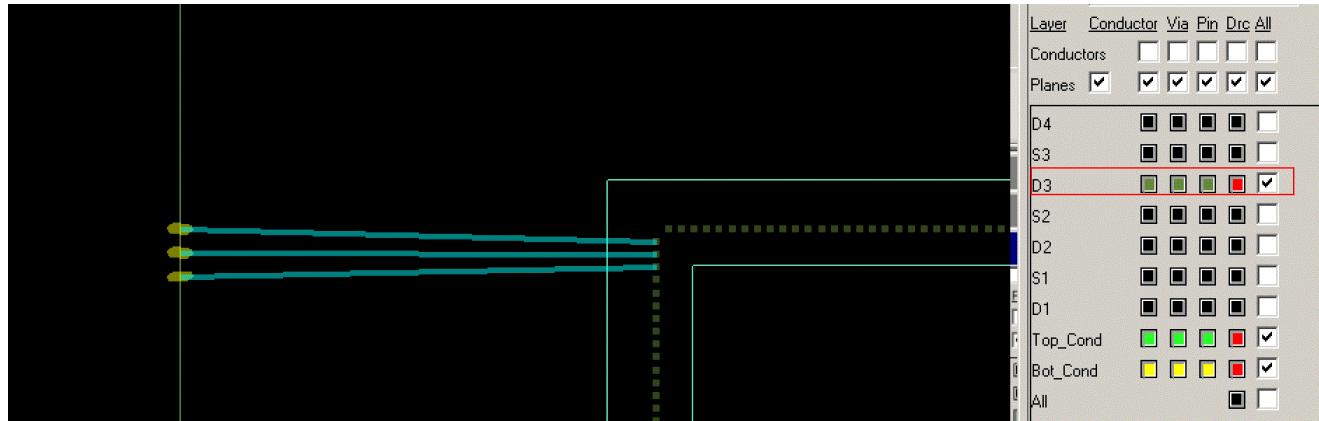
Follow the *Attaching Multiple Wire Bonds to One Bond Finger* procedure described in the `wirebond select` command.

⚠ Make sure that *Bond Wires* is not enabled in the Find Filter while working in the `wirebond select` command, or you may inadvertently move wires instead of adding new ones.

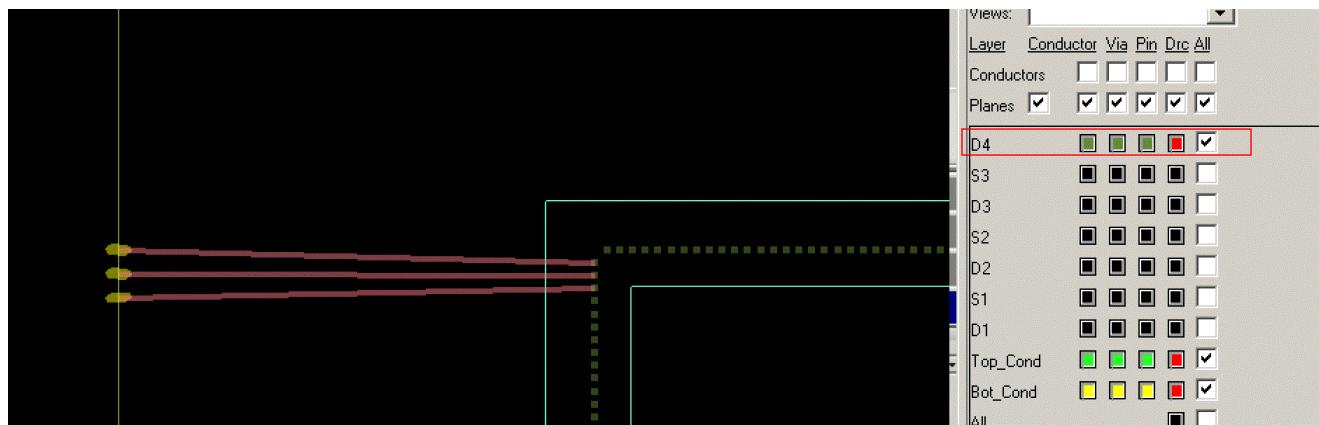
D2 Bonding to Existing Bond Fingers



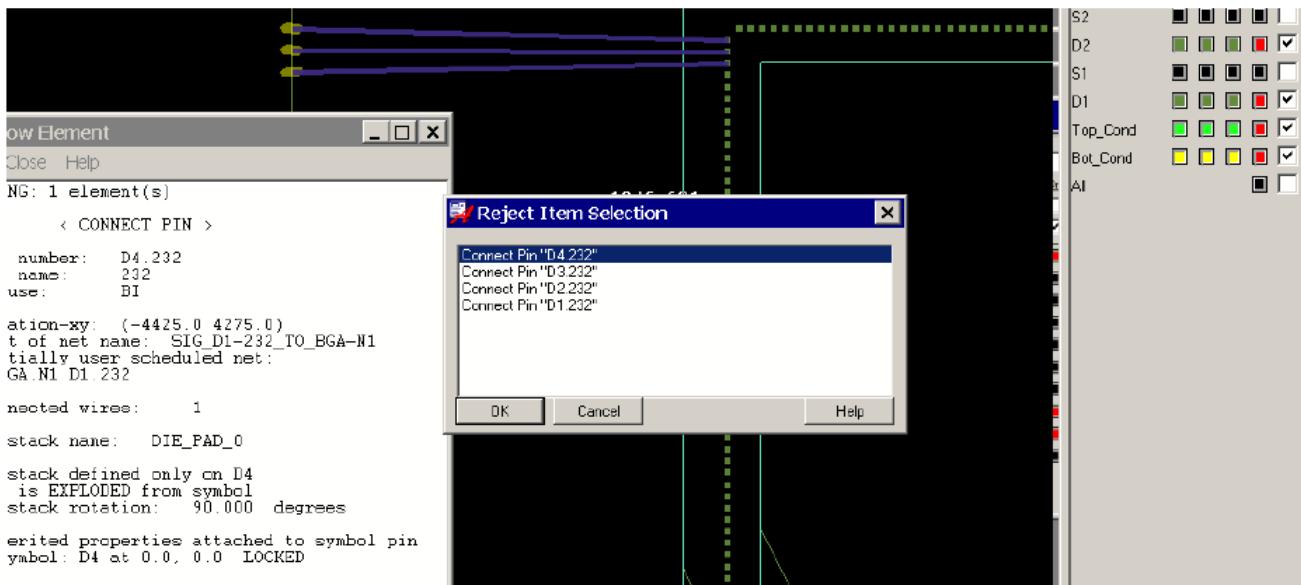
D3 Bonding to Existing Bond Fingers



D4 Bonding to Existing Bond Fingers



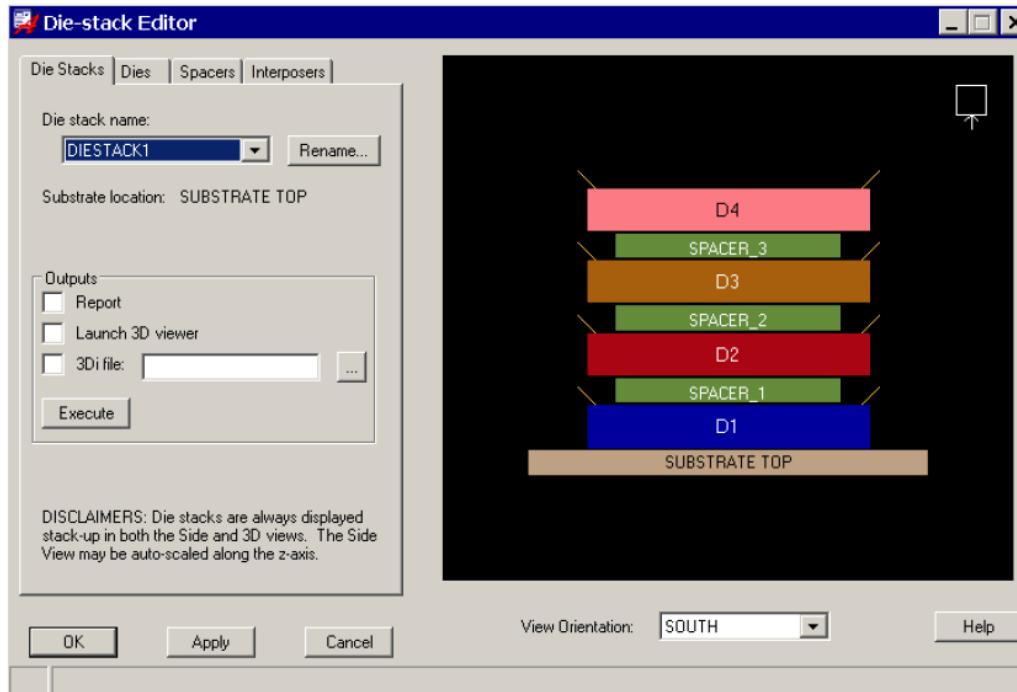
Wire Bonded Dies



The order of wire bonding is important when you move the tack points of the wires on the bond finger. In this case, you need to use the visibility of the bond wire profiles to control which wire is selected for movement. You want to bond the die pad of the highest die in the Die stack Editor to the tack point location farthest away on the bond finger.

To make this process easier, you may want to match the colors of the wire profile to the corresponding die pads, or match the profile name to the layer or reference designator, if that is an option. Then you can use *Visibility* tab on the Control Panel to view the order of the colors. Starting from the bottom (closest to the TOP_COND layer), turn off all profiles except for the current one, and then move the bond wire tack point to the next available location on the bond finger.

For example, the dies in the following die-stack example are D1 through D4 and are assigned profiles in the same order (P1 through P4).



3. Turn off all wire bond profiles except P1 (`color192` command).
4. Run the `wirebond tack point` command.
5. Select the P1 wires and move their tack point on the bond finger close to the die side of the finger.

Top Bond Finger (Profile 1) Tack Point Moved



6. Turn off the P1 wire bond profile and turn on the P2 wire bond profile.

7. Select the P2 wires and move their finger tack point out to the next innermost position on the finger (second closest to the die).

Top Bond Finger (Profile 2) Tack Point Moved

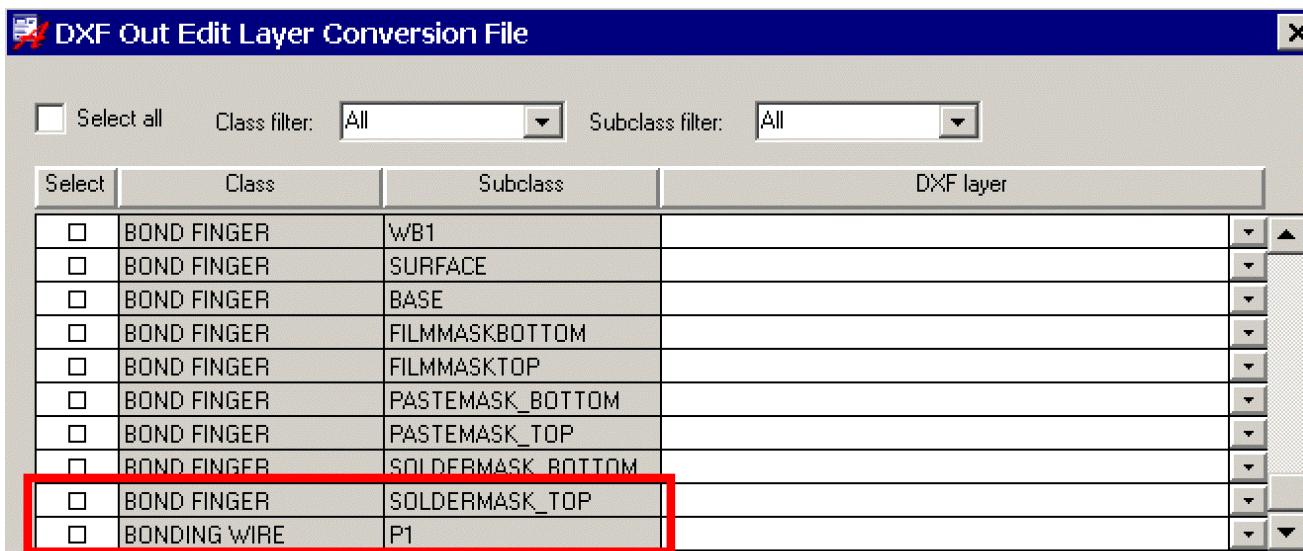


8. Repeat this process until you have selected the P4 wires and moved the tack point.
9. View the wire bonded dies using the Cadence 3D Design Viewer (`view 3d`).
If you follow this process, you will be able to view the wire bonds in the Cadence 3D Design Viewer and note that the wires do not cross.

Wire Bonding: Using the dxf in and dxf out Commands

BONDFINGER is a class in the `dxf cnv` file enabled using the `dxf_bond_finger_class` environment variable. You can, therefore, configure bond fingers separately from vias when exporting DXF. The subclasses are the same as those listed for the VIA class. The VIA class entries give you the vias only, and the BONDFINGER entries give you the bond fingers. You must enable both to get the full layer. This allows you to better control the output if you use DXF to generate wire bond bonding reports.

You can export the DXF for only the wirebond elements, that is, the profiles that you want and the bond fingers and die pads. This can then be used for a bonding diagram or report creation in third-party tools.



The following two environment variables can be used with the `dxf in` and `dxf out` commands:

- `dxf_bond_finger_class` – Lets you control the export of bond fingers separately from via objects. By default the bond fingers are exported using the VIA class. Setting this variable lets you use the BONDFINGER class.
Typically, package designers should enable this variable.
- `dxf_suppress_wire_vias` – Turns off the export of *connector* vias between the bond wire and the connected pin, bond finger, or shape at each end. This is useful if you are doing round trips with the data (`dxf out` and `dxf in`), and do not want to display the connector vias.
If you are exporting to view the DXF in a third-party tool or are trying to obtain connectivity information to make modifications, you would leave the connector vias in.

Related Topics

- [Wire Bond Settings Dialog Box](#)
- [wirebond select](#)
- [enved](#)
- [wirebond tack point](#)
- [color192](#)

Wire Bond Use Models

This section includes the following use models:

- [Single-Chip Wire Bond Use Model](#)
- [Die-to-Die Wire Bond Use Model](#)
- [Wire Bond Stacked-Die Use Model](#)

For a list of generic design tasks, see Generic Wire Bond Design Tasks in the `wirebond select` command.

Single-Chip Wire Bond Use Model

A single-chip component has a single IC component attached to the package substrate. For wire bond dies, this usually means that a die is mounted on the top side of the substrate with the die pads facing away from the package substrate. For this class of designs, the following sequence is the recommended use model using the Cadence wire bond tools.

It is assumed that you set up your design properly. It is also assumed that you already created the BGA, imported the die, set up the package substrate stack-up and constraint information, and checked the *Allow DRC violations* box in the Wire Bond Settings dialog box.

⚠ By default, during wire bond placement, the tool attempts to satisfy wire-to-wire, finger-to-wire, and finger-to-finger constraint values whether or not these values are currently enabled for online DRC checks. By setting the `WIREBOND_IGNORE_DISABLED_CNS` environment variable in the User Preferences Editor (`enved` command), the constraint is not used to guide placement if it is not enabled for DRC. If the same net or differential pair bond finger-to-bond finger spacing constraint is disabled, the tool uses the general finger-to-finger spacing value between these types of fingers. Finger-to-finger spacing and wire-to-wire spacing are always used to ensure that placement does not introduce unnecessary electrical shorts.

1. Define any bond finger padstacks that you need for this design.
These padstacks are normally based on the component technology requirements and the size of the wire bond die. Generally, a larger pad size allows for a cheaper component.
2. Create and wire bond power and ground rings (`pring wizard` command) for your die as

follows:

- For whole ring creation:
 - Create power and ground rings around the perimeter of the die component.
The rings connect power and ground pins on the die to the correct voltage nets on the component.
 - Wire bond the power and ground pins to the rings.
By default, bonds are placed in the middle of the ring and are spaced apart based on the DRC system's constraint for wire-to-wire spacing.
- For split ring creation:
 - Create power and ground rings around the perimeter of the die component.
 - Using the `wirebond select` command with *Shapes* enabled in the Find Filter, select the ring to split into multiple pieces and choose *Cut* from the pop-up menu.
 - Select the area in which you want the ring cut.
 - Repeat steps a and b until you are satisfied with the way the rings look.
 - While still in the `wirebond select` command, select the ring pieces individually and choose *Assign Nets* from the pop-up menu to change the net assignment for each piece.
 - Add the wire bond connections from pins to the ring.

⚠ When performing wire bond operations such as addition and movement, power and ground ring bonds are placed at minimum clearance to one adjacent wire bond. If you enable the `WIREBOND_CENTER_RING_BONDS` environment variable (`enved` command), the tool runs a post-process step to attempt to center the power and ground wires between the adjacent wires on both sides. This improves manufacturability of the design, but may cause a slight performance decrease during interactive wire bonding.

3. Generate I/O bond fingers.

- a. Establish the guide paths.
- b. For each side of the die, select the pins (either by pin or by net) for which you are adding wire bonds:
 - Choose the type of pattern to create (*Orthogonal, Equal Wire Length, On-Path, Direct Connection, or Free Placement*), and set the necessary parameters for

placement.

The wire bonds appear on your cursor.

- Drag them to the appropriate guide path and snap them down.

Power and ground wire bonds automatically adjust based on the placement of new wire bonds and the specified bubble and shove settings.

If they do not automatically adjust, check that the rings have nets on them.

 By default, the tool respects wire profiles when computing legal bond wire placements. As a result if two wires use different profiles, it allows them to cross within a set distance from the die or pin. To avoid the crossing of wires in your design during pushing and shoving of bond fingers (such as when optimizing the design for optical checking), set the `WIREBOND_IGNORE_WIRE_PROFILES` environment variable in the User Preferences (`envedit` command). Setting the `WIREBOND_IGNORE_WIRE_PROFILES` environment variable causes the tool to treat all wires as part of the same profile for placement only. DRC computations, 3D views and 3d analysis are unaffected by this setting.

4. Manipulate the wire bond pattern to eliminate any DRC errors.

At this point, you may have DRC errors in your pattern. You can correct them by:

- Using narrower bond fingers to allow them to space closer together (more expensive).
- Splitting the bond fingers across multiple guide paths at different distances from the die.
Shrink your spacing constraints, if required and possible.
- Using additional different loop profiles to allow more wire bond crossing.
This still requires multiple guide paths. If bond fingers are on the same guide path, the wire profiles do not allow more crosses.

5. Join bond fingers together for same-net bonds.

In situations where neighboring bond fingers are on the same net, you can combine these together into one bond finger with two connection points. This allows the rest of the pattern to compress.

6. Route the design.

7. Refine the wire bond pattern, using a variety of tasks.

Once you perform routing (or if a die change is required), you may need to tweak the wire bond pattern slightly. Perform this task now, with the routing staying snapped to the bond fingers while they are moved.

8. After you create wire bonds, you may have to adjust the tack point (`wirebond tack point` command), which is the contact point of the bond wire to the bond finger to improve the process of manufacturing or adjust the wire-to-bond finger clearance.
9. Perform an assembly or manufacturing analysis for the wire bonds.
Typically, you perform this task near the end of the process. You perform a set of additional checks to ensure that the wire bond pattern not only meets all the physical and electrical constraints, but also can be created by the wire bond machine. This takes into account such items as the sequence in which the wire bonds are created.
10. Generate wire bond documentation.
This includes attaching bond finger labels or text to the pads, writing DXF output for generating diagrams in AutoCAD, and so on.
11. Export the finalized bond finger placements, guide paths, and settings to the library for reuse in future designs.

Die-to-Die Wire Bond Use Model

When two dies are oriented side-by-side in a design and are spaced closely, you may want to run signals directly between them, without ever connecting to the substrate. This is called die-to-die wire bonding. The wire bond machine runs a wire bond from the passivation opening on one die to the passivation opening on a second, adjacent die.

In the following use model, it is assumed that you already created a BGA, placed both dies, and set up the package substrate stack-up with constraint information.

1. Orient dies and perform pin swap for optimal bonding.
Be sure that the dies are ideally oriented with respect to each other. You can use the ratsnest lines between the two dies for gauging the orientation. If either die is a co-design die or is programmable, perform pin swapping to optimize the assignment between dies.
2. Select the pins from one side of the die, and specify direct connection as your layout style.
Then, configure the profile and diameter settings. Finally, drag your cursor in the direction of the destination pins and click them into place once you set the wire bonds according to your specifications.
Pins of the same nets map together. If no match is found, the tool selects the dummy net pins and automatically updates their net assignment.
3. If wire bond errors occur at either end of the wires:
 - a. Select a thinner wire bond for use.

- b. Move the die pads of one of the dies, if either die is a co-design die.
 - c. Re-orient the dies so that the angle of the wires changes sufficiently to allow the bonding to complete without errors.
4. Follow the steps of the [Single-Chip Wire Bond Use Model](#), treating the side-by-side dies as one large, overall die. You can ignore any nets directly connected between the two dies, with all other nets getting connected down to the package substrate.

Wire Bond Stacked-Die Use Model

With the trend of shrinking the overall size of systems in devices such as cell phones and PDAs, there is more stacking of dies inside packages, for example, a memory chip is mounted on top of an ASIC.

For designs with stacked dies, follow this use model using the Cadence wire bond tools. It is assumed that you created a BGA and completed the package substrate stack-up and constraint information.

1. Establish a layer for each die in the stack.
Typically, these layers have the same properties, except for the layer name. Cadence recommends that you name the layer to match the die using it, thus making the visibility of a die easier as you increase the stack height.
2. Import each die in the die stack as follows:
 - a. If this is not the bottom die in the stack, add a spacer on top of the existing die in the stack, where you want to place the die.
The spacer is a mechanical symbol used for spacing two dies far enough apart that they do not interfere with each other electrically or physically.
 - b. Import the die, placing it on the lowest available layer.
3. Configure the vertical die and spacer properties with the die stack editor, and optimize their relative placements using this command and looking at the ratsnest lines.
4. Optimize pin assignments for co-design dies.
If some or all the dies are co-design dies (or are programmable), you may want to swap pin assignments to further optimize the die-die and die-to-component assignments. This allows for easier wire bonding and better overall routing.
5. Turn off visibility except for the lowest die in the stack to allow easier wire bonding of each die.

6. Follow steps 1 through 5 of the [Single-Chip Wire Bond Use Model](#) to wire bond the lower die in the stack.
7. Turn the visibility of the lowest die off, and turn on visibility for the next die in the stack.
8. If any die-to-die direct bonding occurs at this level (side-by-side dies), create these inter-die wire bonds before adding the wire bonds down to the package substrate fingers. See Steps 2 and 3 in the [Die-to-Die Wire Bond Use Model](#) for details on performing this step.
9. If any direct die-pad to die-pad bonding occurs from this die to the die below, create these inter-die connections according to steps 2 and 3 of the Die-to-Die Wire Bond Use Model.



You need to turn on the visibility of the lower die while performing this step.

10. Repeat steps 3 and 4 of the [Single-Chip Wire Bond Use Model](#) to wire bond this die out to the component.
11. Turn the visibility of the lower die on, and correct any DRC errors that occur between the wire bonds of the two dies.
12. Repeat steps 7 through 10 of this use model, gradually working your way up one die further in the stack at each pass.
13. Follow steps 6-11 of the [Single-Chip Wire Bond Use Model](#).

These steps walk you through component routing and the final optimization of the wire bond pattern based on the substrate routing. It also ensures that the known manufacturing and assembly rules have been met.

Related Topics

- [Setting Up a Wire Bond Design](#)
- [wirebond settings](#)
- [Defining Padstacks for Bond Fingers](#)
- [pring wizard](#)
- [Adding Bond Fingers and Fanout Stubs](#)
- [wirebond select](#)
- [Splitting or Reconnecting a Power/Ground Ring into Segments](#)
- [Assigning Nets to a Ring or Ring Segment](#)

- Adding a Bond Finger Guide Path
- Adjusting Bond Fingers to Meet Minimum DRC Requirements
- Redistributing Bond Fingers Across Multiple Bond Finger Guide Paths
- How the Wire Profile Editor Works
- Bonding Multiple Pins to Common Bond Fingers
- Creating a Merged Finger Shape
- Editing the Bond Finger Guide Path
- Adding Non-Wired Bond Fingers by Selecting a Bond Finger
- Moving Bond Fingers
- Swapping Bond Fingers
- Centering Bond Wires Between Bond Fingers
- Spacing Bond Fingers Evenly
- Adding Routing Channels Between Bond Fingers
- Creating a Ring Segment from a Set of Bond Fingers
- Running the Assembly Rules Checker
- wirebonds reports
- dxf out
- wirebond export
- diestack editor

Routing

Allegro® X Advanced Package Designer (APD) contains a number of features that allow you to route your design in a manner consistent with your design methodology and specific requirements.

Prerequisites to Routing

Before you can route your design, you need to perform certain prerequisites. The following list covers routing operations for ICs and packaging. Specific types of routing may not require all these operations.

- Create a layer cross-section of your design.
- Place die (class IC), component (class IO) and, optionally, plating bar (class PLATING_BAR) components and symbols.
- Assign all signal wire bond pins, I/O pins, and plating bar pins.
- Set the physical constraints.
- Ensure that via structures for the bond fingers exist in the *Vias* column of the Constraint Manager (*Setup – Constraints – Physical*)
- Plan bond finger X or Y locations (orthogonal).
- Plan bond finger-to-bond finger spacing.
- Plan minimum and maximum wire bond length.
- Plan maximum wire bond angle (radial).
- Create power and ground pin distribution for component and die (optional, but recommended).
- Assign nets to pins.
- Assign each die pin a layer on which to route it.

Generating Radial Routes

The Radial Router lets you select a number of pins and pull them out in a fanned pattern to increase the spacing between clines, easing automatic routing of the bond fingers to the component pins for a wire bond component.

A radial fanout pattern for escape routes is necessary if the die pins are closer together than the component pins. You can control both the angle and the length of the first route segments in the radial pattern.

When you select the `radial router` command, you can choose to do the following tasks from the Options window pane:

- Set the subclass on which the radial lines are to be added.
- Set the angle of the radial pattern.
- Set the direction in which the radial pattern will emerge from the pins.
- Set the width of the radial lines.
The default value is the specified minimum width of the active layer.
- Drag radial lines using the angle of the pins.

To generate radial routes automatically, choose *Route – Router – Route Radial* ([radial router](#) command).

Using Custom Smoothing

The `custom smooth` command allows you to optimize, or smooth selected clines or cline segments according to parameters set in the Options window pane. Smoothing the angles of clines or cline segments can minimize the distance to pin connections. Before you start, note that you cannot perform custom smoothing on clines or segments that contain DRC errors. You may need to perform a DRC update and appropriate cleanup before using this feature.

To perform custom smoothing, choose *Route – Custom Smooth* (`custom smooth` command).

Routing Automatically with the Allegro PCB Router

APD provides several methods of interactive and semiautomatic routing of wire bonds, component I/O pins, die-to-die pins, die-to-component pins, component-to-plating bar, and plane connections. The tool also handles staggered, radial, and straight orthogonal bond finger and routing configurations. During wire bond routing, the tool can also generate any desired bond finger pattern.

The tool provides quick, automatic Z-direction routing of component I/O pins to power and ground planes (including multiple connections for each I/O pin) with the Zrouter, which places vias on the grid that you specify and also creates vias smaller than the I/O pin. The via grid gives you several locations on a pin or shape from which to route a via. The Zrouter lets you specify vias routed from a module's I/O pins to specific layers (subclasses of class *CONDUCTOR*) in the module. The Zrouter uses the information on the Zrouter dialog box to route one via, or as many vias as possible, between an I/O pin and a layer. Using `zrouter` or choosing *Route – Zrouter* from the menu, you can instruct the tool to use the via that is a "best fit" to make the proper connection to the proper planes. The "best fit" via transcends only the minimum necessary layers to satisfy its associated connectivity.

Use the semiautomatic radial router to achieve die-to-component route fanout patterns. You can use what-if scenarios to quickly optimize fanout angle for routing from the die pins to the I/O pads. Once you establish the pattern, the tool can automatically finish the task with "any angle" routing.

Die-to-die routing is usually complex. Using APD, you can interactively route critical nets (such as clock lines) first, analyze them for signal integrity, adjust as required, and then "fix" them in place so that they cannot be affected by automatic routers. You can do the remainder of the die-to-die routing interactively or automatically.

The Automatic Router dialog box lets you set the routing parameters and run routes. The interface supports *Route – Router – Route Automatic* ([auto_route command](#)) and *Route – Router – Route By Pick* ([route_by_pick command](#)).

-  *Route – Router – Route Automatic* and *Route – Router – Route By Pick* do not automatically protect existing conductor when routing. To protect existing conductor when routing, you must apply the **FIXED** property to any net that you do not want modified by subsequent routing passes.

If you run *Route – Router – Interactive Editor* ([specctra command](#)), or *File – Export – Router* ([specctra_out command](#)), any existing conductor is protected.

For comprehensive information regarding the Package Router, refer to the appropriate online help and manuals. For additional information on running automatic routing, see the router-related chapters in this user guide.

Routing Differential Pairs and Buses

The use of differential pairs and buses in component designs are accounted for during component routing; specifically:

- Automatic Net Assignment
- Layer Assignment
- Route Feasibility

These features act on edge-side differential pairs in your designs. (Route Feasibility does not route broadside differential pairs.) This section describes how these features operate on differential pairs and buses. It does not attempt to provide a comprehensive overview of differential pairs or buses, nor does it provide detailed procedures for assigning differential pairs in designs. For these and other guidelines for working with differential pairs, see the following documents:

- *Best Practices: Working with Differential Pairs*
- The [diff pairs](#) topic in the *Allegro PCB and Package Physical Layout Command Reference*
- [Interactive Routing for Differential Pairs](#)

When automatically assigning nets composed of differential pair or buses, the *Logic – Auto Assign Net* command provides a router-based algorithm that uses the parameters that you have previously established (constraint definitions, layer assignments, pre-routes, and so on) to calculate the best solution for routing your design. By determining the sequence and layers on which routing needs to occur to effect a successful result, the *Logic – Auto Assign Net* command:

- Sorts all pins that represent differential pairs and buses in a continuous set.
- Adjusts pin ordering to compensate for any required layer swapping.
- Places pre-existing assignments and routes.
- Completes assignment of remaining free balls.

In cases where conflicts remain unresolved, you may need to make manual changes to effect a satisfactory conclusion. For example, if an insufficient number of unassigned destination pins prohibits placing all the differential pairs in your design, you might need to:

- Change the layer mapping of nets, balls, or pins to create additional unassigned destination pins.
- Increase the size of your BGA to expand the destination pin set.
- Modify power and ground pin assignments in the destination pin set to make available more destination pins.

Interactive Routing

The layout editor provides several interactive routing options for producing manufacturable results on the following types of designs:

- Analog
- Digital
- Surface-mount
- Through-hole
- High-speed
- Multichip module
- Hybrid

Interactive routing lets you do the following:

- Connect two points, with and without vias
- Start a connection from a Rat T point
- Edit vertices
- Insert blind, buried, and through-hole vias
- Shove vias when adding clines
- Add, slide, or delete connections and vias
- Begin a connection, then run automatic routing to finish the etch for that pin-pair
- Route timing-sensitive circuits
- Route with layer-set constraints
- Centre the route between pins or vias

Interactive routing features complement automatic routing features. You can use interactive routing to:

- Complete critical nets before automatic routing.

- Finish disconnects left after automatic routing.

 Interactive routing is not influenced by manufacturing-based checks.

General Routing Prerequisites

The following table shows general prerequisites, recommendations, and optional considerations for interactive routing. The table also provides links to the areas of the user guide that detail each process.

Prerequisites for Interactive Routing

Prerequisite	Interactive Routing	For Details, See...
Define nets in netlist (for PCB Editor)	Required unless routing in a package symbol	Writing a Netlist
Schedule nets in netlist (for PCB Editor)	If needed	Scheduling a Net
Load netlist (for PCB Editor)	Partial netlist for intelligent conductor optional, but not required	Creating a Database
Place components (for PCB Editor)	Partial placement, reflecting netlist, required	Placing Elements Manually
Define etch/conductor subclasses	As needed	How Etch Shapes Affect Routing
Define etch/conductor width	Required	Defining Line Width
Define route grid	As needed	Specifying Grids
Verify layer information	Recommended	Working with Cross Section Layers
Define blind and buried vias	As needed	Defining Blind and Buried Vias
Define route and via keepouts	As needed	Keepin and Keepout Areas
Define spacing and physical constraint sets for design rule checking	As needed	Working with Constraints
Define electrical constraint sets	As needed	Working with Constraints
Define via padstacks	As needed	Layout Padstacks, Vias, and Etch/Conductor Shapes

Dynamic Etch Editing

When you route connections interactively, you get immediate, real-time feedback by way of dynamic WYSIWYG etch editing. You can see the results of any changes you make when adding or editing etch. Components of dynamic etch editing are integrated in the *Route – Connect* (`add connect` command), *Route – Slide* (`slide` command), and *Edit – Vertex* (`vertex` command) menu items. They include:

- Bubbling in hug- or shove-preferred modes
- Smoothing connect lines
- In-compliance or out-of-compliance feedback on timing constraints
- Dynamic DRC
- Full visibility of elements on dimmed or invisible layers during etch editing operations

All these features let you complete interactive routing tasks quickly and successfully.

The following issues are related to some components of dynamic etch editing:

- Bubble functionality may not perform optimally on arc segments or line segments that need to bubble arcs.
- Dynamic DRC does not support electrical DRC violations.
- In some instances, use of smoothing in bubble mode could result in one cline causing another to bubble.
- Full smoothing on long non-orthogonal segments may result in slower performance when in `vertex` mode.
- Based on your hardware, performance degradation may occur on very dense designs as a result of WYSIWYG functionality.

Related Topics

- [add connect](#)
- [slide](#)
- [vertex](#)

Setting Visibility during Interactive Routing

You can set the visibility (highlighting, dimming, and so on) of elements and layers to assist you while manually editing etch/conductor and other design elements. You can set these functions statically using commands for highlighting nets and viewing ratsnests or dynamically through the use of *Shadow mode* in the Color dialog box, and a user preference.

Critical signal traces must travel over an uninterrupted copper plane to insure a continuous loop of return path current. Even traces that overlap pin voids can be subject to signal disruption. Identifying and locating all of these void overlaps is crucial to assuring the integrity of the signals in question.

The **Highlight Segments Over Voids** feature (`highlight sov` command) locates and highlights the segments of nets where signals overlap voids. The pin where the overlap occurs is also highlighted. This feature streamlines your ability to eliminate potential signal integrity problems during the clean-up phase of your design.

Typically, you use this feature near the end of the design process, after the routing has been completed and all other constraints have been met.

How Highlight Segments over Voids Works

All segments of the selected nets that overlap voids are highlighted, as well as the offending void. Only adjacent voltage planes are considered. The **Highlight Segments Over Voids** feature works across the entire board.

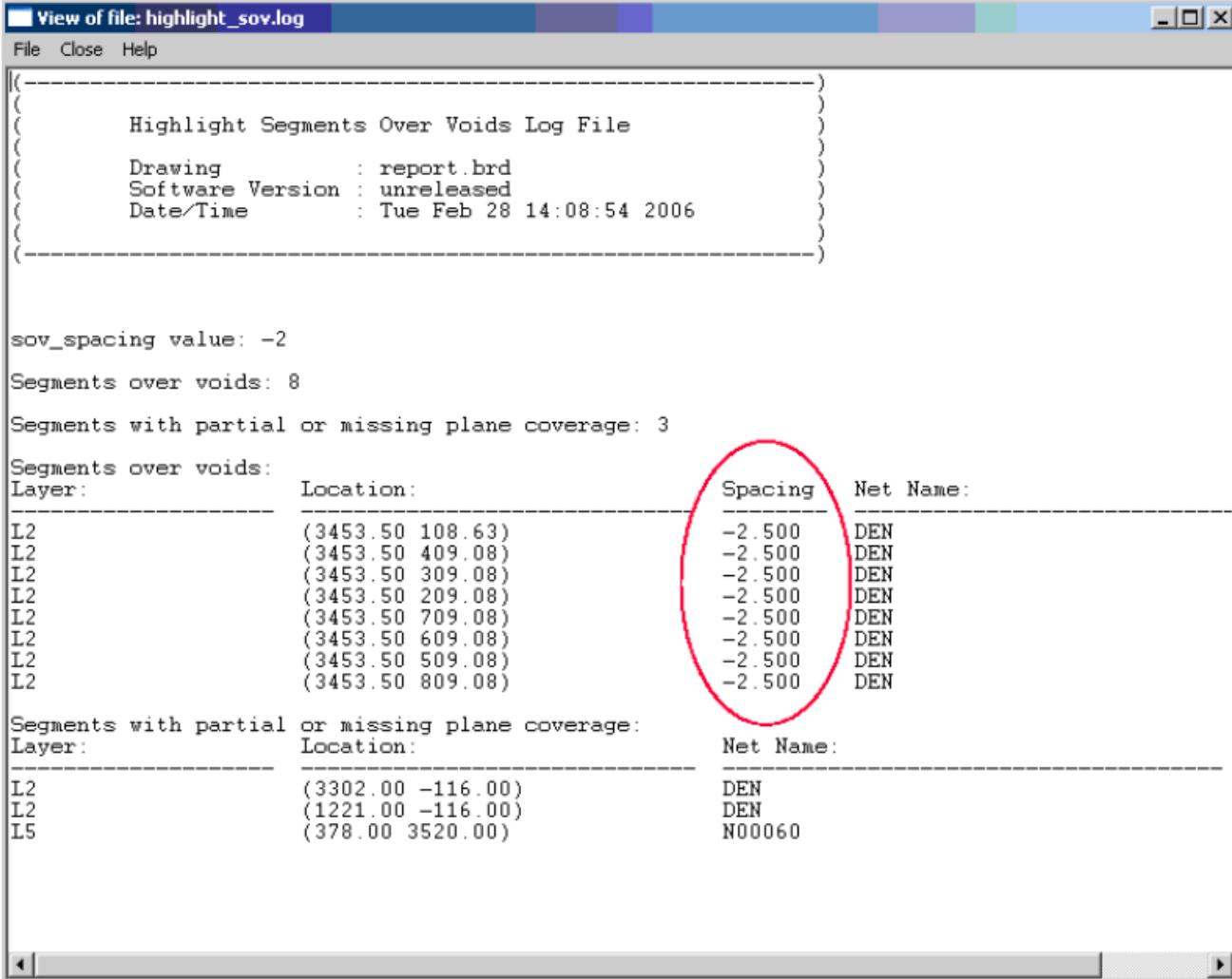
-  In the case of cross-hatched planes, the **Highlight Segments Over Voids** feature highlights only those clines that cross the boundaries of the planes. Such violations for cross-hatched planes appear in the report file under the section *Segments with partial or missing plane coverage*.

Highlight Segments Over Voids checks all nets except those with the **VOLTAGE** property. You can limit the check to particular nets by assigning the **SOV_CHECK** property to one or more nets. If a net is assigned both the **SOV_CHECK** property and the **VOLTAGE** property, the **VOLTAGE** property takes precedence and the net is not checked.

A report file containing the violations is generated. The report file appears in the Viewlog dialog box, which allows you to click on the X,Y coordinates of a violation and center the design window on that object.

The report file shows the spacing of the overlap for each violation. You can scan the list quickly, looking at the spacing values, and determine if there is a common number that would be considered

acceptable (for example, 0.2 mils). You can then set the `sov_spacing` value to offset that (`sov_spacing = -0.2`) and rerun highlight sov to see where the really significant overlaps occur. This process eliminates many "false" violations and helps you choose a reasonable value for `sov_spacing` without a lot of guesswork. By repeating this process incrementally, you can arrive at the smallest acceptable value based on what you have designed.



View of file: highlight Sov.log

File Close Help

```
(-----)
(
  Highlight Segments Over Voids Log File
(
  Drawing      : report.brd
  Software Version : unreleased
  Date/Time    : Tue Feb 28 14:08:54 2006
(
(-----)
```

sov_spacing value: -2

Segments over voids: 8

Segments with partial or missing plane coverage: 3

Segments over voids:	Layer:	Location:	Spacing	Net Name:
	L2	(3453.50 108.63)	-2.500	DEN
	L2	(3453.50 409.08)	-2.500	DEN
	L2	(3453.50 309.08)	-2.500	DEN
	L2	(3453.50 209.08)	-2.500	DEN
	L2	(3453.50 709.08)	-2.500	DEN
	L2	(3453.50 609.08)	-2.500	DEN
	L2	(3453.50 509.08)	-2.500	DEN
	L2	(3453.50 809.08)	-2.500	DEN

Segments with partial or missing plane coverage:

Segments with partial or missing plane coverage:	Layer:	Location:	Net Name:
	L2	(3302.00 -116.00)	DEN
	L2	(1221.00 -116.00)	DEN
	L5	(378.00 3520.00)	N00060

Setting User Preference Variables

You can set the following user preference variables to control how Highlight Segments Over Voids operates:

sov_spacing	Allows you to specify the minimum space that must exist between void and a cline segment. This will suppress highlighting anything less than or equal to that value if it is acceptable to have a small part of a cline exposed by an antipad but the majority covered. You can use this variable to eliminate any "false" failures and highlight only the most troublesome conditions. The value for <code>sov_spacing</code> is expressed in the same units that the design uses (mils, mm, inch, micron). The default value is 0. A positive value means there must be space between the void and the cline segment. A negative value means the segment is allowed to overlap the void by that amount. A value of 0 allows the segment to touch a void without overlapping.
sov_active	If enabled, only the active layer is checked. If you disable <code>sov_active</code> , the Highlight Segments Over Voids checks all conductive layers that have at least one adjacent plane.
sov_skip_plane_check	If enabled, the Highlight Segments Over Voids skips checking of partial/missing plane coverage related violations. If you disable <code>sov_skip_plane_check</code> , the Highlight Segments Over Voids checks plane coverage.

You set these variables in the *User Preferences Editor* (from the *Setup* menu, choose *User Preferences*), under the category *Display_SOV*.

About Bubble Mode

You can choose hug- or shove-preferred bubble modes to avoid spacing DRC errors when adding new etch using *Route – Connect* (`add connect` command) or modifying etch/conductor using *Route – Slide* (`slide` command), or *Edit – Vertex* (`vertex` command) during interactive routing.

When the layout editor:

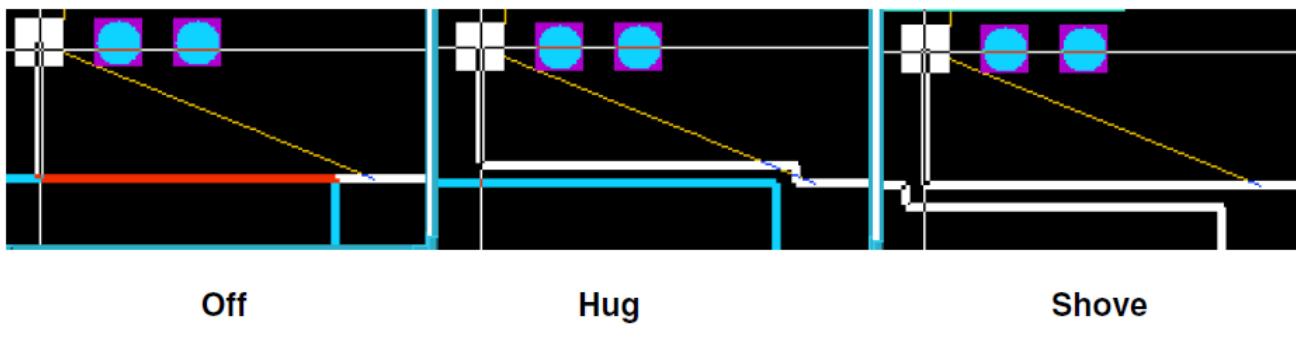
- In hug-preferred mode, encounters an obstacle, it tries to maintain the geometry of the existing etch/conductor to avoid spacing violations. If it cannot successfully hug existing etch with hug-preferred mode enabled, then it tries to shove clines to avoid spacing violations.
- In shove-preferred mode, encounters an obstacle, it tries to maintain the geometry of the new etch/conductor. If it cannot successfully shove existing etch due to obstacles with shove-preferred mode enabled, then it tries to hug obstacles to avoid spacing violations.

See [Obeying Line Angle Controls](#).

Hug and Shove-Preferred Modes When Adding Clines

The following figure shows the differences between hug- and shove-preferred bubble modes when adding clines.

Hug and Shove-Preferred Modes



- *Off*: Shows the original path of the new etch without bubble mode enabled. The new etch/conductor runs on top of the bottom cline, causing DRC errors.
- *Hug-preferred bubble mode*: New etch hugs the bottom cline. The bottom cline retains its original position.
- *Shove-preferred bubble mode*: New etch moves the bottom cline.

Handling DRC Errors

Using *Route – Connect* (`add connect` command) and *Route – Slide* (`slide` command) you can dynamically bubble to fix spacing violations when in hug- or shove-preferred bubble mode. If the layout editor, in the mode, cannot fix some violations, it changes the cursor to the DRC marker shape and highlights segments that have spacing violations as you move the mouse. The layout editor dynamically highlights cline segments affected by DRC errors using the temporary highlight color you defined in the Color dialog box, accessed using *Display – Color/Visibility* (`color192` command).

If the layout editor, in bubble mode, fails to fix DRC errors, takes undesired paths to fix them, or requires too much time to arrive at a solution, you can make intermediate picks to guide more specifically through obstacles. If you disable *Allow DRCs* in the *Options* tab when using *Route – Slide* (`slide` command), the extent of the slide is limited to prevent you from creating spacing violations.

Shove- and hug-preferred bubble modes conform to spacing checks and may consequently cause electrical constraint-set DRC errors, which are not reported until you make the next pick.

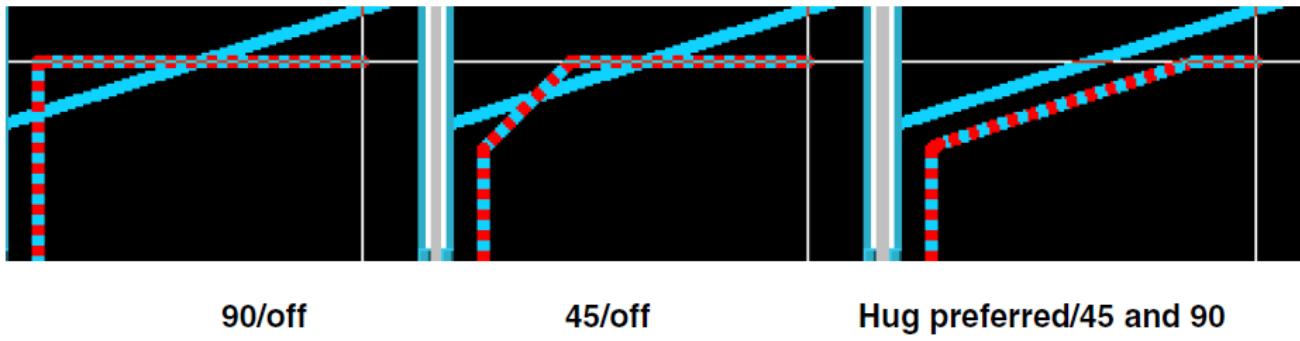
Obeying Line Angle Controls

The initial path for new etch/conductor follows the *Line lock* angle you specified. However, when you enable bubble mode, new or existing etch hugs or shoves other clines to avoid DRC errors, even if it violates the specified angle.

Odd Angle Lines

If the original design contains odd angle lines, bubbling may create more odd angle lines, as follows, even if the *Line lock* value is *90*.

Odd Angle Lines



90/off 45/off Hug preferred/45 and 90

- (Left) A line with a *Line lock* setting of 90 and bubble disabled
- (Center) A line with a *Line lock* setting of 45 and bubble disabled
- (Right) Hug-preferred bubble mode enabled, hugging an odd angle line, regardless of the 45- or 90-degree angle setting

45-Degree Lines

If the original design contains 45-degree angle lines, they layout editor, in bubble mode, may create more 45-degree angle lines (following figure) even if you set the angle to 90 degrees.

Hugging the outside of a 45-degree turn may result in 45-degree segments that exceed the specified value in the Max 45 length field, as follows.

45-Degree Angle with and without Hug-Preferred Mode



- (Left) The specified maximum 45 length obeyed.
- (Right) If the shortest cline was added at the maximum 45-degree length, the layout editor, in hug-preferred mode, adds other segments at greater than maximum 45 length.

Pads

With hug- and shove-preferred bubble mode, circular objects such as pads do not produce circular etch/conductor, as follows.

Handling Pads



- (Left) When the *Line lock* angle is 90 degrees, orthogonal routes handle circular objects.
- (Right) When the *Line lock* angle is 45 degrees or *off*, the layout editor uses 45-degree lines to bubble around circular objects.

Arcs

You cannot add arcs while in shove- or hug-preferred mode. the layout editor does not support hugging or shoving of existing arcs. Consequently, specifying a value of *Arc* in the *Line lock* field disables bubble mode. Conversely, enabling bubble mode to either hug- or shove-preferred sets the *Line lock* value to *Line*.

Constraint Areas

If a line crosses a constraint-area boundary, more than one line-to-line spacing rule may apply to the cline. This could result in bubble using the wrong spacing rule for the line. To avoid this issue, pick just inside the constraint area and then continue routing.

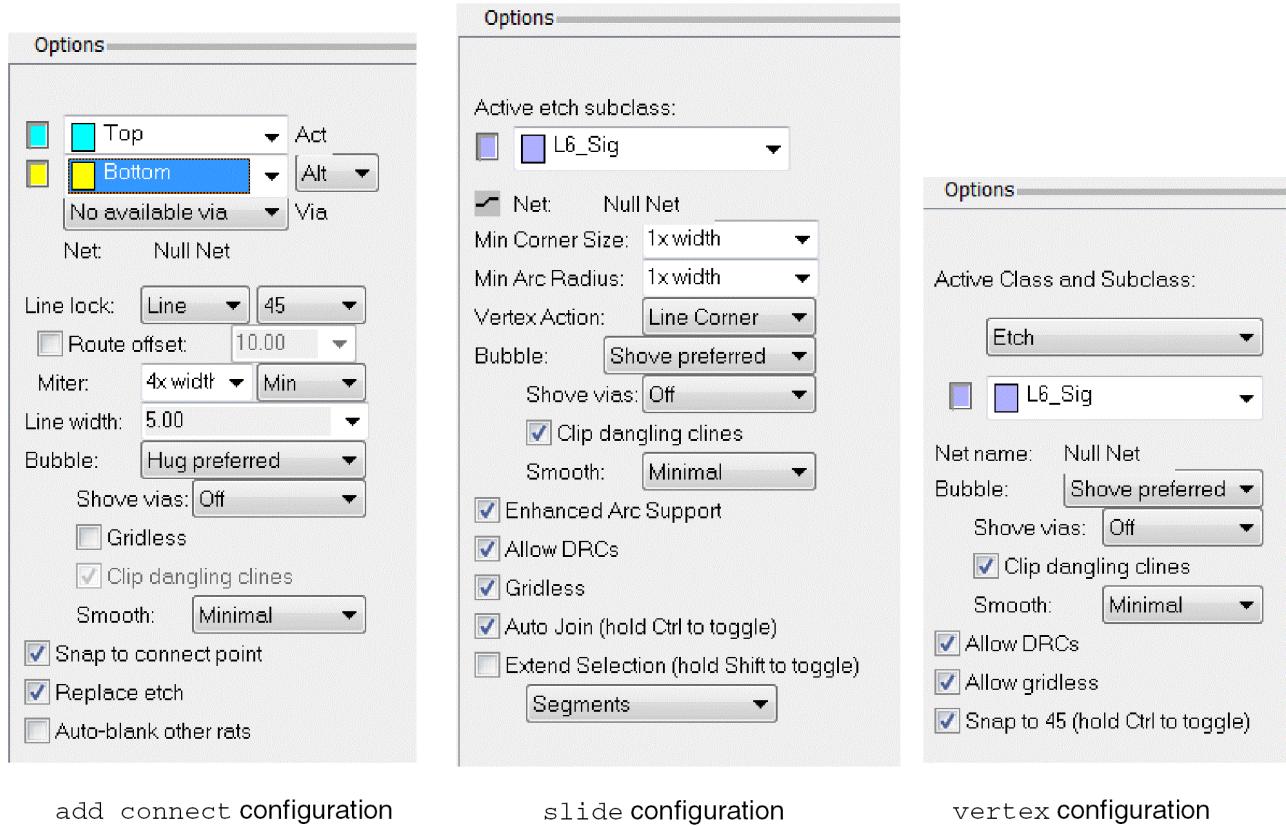
Via Shoving

When you are in bubble mode, you can shove vias when adding or sliding connections or editing vertices. This functionality lets you complete a connection without DRC violations.

You set via shoving in the Options tab when you choose *Route – Connect* (`add connect` command), *Route – Slide* (`slide` command), or *Edit – Vertex* (`vertex` command), as shown in the following figure. Note that in each case, *Shove vias* is disabled when *Bubble mode* is

inactive.

Shove Via Settings in Options Tab



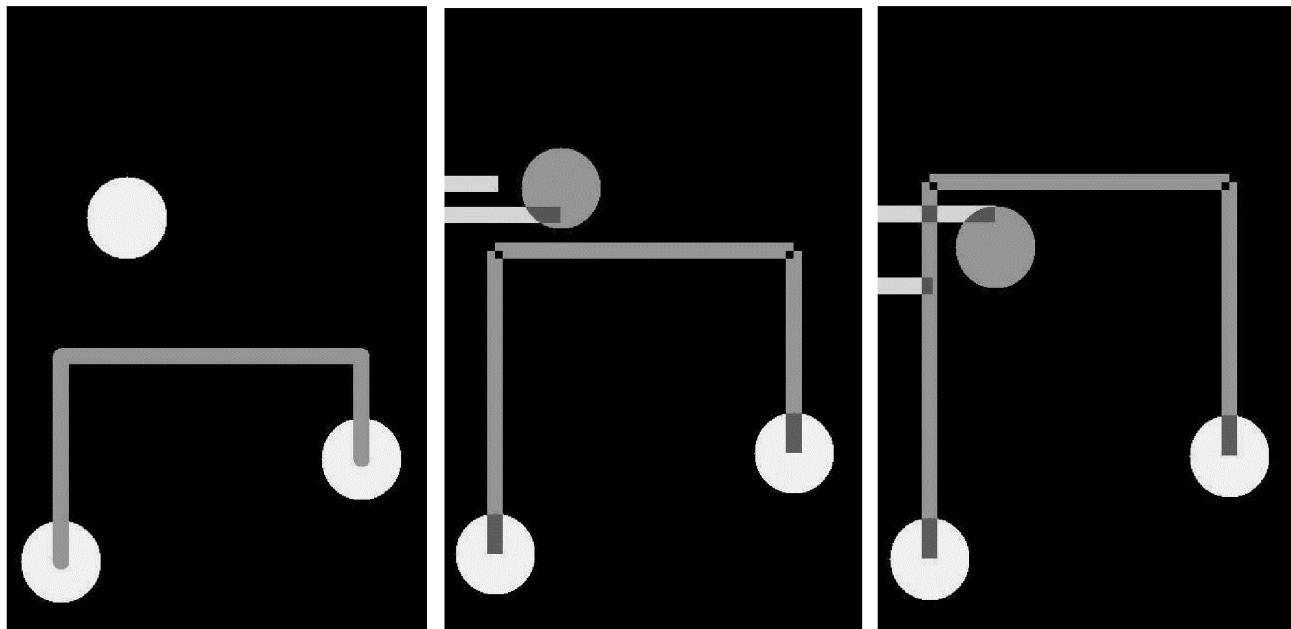
You can set via shoving to operate in three modes. The following table shows how the *Shove vias* modes work with the *Bubble* modes.

Shove Via Modes

Bubble	Shove Vias	
	Minimal	Full
Hug-Preferred Mode	Clines hug the vias unless there is no room, then shoving occurs.	Clines hug the vias unless there is no room, then shoving occurs.
Hug-Only	Clines hug the vias. Other etch/conductor remains the same.	Clines hug the vias. Other etch remains the same.
Shove-Preferred Mode	Clines hug the vias unless there is no room, then shoving occurs.	Vias are shoved. If a via cannot be shoved, the layout editor goes around it.

Regardless of the *Shove via* mode you choose, the behavior of vias when you shove them is identical. A shoved via moves the minimum distance from an element acting upon it. To accommodate this behavior, a via may sometimes hop over a connect line to ensure the least amount of disruption to the overall design. This behavior is shown as follows.

Via Hopping



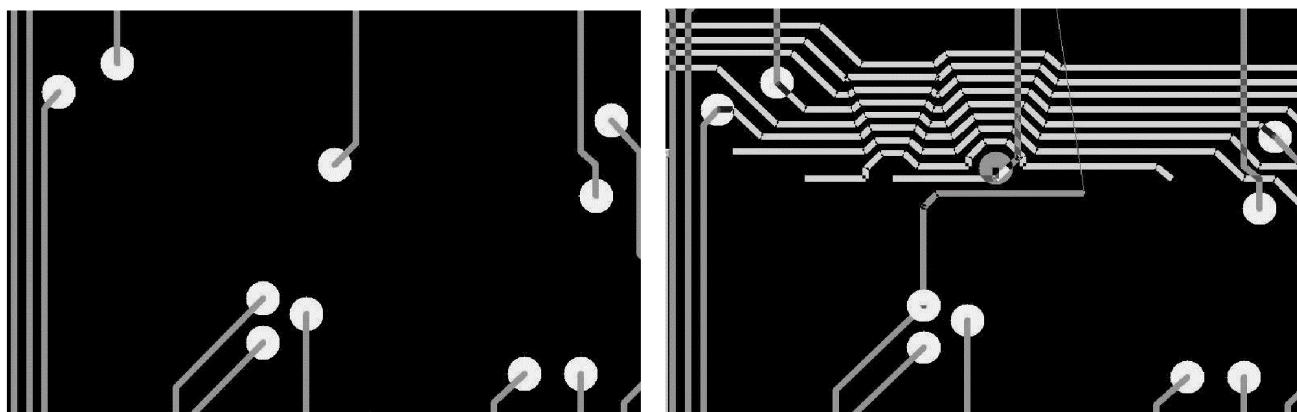
Bond Finger Vias (for APD)

You may not want to implement bond finger via shoving. For this reason, the default mode for via shoving in APD is that via shoving is disabled. Use the `BUBBLE_SHOVE_BONDPADS` environment variable to turn it on. Choose *Setup - User Preferences* (`enved` command) to set this variable.

Display Configurations for Via Shoving

Like all other aspects of etch editing, via shoving appears in WYSIWYG mode. Because vias typically exist on many layers of a design, elements on layers other than the active (visible) layer become visible when acted upon by a shoved via. The following figure shows how a design in non-shadow mode displays etch/conductor when acted upon by a shoved via.

Shoved Via Display Characteristics



In the left design no vias have been shoved. The design displays the vias and connect lines on the active layer. The right design shows when the via at the center of the picture has been shoved upward by the addition of a newly added connect line. All the affected traces on the hitherto invisible layer become visible.

The previous figure shows the default display behavior. You can modify the display of this information by configuring which layer is active and which layers are visible and by setting shadow mode and the `bubble_no_display_invisible` environment variable.

Using the slide Command in Bubble Mode

When you modify existing etch during interactive routing, you can use the two bubble mode options—hug- or shove-preferred—to assist you in avoiding DRC errors. Use hug-preferred mode when you want to model the etch/conductor you are sliding to other etch/conductor. Use shove-preferred mode when you want the etch you are sliding to move other etch out of its path.

Using the add_connect Command in Bubble Mode

When you add new etch during interactive routing, you can use hug- or shove-preferred or hug-only bubble mode to avoid creating DRC errors. Use hug-preferred mode to place routes as closely as possible to existing routes. Use hug-only mode to contour without changing other etch objects. Use shove-preferred mode to route through a path blocked by other clines. To ensure that existing critical clines meeting your requirements remain unchanged, choose *Edit – Properties* (property edit command) or use the toolbar icon to attach the FIXED property to these clines after you complete the route.

Guidelines

Hug and shove-preferred bubble modes are not auto-routing modes. When in these modes, the layout editor does not always locate the optimal path from one point to another. It starts with the line lock-based segments that go from the last pick to the cursor. It then attempts to correct any spacing violations by hugging or shoving etch. Consequently, as you use the shove-preferred and hug modes, the *Line Lock* and toggle settings are critical. You can access the toggle options from the pop-up menu.

Note that setting the *Line Lock* field to *Arc* disables bubble mode. Conversely, enabling bubble mode (to either hug-preferred or shove-preferred mode) sets the *Line Lock* field to *Line* to prevent adding arcs while in shove/hug mode. Shoving or hugging existing arcs is not supported.

During hug or shove-preferred routing, excessive delays typically indicate the result may be undesirable. To maximize performance when you use bubble modes during etch editing:

- Avoid particularly dense or complicated or both types of regions.
- Do not shove many traces simultaneously.
- Minimize distances between selections to preclude crossing constraint-area boundaries.

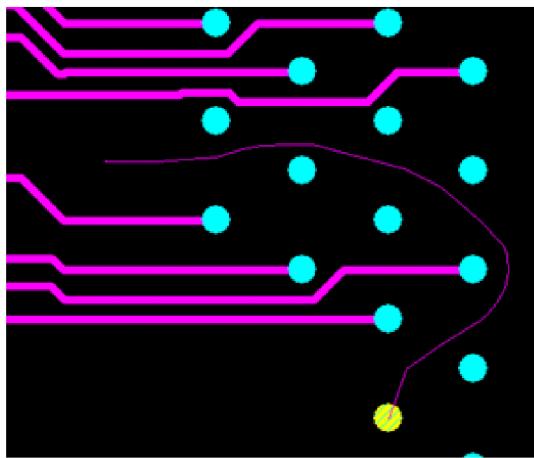
If you do not like the result, you can try again by adding smaller amounts of etch with each pick.

Related Topics

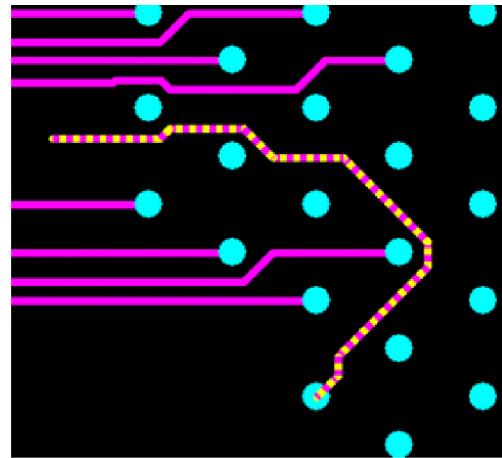
- [add connect](#)
- [slide](#)
- [vertex](#)
- [color192](#)
- [vertex](#)
- [enved](#)
- [Setting Visibility During Interactive Routing](#)
- [property edit](#)

About Scribble Mode

Scribble mode is designed to route in complex routing areas using `add_connect` command. The scribble routing mode lets you to scribble a route path between two points using smart shove and push techniques. Once the routing is completed the etch solution is generated by the application for the scribbled path.

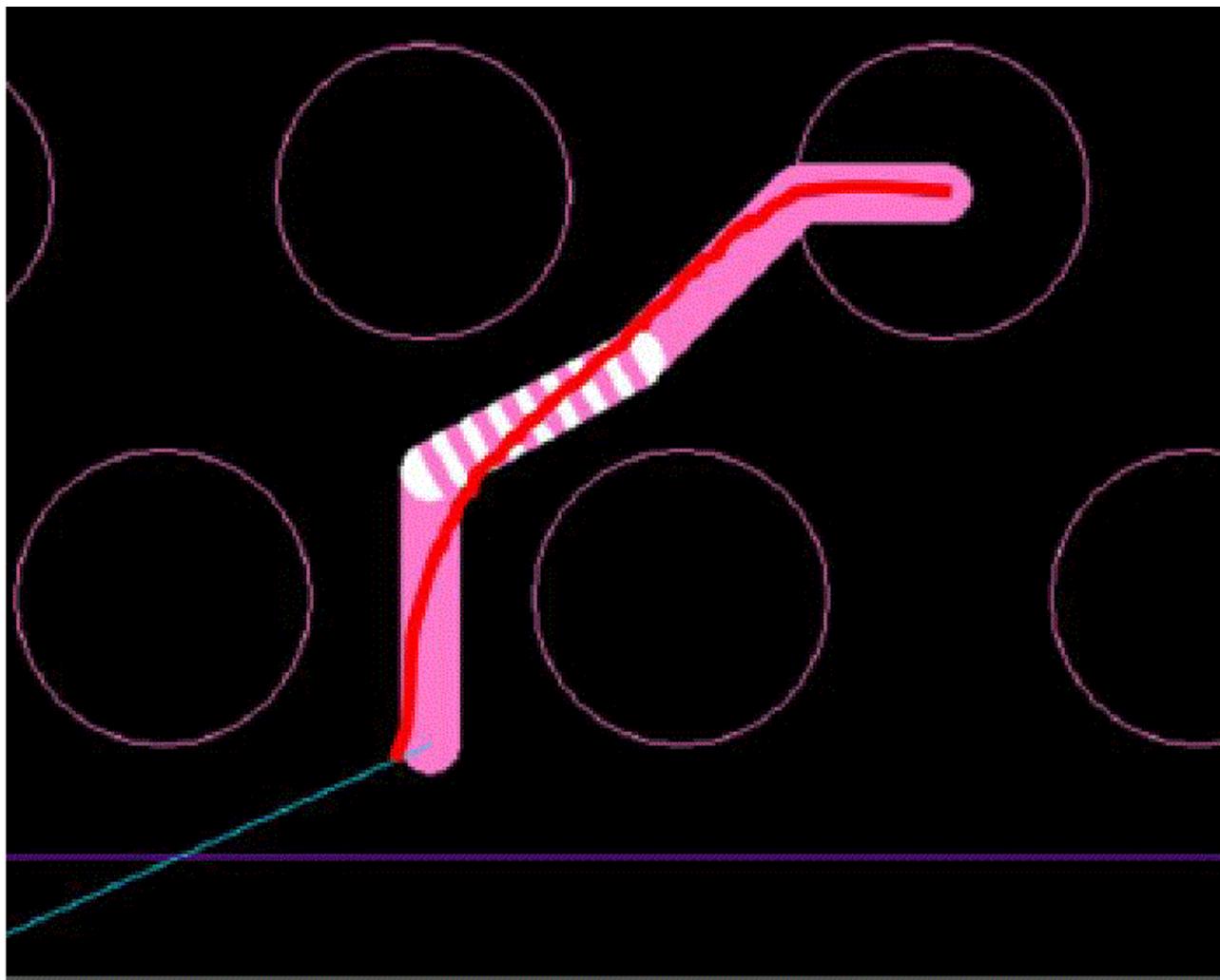


Scribble Path



Etch Solution for Scribble Path

Scribble mode creates an off-angle path through pinch point, keeping the rest of the route path at 45 degree.

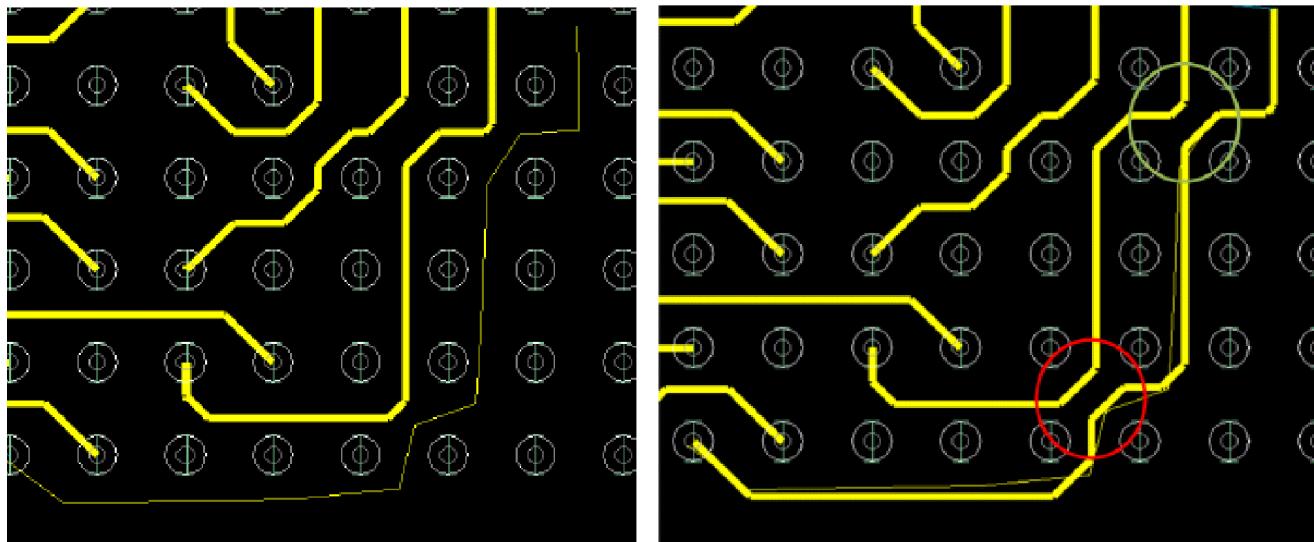


Off-angle Scribble routing

⚠ The off-angle direction is determined based on clicks and pad entry information. The maximum length for a connection is determined based on pad entry direction and a snap band is determined.

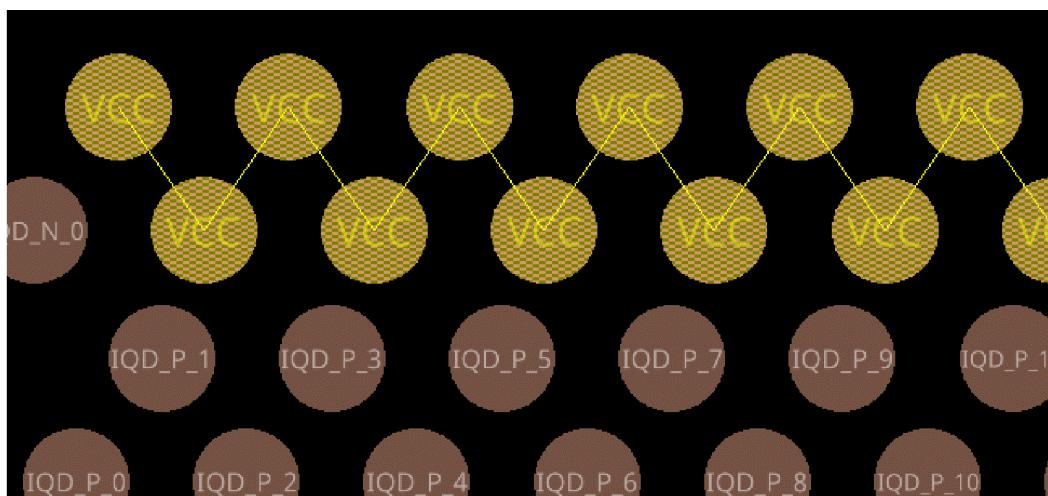
It moves the existing cline routes only when necessary, to avoid a DRC. In the following figure, scribble shoves existing clines at few places, whereas the etch solution remains in the same channel defined by the scribble path.

Scribble Solution: Localized Pushing



Use TAB key to toggle between scribble mode and normal routing mode.

Use scribble mode when routing inside a pin field for creating power or ground connections. It recognizes the same net pins and snaps to the center of the pin allow you to continue your routing path without making several clicks. You can guide the cursor through a pin field to route a single connection and do not need to select multiple points.



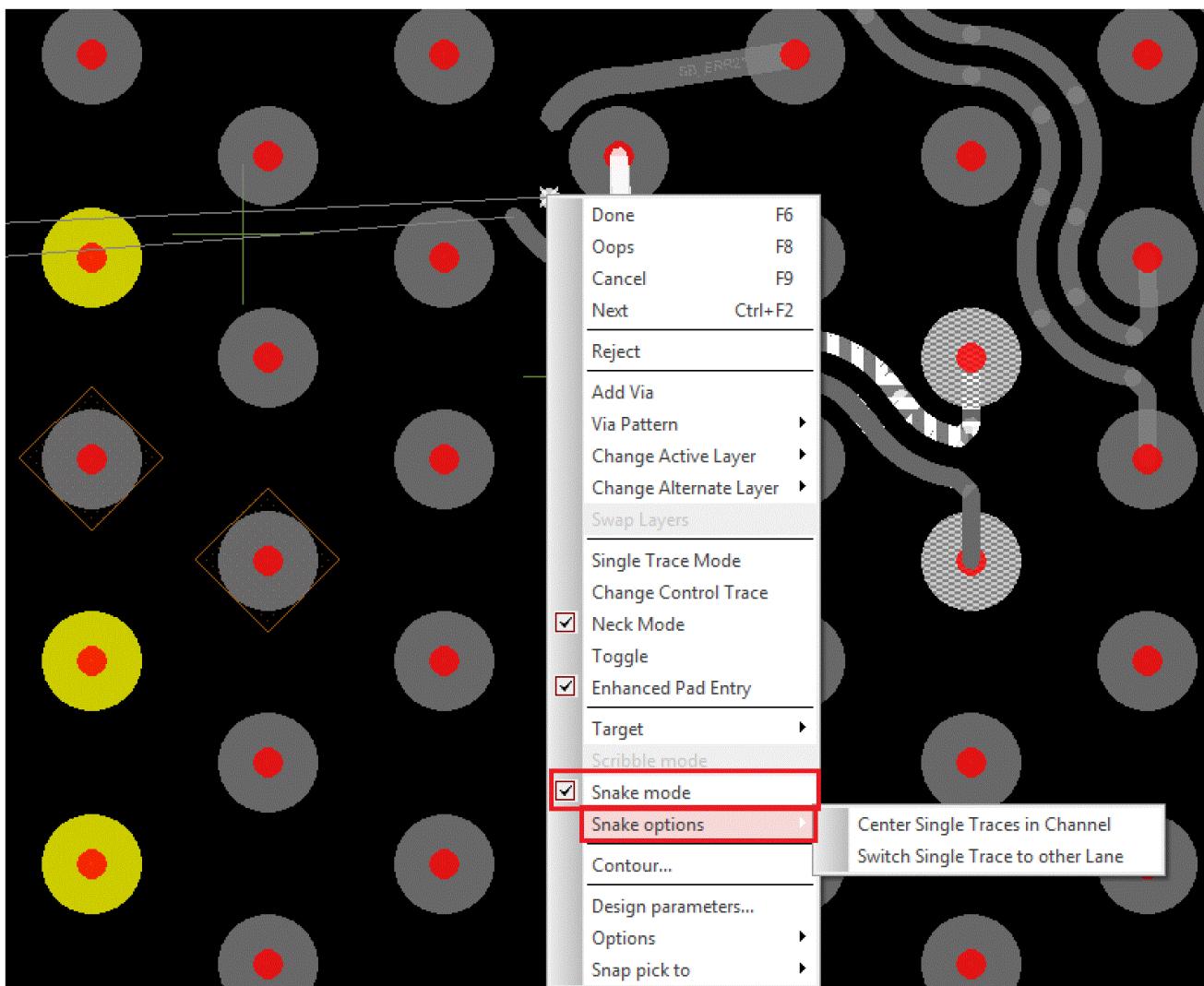
To avoid conflict during interactive routing following menu options are disabled when using scribble mode:

- Clip dangling lines

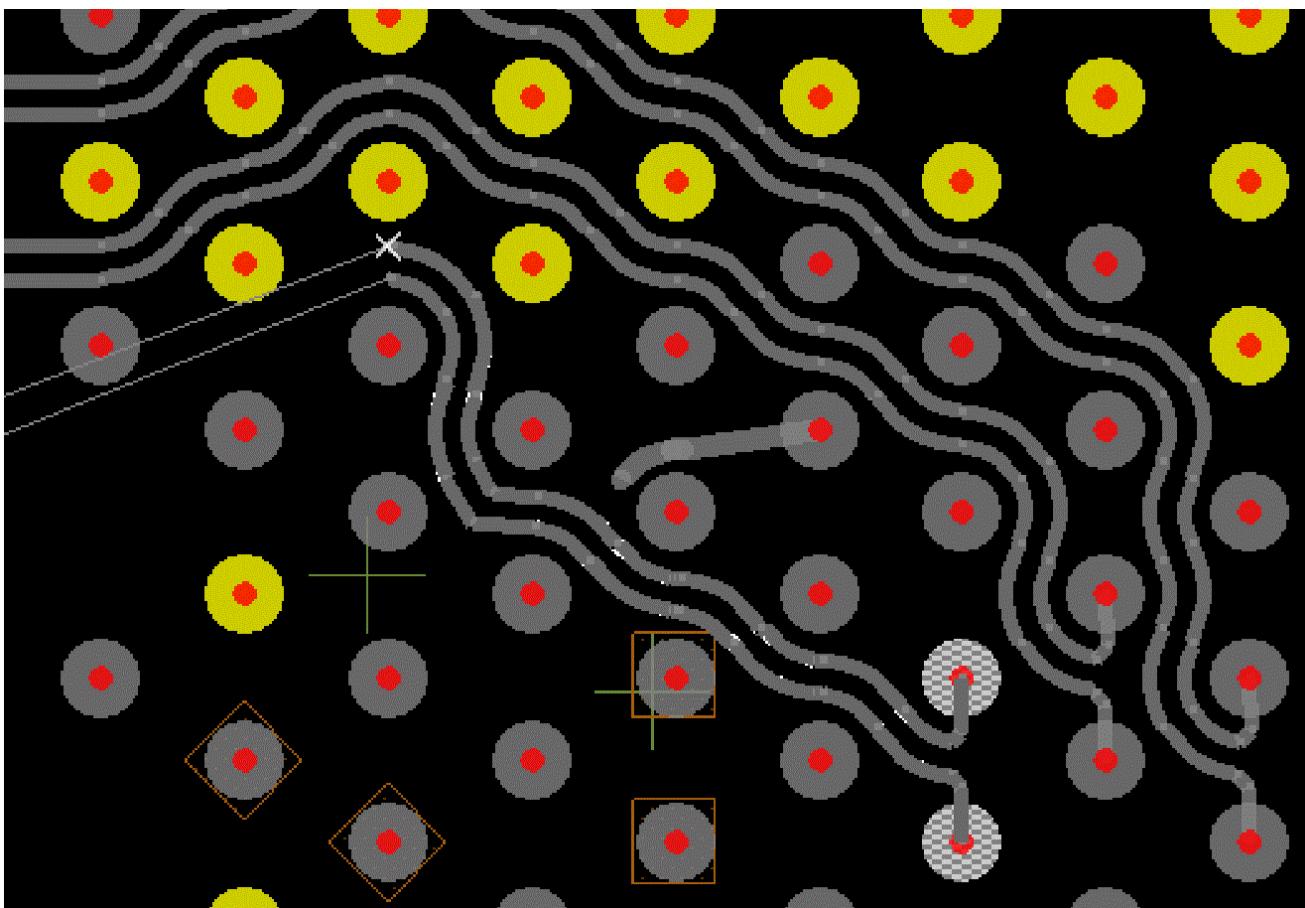
- Route offset

About Snake Mode

The Snake mode is available in the context-sensitive menu of add connect command. In a pin/via field pattern where pins make equilateral triangle, the Snake mode routing creates arcs. You can route a single or double traces within a channel between pins/vias.



Using Snake mode, you can primarily route differential pairs within a channel. You can also route partially completed lines, dangling lines, and single traces. To change the direction within a channel, click left-mouse button and continue to route.



Snake traces are centered between the channels. A single trace can be routed in either left lane or right lane of the channel. Two options are available for routing a single trace.

- Center Single Traces in Channel: Routes a trace in the center of a channel.
- Switch Single Trace to other Lane: Changes the lane for a single trace from left to right or vice-versa within a channel.

⚠ The *Bubble* option does not work if *Snake mode* is enabled.

Adding Connections to a Design Interactively

Choose *Route – Connect* ([add connect](#) command) to add connections to a design interactively. You can add etch/conductor interactively before or after automatic routing. Before adding connections, you may want to familiarize yourself with some aspects of interactive routing:

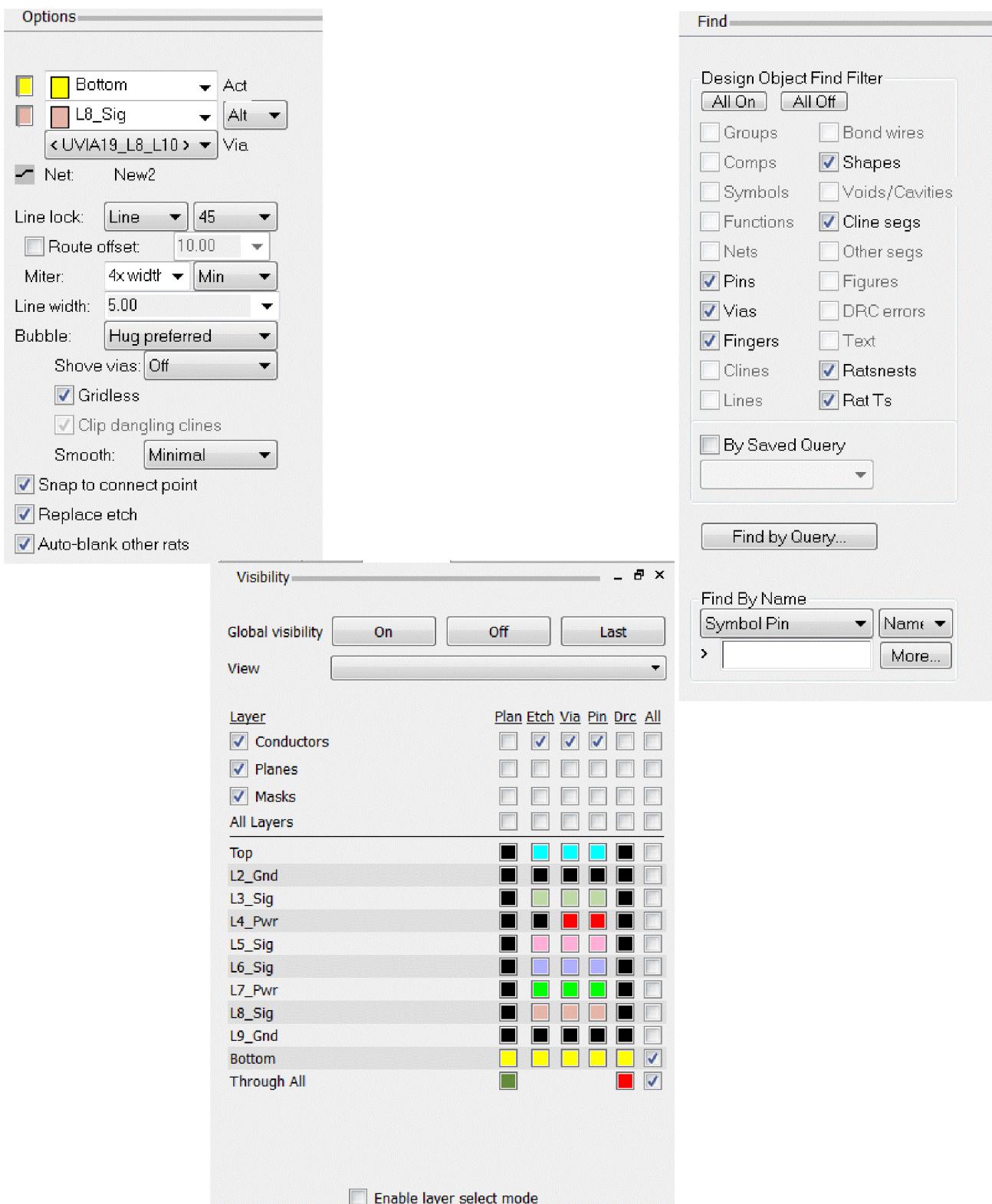
- [Setting Visibility During Interactive Routing](#)
- [About Bubble Mode](#)

For information related to time-sensitive connections, see [Routing High-Speed Circuits](#).

When you add connections using *Route – Connect* ([add connect](#) command), the layout editor displays the following tabs.

Routing the Design

Interactive Routing--Adding Connections to a Design Interactively



- The *Options* tab lets you choose various routing controls.

- The Find Filter lets you identify and choose pins, vias, and etch segments for routing.
- The *Visibility* tab controls visibility of those elements as well as selection of visibility settings.

Routing with Enhanced Pad Entry

The *Enhanced Pad Entry* option is introduced to avoid the manufacturing issues such as acid traps in high-density multi-layer PCBs. This option is available in the right-click menu of `add_connect` and `slide` commands.

Done F6
Oops F8
Cancel F9

Select by Polygon

Select by Lasso

Select on Path

Temp Group

Reject

Change Active Layer ▶

Change Alternate Layer ▶

Swap Layers

Multi-Line Route...



Enhanced Pad Entry

Design parameters...

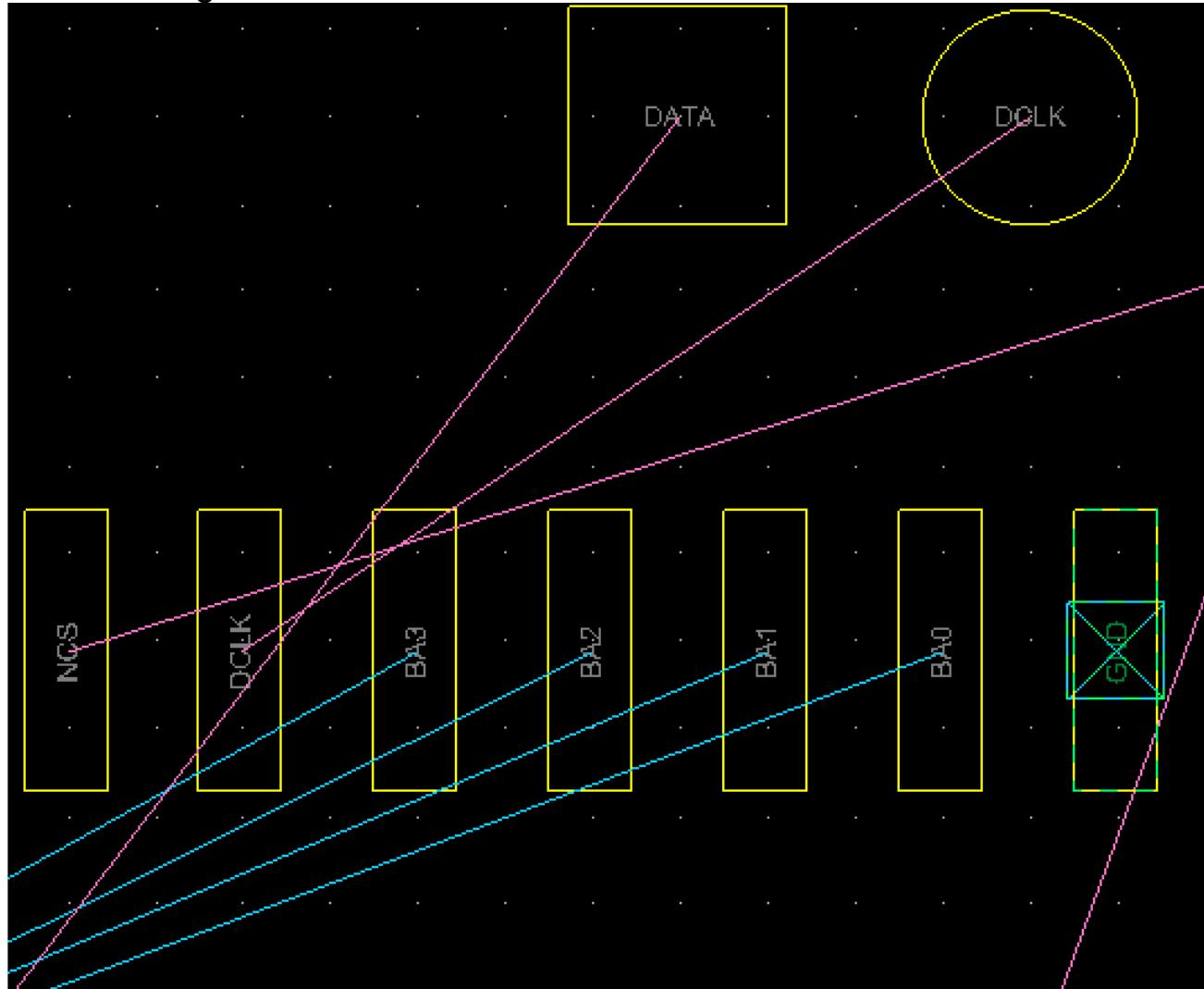
Options ▶

Snap pick to ▶

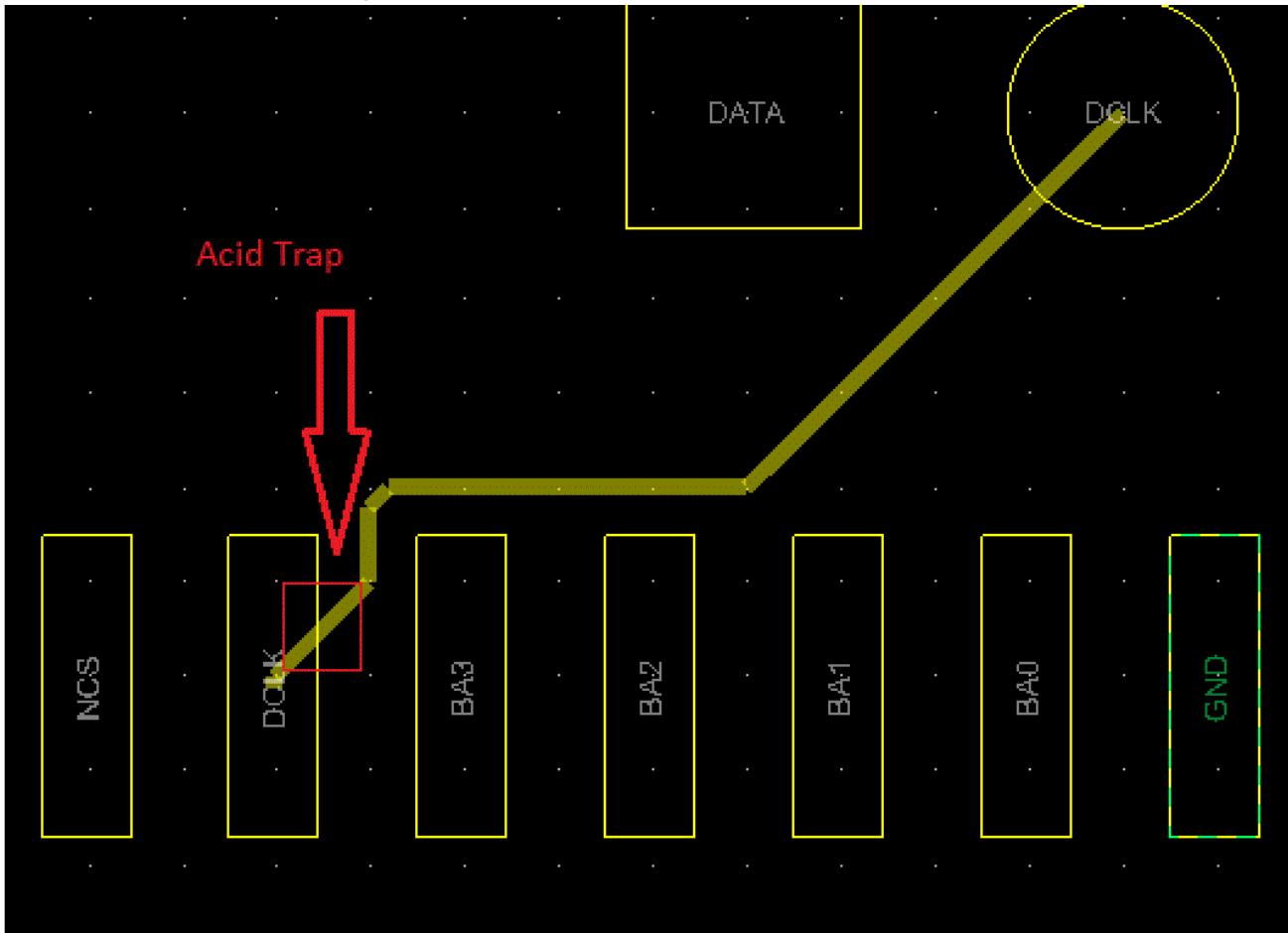
If this option is enabled then a cline exits a pad perpendicularly to or at an angle to the pad edge that does not create an acute angle.

Example

Before Routing

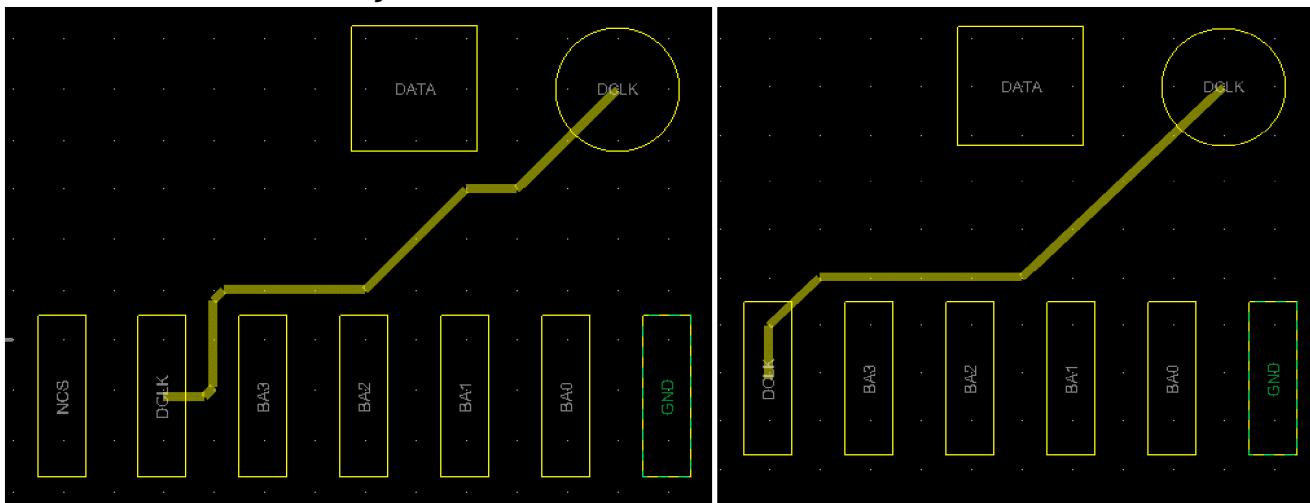


The Enhanced Pad Entry Disabled with `add_connect` Command



Cline exits at an acute angle with the pad edge.

The Enhanced Pad Entry Enabled with `add_connect` Command



Cline exits at an acute angle with the pad edge

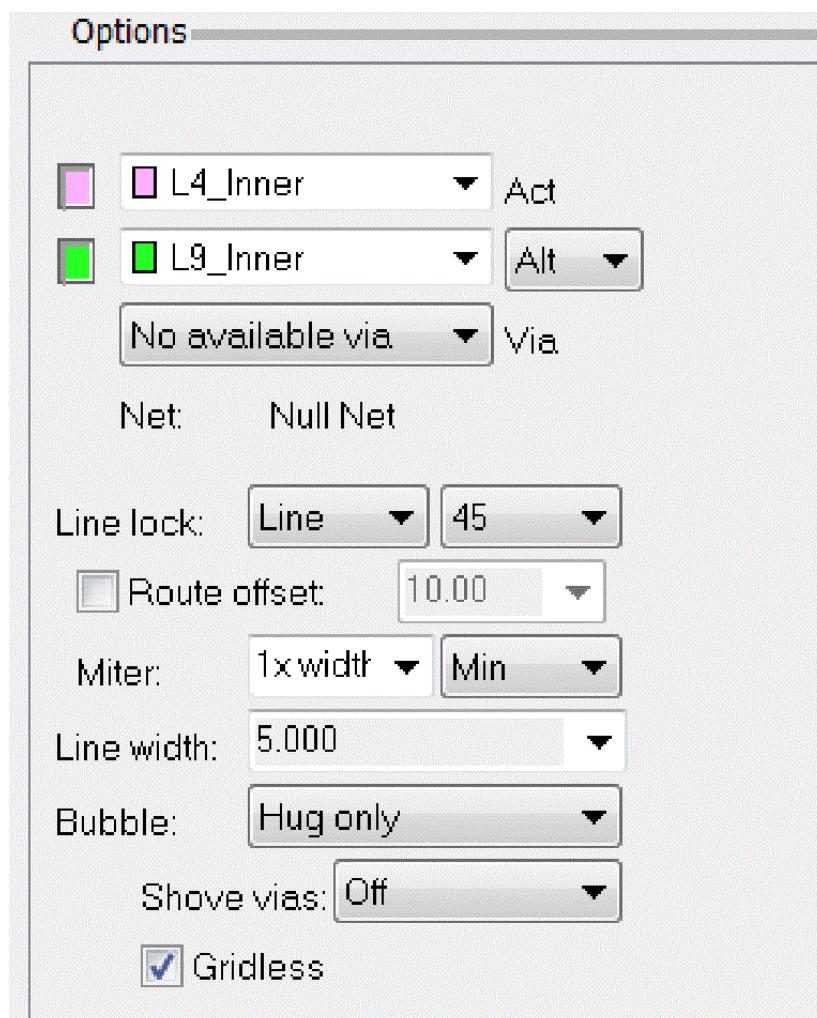
Cline exits at the corner of the pad edge

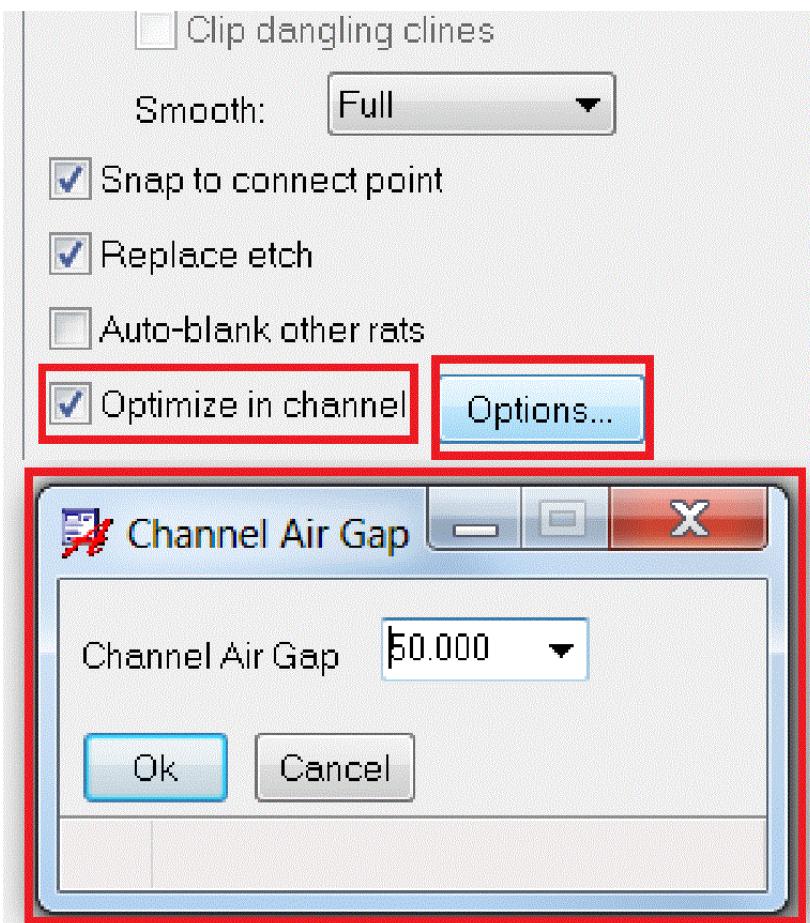
This mode works on all types of pads including shape-based pads, placed at any angle. When a line exits a pad it will not create a bend-point while vertex is less than the same net pin/via to line constraint. The behavior for exiting pads depends on line-lock values, location of the cursor, and picks relative to the pad position.

Optimizing Routes in Channels

Route optimization is a correct-by-construction mechanism that automatically centers the routes within a channel formed between two pins/vias during interactive routing. This technique results in better manufacturing yield and/or electrical performance. While regular routing tends to hug one side of the channel, optimized routing maximizes pad-to-trace spacing while keeping undesired trace jogs to a minimum.

Optimization during interactive routing is set by enabling *Optimize in channel* checkbox in the *Options* tab of the `add connect` command.

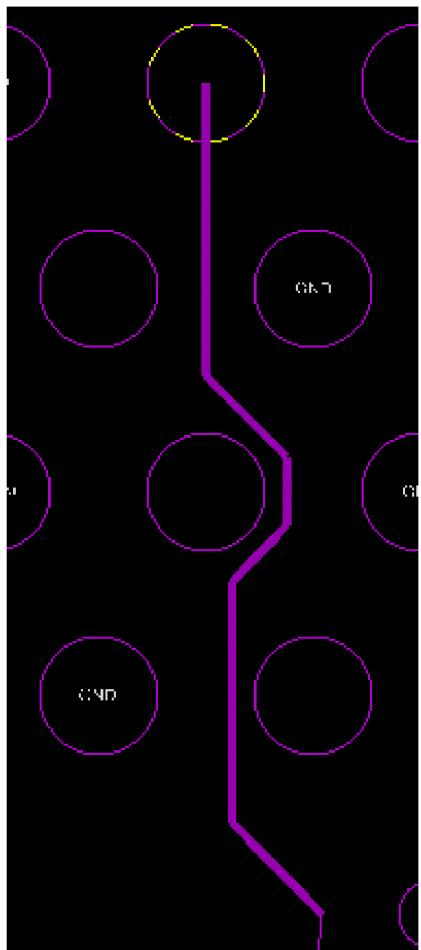




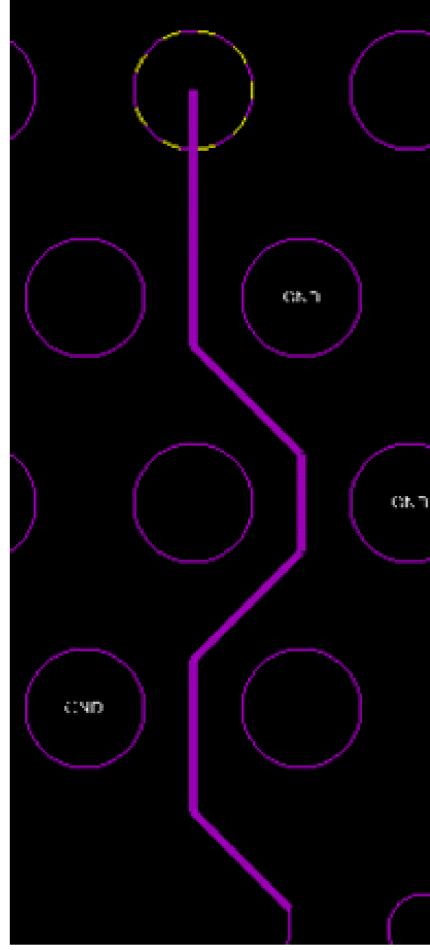
The *Channel Air Gap* defines the distance between the two pads (pins/vias) where clines are centered for optimization. Like other settings, the value of *Channel Air Gap* is saved in the `.env` file and is available in all subsequent sessions of the layout editor. To achieve maximum optimization, the *Channel Air Gap* value should be kept small.

Example

The following example shows the legacy routing and optimized routing between channels.

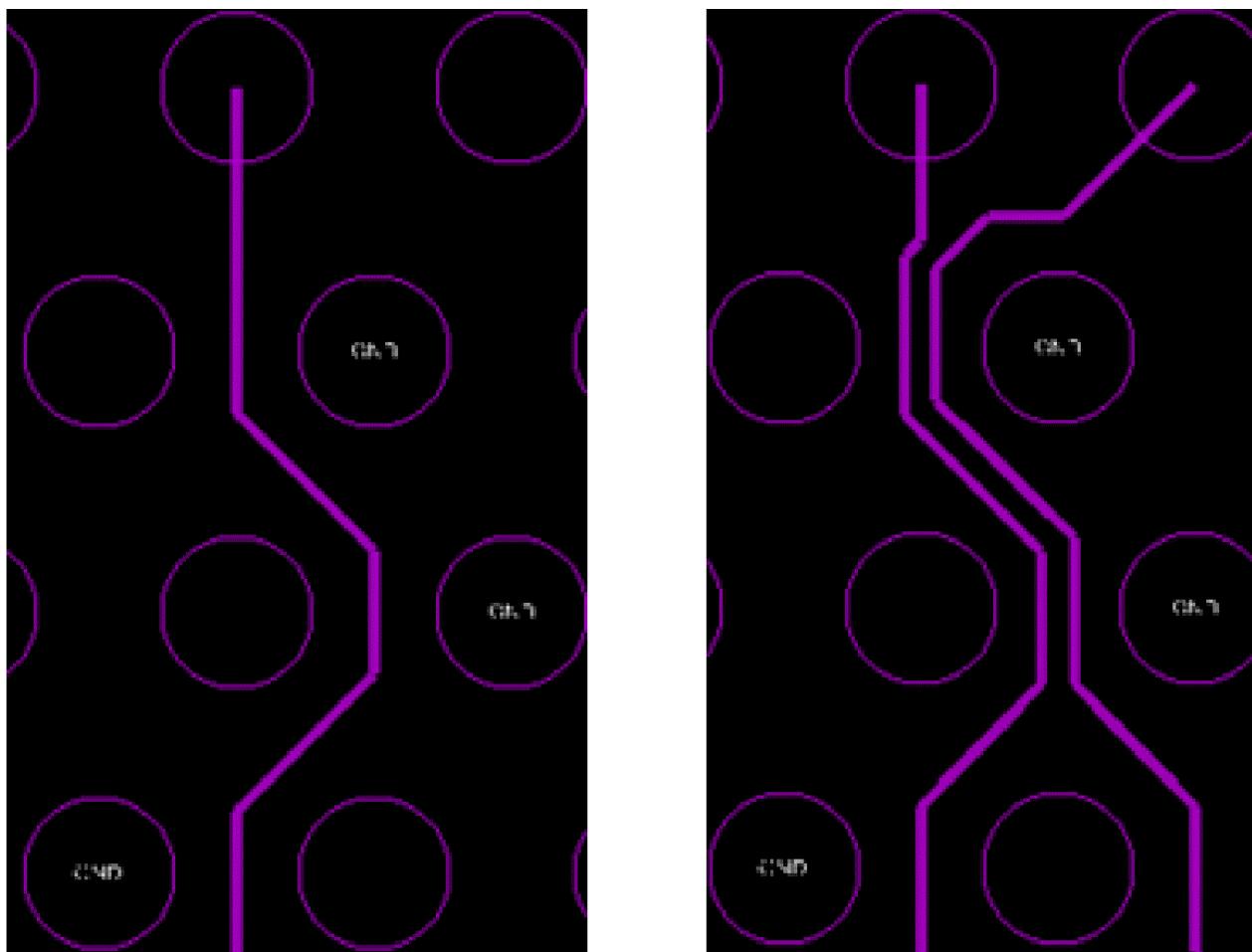


Non-optimized Routing: Clines not centered



Optimized Routing: Clines are centered

When *Optimize in channel* option is on, the `add connect` command centers all new cline segments in a channel. As a consequence, existing cline segments on different nets, which share the channel with the new cline segment, are also re-centered.



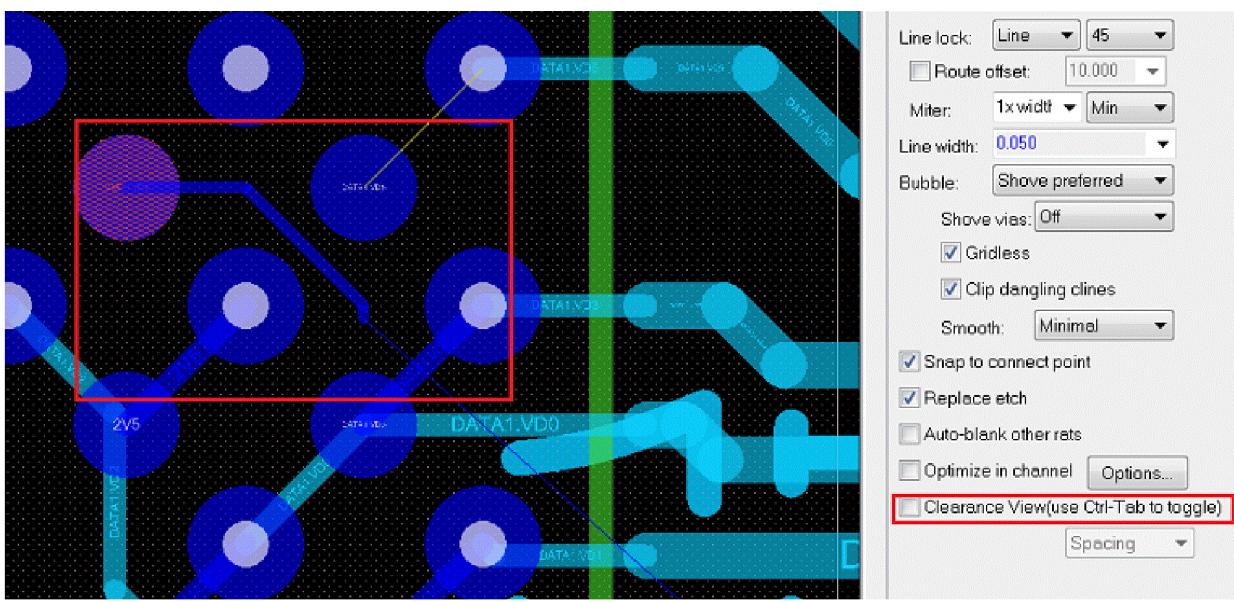
Re-centred existing cline segments with optimized routing

To achieve ideal results with *Optimize in channel* option, the following settings are recommended:

- Disable the option when routing dangling cline segments.
- Disable the option before making the final connection if *Replace etch* is on.
- Reduce the size of the channel if it is greater than four times of minimum line width.

Viewing Clearances in Channels

When routing in channels it is difficult to assess if the sufficient space is available so that no DRC error is generated.



Enabling *Route Clearance View* option in the `add connect` command acts as a visual guide and show the space left within channel after meeting the specified spacing constraints. If enabled, the command generates polygons around objects to show the amount of space available for routing.

The clearance calculation is based on the value of spacing constraint between cline and the object and line width constraint of the cline.

Two modes of operation are provided: Spacing and Channel. Depending on which mode is selected the calculation of clearance changes.

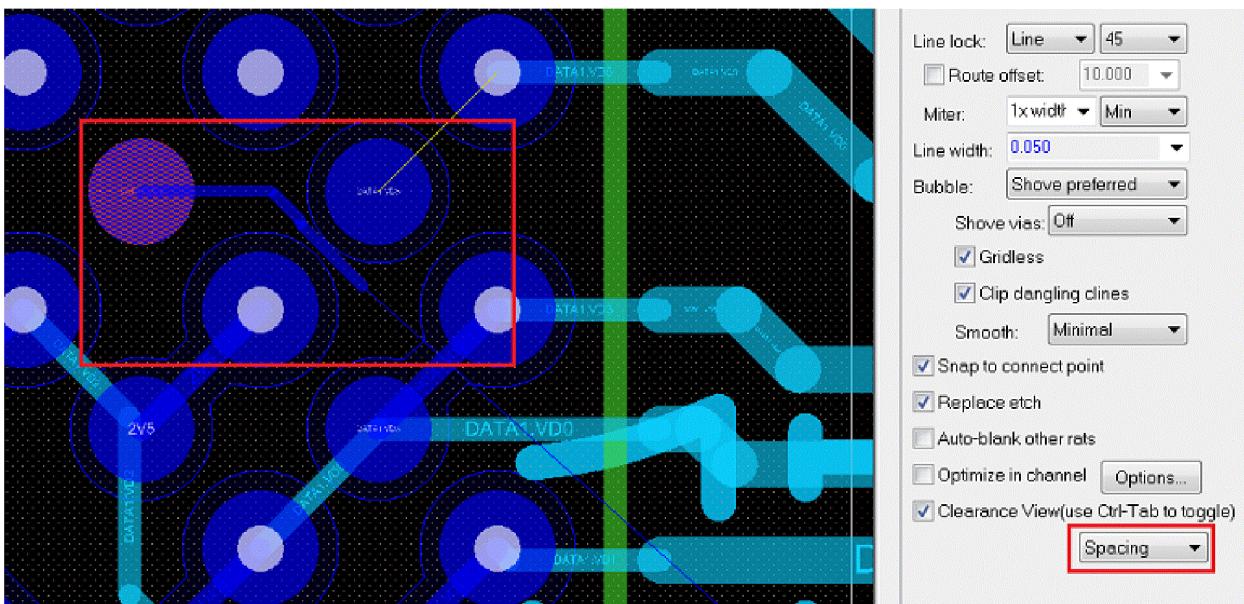


Figure 8-12 Route Clearance View: Spacing mode

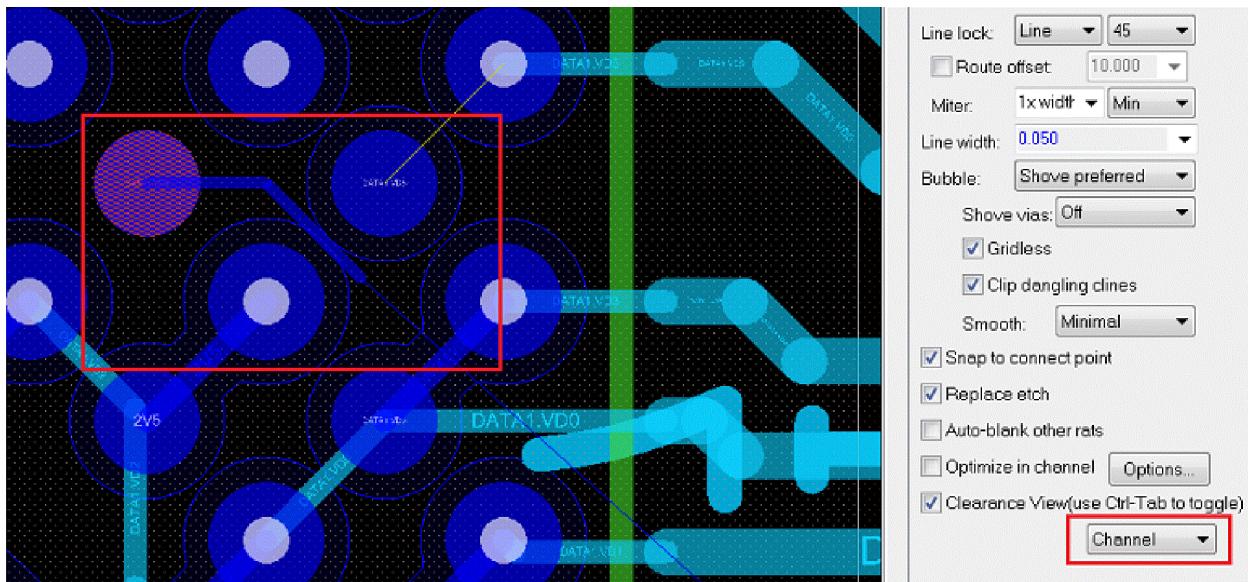


Figure 8-13 Route Clearance View: Channel mode

⚠ The option is enabled only when routing a single cline. It becomes inactive during group routing or multi-line route.

When routing a differential pair, clearance in Channel mode depends on the values of spacing constraint between cline and the other objects, line width of differential pair, and half the air gap between differential pair being routed.

APD: Routing or Sliding in Super Smooth Mode

You may spend a significant amount of time cleaning up traces after performing point-to-point routing. Although you can use the *Route – Custom Smooth* (*custom smooth*) command, it is an additional step to routing.

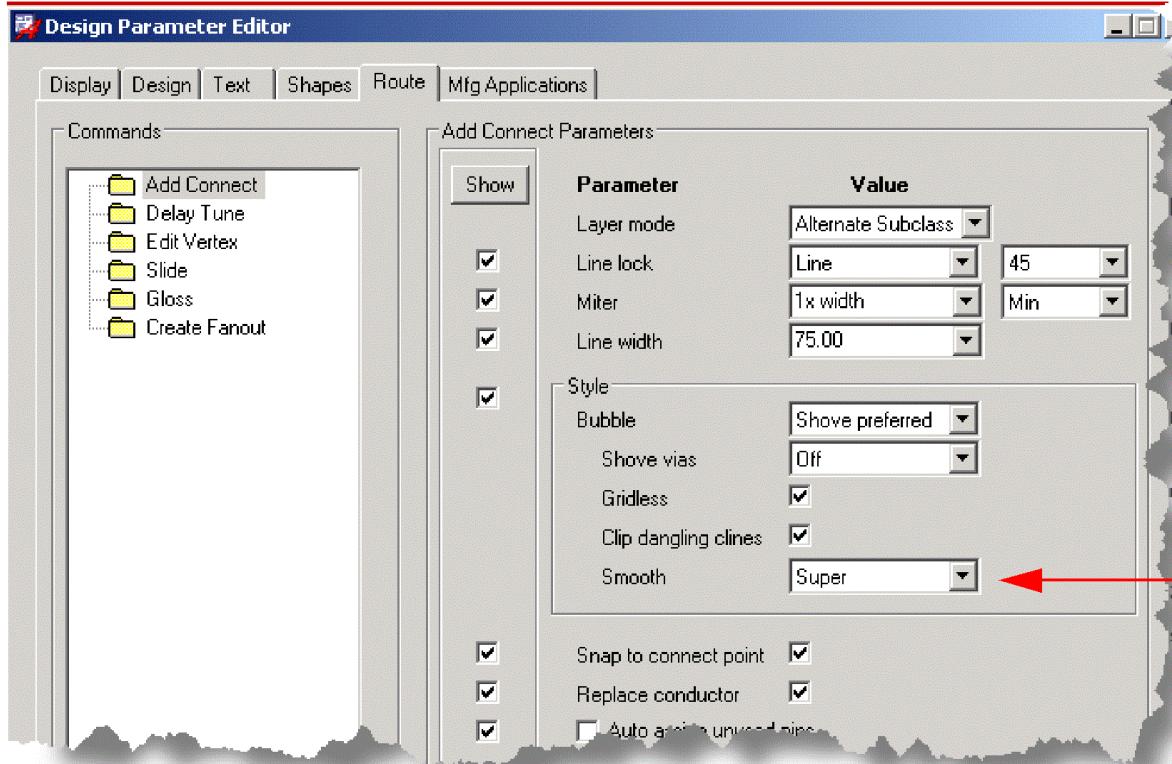
With the *Super Smooth* option, you can easily remove unnecessary vertices for the entire cline, as well as the shoved traces, during routing or sliding. This option is available when you run the `add connect` or `slide` commands. The *Super Smooth* mode affects the modified trace the same way as the *Full Smooth* mode, but it aggressively smooths out the entire clines for the shoved traces.

The *Super Smooth* mode ignores any timing rules, and smooths out delay loops if any exist.

How to Access the Super Smooth Option

You can set the Super Smooth option as follows:

- Run the `prmed` command, select the *Route* tab in the Design Parameter Editor, click either the *Add Connect* or *Slide* command in the *Commands* pane, then click the drop-down list in the *Smooth* field in the *Style* pane, and select *Super*.



- Run the `add connect` or `slide` commands and choose *Super* in the *Smooth* field in the

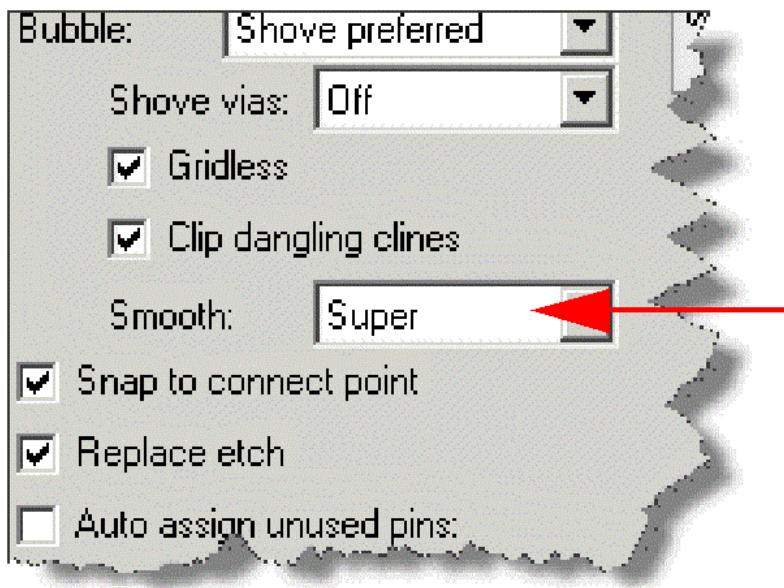
Options tab.

- Right-click and choose *Options – Smooth – Super* from the pop-up menu when running the `add connect` or `slide` commands.

Examples of Super Smooth Mode

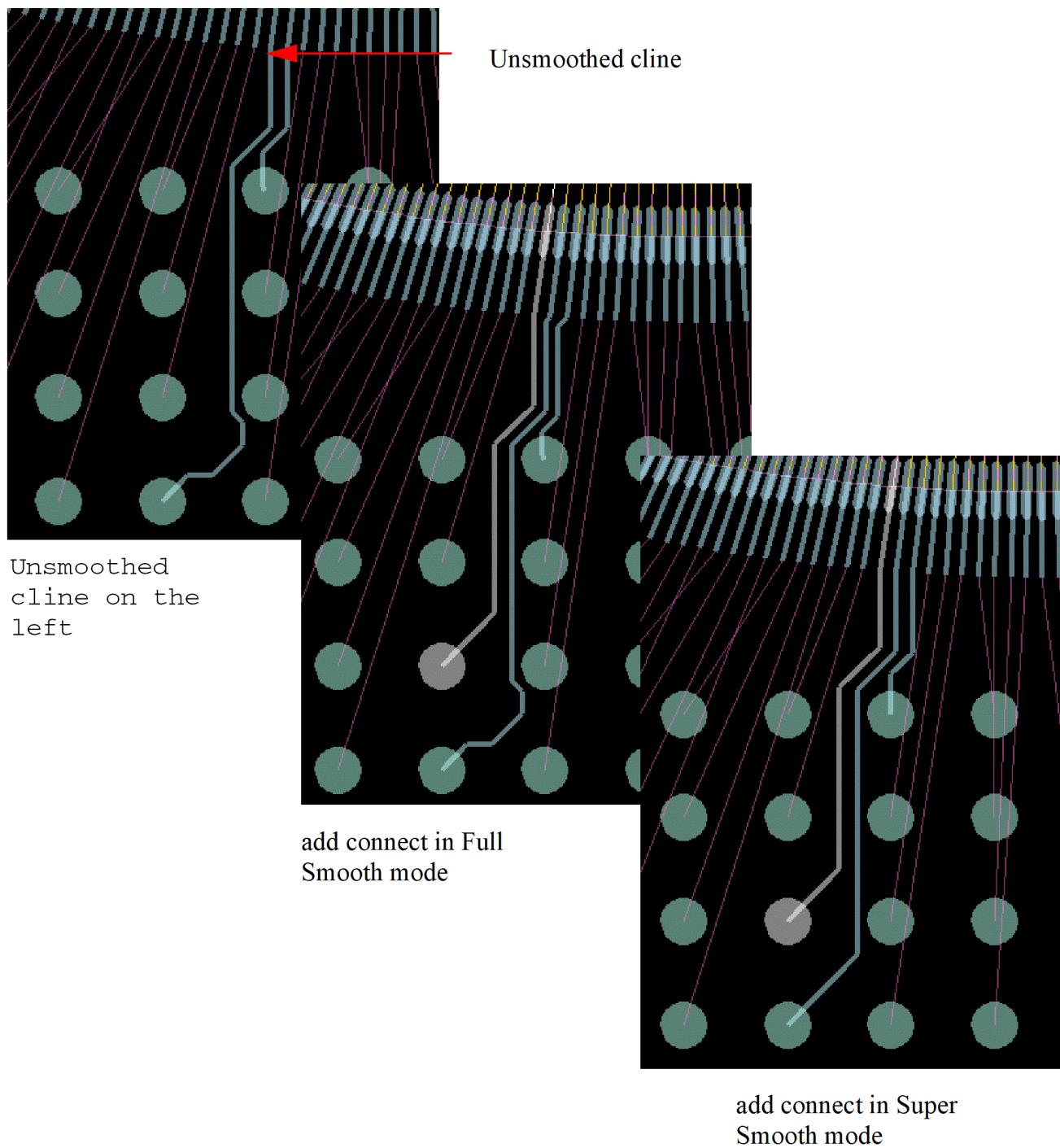
The following figure shows three graphics. The first graphic on the left shows a routed trace with unsmoothed vertices.

In the middle graphic, a third trace was routed (*Full Smooth* mode) and shoved against the two traces on the right, but still, the tool did not remove smooth the vertices.



In the graphic on the right, when the third trace was routed again using the *Super Smooth* option, the tool smoothed the vertices on both traces.

Comparing Full Smooth and Super Smooth Modes in add connect

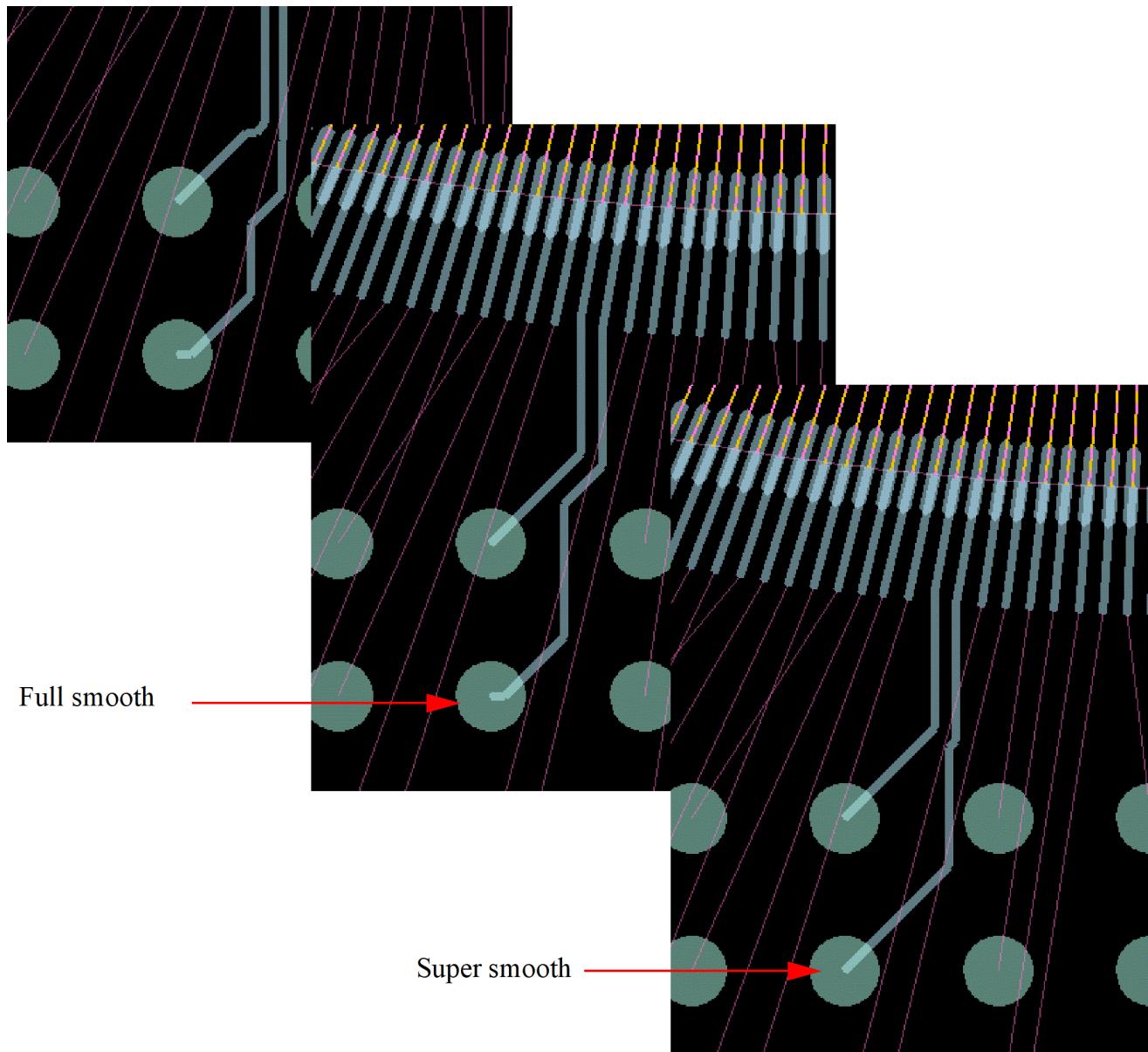


The following figure shows three graphics. In the left graphic, both traces have removable vertices.

In the middle graphic, use the `slide` command on the left trace to shove it against the right trace in *Full Smooth* mode. It does not remove the extra vertex of the right trace near the via center.

In the right graphic, use the `slide` command in *Super Smooth* mode on the left trace to remove the extra vertex from the round via pad in the right trace.

Comparing Full Smooth and Super Smooth Modes in `slide`



Interactive Routing with Layer Set Constraints

The `add connect` command lets you route nets that have layer-set constraints.

 You can place layer-set constraints on nets, XNets, differential pairs, buses, or ECSets.

The *Options* tab helps you to route with layer-set constrained nets in the following ways:

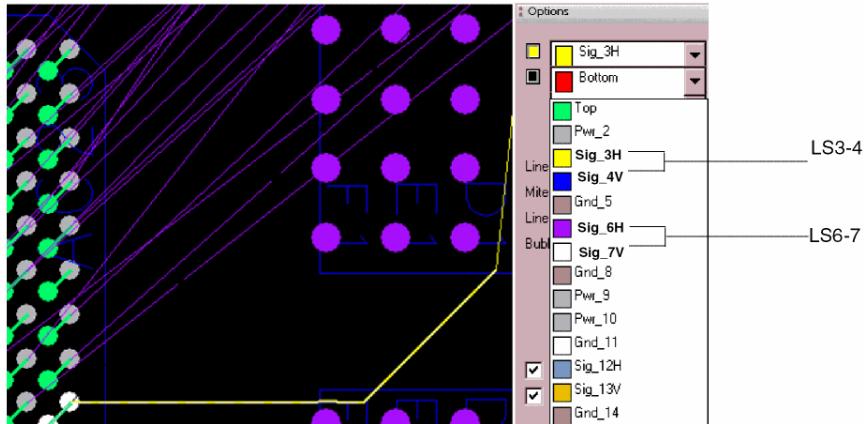
- Displays the legal routing layer-set layers (also referred to as subclasses) in bold-faced type.
- Automatically sets the *Act* (active) and *Alt* (alternate) drop-down list boxes to layers that are in the applicable layer sets.

 Uncheck the *Planes* box in the *Visibility* tab to reduce the number of layers that appear in the drop-down list boxes of the *Options* tab.

The layout editor defaults to the layer closest to the initial active subclass setting. When you start routing on a net, the layer-set constraint and the existing clines determine the layer set that applies to the current route. In cases where the layer-set constraint contains more than one layer set, the pre-existing clines determine the legal layers for both the active and alternate subclasses. If no clines exist, all layers in all layer sets of the layer-set constraint are legal for the active and alternate subclasses. If you choose a layer-set subclass in the *Act* drop-down list box, the layout editor displays the applicable alternate subclass in the *Alt* drop-down list box.

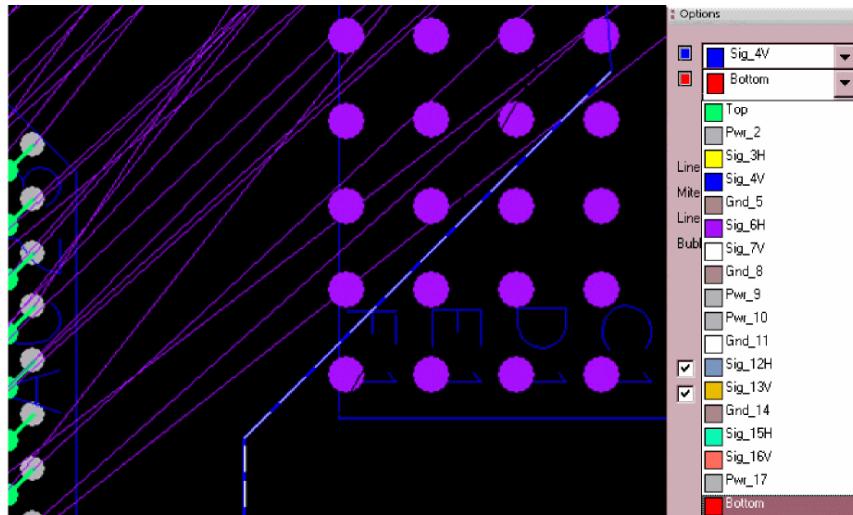
For example, in the following figure, a 64-bit bus has a layer-set constraint that allows `LS3-4` (layer set 3-4) and `LS6-7` as legal routing layer sets. When you choose the first element, the four layers contained in the two layer sets appear in bold-faced type in the *Act* and *Alt* subclass drop-down list boxes. For example, `LS3-4` consists of layers `Sig_3H` and `Sig_4V`. `LS6-7` includes layers `Sig_6H` and `Sig_7V`. The *Act* subclass defaults to `Sig_3H`, as a result of the previous setting of *Top*, and the *Alt* subclass becomes `Sig_4V`. After you pick the first element and the *Act* subclass was previously set to *Bottom*, the default *Act* subclass is `Sig_7V` and the *Alt* subclass becomes `Sig_6H`. If you want, you can change the active layer to a layer from the second layer set (`LS6-7`) at the point of the initial pick.

Picking the First Element



When you add the first cline, the layout editor locks the routing on the `LS3-4` set containing routing layers `Sig_3H` and `Sig_4V`. Subclasses `Sig_6H` and `Sig_7V` revert back to normal font in the *Options* tab, as follows.

Committing to a Layer Set



If a layer set does not include top and bottom etch layers, then surface-mount pins associated with the layer set require pin escaping to access the legal routing layers. In this case, DRC ignores the etch from the surface-mount pin to the via. The accumulated amount of etch length on non-layer set subclasses appears in the *Length* column of the Electrical Constraint Spreadsheet.

- ❗ The layout editor does not prevent routing on an illegal layer; however, DRC reports an error.

Defining Line Width

To determine the line width of the etch you add to the design, the layout editor uses several locations to determine the width of each line segment (lines and arcs). An impedance rule can control width, and as with minimum line width, it can come from the constraint set (except when it is electrical) or from a property (IMPEDANCE_RULE) on the net. The net property overrides the constraint set impedance rule.

The precedence is as follows:

- If you type in a new line width in the *Options* tab during operation, it takes precedence over values at any other locations.
- If the route is constrained by an impedance rule, a line width based on that impedance rule is used as long as that line width is greater than the minimum line width defined in the physical rule set. Line width, subclass, board materials, and cross-section affect impedance.
- If the route is constrained by an impedance rule and a neck width greater than zero that allows the cline to meet the target impedance, necking is used and automatically adjusts to meet the target impedance. If the minimum neck width is zero, necking is not used, so the neck width is not considered for automatic width generation. Instead the minimum line width is the constraint limit to meet the target impedance. If the target impedance is not met, a DRC error occurs.
- If no impedance rule exists, or if the width based on impedance is less than the minimum line width, the minimum line width is used.

⚠ Line widths that are derived from an impedance rule are rounded to the database accuracy. If database accuracy is greater than one decimal place, the resulting line width may not be an integer multiple of the base design unit; for example, 1 mil or 1 micron.

Overriding the Minimum Line Width Setting

By entering a new value in the *Line width* field of the *Options* tab, you can override the minimum line width determined by an impedance rule on a property or by a constraint setting.

If you are adding etch/conductor to an existing cline segment whose line width differs from that of the constraint/property setting, the new etch uses the width of the existing segment. This applies to new routes that extend a dangling cline or which ‘tee’ into a cline.

Unless you specify a width override in the *Options* tab and you connect from an existing segment, the layout editor automatically uses the width of the existing segment. The following table shows

behaviors that apply to minimum line width settings and manual overrides when you are connecting from existing segments.

Line Width Behavior When Connecting from Existing Segments

If...	Then the layout editor...
You override the line width for an existing segment	Uses the line value specified as an override for the connection.
You begin a new route or route from an existing segment that does not have an override value	Uses the line width setting defined in the constraint/property.
You connect from other element types, such as pins, vias, or shapes	Uses the minimum line width specified by the constraint/rule. This behavior, however, is partially dependent on the type of element selected and the element types selected in the <i>Visibility</i> tab.
You add a via while a minimum line width override is in effect	Reverts to its property constraint setting.
The line width of an existing segment is less than the constraint-based line width	Uses the constraint-based line width.

 The override value remains in effect until you manually reset in the *Options* tab.

Reverting to Constraint-Set Minimum Line Width

You can revert a manually overridden minimum line width to constraint/property settings. The drop-down menu associated with the *Line Width* field shows previous values that were set. If the current value in the field is not the default value (the minimum line width) the drop-down menu shows an item called *Constraint*. Choosing this item resets the line width so that the layout editor uses the minimum line width from the applicable physical constraint set. This feature replaces the *Reset* button.

When you use this function, the following conditions occur:

- The constraint/property-based minimum line width takes effect immediately and under all the conditions described above.
- The *Line width* field in the *Options* tab displays the constraint/property-based setting.

Cornering

During interactive routing, you now have more control in cornering with arcs and 45-degree angles.

If you set the *Line lock* option as *Arc* with 45 or 90 degrees, the radius controls appear. These controls let you specify a minimum or fixed radius.

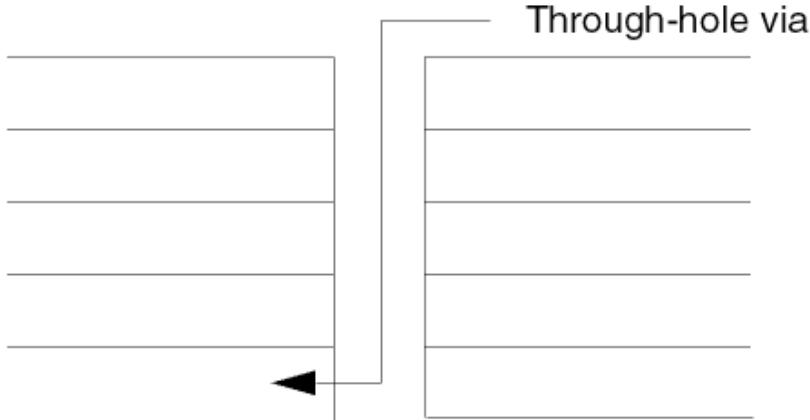
If you set the *Line lock* option as *Line* with a 45-degree angle, the miter controls appear. These controls allow you to specify a minimum or fixed miter size. Miter means that the layout editor cuts a corner with a 45-degree angle.

Adding Vias

There are two types of vias: through-hole and blind/buried. You can add either type as part of a connection or as a stand-alone via. A stand-alone via is either a through-hole or a blind/buried via that is not added to a particular connection.

Through-Hole Via

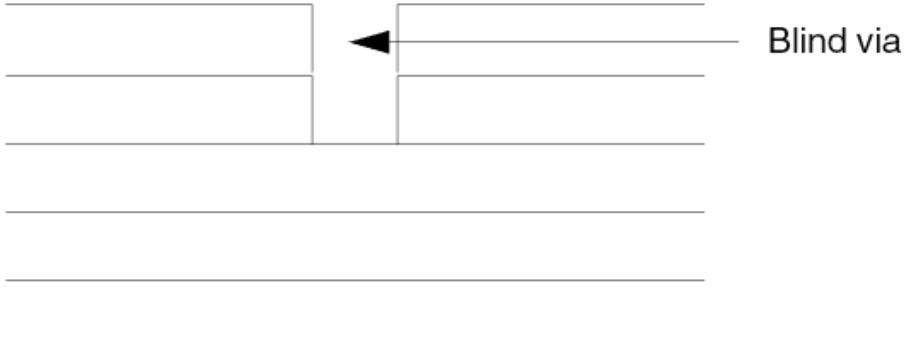
A through-hole via penetrates all layers and allows a connection to travel between the top and bottom etch layers.



To add a through-hole via, choose *Route – Connect* (`add connect` command).

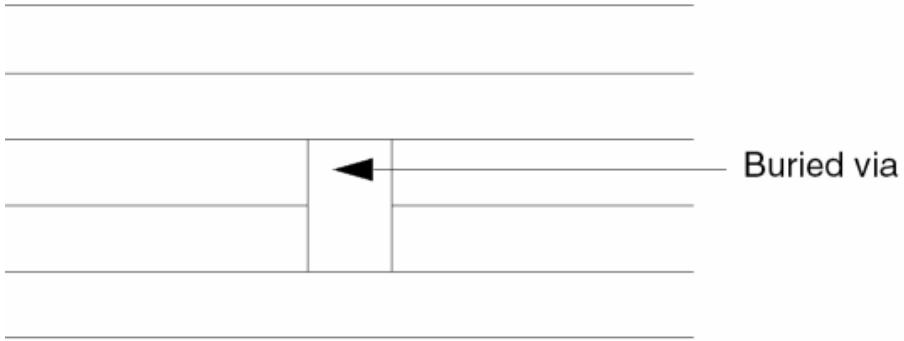
Blind Via

A blind via travels between an outer layer and an inner layer.



Buried Via

A buried via travels between two internal layers.



Defining Blind and Buried Vias

Before you can add blind or buried vias to a design, first create a blind or buried padstack. You can do this automatically by choosing *Setup – Vias – Auto Define B/B Via* (`auto_define_bbvia` command) or manually by choosing *Setup – Vias – Define B/B* (`define_bbvia` command). You can also define a via in the Padstack Designer by choosing *Tools – Padstack – Modify Design Padstack* (`padeditdb` command).

To assign vias that you are going to use on a net for routing, choose *Setup – Constraints – Physical*, then in the Physical worksheet of Constraint Manager, click in the cell under the *Vias* column for the net and select the vias from the Edit Via List dialog box.

Adding Via Structures

There are two types of via structures: standard and high speed. To use via structures in a design, you need to create, define, and add them to the design.

Standard Via Structure

A standard via structure is created from a single via or connect line, a via and a connect line, or multiple vias of different pin sizes and multiple connect lines of different widths.

High Speed Via Structure

High-speed via structures can contain dif-pairs with return path vias and can include route keepouts for custom voiding.

Adding Via Structures in Constraint Manager

For adding via structures during routing (`add connect` command), you must add via structures in the Constraint Manager in the Via Structure column available at

- Electrical Constraint Set – Routing – Vias worksheet
- Electrical Constraint Set – All Constraints – Signal Integrity/Timing/Routing worksheet

Via structures are assigned to nets and diff pairs by referencing an ECSet. Via Structures column in the Net worksheet displays values per ECSet assigned. You cannot edit the Via Structures column. When routing diff pairs, the `add connect` command uses via structures that are added to ECSet and assigned to DPr object. If no ECSet assigned (or inherited) to the diff pair, the 2 ECSets for each diff pair net is merged.

Setting Up Ordered Via Structures List

The ordered via structures lists reside in the Edit Via Structure List dialog box. The order of the via structures in the list determines whether a via structure is preferred or is an alternative via. A preferred via structure should be placed higher in the list.

To view/edit the via structures:

1. Select *Setup – Constraints – Electrical*. This command launches Constraint Manager in the Electrical Constraint view, a detail of which is shown as follows.

Constraint Manager, Electrical

The screenshot shows the Constraint Manager interface. The left pane is the Worksheet Selector with 'Electrical' selected. The right pane is the 'start1' worksheet, which contains a table with columns for Objects, Max Via Count, and Via Structures. The table has three rows: one for 'Dsn' (start1) with Max Via Count set to '*' and Via Structures set to '*', and two for 'ECS' (DIFF_PAIR_ELEC and ECS11) where Max Via Count is '*' and Via Structures are set to 'VS_2:VS_1'.

Objects			Max Via Count	Via Structures
Type	S	Name		
*	*	start1	*	*
ECS		DIFF_PAIR_ELEC		
ECS		ECS11		VS_2:VS_1

2. In the *Via Structures* column, click on the worksheet cell for the selected Electrical Cset (ECSet) to display the Edit Via Structure List, shown as follows.

Via Structure List Setup for Working Layers

The screenshot shows the 'Edit Via Structure List' dialog. On the left is a list of available via structures, and on the right is a 'Preferred vs. Alternative via structures' list. Arrows point from the 'Preferred' section to the 'Preferred' list and from the 'Alternative' section to the 'Alternative' list.

Name	Start	End	Signal Paths	Return Paths
VS_4		0	0	0
VS_TOP		0	0	0
VS_3	TOP	TOP	1	0
VS_2	TOP	TOP	1	0
VS_1	TOP	TOP	1	0

Name	Start	End	Signal Paths	Return Paths
DP_OFFSET	TOP	L4_SIGY	2	1
DP_EQUI	TOP	L3_SIGX	2	1
DP_RECTANGULAR	TOP	L3_SIGX	2	1
VS_TEST	TOP	TOP	1	0
DP_TEST	TOP	TOP	1	0

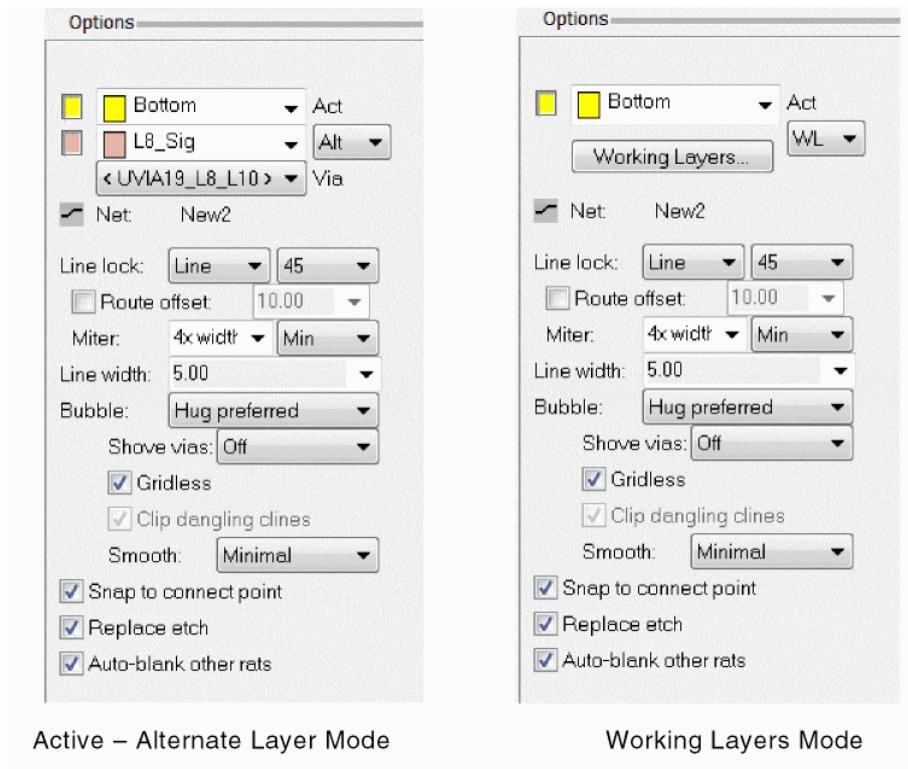
In the example shown above, two vias are defined for each set of adjacent HDI layers in the current design (for the associated cset).

The via structures above the imaginary red line in the previous figure are defined as *preferred via structures* because the Add Via Structure operation will select, by default, the appropriate via structure for the layer sub-set that is highest in the via structure list.

Use Models for Adding Vias and Via Structures

Allegro provides two use modes for adding vias/via structures to your design: the *Active layer–Alternate layer* mode and the *Working Layer* mode. The *Options* tab reflects the differences in these two modes.

Options Tab Configurations for Adding Vias



- **Active Layer – Alternate Layer Mode**

In this mode, the *Act* (active subclass) drop-down list box displays the current value and provides choices for modifying the value. While actively routing a net, you can change the value, if you are routing from a via or multi-layer pin. If layer-set constraints exist for the net, the layers (also referred to as subclasses) in the legal layer sets appear in bold-faced type. The *Alt* (alternate subclass) option in the drop-down displays the current value and provides choices for modifying the value. The alternate layer becomes the active layer when you double-click on the canvas to add a via or choose *Swap Layers* or *Add Via* from the right-click pop-up menu when an element is active. If no element is active or the active element also exists on an alternate subclass, *Swap Layers* lets you alternate active and alternate layers. If layer-set constraints exist for the net, the layers in the legal layer set appear in bold-faced type.

The *Via* field lists the available vias between the active and alternate subclasses. The *VS* field lists the available via structures between the active and alternate subclasses. You must

have already defined the vias/via structures in the constraint set via/via structure list.

- Working Layers Mode

The Working Layers mode addresses the challenges of working on HDI designs (though you can use it on any layout). In this mode, the *Working Layers* dialog box (Figure 8-23) displays all the etch layers in the current design and is used to control the layers that appear in the Via/Via structure pop-up GUI. Instead of being confined to routing from an active layer to a single alternate layer, a double-click in this mode launches a pop-up GUI with all working layers available for selection. A single pick on any of the layers resumes routing on that respective layer. When routing to HDI Rules, you can automatically add stacked vias or semi-automatically add staggered vias or via structures across multiple layers. Additionally, you do not need to continually navigate from your design to the Options tab in order to select individual vias for each layer. Features of the Working Layers mode are covered in more detail below.

Adding Vias Using the Working Layers Mode

As is the case in the Active Layer – Alternate Layer mode, in Working Layers mode you must have predefined vias in the constraint set via list for the layers upon which you want to route; however, this mode assumes that you have defined an "ordered via list" for csets. The vias listed in the Edit Via List will be available for adding when you route interactively.

 To ensure maximum routing flexibility, we recommend that the vias in your via list span the entire cross-section of your design.

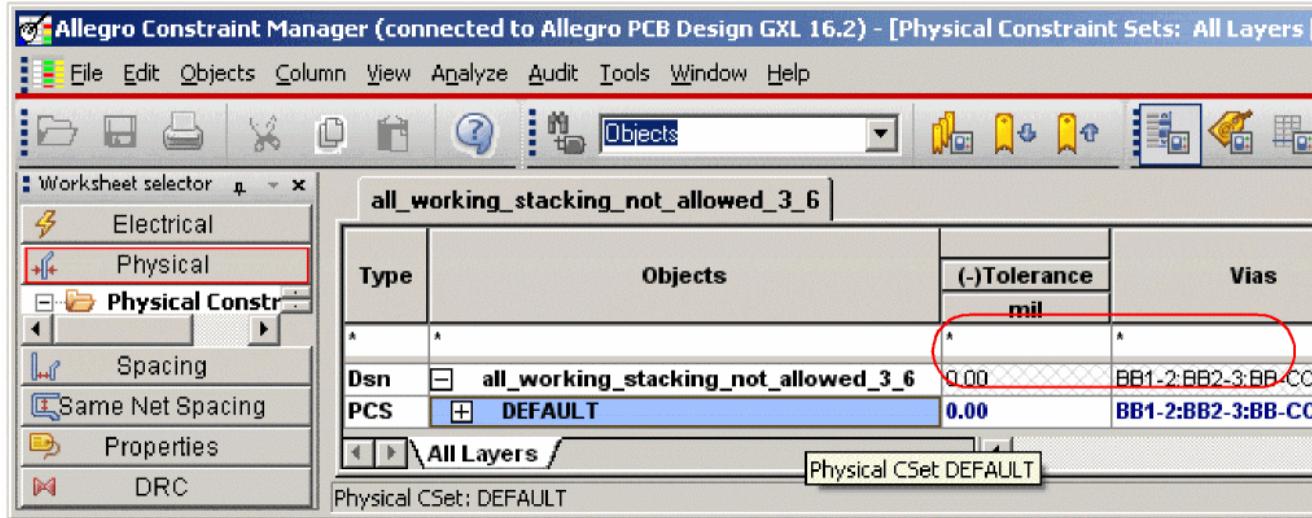
Setting Up Ordered Via Lists

The ordered via lists for your designs reside in the Edit Via List dialog box. The order of the vias in the list determines whether a via is preferred or is an alternative via. A preferred via is one that you expect to use most frequently. Your preferred via should be placed higher in the list than other vias with similar characteristics; for example, BB1-2 is listed above BB1-2B because BB1-2 is the via you will use more often.

To view/edit the vias:

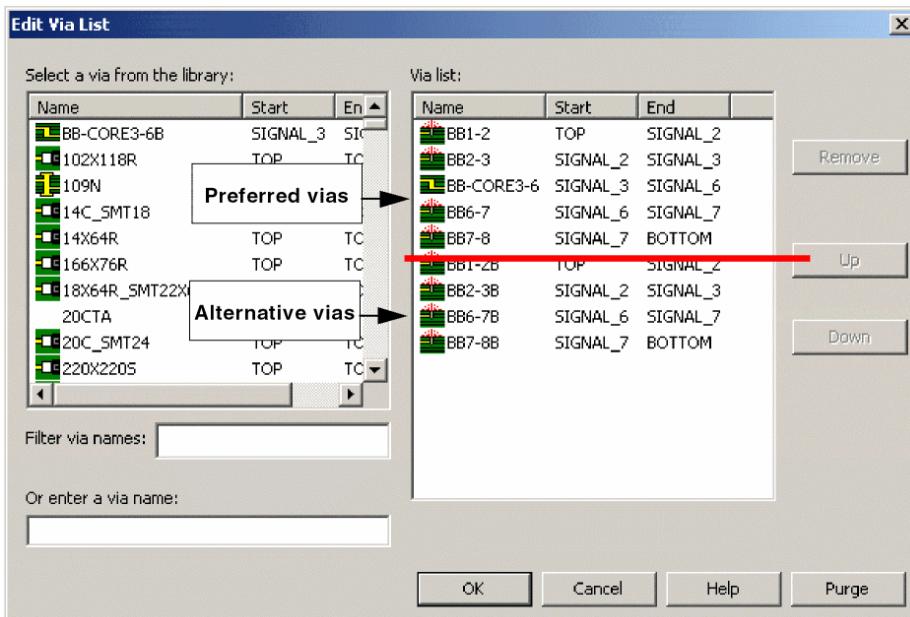
1. Select *Setup – Constraints – Physical*. This command launches Constraint Manager in the Physical Constraint view, a detail is as follows.

Constraint Manager, Physical



2. In the *Vias* column, click on the worksheet cell for the selected Physical Cset (PCS) to display the Edit Via List, as follows.

Via List Setup for Working Layers



In the example shown above, two vias are defined for each set of adjacent HDI layers in the current design (for the associated cset).

The vias above the imaginary red line in the previous figure are defined as *preferred vias* because the Add Via operation will select, by default, the appropriate via for the layer sub-set that is highest in the via list.

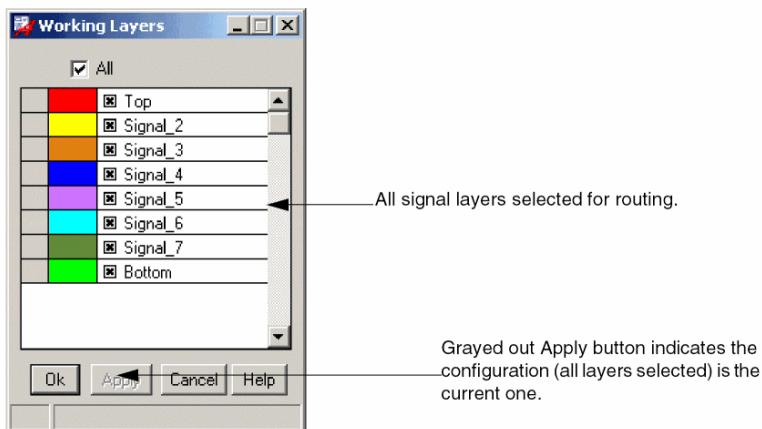
Determining the Working Layers

The layers you select for routing in the Working Layers dialog box determines which layers will appear in the Add Via dialog when you double-click to add a via.

- Select only the working layers you know you will route on to minimize the number of layers that will appear in the Add Via dialog box.

The following figures illustrate the features of the Working Layers dialog box.

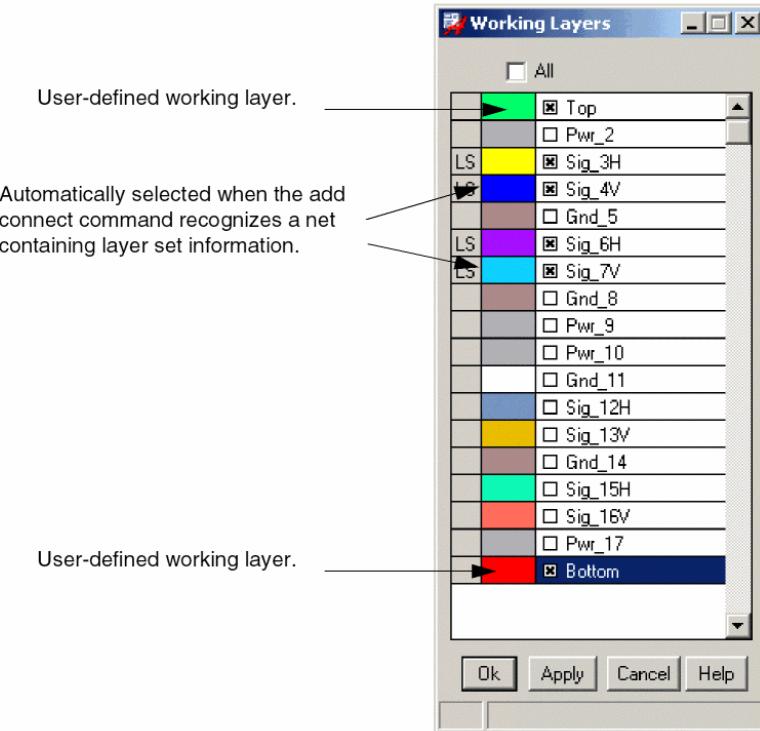
Working Layers Dialog Box



In this figure, all the etch layers have been selected for routing and the *Apply* button clicked to allow routing and via adding.

- Whenever a condition in the dialog box changes (for example, turning off one or more layers), you must "refresh" the dialog with the *Apply* button.

Working Layers Dialog Box with Layer Sets Defined



In the previous figure , the Top and Bottom layers are selected for routing. However, the selected net contains a layer set constraint that defines routing between layers 3–4 and 6–7. Therefore, those layers are automatically turned on and flagged with the LS (layer set) designation. These layers automatically turn off when you select another net.

Adding Preferred and Alternative Vias

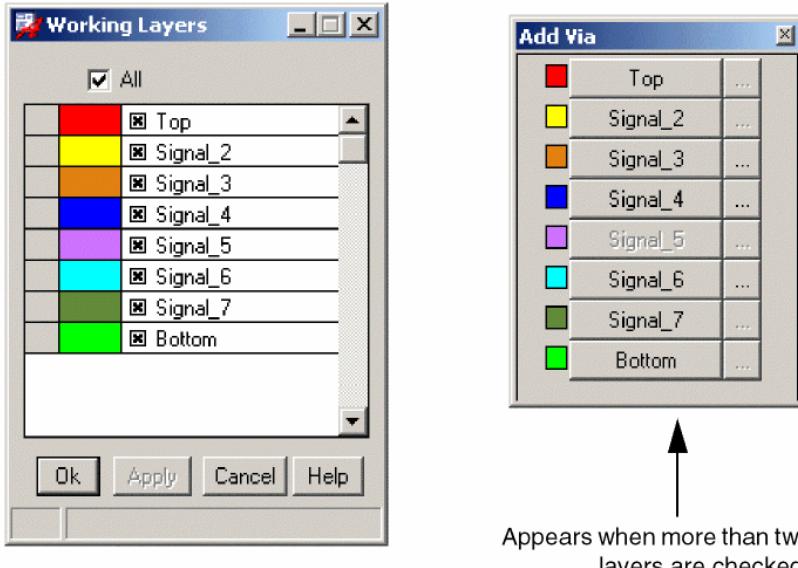
When you double-click to add a via in the design canvas, the Add Via dialog appears only when you have *more* than two layers turned on in the Working Layers dialog (as illustrated in Figure 8-25). When you have only two working layers (that is, an active layer and an alternate layer), the preferred via (if there is more than one via defined for those layers) is selected by default and uses those two layers as the start and end points.

Note the following conditions related to the Add Via dialog box:

- The active layer is always greyed-out, because you will always be routing *from* it to another layer.
- Greyed-out browse (...) buttons next to routable layers indicate that alternative vias are not available for adding on that layer.
- An inactive layer other than the active layer indicates that the layer is not accessible, probably

due to an incomplete via list.

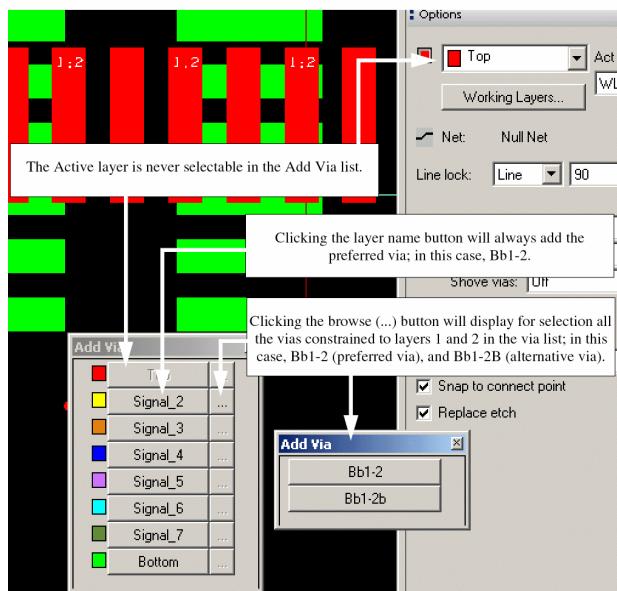
- **Add Via Dialog Box**



Appears when more than two working layers are checked.

The example shown in the following figure illustrates the operation of the Add Via dialog box. The active layer is Top (the starting layer of your route) and you want to add a via to the adjacent layer, Signal_2. Note the conditions called out in the graphic.

Via Selection Conditions



The conditions under which you can add vias vary according to the type of via needed to satisfy routing requirements:

- Single vias

For this requirement, one via is created from the active (start) working layer to the end working layer. The end layer then becomes the active layer on which a cline segment is created and a via attached. This process continues, layer-by-layer, until routing is complete.

- Multiple vias with stacking allowed

Similar to the single via process, two or more same-net vias are created directly on top of each other, layer-by-layer.



To allow vias to be stacked, two physical constraints must be adhered to:

A straight one-segment cline connects the spiral via to the previous via. The spiral via cannot be dropped outside the spacing restrictions *unless* you override the rules by pressing the control (`Ctrl`) key to release the via. You can then move it freely to any location on the canvas along a straight-line segment, as shown in the following figure.

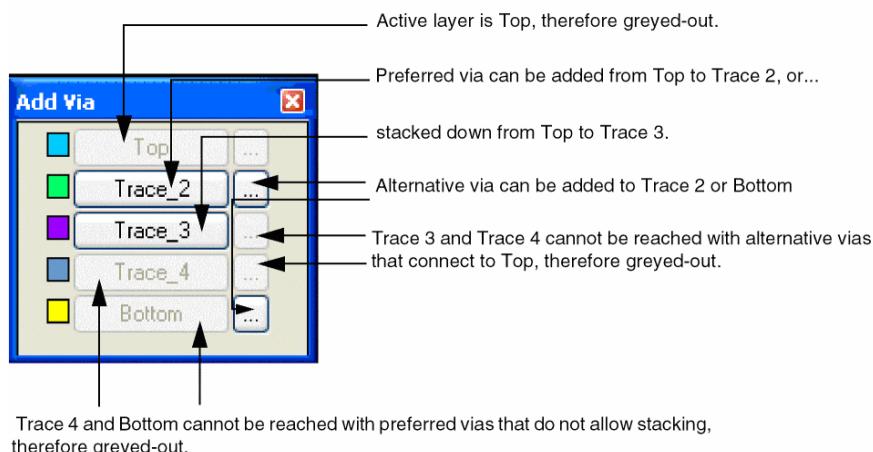


Adding Vias for Differential Pair Nets

When you add vias for differential pair nets, the end-layer is restricted to those that allow stacked vias. In the Add Via dialog, a layer that does not allow stacking between it and the active layer will be greyed-out. In order to create staggered vias on a differential pair net, you must add individual via stacks on a layer-by-layer basis, using alternative vias where necessary.

The following figure explains the meaning of the active and inactive status of the layer name buttons and associated browse buttons for the differential pair net process.

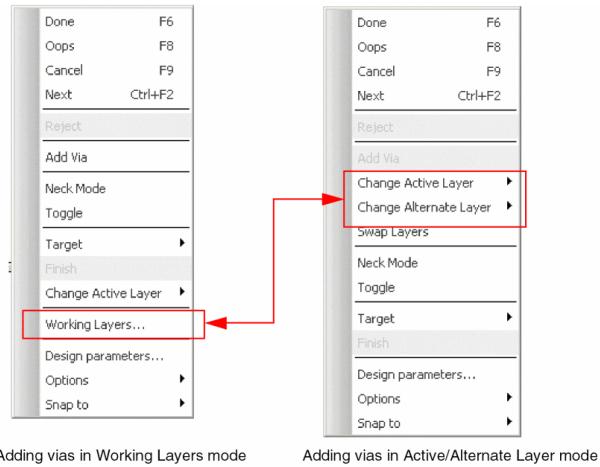
Allowed and Prohibited Actions for Adding Vias to a Differential Pair Net



Context-Sensitive Menus for Adding Vias

The two operating modes available for adding vias display different options in the right-button pop-up menus when the `add connect` command is active. The following figure illustrates these differences.

Pop-Up Menu Configurations



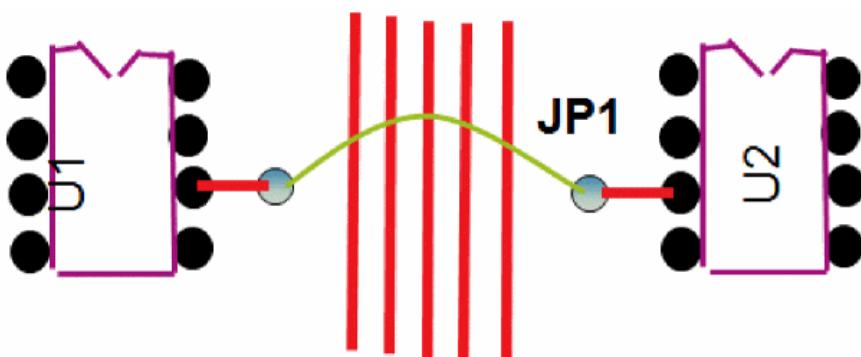
Adding vias in Working Layers mode

Adding vias in Active/Alternate Layer mode

Adding Jumpers

While routing a single layer design you can find situations when there are overlapping traces. Such situations cannot be avoided by replacement of components or by rerouting multiple times. In such cases a jumper is used.

A jumper is a short piece of wire that jumps over a trace to avoid the overlap. A wire jumper connects two points electrically.



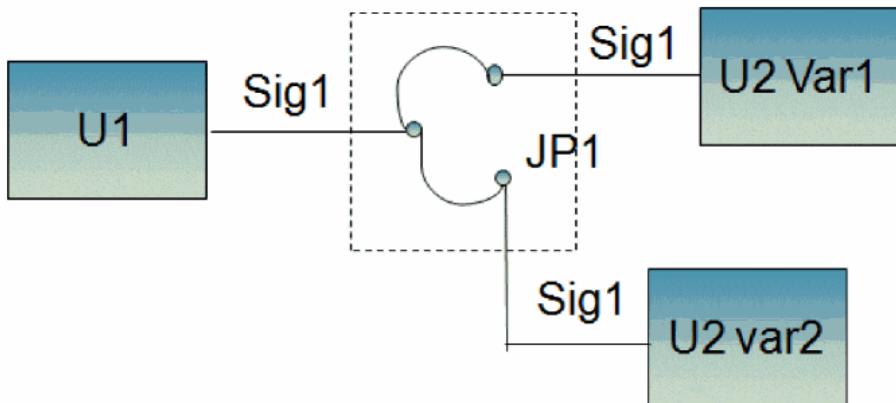
Jumpers are flat, stand-alone copper plated traces with sized pads and holes matching the circuit boards.

Types of Jumper

Jumpers can either be driven from a schematic or added via routing. A single design can include both types of jumpers.

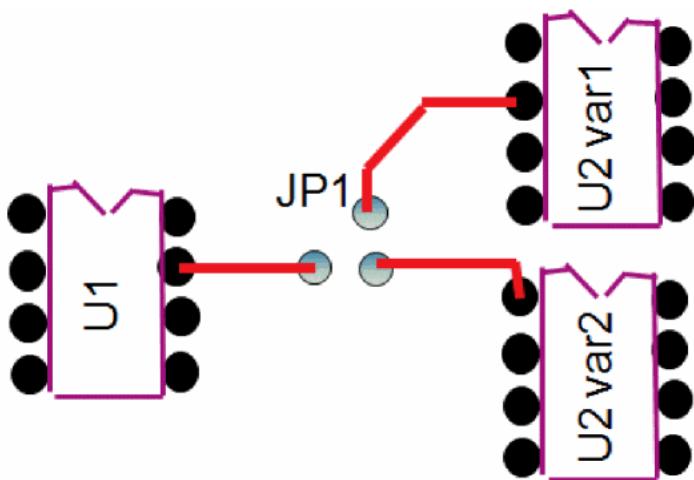
Model-based jumper

- Driven by front-end tools
- A component like element used to control variant design features.
- Two or more jumper points.
- Functions similar to a switch.



Routing Based jumper

- Driven from the back-end tools.
- A model that represents a hard wired connection.
- Used to connect a single net when etch connections are not possible.
- Two jumper points (Pin/via) only



⚠ Jumpers are not supported in modules and design partitions.

Jumper Package symbol

You can create a jumper package symbol in symbol editor.

Assigning the JUMPER_LIST Property to the Board

In order to use a jumper in design you need to add the JUMPER_LIST property to the drawing. The JUMPER_LIST is a board level property. This defines the list of jumpers allowed in the design. You can add multiple jumpers in a design.

To add the JUMPER_LIST property to a drawing complete the following steps:

1. Choose *Edit – Properties*.
2. In the *Find* tab and choose *Drawing* from the drop-down list.
The *Edit Property* dialog box appears.
3. Choose the *Jumper_List* property from the list of *Available Properties*.
The *Jumper_List* property appears in the right side of the dialog box.
4. Enter jumper symbol name. You can add multiple jumpers or names for the *Jumper_List* property value field.

Adding Jumper to Routes

To add jumpers during etch editing a new right-click pop-up menu is added. The new menu *Add Jumper* is enabled if the active layer is TOP or BOTTOM.

You can choose a jumper from the list. The jumpers that are defined in the PSMPATH are displayed in bold. The jumpers that are not defined in the PSMPATH are disabled in the list. Click in the design to complete the jumper placement.

Related Topics

- [add connect](#)
- [Layer Sets](#)
- [add connect](#)
- [Creating Jumper Package Symbol](#)

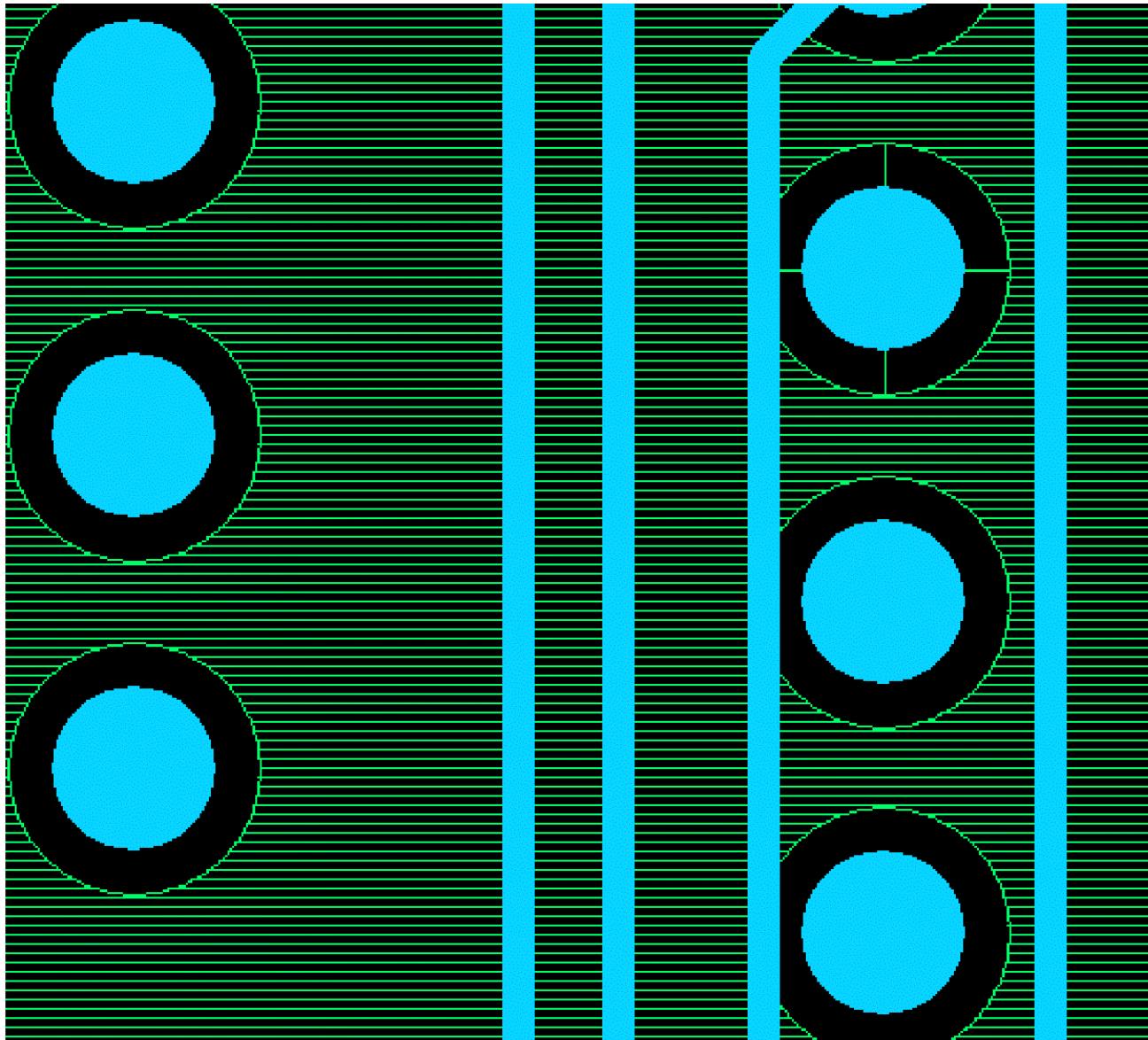
Editing Connections and Vias

You can use menu items or routing commands described in the *Allegro PCB and Package Physical Layout Command Reference* to perform the tasks listed in this section.

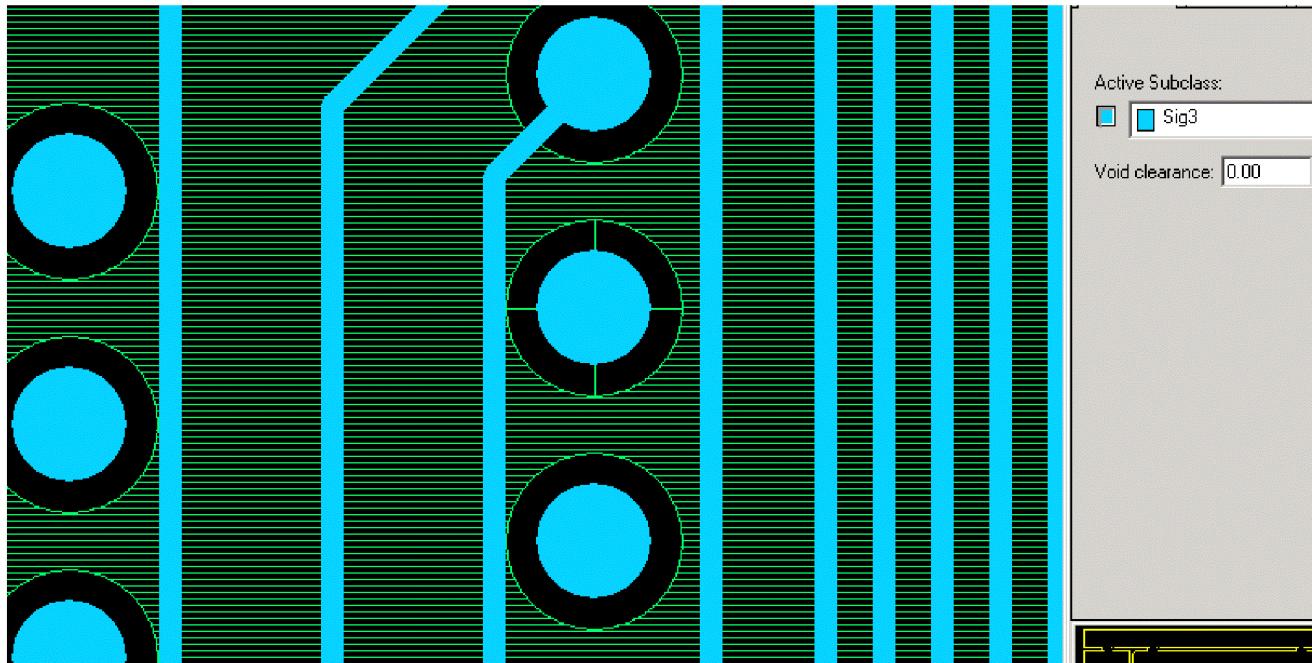
The Spread Between Voids feature spreads out the clines in a routing channel you specify. You can use this feature to correct return path issues that occur when clines overlap pad voids on adjacent layers. Typically, you apply the spreading function at the end of the design process after you complete routing, meet all other design constraints, and execute the [highlight sov](#) command to highlight any problems.

You choose two objects (a combination of two pins, two vias, or one of each) that define a routing channel. The clines within the channel are pushed toward voids found on adjacent planes. The spacing between routes is the largest minimum spacing for the group of clines in that channel. If the clines cannot space evenly, based on the spacing rule applied for spreading, then no spreading occurs. The following examples illustrate what happens to the clines using different void clearances.

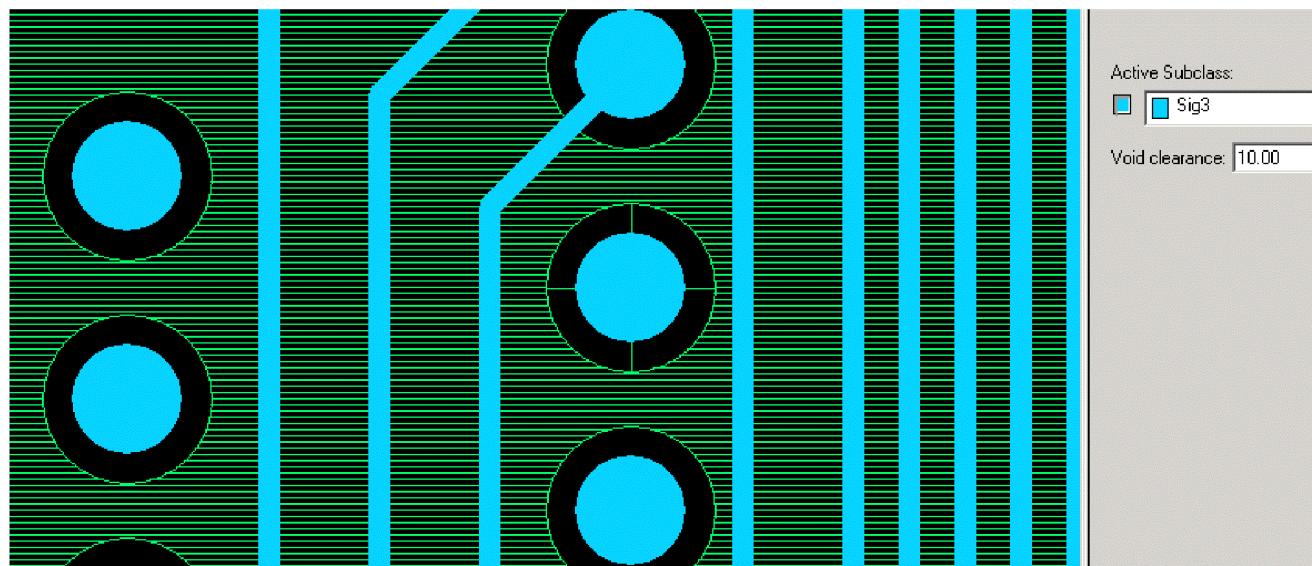
Before Spreading between Voids



Spreading between Voids with Zero Clearance



Spreading Between Voids with 10 mil Clearance



Routing High-Speed Circuits

Adding connections to high speed circuits in a design may make it necessary for you to use rat Ts and delay rules that are not present in other nets.

Routing Rat Ts

Rat Ts are logical database (not physical) objects that you can use to insert a branch in a net schedule at a point other than at a component pin. A rat T's physical location is typically an approximate location for a T or a via in the net's interconnect. However, once located in the design, you can use the `optimize_ts` command to further optimize the location of rat T's automatically.

To route rate Ts, choose *Route – Connect* (`add connect` command).

Displaying Timing Feedback

High-speed circuits often require timing constraints to ensure successful routing. Delay rules are timing constraint variations that you can attach to timing-sensitive nets. The layout editor provides you with dynamic timing feedback on nets, extended nets (Xnets), buses, differential pairs, and pin pairs that have properties such as the following attached:

- PROPAGATION_DELAY
- RELATIVE_PROPAGATION_DELAY
- TOTALETCH_LENGTH

See the *Allegro Platform Properties Reference* for the syntax of these properties.

Dynamic timing feedback, a type of heads-up display, lets you determine if connections you are adding or modifying are within the acceptable timing parameters of the properties. The display updates as you route using the following commands:

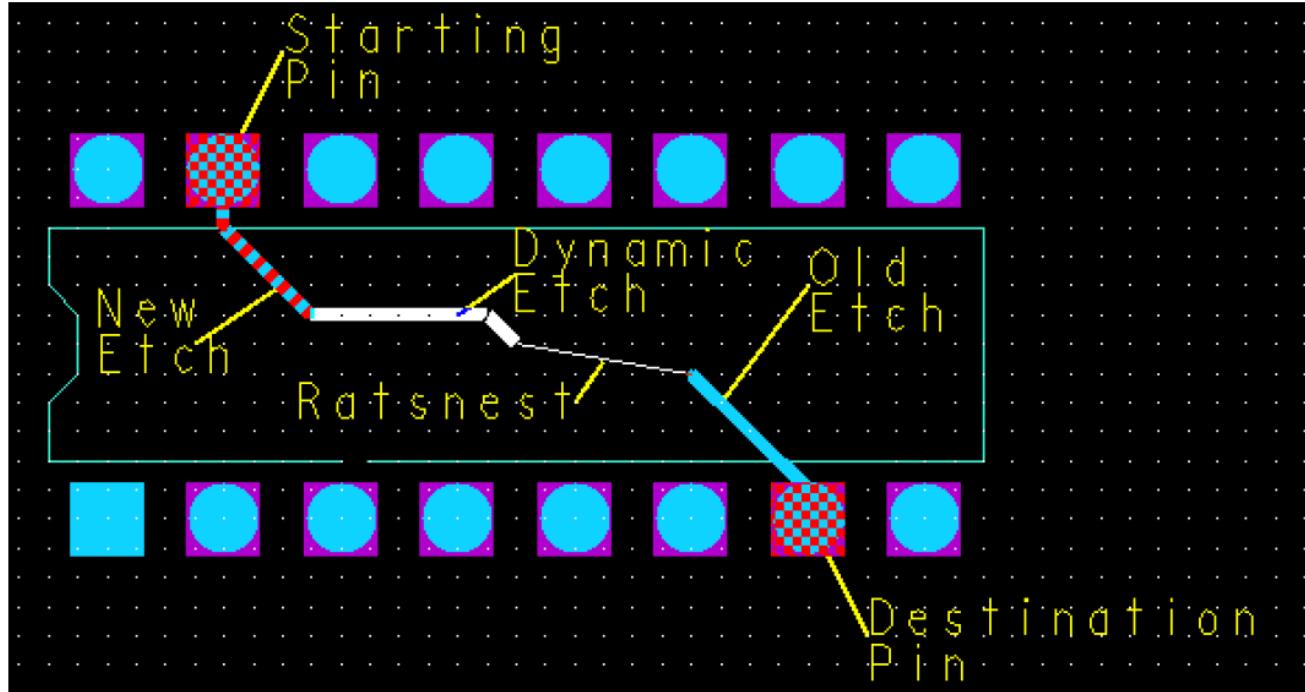
- `add connect`
- `delay tune`
- `slide`
- `vertex`

The feedback displayed shows you whether the etch you are adding or changing is within the acceptable range, determined by the values of the attached timing properties and the overall

proposed etch of the interconnect. Proposed etch, as the `add connect` example shows in the following figure, is the total of all interconnect between start and end points. It includes:

- Existing etch (shown below as old and new etch)
- Dynamic etch (the rubberband display that appears as you move the cursor)
- Expected etch (represented as a ratsnest line)

Components of Proposed Etch



⚠️ Proposed etch/conductor must provide at least some of the connection between pin pairs specified in attached timing constraints. For timing constraints on Xnets, pin pairs do not have to be part of the current net.

In addition to the components of proposed etch described above, timing displays also take into account unrouted portions of a net (or other supported object). The calculation of unrouted portions of a proposed etch assumes routing at manhattan length with current line width on the current subclass. Timing constraints for pin pairs that are not connected (fully routed) have the following conditions:

- Reference pin pairs of a RELATIVE_PROPAGATION_DELAY constraint are assumed to be routed at manhattan length and are assigned the default propagation velocity, which is defined on *Interconnect Models* tab of the Analysis Preferences dialog box, accessed by choosing *Analyze – SI/EMI Sim – Preferences* (`signal prefs` command).

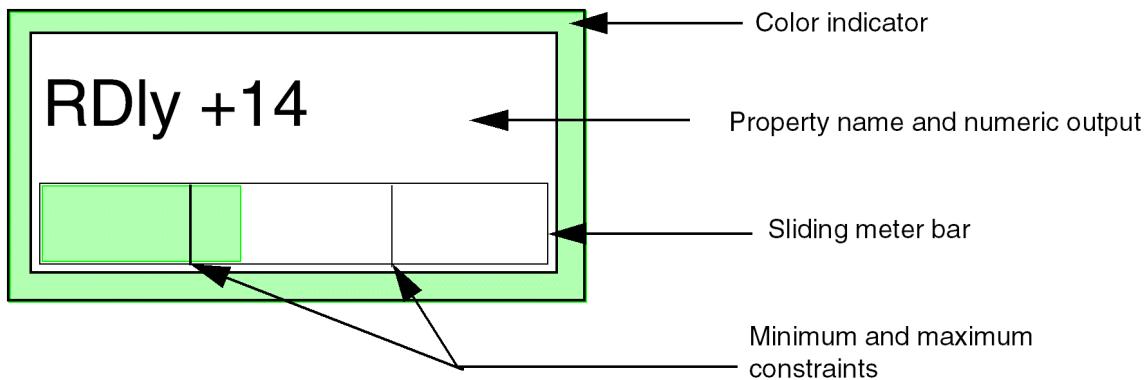
- Unreferenced pin pairs are ignored; no timing feedback is displayed.

The Timing Feedback Window

The dynamic timing feedback window is a heads-up display that provides feedback on how well you satisfy timing constraints when editing an electrically-constrained net. These constraints include properties such as PROPAGATION_DELAY, RELATIVE_PROPAGATION_DELAY, TOTALETCH_LENGTH, and so on.

The following figure shows the basic components of the timing feedback window, and the sections below the figure describe the components.

Dynamic Timing Feedback Window



Color Indicator

The frame of the window and the slide bar show a color representing one of the following conditions. The layout editor uses the length of the current etch/conductor plus the estimated manhattan distance of the unrouted segment when determining these conditions.

Color	Description
Green	Indicates that the current etch is within the minimum/maximum constraint range.
Red	Indicates that the current etch is not within the minimum/maximum constraint range.
Yellow	Indicates that you have a violation, but possibly, it can be corrected by a different as yet unrouted connection.

Numeric Output

The top of the window shows text representing the timing constraint and the numeric output of the timing margin. The display changes as you add, subtract, or move dynamic etch (the interconnect that changes as you move the cursor). The numeric output display has the following characteristics:

- The displayed value of the timing margin is in length user units of the design (mils, inches, and so on).
- The margin is relative to either the minimum or maximum timing constraint, depending on which makes a smaller value. In the following figure, the margin value (+14) is relative to the minimum timing constraint, so it appears on the left side of the display. When the margin is relative to the maximum timing constraint, the value appears on the right side.
- The margin value displays a decimal accuracy based on the accuracy settings in the *Design* tab of the Design Parameter Editor, available by choosing *Setup – Design Parameters* (`prmed` command). The maximum number of digits after the decimal point in the display is the same as the value in the *Design* tab of the Design Parameter Editor. However, trailing zeros to the right of the decimal point are suppressed.

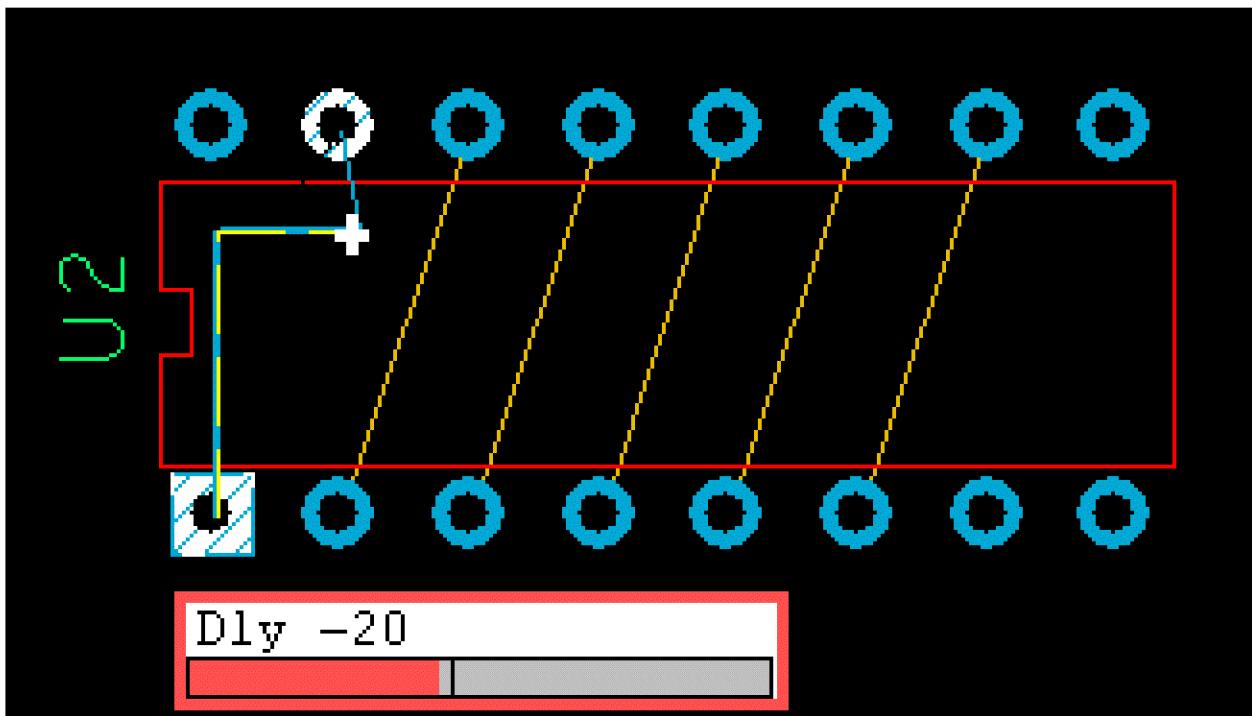
Sliding Meter Bar

The bottom of the window has a meter bar that slides to the right as you increase the proposed etch/conductor length. The meter bar has these characteristics:

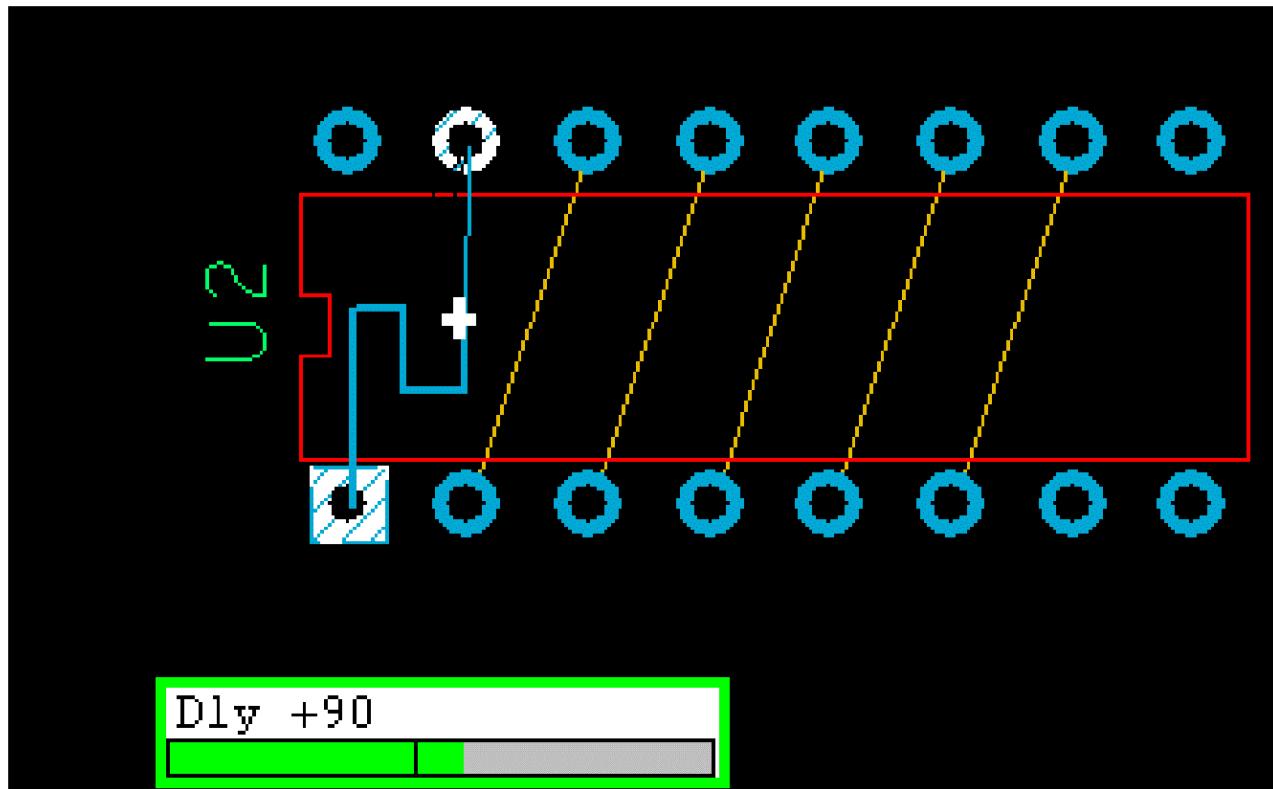
- Vertical lines represent the minimum and maximum timing constraint values. If either a minimum or maximum timing constraint is not set, then only one line, representing the constraint that is set, appears.
- The meter reaches its maximum extent to the left or right if the proposed etch extends too far outside the acceptable range. Beyond that point, you must read the numeric output for valid data.

The following figures show the timing display for conditions when there is only a minimum value set for the constraint setting.

Red Condition – Delay Less Than Minimum Value

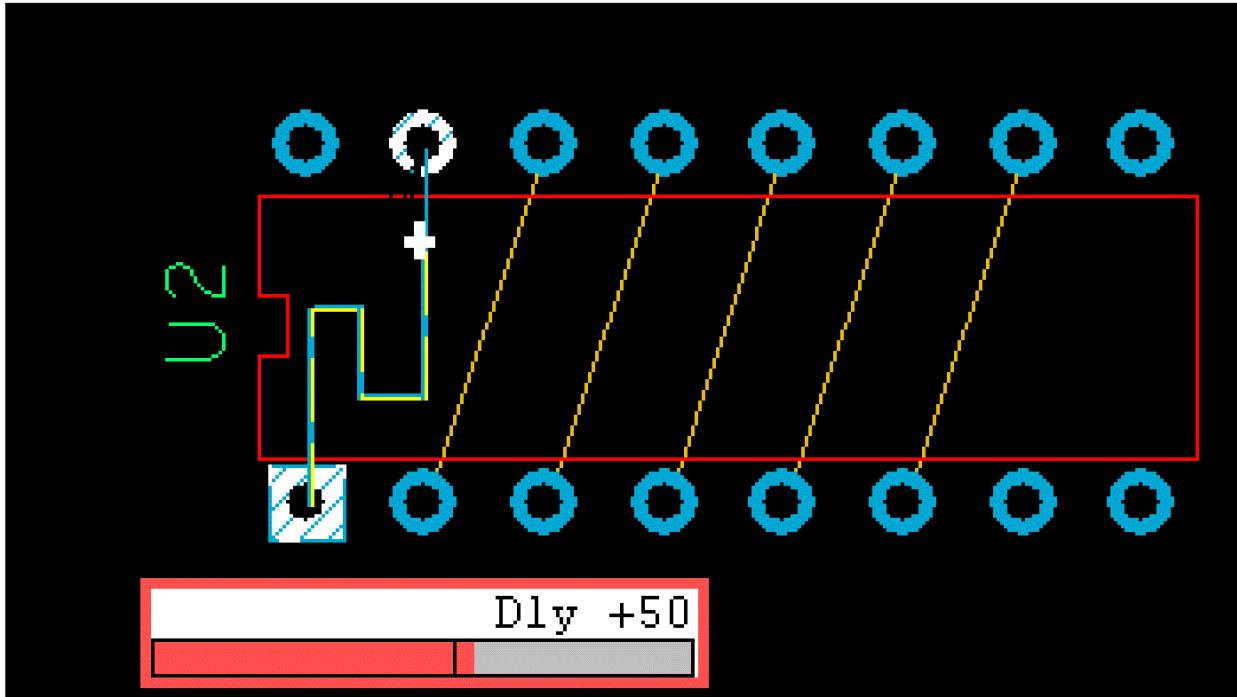


Green Condition – Delay Satisfies the Minimum Value

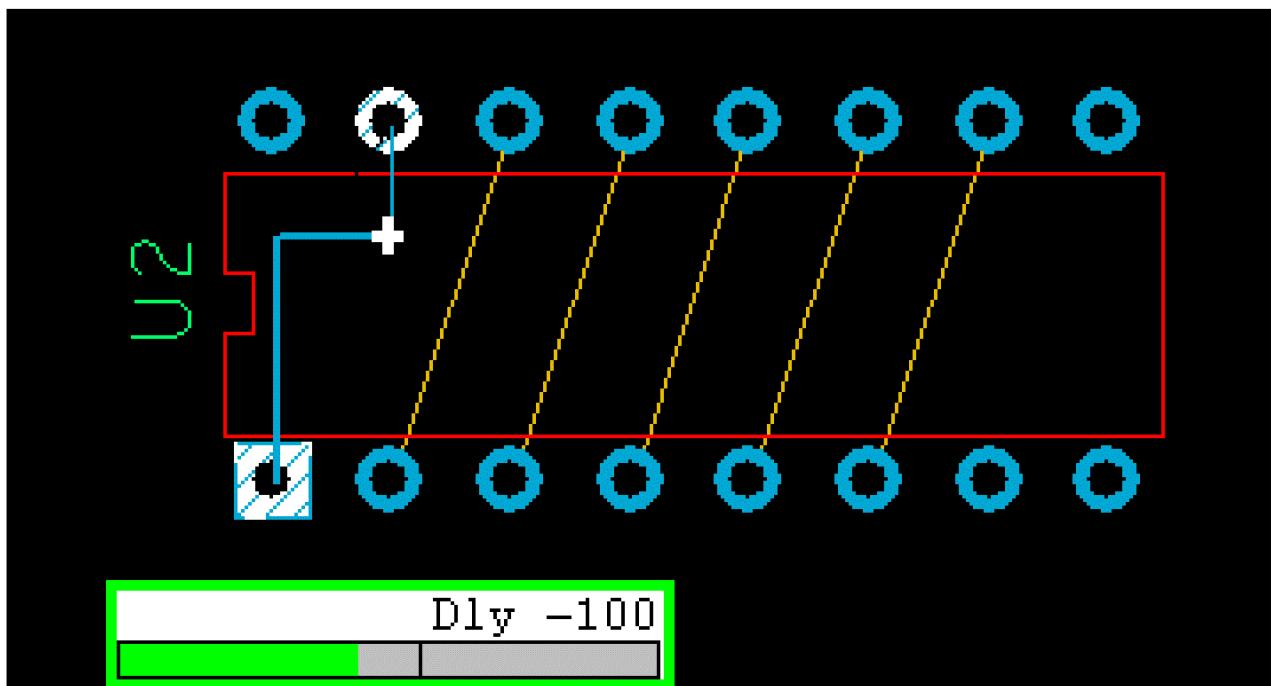


The following figures show the timing display for conditions when there is only a maximum value set for the constraint setting.

Red Condition – Delay Exceeds Maximum Value

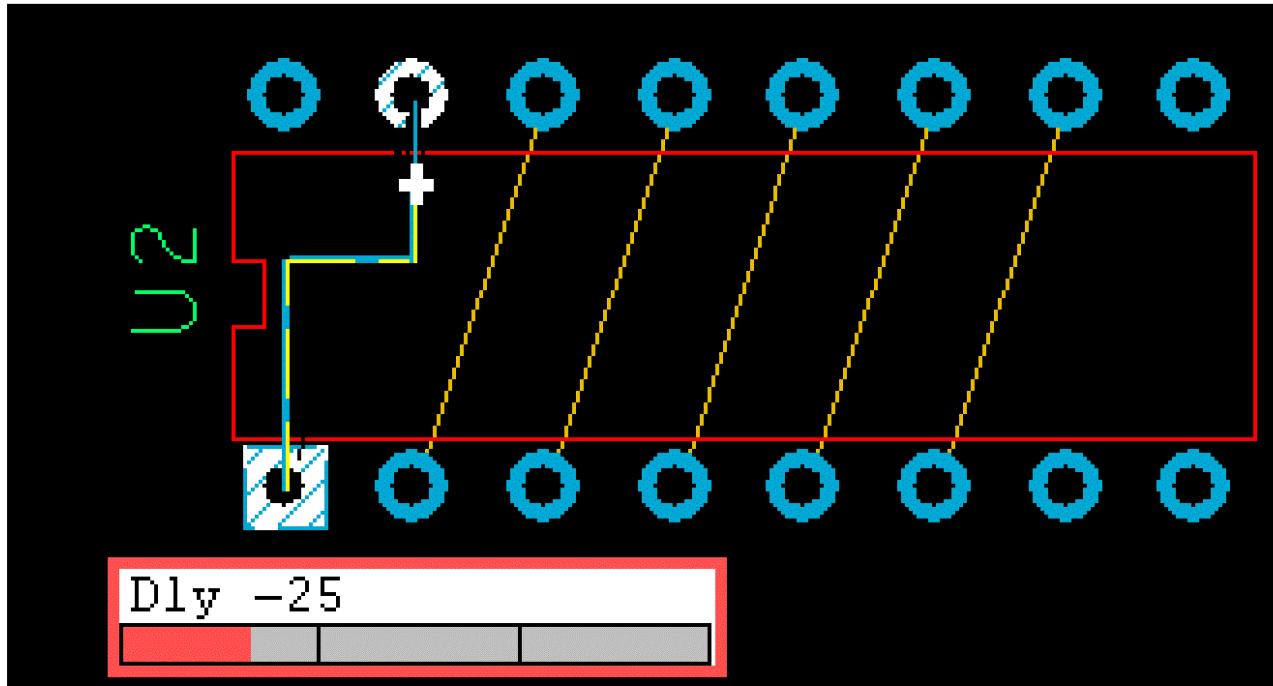


Green Condition – Delay Satisfies the Maximum Value

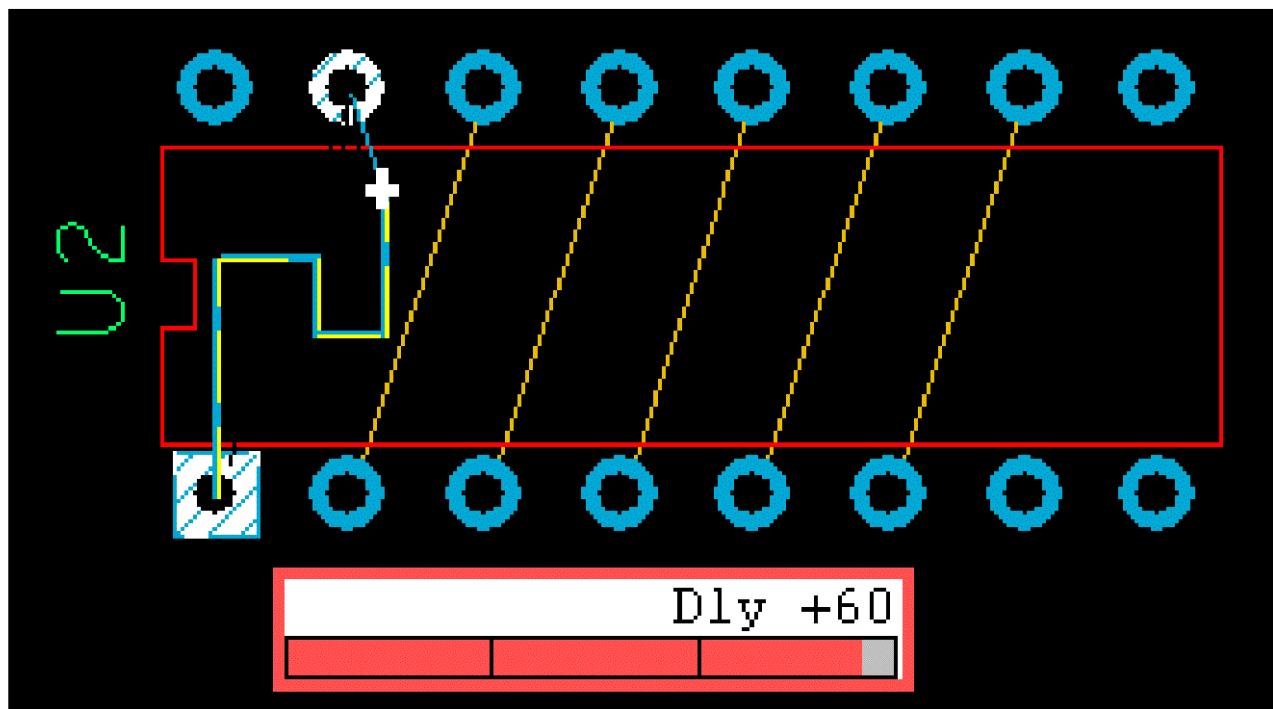


The following figures show the timing display for conditions when both minimum and maximum values are set for the constraint setting.

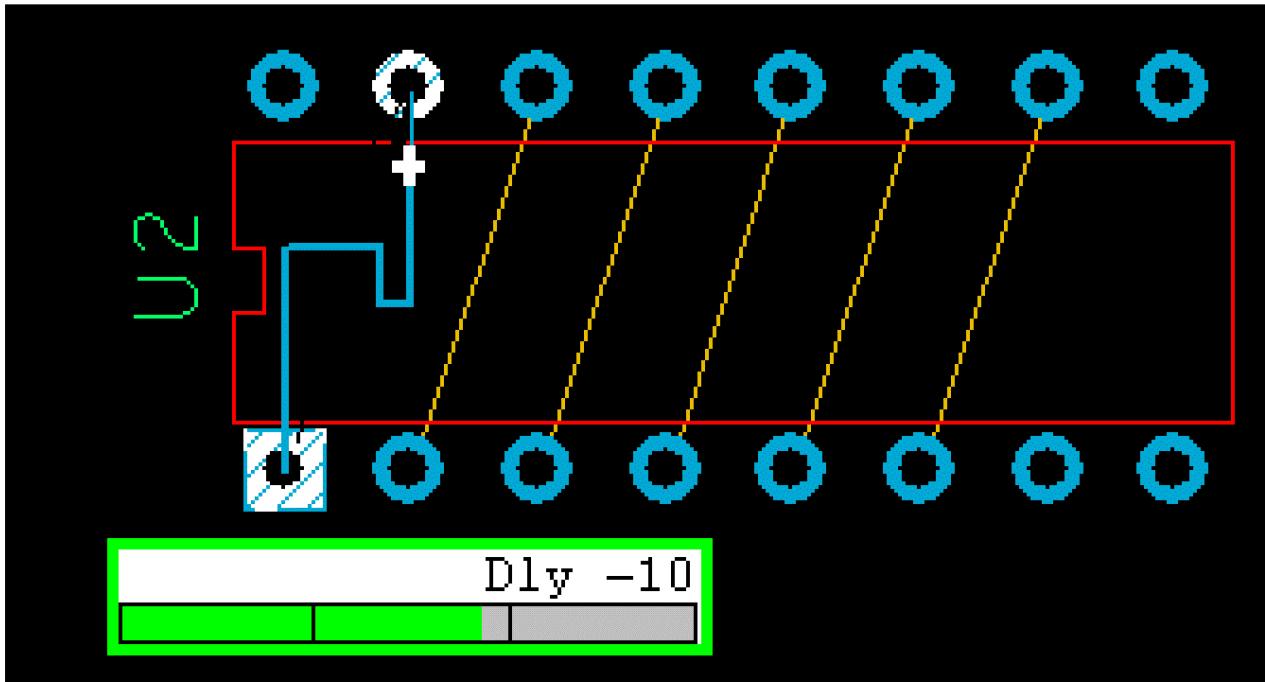
Red Condition – Delay is Less than Minimum Value



Red Condition – Delay Exceeds Maximum Value



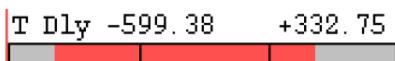
Green Condition – Delay Is in Acceptable Range for Both Min/Max Values



Target Net Identified in Matched Groups

If you create a matched group using relative conditions, for example, `delta:tolerance`, the word `Target` is assigned to the member with the longest manhattan length. The letter `T` has been added to the heads-up display (see below) to indicate that the net being editing is the target net of a relative matched group.

 To obtain this feedback, you must dock the heads-up display.



Docking the Heads-Up Display in the Options Tab

To dock the heads-up display, choose *Setup – User Preferences* from the menu bar and enable the `allegro_dynam_timing_fixedpos` environment variable. When you dock the heads-up display in the *Options* tab in newer versions of the layout editor, each length or timing constraint margin displays its own meter.

Turning Off the Timing Feedback Window

If you do not want to automatically display feedback on timing-sensitive circuits, set the `allegro_dynam_timing` environment variable in the User Preferences Editor to *off*. Choose *Setup – User Preferences* (`enved` command) to access the User Preferences Editor. If you are viewing etch length feedback on circuits that do not have timing constraints attached, using *Route – Connect* (`add connect` command) displays only that information until you unset the variable.

Displaying Timing Feedback With Constraint Manager

If you are working with the layout editor supporting simulation, you can view detailed timing feedback using Constraint Manager. This may be required in instances when complicated timing situations make it difficult for you to route a net using only a worst case feedback display.

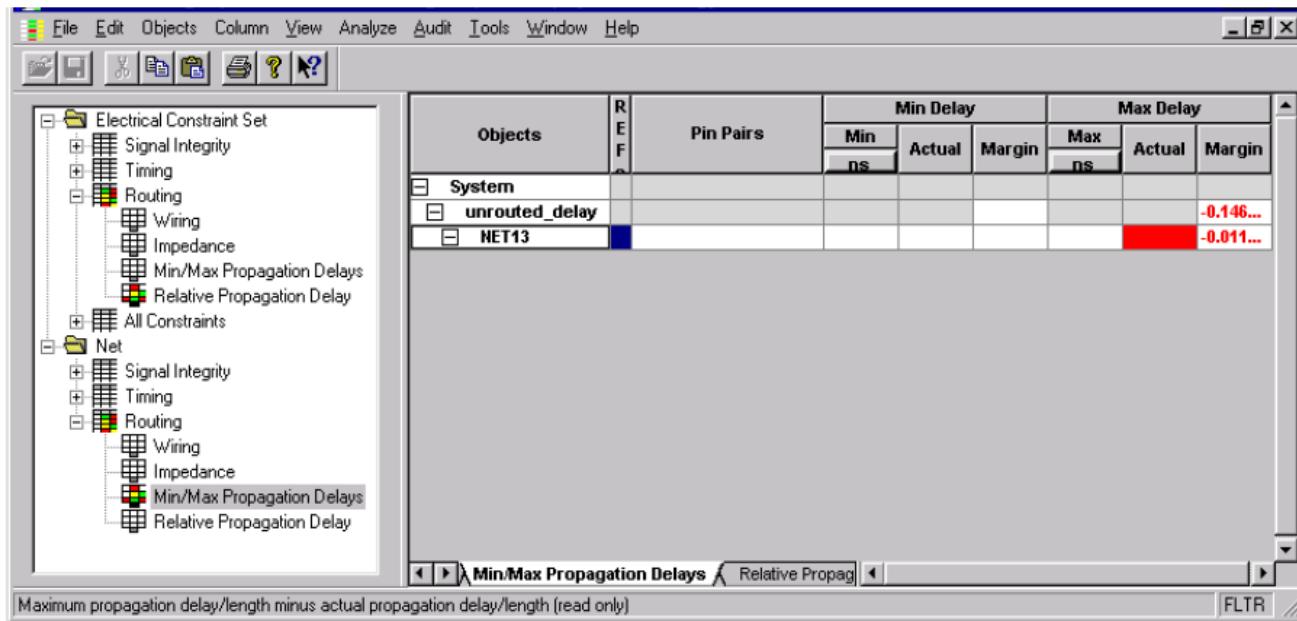
 This section describes only the procedure for viewing timing feedback in Constraint Manager. For complete information on Constraint Manager, see the Constraint Manager documentation.

To display timing information in Constraint Manager:

1. Choose *Setup – Constraints – Electrical* (`cmgr_elec` command) to open the Electrical worksheet of Constraint Manager.
2. In the tree view, expand the *Net* folder and then the *Routing* icon.
3. Open the timing worksheet appropriate to the timing constraint you are using, for example: Min/Max Propagation Delays, Relative Propagation Delay.
4. As an optional step, you can use the pop-up menu in the *Objects* header of the worksheet to open a Filter dialog box.
 - a. Check Selected nets/Xnets only.
 - b. Click *Apply*.
 - c. Click *Close*.

While you are editing the net, its Constraint Manager object is highlighted. If you chose the *Filter* option, only the net you are editing appears. Figure 8-41 shows the Constraint Manager worksheet for a timing-sensitive net. Filtering is used to display only the active net.

Constraint Manager Worksheet



Highlight Limitations

- Only nets with timing constraints are highlighted in the Constraint Manager.
- Constraint Manager has to be open before you begin editing the net for information in the worksheet to be highlighted.
- Only nets and extended nets are highlighted, not pin pairs, differential pairs, or bus objects.
- Differential pair length tolerance is not supported.

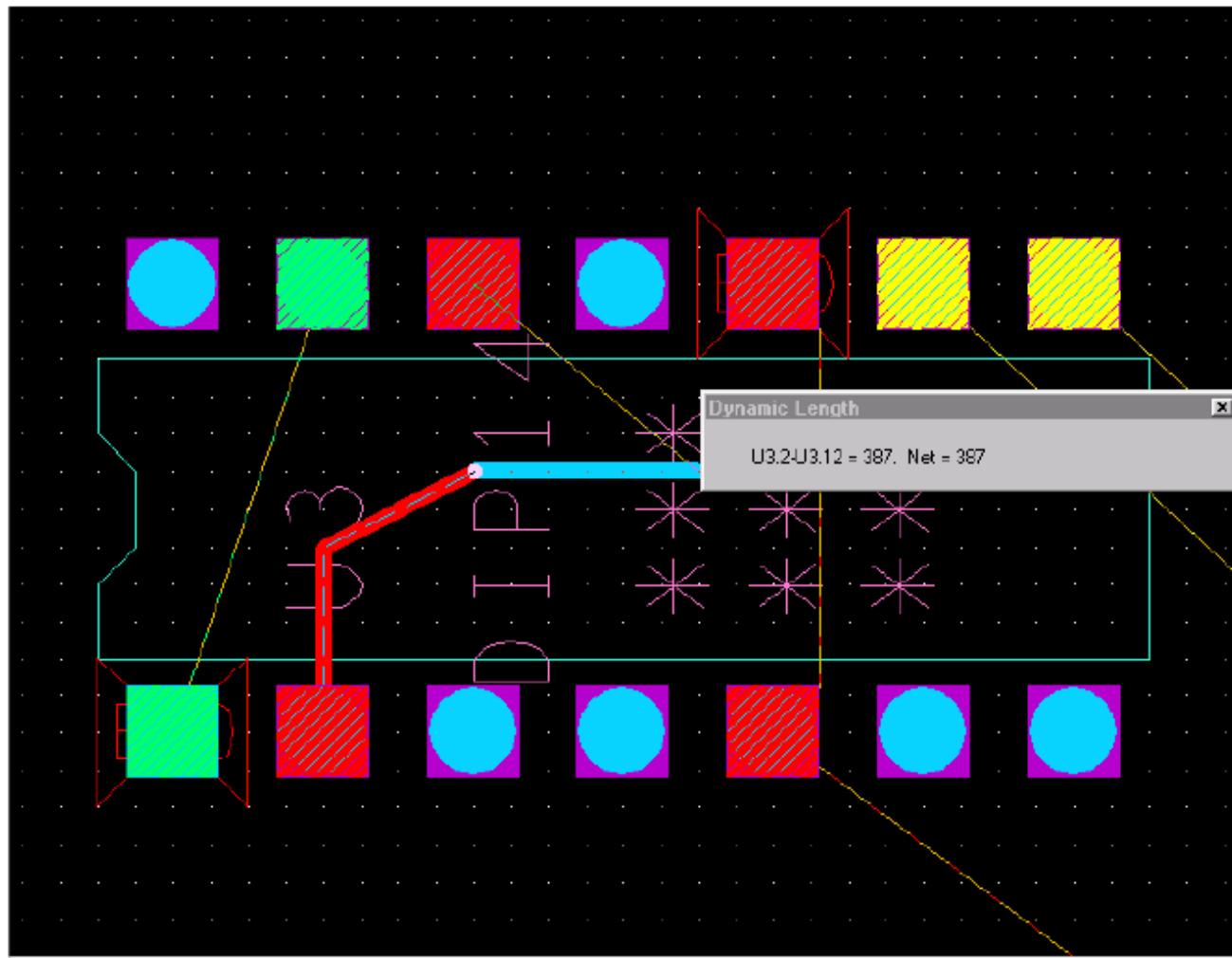
Displaying Etch Length

When you choose *Route – Connect* (*add connect* command), you can receive dynamic feedback on the length of the etch you are adding. The feedback appears when you set the `allegro_etch_length_on` environment variable in the User Preferences Editor. Choose *Setup – User Preferences* (`enved` command). The Dynamic Length window displays the names of the design objects logically connected; for example, ADDR4T.1 (a connect point) to U13.11 (a pin).

The Dynamic Length window displays when you have nets that do not have timing constraints attached to them or because you have set the `allegro_dynam_timing` environment variable to *off*.

The following figure shows the Dynamic Length window.

Dynamic Etch Length



Related Topics

- [net schedule](#)
- [optimize_ts](#)
- [add connect](#)
- [delay tune](#)
- [slide](#)
- [vertex](#)
- [signal prefs](#)
- [prmed](#)
- [Displaying Etch Length](#)
- [Displaying Timing Feedback](#)

Delay Tuning

Nets containing minimum propagation delay, minimum total etch, or relative delay rules often require additional compensation etch to meet these respective constraint conditions. Today, most designs require levels of delay matching between groups of signals, either signals in a data bus or groups of differential pairs. For high-speed serial interfaces, such as PCI Express, the differential pair should be kept in phase across its entire length and not just at the gather points.

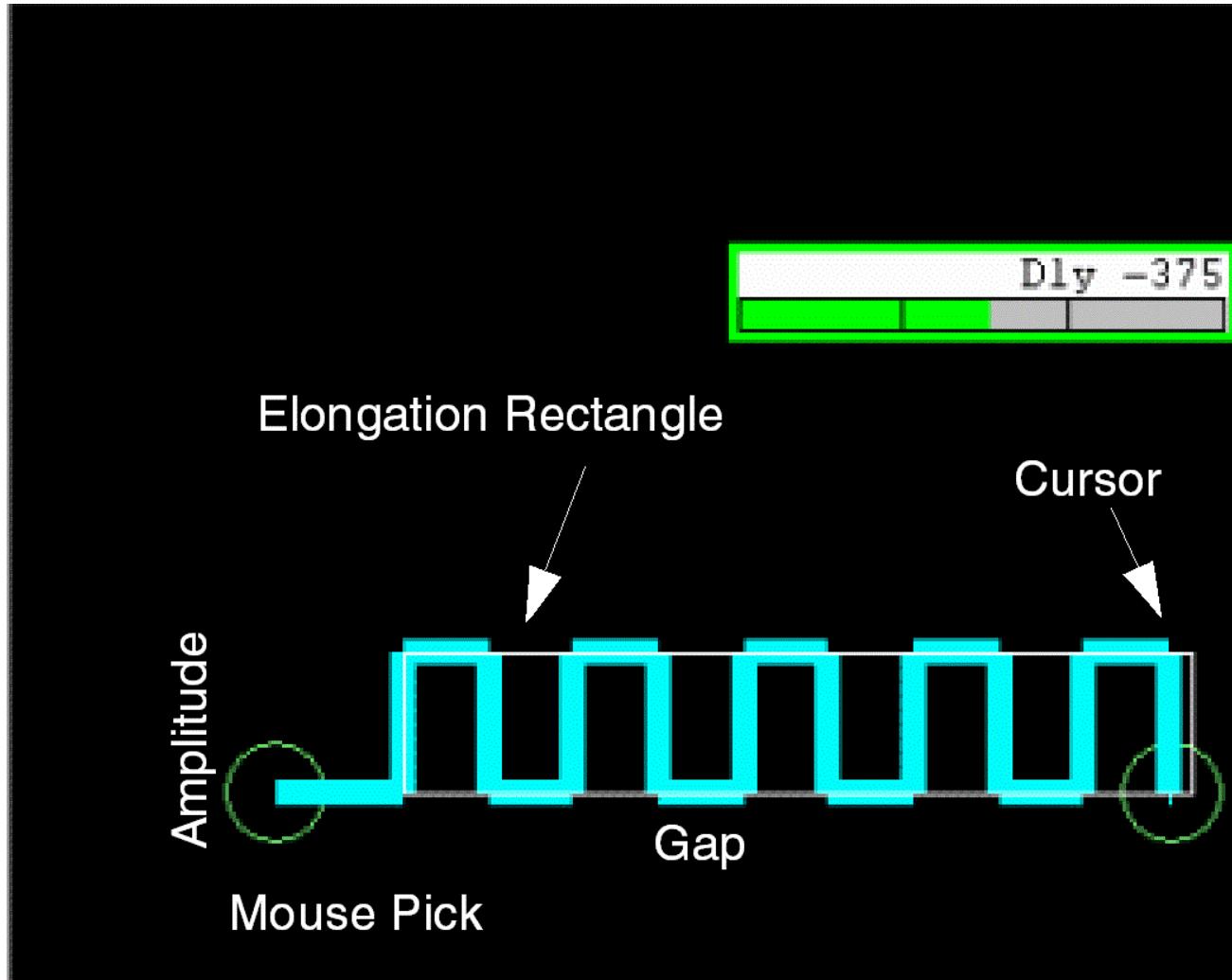
The interactive tuning capability lets you select and control the delay matching style, with the supported accordion, trombone, or sawtooth (preferred for PCI Express) elongation styles, and control gap and amplitude. You can also choose the exact area or areas for tuning. The real-time heads-up display provides direct feedback to guide you during tuning.

Once you determine the violations, either through visible DRC markers or from Constraint Manager information, you can interactively adjust etch length on your delay-constrained nets or differential pairs to match the required delay on the net. To create the delay tuning area, called the elongation rectangle, place your cursor on the line segment that you want to tune. Click the mouse and then using the heads-up display feedback as a guide, move your cursor toward the location that gives the specified results. The elongation rectangle appears, displaying the location, amplitude, and length of the delay tuning area. When you are satisfied with the results, click the mouse to complete tuning.

Elongation Styles

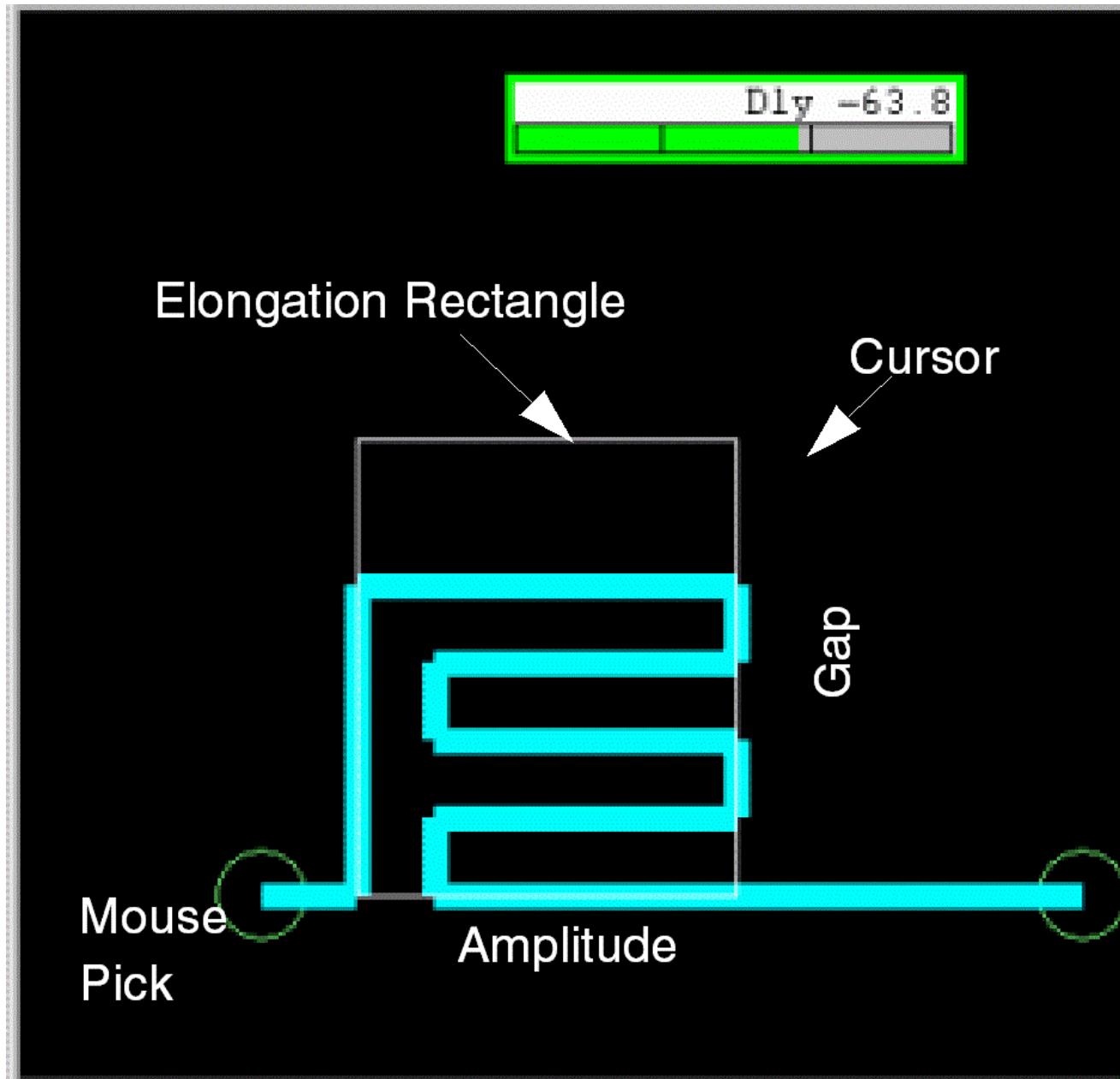
The following figure shows the accordion elongation style.

Accordion Elongation Style



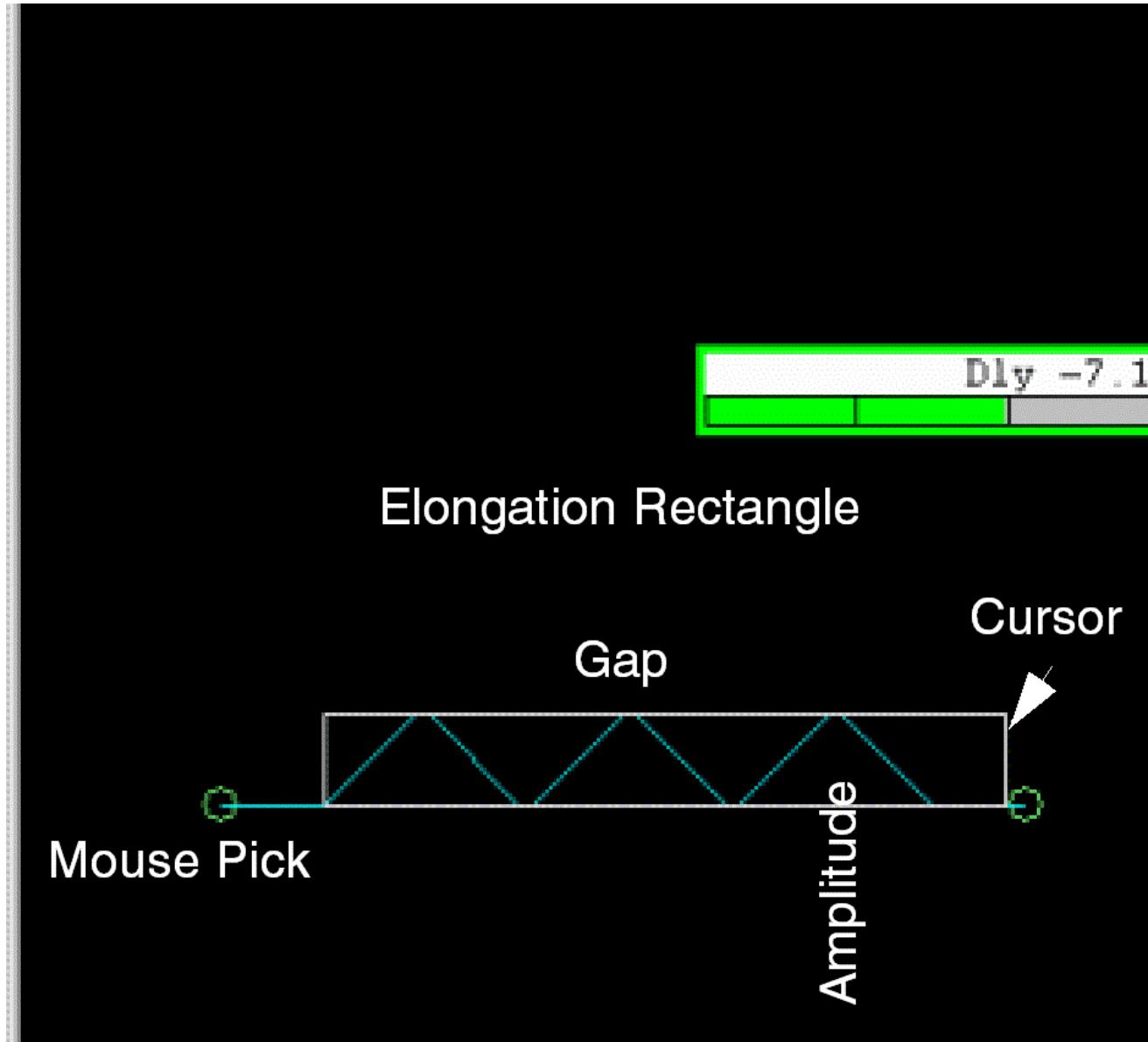
The following figure shows the trombone elongation style.

Trombone Elongation Style



The following figure shows the sawtooth elongation style.

Sawtooth Elongation Style



Although the layout editor ignores constraint areas during delay tuning, you can perform separate elongations on each side of the constraint boundary. The layout editor uses the width of the selected cline segment for the entire elongation etch/conductor. If you specify a gap or corner size using N times the line width, the calculation is based on that same width.

Differential Pairs

Delay tuning for differential pairs is similar to single net delay tuning. You select the cline segment with a cursor pick, and you define the amplitude and direction with the mouse position. The elongation gap requires that the innermost parallel lines of the elongation pattern have the required gap clearance. If the selected cline segment is not currently coupled, or if you have set single trace mode, the layout editor elongates the selected net. When you specify corners, the value applies to the inner segment.

Related Topics

- [delay tune](#)
- [Displaying Timing Feedback](#)
- [Interactive Routing for Differential Pairs](#).

Phase Tuning

Editing or Deleting a Vertex

You can insert vertices (corners) into existing lines. Line elements include connect lines, shapes, and void boundaries. When you choose edit vertex command, the Find Filter window pane is displayed. Fill in the Find Filter window pane as appropriate, and choose the vertex to edit or choose any point along a line segment at which to create a new vertex. APD displays a rubberband from the corners on each side of the vertex. Position the cursor at the new location and click left.

For additional details, see *Creating or Moving Vertices*.

Deleting a Vertex

You can delete vertices of line and arc segments, but you cannot delete vertices of rectangles. For additional details, see *Deleting Vertices*.

Changing the Width of Clines

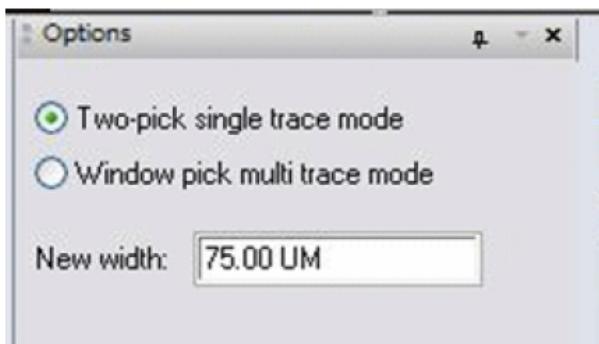
In IC package design, sometimes it is necessary to change a width of a cline in a specified region either for signal integrity reasons or to allow the traces to pass through a dense region. Although, you can make some changes by using constraint areas, glossing, and the change command, there are limitations. For example, constraint areas can be time-consuming to construct. If a cline serpentine around itself for delay reasons, it is difficult to accurately specify the region. And you can only use the change command to modify an entire segment; you cannot add new vertices.

The *Edit - Cline Change Width* menu command ([cline change width](#)) lets you update key sections of your design more efficiently (for example, the fanout area beneath a flip-chip die).

How the Cline Change Width Command Works

When you run this command, using the parameters in the *Options* window pane of the Control Panel, you can either select a start and end point on a single cline to change the width of the cline between those two points or you can window select a region and change the width of all clines within the specified region.

Change Cline Width Parameters in the Options Window Pane



Before and after Cline Change Width



When to Use the `cline change width` Command

Use the `cline change width` command when you have almost completed the design of a certain area, and you will no longer be extensively modifying the clines.

However, if you prematurely use this command and then have to make extensive modifications, run the `cline change width` command, set the new width to the default line width for the specified layer, and window the entire area. All the clines are restored to the default line width, and you can begin necking and fattening traces again.

For additional information, see the [cline change width](#) command.

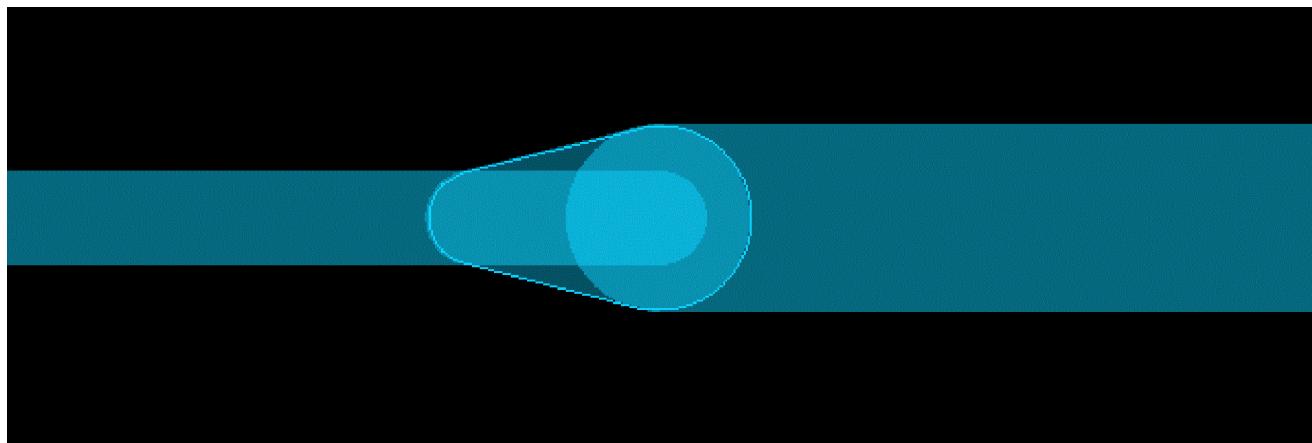
Adding Teardrops Interactively

Adding Teardrops Automatically

Tapering Traces

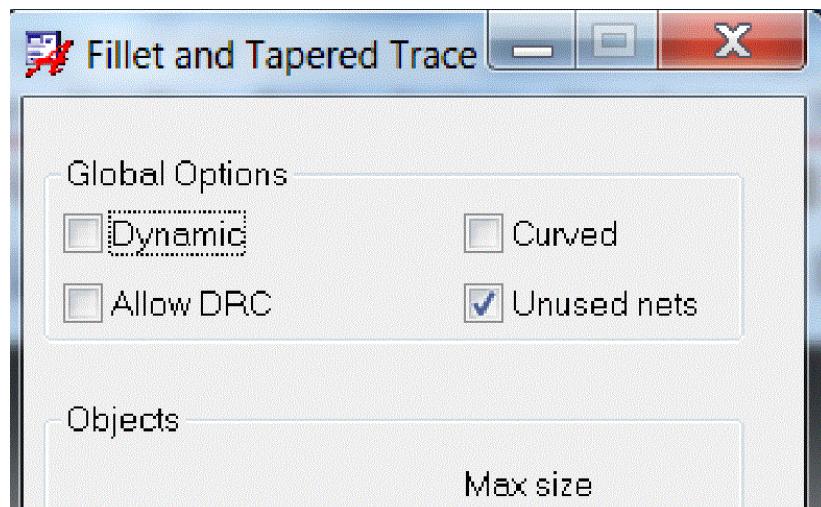
Sharp edges are created when a cline consists of segments of different widths. You can taper such cline segments to smoothen the transition. The purpose of tapering is to prevent abrupt changes in line width which is common in RF and Rigid Flex applications. Tapering reduces the stress at the location of the line width transition.

When you add a tapered trace, on the two cline segments of different widths a fillet shape is added.



You can taper the clines using the [add taper](#) command from the *Route – Teardrop/ Tapered Trace – Add Tapered Trace* menu. To create or modify tapered traces automatically during interactive routing choose *Dynamic* option in the *Fillet and Tapered Trace* dialog box.

⚠ Turning off the *Dynamic* option in the *Fillet and Tapered Trace* dialog box, deletes existing tapers on moving/sliding clines.



<input checked="" type="checkbox"/> Circular pads	100.0
<input checked="" type="checkbox"/> Square pads	100.0
<input checked="" type="checkbox"/> Rectangular pads	100.0
<input checked="" type="checkbox"/> Oblong pads	100.0
<input checked="" type="checkbox"/> Octagon pads	100.0
<input type="checkbox"/> Complex pads	
<input type="checkbox"/> Pads as shapes	<input checked="" type="checkbox"/> Pins
<input checked="" type="checkbox"/> Pads without drills	<input checked="" type="checkbox"/> Vias
<input type="checkbox"/> Bond fingers	<input checked="" type="checkbox"/> T connections

Fillet Options

Fillet Object:	All
Desired angle:	90
Max angle:	90
Max offset:	25.0
Max arc offset:	
Min arc offset:	
Min line width:	3.0
Max line width:	25.0

Tapered Trace Options

<input type="checkbox"/> Tapered traces	
Desired angle:	60



-  Teardrops and tapers are non-editable objects. To view properties, run the `show element` command.

Glossing a Design

You can gloss a design interactively or in batch mode from the APD command line.

To gloss a small area of the design or run one of the faster types of gloss, run it in the Design Window completes quickly. If you are running a complete execution of line and via cleanup, batch mode is most efficient.

When you run glossing, APD writes statistics into a log file, gloss.log. Check this file for warnings and errors encountered during the glossing process.

To gloss your design, choose *Route – Gloss – Parameters* ([gloss param](#) command).

Route - Gloss - List ([gloss area list](#) command) displays the LIST AREA dialog box showing the current glossing mode and the areas selected for automatic glossing.

Interactive Routing for Differential Pairs

Differential pair routing applies to edge coupled differential pairs. An edge coupled differential pair consists of two signals that are routed side-by-side on the same layer. You can perform routing on the differential pairs after you set up the nets as a differential pair.

Once you set up the differential pair, you can route the paired nets concurrently (default) or in single trace mode. When you concurrently route or edit two nets of a differential pair, the specified net is routed or modified while the companion net automatically follows.

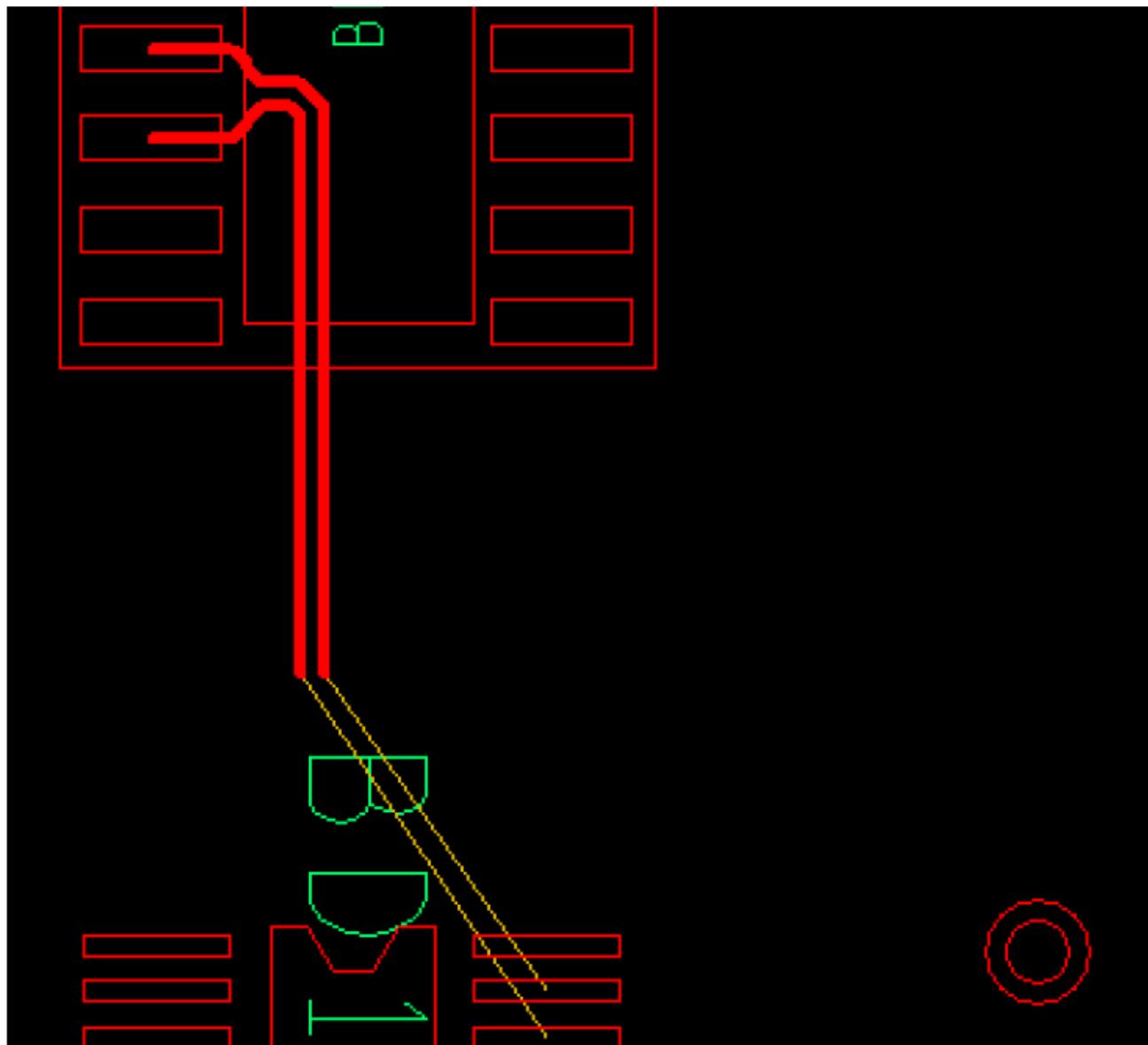
-  Ensure to specify PINUSE property to either IN or OUT for differential pair pins to enable proper routing.

Single Trace Mode

You can also individually route or edit the nets of a differential pair using single trace mode. All the etch edit commands that support concurrent differential pair routing or editing let you enter and exit single trace mode during routing or editing. When you turn on single trace mode, the companion net is immediately dropped. Turning off single trace mode results in differential pair routing or editing as long as the layout editor can identify the companion net.

The following figure shows the routing of a differential pair.

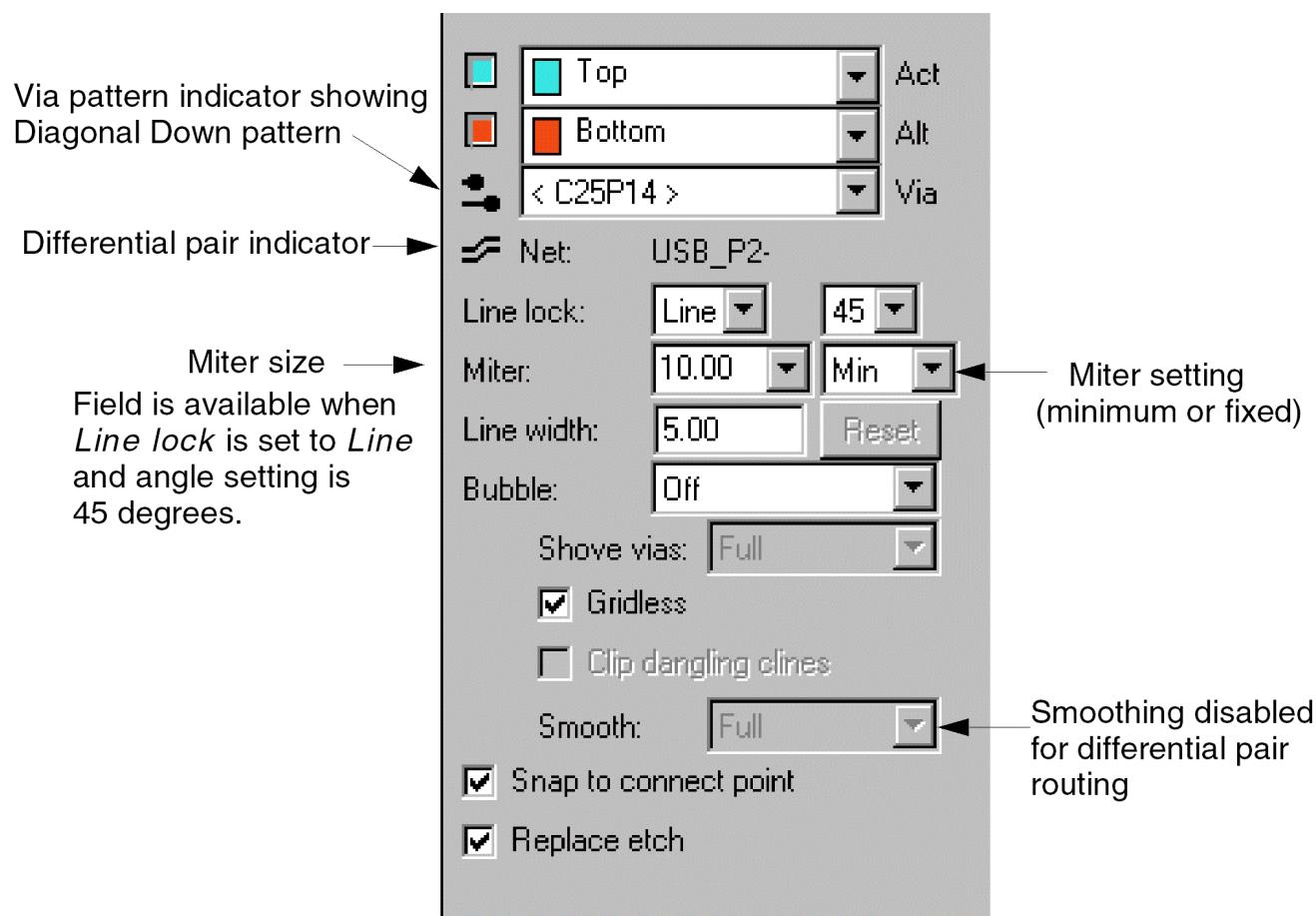
Unfinished Routing of Differential Pair



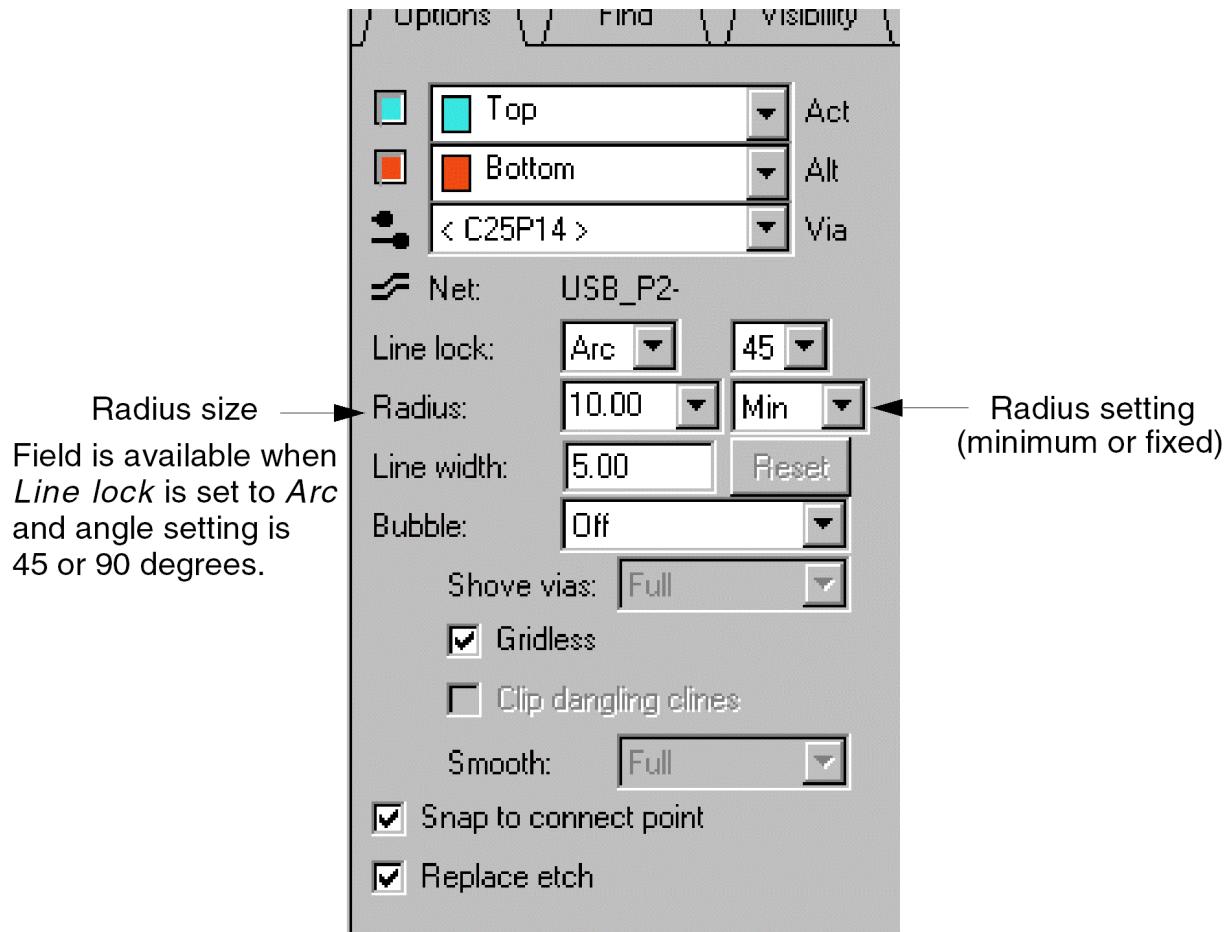
You can manually route differential pairs by choosing *Route – Connect* (`add connect` command). Once you pick a starting point – a pin, via, cline, rat, or rat T – on a net, the layout editor determines whether the net is part of a differential pair, finds a starting point on the companion net, and enters concurrent routing mode.

The following figures show the options available on the *Options* tab when routing differential pairs.

Options Available with Routing Differential Pairs



Options Available with Routing Differential Pairs



Diffpair Driver-Receiver Model Translation

Allegro PCB Router Translator translates the diffpair driver-receiver pin pairs from PCB Editor to PCB Router. Allegro X PCB Editor performs the phase match and uncouple rules calculation based on the driver-receiver pairs. Based on internal pin data, PCB Editor engine then extract all possible pin pairs and evaluates them during diffpair checking and displays them in Constraint Manager. The Translator accounts for the data and maps it into paired wires as part of paired nets.

Related Topics

- [Setting up and Editing Differential Pairs Using the Etch Edit Tools](#)
- [Gathering and Splitting](#)
- [Designating Nets as Differential Pairs](#)
- [Add Connect Command: Pop-Up Menu Options](#)
- [Add Connect Command: Options Panel](#)
- [add connect](#)

Setting Up and Editing Differential Pairs Using the Etch Edit Tools

This section describes how the layout editor routes and edits differential pairs. Topics include:

- [Line Spacing](#)
- [Cornering](#)
- [Grid Snapping](#)
- [Route Necking](#)
- [Single Trace Mode](#)
- [Slide](#)

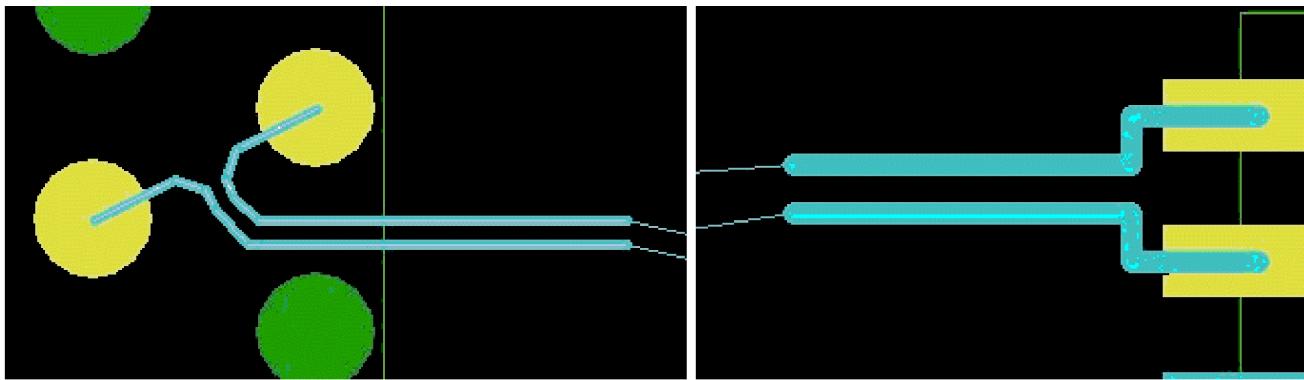
Line Spacing

In handling line spacing for differential pairs during concurrent routing, the layout editor uses the values described in [Differential Pairs](#) in the *Allegro User Guide: Creating Design Rules*. It also uses the values of *Line To Via* and *Via To Via* from the applicable spacing constraint set for spacing between vias to lines and vias to vias. You can also control the via to via spacing when you are in the `add connect` mode and enable via pattern spacing.

The following figure shows how the layout editor gathers traces during differential pair routing. The line lock is 45 degrees in the differential pair on the left side of the figure and 90 degrees in the

differential pair on the right side of the figure.

Differential Pair Spacing



The layout editor tries to maintain the specified gap value during routing and editing except when you choose *Route – Gloss* (`gloss` command).

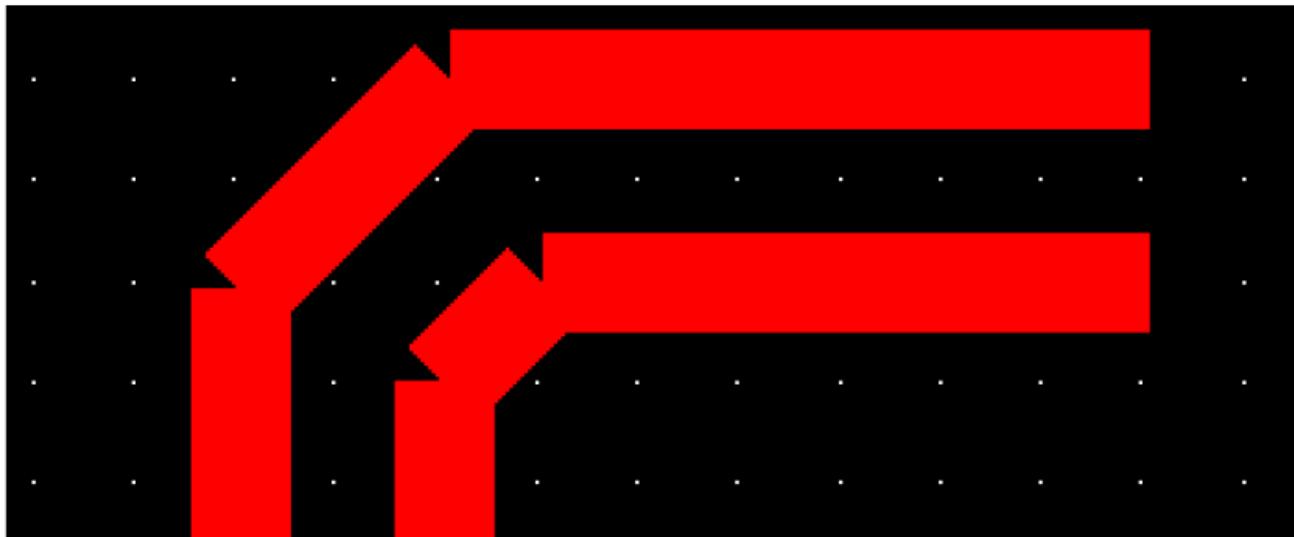
Cornering

With Release 15.0, you now have more control in cornering with 45-degree angles and with arcs.

During editing of differential pairs, based on your options, the specified minimum value for corner size applies unless you specify a miter size. When you specify a minimum or a fixed corner size, it applies to the inner trace.

If you enable the *Line lock* option as *Line* with a 45-degree angle, the corners are mitered even when you position the cursor for a 90-degree turn. The layout editor uses the miter size you specify in the Options tab. The following figure shows an example of mitering.

Mitering Example



You can use arcs to corner differential pairs. For concurrent differential pair editing, the layout editor properly nests the arcs while turning the corner. The nested arcs have a common center point. The trace on the outside of the corner has a larger radius, providing the proper gap for the differential pair. The following figure shows an example of cornering with arcs.

Cornering with Arc



For interactive routing, you can specify the value to be used for smaller radius, either as a minimum or as a fixed radius. Bubble mode is automatically disabled for interactive routing with arcs.

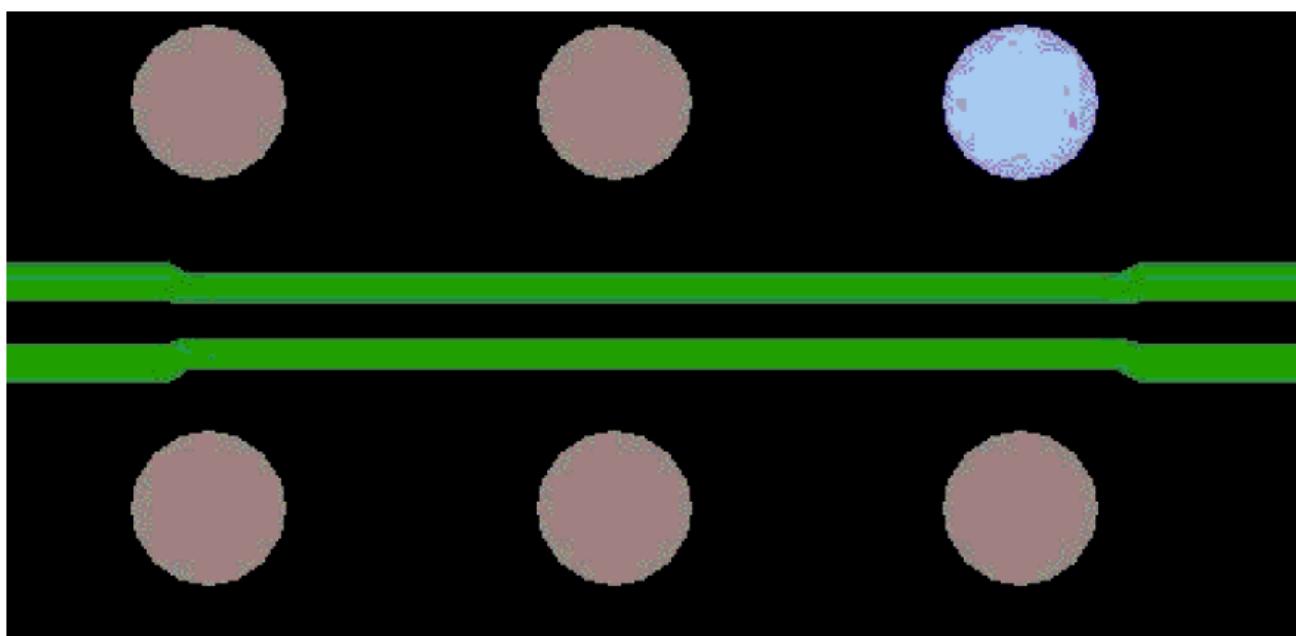
Grid Snapping

The specified net of the differential pair snaps to the specified grid when it can. This means that even if you have specified *Gridless* during setup, the layout editor tries to put at least one of the differential pair traces on grid without causing unnecessary bends or rules violations. When routing differential pairs, the primary gap separation takes precedence over grid snapping.

Route Necking

Neck mode during differential pair routing allows you to route with reduced line width and gap (specified by the *Neck Gap* and *Neck Width* parameters) to keep the differential pair impedance. The layout editor remains in neck mode as long as you continue with the current route. See the following figure.

Neck Mode

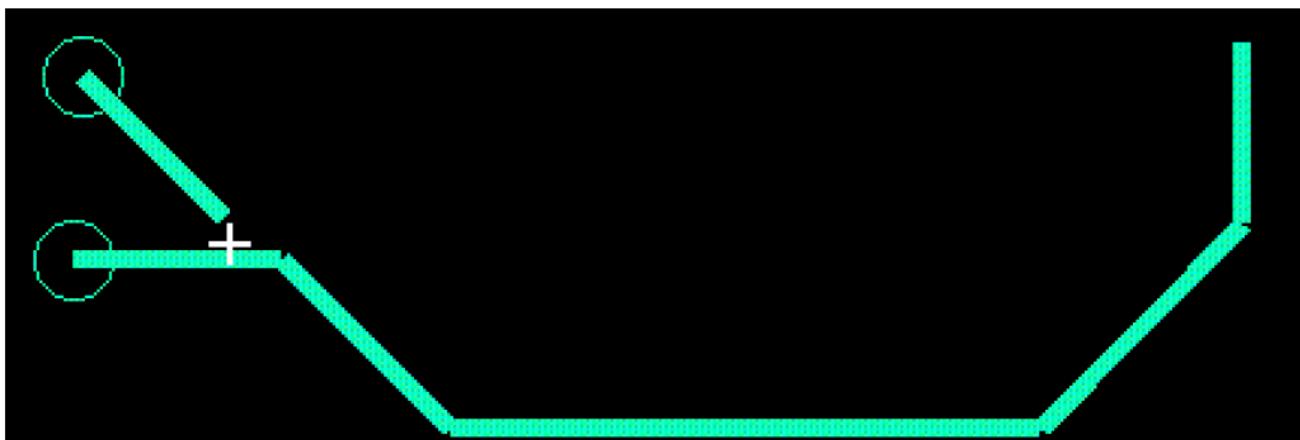


Single Trace Mode

Sometimes, you must route in single trace mode to get around obstacles and complete the differential pair route. During routing or editing of differential pairs, when you enter single trace mode for one command, it causes all the other commands to operate in single trace mode until you turn it off or open another board. When you open another board, the current mode is the one that was in effect when the board was saved. So if you saved a board with single trace mode turned on, when you open the board again, single trace mode is automatically turned on.

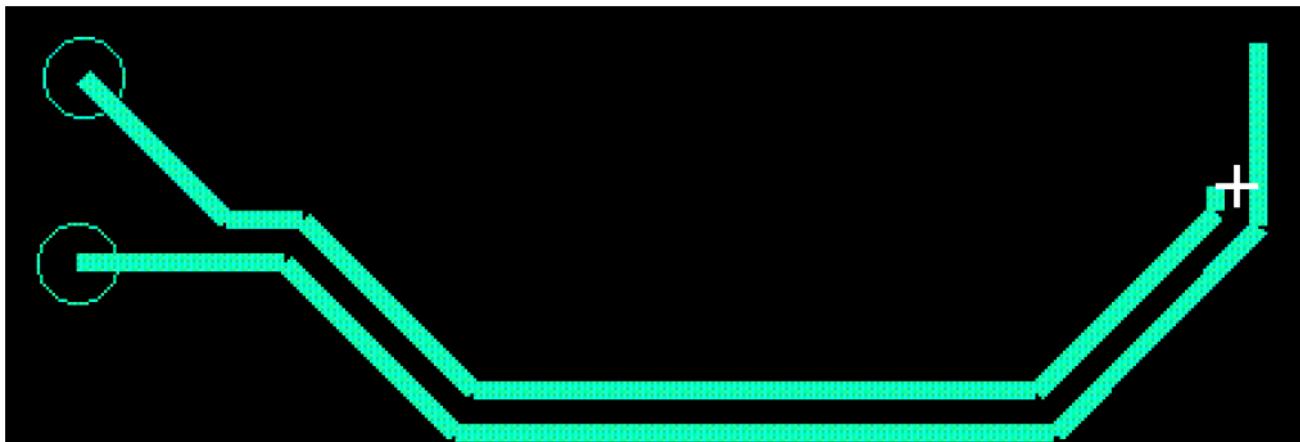
The following figures show how single trace mode works with differential pair routing. The following figure shows that if you position the cursor close to a cline segment of the companion net during single trace mode, the route endpoint snaps to a point that is spaced from the companion net segment by the specified gap value.

Differential Pair Snapping During Single Trace Mode



The following figure shows that if you choose a companion net while in single trace mode, the layout editor follows the route of the first net, while maintaining the specified gap value.

Differential Pair Follows Route



Gathering and Splitting

During routing, the layout editor automatically gathers routes when they do not meet the specified gap value. Gathering is the point where the two nets of the differential pair meet the specified gap value within tolerance for the first time.

The layout editor also splits routes, when necessary, to avoid obstacles or connect with the destinations (vias, pins, and so on). Splitting automatically occurs when you position the cursor at the specified net's destination, unless the destination connect points are spaced by the differential pair gap. The layout editor then attempts to route the companion net to its destination. If the layout editor cannot route the companion net, it automatically enters single trace mode. Then you can finish routing the specified net, followed by the companion net, without having to turn on and turn off single trace mode.

The layout editor gathers together or splits the nets of a differential pair during these times:

- [Routing to and from Pins or Vias](#)
- [Routing with Vias](#)
- [Encountering Obstacles During Routing](#)

Routing to and from Pins or Vias

When routing to and from pins or vias, the layout editor determines the gather point as close to the pins as possible, while adhering to cornering requirements and good pad entry practices.

If the layout editor is unable to gather close to the pins because it would result in a DRC violation, it gathers near the positioned cursor where you define the gathering or splitting point. To lock in the gathering location, you must make a pick. When splitting, the last pick defines the split location.

Finally, if you prefer, you can individually route the nets to or from a gather point using single trace mode. This gives you the most control over the routes to the gathering points, but you have to individually route each signal to or from the gathering point.

Routing with Vias

During routing with differential pairs, the layout editor splits the routes and adds two vias concurrently. While you are in add via mode:

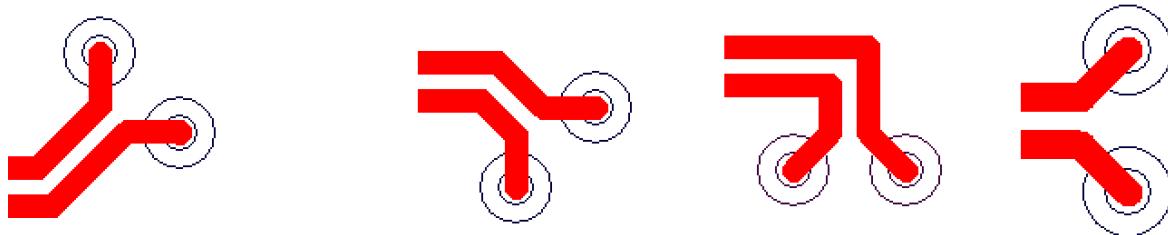
- The two vias connected to the differential pair traces move or float with the cursor.
- Although the vias move with the cursor, the via pattern remains the same. For additional information on via patterns, see [Via Patterns](#).
- You can see the effects of bubble as you specified in the *Options* tab.
- You must pick to place the vias.
- Routing continues with traces gathering as soon as possible.

Via Patterns

The layout editor supports four styles of via patterns. To change a pattern directly, choose a new pattern from the pop-up menu. The layout editor remembers the values and uses them the next time you add vias.

The following figure shows examples of the patterns.

Diagonal Down, Diagonal Up, Horizontal, and Vertical Via Patterns



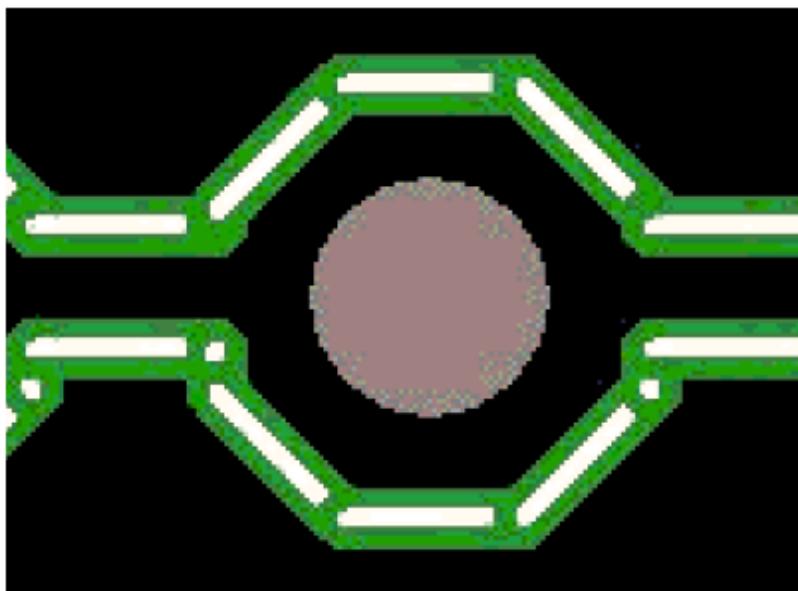
Via Spacing

The layout editor supports three options for via spacing with differential pairs. You can set the spacing for vias using the Differential Pair Via Space dialog box.

Encountering Obstacles During Routing

Differential pair nets are gathered or split when the pair of nets cannot get by as a unit. This can result when you turn on bubble mode. The split lasts as long as obstacles continue to prevent the signals from coming back together.

Gathering and Splitting



Slide

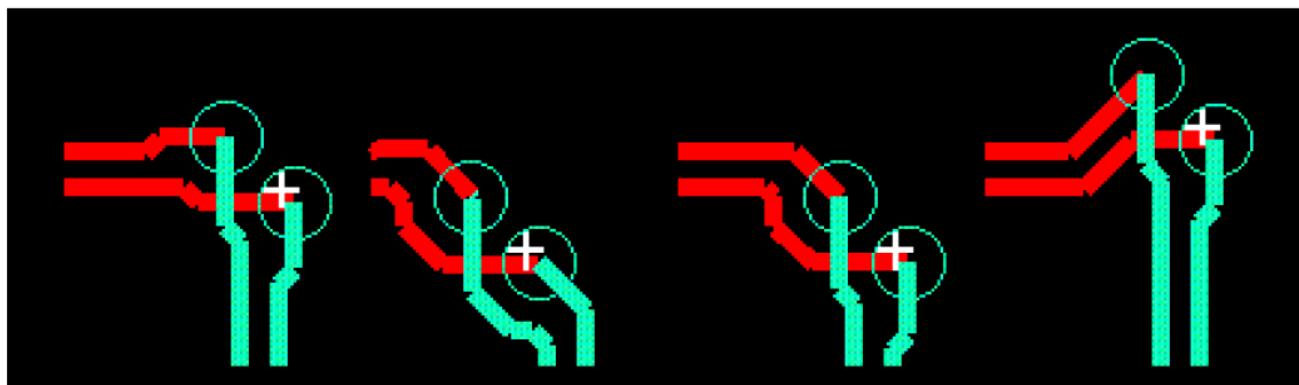
The slide feature supports the differential pair as a single object. For each operation listed below, the layout editor needs to be able to identify the companion net. If you choose elements from multiple nets, the layout editor slides these nets and maintains relative spacing between each element.

- If you choose a segment of a differential pair net, the layout editor chooses those segments. The segments slide together, maintaining the applicable differential pair gap.
- Using the *Cut* option (unavailable in pre-selection use model), you can slide a piece of a line segment on the specified net and a compatible segment on the companion net. The two cut segments slide together.
- Differential pair T sliding is enabled if the pair of physical Ts are associated with Rat Ts. Then the Ts move together with the attached etch. If you choose a physical T and there is no Rat T, the corresponding segments of the differential pair companion move with the specified tracks, but the corresponding T remains in place. You can use the Group Slide feature to move the physical Ts together.
- If you choose a via connected to a differential pair, the layout editor also chooses the companion via. The vias slide together, maintaining the original pattern and space, unless Bubble mode moves them to fix a spacing violation.

- If you choose a segment that connects to a via, and you enable *Vias with Segments* in the *Options* tab, the via slides with the segment. If the segment is coupled with a segment on the companion net, the corresponding via on the companion net also slides.

The following figure shows how vias slide when part of a differential pair.

Sliding of Differential Pair Vias

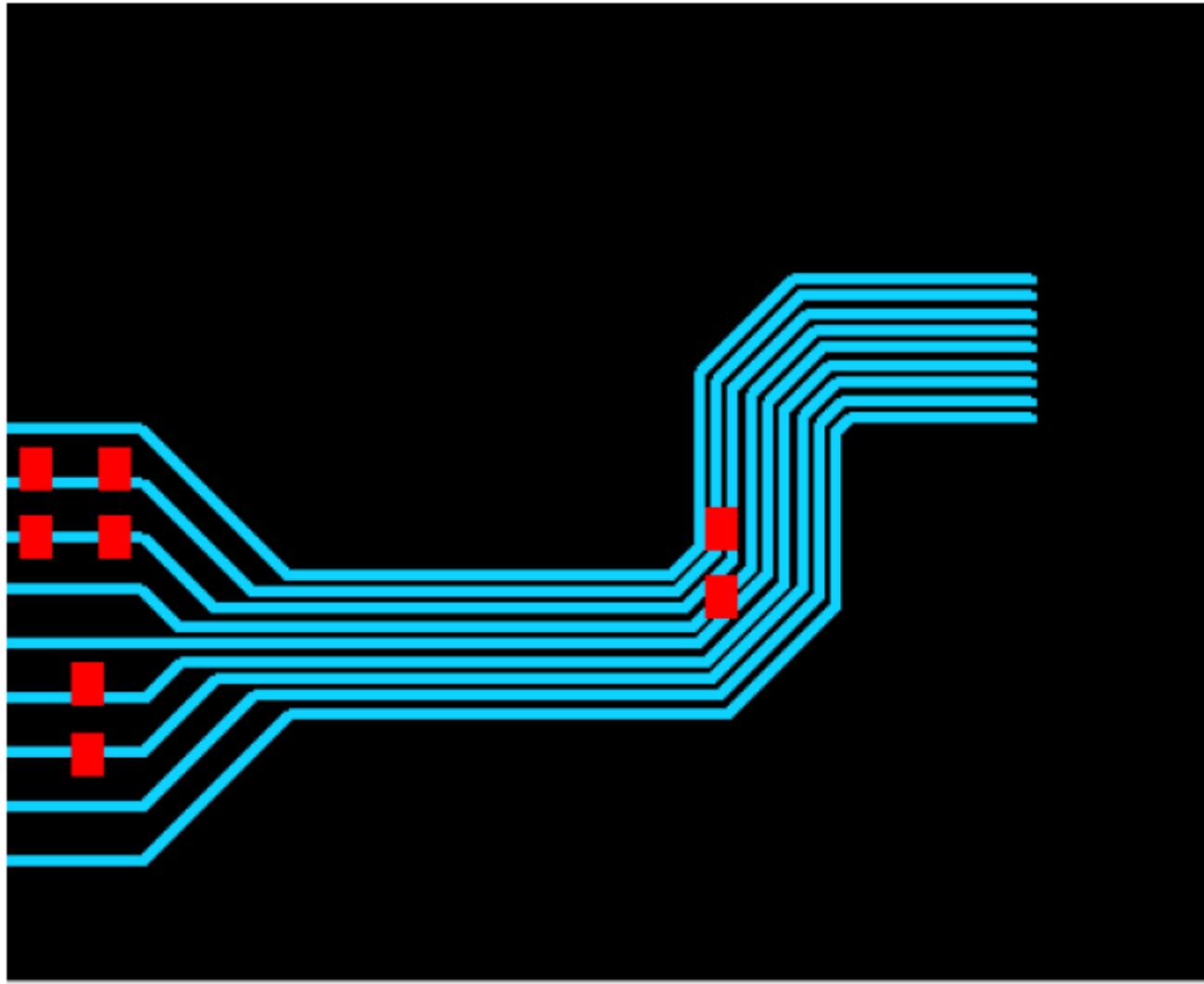


Related Topics

- [cmgr_elec](#)
- [cmgr_phys](#)
- [cmgr_spac](#)
- [Cornering.](#)
- [gloss](#)
- [Add Connect Command: Pop-Up Menu Options](#)
- [Using Single Trace Mode With Differential Pairs](#)
- [Adding Vias to a Differential Pair](#)
- [Changing Via Patterns](#)
- [Changing Via Spacing Using the Diff Pair Via Space Dialog Box](#)

Interactive Group Routing

All tiers of the layout editor support interactive group routing. Interactive group routing is the routing of more than one net concurrently. You can use this feature when routing a bus with traces that follow the same path and have common physical and electrical rules.



To specify the nets for group routing, select the elements (such as clines, pins, vias, and ratsnests) from which to route either by using the *Temp Group* option from the `add connect` pop-up menu or selecting the elements with a window. Routing proceeds from the selected elements.

- ⚠ You can initiate a route by selecting ratsnest lines provided that you have enabled *Ratsnests* in the Find Filter. To reduce the incidence of accidental ratsnest selection, the editor ignores the ratsnests if you also select other types of elements.

⚠ If you are routing from a component with a complicated pin pattern, route from each pin to a location outside the component area. Then group the routes together (outside the component area) in the order that you want to route them as a group— that is, organize the routes outside the component area so that the layout editor can order and space them properly.

The following topics describe how group routing works:

- [Routing Spacing](#)
- [Control Trace](#)
- [Cornering](#)
- [Snapping and Hugging](#)
- [Routing in Single Trace Mode](#)
- [Via Patterns during Group Routing](#)

Routing Spacing

When group routing, you can change the spacing mode between traces to one of the following:

- Current
- Minimum DRC
- User-defined

⚠ Spacing mode is supported only in Allegro Series 600, and Allegro SI.

When you choose *Current* mode, which is the default mode, traces continue at the same spacing with which they started. No gathering or splitting is done unless there is a line-width change. Differential pairs gather, if necessary, to maintain the differential pair gap.

When you choose *Minimum DRC* mode, adjacent traces are separated by the line-to-line space specified in the applicable spacing constraint set. Traces from the same differential pair traces are spaced by the applicable differential pair gap.

You can specify a value for spacing between the traces when you choose *User-defined* and then enter a spacing value. For any trace pair, if the spacing value you specify is less than the *Minimum DRC* spacing value, the *Minimum DRC* spacing value is used instead of the specified value. Traces from the same differential pair are always spaced by the differential pair gap. The layout editor performs gathering or splitting when you change the spacing value.

Gathering and Splitting

Gathering and splitting during group routing occurs when the spacing of your traces does not match the value specified by the spacing mode, for example, the line width changes because you modified the *Options* tab of the Control Panel or you switched to neck mode.

 Gathering and splitting are supported only in Allegro Series 600, and Allegro SI.

Control Trace

During group routing, the editor chooses one of the traces as the control trace. The control trace routes to the cursor location and the other traces follow along with it. You can identify the control trace by the X at the dangling end of the control trace near the cursor.

To determine which trace is the control trace, the editor uses the following:

- If you choose *Temp Group* from the pop-up menu when selecting the elements for group routing, the layout editor chooses the first element as the control trace.
- If you select elements with a window, the layout editor chooses the trace closest to the finish point of the selection box as the control trace.

You can change the control trace for routing when you choose *Change Control Trace* from the add connect pop-up menu.

Cornering

Interactive group routing uses the *Miter* and *Radius* controls in the *Options* tab. As with differential pairs, the *Min* or *Fixed* corner size applies to the inner trace. The control trace corner size adjusts to a larger value if it is not the inside trace of the corner.

Snapping and Hugging

During group routing, snapping and hugging occurs as it does with differential pair nets. The snap-to- or hug-to-cline is established by positioning the cursor close to the cline. This is enabled unless you set *Bubble* mode to *Shove preferred*. To use this feature, you must set the control trace to the trace in the group that is closest to the snap-to or hug-to cline.

Routing in Single Trace Mode

Single trace mode allows you to suspend concurrent routing in favor of routing a single trace of the group. This is helpful for drilling to a different layer or for connecting into or out of a component, for example, getting in and out of a BGA. Single trace mode during group routing works in a similar manner as single trace mode for differential pairs.

When you switch to single trace mode, the companion nets are immediately dropped. Routing continues with just the control trace from the same route-from point. To change the trace used in single trace mode, choose *Change Control Trace* from the pop-up menu. When you switch from single trace mode to group routing, the current trace becomes the group route control trace. When you switch to group routing after adding single trace routes, the other traces either snuggle up to the control trace to catch up to the endpoint of the control trace or are trimmed back to the endpoint of the control trace, depending on which net extends farther.

Single trace mode resets at the beginning of the route, that is, you always start out in multi-trace mode.

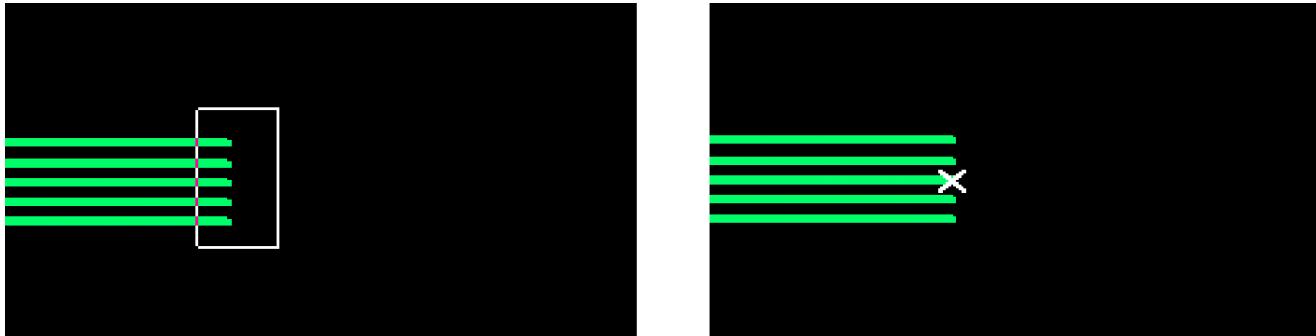
Via Patterns during Group Routing

Via pattern support during group routing is available when you are in the `add connect` command. You can add vias during group routing in both the modes-Alternate mode and Working layer mode. With the *Alternate* use-model enabled, you can select the via from the *Options* tab. With the *Working Layer* use-model enabled, you can pick the target-layer from the *Add-Via* dialog box.

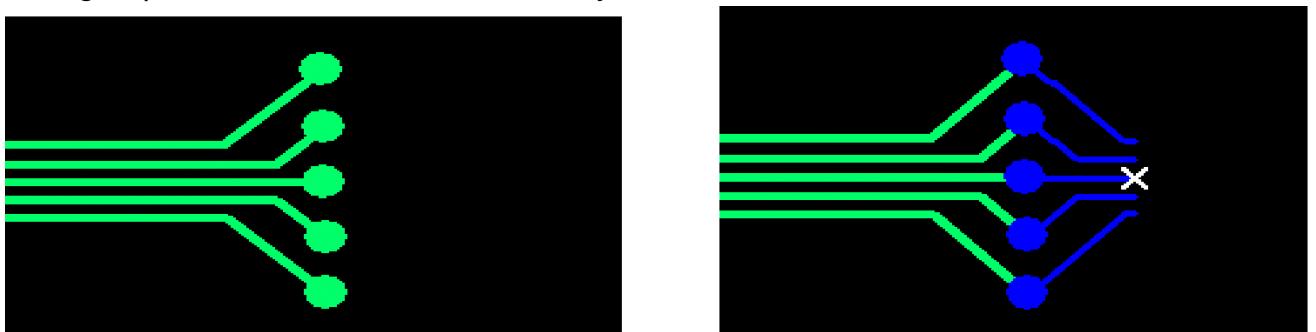
For adding vias in group routing, the same padstack (or via-stack) is used for all selected clines, and is determined by the control-trace. A DRC may appear if a padstack is invalid for one or more of the selected clines.

Adding Via Patterns during Group Routing

1. Select `add connect` command and create group to add vias. In the following figure five cline segments are selected. The control-trace is shown by the white X.



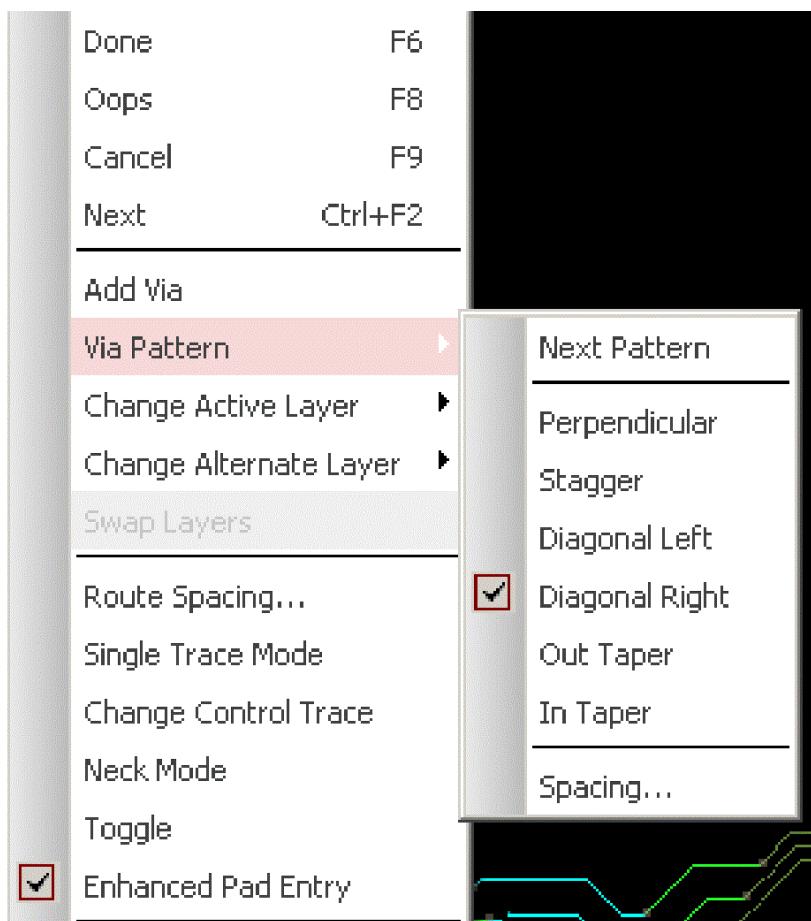
2. Now select via-pattern from pop-up menu and add the via by double clicking the cline segments. The vias remain in the floating state until one additional click is made. However, you can also use the function key while in the floating state. New clines (blue) will gather, and then group route continues on the new layer.



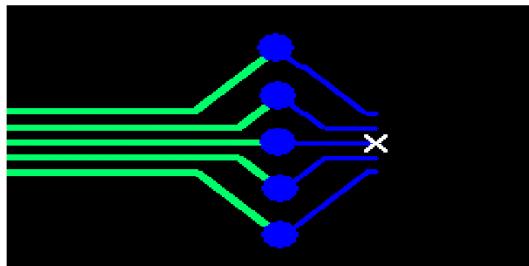
The via-pattern is created, and all the vias will slide dynamically as a group in the direction of the control-trace. The control-trace via is placed directly along the control-trace cline, with no extra vertices added. Extra vertices are added for the other traces if needed.

Types of Via Patterns

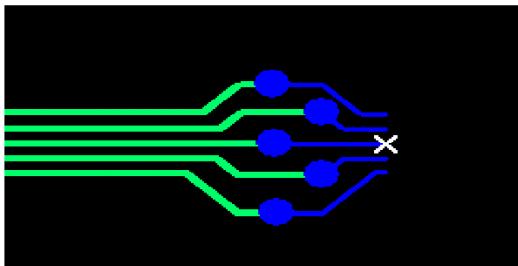
There are six type of via patterns. You can select the via pattern from pop-up menu. The *Next Pattern* option can be used to cycle to the next via pattern in the list.



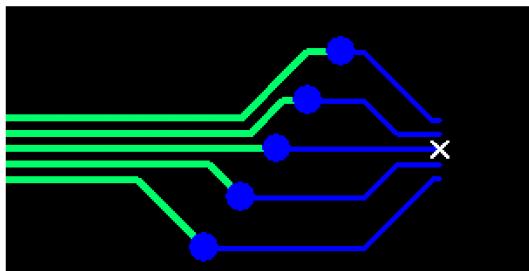
The figure shows the example of via patterns:



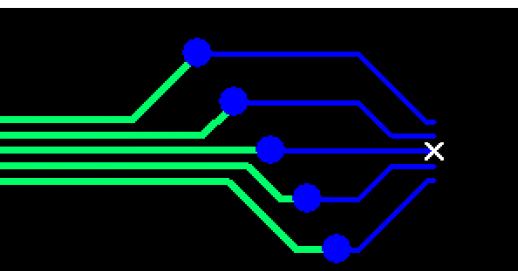
Perpendicular



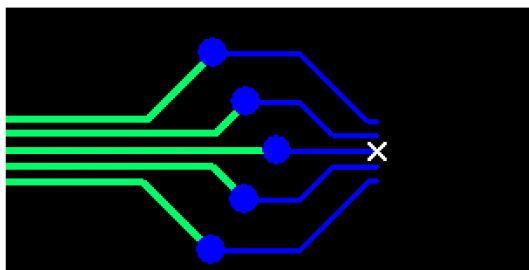
Stagger



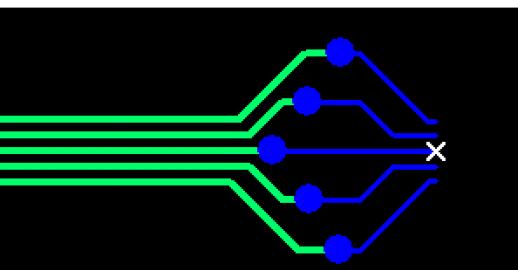
Diagonal Right



Diagonal Left

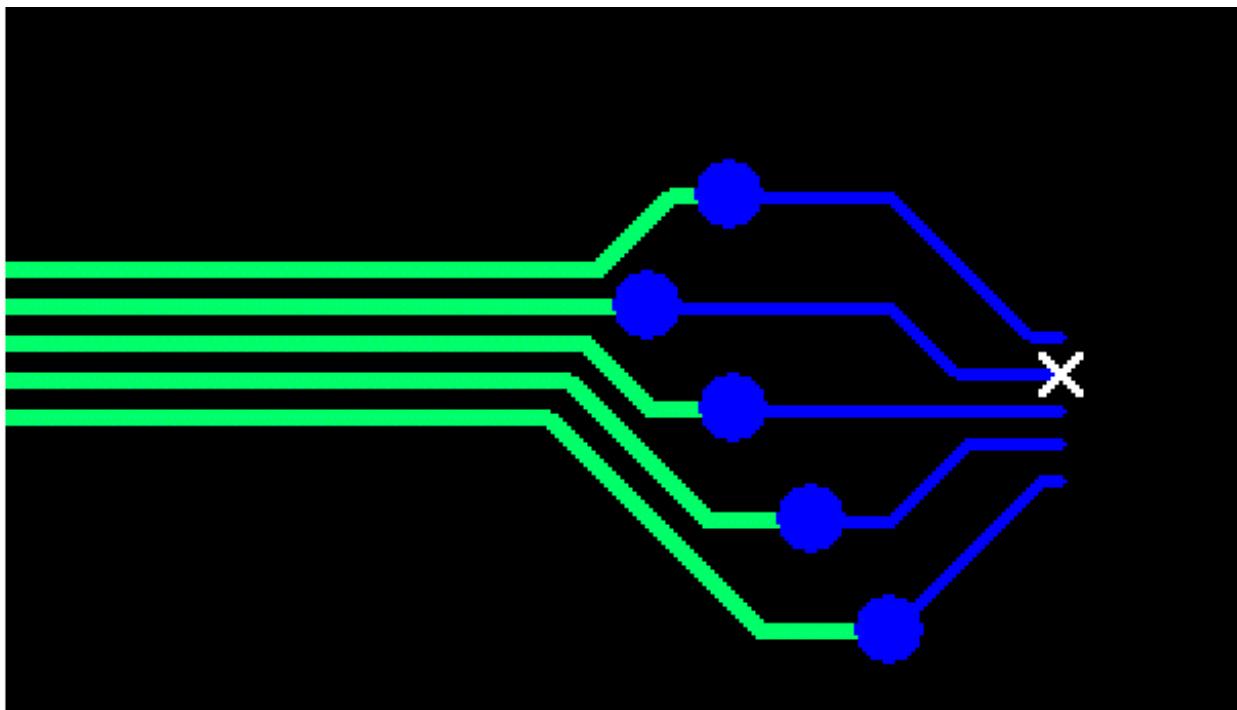


Out Taper

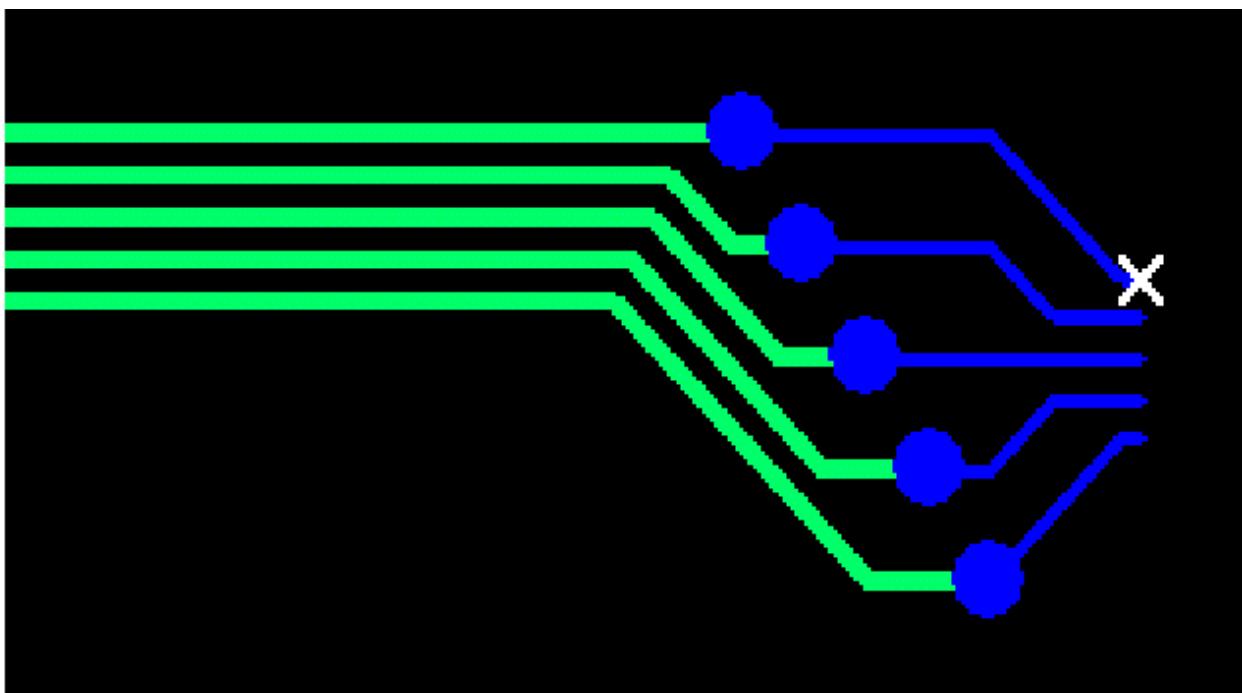


In Taper

The shape of the via-pattern can change depending on which cline is the control-trace. To change the control-trace use pop-up menu. The following figure shows vias added with in taper pattern with the second cline as control-trace.

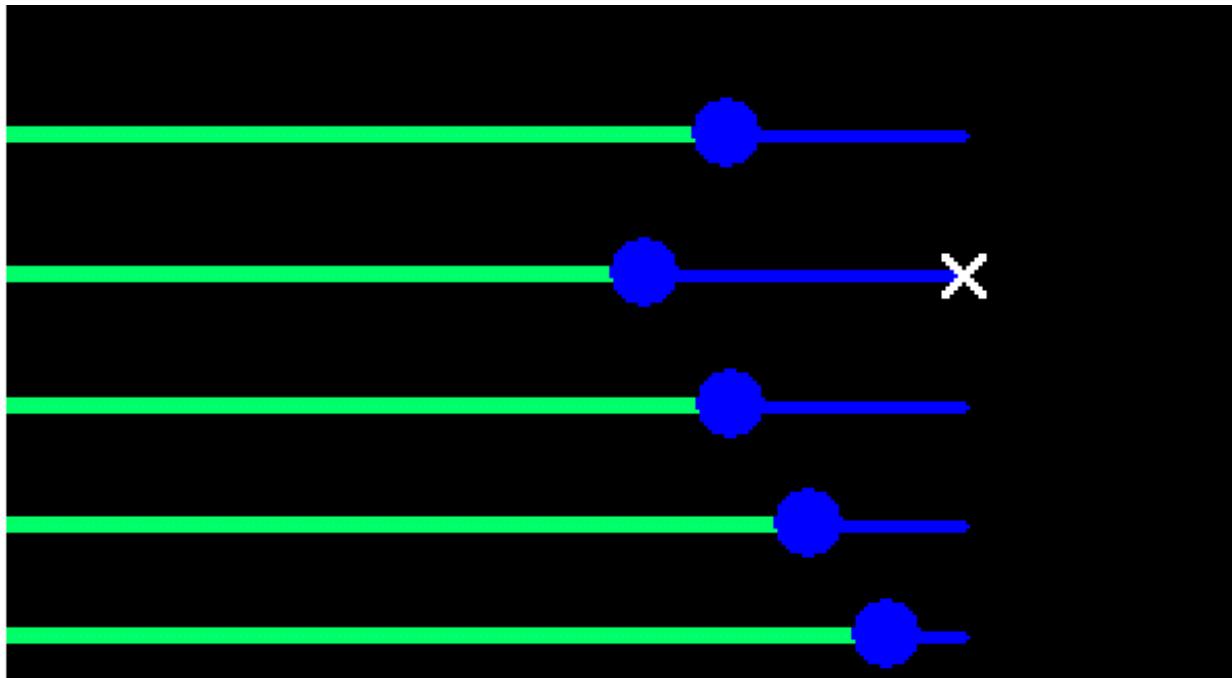


Taper patterns produce the same result as one of the diagonal patterns if the control-trace is at either end of the end.



If the vias are small, and/or the selected c-lines are already far enough apart, in group routing vias

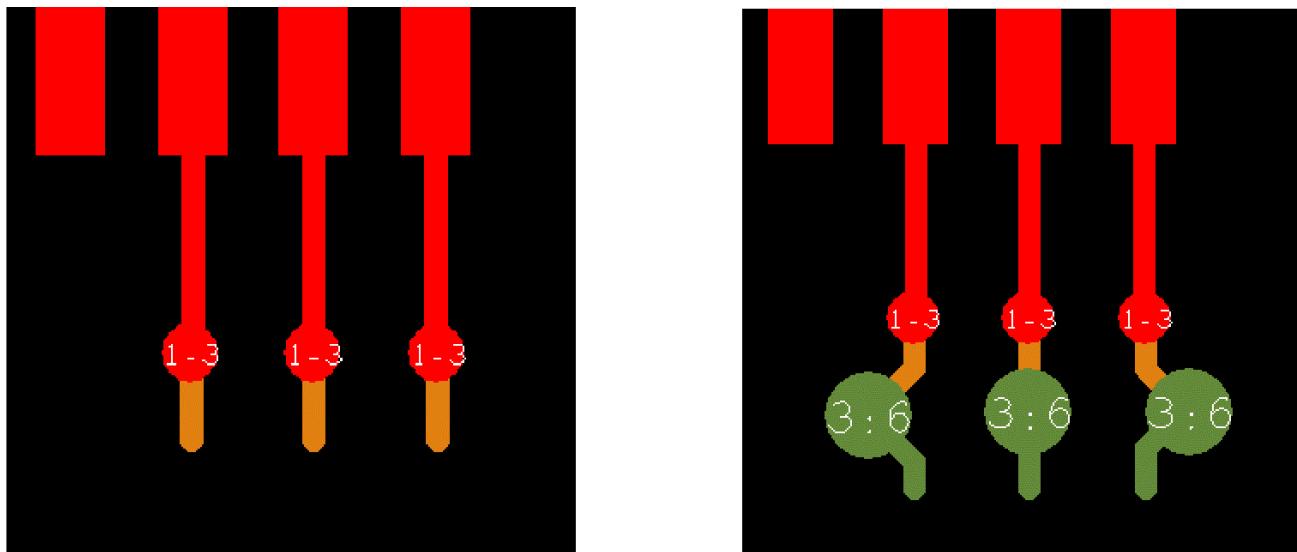
are added in-line, with no extra vertices.



Adding Stacked Blind/Buried Vias during Group Routing

For designs using stacked vias, you can select only those layers that can be reached with a single via-stack. The layers that can only be reached with staggered vias cannot be selected for adding vias in group routing.

The example in the following figure shows three via-stacks (labeled "1-3"). You can add stacked vias during group routing by invoking the command once.



If via-stacking is not allowed on layer three, then in order to add the vias from layers 3-to-6 you need to select add via second time, with layer six as the target layer. You can move vias labeled "3:6" vertically up or down until you click to drop them. To avoid any DRCs with the "1-3" via-stacks the "3:6" vias are placed in staggered form.

Related Topics

- [add connect](#)
- [Changing the Spacing Mode During Group Routing](#)

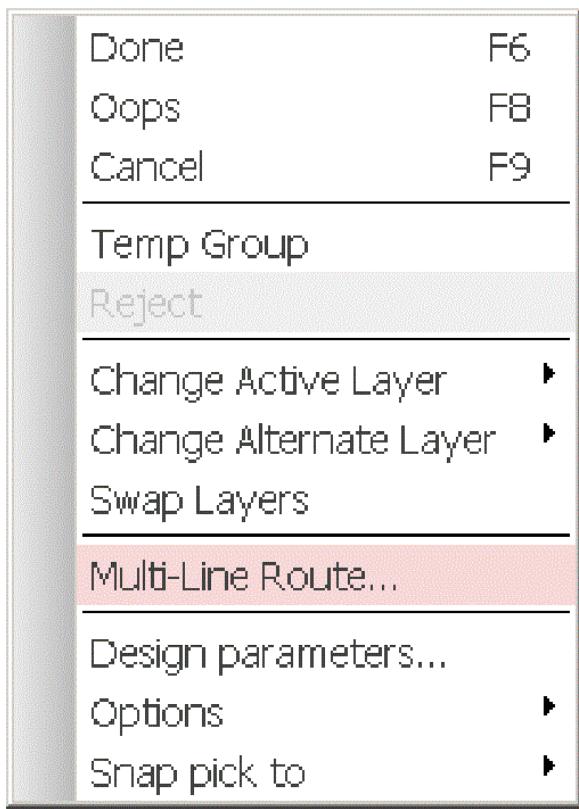
Interactive Freestyle Multi-Line Routing

When planning a bus route strategy, it is often convenient for floor-planning purposes to add the trunk of the bus into the design without actually connecting the ends. For example, you may know how to plan the bus route but are awaiting pin-out changes at the ASIC level.

By having the trunk of the bus already routed, you can also influence pin-out changes to accommodate the routing. Although interactive group routing also supports routing multiple connections simultaneously, it is designed to key off of existing design elements such as vias and cline segments to initiate the route. In other words, you cannot simply add multiple connect lines from point A to point B as with multi-line routing.

You initiate a freestyle multi-line route by choosing *Route – Connect*, right-clicking in the canvas and choosing *Multi-line Route* from the pop-up menu as shown in the following figure.

Multi-Line Route Option in the Add Connect Pop-Up Menu



For a step-by-step procedure, see [Performing a Freestyle Multi-line Route](#).

The following topics describe how freestyle multi-line routing works.

- [Setting Multi-Line Route Parameters](#)
- [Multi-Line Routing and Graphic Feedback](#)
- [Leveraging Design Intent to Route a Bus](#)
- [Using Contour to Route Rigid-Flex Designs](#)

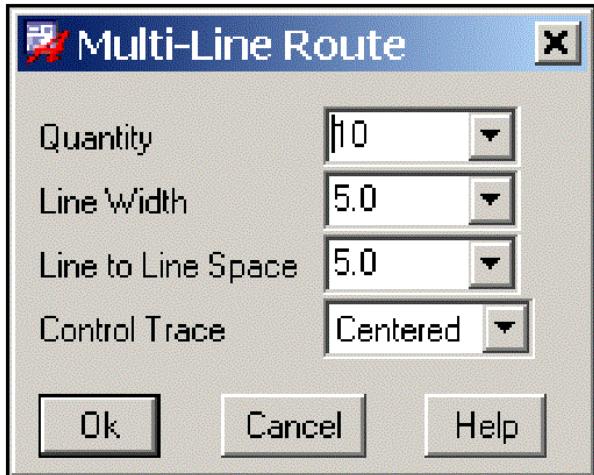
Setting Multi-Line Route Parameters

Before you begin a multi-line route, you need to set the physical route parameters for the bus. After choosing the Multi-Line Route option from the pop-up menu, the following prompt appears in the Console window.

Waiting for the Multi-Line Route origin pick.

Upon picking the origin point for the route, the Multi-Line Route dialog box appears as shown in the following figure. You enter the desired parameter values, then click *Ok* to begin the route.

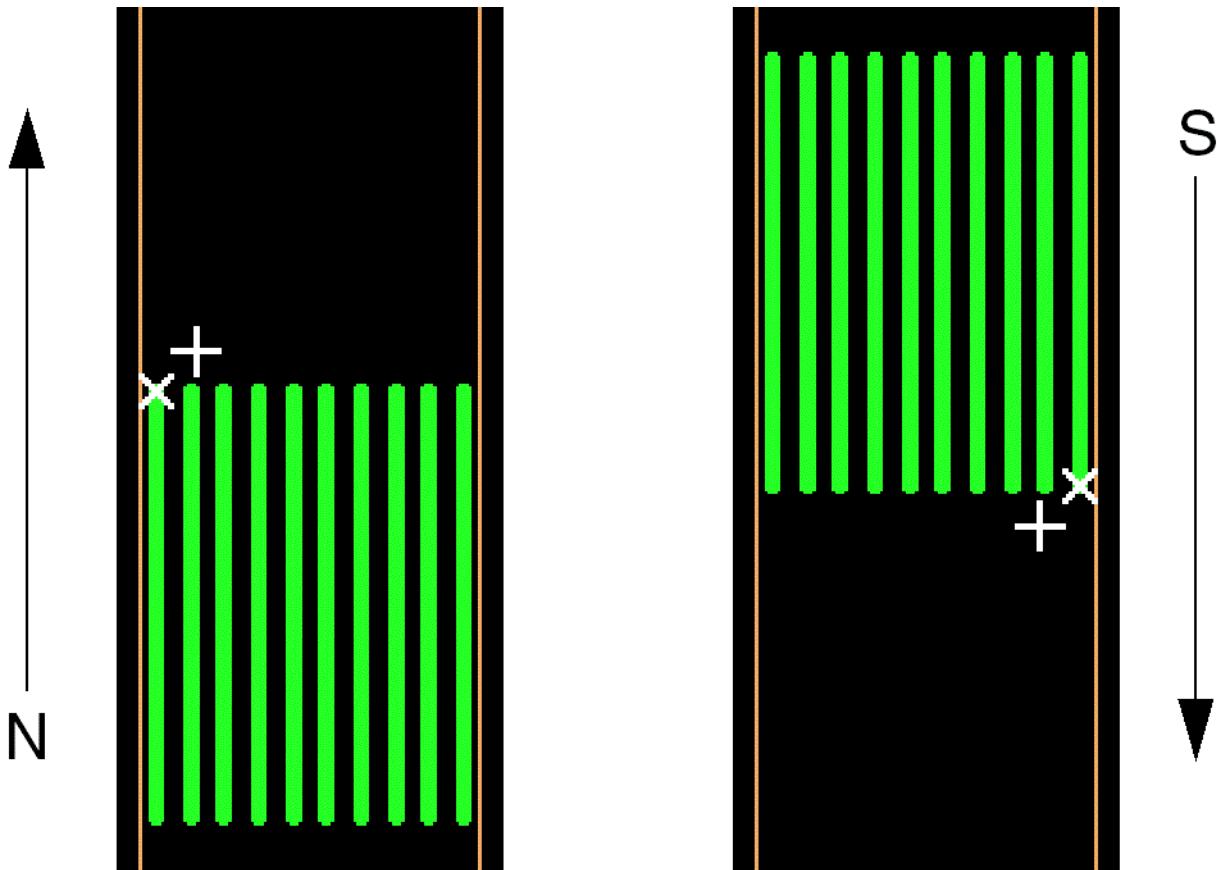
The Multi-Line Route Dialog Box



Control Trace

You can choose the left, right, or centered trace of the bus to closely track your mouse cursor when you route. Note that left and right are determined according to route direction. For example, if your route direction is north and you choose *on Left* for the control trace, the left-most trace of the bus is used. However, if the route direction is south, the right-most trace is used as follows.

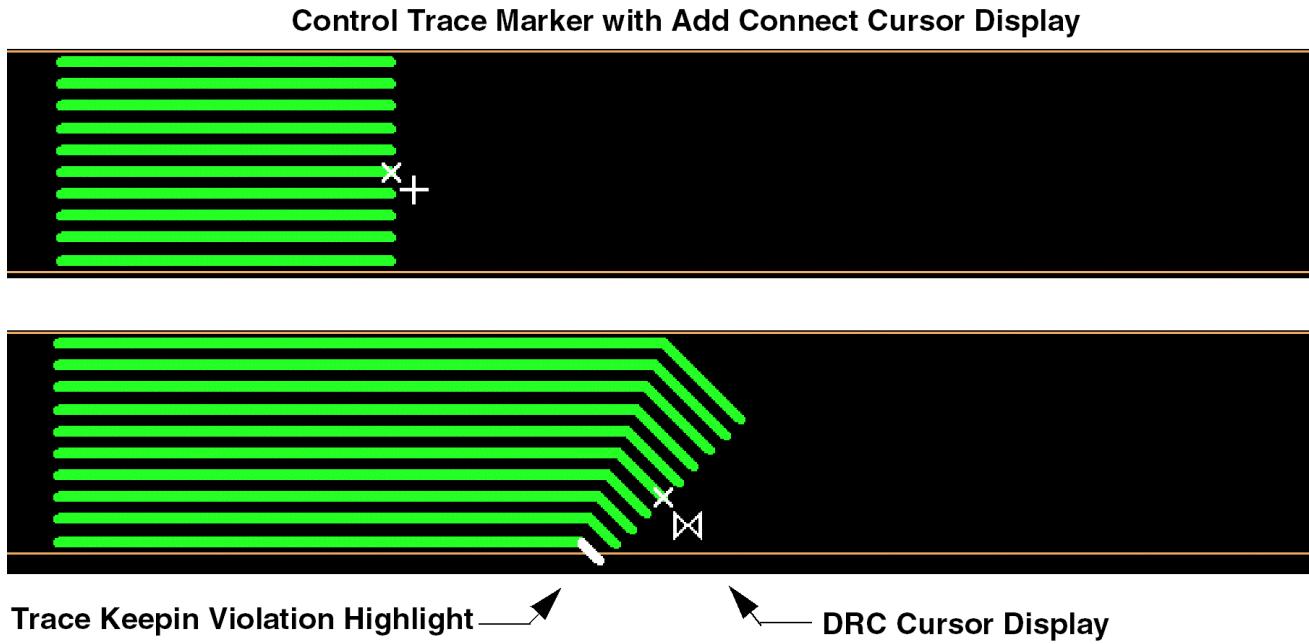
Control Trace on Left and Route Direction



Multi-Line Routing and Graphic Feedback

As you route, graphic feedback is displayed whenever your multi-line bus exceeds its boundary area (route keepin or board outline). The offending c-lines highlight where the violation occurs and the add connect cursor changes to a bow-tie (DRC) display as follows.

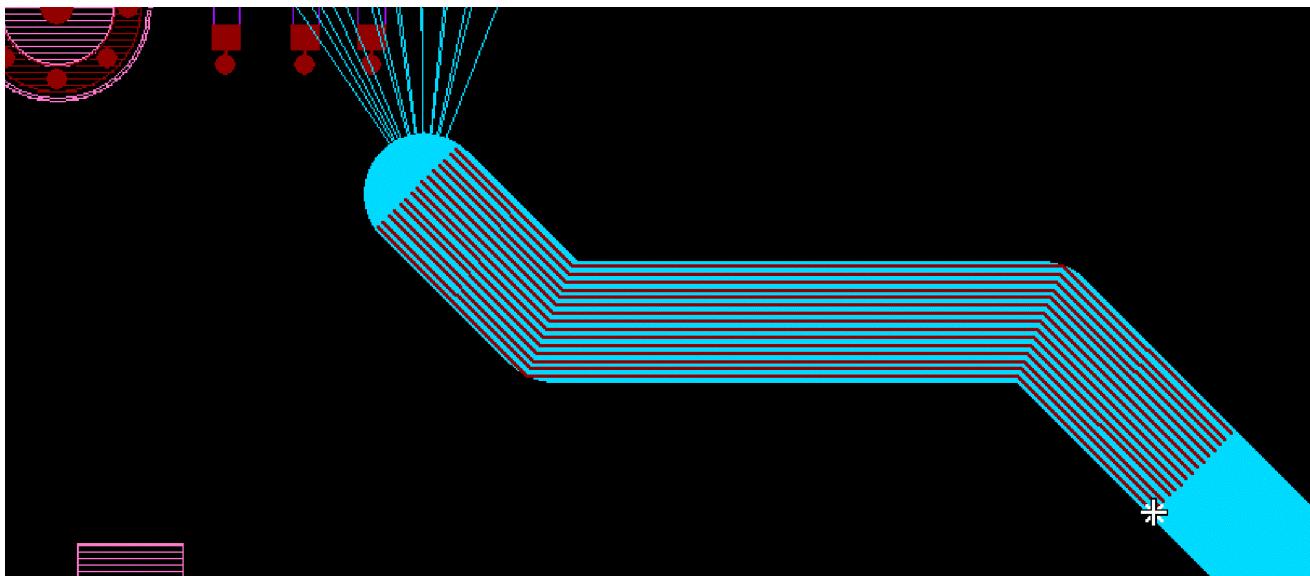
Routing a Bus Trunk Using a Route Keepin



Leveraging Design Intent to Route a Bus

Another benefit of freestyle multi-line routing is the ability to leverage existing design intent by having the bus follow the graphical flow of an IFP bundle within your design as shown in the figure. Note that bundle flow planning capability requires an Allegro advanced routing license. For further details, see the chapter [PCB Editor: Developing Interconnect Flows](#) in this guide.

Routing a Bus Using a Bundle Flow



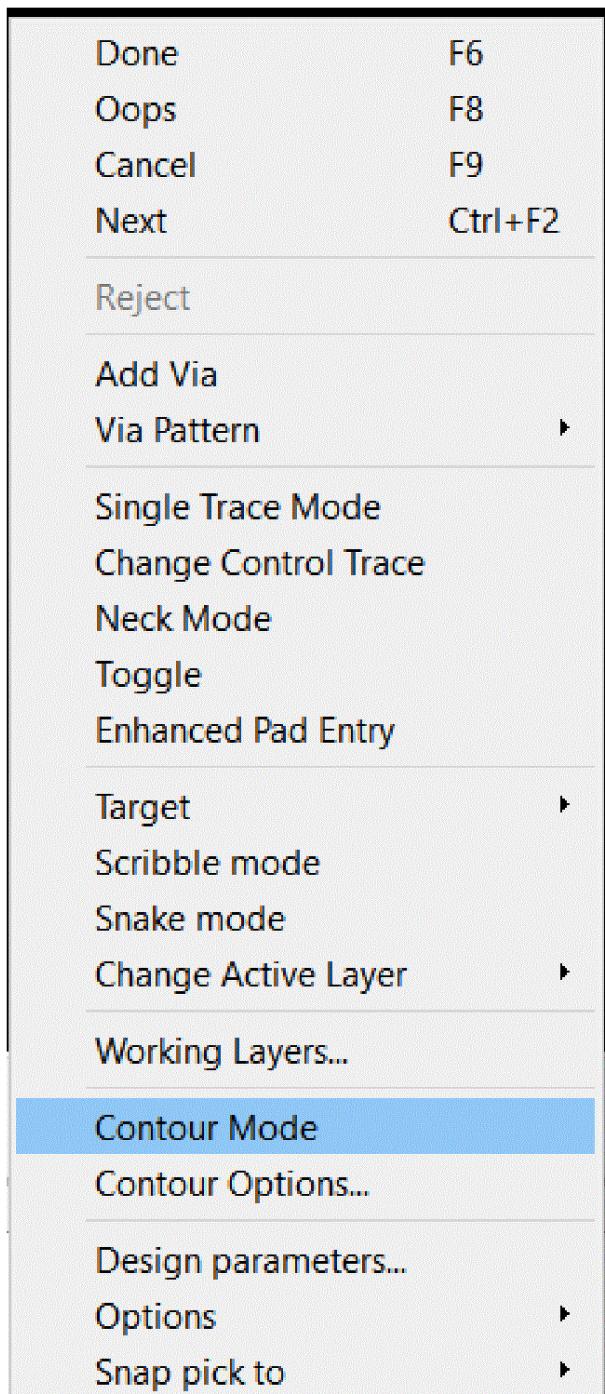
Using Contour to Route Rigid-Flex Designs

Rigid-flex designs require curved bus lines. A flex board outline can change during the design cycle - often after the original connect lines are committed. The ability to easily adjust the connect lines to the new form factor is critical. The *Contour Mode* option lets you quickly re-route the bus by hugging the contour of the route keepin as shown in the previous.

You initiate contour routing on-the-fly as you route by right-clicking and choosing *Contour Mode* from the pop-up menu as shown in the following figure.

- ⚠** When you right-click, if your cursor location is in appropriate position relative to the route boundary that you intend to hug, that location is automatically used as the start location for contour routing once contour routing parameters have been entered.

Contour Mode in the Add Connect Pop-Up Menu

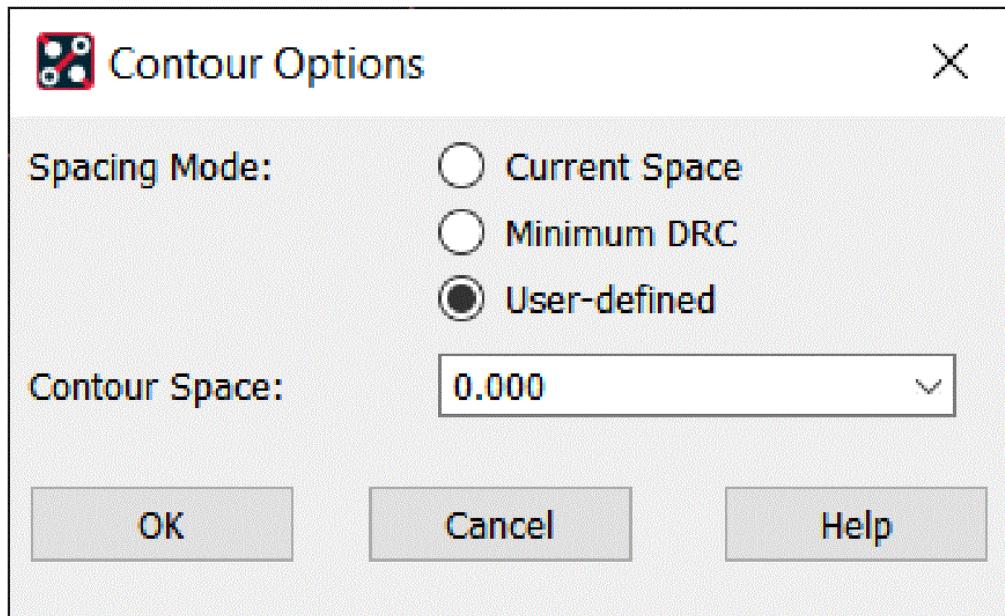


Setting Contour Route Parameters

Upon choosing the *Contour Options*, the Contour Options dialog box appears as shown in the following figure. Enter the desired parameters, click *OK* to dismiss the dialog box, then continue to slide your cursor along the curved boundary to begin contour routing.

-  If your cursor location was not in the appropriate position to the boundary when you right-clicked to choose the *Contour Mode* option, you need to click near the selected boundary to designate the start location for contour routing.

The Contour Options Dialog Box

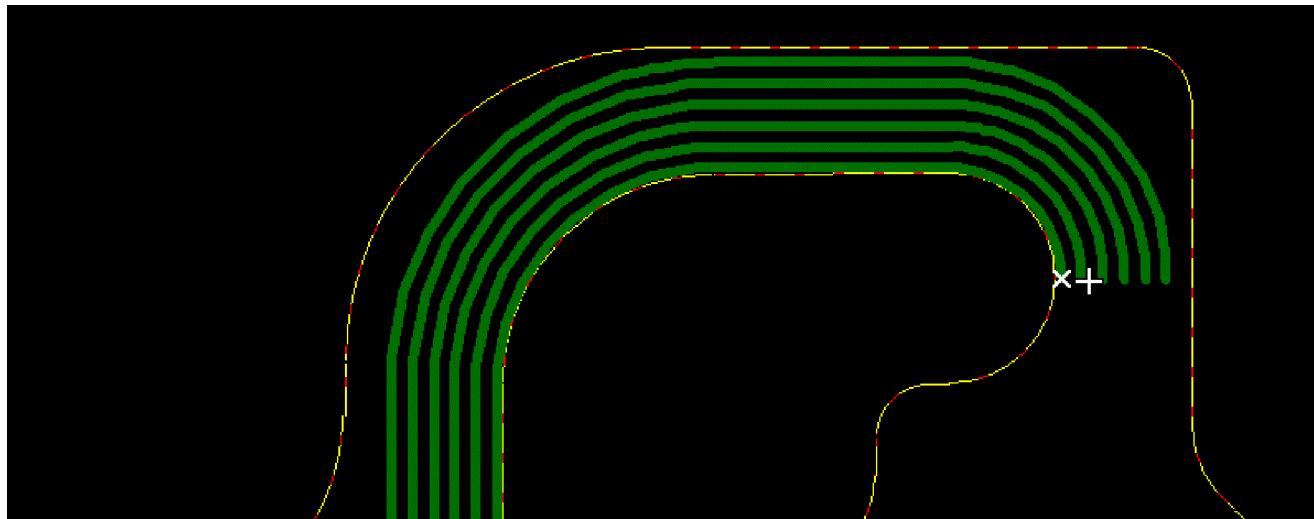


For a step-by-step procedure, see [Routing Connections Using the Contour Option](#).

Contouring Mode Examples

Contour mode can be used for fast and easy routing of buses and in rigid-flex designs. When using contour mode during group route, the contour spacing only applies between contour line and control trace. The following images show some examples of contour routing.

Curved Bus Routing Using Contour



Generating Reports on Interactive Routing

Setting Ratsnest Schedule for a Net

To set ratsnest schedule (*Ratsnet_Schedule*) to a net, choose *RATSNEST_SCHEDULE* in the Edit Property dialog box (*Edit - Properties*) and specify a value from the list.

Property	Value
Ratsnest_Schedule	
	MIN_TREE
	MIN_DAISY_CHAIN
	SOURCE_LOAD_DAISY_CHAIN
	STAR
	FAR-END_CLUSTER
	POWER_AND_GROUND

For example, to attach the *POWER_AND_GROUND* schedule to a ground (GND) net, do the following steps.

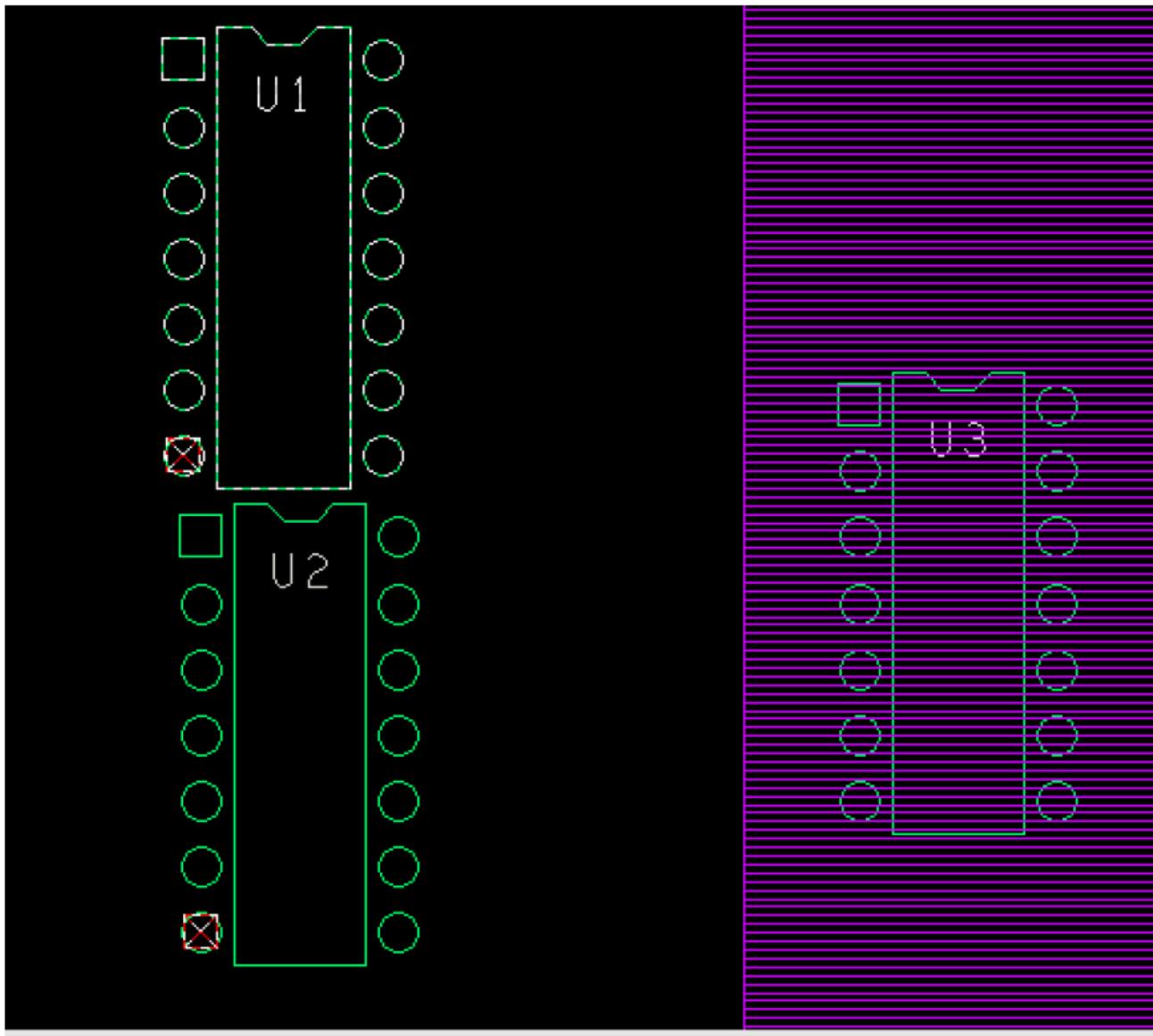
 You can also attach schedules to one or more selected nets in Constraint Manager by choosing *Tools - Setup Property Definitions*.

1. Choose *Edit – Properties (property edit* command).
2. Attach the *VOLTAGE* property to the net with a value.
3. Attach *RATSNEST_SCHEDULE* to the net and set the value to *POWER_AND_GROUND*.

Procedures and dialog boxes for these menu items and commands are described in the *Allegro PCB and Package Physical Layout Command Reference*.

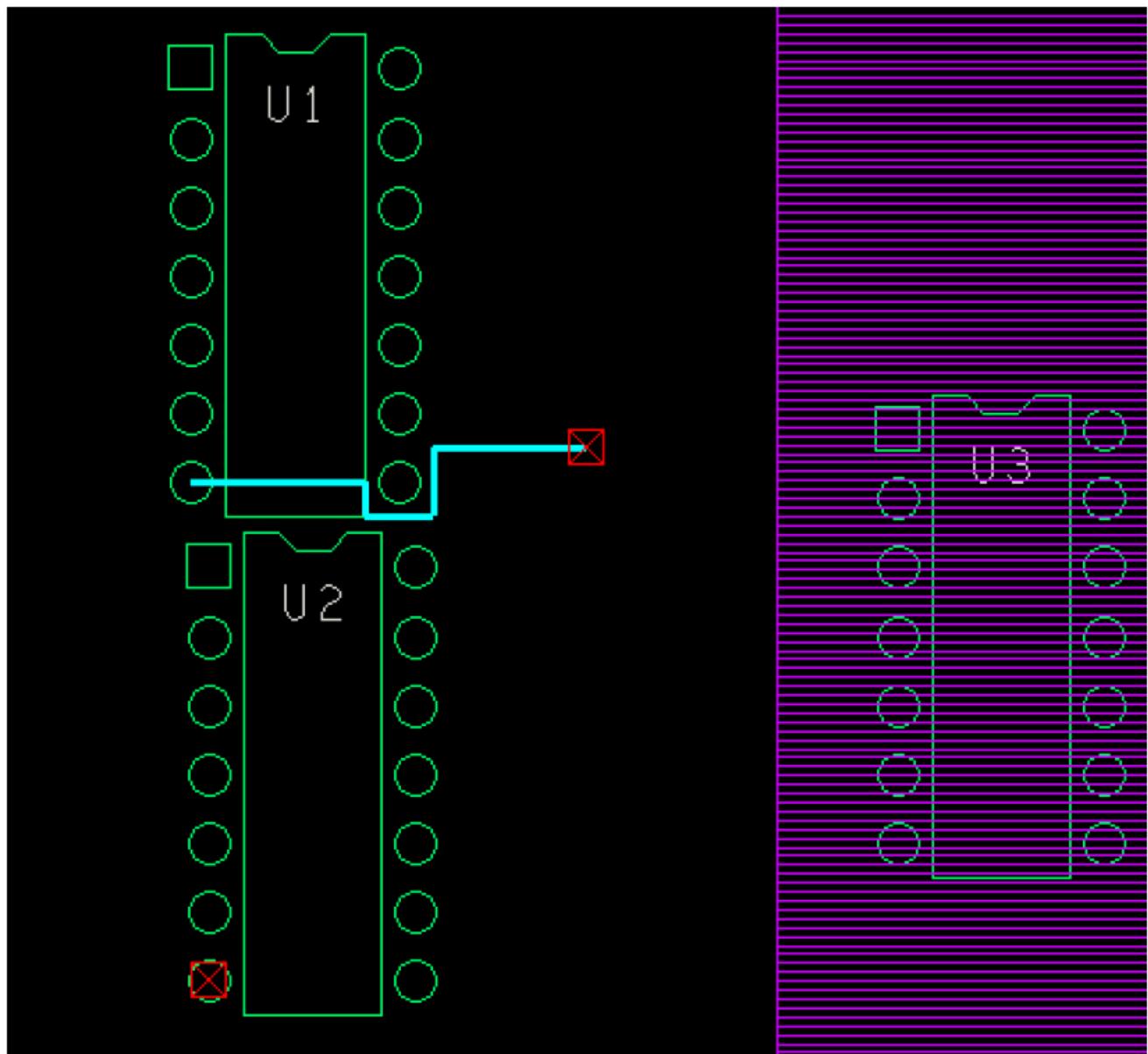
The net that you attach the schedule to displays the boxed-X figures on any unconnected pins, as shown in the figure below. In this example, U3 on active class ETCH, subclass GND is fully connected, therefore no pins display the boxed-X display.

Boxed X Warnings on Unconnected Pins



1. Choose *Setup – Design Parameters* (`prmed` command) to display the Design Parameter Editor.
2. In the Display tab, set *Ratsnest points* to *Closest endpoint*, and click *OK* to save the change.
3. Choose *Route – Connect* (`add connect` command) to add some etch from an unconnected power rat pin—in this example, the ground pin in U1.
The boxed-X display appears at the end of the dangling cline, as follows.

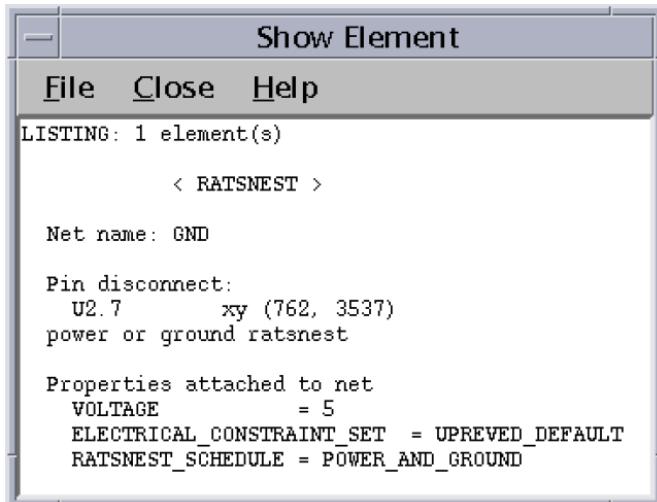
Warning Display on Dangling Cline



4. Choose *Display – Element* ([show element command](#)) and choose the figure at the end of the cline on the U2 ground net.

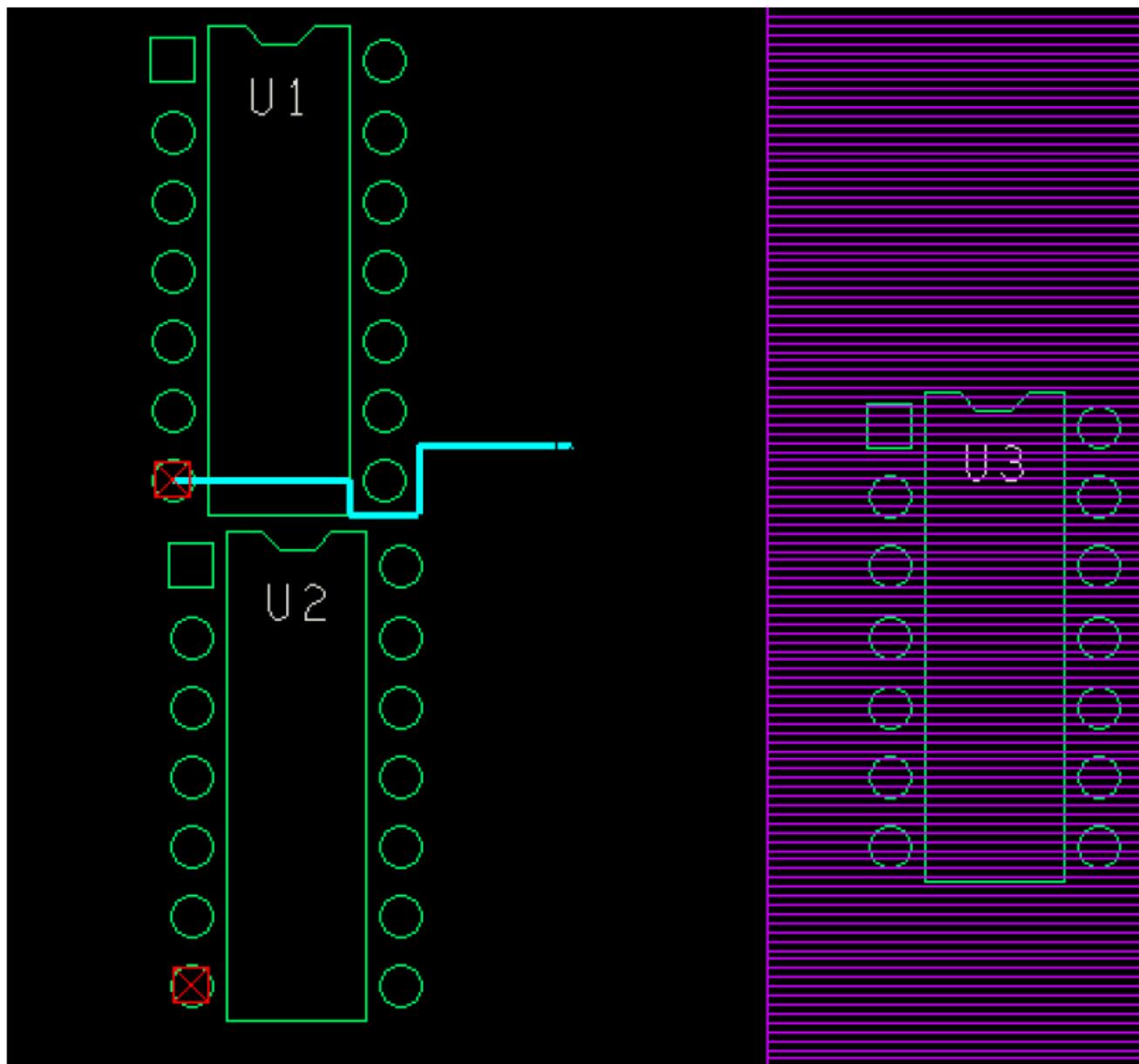
The Show Element dialog box in the following figure shows the lower left coordinates of the U2.7 display point (the other rat end being arbitrary). The net is labeled "power or ground ratsnest."

Show Element Data



5. Choose *Setup – Design Parameters* ([prmed](#) command) to display the Design Parameter Editor to reset the *Ratsnest points* option to *Pin to pin*.
As shown in following figure, the display reappears on the ground pin.

Boxed-X Display on Ground Pin



Prerequisites for Allegro PCB Router Automatic Routing

This section provides information on setting up prerequisites for the Allegro PCB Router used with the layout editor. Some of the prerequisites may apply to third-party routers. Where specifics differ, consult the user documentation for the specific router.

 Automatic routing is not available in all the layout editor versions.

General Routing Prerequisites

The following table shows general prerequisites, recommendations, and optional considerations for Allegro PCB Router as well as many third-party automatic routing tools. The table also provides pointers to the areas of the user guide that detail each process.

Prerequisites for Automatic Routing

Prerequisite	Automatic Routing	For details, see...
Define nets in netlist	Required	Writing a Netlist
Schedule nets in netlist	Recommended	Scheduling a Net Interactively
Load netlist	Required	Creating a Database
Place components	Required	Placing Elements Manually
Define etch subclasses	Required	Affect of ETCH or CONDUCTOR Shapes on Routing

Define etch width	Required	Defining Line Width
Define route grid (This can be done either in Allegro PCB Router or with layout tool's <code>auto_route</code> command)	Required	Specifying Grids
Verify layer information	Recommended	<u>Working with Cross-section Layers</u>
Define route keepin	Required	Keepin and Keepout Areas
Define blind and buried vias	As needed	Defining Blind and Buried Vias
Define route and via keepouts	As needed	Keepin and Keepout Areas
Define constraint sets for design rule checking	Required	Working with Constraints
Assign properties to nets and components	As needed	Working with Properties
Define via padstacks	As needed	Layout Padstacks, Vias, and Etch Shapes

Grids and Automatic Routing

The layout editor automatic routers typically place connections on grid locations. The only time etch is placed off grid is:

- When you run a pattern router
- When you connect to off-grid pins

The spacing of routing grids, together with etch width and spacing of etch and pads, determines how efficiently etch is placed on the design. Grids are one of the critical variables to check when results do not meet expectations.

In addition to routing grids, via grids define where vias can be located during executions of a via

router, DRC router, and delay router.

Defining Routing Grids

The layout editor lets you interactively define a variable routing grid and a different grid for each etch subclass by choosing the *Display* tab of the Design Parameter Editor, available by choosing *Setup – Design Parameters* (`prmed` command). If using Allegro PCB Router, you must set the grid within the router.

You can define a variable grid for etch subclasses by entering multiple spacing values. Variable grid points are spaced by the values in the sequence you provide. After the last value has been used, the series repeats from the first increment value, as shown in the following example:

Via Grids

You set via grids in Allegro PCB Router or in the *Via Grid* section of the Automatic Router dialog box when you choose *Route Automatic* (`auto_route` command).

Related Topics

- [Automatic Routing with Allegro PCB Router](#).
- [prmed](#)
- [auto_route](#)

Controlling How Vias Are Used during Routing

 For successful routing results, Cadence recommends that the only shapes on planes during routing be negative solid shapes. Routing does not put vias through crosshatched and positive shapes. Positive shapes or any kind of crosshatched shapes should not be added until you are ready to generate artwork.

Defining Vias for Use during Routing

You can assign the via types that are used during routing using the *Vias* column of the Physical Constraints worksheet within Constraint Manager. To access the worksheet, choose *Setup – Constraints – Physical* ([cmgr_phys](#) command). Click in the cells under the *Vias* column to access the Edit Via List dialog box, then select the via types you want to assign to the constraint set for each layer.

 Before you can establish a list of vias, you must define all necessary padstacks, both through-hole and blind or buried. Choose *Tools – Padstack – Modify Design Padstack* ([pateditdb](#) command) to complete this task.

Via Selection Priority

When multiple vias are available during routing, the router uses the via that spans the fewest layers. Therefore, blind and buried vias are selected before a through-hole via. If multiple blind and buried vias span the correct layers, the one that spans the fewest is chosen.

If both through-hole and blind/buried vias are legal for a net, the router prefers a sequence of blind/buried vias. If more than one sequence of blind/buried vias is possible, the layout editor selects the sequence with the fewest vias. The router selects a through-hole via only if no sequence of blind/buried vias is possible. If multiple through-hole vias are available, then the first one in the list is used.

Restricting Via Type by Area

You can use constraint regions to control the types of vias that are used within specific areas of your design. This is especially useful for Flex/Rigid Flex designs where HDI is prevalent. Once a constraint region is created, click in the cell under the *Vias* column to access the Edit Via List dialog box, then assign a via list to its constraint set as shown in the following figure.

Region Object with a Via List Assignment in Constraint Manager

For details on how to define constraint regions, see the [Objects - Create](#) - command in the *Constraint Manager Reference*.

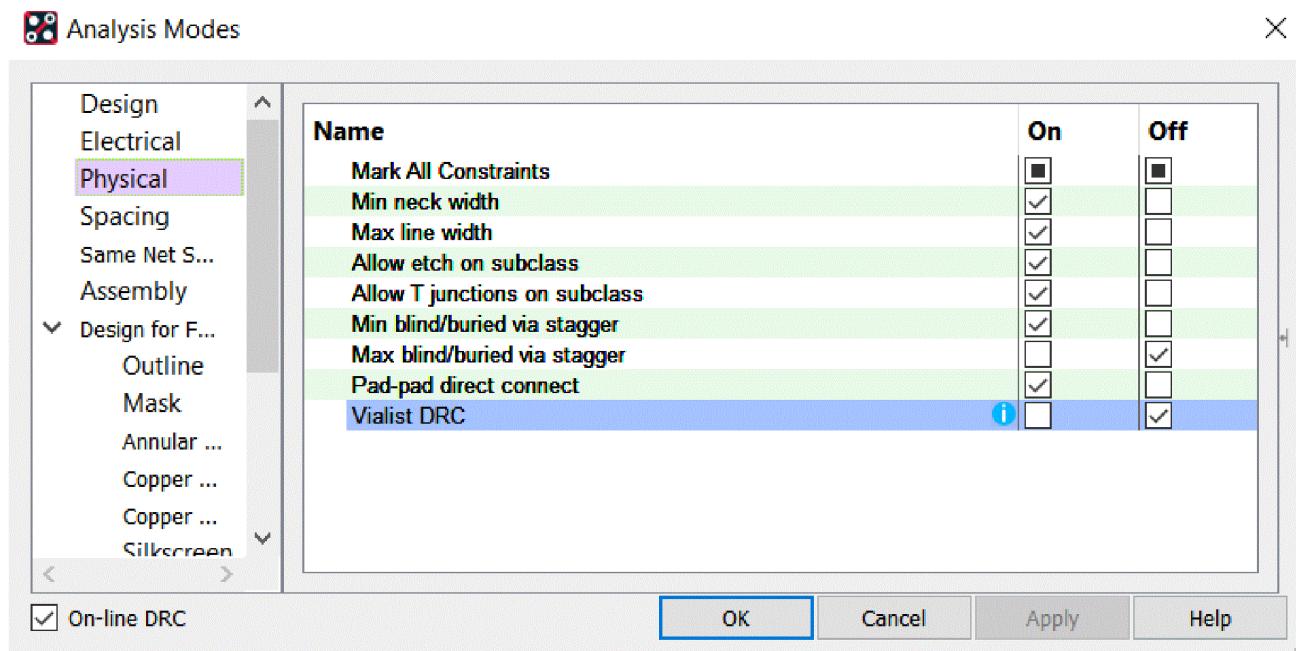
Via Type Audit

Type	Objects	Neck		Differential Pair Gap		Via	BB Via Stagger	
		Min Width	Max Length	Primary	Neck		Min	Max
		mil	mil	mil	mil		mil	mil
Dsn	module6_CM	0.00	0.00			V26C14P_BGA	5.00	0.00
Rgn	BGA_1MM					VIA019		

As your design is modified, the Via List DRC can be used to detect when illegal vias are used, or in the case of constraint regions, when illegal vias have been moved inside or outside of a constraint region boundary. You enable the via list DRC by choosing *Setup – Constraints – Modes* to access the Analysis Modes dialog box as shown in the following figure.

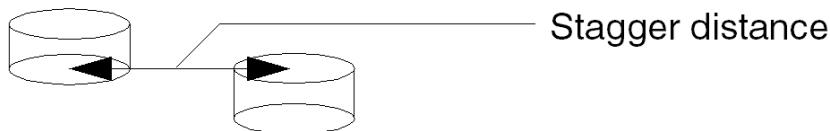
For further details on the Via List constraint, see Physical Constraint Data Sheets in the *Allegro Platform Constraints Reference*.

Via List DRC

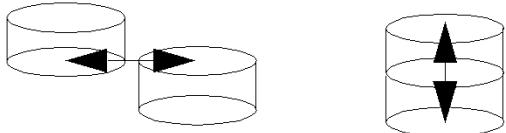


Controlling Via Staggering

Staggered vias are two pins or vias on the same net with a single connect line connecting them on a particular layer. The stagger distance is the distance from the connect point of one pin or via (x,y location) to the connect point of another.



A minimum stagger distance lets you control how close staggered vias can be to each other. Zero is a legal value for Min BBvia Stagger, which allows a pad to be placed on top of the via to which it is connected, as in the following example:



This establishes a direct connection, without a connect line. For details on direct pad-to-pad connections, see [Allowing Via Placement on Pads](#).

After you have minimum and maximum stagger distances, a DRC marker is generated if a stagger distance is less than the minimum stagger value or greater than the maximum stagger value. The DRC marker is placed on the common subclass.

Defining Minimum and Maximum Stagger Distances

There are two methods for defining stagger distances:

- Assign these distances to nets using a constraint set. Choose *Setup – Constraints – Physical* ([cmgr_phys](#) command) to enter values in the *Min* and *Max* cells under the *BB Via Stagger* column in the Physical worksheet of Constraint Manager. You can specify these fields by subclass.
- Add these distance constraints to individual nets. Choose *Edit – Properties* ([property edit](#) command) to assign these properties, described in the *Allegro Platform Properties Reference*:
 - MIN_BVIA_STAGGER
 - MAX_BVIA_STAGGER

These menu items and commands are described in the *Allegro PCB and Package Physical Layout*

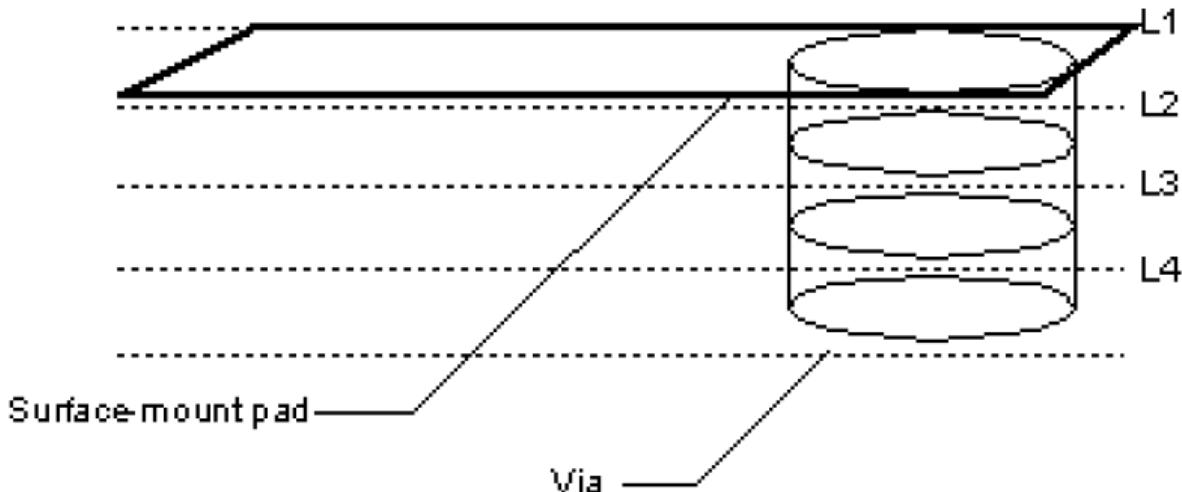
Command Reference.

Allowing Via Placement on Pads

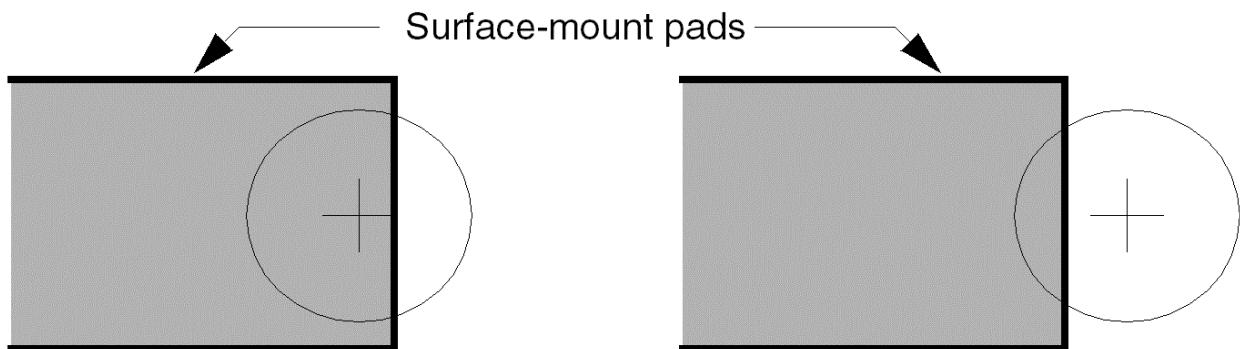
You can set the placement of vias on pads in the Physical worksheet of Constraint Manager. Choose *Setup – Constraints – Physical* (`cmgr_phys` command) and click in cells under the column *Allow - Pad-Pad Connect* to select a via placement option from the pull-down list for each layer.

To create a pad-to-pad direct connections, follow these rules:

- Both via and pad must be on the same layer (ETCH subclass).



- The origin of one pad must be within the edge of the other pad.



Connection:

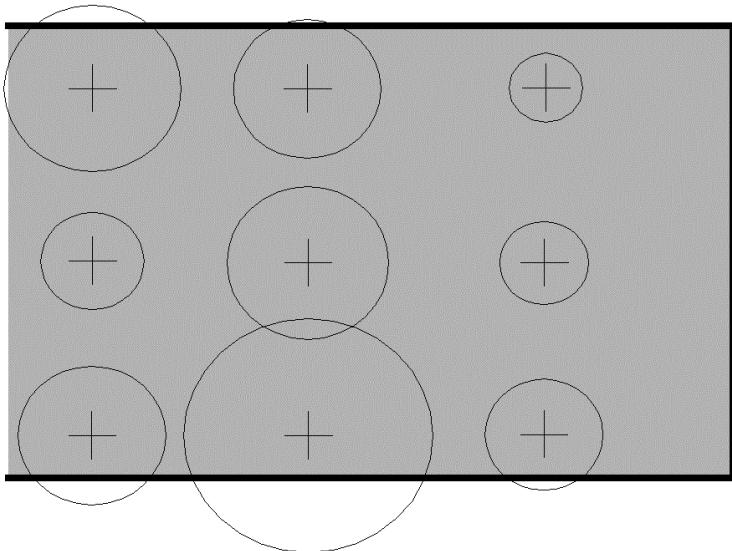
Origin of via inside the edge of the surface-mount pad

No Connection:

Via origin is not inside the edge of the surface-mount pad

- The two pads must be on the same net.

There is no limit to the number of vias that can connect to a single pad, as shown below:



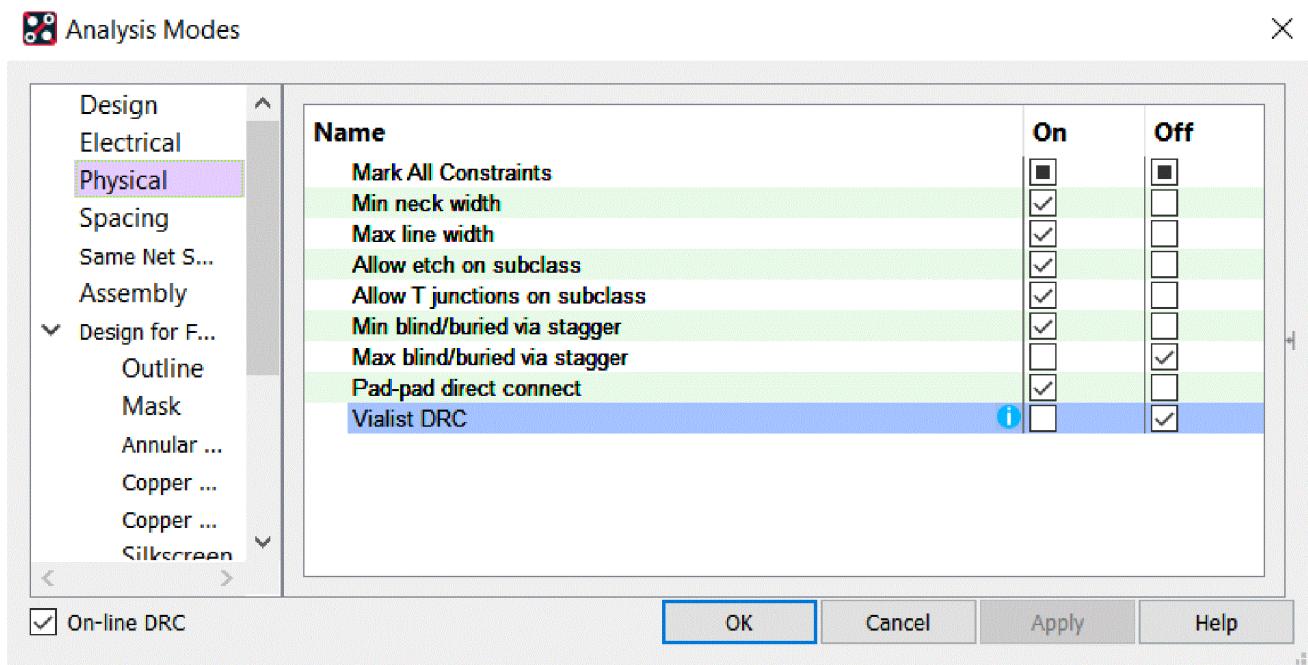
Controlling Via Stackups

You can control the types of vias that are allowed to stack on each other. For example, there are many types of HDI stackups, some that allow via stacking. The stacking of microvias is quite common whereas the stacking of microvias to core vias tends to be prohibitive due to cost.

Exclusive Microvia Stacking

You can restrict via stacking to microvia types. Choose *Setup – Constraints – Physical* ([cmgr_phys](#) command) to access the Physical worksheet in Constraint Manager. Click in the cells under the column *Allow - Pad-Pad Connect* to select a microvia stacking option from the pull-down list for each layer. This allows heterogeneous via stacking to be detected by way of the constraint set. Be sure the *Pad-pad direct connect* DRC mode is enabled as shown in the following figure.

Pad-Pad Direct Connect DRC



Controlling the Distance between Buried Vias

You can control the minimum distance allowed between the centers of two vias that do not share a common layer. The distance is measured from the connect point of one pin or via (x,y location) to the connect point of the other.

There are two methods for controlling the distance between buried vias:

- Assign the distance using a constraint set. Choose *Setup – Constraints – Spacing* ([cmgr_spac](#) command) to access the Spacing worksheet of Constraint Manager. In the tree view, expand the *Spacing Constraint Set* folder and then the *All Layers* icon. Select the *Same Net/BB Via Gap* icon, then enter a default value in the cell under the column *Min BB Via Gap*.
- Add this distance constraint to individual nets. Choose *Edit – Properties* ([property edit](#) command) to assign the *MIN_BVIA_GAP* property, described in the *Allegro Platform Properties Reference*.

These menu items and commands are described in the *Allegro PCB and Package Physical Layout Command Reference*.

- ⓘ If the gap is less than the specified constraint value, a DRC marker appears *except* if the vias in question share a common layer, in which case no DRC error is generated.

Controlling the Number of Vias on a Net

There are two methods for controlling the number of vias on a net:

- Assign this number using a constraint set. Choose *Setup – Constraints – Electrical* ([cmgr_elec](#) command) to access the Electrical worksheet of Constraint Manager. In the tree view, expand the *Net* folder and then the *Routing* icon. Select the *Wiring* icon, then enter values in the cells under the *Via Count - Max* column.
- Add this constraint to individual nets. Choose *Edit – Properties* ([property edit](#) command) to assign the **MAX_VIA_COUNT** property, described in the *Allegro Platform Properties Reference*.

These menu items and commands are described in the *Allegro PCB and Package Physical Layout Command Reference*.

Scheduling Nets Interactively

After placing design elements on a design, you might need to reschedule the order that the pins are routed on a particular net to satisfy electrical constraints and improve routability.

Choose *Logic – Net Schedule* (`net schedule` command), described in the *Allegro PCB and Package Physical Layout Command Reference*, to:

- Schedule entire nets and parts of nets.
- Schedule partially placed nets.
- Schedule Tpoints in nets.
- Unschedule nets.

Constraints That Affect Automatic Routing

This section explains key properties that affect automatic routing.

To apply properties, Choose *Edit – Properties* (`property edit` command), described in the Allegro PCB and Package Physical Layout Command Reference.

Net Properties

The following properties can be added to particular nets. The basics of each property and its values are described in the *Allegro Platform Properties Reference*. If there is additional information on routing for the property, it appears in this section.

- PROPAGATION_DELAY (For routing details, see below.)
- DIFFP_COUPLED_MINUS
- DIFFP_COUPLED_PLUS
- DIFFP_GATHER_CONTROL
- DIFFP_MIN_SPACE
- DIFFP_NECK_GAP
- DIFFP_PHASE_TOL
- DIFFP_PRIMARY_GAP
- DIFFP_UNCOUPLED_LENGTH
- ECL
- ELECTRICAL_CONSTRAINT_SET
- FIXED
- IMPEDANCE_RULE
- RELATIVE_PROPAGATION_DELAY
- LAYERSET_GROUP
- MAX_BVIA_STAGGER
- MAX_XTALK
- MAX_VIA_COUNT

- MIN_BVIA_GAP
- MIN_BVIA_STAGGER
- MIN_LINE_WIDTH
- NO_PIN_ESCAPE
- NO_RAT
- NO_RIPUP
- NO_ROUTE
- RATSNEST_SCHEDULE
- ROUTE_PRIORITY
- SAME_NET
- STUB_LENGTH
- TS_ALLOWED
- VIA_LIST

PROPAGATION_DELAY

Add this property to nets to control either the delay or the connection length between any two pins or for an entire net.

 For a complete explanation of how the Allegro PCB Router uses PROPAGATION_DELAY, see the *Allegro PCB Router* documentation.

The glossing applications do not recognize the PROPAGATION_DELAY property. Both the Line And Via Cleanup and Line Smoothing applications can make changes in the lengths of nets, so it is best to avoid running either of these two functions on any net with a PROPAGATION_DELAY. Add the NO_GLOSS property to delay nets. The other glossing functions make minor length changes and should not adversely affect nets with the PROPAGATION_DELAY attached.

Applying PROPAGATION_DELAY to a Net

You can add the PROPAGATION_DELAY property to a net using any of the following methods:

- Place it on the schematic with Design Entry HDL, System Connectivity Manager, or Design

Entry CIS (if available).

- Add it through a netlist.
- Choose *Edit – Properties* (`property edit` command) to assign it interactively.
- Choose *Setup – Constraints – Electrical* (`cmgr_elec` command) to specify it in the *Propagation Delay* column of the Electrical worksheet in Constraint Manager.

Because the syntax is complex, adding the property through Constraint Manager or a topology template is the recommended method.

Reports for Checking PROPAGATION_DELAY Results

Two reports let you verify etch results with the PROPAGATION_DELAY property:

ECL Actual/Schedule Report

This report is useful for verifying if the connection length restrictions for a board are reasonable.

 This report does not support non-ECL nets or extended nets.

To generate this report, type the following command:

```
report -v ean <design name> <output report name>
```

For every net marked with the ECL property, this report lists the manhattan distance for each pin-to-pin connection within the net. By comparing the required length with the manhattan distance for each connection, you can determine whether or not the lengths are acceptable.

If the requested maximum is shorter than the manhattan length or if the maximum is much longer than the manhattan distance, the router might not achieve the required results. If any of these conditions exist, fix them by modifying placement before running the router.

DRC Report

This report lists all violations that exist on the design. To generate this report, type the following command:

```
report -v drc <design name> <output report name>
```

Any connection that is longer than a given maximum length, or shorter than a specified minimum length, is listed in this report. This report also lists errors created when using interactive editing commands that do not adhere to the PROPAGATION_DELAY property.

IMPEDANCE_RULE

This property specifies an impedance restriction between any two pins on a net or between any pin and Tpoint connection on a net. When this property is assigned to a net its impedance value overrides any impedance value constraint defined in the net's Electrical Constraint Set.

IMPEDANCE_RULE is used to establish line widths used by Allegro PCB Router. The router passes a width rule that satisfies impedance unless the width is smaller than the minimum line width rule.

Applying IMPEDANCE_RULE to a Net

You can add the IMPEDANCE_RULE property to a net using any of the following methods:

- Add it through a netlist.
- Choose *Edit – Properties* (`property edit` command) to assign it interactively.

The netlist is the recommended method.

Component Properties

The following properties can be added to components:

- NO_PIN_ESCAPE
- PIN_ESCAPE

Optimizing Tpoint Location

After fanout is complete and before any critical routing is performed, you can use the `optimize_ts` command to help optimize the location of Tpoints in designs containing complex topologies such as dense H-tree and Tpointed differential pair.

This command uses the first of two separate modules that Allegro PCB Router employs to optimize Tpoints for routing. The first module is a coarse optimizer that works during the opening of a `.dsn` or `.do` file. The second module runs during the first two route passes and optimizes locations according to routing when the dynamic virtual pin setting is on (default).

Graphic feedback from this command enables you to visually inspect the optimized Tpoint locations to ensure that they are reasonable. If so, you can begin routing. However, if one or more Tpoint locations are unsatisfactory, you can make further adjustments interactively by using the following procedure.

To interactively adjust and refine the location of Tpoints

1. Move one or more Tpoints to more satisfactory locations in the design taking routing rules into consideration.

 Be sure to check (select) Rat Ts in the Find Filter.

2. Apply a FIXED_T_TOLERANCE property to each Tpoint that you move to preserve it's new location.

 The value of this property is a radius (0 (fixed) to 500 (default) mils) from the supplied location that defines how much the router is allowed to move the Tpoint once routing commences.

For details on how to apply a property to a design element, see the procedures for the `property_edit` command.

3. Check your design carefully to ensure that vias at these new Tpoint locations do not create conflicts.
4. Using the Route Editor, type the following commands in the Command line.

```
protect all wires route 2 delete all wires quit (and save)
```

 This is a more involved process of Tpoint location refinement and takes longer to optimize.

This result is the Allegro PCB Router's best effort to optimize Tpoint locations.

5. Repeat steps 1 through 3.
6. Repeat step 4 using a five pass routing strategy (`route 5`).
7. Make any final adjustments by repeating steps 1 through 3 again.

Routing the Design

Prerequisites for Allegro PCB Router Automatic Routing--Optimizing Tpoint Location

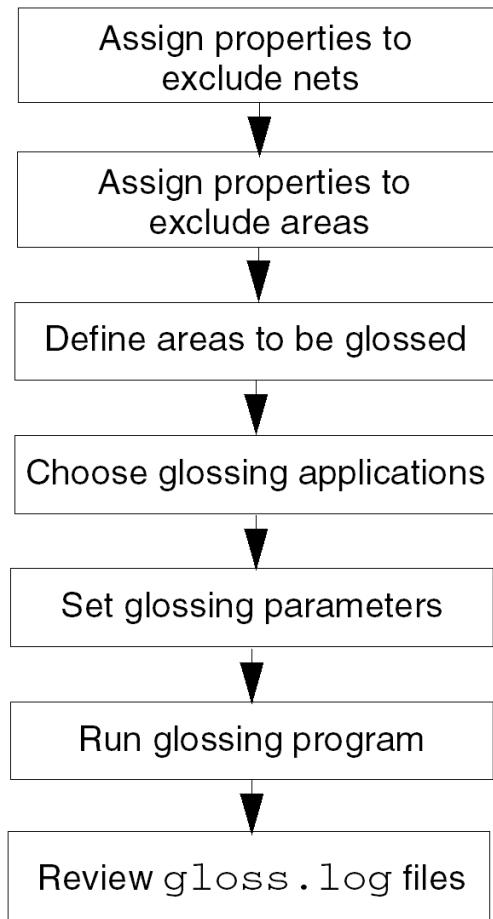
Post-Route Clean Up

The automatic glossing process eliminates vias and straightens routed traces on the board to improve the appearance and manufacturability of the design. A suite of nine glossing functions are accessible from the *Glossing Controller* form, which lets you edit their parameters, listed below, and control their execution.

Glossing Application	Availability
Line and Via Cleanup	All except Allegro PCB Design L
Via Eliminate	All except Allegro PCB Design L
Line Smoothing	All except Allegro PCB Design L
Center Lines Between Pads	All except Allegro PCB Design L
Improving Line Entry Into Pads	All except Allegro PCB Design L
Line Fattening	All except Allegro PCB Design L
Converting Corner to Arc	All except Allegro PCB Design L
Pad and T Connection Fillet	All except Allegro PCB Design L and <i>Dynamic Fillets</i> option is unavailable in Orcad PCB L and PCB L Performance
Dielectric Generation	All except Allegro PCB Design L and Allegro PCB Performance option L

The gloss routines can be run on an entire design, individual areas of the design, or individual nets on a design. The following figure shows the glossing process.

Glossing Process



Before Glossing

Before glossing, do the following:

- Determine if you want to gloss the entire design, individual areas, or individual nets.
- Exclude nets from glossing by assigning the NO_GLOSS or FIXED properties.
- Exclude areas from glossing by assigning the NO_GLOSS_TOP, BOTTOM, INTERNAL properties.
- Choose glossing applications.
- Set glossing parameters.

Excluding Nets from Glossing

When you route an entire design or areas of a design, there are often critical nets such as clock nets or analog nets that you do not want the gloss routines to modify.

You prevent a net from being changed by the glossing routines by attaching the NO_GLOSS property to a net. You can also define areas of the board, such as analog areas, that should not be modified with the glossing routines, by adding an area to the board on the MANUFACTURING class. To designate nets that require special treatment, assign the following properties by choosing *Edit – Properties* (`property edit` command):

- NO_GLOSS: Prevents a net from being changed by the automatic glossing applications.
- FIXED: Prevents a net or group from being changed by any automatic routine.

Excluding Areas

By enclosing an area of the design with a no-gloss polygon, you can exclude that area from being changed by glossing. A no-gloss polygon is a shape on class MANUFACTURING. It can be placed in any of the following subclasses:

- NO_GLOSS_TOP
- NO_GLOSS_BOTTOM
- NO_GLOSS_ALL
- NO_GLOSS_INTERNAL

To add a no-gloss rectangular shape, choose *Setup – Areas – Gloss Keepout* (`keepout`

`gloss` command) with the *Active Class and Subclass* in the *Options* tab set to *Manufacturing* and *No_gloss_all*.

- ⚠** To add a polygon or circular shape, choose *Shape – Polygon* (`shape add` command) or *Shape – Circular* (`shape add circle` command) with the *Active Class and Subclass* in the *Options* tab set to *Manufacturing* and *No_gloss_all*.

Defining Areas to be Glossed

You can specify any of the following areas for glossing with the appropriate menu path or command.

Area	Menu Path/Command	Notes
Design	<i>Route – Gloss – Design</i> (<code>gloss area design</code>)	The layout editor automatically selects the area defined by the route keepin for glossing.
Individual nets or components	<i>Route – Gloss – Highlight</i> (<code>gloss area highlight</code>)	You select the nets or components for glossing before you select this option.
Room(s)	<i>Route – Gloss – Room</i> (<code>gloss area room</code>)	You specify one or more rooms as glossing areas.
Window	<i>Route – Gloss – Window</i> (<code>gloss area window</code>)	You specify a rectangular window area for glossing.

To view the current, active area of the design for glossing, choose *Route – Gloss – List* (`gloss area list` command).

Related Topics

- [property edit](#)
- [keepout gloss](#)
- [shape add](#)
- [shape add circle](#)
- [gloss area design](#)
- [gloss area highlight](#)
- [gloss area room](#)
- [gloss area window](#)
- [gloss area list](#)

Glossing Applications

Access the glossing routines briefly described below and set glossing parameters from the Glossing Controller, available by choosing *Route – Gloss – Parameters* (`gloss param` command).

Line and Via Cleanup

The Line and Via Cleanup option processes one net at a time, ripping up every connect line and via and rerouting it using a high via cost. If the rerouted path is an improvement, the new path replaces the existing one.

Via Eliminate

The Via Eliminate glossing routine tries to remove unnecessary vias in the design without rerouting each of the nets. You can have the Via Eliminate glossing routine remove pin escape vias, standalone vias, and standard through vias. A standalone via is a via that is not connected to a net. A standalone via could have happened from interactively editing traces. Vias that were added as built-in pin escapes to a package symbol will also be eliminated in the glossing routines.

Line Smoothing

The Line Smoothing glossing routine removes jogs in lines that could be created interactively or by the automatic router. You could use the Line Smoothing glossing routine to change 90-degree bends to a set of two 45-degree bends. The Line Smoothing glossing routine generally reduces the total etch length of the design and also frees up routing channels. Line smoothing removes extra jogs and line segments in the design. Line smoothing is a good tool to use to help open channels during routing.

Center Lines between Pads

The Centering Lines glossing routine centers lines between component pins that are lined up in the horizontal or vertical direction. The design rules and grids used in most of today's designs allow multiple routing channels to fit between adjacent component pins. If only one of these channels is used, the route will not be centered between the pads. The manufacturing yield of the printed circuit board can be increased by centering these lines between the component pins. It is best to run this glossing routine only after the board has been completely routed, because many of the connections will be moved off grid when they are centered between pads.

Improving Line Entry into Pads

After routing the board, pin connections often exist that create acute angles from the pad to the connection leaving the pad. This can happen on circular pads when the connection exits the pad from a point that is not at the center of the pad. These acute angles lower the manufacturing yields of your printed circuit boards. The acid from the etching process can continue to eat away in that acute angle area. This will weaken and could cause a break in the signal. You can use the Improve Line Entry glossing routine to force connections to leave pads at specific angles.

Line Fattening

The Line Fattening glossing routine widens lines on your printed circuit board wherever it can without causing DRC violations to improve reliability when the design is manufactured. The Line Fattening form lets you create a set of new widths for every existing line width. The set of widths for each line can be unique for each etch subclass.

When you run *Line Fattening*, the editor examines all connect line segments within the defined area and compiles a list of candidates for widening. These are the criteria that eliminate a connect line from this list:

- NO_GLOSS or FIXED property assignment
- Location in a NO_GLOSS area

- Fattening a line generates a DRC marker
- Fattening a line makes the line wider than the pad it connects to

When a list of candidates is created, the editor makes one pass through the entire design and increments each line segment to the first width listed on its step. After one pass, the editor returns to the beginning and completes another pass to the next defined increment.

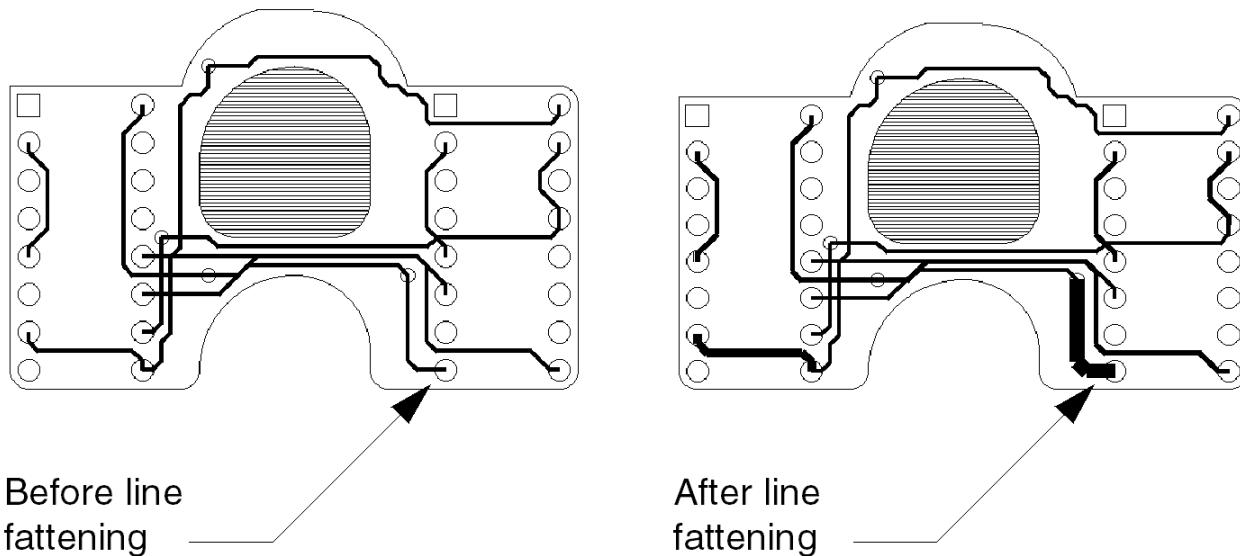
If widening a connect line causes a DRC error, the segment is returned to its previous width, and marked for exclusion on the next pass.

⚠ However, if you set the `gloss_fatten_single_seg` environment variable (`enved` command), the tool fattens all segments as wide as they can go without creating DRCs, and does not abort the fattening process when the first segment to hit its limitation occurs.

Additionally, all other line segments on the same connect line are reduced back to the current segment's previous width and are also marked for exclusion. The application orders each step to progress in consistently larger increments, regardless of how you typed them into the step.

Because the program examines individual connect lines and not entire nets, the following figure shows how a typical substrate might look following an execution of Line Fattening.

Line Fattening



Converting Corner to Arc

The Converting Corners glossing routine converts 45- and 90-degree bends to arcs. This routine is often used on analog designs and very-high-speed designs. It might also be used for flex circuits. The resulting arcs can be difficult to hand edit.

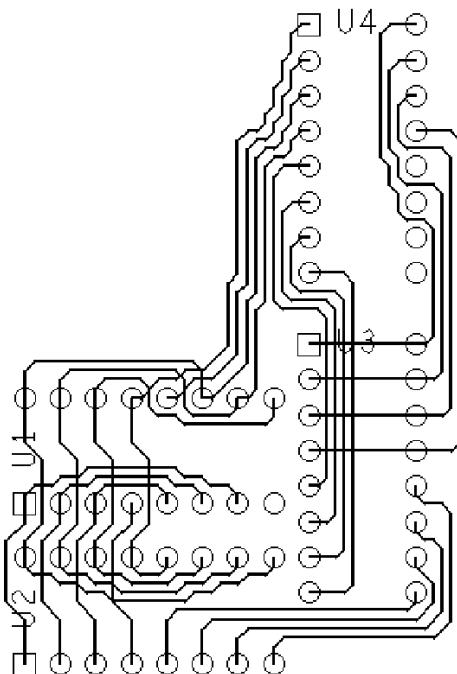
This glossing application examines each etch/conductor subclass for connect lines with 45- and 90-degree corners. When a corner is identified, the process attempts to create an arc at the maximum radius specified.

If a DRC is created, it decreases the radius and tries again. This process is repeated until it either reaches the minimum radius value or completes with no DRC. The default number of executions is one.

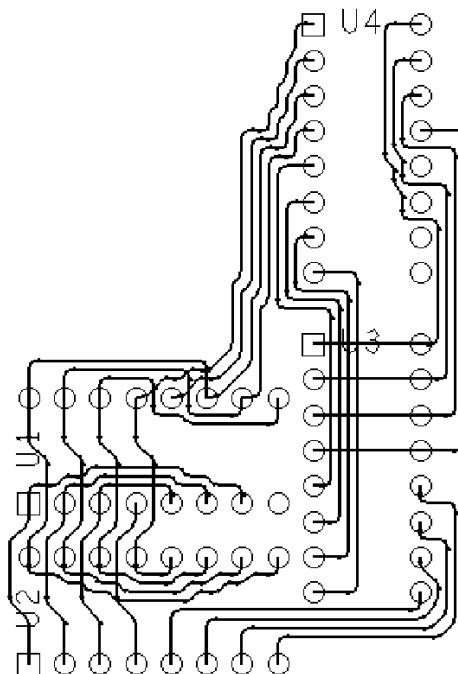
If corners were unchanged, you can change the parameters and rerun the program. Note that nested connections typically require multiple executions. When you run a second or third iteration of the program, corners already converted are not changed.

The following figure shows a layout after the program has been run, using the default parameters.

Converting Corners to Arcs



Before converting corners

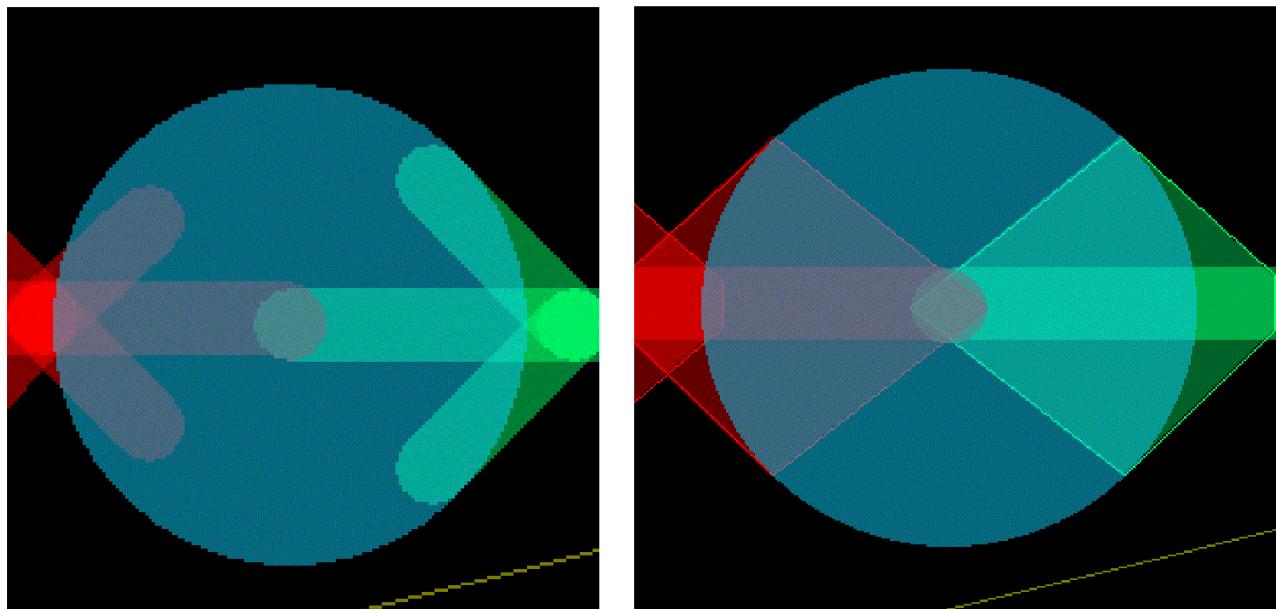


After converting corners

Pad and T Connection Fillet

The Pad and T Connection fillet glossing create shape-based fillets, rather than line-based fillets, also known as teardrops. Fillets are extra etch added to pads or etch Ts at the point where a cline enters the pad, to improve manufacturability. Shape-based fillets offer flexibility for style requirements, such as rounded edges, required for flex designs, which rely on shape-based fillets to transition the stress on a trace as it enters the pad, allowing a wider entry point.

Line-Based Fillet (l) and Shape-Based Fillet (r)



Older versions of layout editors use line-based filleting. Fillets extend a pin pad at the trace-to-pad intersection to ensure that the hole will not create a possible open at this point. Consequently, the application assumes pin-pad spacing DRC rules than shape rules.

⚠ To restore the old behavior of using line-based fillet DRCs, enable line-based spacing check in the *Design* tab of the Analysis Modes.

Analysis Modes

Design	Name	Value	On
Electrical	Mark All Constraints		<input type="checkbox"/>
Physical	► General		<input type="checkbox"/>
Spacing	► Soldermask		<input checked="" type="checkbox"/>
Same Net Spacing	► Acute Angle Detection		<input checked="" type="checkbox"/>
Assembly	► Package		<input type="checkbox"/>
▼ Design for Fabric...	► SMD Pin		<input checked="" type="checkbox"/>
Outline			
Mask			
Annular Ring	▼ Spacing Options		
Copper Featu...	Check holes within pads	<input type="checkbox"/>	
Copper Spaci...	Backdrill Min Space	<not set>	
Silkscreen	Supress DRC on exploded pins	<input type="checkbox"/>	
▼ Design for Assem...	Enable Line based check for fillets	<input checked="" type="checkbox"/>	
	► Mechanical Spacing		<input type="checkbox"/>

For pre-16.2 boards, cline fillets coexist with shape-based fillets; however, any new fillets are shape-based. Existing cline-based fillets are not changed during `uprev`. Running `downrev` on boards with shape-based fillets removes them while retaining cline-based fillets. Shape-based fillets must then be recreated after `downrev`. Voiding for shape-based fillets uses pad spacing for pads or cline spacing for Ts rather than shape spacing rules.

Invoking Fillet Modes

Three fillet modes are available:

- Static: Disable the *Dynamic Fillets* option on the *Pad and T Connection Fillet* dialog box. Choose *Route – Gloss – Parameters*, and on the *Glossing Controller*, choose *StaticRun* next to the *Pad and T Connection Fillet* dialog box. Click the *Gloss* button to add shape-based fillets in a static, or batch, update. Whenever you modify a pin, via, or cline, the tool deletes the fillets and does not regenerate them.
- Dynamic: Enable the *Dynamic Fillets* option on the *Pad and T Connection Fillet* dialog box. When you initially enable this option, the entire board updates with shape-based fillets. During subsequent interactive route editing, fillets are deleted and then regenerated based on the specified parameters (unless an element has the `NO_FILLET` property assigned).
- Interactive: For instance editing of fillets, choose *Route – Gloss – Add Fillet* (`add fillet` command) or *Route – Gloss – Delete Fillet* (`delete fillet` command). You cannot run this command if the *Dynamic Fillets* option is enabled.

If you are filleting manually rather than dynamically, Cadence recommends that you run any other glossing options first, including the *Improve Line Entry into pads* glossing application, available by

choosing *Route – Gloss – Parameters*. To further improve manufacturability, use *Route – Custom Smooth* (`custom smooth` command) interactively to optimize the angles of a single net or a group of nets.

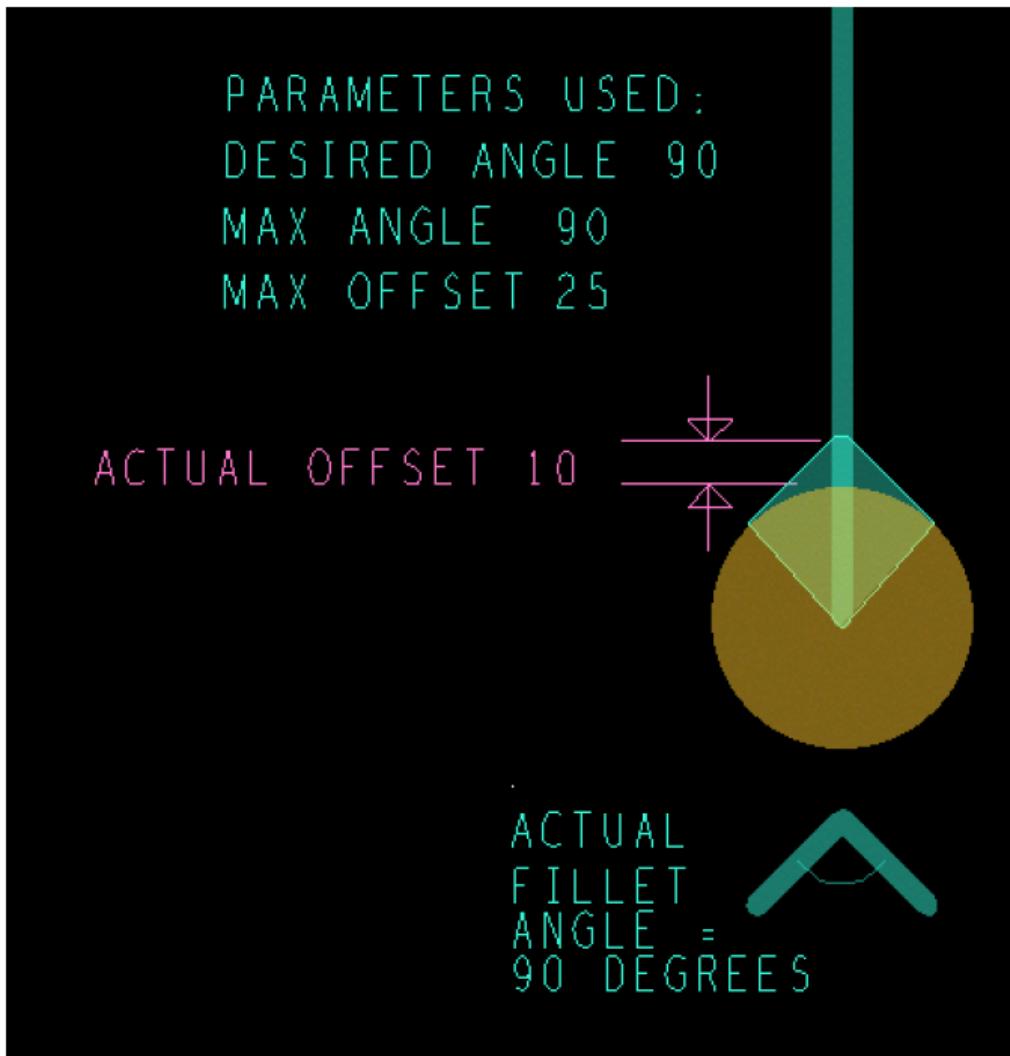
Setting the Desired Angle, Max Angle, and Max Offset Parameters

The algorithm first tries to create the fillet at the *Desired Angle*, tangent to the pad. If the fillet cannot be created, the angle is incremented up to the *Max Angle*. If the fillet length from pad tip to the vertex of fillet is greater than the *Max Offset*, the vertex is adjusted by an amount to satisfy the *Max Offset* requirement, and the end points of the fillet are adjusted by the same amount to maintain the angle. A larger *Desired Angle* and a smaller *Max Offset* gives a short teardrop and a smaller *Desired Angle* and larger *Max Offset* gives a long teardrop.

The following examples highlight how these parameters interact with one another to produce various filleting effects.

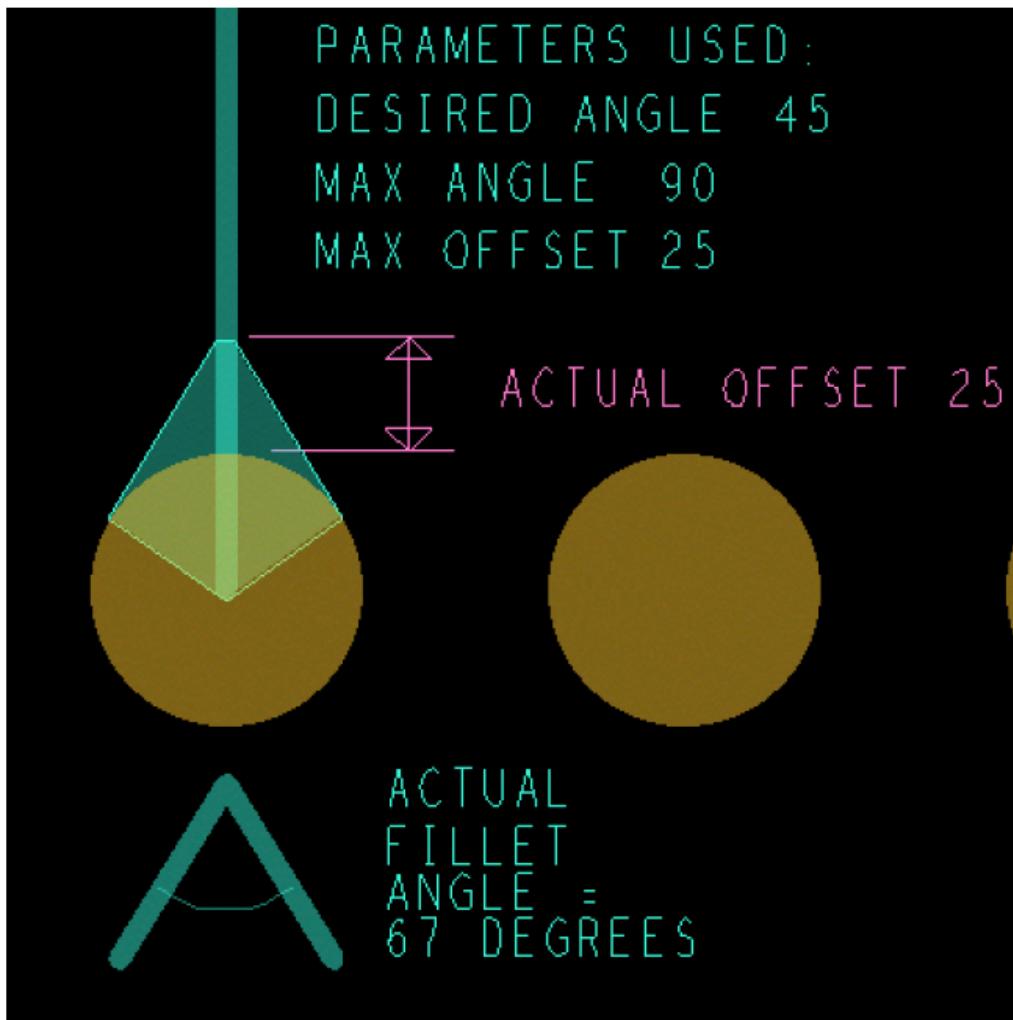
The following figure adds fillet at the *Desired Angle* within the Max offset.

Fillet Example 1



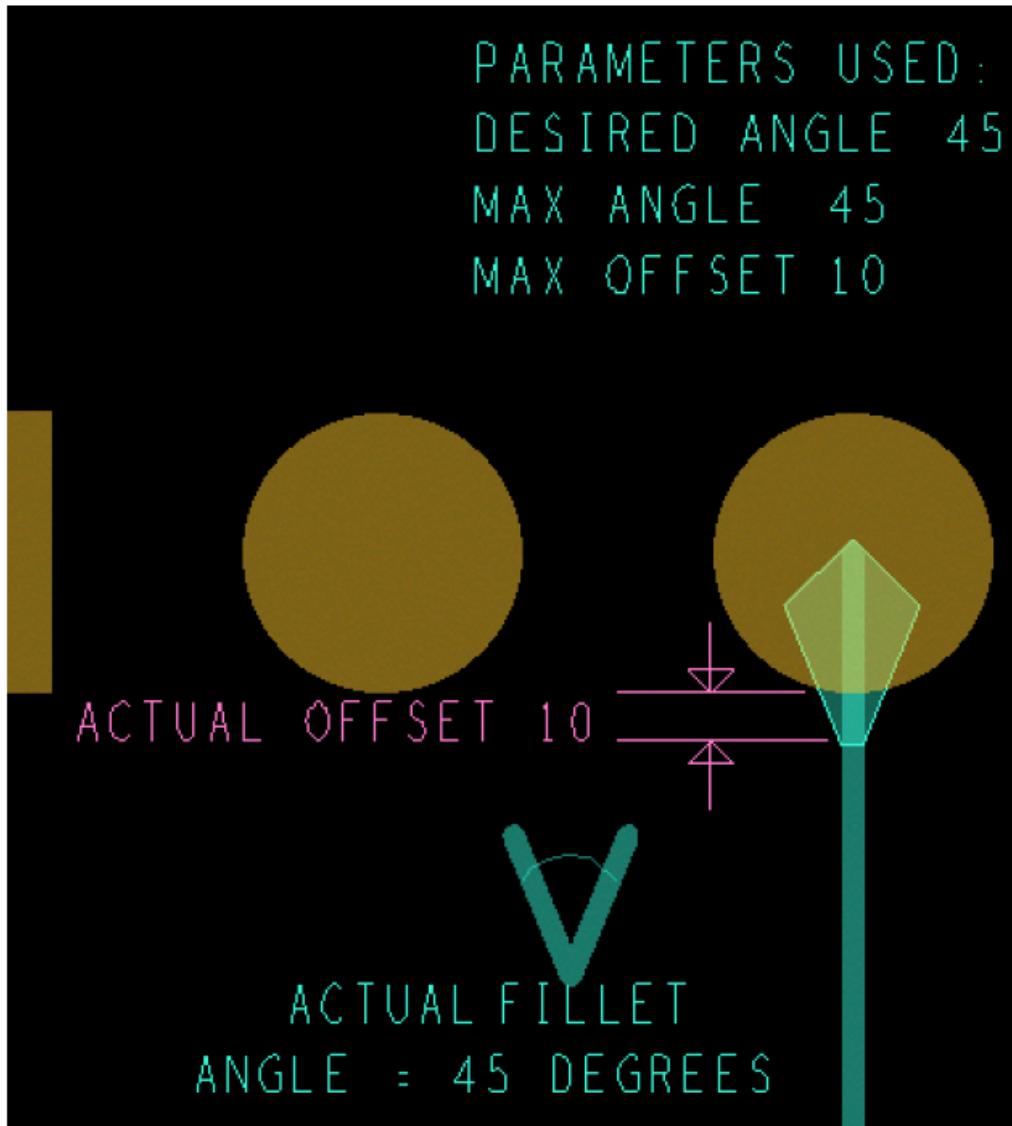
In the following figure, a fillet with a 45 degree angle cannot fit within the offset of 25. The fillet generates at the *Max Offset*, and the needed angle is internally calculated.

Fillet Example 2



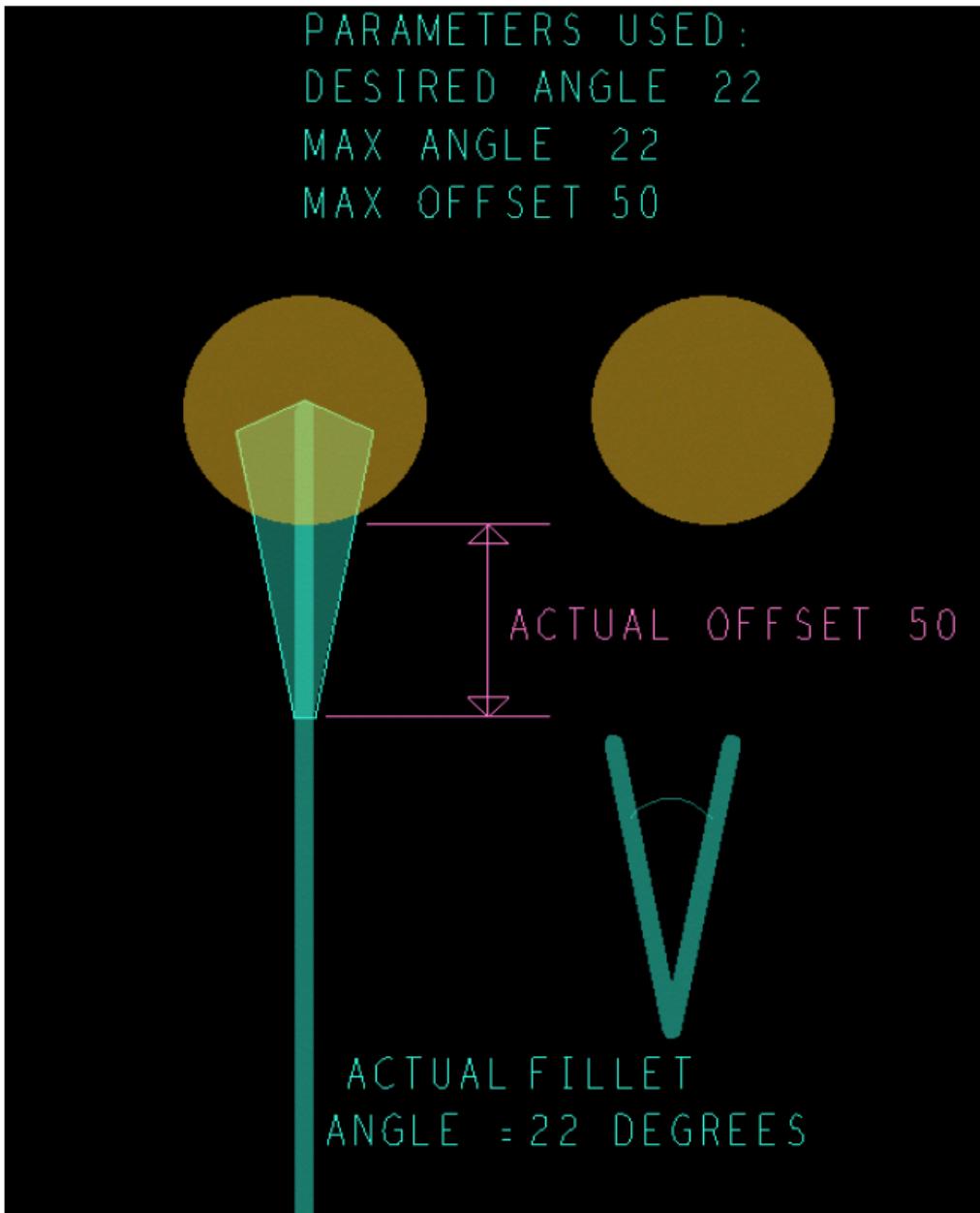
The following figure shows a fillet forced to draw with one specific angle. Both the *Desired* and the *Max Angle* are equal.

Fillet Example 3



The following figure illustrates how the offset can be used to manipulate a specific angle fillet. Both *Desired* and *Max Angle* are equal. Tangency with the pad is lost due to the adjustment required to meet the *Max Offset* and maintain the calculated angle, resulting in an elongated fillet.

Fillet Example 4



Executing the Fillet Operation

When Pad and T Connection filleting executes in either static or dynamic mode, the application compares each pin, via, and T with the parameter definitions to determine whether it is a candidate for glossing.

Editing a Design Filleted by Static Mode

If minor edits are required after filleting in static mode, use *Route – Gloss – Add Fillet* to interactively create fillets.

To remove fillets from larger areas, you can window select around a group of fillets and delete them using *Route – Gloss – Delete Fillet* (`delete fillet` command). Replace them by re-running the *Pads and T connection fillet* glossing application on that area in static mode (with the *Dynamic Fillets* option disabled).

Dielectric Generation

Use the Dielectric Generation to define the size of one or two dielectric areas, depending on the number needed in the design. When the Dielectric Generation glossing application is executed, dielectric patches are placed between intersecting connections.

Before completing this dialog box make sure you define the etch/conductor subclass for the following layers:

- Crossover Layer
- First Dielectric Layer

The etch subclass is optional for the second dielectric layer, but it is necessary if the design manufacture requires a two-screen dielectric process.

Blind and buried vias must also be defined.

When you run Dielectric Generation, consider the following:

- If you use the automatic router to place connections, all DRC errors are displayed by the editor and listed on the Design Rule Check Error Report. If you place the connections manually, the editor still sees the intersections and places the dielectric patches when this program is run but does not display DRC markers. You must track DRCs manually.
- Cadence recommends running the program at the end of an editing session. If you move connections that are over or under a dielectric patch, the patch remains in its original location and might cause a short circuit in the design.
- If you need vias to make a connection in the case of a large dielectric patch, you must place them yourself. The automatic void program does not see these even though they are listed as shapes.
- The application is most efficient when operating on designs without arcs.

There are two modes of running Dielectric Generation:

- Active: You place the crossover connections interactively on a separate layer.
- Passive: the editor finds instances of crossover DRC violations and determines which one belongs on the separate layer.

In both cases, the editor reconciles the possibility of a short circuit by placing a rectangle of dielectric material between the two crossover connections. The editor determines whether to place one or two layers of dielectric material, and the dimensions of those rectangles from your parameter definitions.

When you interactively place the etch in a crossover situation, you decide which piece of etch to place on the crossover subclass. When the editor corrects a DRC, it considers both pieces of etch to be candidates for the crossover layer. To determine which connect line is placed on the crossover subclass, the editor creates a connect line set for each connect line involved. This set consists of the connect line and all other connect lines that T into it. The set that contains the greatest number of line/line DRCs that are crossovers is placed on the crossover layer.

Related Topics

- [gloss param](#)
- [enved](#)

Glossing Error Reporting

When glossing completes, check the results:

- Click *File – Viewlog* to check the `gloss.log` file (generated in Static mode).
- Click *File – Viewlog* to check the Missing Fillets Report (generated in Dynamic mode), which lists the parameters used to generate the fillets in that design as well as information on missing and partial fillets, including net, item, location, and subclass. This report is also available by choosing *Tools – Reports* (`report` command).

The following sample `gloss.log` file shows the results of the Pad and T Fillet glossing application.

Sample `gloss.log` File

```
GLOSSING on drawing C:/boards/sh_merge.brd Glossing started on Tue May 20 10:50:15 2008
**WARNING: Disabling dynamic shapes
```

Pad and T Fillet glossing in progress.

Started on Tue May 20 10:50:15 2008

Round pads: YES max size: 100.0

Square pads: YES max size: 100.0

Rectangular pads: YES max size: 100.0

Oblong pads: YES max size: 100.0

Octagon pads: NO max size: 100.0

Pads as shapes: NO

Pins: YES

Vias: YES

Bond Fingers: NO

Ts: YES

Unused nets: YES

Pads Without Drills: YES

Desired Angle: 90

Maximum Angle: 90

Maximum offset: 25.0

Minimum line width: 3.0

Allow drc: NO

Dynamic: NO

Curved lines: NO

*WARNING: Can't fillet T at pad (2800.0 3800.0)

Unable to find segment with one end inside pad and the other end outside pad.

Total number of fillets added : 66

Number of fillets rejected due to DRC errors : 0

Elapsed time: 0 hr 0 min 0 sec (0.00 hr)

Actual time of completion: Tue May 20 10:50:15 2008

There were 2 errors reported in logfile

Warning Messages for Pad and T Connection Filleting

The following warning messages pertaining to Pad and T Connection Filleting appear in the `gloss.log` log file when a fillet pattern creates a DRC error, and the fillet is not added.

If part of a fillet cannot be created due to a DRC error, the problem is reported in the `gloss.log` file with the following message:

```
WARNING: Fillet segment from (3450,6168) to (3431,5952) on subclass BOTTOM removed  
due to DRC error.
```

In addition, the number of fillets that only partially fail due to a DRC error are tallied and reported at the end of the log file with the following message:

```
Number of fillets rejected due to DRC errors : 1
```

This message indicates that one partial fillet was created. If an entire fillet cannot be added due to DRC errors or for any other reason, then the following message appears in the log file:

```
*WARNING: Can't fillet T at pad (2800.0 3800.0)
```

```
        Unable to find segment with one end inside pad and the other end outside pad.
```

If the fillet fails completely due to DRC errors (no partial fillet is created), then it is excluded from the "Number of fillets rejected due to DRC errors" total.

Error Reporting for Dielectric Generation

The log file reflects the parameter sets defined for the execution and the glossing activity that occurred, as shown in the following example.

```
***Parameters for trace layer: TOP  
  
Crossover Layer: INTERNAL-SIGNAL4  
  
1st Dielectric Layer: INTERNAL-SIGNAL9  
  
X Size: 15 Y Size: 30 Incremental: Yes  
  
Dielectric created at location 475 1500 of size 33 wide by 48 high for nets: GND TN-31  
  
Can't correct drc - both nets not movable: TN-31 VCC  
  
Drc at location 600 1500  
  
TN-31 not movable due to etch already moved to crossover layer.  
  
VCC not movable due to an attached pad not going through to crossover layer.  
  
Dielectric created at location 1200 1438 of size 66 wide by 18 high for nets TN-3 TN-31
```

Related Topics

- [report](#)

Using the Allegro PCB Router Translator

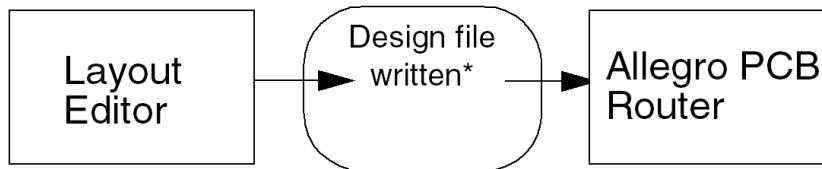
The Allegro® PCB Router Translator, a.k.a the SPECCTRA Interface (SPIF), lets you:

- Translate and export data from an Allegro .brd or .mcm, to an Allegro PCB Router design (.dsn) file.
The design file contains information needed by Allegro PCB Router to place and route the design.
- Translate and import data from an Allegro PCB Router session (.ses) file to a .brd or .mcm file.
The session file contains information regarding any placement or routing changes made to the design during the router session. The session file is all that is needed to read place and route data back into Allegro.

You can run the translator interactively from within Allegro or from your system console.

The following figure shows the translation in both directions.

Translation



Layout Editor to Allegro PCB Router translation



Allegro PCB Router to Layout Editor translation

* Design and Session files are similar, but design (.dsn) files contain detailed pre-route information for the purpose of identifying new place-and-route (and gate/pin swap) data. Session files contain information related to changes in placement/routing.

Prerequisites to Running the Translator

To minimize the potential for violations and mapping errors, prepare the Allegro layout files *before* running the translator. You can perform these preparations in any order:

- Set up constraints and properties as desired in the Allegro X layout editor.

⚠ Special constraints that are defined in Allegro PCB Router do not get passed back to Allegro when you import the .ses file.

- Set plane layers to *Plane* (other layers are translated as *Signal* by the translator).
- Edit null padstacks in Allegro to avoid generating errors or warnings when starting Allegro PCB Router.
- Set the analysis modes in Constraint Manager as desired to control constraint translation and rule checking within the router.
- Run the pre-route checker in Allegro.

Restrictions and Considerations

In addition to the prerequisites listed previously, be aware of the following before running the translator.

- Pad suppression is not implemented in Allegro PCB Router. Therefore, unused pads that were suppressed in the layout editor prior to translation are restored in Allegro PCB Router, used to route the design, and may produce:
 - valid but unexpected routing results.
 - error messages in Allegro PCB Router if pre-routes were created in the layout editor with pad suppression switched on.
- Teardrops that are protected in the layout editor are translated as wiring polygons to Allegro PCB Router and interpreted as conductive areas during routing. Any teardrops that are unprotected are removed during the automatic routing process. On the return trip back to the layout editor, the new routes are imported into the design devoid of any wiring polygons (teardrops) used in routing. However, teardrops can be automatically re-inserted by turning on the Dynamic Fillets option in the layout editor prior to routing.
You can also protect the teardrops in a design during the round-trip from the layout editor to Allegro PCB Router and back by doing one of the following.
 - Using Route Automatic: Protect all layers.
 - Etch editing and further routing within Allegro PCB Router:
 - mark nets as FIXED.
 - or -
 - when saving the resulting wires file within Allegro PCB Router, choose the option to save just unprotected (new) wires. Existing wires with teardrops are not altered.
- The FIXED property in the layout editor translates to a net of TYPE FIX in Allegro PCB Router. It cannot be changed in Allegro PCB Router.
- The translator "protects" pre-existing etch in the layout editor . Etch that is carried over to Allegro PCB Router can be unprotected, then removed/modified there.
- The translated board file should not be changed between the time the Allegro PCB Router design file is written and the session file is read back into the layout editor .
- A session file must be written before exiting Allegro PCB Router to update the Allegro .^{brd} or .^{mcm} file.

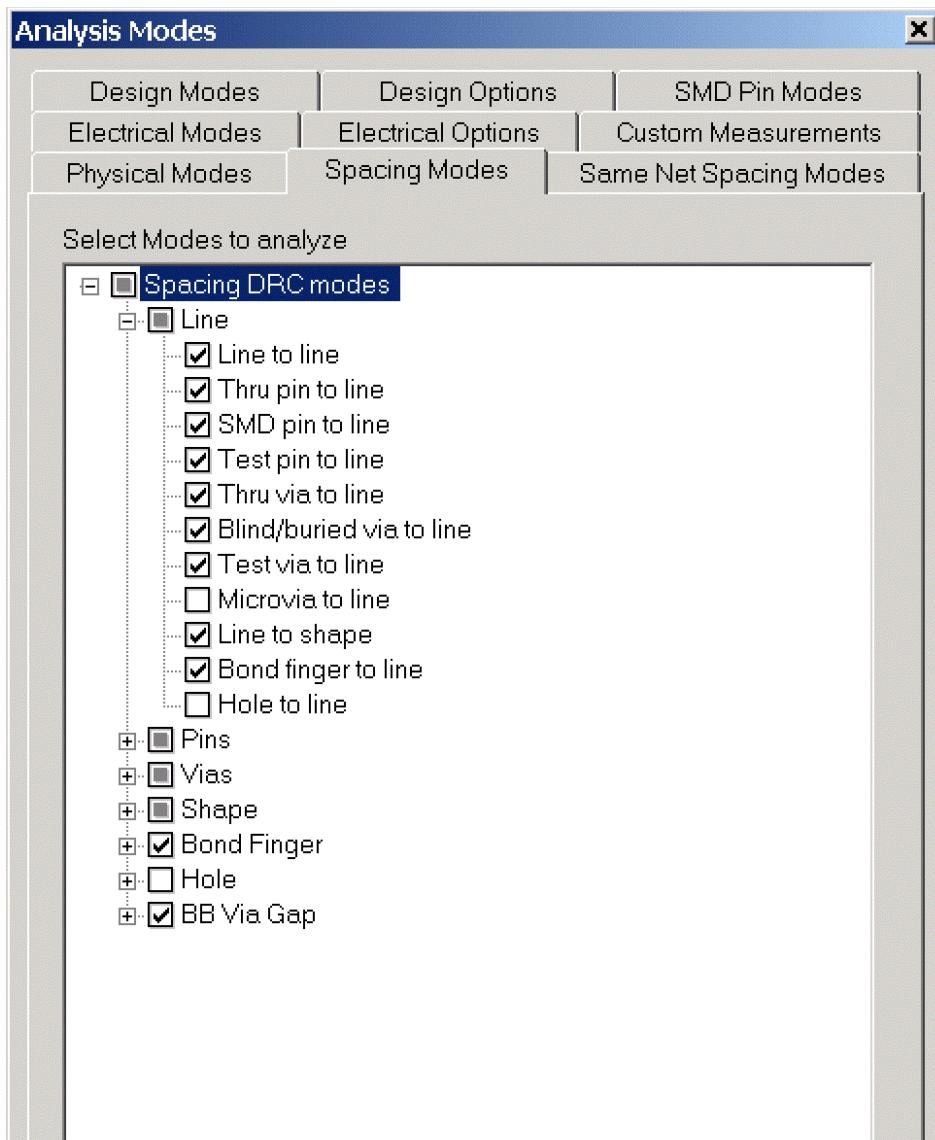
- Only place and route results (not rules) created in Allegro PCB Router translate to the layout editor.
- Electrical constraints are not passed from the layout editor to Allegro PCB Router if you set DRC mode to *Off* in the *Analysis Modes* dialog box, accessed when you choose *Setup – Constraints – Modes*. You must have all DRC modes set to *On* or *Batch* to pass electrical constraints.
- Dynamic shapes passed from the layout editor to Allegro PCB Router without the **FIXED** property enable Allegro PCB Router to allow wires and vias inside their area fills.
- All shapes with voids passed from Allegro PCB Router back to the layout editor behave in the following ways:
 - For shapes with the *Dynamic Fill* parameter set to *Disabled*; voiding does not take place.
 - For shapes with the *Dynamic Fill* parameter set to either *Rough* or *Smooth*; voiding automatically takes place.

Analysis Mode Settings in Constraint Manager

Constraints

You can control whether certain non-HDI routing constraints are translated to the Allegro PCB Router. You can also control whether the Allegro PCB Router rule checking for HDI routing constraints is turned on or off during routing. You do this by enabling or disabling DRC modes within Constraint Manager before translation to the Allegro PCB Router. Choose *Analyze – Analysis Modes* from the menu bar. The Analysis Modes dialog box appears as shown in the following figure. The following table describes how these settings control constraint and rule checking translation.

Analysis Modes Dialog Box



Constraint Translation and Rule Checking Control

DRC Mode Setting	Non-HDI Constraints	HDI Constraints
On	Constraint is translated.	Constraint is translated and corresponding rule check is turned on in the router.
Off	Design-level constraint is translated and -1 assigned as a value in the router. Otherwise, constraint is not translated.	Constraint is translated and corresponding rule check is turned off in the router.

Example

These spacing constraint / DRC mode settings translate to the following clearance rules within the router.

Allegro / Constraint Manager

Constraint / DRC Mode	Value / Setting
Design-level Constraints	
line to line	5
line to pin	6
hole to line	7
hole to pin	8
Net 1 Constraints	
line to line	5.5
line to pin	6.5
hole to line	7.5
hole to pin	8.5
Spacing DRC	
line to line	on
line to pin	off
hole to line	on
hole to pin	off

Allegro PCB Router

```
rule PCB (clearance 5 (type wire_wire))  
  
rule PCB (clearance -1 (type wire_pin))  
  
rule PCB (clearance 7 (type nhole_wire))  
  
set nhole_wire on  
  
rule PCB (clearance 8 (type nhole_pin))  
  
set nhole_pin off  
  
rule net NET1 (clearance 5.5 (type wire_wire))  
  
rule net NET1 (clearance 7.5 (type nhole_wire))  
  
rule net NET1 (clearance 8.5 (type nhole_pin))  
  
set same_net_checking on
```

⚠ When at least one Same Net Spacing DRC mode is turned on in Constraint Manager, same net rule checking is enabled in the Allegro PCB Router for the design. The following rule appears before corresponding same net clearance rules in the router.

Running the Pre-Route Checker

Before routing a design in Allegro PCB Router, you should perform a pre-route check to help ensure that the design does not contain non-conforming conditions that may cause Allegro PCB Router to fail or route poorly. The check identifies deficiencies in a design that could cause the router to fail and finds the layout editor constraints that do not correspond with constraints in Allegro PCB Router.

You can use one of these menu items or commands to run routing and alignment checks:

- *Route – Router Checks* (`specctra checks` command) runs the checker from the layout editor on the current design.
- `spif` is a batch command that runs the checker against the specified design. You can also run the Allegro PCB Router with this command.
- `spif_batch` is a batch command that runs the checker against the specified design.

With *Route – Router Checks* (`specctra checks` command) and the `spif` command, the pre-route check displays a list of warnings and errors that provides you with the opportunity to remove or otherwise work around non-conforming constraints.

Listing of Pre-Route Design Checks

The following design checks are performed for Allegro PCB Router when you run the pre-route program. Additional checks may be added in subsequent versions to identify non-conforming conditions in the two systems.

- **checkKeepin**
route_keepin is not defined.
- **checkFiguresOnEtch**
There are figures on ETCH subclass.
- **checkVias**
There are no vias available for routing. Checks for default vias only.
- **check0LineWidth**
One or more nets have a default line of zero units.
- **checkConsOutsideKeepin**
One or more connection points of a route is outside the route_keepin area. Connections are checked for placed components only.
- **checkPositiveShapes**
One or more routing layers have more than 50% of their areas covered with positive shapes.
- **checkPinsUnderKeepout**
A through pin is blocked by ROUTE_KEEPOUT ALL, or a surface-mount pin is blocked by KEEPOUT on its layer.
- **checkDanglingConnections**
Unconnected clines/vias that would translate to Allegro PCB Router as traces/vias on an orphan net. This can cause unwanted blockages when routing.
- **checkXtalkTable**
DRC is turned on, but no crosstalk table (`design_name.xtb`) file can be found.
- **checkRatTs**
One or more Rat Ts are found without a FIXED_T_TOLERANCE property set.
- **checkPadstackShape**

A high number of custom shapes are found in one or more padstacks. This can cause possible degradation in autorouter performance.

- **checkHighNumberDRC**

A high number of pre-existing DRC errors are found. This can cause possible degradation in autorouter performance.

- **checkShapeVertices**

A high number of vertices are found on one or more shapes on a signal layer. This can cause possible degradation in autorouter performance.

- **checkDiffPairConstraintArea**

One or more constraint areas are found. Differential pair rules in constraint areas are not followed by the Allegro PCB Router.

Running the Translator

You can run the translator in one of two ways.

You can choose *File – Import – Router* (`specctra_in` command) or *File – Export – Router* (`specctra_out` command) from within layout editor when you want to import to or export from the active database.

Alternatively, you can transfer design data between the layout editor and Allegro PCB Router using the `spif` command with the `-io` argument at an operating system prompt.

Translation Procedures

Layout Editor to Allegro PCB Router

1. In Allegro, set up a board (`.brd`) or multi-chip module (`.mcm`) file (see [Prerequisites to Running the Translator](#)) to translate.
2. Choose *File – Export – Router* (`specctra_out` command) from within Allegro
or
run `spif` from an operating system prompt.
The SPECCTRA Automatic Router Open dialog box appears.
3. Enter the name of the Allegro file to translate into the dialog box, then click *Open*.
The translator starts and a design (`.dsn`) file is generated in your working directory.
4. Launch Allegro PCB Router and load the resulting design (`.dsn`) file.
5. Place and route the design as required. You may want to set up routing controls within Allegro PCB Router using "Do" files.

Allegro PCB Router to the Layout Editor

1. In Allegro PCB Router, choose *File – Write – Session* to write out a session (`.ses`) file for the design translation.
2. In Allegro, choose *File – Import – Router* (`specctra_in` command)
or
run `spif` from an operating system prompt.
The Import from Auto-Router dialog box appears.
3. Enter the name of the session file generated in step 1, then click *Run*.
The translator starts and the design appears in the Allegro canvas.
4. Choose *Tools – Update DRC* (`drc update` command) to update the design DRCs.

Related Topics

- [Mapping Properties, Assignment Tables, Rule Sets, and Constraints.](#)
- [specctra checks](#)
- [spif](#)
- [spif_batch](#)
- [specctra_in](#)
- [specctra_out](#)
- [spif](#)

Mapping of Properties, Assignment Tables, Rule Sets, and Constraints

This section describes how the layout editor properties, assignment tables, and rule sets are mapped to corresponding elements in the Allegro PCB Router design file.

Layout Editor Properties to Allegro PCB Router

The SPIF translator extracts the following the layout editor properties:

ALT_SYMBOLS

BUS_NAME

ECL

ELECTRICAL_CONSTRAINT_SET

ETCH_TURN_UNDER_PAD

FIXED

FIXED_T_TOLERANCE

LAYER_SET_GROUP

MAX_VIA_COUNT

MIN_LINE_WIDTH

NO_PIN_ESCAPE

NO_RIPUP
NO_ROUTE
NO_SWAP_GATE
NO_TEST
PINUSE
PIN_DELAY
PROPAGATION_DELAY
RELATIVE_PROPAGATION_DELAY
ROOM
ROUTE_PRIORITY
SHIELD_NET
SHIELD_NET (user defined)
SHIELD_TYPE
STUB_LENGTH
TS_ALLOWED
VIA_AT_SMD_FIT
VIA_AT_SMD_THRU
VIA_LIST

The mapping of these properties to a Allegro PCB Router design file are detailed below.

ALT_SYMBOLS	This property translates to a pair of image descriptors in Allegro PCB Router that specify symbol images for top side and back side component placement.
-------------	--

Layout Editor:

```
CLASS DISCRETE  
  
PACKAGE RES400  
  
PACKAGEPROP ALT_SYMBOLS '(RES500;B:RES400B)'
```

Allegro PCB Router:

```
(image RES400_-_RES400B  
(side front)
```

```
...
)
(image RES400_-_RES400B
(side back)
...
)
```

In this example, the translator defines front and back side images for Allegro PCB Router, but uses the same name for both, with a side descriptor to specify image usage for either front or back placement. The translator combines the layout editor package symbol name (RES400) and the first symbol name following the "B" in the ALT_SYMBOLS property value (RES400B) to form the Allegro PCB Router image name (RES400_-_RES400B).

BUS_NAME	This property translates to a Allegro PCB Router class that contains all nets that have an identical BUS_NAME value.
----------	--

Layout Editor:

Net Sig1:

BUS_NAME = DATA

Net Sig2

BUS_NAME = DATA

Net Sig3

BUS_NAME = DATA

Allegro PCB Router:

(class DATA Sig1 Sig2 Sig3)

NO_SWAP_GATE	Components with this property do not have any of their gates swapped, either within the component or with gates in other components.
--------------	--

Layout Editor:

```
component U1: NO_SWAP_GATE = YES
```

Allegro PCB Router:

```
(component DIP14
(place U1 ... (property (lock_type gate))))
```

NO_SWAP_PIN	Components with this property do not have any of their pins swapped with other pins in the component.
-------------	---

Layout Editor:

```
component U1: NO_SWAP_PIN = YES
```

Allegro PCB Router:

```
(component DIP14  
  (place U1 ... (property (lock_type pin))))
```

ROOM	Components with this property are included in a Allegro PCB Router floor planning room.
------	---

Layout Editor:

```
component U1: room = left_side
```

Allegro PCB Router:

```
(floor_plan  
  (room left_side  
    (polygon ...)  
    (include U1 ... (property (type hard))  
    (exclude remain)))
```

PROPAGATION_DELAY	Nets with this property have a length rule in the Allegro PCB Router design file. Pin pairs included in the layout editor PROPAGATION_DELAY translate as fromto descriptors in the Allegro PCB Router design file. Within the same net, pin pairs are (type soft); in extended nets (XNets), a group is created.
-------------------	--

Layout Editor:

```
C18.1:U6.4:800:1000:U16.2:U4.3:900:1100:
```

Allegro PCB Router:

```
(net SIG_1  
  (pins C18-1 U16-2 U4-3 U6-4)  
  (fromto C18-1 U6-4 (circuit (length 1000 800))  
   (fromto U16-2 U4-3 (circuit (length 1100 900))))
```

)

If pin pairs are not included in the PROPAGATION_DELAY, fromto descriptors are created for AD:AR or D:R formats.

Layout Editor:

```
::1500:2000
```

Allegro PCB Router:

```
(net SIG_1
  (pins C18-1 U16-2 U4-3 U6-4)
  (circuit (length 2000 1500)
)
```

If the delay value is specified in time instead of length (for example, ns), then the translator calculates and assigns a Time/Length Factor to the design file so that Allegro PCB Router can translate the time to an appropriate length.

Layout Editor:

```
ECset ESET1:
```

```
  Primary Gap = 8 mil
  Line Width = 5 mil
  Neck Gap = 6 mil
  Neck Width = 4 mil
  Coupled Tolerance (+) = 3 mil
  Coupled Tolerance (-) = 2 mil
  Min Line Spacing = 1 mil
  Gather Control = Ignore
  Max Uncoupled Length = 25 mil
  Phase Control = Dynamic
  Phase Tolerance = 15 mil
```

Net Sig1:

```
  DIFF_PAIR_NAME = MyPair
```

```
ELECTRICAL_CONSTRAINT_SET = ESET1  
PRIMARY_GAP = 9 mil  
  
Net Sig2  
  
DIFF_PAIR_NAME = MyPair  
ELECTRICAL_CONSTRAINT_SET = ESET1
```

Allegro PCB Router:

```
define (pair (nets Sig1 Sig2))  
  
define (class ESET1 Sig1 Sig2)  
  
rule class ESET1 (edge_primary_gap 8)  
rule class ESET1 (diffpair_line_width 5)  
rule class ESET1 (neck_down_gap 6)  
rule class ESET1 (neck_down_width 4)  
rule class ESET1 (edge_coupling_tolerance_plus 3)  
rule class ESET1 (edge_coupling_tolerance_minus -2)  
rule class ESET1 (clearance 1 (wire_wire))  
rule class ESET1 (ignore_gather_length on)  
rule class ESET1 (max_uncoupled_length 25)  
rule class ESET1 (phase_control on)  
rule class ESET1 (phase_tolerance 15)  
rule net Sig1 (edge_primary_gap 9)
```

ECL

Nets with the ECL property translate to a series of fromto descriptors and a reorder daisy net rule in Allegro PCB Router.

Allegro PCB Router:

```
(net SIG-1  
(pins C18-1 U16-2 U4-3 U6-4)  
(fromto C18-1 U4-3)  
(fromto U4-3 U6-4)
```

Routing the Design

Using the Allegro PCB Router Translator--Mapping of Properties, Assignment Tables, Rule Sets, and Constraints

```
(fromto U6-4 U16-2)  
(rule (reorder daisy))  
)
```

ELECTRICAL_CONSTRAINT_SET	This property determines the electrical rule set for this net. If this property is not defined, the default electrical rule set is used.
ETCH_TURN_UNDER_PAD	Components, symbols, or pins with this property enabled (on) are allowed to have wires route and turn under their pads.

Layout Editor:

SYMBOL: DIP14_3 at 1300.00,1800.00 ETCH_TURN_UNDER_PAD = ON

Allegro PCB Router:

```
component_property U1 (turn_under_pad on)  
rule pcb (turn_under_pad on)
```

FIXED	Nets with this property are type fix in Allegro PCB Router. Allegro PCB Router cannot move, alter, or route to any point on these nets. Nets with this property cannot be deleted or replaced by the cct2cadence program.
-------	---

Allegro PCB Router:

```
(net SIG_1  
(pins C18-1 U16-2 U4-3 U6-4)  
(type fix)  
)
```

If nets are type "fix" and not routed, Allegro PCB Router does not recognize them as unconnects, although the layout editor does. Therefore, you can have a report of 100% completion in Allegro PCB Router, but when routing is merged with the layout editor file, the completion rate is less than 100%.

FIXED_T_TOLERANCE	Nets with this property have a RADIUS rule in Allegro PCB Router. Allegro PCB Router cannot move the location of any Rat-Ts outside the specified tolerance radius.
-------------------	---

Layout Editor:

FIXED_T_TOLERANCE = 50 mil

Allegro PCB Router:

```
(virtual_pin TR14.2_Sig1  
(position 1500 2500 (radius 50))  
)
```

LAYERSET_GROUP

Nets with this property have a USE_LAYER rule in Allegro PCB Router.

Layout Editor:

LAYERSET_GROUP = LS1:LS2

Allegro PCB Router:

```
(circuit net SIG_1  
(use_layer LS1 LS2)  
)
```

MAX_VIA_COUNT

Nets with this property have a MAX_VIA_COUNT net rule in Allegro PCB Router.

Layout Editor:

MAX_VIA_COUNT = 5

Allegro PCB Router:

```
(net SIG_1  
(pins C18-1 U16-2 U4-3 U6-4)  
(rule (max_via_count 5))  
)
```

MIN_LINE_WIDTH

Nets with this property have a width net rule in Allegro PCB Router.

Allegro PCB Router:

```
(net SIG_1
```

Routing the Design

Using the Allegro PCB Router Translator--Mapping of Properties, Assignment Tables, Rule Sets, and Constraints

```
(pins C18-1 U16-2 U4-3 U6-4)  
(rule (width 10))  
)
```

NO_RIPUP	All wires and vias that have this property are type protect in Allegro PCB Router. Wires and vias assigned this property cannot be altered by Allegro PCB Router. The autorouter can route to a terminal of a protected wire, or if tjunctions are allowed, tee off a protected segment.
----------	--

Allegro PCB Router:

```
(wire  
(path TOP 10 8043 9090 8103 9090 8103 9179)  
(net SIG_1) (type protect)  
)
```

NO_ROUTE	Nets with this property are of type fix in Allegro PCB Router. Allegro PCB Router cannot route nets that have this property and cannot move or alter routes that have this property.
----------	--

Allegro PCB Router:

```
(net SIG_1  
(pins C18-1 U16-2 U4-3 U6-4)  
(type fix)  
)
```

When you assign a NO_ROUTE property to a net in the layout editor , the property translates to a type fix in Allegro PCB Router. Allegro PCB Router does not alter nets that have the fix property. If a net with the fix property is not prerouted, it remains unrouted. Nets with the FIXED property cannot be completed by Allegro PCB Router unless you apply the unfix command.

PINUSE	This pin property translates to source and load keywords in a Allegro PCB Router net descriptor statement. Values of OUT, OCA, OCL, and TRI translate to Allegro PCB Router source keywords. Values of NC, IN, UNSPEC, and BI translate to Allegro PCB Router load keywords.
--------	--

Layout Editor:

NET: SIG1

PINS: U1-1 U2-2 U3-1 U4-1

PIN: U3-1;

PROPERTY: PINUSE

VALUE: OUT

Allegro PCB Router:

```
(net SIG
  (pins U1-1 U2-1 U3-1 U4-1)
  (source U3-1)
  ...
)
```

In this example, the layout editor assigns the PINUSE property with a value of "OUT" to pin U3-1. The translator defines U3-1 as a source pin in the Allegro PCB Router net descriptor.

The translator ignores the PINUSE property if its associated pins are in a net that has an ECL property.

ROUTE_PRIORITY	Nets with this property have a priority net rule in Allegro PCB Router. In the layout editor , the lower the value the higher the route priority. A value of 1 is the highest the layout editor route priority. In Allegro PCB Router, the higher the value the higher the route priority. A value of 255 is the highest Allegro PCB Router route priority. If the route priority has a value of 255 or greater, the route priority maps to Allegro PCB Router as 1 (a low Allegro PCB Router route priority). Otherwise, route priority is calculated as follows:
----------------	--

Allegro PCB Router route priority = (256 – Layout Editor route priority)

For example, if the route priority in the layout editor is 5, the Allegro PCB Router route priority is represented as

```
(net SIG_1
  (pins C18-1 U16-2 U4-3 U6-4)
  (circuit (priority 251))
)
```

STUB_LENGTH	Nets that have a STUB_LENGTH property with a value greater than 0 have a max_stub net rule in Allegro PCB Router.
-------------	---

Allegro PCB Router:

```
(net SIG_1  
(pins C18-1 U16-2 U4-3 U6-4)  
(rule (max_stub 50))  
)
```

TS_ALLOWED	Nets with this property have a tjunction net rule in Allegro PCB Router. There are four different settings for tjunctions in the layout editor .
------------	--

not_allowed	No tjunctions are allowed on these nets.
-------------	--

Allegro PCB Router:

```
(net SIG_1  
(pins C18-1 U16-2 U4-3 U6-4)  
(rule (tjunction off))  
)
```

anywhere	Tjunctions are allowed anywhere on these nets.
----------	--

Allegro PCB Router:

```
(net SIG_1  
(pins C18-1 U16-2 U4-3 U6-4)  
(rule (tjunction on) (junction_type all))  
)
```

pins_only and pins_and_vias_only	Both pins_and_vias_only and pins_only allow tjunctions at terminals only (pins and vias).
----------------------------------	---

Allegro PCB Router:

```
(net SIG_1  
(pins C18-1 U16-2 U4-3 U6-4)  
(rule (tjunction on) (junction_type term_only))  
)
```

VIA_AT_SMD_FIT	Symbols or symbol pins with this property have a via_at_smd rule in Allegro PCB Router. When this property is enabled (on), a via at an SMD pads is permitted only if it is completely covered by the pad. When disabled (off), a via at an SMD pad is permitted only if its center is inside the pad. Wildcards are not allowed when you use this property. Note that this property must be user-defined for components or component pins in the router.
----------------	---

Layout Editor:

SYMBOL: DIP14_3 at 1300.00,1800.00 VIA_AT_SMD_FIT = ON

Allegro PCB Router:

```
component_property U1 (via_at_smd_fit on)
rule pcb (via_at_smd on
(fit on))
```

VIA_AT_SMD_THRU	Symbols or symbol pins with this property have a via_at_smd rule in Allegro PCB Router. When this property is enabled (on), a thru via, blind via, or microvia at an SMD pad is permitted. When disabled (off), a thru via, blind via, or microvia at an SMD pad is not permitted. Wildcards are not allowed when you use this property. Note that this property must be user-defined for components or component pins in the router.
-----------------	---

Layout Editor:

SYMBOL: DIP14_3 at 2300.00,2600.00 VIA_AT_SMD_THRU = ON

Allegro PCB Router:

```
component_property U2 (via_at_smd_thru on)
rule pcb (via_at_smd on
(thru on))
```

VIA_LIST	Nets with this property have a use_via net rule in Allegro PCB Router. Only vias in the via_list are used by Allegro PCB Router to route a net with this property. Wildcards are not allowed when you use this property.
----------	--

Layout Editor:

```
via_list - vial, testvia
```

Allegro PCB Router:

```
(net SIG_1  
(pins C18-1 U16-2 U4-3 U6-4)  
(circuit (use_via vial testvia)))
```

NO_PIN_ESCAPE	Nets with this property have a 'noexpose' rule in the Allegro PCB Router design file. All pins in the net are included in the 'noexpose' rule and will not be fanned out.
---------------	---

Allegro PCB Router:

```
(net SIG1  
(pins U1-3 R3-1 U1-4)  
(noexpose U1-3 R3-1 U1-4)  
)
```

NO_TEST	Nets with this property have the testpoint rule 'insert off' in the Allegro PCB Router design file. The Allegro PCB Router does not insert testpoints on nets that have this property.
---------	--

Allegro PCB Router:

```
(net SIG1  
(pins U1-3 R3-1 U1-4)  
(rule (testpoint (insert off)))  
)
```

Routing the Design

Using the Allegro PCB Router Translator--Mapping of Properties, Assignment Tables, Rule Sets, and Constraints

SHIELD_NET	Nets with this property have the circuit rule 'shield' in the Allegro PCB Router design file. The value of this property is the name of the net to use for shielding the net that the property is attached to.
------------	--

Layout Editor:

```
SHIELD_NET = GND
```

Allegro PCB Router:

```
(net SIG1  
  (circuit (shield on (use_net GND)))  
)
```

SHIELD_TYPE	Nets with this property include a type for the circuit rule 'shield' in the Allegro PCB Router design file. The value should be one of the Allegro PCB Router keywords "parallel", "tandem", or "coax".
-------------	---

Layout Editor:

```
SHIELD_NET = GND  
SHIELD_TYPE = PARALLEL
```

Allegro PCB Router:

```
(net SIG1  
  (circuit (shield on (use_net GND) (type parallel)))  
)
```

PIN_DELAY	Pins with this property include a 'pin_length' rule in the Allegro PCB Router design file. The value should represent the internal length from the package to the pin. This length will be considered when calculating length rules in the Allegro PCB Router.
-----------	--

Layout Editor:

```
PIN_DELAY on U1-3 = 2 mill
```

Allegro PCB Router:

```
(net SIG1  
  (pins U1-3 R3-1 U1-4))
```

```
(pin_length 2 u1-3))
```

Layout Editor Rule Sets to Allegro PCB Router

The layout editor spacing rules map to the Allegro PCB Router as pcb rules, class rules, or class_class rules. The layout editor physical rules map to Allegro PCB Router as pcb rules or class rules.

Spacing Rules

If a spacing rule set is the default set (that is, no_type, notype, no_type entry), this maps to Allegro PCB Router pcb rules.

Types of clearance rules that translate to Allegro PCB Router include:

wire_wire	area_wire	smd_smd	pin_pin	via_via	test_wire	bbvia_bbvia
wire_smd	area_smd	smd_pin	pin_via		test_tes	bbvia_smdpin
wire_pin	area_area	smd_via			test_smd	bbvia_smdpinn
wire_via	area_pin				test_pin	bbvia_thruvia
	area_via				test_via	bbvia_testvia
	area_test					bbvia_testpin
						bbvia_bondpad
						bbvia_area

Spacing rule set examples (in Allegro PCB Router):

```
rule pcb (clearance 6 (type wire_wire))
rule pcb (clearance 6 (type wire_smd))
rule pcb (clearance 6 (type wire_area))
rule pcb (clearance 6 (type wire_pin))
rule pcb (clearance 7 (type wire_via))
rule pcb (clearance 6 (type area_smd))
rule pcb (clearance 6 (type area_area))
```

```
rule pcb (clearance 6 (type area_pin))
rule pcb (clearance 7 (type area_via))
rule pcb (clearance 6 (type smd_smd))
rule pcb (clearance 6 (type smd_pin))
rule pcb (clearance 7 (type smd_via))
rule pcb (clearance 6 (type pin_pin))
rule pcb (clearance 7 (type pin_via))
rule pcb (clearance 7 (type via_via))
rule pcb (clearance 10 (type bbvia_via))
rule PCB (clearance 0.12 (type bbvia_testvia))
rule PCB (clearance 0.12 (type bbvia_testpin))
rule PCB (clearance 0.12 (type bbvia_bondpad))
rule PCB (clearance 0.12 (type bbvia_area))
```

Physical Rules

The layout editor physical rules that translate include:

- Allow etch on subclass
 - If one or more signal layers have this rule turned off (no routing on those layers), this rule maps to Allegro PCB Router as a circuit net rule. All signal layers that have this rule turned on are included in a Allegro PCB Router use_layer rule.

Allegro PCB Router:

```
(class SIG_1
  (pins C18-1 U16-1 U4-3 U6-4)
  (circuit (use_layer TOP SIG_3 BOTTOM))
)
```

- Buried Via Gap maps to Allegro PCB Router as a spacing rule.

Allegro PCB Router:

(rule (clearance 3 (type buried_via_gap))

- Blind and Buried Via Gap maps to Allegro PCB Router as a spacing rule.

Allegro PCB Router:

(rule (clearance 3 (type bbvia_gap))

- Minimum Line Width maps to Allegro PCB Router as a width rule.

Allegro PCB Router:

(rule (width 10))

- Minimum Neck Width maps to Allegro PCB Router as a neck_down_width rule.

Allegro PCB Router:

(rule (neck_down_width 0.003))

- Neck Gap maps to Allegro PCB Router as a neck_down_gap rule.

Allegro PCB Router:

(rule (neck_down_gap 0.003))

- Primary Gap maps to Allegro PCB Router as a edge_primary_gap rule.

Allegro PCB Router:

(rule (edge_primary_gap 0.005))

- Tjunctions map to Allegro PCB Router as a tjunction rule. the layout editor has four different settings for tjunctions:

- Not_Allowed

No tjunctions are allowed.

Allegro PCB Router:

(rule (tjunction off))

- Anywhere

Tjunctions are allowed anywhere on a wire segment.

Allegro PCB Router:

(rule (tjunction on))

(junction_type all))

- Pins_Only

- Pins_and_Vias_Only

Both of these settings allow tjunctions at terminals only (pins and vias)

Allegro PCB Router:

```
(rule (tjunction on))
```

```
(junction_type term_only))
```

- Tolerance + maps to Allegro PCB Router as an edge_couple_tolerance_plus rule.

Allegro PCB Router:

```
(rule (edge_couple_tolerance_plus -1))
```

- Vialist - maps to Allegro PCB Router as an use_via rule.

Allegro PCB Router:

```
circuit region R1 (use_via V40)
```

 This rule is supported for all levels of the rules hierarchy.

- pad-pad connect - maps to Allegro PCB Router as an stacking rule.

Allegro PCB Router:

```
rule layer SIG1 (stack_via on overlap_only microvia_only)
```

 This rule is supported for all levels of the rules hierarchy.

- Tolerance - maps to Allegro PCB Router as an edge_couple_tolerance_minus rule.

Allegro PCB Router:

```
(rule (edge_couple_tolerance_minus -1))
```

- Via_List

The default physical rule set maps to Allegro PCB Router as a via list. Allegro PCB Router can use any via in the list for autorouting. If a physical rule set is not the default set, vias in the list are represented in the Allegro PCB Router design file as spares.

Allegro PCB Router:

```
(via VIA
```

```
(spare VIA1 VIA_TEST) )
```

Spare vias can be assigned to a net in Allegro PCB Router by using a circuit command. If a net uses a physical rule set that is not the default, the layout editor VIA_LIST maps to Allegro PCB Router as a circuit net rule, and all vias in the list are included.

Allegro PCB Router:

```
(net SIG_1  
(pins C18-1 U16-2 U4-3 U6-4)  
(circuit (use_via VIA_TEST VIA1))
```

Electrical Rules

The electrical rule that translates to Allegro PCB Router is:

max_stub_length	If this rule has a value greater than 0, and the nets that use this rule set don't have a max_stub property, they are assigned a max_stub net rule in Allegro PCB Router.
-----------------	---

Allegro PCB Router:

```
(net SIG_1  
(pins C18-1 U16-2 U4-3 U6-4)  
(rule (max_stub 50))  
)
```

Same Net Spacing Rules

Each Allegro spacing rule, both HDI and non-HDI has a same_net clearance rule equivalent in the router. The format used in the router for same net rules is:

```
<clearance descriptor> ::=  
(clearance <positive_dimension> [same_net] [(type{<clearance_type>})])
```

When at least one Same Net Spacing DRC mode is turned on in the Analysis Modes dialog box within Constraint Manager, same net rule checking is enabled in the router for the design as shown in the following example.

Allegro PCB Router:

```
set same_net_checking on
```

```
rule pcb (clearance 32 same_net (type wire_wire))  
rule pcb (clearance 10 same_net (type microvia_wire))  
rule PCB (clearance 0.1 same_net (type bbvia_area))
```

SMD Pin Rules

The SMD pin rules that translate to Allegro PCB Router are:

via_at_smd	If this rule is enabled (on), vias are allowed on SMDs in Allegro PCB Router. This rule is modified by the Allegro properties VIA_AT_SMD_FIT and VIA_AT_SMD_THRU. See Layout Editor Properties to Allegro PCB Router for further details.
------------	---

Allegro PCB Router:

```
rule pcb (via_at_smd on (fit on) (thru off))
```

etch_turn_under_pad	If this rule is enabled (on), wires are allowed to route and tune under SMD pads in Allegro PCB Router.
---------------------	---

Allegro PCB Router:

```
rule pcb (turn_under_pad on)
```

HDI Objects Support

Microvias

Allegro microvias are translated to the router using padstack micro objects. All microvia spacing rules are applied only to these objects.

Allegro PCB Router:

```
(padstack microvia_97  
(type micro)  
(plating plated)  
(shape (circle top 10 0 0))  
(shape (circle L1 10 0 0)))
```

Padstack Holes

Allegro holes are translated to the router using padstack hole shape objects. All net-based holes and mechanical hole rules are applied only to these objects.

Allegro PCB Router:

```
(padstack hole_125
  (plating plated)
    (shape (circle power 125 0 0))
    (shape (circle top 110 0 0))
    (antipad (circle top 157 0 0))
    (hole (circle signal 125))
    (shape (circle bottom 110 0 0))
    (antipad (circle bottom 157 0 0)))
```

Mechanical Pins / Holes

Standalone Allegro mechanical component and mechanical pins are translated to the router using image pin objects.

⚠ The *Enable Antipad as Route Keepouts (ARK)* Padstack Editor option in Allegro is mapped to PCB Router through the ARK option of the padstack_descriptor.

Allegro PCB Router:

```
(padstack PAD125
  (plating nonplated)
  (ark on)
  (type thrupad))
```

Layout Editor Constraints to Allegro PCB Router Rules

The following table shows the mapping of the layout editor constraint hierarchy to the router routing rules hierarchy. This mapping equivalence applies to Physical and Spacing constraints only.

Layout Editor to Allegro PCB Router Constraint Hierarchy Mapping

Design Constraint Level	Allegro PCB Router Rule Level	Comments
Design	pcb (or layer)	Layer rules are used when the constraint varies by layer.
NetClass	class	
Bus	class	The router only supports nets as members of a class. Only direct net members of a NetClass will be included in the router class.
Diffpair	class, pair	The router only supports nets as members of a class. Only direct net members of the diffpair will be included in the router class. The rules will be inherited from the parent NetClass (if applicable) and then overridden with the Diffpair rules. Additionally, a pair object is defined with the nets as members.
Xnet	n/a	Layout editor physical and spacing constraints on Xnets are flattened to net rules – unless overridden by net constraints.
Net	net	
PinPair	fromto	
Class-Class	class_class	
Region	region	
Region-Class	region_class	
Region-Class-Class	region_class_class	

The following table shows the mapping of the layout editor constraints to the Allegro PCB Router routing rules.

Layout Editor Non-HDI Constraint to Allegro PCB Router Rule Mapping

Layout Editor Constraint	Allegro PCB Router Rule	Comments
Buried Via to Buried Via Spacing	none	The router uses via_via
Buried Via Gap	buried_via_gap	
Buried Via to Line Spacing	none	The router uses wire_via
Buried Via to Pin Spacing	none	The router uses pin_via
Buried Via to Shape Spacing	none	The router uses area_via
Buried Via to SMD Spacing	none	The router uses smd_via
Etch Allowed	(see note below table)	(see note below table)
Line to Line Spacing	wire_wire	
Line to SMD Spacing	wire_smd	
Line to Pin Spacing	wire_pin	
Line to Via Spacing	wire_via	
Maximum BBVia Stagger	max_stagger	
Maximum Line Width	none	Not translated by SPIF
Maximum Neck Length	none	Not translated by SPIF
Minimum BBVia Stagger	staggered_via	
Minimum Line Width	width	
Minimum Neck Width	neck_down_width	
Neck Gap	neck_down_gap	
Pin to Pin Spacing	pin_pin	
Pin to Via Spacing	pin_via	
Primary Gap	edge_primary_gap	

Routing the Design

Using the Allegro PCB Router Translator--Mapping of Properties, Assignment Tables, Rule Sets, and Constraints

Shape to Line Spacing	area_wire	
Shape to Pin Spacing	area_pin	
Shape to Shape Spacing	area_area	
Shape to SMD Spacing	area_smd	
Shape to Test Via Spacing	area_test	
Shape to Via Spacing	area_via	
SMD to Pin Spacing	smd_pin	
SMD to SMD Spacing	smd_smd	
SMD to Via Spacing	smd_via	
Test Via to Test Via Spacing	test_test	
Test Via to Line Spacing	test_wire	
Test Via to SMD Spacing	test_smd	
Test Via to Pin Spacing	test_pin	
Test Via to Via Spacing	test_via	
Test Pin to Shape Spacing	none	The router uses area_test
Test Pin to Test Pin Spacing	none	The router uses test_test
Test Pin to Test Via Spacing	none	The router uses test_test
Test Pin to Line Spacing	none	The router uses test_wire
Test Pin to SMD Spacing	none	The router uses test_smd
Test Pin to Pin Spacing	none	The router uses test_pin
Test Pin to Via Spacing	none	The router uses test_via
Test Pin to Buried Via Spacing	none	The router uses test_via
Test Via to Buried Via Spacing	none	The router uses test_via
Ts Allowed	junction_type	

Routing the Design

Using the Allegro PCB Router Translator--Mapping of Properties, Assignment Tables, Rule Sets, and Constraints

Tolerance +	edge_couple_tolerance_plus	
Tolerance -	edge_couple_tolerance_minus	
Via to Via Spacing	via_via	
VIA to VIA Stacking	none	Not translated by SPIF

⚠ The layout editor Etch Allowed constraint's default value is `TRUE`. This means that etch is permitted on that layer. When `Etch Allowed` is set to `FALSE`, etch is not allowed on that layer. Both the layout editor and the router assume that all layers are available for routing, unless otherwise specified. However, in the router there is no way to directly restrict routing from a specific layer. This can be accomplished indirectly with the `use_layer` constraint, listing the layer names where routing is permitted and omitting the layer names where routing is not permitted.

The layout editor supports the `Etch Allowed` constraint within a Physical CSet and as an override on a region and region-class. In the router, the `use_layer` constraint can only be set on a class or net object. When the `Etch Allowed` constraint is `FALSE` for one or more layers within a Physical CSet and that CSet is referenced by the layout editor net-based object (class, bus, diffpair, xnet, net, pinpair), SPIF will generate the appropriate `use_layer` constraints on the translated router net, class, or fromto object.

SPIF does not translate an `Etch Allowed` constraint from a layout editor region or region-class.

Layout Editor HDI Constraint to Allegro PCB Router Rule Mapping

Design Editor Constraint	Allegro PCB Router Rule	Comments
Hole to Line Spacing	nhole_wire	
Hole to Shape Spacing	nhole_area	
Hole to Hole Spacing	nhole_nhole	
Hole to Pin Spacing	nhole_pin	
Hole to Via Spacing	nhole_via	
Mechanical Hole to Shape Spacing	mhole_area	

Routing the Design

Using the Allegro PCB Router Translator--Mapping of Properties, Assignment Tables, Rule Sets, and Constraints

Mechanical Hole to Line Spacing	mhole_wire	
Mechanical Hole to Mechanical Hole Spacing	mhole_mhole	
Mechanical Hole to Net-based Hole Spacing	mhole_nhole	
Mechanical Hole to Pin Spacing	mhole_pin	
Mechanical Hole to Via Spacing	mhole_via	
Microvia to Line Spacing	microvia_wire	
Microvia to SMD Spacing	microvia_smdpin	
Microvia to Microvia Spacing	microvia_microvia	
Microvia to Pin Spacing	microvia_thrupin	
Microvia to Test Pin Spacing	microvia_testpin	
Microvia to Via Spacing	microvia_thruvia	
Microvia to Blind/Buried Via Spacing	microvia_bbvia	
Microvia to Test Via Spacing	microvia_testvia	
Microvia to Shape Spacing	microvia_area	
Microvia to Bond Finger Spacing	microvia_bondpad	
bbvia to Line Spacing	bbvia_wire	
bbvia to SMD Spacing	bbvia_smdpin	
bbvia to Microvia Spacing	bbvia_microvia	
bbvia to Pin Spacing	bbvia_thrupin	
bbvia to Test Pin Spacing	bbvia_testpin	
bbvia to Via Spacing	bbvia_thruvia	
bbvia to Blind/Buried Via Spacing	bbvia_bbvia	
bbvia to Test Via Spacing	bbvia_testvia	
bbvia to Shape Spacing	bbvia_area	

Routing the Design

Using the Allegro PCB Router Translator--Mapping of Properties, Assignment Tables, Rule Sets, and Constraints

bbvia to Bond Finger Spacing

bbvia_bondpad

Troubleshooting Translation Problems

This section offers possible solutions to common translation difficulties. The information is presented in a question and answer format.

Q: After I run ‘Write Allegro PCB Router’, my padstacks are the wrong shape, or they overlap, or both. What causes this?

A: If the padstacks contain custom shapes, they’re translated to rectangles that fully enclose the custom shapes.

Q: Why does Allegro PCB Router route my power nets on signal layers?

A: This occurs when etch subclasses are not designated as power layers and set to type plane in the layout editor . Also, check whether you have shapes on the plane layers and whether the shapes are assigned to power nets.

Q: After I run ‘Update,’ I see DRC violations that are not present in Allegro PCB Router. Why?

A: If you change clearance rules in Allegro PCB Router (which is not recommended), update the DRC rules in the layout editor to match those in Allegro PCB Router before you run a design rule check.

Q: After I run ‘Update,’ my cross-hatched shapes look different. Why?

A: Allegro PCB Router does not support cross-hatched shapes, so it converts them to solid fills. When possible, convert cross-hatched shapes to solids in the layout editor before running the translator.

Q: After I run ‘Update,’ I have unroutes in the layout editor that were not in Allegro PCB Router. Why?

A: There may be voids in the power plane in the layout editor . Voids in shapes don’t translate into Allegro PCB Router. If these voids are unnecessary, delete them.

Q: Why does ‘Update’ sometimes run slowly?

A: This occurs when the layout editor ’s DRC checking is turned on AND there are a lot of electrical rules applied. Normally you would want to leave DRC checking on, but in this case, it translates faster if you turn it off in the layout editor first and then run ‘Update DRC’ after translating.

Automatic Routing with Allegro PCB Router

From the layout editor, you can automatically route all or part of your design (`.brd` or `.mcm` file) using Allegro PCB Router. Depending on the characteristics of your design and the level of routing interaction and control that you require, you can choose to follow one of three different Cadence recommended autorouting task flows.

Modes of Operation

Depending on the command that you choose for autorouting, the router is launched in either background or foreground mode. In background mode, the router works behind the scenes to route your design while you remain active in the layout editor. In foreground mode, the router becomes active to provide you with an enhanced level of interactivity and a higher degree of routing control.

You can begin autorouting by using one of the following commands:

Command	Menu Path	Mode of Operation	Level of User Interaction Required
<code>fanout_by_pick</code>	<i>Route – Fanout by Pick</i>	Background	Low
<code>route_by_pick</code>	<i>Route – Route Net(s) by Pick</i>	Background	Low
<code>elong_by_pick</code>	<i>Route – Elongation by Pick</i>	Background	Low
<code>auto_route</code>	<i>Route – Route Automatic</i>	Background	None
<code>custom_route</code>	<i>Route – Route Custom</i>	Foreground	Low to Medium
<code>specctra</code>	<i>Route – Route Editor</i>	Foreground	Medium to High
<code>miter_by_pick</code>	<i>Route – Miter by Pick</i>	Background	Low
<code>unmiter_by_pick</code>	<i>Route – Unmiter by Pick</i>	Background	Low

- i** Before running an autorouting command, you should review and complete all required procedures that apply to your design.

Related Topics

- [Autorouting Task Flows](#)
- [Prerequisites for Allegro PCB Router Automatic Routing](#)
- [Using the Allegro PCB Router Translator](#)
- [fanout_by_pick](#)
- [route_by_pick](#)
- [elong_by_pick](#)
- [auto_route](#)
- [custom_route](#)
- [specctra](#)
- [miter_by_pick](#)
- [unmiter_by_pick](#)

Autorouting File Generation

When you run an automatic routing command, the following file types of routing files are generated from your design and passed to the Allegro PCB Router for processing.

Type	Name	Purpose
Design	<filename>.dsn	Communicates design data regarding components, connectivity, and constraints to Allegro PCB Router.
Rules	<filename>_rules.do	Specifies design rules such as clearance, wiring, timing, cross-talk, and so on.
Forget Rules	<filename>_forget.do	Specifies certain rules defined in the <code>rules.do</code> file that can be ignored during the current routing session.

These files can be used as-is, or you can modify them to change routing rules and parameters or possibly relax or tighten constraints.

Related Topics

- [Autorouting with Do Files](#)

Autorouting Task Flows

Cadence recommends using one of the following autorouting task flows depending on the technology characteristics of your design and the level of interaction required.

Flow	Use for . . .
Mainstream	automatic routing of designs from within the layout editor where little to no user interaction is needed.
High-speed	routing high-speed designs using the router interface that require a certain degree of routing interactivity or possible editing of the various .do files created for Allegro PCB Router.
High-speed Power User	routing high-speed designs using the router interface that may require significant routing interactivity and editing of the various .do files created for Allegro PCB Router.

Mainstream Flow

The Mainstream Flow performs automatic routing on the entire design or certain portions of the design using the router in background mode.

 This flow does not allow for modification of routing files.

The following commands are considered part of the Mainstream Flow. Click on a command for

specific usage details described in the *Allegro PCB and Package Physical Layout Command Reference*.

Command	Use to ...
auto_route	route the entire design without further intervention.
route_by_pick	route certain portions of the design. For example, just critical nets.
fanout_by_pick	route short pin escape wires from pins to vias.
elong_by_pick	increase etch length to adhere to timing rules.
miter_by_pick	change 90-degree wire corners to 45 degrees for wires, exiting pins, and vias.
unmiter_by_pick	remove 45-degree wire corners and change them to 90-degree corners.

High-Speed Flow

Use this flow if you want to review and possibly modify the generated .do files, then route the design with Allegro PCB Router running in foreground mode.

- ⓘ Be aware that the original .do files generated and modified using this flow are automatically deleted when you terminate the current session of the layout editor. You must rename these .do files to have them retained between sessions.

The following commands are considered part of the High-speed Flow. Click on a command for specific usage details in the *Allegro PCB and Package Physical Layout Command Reference*.

Command	Use to ...
custom_route	generate and pass a set of custom rules files from your design to the router running in foreground mode to autoroute your design. For further details on this strategy, see the description for custom route in the <i>Allegro PCB and Package Physical Layout Command Reference</i> .
specctra	generate and pass a single (standard) rules file from your design to the router running in foreground mode to autoroute your design.

High-Speed Power User Flow

Use this flow if you want to actively edit the generated `.do` files before loading and routing them with Allegro PCB Router running in foreground mode.

The following commands are considered part of the High-speed Power User Flow. Click on a command for specific usage details described in the *Allegro PCB and Package Physical Layout Command Reference*.

Command	Use to ...
<code>specctra_out</code>	export data from your design database and generate routing files for use in the router.
<code>specctra</code>	invoke the router in foreground mode in preparation to load routing files and autoroute the design.
<code>specctra_in</code>	translate and import data from a router session (<code>.ses</code>) file to update your design.

Flow Procedure

1. Choose *File > Export > Router* (`specctra_out`) .

When you run this command, the following actions occur:

- The layout editor writes a design, rules, and forget file from the current database.
- The Export to Auto-router dialog box opens.

2. Click *Run* to export the design file to Allegro PCB Router.
3. When translation of the design file is complete, close the dialog box.
4. Open a text editor to review and edit the `rules.do` file. Cadence recommends that you do the following when editing any `.do` files:
 - a. Copy the generated `rules.do` file to a different file name.
 - b. Edit the renamed file as needed.
5. Choose *Route > Route Editor* (`specctra`) to launch Allegro PCB Router.
6. Load the forget file and the renamed `.do` files into Allegro PCB Router and perform an initial route of the design.
7. If the initial route is completed to satisfaction, load the forget file and the original `.do` file(s).
8. Issue the `check` command to verify any design rule violations.
9. If you are satisfied with the results, write out a session file, and load the original files back into the layout editor .
 - a. Choose *File – Import – Router* (`specctra_in`).
The Import from Auto-Router dialog box opens.
 - b. Enter the name of the session file.
 - c. Click *Run* to import the file into the layout editor .

Autorouting Parameters

Setting Parameters in the Mainstream Flow

Prior to initiating autorouting using the mainstream flow, you have the opportunity to set routing parameters within the layout editor using the Automatic Router Parameters dialog box shown.

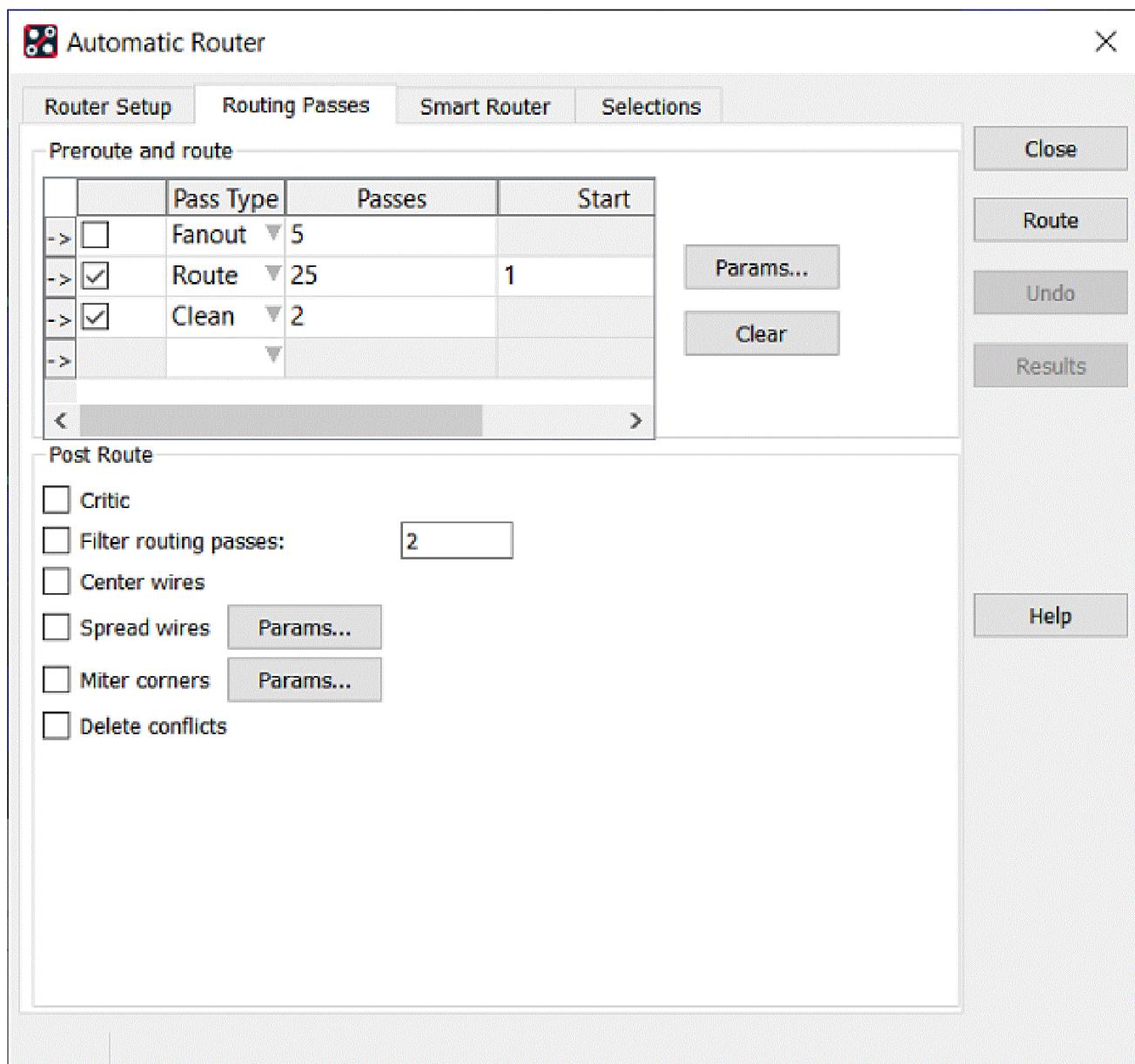
You can set the following parameters:

Parameter Type	Specifies ...
Fanout	pin and via sharing, layer depth, escape direction, and a temporary grid.
Blind / Buried Via Depth	both direction and depth of the routing for blind and buried vias.
Pin Types	all pin types.
Bus Routing	how component pins sharing the same, or nearly the same, x or y coordinate are routed.
Seed Vias	when single connections break into two shorter connections by adding a via.
Testpoint	how testpoints are assigned to signal nets.
Miter Corners	how and when 90-degree wire corners are changed to 45 degrees for wires exiting pins and vias.
Spread Wires	how extra space is added between wires and pins to improve design manufacturability without moving or adding vias.
Elongation	how etch length is increased to adhere to timing rules.

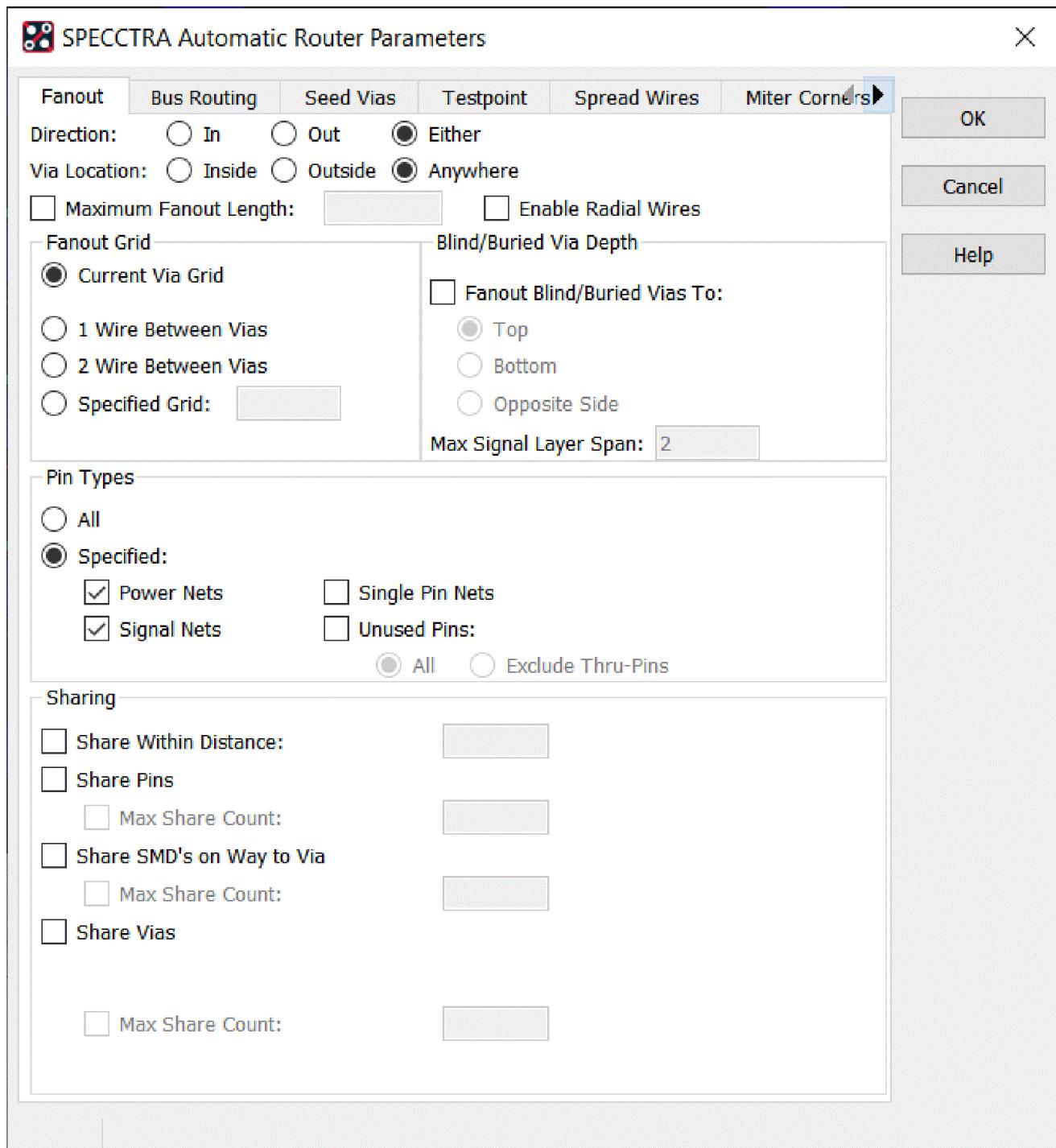
The Automatic Router Parameters dialog box can be accessed within the layout editor by:

- Clicking *Params* on the *Routing Passes* tab of the Automatic Router dialog box (choose *Route – Route Automatic*).

Accessing the Automatic Router Parameters Dialog Box



Automatic Router Parameters Dialog Box



Setting Parameters in the High-Speed or High-Speed Power User Flows

When using either the High-speed or High-speed Power User flows, Allegro PCB Router runs in foreground mode. Therefore, autorouting parameters must be set directly within the router environment or by editing the routing .do files before you commence autorouting.

Related Topics

- [Automatic Router Dialog Box](#)
- [Autoroute Menu](#)

Autorouting File Examples

This section provides examples of the rules and forget files generated when you invoke Allegro PCB Router from the layout editor .

Sample Rules File

```
# start of do file
#####
##

# CLEARANCE RULES
#
#####
##

# rule assignments for design clearances
#
#####
##

rule PCB (width 6)

rule PCB (clearance 5.50 (type wire_wire))

rule PCB (clearance 9 (type wire_smd))

rule PCB (clearance 9 (type wire_pin))

rule PCB (clearance 5.50 (type wire_via))
```

```
rule PCB (clearance 9 (type smd_smd))
rule PCB (clearance 20 (type smd_pin))
rule PCB (clearance 11.50 (type smd_via))
rule PCB (clearance 14 (type pin_pin))
rule PCB (clearance 9 (type pin_via))
rule PCB (clearance 5.50 (type via_via))
rule PCB (clearance 5 (type test_test))
rule PCB (clearance 5 (type test_wire))
rule PCB (clearance 5 (type test_smd))
rule PCB (clearance 5 (type test_pin))
rule PCB (clearance 5 (type test_via))
rule PCB (clearance 5 (type buried_via_gap))
rule PCB (clearance 0 (type area_wire))
rule PCB (clearance 0 (type area_smd))
rule PCB (clearance 0 (type area_area))
rule PCB (clearance 0 (type area_pin))
rule PCB (clearance 0 (type area_via))
rule PCB (clearance 0 (type area_test))
rule PCB (clearance 0.12 (type bbvia_wire))
rule PCB (clearance 0.12 (type bbvia_bbvia))
rule PCB (clearance 0.12 (type bbvia_smdpin))
#####
##

# rule assignments for layer clearances
#
#####
##

rule layer BOTTOM (clearance 9 (type wire_via))
rule layer BOTTOM (clearance 20 (type smd_smd))
rule layer BOTTOM (clearance 20 (type smd_via))
```

```
rule layer BOTTOM (clearance 14 (type pin_via))
rule layer BOTTOM (clearance 14 (type via_via))
#####
## WIRING RULES
#
#####
## rule assignments for pcb wiring
#
#####
## rule assignments for layer wiring
#
#####
## rule assignments for net wiring
#
#####
## TIMING RULES
#
#####
##
```

```
# rule assignments for layer timing
#
#####
##rule layer TOP (restricted_layer_length_factor 1)
rule layer BOTTOM (restricted_layer_length_factor 1)
#####
##
# Shielding RULES
#
#####
###
#####
###
# NOISE RULES
#
#####
###
#####
###
# rule assignments for net noise
#
#####
###
# end of do file
```

Sample Forget File

```
#####
###
# FORGET PAIR DEFINITIONS
#
#####
###
forget pair (nets A3 A4)
```


Allegro Integrated Analysis and Checking

The Sigrity™ technology driven high-speed analysis and checking environment provides analysis and checking capability within the Allegro® PCB Editor framework. Workflows for the following six analyses types are available from PCB Editor:

- Virtual Proto
- Design Setup
- Impedance
- Coupling
- Crosstalk
- Return Path
- Reflection
- IR Drop
- Power Inductance
- Topology Extraction
- Via Wizard
- Interconnect Model Extraction

The workflows are available from Allegro X PCB Editor and Allegro X Advanced Package Designer when specific licenses are selected. The workflows that you see will depend on the license you select. Virtual Prototyping, Return Path, Power Inductance, and Topology Extraction are not supported in Allegro X Advanced Package Designer.

 This section, along with the section on Vision Manager, describes only the listed flows; for the Advanced ERC capabilities, see the Constraint Manager documents.

The easy-to-use workflows detect impedance discontinuities of routed signals and excessive coupling between routed signals, without needing any models. Detection of problems and identification of issues that can be resolved by design engineers early in the cycle, improves productivity and reduces the turnaround time.

The Analysis Workflows pane (*Analyze – Workflow Manager*) guides you through setting up, running, and viewing results for the different analyses. The options in the workflow pane change depending on the analysis selected.

ⓘ The environment variable `SIGRITY_EDA_DIR` must point to an installation of Sigrity™ release 2022.1, Hotfix 005 (ISR 5) to run many of the steps in this document.

ⓘ To simulate unrouted nets, set the `IDA_RATS_ENABLE` environment variable before starting PCB Editor.

Performing Virtual Prototyping

Use Virtual Proto Workflow to create quick prototypes for analysis tasks.

1. Choose *Analyze – Workflow Manager*.
2. Choose *Virtual Proto Workflow*.

The following tasks are listed for the flow:

- Define Cross-section
- Create Board Outline
- Select Components
- Assign Rooms
- Place Components
- Create Nets
- Create Planes
- Set Voltage
- Route Nets

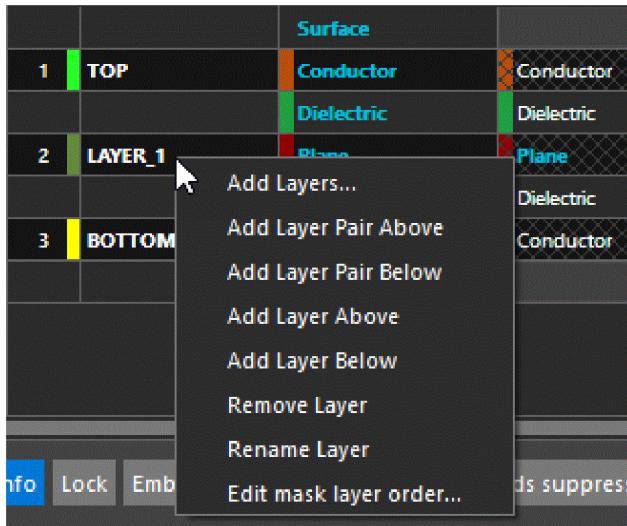
Selecting a task runs the required command and brings up necessary interfaces to accomplish the task.

You need not run all tasks in the given sequence. For example, you can select components again after setting voltage.

3. Click *Define Cross-section* to open Cross-section Editor.

The Cross-section Editor window lists the cross-sections available for the design. You can use Cross-section Editor to add or remove cross-section layers and to change material, layer type, layer name, layer thickness and so on.

Right-click in the layer table to choose an available command from the options.



For more information see the [xsection](#) command in *Allegro PCB and Package Physical Layout Command Reference*.

4. Click *OK* to close Cross-section Editor.

5. Click *Create Board Outline* to open the Design Outline dialog box.

You can create, edit, delete, or move a board outline using the Design Outline dialog box. You can also create a new segment within an existing segment of an outline.

You can create a new outline by either drawing a rectangle or a polygon or by placing a rectangle. To draw a rectangle or a polygon you must click to specify the corners. To place a rectangle, specify the width and height and then click to place the rectangle.

For more information see the [board outline](#) command in *Allegro PCB and Package Physical Layout Command Reference*.

6. Click *OK* to close Design Outline.

7. Click *Select Components* to open the Parts List dialog box.

Use the browsers in the Browsers group to add parts to the list. For example, click Physical Devices for the device files which are used to define the logical part information in a third-party netlist. Scroll down and select *pga68.txt*. Under *Part Modification Area*, in the RefDes field type U1-2 and then click Add U1 and U2. Similarly, add other devices of your choice. This will add the devices to the database. You need to place the devices.

For more information see the [partlogic](#) command in *Allegro PCB and Package Physical Layout Command Reference*.

8. Click *OK* to close the Parts List dialog box.

9. Click *Assign Rooms* to open the Room Assignment dialog box.

Choose groupings to assign to rooms and to assign components to any single group.

For more information see the [component assign](#) command in *Allegro PCB and Package Physical Layout Command Reference*.

10. Click *OK*.
11. Click *Place Components* to open the Placement dialog box.
The components that you had selected using the Parts List dialog box are listed by RefDes, by default, in the Placement dialog box. Select one or more or all listed components and click within the outline on the canvas to place the components.
You can move around the components after placing them.
For more information see the [place manual](#) command in *Allegro PCB and Package Physical Layout Command Reference*.
12. Click *Close* to close the Placement dialog box.
13. Click *Create Nets* to open the Edit Nets dialog box.
Use Edit Nets to create nets, add pins to a net, delete nets, modify nets, or rename nets.
For more information see the [edit nets](#) command in *Allegro PCB and Package Physical Layout Command Reference*.
14. Click *OK* to close Edit Nets.
15. Click *Create Planes* to open the Plane Outline dialog box.
Use the Plane Outline dialog box for creating a new plane outline or modifying, moving, or deleting an existing outline.
You can select a layer and specify a plane voltage level and assign a net to the outline shape.
You can choose to draw a rectangle or polygon for the outline or place a rectangle with specified width and height. You can also choose to copy the outline from a specified plane or copy the board outline.
For more information see the [board plane](#) command in *Allegro PCB and Package Physical Layout Command Reference*.
16. Click *OK* to close Plane Outline.
17. Click *Set Voltage* to open the Identify DC Nets dialog box.
DC Net assignment is required for signal integrity analysis, a simplified ratsnest display, and performance improvements on large pin count nets. In addition, a DC Net provides a stop point when generating XNets topology preventing a large number of nets from being combined into one XNet.
For more information see the [identify nets](#) command in *Allegro PCB and Package Physical Layout Command Reference*.

18. Click *OK* to close Identify DC Nets.
19. Click *Route Nets* to enable the Etch Edit application mode.

Perform etch-editing tasks such as adding and sliding connections, delay tuning, and smoothing cline or cline segment angles in the Etch Edit application mode.

Start routing by selecting a pin and moving the cursor. A trace is attached to the cursor. You can choose various commands by right-clicking. For example, you can choose to add a via or change layers. To complete the routing, make your way to the other pin in the net and select the pin.

For more information see the [etchedit](#) and [add connect](#) commands in *Allegro PCB and Package Physical Layout Command Reference*.

Using the Design Set Up Workflow

Use the Design Setup workflow to conveniently specify different design level settings from one cockpit. The settings you specify will be honored across different workflows and ensure that you are using consistent settings.

You can specify the following settings by clicking the tasks listed when you select the Design Setup workflow:

- Define Materials
- Define Cross-section
- Identify DC Nets
- Set up Components
- Set up Xnets
- Set up XNets
- Assign Diff Pairs

Refresh the design after making the setting changes by clicking *Refresh Data*. A red cross mark is displayed if a refresh is required.

 Click *Refresh Design Data* after adding or deleting properties for the change to be retained and visible in filters.

You can also perform Component Model Setup and Analysis Setup.

Click *Set up Padstack Plating Parameters* to specify padstack plating by selecting the plating material, which is copper by default, and choosing a method to determine plating thickness. By default, the thickness is defined entering a value, which is 1 MIL by default. You can also choose to specify *Filled with plating* or *Calculate plating thickness from drill and finished hole sizes in padstack*.

Related Topics

- [Setting Up Component Models](#)

Performing Impedance Analysis

The impedance presented here is the characteristic impedance of a transmission line. At the frequencies of interest for most PCBs, this is $Z_0 = \sqrt{L/C}$ which does not depend on frequency. The L and C values are computed from the 2D cross-sectional geometry of the trace and reference. At lower frequencies, where the characteristic impedance does depend on frequency, matching impedances is usually not a concern because reflections will usually be minimal.

1. Choose *Analyze – Workflow Manager*.
2. Choose *Impedance Workflow*.
3. Choose either *Net Based* or *Directed Group* from *Analysis Modes*.
Net Based is chosen by default.
4. Select nets or specify directed groups based on the analysis mode.
5. To include coplanar data in the simulation results, click *Analysis Options* and select *Detect and model the coplanar traces*.
6. Click *Start Analysis*.

Progress of analysis is shown in the Analysis section of the Analysis Workflows window. When analysis is complete, a green check will appear for the *Start Analysis* task. If a previous analysis is stored, a prompt appears for overwriting the old analysis.

7. Click *Save Analysis Results* to save the results. Specify a name and location for the results file.

The extension for the result file is `.impida` for impedance analysis.

8. Specify *View Modes*. Choose:

- *Net Based* to view results for all segments of selected nets. Available for all analysis results.
- *Directed Group* to view results for segments defined in selected directed groups. Available only if analysis results contain directed group data.
- *Single Ended* to view single-ended analysis results for both single-ended and differential pair nets.
- *Diff Pair* to view differential pair analysis results for differential pair nets and single-ended results for single-ended nets.
Differential analysis results are reflected for coupled segments of differential pairs.

- [Selecting Nets](#)

- Specifying Directed Groups
- Impedance Table
- Impedance Vision

Impedance Vision

Clicking *Impedance Vision* displays color-coded impedance segments in the canvas.

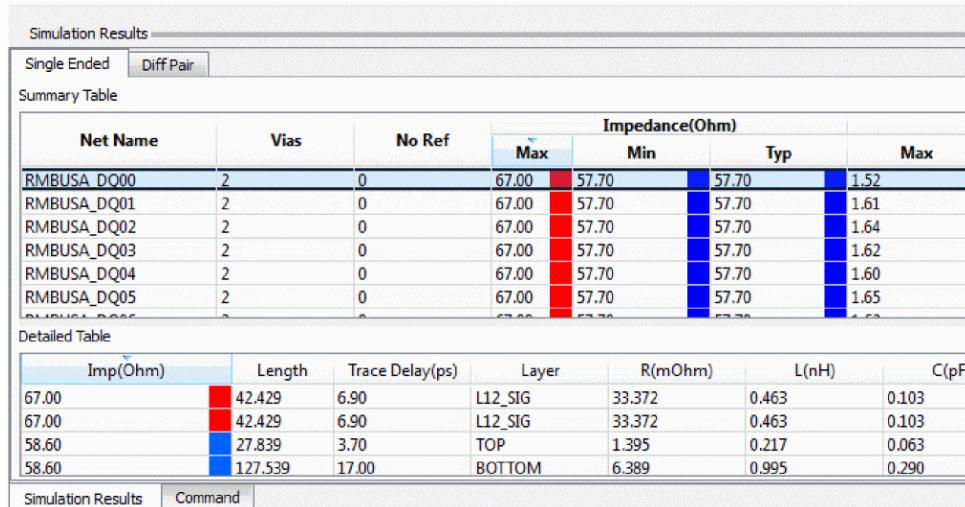
The color coding is based on the segment or sub-segment impedance value as mapped into the impedance gradient, which is the continuous transition of colors from the minimum to maximum range of impedance values obtained by analysis. Any nets part of the analysis results which have segments not associated with impedance analysis data will be displayed in the default color based on layer/net color assignments. Any nets which are not part of the analysis are dimmed in the default color.

The Single Ended view mode setting displays all Singled Ended impedance result for the selected nets; both single ended nets and differential pair nets of the results use single ended impedance data to color segments and for data tips.

The Diffpair view mode setting displays Single Ended impedance data for all single ended nets and all Diffpair impedance data for all differential pair nets.

Impedance Table

Summary Table lists summary impedance results for each net selected for analysis. Detailed Table lists the impedance results for each segment on the net selected in Summary Table.



Simulation Results

Single Ended Diff Pair

Summary Table

Net Name	Vias	No Ref	Impedance(Ohm)			
			Max	Min	Typ	Max
RMBUSA_DQ00	2	0	67.00	57.70	57.70	1.52
RMBUSA_DQ01	2	0	67.00	57.70	57.70	1.61
RMBUSA_DQ02	2	0	67.00	57.70	57.70	1.64
RMBUSA_DQ03	2	0	67.00	57.70	57.70	1.62
RMBUSA_DQ04	2	0	67.00	57.70	57.70	1.60
RMBUSA_DQ05	2	0	67.00	57.70	57.70	1.65
RMBUSA_DQ06	2	0	67.00	57.70	57.70	1.63

Detailed Table

Imp(Ohm)	Length	Trace Delay(ps)	Layer	R(mOhm)	L(nH)	C(pF)
67.00	42.429	6.90	L12_SIG	33.372	0.463	0.103
67.00	42.429	6.90	L12_SIG	33.372	0.463	0.103
58.60	27.839	3.70	TOP	1.395	0.217	0.063
58.60	127.539	17.00	BOTTOM	6.389	0.995	0.290

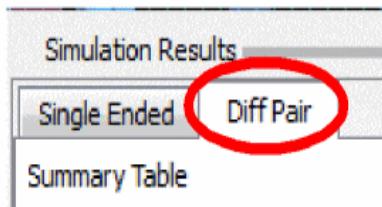
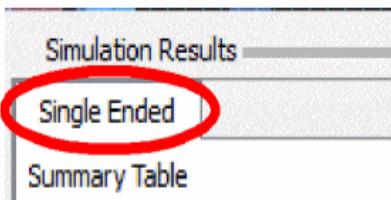
Simulation Results Command

To sort data in the table, click any column header.

Select rows in the Detailed Table section to zoom/center the canvas on the impedance variance for investigation.

If view mode is Single Ended, a Single Ended tab appears at the top of the summary table and all net results appear in the summary table.

If view mode is Diffpair, both the *Single Ended* and *Diffpair* tabs appear at the top of the summary table. Select the Single Ended tab to view single ended nets or select the Diffpair tab to view differential pair nets.



Performing Coupling Analysis

Coupling is not computed at a single frequency. Since PCB traces carry bit patterns that are made of pulses containing many frequencies, the coupling coefficient result takes into account a rise time in picoseconds. The maximum frequency relevant to the trace is inversely proportional to rise time parameter.

1. Choose *Analyze – Workflow Manager*.
2. Choose *Coupling Workflow*.
3. Choose either *Net Based* or *Directed Group* from *Analysis Modes*.
Net Based is chosen by default.
4. Select nets or specify directed groups based on the analysis mode.
5. Click *Analysis Options*.
The Coupling Analysis Parameters dialog appears.
Specify the following parameters:
 - *Detect and model the coplanar traces*: Choose to include coplanar data in the simulation results.
 - *Coupling*: Specify the minimum net level coupling coefficient threshold in percentage. The default value is 2%.
 - *Rise Time*: Specify the rise time for minimum coupled length in μs (picoseconds). The default is $50\mu\text{s}$.
 - *GeoWindow*: Set a geometry window value in design units for aggressor inclusion for the selected victim nets. All net segments within the specified distance from the selected net across layers will be considered as aggressors. This is optional for coupling analysis and allows for additional potential aggressors to be found based on the window.
6. Click *Start Analysis*.
Progress of analysis is shown in the Analysis section of the Analysis Workflows window.
When analysis is complete, a green check will appear for the *Start Analysis* task. If a previous analysis is stored, a prompt appears for overwriting the old analysis.
7. Click *Save Analysis Results* to save the results. Specify a name and location for the results file.
The extension for the result file is `.cplida` for coupling analysis.

8. Specify *View Modes*. Choose:

- *Net Based* to view results for all segments of selected nets. Available for all analysis results.
- *Directed Group* to view results for segments defined in selected directed groups. Available only if analysis results contain directed group data.
- *Worst case* to view the maximum coupling coefficient for all segments of selected nets.
- *Victim* to view aggressor segments for a selected victim net.

9. Click *Coupling Table* or *Coupling Vision* to view results.

- [Selecting Nets](#)
- [Specifying Directed Groups](#)
- [Coupling Table](#)
- [Coupling Vision](#)

Coupling Vision

Clicking *Coupling Vision* displays color-coded coupling segments in the canvas.

The color coding is based on the segment or sub-segment coupling coefficient value as mapped into the coupling color gradient, which is the continuous transition of colors from the minimum to maximum range of coupling values obtained by analysis. Nets which are not part of the analysis are dimmed in the default color. Even for a net that is part of the analysis, segments not associated with the analysis will be displayed in the default color based on layer/net color assignments.

The Worst Case view mode setting highlights segments on victim nets. The color coding is based on the maximum coupling coefficient for any aggressor segment to that victim segment. Victim segments not having a coupling coefficient are displayed dimmed in the default color.

The Victim view mode setting highlights aggressor segments and sets the color to active victim net/segment. The color coding is based on the coupling coefficient for each aggressor segment to the cross-probed net/segment. The victim net is displayed in the default color. Aggressor segments not having a coupling coefficient applied to the victim net/segment are displayed dimmed in the default color.

Coupling Table

In the Coupling Table, Worst Case and Victim modes differ in the way aggressors are color coded in the table.

Simulation Results						
Worst Case Mode						
Summary Table						
Net Name		Max Coupling		% Length with Coupling Coef		Total Coupling Index (mils-%)
Aggressor Net Name	Coeff(%)	Length(%)	> 5%	2%~5%		
P0_MEM_MA0_CLK_H<0>	P0_MEM_MA_ADD<0>	12.50	3.34	3.34	11.00	3880.13
P0_MEM_MA0_CLK_H<1>	P0_MEM_MA_ADD<0>	0.90	34.36	0.00	0.00	607.52
P0_MEM_MA0_CLK_L<0>	P0_MEM_MA_RAS_L	3.30	0.82	0.00	5.60	1783.28
P0_MEM_MA0_CLK_L<1>	P0_MEM_MA_ADD<0>	3.20	34.73	0.00	14.58	2206.02
P0_MEM_MA0_CS_L<0>	P0_MEM_MA_WE_L	11.10	0.64	6.16	37.61	4560.62
P0_MEM_MA0_CS_L<1>	P0_MEM_MA_ODT<0>	3.80	1.01	0.00	49.94	2998.60

Detailed Table						
Victim Trace Ref	Aggressor Net	Aggressor Trace Ref	Coupling Coef	Length	Layer	Victim
▷ (1) Trace27221::P0_MEM_MA0_CLK_H<0>	P0_MEM_MA_RAS_L	Trace25418::P0_MEM_MA_RAS_L	0.68	8.482	L4_INNER	(2758.055 :)
▷ (3) Trace27221::P0_MEM_MA0_CLK_H<0>	P0_MEM_MA_ADD<0>	Trace27068::P0_MEM_MA_ADD<0>	3.00	210.469	L4_INNER	(2766.537 :)
▷ (2) Trace27221::P0_MEM_MA0_CLK_H<0>	P0_MEM_MA_ADD<0>	Trace27068::P0_MEM_MA_ADD<0>	3.00	21.708	L4_INNER	(2977.006 :)

Summary Table lists summary coupling results for each net selected for analysis. Detailed Table lists the coupling results for each segment on the net selected in Summary Table.

Detailed Table provides a hierarchical view where each victim segment and all related aggressors are listed in one block of information as contiguous rows. Within the victim segment block, the aggressors are listed in the order of the highest coupling coefficient. If coupling coefficients are same, length is considered for ordering.

The first row in each block is the victim summary line and displays the total aggressor count for the block in the first column. If the block hierarchy is contracted, the aggressor with the highest coupling coefficient and longest length is displayed.

In the Worst Case mode, the coupling coefficient (*Coupling Coef*) column is color coded on the victim summary line and the maximum aggressor within the block. The coupling coefficient column for non-maximum aggressors is colored gray.

Detailed Table				
Victim Trace Ref	Aggressor Net	Aggressor Trace Ref	Coupling Coef	Length
▷ (1) Trace1004::CLK32K_GATE_LPOIN	GPS_IFVALID	Trace922::GPS_IFVALID	3.33	0.0823
▷ (2) Trace1005::CLK32K_GATE_LPOIN	GPS_IFVALID	Trace965::GPS_IFVALID	7.44	0.0336
Trace1005::CLK32K_GATE_LPOIN	GPS_IFVALID	Trace965::GPS_IFVALID	7.44	0.0336
Trace1005::CLK32K_GATE_LPOIN	UIM_DATA	Trace901::UIM_DATA	1.58	0.0336
▷ (2) Trace1005::CLK32K_GATE_LPOIN	GPS_IFVALID	Trace964::GPS_IFVALID	7.47	0.1662

In the Victim mode, the coupling coefficient column is color coded for all aggressors but the victim summary line is gray. On the victim summary line, the victim segment column is color coded with the default color as it appears on the canvas.

Detailed Table

Victim Trace Ref	Aggressor Net	Aggressor Trace Ref	Coupling Coef	Length
(2) Trace982::CLK32K_GATE_LPOIN	GPS_IFVALID	Trace939::GPS_IFVALID	5.07	0.7301
Trace982::CLK32K_GATE_LPOIN	GPS_IFVALID	Trace939::GPS_IFVALID	5.07	0.7301
Trace982::CLK32K_GATE_LPOIN	VREG_UIM	Trace469::VREG_UIM	1.60	0.7301

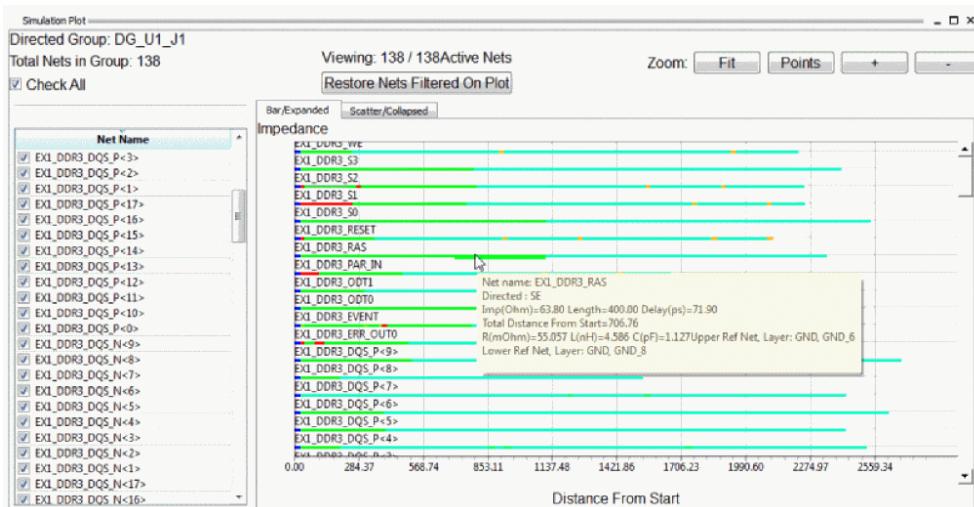
Impedance and Coupling Plots

Impedance Plot or Coupling Plot is available only for directed groups and shows results as bar (expanded) and scatter (collapsed) plots using the distance from start component on the x-axis as determined in the Directed Groups definition.

Clicking Impedance Plot or Coupling Plot opens the dockable Simulation Plot pane, which has two tabs, *Bar/Expanded* and *Scatter/Collapsed*.

The Bar/Expanded tab can be filtered using the options on the right while the Scatter/Collapsed tab offers a global view and can be used to quickly identify outliers. Both tabs support data tips.

A net selection panel on the left lists all the nets in the active directed groups. All nets are selected by default. You can choose to select only specific nets.



Performing Crosstalk Analysis

1. Choose *Analyze – Workflow Manager*.

2. Choose *Crosstalk Workflow*.

3. Set up component models.

a. Set up default models

b. Manages libraries

Only IBIS buffer models are considered for crosstalk.

4. Select nets to be simulated.

5. Click *Analysis Options*.

The Crosstalk Analysis Parameters dialog appears.

Specify the following parameters:

- *Include Each Neighbor*: Run for every neighbor based on All Neighbor results and filtered by the value specified in *Worst Case Each Victim Highest*.
- *Worst Case Each Victim Highest*: Specify the filter value in percentage that is to be applied to results of All Neighbor run.
- *Coupling*: Specify the minimum net level coupling coefficient threshold in percentage. The default value is 2%.
- *Rise Time*: Specify the rise time for minimum coupled length in μs (picoseconds). The default is $50\mu\text{s}$.
- *GeoWindow*: Set a geometry window value in design units for aggressor inclusion for the selected victim nets. All net segments within the specified distance from the selected net across layers will be considered as aggressors. This is required in the crosstalk flow.

A segment on a layer other than victim segment will only be considered if the GeoWindow is greater than the layer distance span.

- *Corner*: Specify the corner setting as *Typical* (default), *Fast*, or *Slow*.

If buffer models do not have corner data for Fast and Slow corners, the results will be produced for Typical setting.

6. Click *Start Analysis*.

Progress of analysis is shown in the Analysis section of the Analysis Workflows window. When analysis is complete, a green check will appear for the *Start Analysis* task. If a previous analysis is stored, a prompt appears for overwriting the old analysis.

7. Click *Save Analysis Results* to save the results. Specify a name and location for the results file.
The extension for the result file is `.xtalkida` for crosstalk analysis.
8. Click *Crosstalk Table* or *Crosstalk Vision* to view results.

Related Topics

- [Setting Up Component Models](#)
- [Selecting Nets](#)
- [Crosstalk Table](#)
- [Crosstalk Vision](#)

Crosstalk Vision

Clicking *Crosstalk Vision* displays net-level, color-coded crosstalk segments in the canvas.

Victim net is displayed in default color with coupled sections color-coded based on coefficient. Click in the summary table to see simulation information. Double-click to zoom or pan.

Hover over a net to see datatips for net-level crosstalk.

Crosstalk Table

The table for crosstalk analysis, *Simulation Table*, has two tabs, *All Neighbor* and *Each Neighbor*.

The *All Neighbor* tab shows four simulation results for all victims; listed hierarchically with the summary row being the simulation with the highest crosstalk on that victim. The types of simulation results for All Neighbor are unique combinations of high-state (*HS*) and low-state (*LS*) with even (*EVEN*) or odd (*ODD*) switch.

- ⚠ In a high state (*HS*), a victim is held high whereas a low state (*LS*) means the victim is held low. Even (*EVEN*) is when an aggressor is switching towards and odd (*ODD*) when the aggressor is switching away from a victim state.

Victim net/Xnet	Result/Waveform L	Victim Receiver	Simulation Type
U11_U12_H	402.8 mV	CDS_IN_GEN_2...	ALL; HS; EVEN; ...
	402.8 mV	CDS_IN_GEN_2...	ALL; HS; EVEN; ...
	359.9 mV	CDS_IN_GEN_2...	ALL; HS; ODD; ...
	400.2 mV	CDS_IN_GEN_2...	ALL; LS; EVEN; ...
	361.5 mV	CDS_IN_GEN_2...	ALL; LS; ODD; T...
U11_U12_G	223.0 mV	CDS_IN_GEN_2...	ALL; HS; EVEN; ...
	235.5 mV	CDS_IN_GEN_2...	ALL; HS; EVEN; ...
	201.5 mV	CDS_IN_GEN_2...	ALL; LS; ODD; T...
	314.9 mV	CDS_IN_GEN_2...	ALL; HS; EVEN; ...
	139.2 mV	CDS_IN_GEN_2...	ALL; HS; ODD; ...
DP_ED	194.4 mV	CDS_IN_GEN_2...	ALL; LS; ODD; T...

The *Each Neighbor* tab shows simulation results for each victim-aggressor; listed hierarchically with the summary row being the simulation with the highest crosstalk on victim.

- ⚠ Differential pairs are listed as a single entry in the Simulation table; for example, DP_ED in the following image. The value shown is the maximum crosstalk value for each individual mate.

Victim net/Xnet	Result/Waveform Link	Victim Receiver	Simulation Type
DP_ED	194.4 mV	CDS_IN_GEN_2P5V_10PF_50...	ALL; LS; ODD; TYP
	134.4 mV	CDS_IN_GEN_2P5V_10PF_50...	ALL; HS; EVEN; TYP
	190.9 mV	CDS_IN_GEN_2P5V_10PF_50...	ALL; HS; ODD; TYP
	132.7 mV	CDS_IN_GEN_2P5V_10PF_50...	ALL; LS; EVEN; TYP
	194.4 mV	CDS_IN_GEN_2P5V_10PF_50...	ALL; LS; ODD; TYP

Select a victim to see simulation coupled data in Detail Couple Table.

Simulation Table

All Neighbor	Each Neighbor			
Summary Table				
Victim net/Xnet	Aggressor net/Xnet	Result/Waveform Link	Victim Receiver	Simulation Type
U11_U12_B	U11_U12_H	234.7 mV	CDS_IN_GEN_2...	EACH; HS; EVE...
U11_U12_F	U11_U12_G	205.3 mV	CDS_IN_GEN_2...	EACH; HS; EVE...
U11_U12_H	U11_U12_B	232.4 mV	CDS_IN_GEN_2...	EACH; HS; EVE...

Detail Table

Victim Trace Ref	Aggressor Net	Aggressor Trace Ref	Coupling Coef	Length	Layer	Coupling Index	Victim Segment	Aggressor Segment
Trace11::U11_U...	U11_U12_B	Trace140::U11_...	7.28	4.24	L2	30.87	(733.56 1571.44)...	(724.00 1581.00)...
Trace11::U11_U...	U11_U12_B	Trace138::U11_...	16.15	4.39	L2	70.88	(740.46 1578.34)...	(734.80 1584.00)...
Trace11::U11_U...	U11_U12_B	Trace138::U11_...	16.40	7.22	L2	118.44	(743.56 1581.44)...	(737.90 1587.10)...
Trace12::U11_U...	U11_U12_B	Trace141::U11_...	7.96	13.00	L2	103.47	(711.00 1568.00)...	(711.00 1581.00)...
Trace5::U11_U1...	U11_U12_B	Trace130::U11_...	7.85	611.00	L2	4795.75	(1355.00 523.00)...	(1371.00 523.00)...
Trace5::U11_U1...	U11_U12_B	Trace130::U11_...	17.04	30.00	L2	511.06	(1355.00 1134.00)...	(1371.00 1134.00)...
Trace5::U11_U1...	U11_U12_B	Trace130::U11_...	7.85	80.63	L2	632.87	(1355.00 1164.00)...	(1371.00 1164.00)...
Trace6::U11_U1...	U11_U12_B	Trace131::U11_...	2.68	9.45	L2	25.33	(983.63 1616.00)...	(1000.81 1633.1...)...
Trace6::U11_U1...	U11_U12_B	Trace131::U11_...	2.62	480.83	L2	1257.43	(990.31 1609.31)...	(1007.50 1626.5...)...
Trace6::U11_U1...	U11_U12_B	Trace131::U11_...	2.68	33.23	L2	89.06	(1330.31 1269.3...)...	(1347.50 1286.5...)...
Trace7::U11_U1...	U11_U12_B	Trace132::U11_...	5.19	90.01	L2	467.50	(839.00 1616.00)...	(839.00 1635.00)...
Trace7::U11_U1...	U11_U12_B	Trace132::U11_...	5.19	31.99	L2	166.15	(929.01 1616.00)...	(929.01 1635.00)...
Trace7::U11_U1...	U11_U12_B	Trace132::U11_...	5.19	18.00	L2	93.49	(961.00 1616.00)...	(961.00 1635.00)...
Trace7::U11_U1...	U11_U12_B	Trace132::U11_...	5.27	4.63	L2	24.39	(979.00 1616.00)...	(979.00 1635.00)...
Trace8::U11_U1...	U11_U12_B	Trace135::U11_...	4.11	5.90	L2	24.27	(779.00 1616.00)...	(779.00 1634.00)...
Trace8::U11_U1...	U11_U12_B	Trace133::U11_...	3.62	54.10	L2	196.05	(784.90 1616.00)...	(785.80 1635.00)...
Trace9::U11_U1...	U11_U12_B	Trace136::U11_...	10.04	16.74	L2	168.01	(751.00 1592.00)...	(743.00 1600.00)...
Trace9::U11_U1...	U11_U12_B	Trace136::U11_...	10.02	14.59	L2	146.25	(762.84 1603.84)...	(754.84 1611.84)...

In both the tabs, you can see details about the simulation results, save results to a comma-separated value (CSV) file, and see the waveform view.

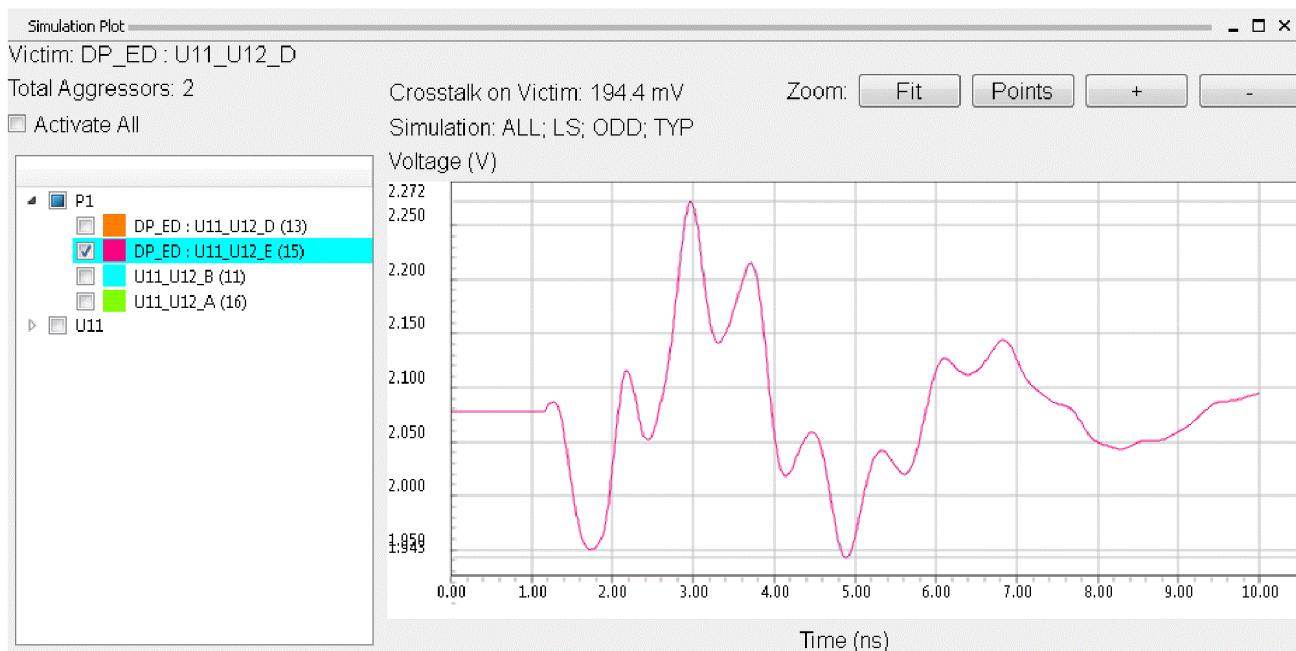
Right-click a victim and choose:

- *Show Details* to see details about the simulation, such as victim net and driver, simulation type, and aggressor nets.

A	B	C	D
1	Victim net/Xnet	Result/Waveform Link	Victim Receiver
2	U11_U12_H	402.8 mV	CDS_IN_GEN_2P5V_10PF_NOTERM P1 10
3	U11_U12_H	359.9 mV	CDS_IN_GEN_2P5V_10PF_NOTERM P1 10
4	U11_U12_H	400.2 mV	CDS_IN_GEN_2P5V_10PF_NOTERM P1 10
5	U11_U12_H	361.5 mV	CDS_IN_GEN_2P5V_10PF_NOTERM P1 10
6	U11_U12_G	223.0 mV	CDS_IN_GEN_2P5V_10PF_NOTERM P1 7
7	U11_U12_G	215.4 mV	CDS_IN_GEN_2P5V_10PF_NOTERM P1 7
8	U11_U12_G	221.0 mV	CDS_IN_GEN_2P5V_10PF_NOTERM P1 7

To see the waveform view for a result row, click the *Result/Waveform Link* column of that row. In the waveform view, the highest crosstalk on receiver is auto-selected and highlighted.

⚠ By default, only the victim waveform is displayed. Click in the tree view to display aggressor waveforms



Performing Return Path Analysis

1. Choose *Analyze – Workflow Manager*.
2. Choose *Return Path Workflow*.
3. Specify the reference net.
4. Specify the directed groups to be simulated.
5. Click *Start Analysis*.

Progress of analysis is shown in the Analysis section of the Analysis Workflows window.

When analysis is complete, a green check will appear for the *Start Analysis* task. If a previous analysis is stored, a prompt appears for overwriting the old analysis.

6. Click *Save Analysis Results* to save the results. Specify a name and location for the results file.
The extension for the result file is `.rpida` for return path analysis.
7. Click *Return Path Table* to view results.
Simulation Table appears listing the results in a tabular format.
8. Click *Start Simulation* in the *Vision* column or right-click a row and choose *Start Simulation* to view color-coded results in the canvas.

Related Topics

- [Specifying Directed Groups](#)
- [Return Path Table](#)
- [Return Path Vision](#)

Return Path Vision

Click *Return Path Table* and then click *Start Simulation* or *Restart Simulation* in the *Vision* column to view color-coded results in the canvas for a net and layer. Select *Vision Net* and *Vision Layer* to view the results of your choice.

Return Path Table

Net/Xnet Name	Quality Factor	Return Path		Start Signal Pin	Start Return Pin	End Signal Pin	End Return Pin
		Vision	Start Simulation				
A<22>	1.237	Start Simulation	U2.2	U2.11	U15.38	U15.39	
A<23>	1.235	Start Simulation	U2.1	U2.11	U15.37	U15.39	
A<19>	1.231	Start Simulation	U2.5	U2.11	U15.43	U15.45	
A<21>	1.228	Start Simulation	U2.3	U2.11	U15.40	U15.39	
A<18>	1.226	Start Simulation	U2.6	U2.11	U15.44	U15.45	
A<20>	1.223	Start Simulation	U2.4	U2.11	U15.41	U15.39	
DDIR	1.206	Start Simulation	U2.20	U2.11	U15.24	U15.21	
DEN	1.184	Start Simulation	U2.21	U2.11	U15.48	U15.45	
DHEN_R	1.158	Start Simulation	U2.22	U2.11	U15.25	U15.28	
MW_P	1.152	Start Simulation	U2.9	U2.11	U15.3	U15.4	

The Simulation table lists the *Return Path Quality Factor* (RPQF) along with nets, vision, and start and end signal pins.

RPQF is a ratio of the actual loop inductance and the ideal loop inductance based on ideal reference plane conditions.

 A directed group that cannot be simulated because of errors is listed with the value 0.

Click in the *Vision* column start or restart a simulation and enable Return Path Vision.

Return Path Vision

Click *Return Path Table* and then click *Start Simulation* or *Restart Simulation* in the *Vision* column to view color-coded results in the canvas for a net and layer. Select Vision Net and Vision Layer to view the results of your choice.

Performing Reflection Analysis

This analysis is fully functional only if Allegro Sigrity SI or SI/PI Base Suite available. For High-Speed option, *Save Workflow Settings*, *Load Workflow Settings* and *Load Existing Analysis* are available initially, and on loading analysis, the *View Results* options are enabled.

1. Choose *Analyze – Workflow Manager*.
2. Choose *Reflection Workflow*.
3. Set up component models.
 - a. Set up default models
 - b. Manages libraries
4. Click *Set up Analysis Options*.

The Reflection Analysis Parameter Setup dialog appears, with two tabs, Simulation and Interconnect Models. This dialog has a Reflection Analysis Parameters section with various parameters. The dialog also includes an expandable section listing the selected nets.

- a. Specify the following parameters in the Simulation tab of the dialog:

The options set in the dialog will not override any existing net pulse parameters by default. To override parameters, use the *View/Apply Net Override Parameters* section of the dialog box.

- *Bit Pattern*: Specify a bit pattern or choose from the list. The default value is 0101.
- *Data Rate*: Specify the data rate in Gbps (giga-bits per second). Default value is 0.5Gbps . The *frequency* and unit interval are calculated based on the specified value. The unit interval is calculated in nano-seconds, which is 2ns for the default value.
- *Gbps Frequency*: Specify the frequency in MHz. Default value is 500MHz . The Data Rate value is calculated based on this value, as is the unit interval.
- *Offset/Delay*: Enter a time delay for the stimulus in nanoseconds to be applied before the stimulus switches. The default value is 0ns .
- *Corner*: Choose a corner for the IBIS models of the drivers and receivers. Valid values are *Typical*, *Fast*, and *Slow*. *Typical* is the default value. Choose *Fast Driver/Slow Receiver* or *Slow Driver/Fast Receiver* to test worst case scenario corners.

- b. If needed, expand *View/Apply Net Override Parameters* to view and change selected nets.

The number of selected nets and the number with existing pulse parameter properties are shown as *Selected Nets* and *Nets with Overrides*, respectively.

Differential pairs are listed as a single net starting with *DP_<pair_name>:<P-net_name>*.

Any change made in this table updates the database. The number of nets with overrides is listed at the top of the section along with the number of selected nets.

- c. In the Interconnect Models tab, specify the model type and the related parameters for unroute or partially routed nets.

You can select *T-element* or *W-element* as the model type for rats. By default, *T-element* is selected.

5. Click *Start Analysis*.

Progress of analysis is shown in the Analysis section of the Analysis Workflows window.

When analysis is complete, a green check appears for the *Start Analysis* task. If a previous analysis is stored, a prompt appears for overwriting the old analysis.

6. Click *Save Analysis Results* to save the results. Specify a name and location for the results file.

The extension for the result file is *.rfltida* for reflection analysis.

7. Click *Reflection Table* or *Reflection Vision* to view results.

Related Topics

- [Setting Up Component Models](#)
- [Reflection Table](#)
- [Reflection Vision](#)

Reflection Vision

Reflection Vision shows color-coded segments based on a color gradient within the canvas.

Hover over a segment to view data tips. The focus data color coding and data tips vary depending on the Reflection Table focus data setting. Text at the bottom of the gradient legend shows the measurement being shown in the Vision if the Table is not open.

The color is from blue for the smallest gradient to red for the largest gradient. The color code is reversed for Ring Back Margin since smaller (red) values are worse than large values (blue).

Drag the pointer at the top or bottom of the gradient on the canvas to select specific regions of the results to display in the vision. For nets with multiple driver/receiver pairs, only the portion of the net between the worst-case driver/receiver pair for the focus measurement is colored in the vision.

Reflection Table

The Simulation Table displays two voltage measurements and three timing measurements for a selected measurement of focus, which is by default *Ring Back Margin High*. Measurements are taken on the receiver side for both rising and falling edges of the stimulus. If the stimulus does not contain a rising or falling edge, the rise or fall measurements will be listed as *N/A* (not applicable).

The two voltage measurements are *Ring Back Margin* and *JEDEC Overshoot*. The timing measurements are Propagation delay, Minimum First Switch, and Maximum Final Settle.

The focus can be any one of:

- *Ring Back Margin High*: Difference between lowest signal voltage in high state and V_{ih} .
- *Ring Back Margin Low*: Difference between highest signal voltage in low state and V_{il} .
- *JEDEC Overshoot High*: Difference between high point and DC high levels.
- *JEDEC Overshoot High/Low*: Difference between low point and DC low levels.
- *Prop. Delay* (Propagation Delay): Time between beginning of receiver and driver transition.
- *Min First Switch Rise*: Time between receiver crossing V_{il} and driver beginning to rise.
- *Min First Switch Fall*: Time between receiver crossing V_{ih} and driver beginning to fall.
- *Max Settle Switch Rise*: Time between receiver's last crossing of V_{ih} before settling and driver beginning to rise.
- *Max Settle Switch Fall*: Time between receiver's last crossing of V_{il} before settling and driver beginning to fall.

The results are color coded and map to the Reflection gradient in the Reflection Vision, from blue for the smallest to red for the largest gradient. The color code is reversed for Ring Back Margin since smaller (red) values are worse than large values (blue).

The results are shown for each net. The table is hierarchical if a net has multiple drivers or receivers. Expand the net to view the results for each driver/receiver pair. The top-most driver/receiver pair is the one with the worst-case focus measurement. Therefore, the display might change the driver/receiver pair shown in the collapsed view when the focus measurement selection changes.

Results for differential pairs is measured based on the differential waveform (p-net - n-net). The net name is shown in the format `<diffpair_name>:<p-net_name>`. On expanding a diffpair, results are shown in separate rows for each driver/receiver pair. X-nets are shown as `<X-net name>`.

Right-click on the Reflection Table to select an option to export the table to a CSV file or see more details about a driver/receiver pair. You can also sort the columns in ascending or descending order.

Double-click any row in the Reflection table to cross-probe the worst-case pin pair.

Click a value in the measurement of focus column to open Reflection Waveform viewer, which shows the full waveform for a net. The viewer shows the receiver waveform by default., but the driver and any other receiver waveforms will also be available to view from a menu on the left of the waveform view. The high and low threshold values from the IBIS model are shown using dashed lines. The waveform viewer header includes the net name, driver pin, and receiver thresholds. If multiple receivers are available, only the thresholds for the row selected to originally launch the waveform viewer is shown in the waveform viewer. These are listed in the header with the pin name of the receiver originally selected.

Reflection Vision

Reflection Vision shows color-coded segments based on a color gradient within the canvas.

Hover over a segment to view data tips. The focus data color coding and data tips vary depending on the Reflection Table focus data setting. Text at the bottom of the gradient legend shows the measurement being shown in the Vision if the Table is not open.

The color is from blue for the smallest gradient to red for the largest gradient. The color code is reversed for Ring Back Margin since smaller (red) values are worse than large values (blue).

Drag the pointer at the top or bottom of the gradient on the canvas to select specific regions of the results to display in the vision. For nets with multiple driver/receiver pairs, only the portion of the net between the worst-case driver/receiver pair for the focus measurement is colored in the vision.

Performing IR Drop Analysis

1. Choose *Analyze – Workflow Manager*.
2. Choose *IR Drop Workflow*.
3. Click *Setup Analysis Options* to set up topology extraction, default sink properties, current density threshold for vias and traces, design cutting and error checking for short circuits, and ideal ground for simulation using the [IR Drop Analysis Parameter Setup](#) window.
4. Under *Setup Modes*, select one of the options from the following:
 - *VRMs and Sinks*
 - *PowerTree File* (Default)
 - *Workspace File*

If you select VRMs and Sinks, define the voltage regulator modules and sinks using the *Set up VRMs* and *Set up Sinks* links.

If you select Powertree File, click *Select Power Tree* or edit the field below to specify the power tree file (.pwt).

Similarly, if you select Workspace File, click *Select Workspace File* or edit the field below to specify the workspace file (.pdcx).

5. Under Distributed Computing Setup, select *Enable Distributed Computing* to take advantage of available distributed computing.
This option is not selected by default.
6. Select the options to enable log scale, show vectors, and enable vertical range, if needed.
The options are not selected by default.
7. Click *Start Analysis*.
Progress of analysis is shown in the Analysis section of the Analysis Workflows window.
When analysis is complete, a green check appears for the *Start Analysis* task. If a previous analysis is stored, a prompt appears for overwriting the old analysis.
8. Click *Save Analysis Results* to save the results. Specify a name and location for the results file.
The extension for the result file is .irida for IR Drop analysis.
9. Select a value from the *View Modes* list. The available values are *IR Drop*, *Voltage*, and *Current Density*.

10. Click *IR Drop Table* or *IR Drop Vision* to view results.

Related Topics

- [IR Drop Analysis Parameter Setup Window](#)
- [IR Drop Table](#)
- [IR Drop Vision](#)

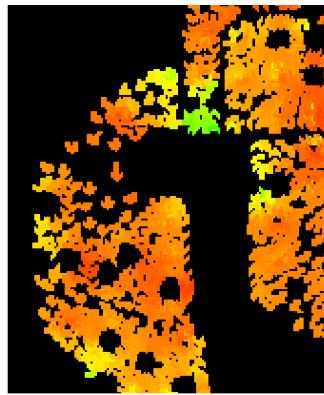
IR Drop Vision

Depending on the View Modes selection, color coded IR Drop, Voltage, or Current Density is displayed in the canvas.

Select a layer from *Vision Layer* to display information specific to the layer.

Select *Log Scale* to display the color gradient in log scale instead of a linear scale.

Select *Show Vectors* to display vector field for Current Density. Colored arrows show you the current flow.



Hover over pins and cline segments to see datatips.

IR Drop Table

Depending on the View Modes value chosen, you will see either the shrink voltage table or the current/current density table.

Shrink Voltage Table

The shrink voltage table is displayed when you click *IR Drop Table* after selecting either *IR Drop* or *Voltage* from the *View Modes* list. The Simulation table contains a net control section, a summary result section, and a detail result section.

The net control section displays power nets that have VRM defined. Select one or more nets to display the related sinks in the summary section. Choose the Vision column for a net to include in IR Drop vision.

The summary section lists the worst case for a component (sink). Select a row to display information in the details section. Double-click a row to perform a cross-probe the component.

The detail section lists the IR Drop results for each power pin on the component which is connected with the sink selected in summary result table. Double-click a row to perform a cross-probe for the selected pin.

The summary and detail section lists the *IR Drop*, Margin, and Status.

IR Drop (mV) is the difference of the actual or absolute voltage (V_{actual}) at the simulated sink or pin simulated by PowerTree and the nominal voltage ($V_{nominal}$) set in Power Tree.

$$V_{IRDrop} = V_{nominal} - V_{actual}$$

Margin (mV) is the difference of the actual voltage at the simulated sink or pin and either the lower or the upper voltage, depending on whether the actual voltage is less than or greater than or equal to the nominal voltage set in Power Tree.

If $V_{actual} < V_{nominal}$,

$$V_{margin} = V_{actual} - V_{lower}$$

If $V_{actual} \geq V_{nominal}$,

$$V_{margin} = V_{upper} - V_{actual}$$

Upper and lower voltages are calculated using upper and lower tolerances set in Power Tree as follows:

$$V_{upper} = V_{nominal} + |V_{nominal}| \times \frac{upper}{\%} tolerance$$

$$V_{lower} = V_{nominal} - |V_{nominal}| \times \frac{lower}{\%} tolerance$$

Status is *PASS* if *Margin* is greater than or equal to 0.

Current/Current Density Table

The current/current density table is displayed when you click *IR Drop Table* after selecting *Current Density* from the *View Modes* list. Three tabs are displayed *Via Current*, *Via Current Density*, and *Trace Current Density*.

The net control section displays power nets that have VRM defined. Select one or more nets to display the related thresholds in the summary section. Choose the *Vision* column for a net to include in IR Drop vision.

Double-click a row in any of the tables to highlight and zoom the corresponding via or trace in the design.

- *Via Current* table: In addition to the columns specified in the IR Drop Analysis Parameters Setup window, this table displays the following columns:
 - *Start Layer*: The start layer of a via section. Each via is translated into more than one via section based on its connection.
 - *End Layer*: The end layer of a via section. Each via is translated into more than one via section based on its connection.
 - *Actual Current (A)*: The current flow through a via determined from simulation result.
 - *Ratio*: The absolute value of the ratio between the actual current, as determined, and the maximum current (Max Current), as specified in the IR Drop Analysis Parameters Setup window.
 - *P/F Status*: Specifies the *PASS* or *FAIL* status. A *Ratio* of less than or equal to 1 is a

PASS status.

- *Via Current Density* table: In addition to the columns specified in the IR Drop Analysis Parameters Setup window, this table displays the following columns:
 - *Start Layer*: The start layer of a via section. Each via is translated into more than one via section based on its connection.
 - *End Layer*: The end layer of a via section. Each via is translated into more than one via section based on its connection.
 - *Actual Current Density (A/mm²)*: The current density of a via determined from simulation result.
 - *Ratio*: The absolute value of the ratio between the actual current density, as determined, and the maximum current density (Max Current Density), as specified in the IR Drop Analysis Parameters Setup window.
 - *P/F Status*: Specifies the PASS or FAIL status. A *Ratio* of less than or equal to 1 is a PASS status.
- *Trace Current Density* table: In addition to the columns specified in the IR Drop Analysis Parameters Setup window, this table displays the following columns:
 - *Actual Current Density (A/mm²)*: The current density of a trace determined from simulation result.
 - *Ratio*: The absolute value of the ratio between the actual current density, as determined, and the maximum current density (Max Current Density), as specified in the IR Drop Analysis Parameters Setup window.
 - *P/F Status*: Specifies the PASS or FAIL status. A *Ratio* of less than or equal to 1 is a PASS status.
- *Plane Current Density* table:
 - *Layer Name*: The layer containing the selected area of constraint.
 - *Net*: The net in which the density is being calculated.
 - *Metal Area*: The area for which the constraint is added. The coordinates of the lower left and upper right points of the area are listed.
 - *Maximum Current Density*: The specified maximum current density.
 - *Actual Maximum Current Density*: The current density determined from simulation result.

- Ratio: The absolute value of the ratio between the actual current density, as determined, and the maximum current density (Max Current Density), as specified in the IR Drop Analysis Parameters Setup window.

IR Drop Analysis Parameter Setup Window

Use the IR Drop Analysis Parameters Setup window to set the threshold for vias and traces.

Option	Description
Topology Extraction	
Component Type Identification	Specify the identifiers for topology extraction. Default values are populated.
Ignore resistors resistance larger than	Select to ignore resistances with resistance larger than a specified value. Selected by default. The default value is 100 Ohms.
Default Properties	
Default Sink Properties	Specify the sink properties in terms of current and upper and lower voltage tolerances. The tolerances are specified as a percentage value in a plus and minus range. The default current is set at 1A. The default value for lower and upper voltage tolerance is +/-5%.
Current Threshold	
IPC	<p>Select to calculate the global threshold automatically using the specified temperature rise. Calculation is based on IPC-2221A. The minimum width and thickness of conductors on board are determined primarily on the basis of the current-carrying capacity required, and the maximum permissible conductor temperature rise.</p> $I = k\Delta T^{0.44} A^{0.725}$ <p>Where,</p> <ul style="list-style-type: none"> • I is the current in amperes • A is the cross-section in square mils • ΔT is the temperature rise in $^{\circ}C$ • k is constant with value 0.048 for outer layers and 0.024 for inner layers

Apply	Click to calculate the global threshold if you selected IPC.
Manual	Specify the global threshold manually.
Threshold Table	Contains three tabs, Via Current/Current Density, Trace Current Density, and Plane Current Density and two options to specify color of selectable and selected objects. You can specify global threshold values for maximum current and current density, either through calculations based on IPC-2221A or manually. You can then add vias or traces to the appropriate table by right-clicking and choosing <i>Add</i> . When you choose <i>Add</i> , the valid vias and traces are highlighted in the canvas and you can click to select and populate the tables. You can only select through vias and traces that are connected to nets defined in the Power Tree you specified. SMD pins are ignored. You can also select one or more rows in the tables, right-click, and choose <i>Delete</i> , to delete one or more vias or traces.
Via Current/Current Density	Contains via information in tabular format, which can be set up by via or by padstack.
Threshold on Global Via Current (A)	Specifies the threshold value for via current either based on calculations if IPC is selected for Threshold Mode or the entered value if Manual is selected. Changes to Manual if the value is edited.
Threshold on Global Via Current Density (A/mm^2)	Specifies the threshold value for via current density either based on calculations if IPC is selected for Threshold Mode or the entered value if Manual is selected. Changes to Manual if the value is edited.
Setup by Via	Select to set up the threshold value table by via. The table will display the following columns: <i>Location (x,y)</i> , <i>Padstack Name</i> , <i>Net Name</i> , <i>Max Current (A)</i> , and <i>Max Current Density (A/mm²)</i> . Selected by default.
Setup by Padstack	Select to set up the threshold value table by padstack. The table will display the following columns: <i>Padstack Name</i> , <i>Max Current (A)</i> , and <i>Max Current Density (A/mm²)</i> .
Location (x y)	Specifies the coordinates of via. Available only if <i>Setup by Via</i> is selected.
Padstack	Specified the padstack of the via.
Net Name	Specified the name of the net connected to the via. Available only if <i>Setup by Via</i> is selected.

Max Current (A)	Specifies the threshold defined for the via current. If edited, overrides global value.
Max Current Density (A/mm ²)	Displays the calculated value of the threshold current density based on the threshold current specified and assuming a solid fill for the via drill. Cannot be edited.
Trace Current Density	Shows the threshold value for trace current density and threshold information for traces in tabular format, namely, <i>Location (x1 y1), (x2 y2), Net Name, Layer</i> , and <i>Max Current Density (A/mm²)</i> .
Threshold on Global Trace Current Density (A/mm ²)	Specifies the threshold value for trace current density either based on calculations if IPC is selected for Threshold Mode or the entered value if Manual is selected. Changes to Manual if the value is edited.
Location (x1 y1), (x2 y2)	Specifies the start (x1 y1) and end (x2 y2) locations of the trace.
Net Name	Specifies the name of the net connected to the trace.
Layer	Specifies the layer where the trace is located.
Max Current Density (A/mm ²)	Specifies the threshold defined for trace current density.
Layout	
Error Checking	Select options to check for short circuits and to stop simulation if short circuits are found. By default, both the options to find errors and stop simulation are selected.
Design cutting	Select to cut design by enabled nets. Not selected by default.
Simulation	

Ground	<p>Select to use ideal ground. Selected by default. If Use ideal ground is not selected, a real ground net must be present in the design.</p> <div style="border: 1px solid #ccc; padding: 5px; border-radius: 5px; background-color: #e0f2e0; margin-top: 10px;"> ✓ Use <i>Identify DC Nets</i> to check if there is any ground net exists in the design. A ground net has 0V assigned to the VOLTAGE property. </div>
--------	---

IR Drop Table

Depending on the View Modes value chosen, you will see either the shrink voltage table or the current/current density table.

Shrink Voltage Table

The shrink voltage table is displayed when you click *IR Drop Table* after selecting either *IR Drop* or *Voltage* from the *View Modes* list. The Simulation table contains a net control section, a summary result section, and a detail result section.

The net control section displays power nets that have VRM defined. Select one or more nets to display the related sinks in the summary section. Choose the Vision column for a net to include in IR Drop vision.

The summary section lists the worst case for a component (sink). Select a row to display information in the details section. Double-click a row to perform a cross-probe the component.

The detail section lists the IR Drop results for each power pin on the component which is connected with the sink selected in summary result table. Double-click a row to perform a cross-probe for the selected pin.

The summary and detail section lists the *IR Drop*, Margin, and Status.

IR Drop (mV) is the difference of the actual or absolute voltage (V_{actual}) at the simulated sink or pin simulated by PowerTree and the nominal voltage ($V_{nominal}$) set in Power Tree.

$$V_{IRDrop} = V_{nominal} - V_{actual}$$

Margin (mV) is the difference of the actual voltage at the simulated sink or pin and either the lower or the upper voltage, depending on whether the actual voltage is less than or greater than or equal to the nominal voltage set in Power Tree.

$$\text{If } V_{actual} < V_{nominal}, \\ V_{margin} = V_{actual} - V_{lower}$$

$$\text{If } V_{actual} \geq V_{nominal}, \\ V_{margin} = V_{upper} - V_{actual}$$

Upper and lower voltages are calculated using upper and lower tolerances set in Power Tree as follows:

$$V_{upper} = V_{nominal} + |V_{nominal}| \times \text{upper tolerance } \%$$

$$V_{lower} = V_{nominal} - |V_{nominal}| \times \text{lower tolerance } \%$$

Status is *PASS* if *Margin* is greater than or equal to 0.

Current/Current Density Table

The current/current density table is displayed when you click *IR Drop Table* after selecting *Current Density* from the *View Modes* list. Three tabs are displayed *Via Current*, *Via Current Density*, and *Trace Current Density*.

The net control section displays power nets that have VRM defined. Select one or more nets to display the related thresholds in the summary section. Choose the *Vision* column for a net to include in IR Drop vision.

Double-click a row in any of the tables to highlight and zoom the corresponding via or trace in the design.

- *Via Current* table: In addition to the columns specified in the IR Drop Analysis Parameters Setup window, this table displays the following columns:
 - *Start Layer*: The start layer of a via section. Each via is translated into more than one via section based on its connection.

- *End Layer*: The end layer of a via section. Each via is translated into more than one via section based on its connection.
- *Actual Current (A)*: The current flow through a via determined from simulation result.
- *Ratio*: The absolute value of the ratio between the actual current, as determined, and the maximum current (Max Current), as specified in the IR Drop Analysis Parameters Setup window.
- *P/F Status*: Specifies the PASS or FAIL status. A *Ratio* of less than or equal to 1 is a PASS status.
- *Via Current Density* table: In addition to the columns specified in the IR Drop Analysis Parameters Setup window, this table displays the following columns:
 - *Start Layer*: The start layer of a via section. Each via is translated into more than one via section based on its connection.
 - *End Layer*: The end layer of a via section. Each via is translated into more than one via section based on its connection.
 - *Actual Current Density (A/mm^2)*: The current density of a via determined from simulation result.
 - *Ratio*: The absolute value of the ratio between the actual current density, as determined, and the maximum current density (Max Current Density), as specified in the IR Drop Analysis Parameters Setup window.
 - *P/F Status*: Specifies the PASS or FAIL status. A *Ratio* of less than or equal to 1 is a PASS status.
- *Trace Current Density* table: In addition to the columns specified in the IR Drop Analysis Parameters Setup window, this table displays the following columns:
 - *Actual Current Density (A/mm^2)*: The current density of a trace determined from simulation result.
 - *Ratio*: The absolute value of the ratio between the actual current density, as determined, and the maximum current density (Max Current Density), as specified in the IR Drop Analysis Parameters Setup window.
 - *P/F Status*: Specifies the PASS or FAIL status. A *Ratio* of less than or equal to 1 is a PASS status.
- *Plane Current Density* table:

- Layer Name: The layer containing the selected area of constraint.
- Net: The net in which the density is being calculated.
- Metal Area: The area for which the constraint is added. The coordinates of the lower left and upper right points of the area are listed.
- Maximum Current Density: The specified maximum current density.
- Actual Maximum Current Density: The current density determined from simulation result.
- Ratio: The absolute value of the ratio between the actual current density, as determined, and the maximum current density (Max Current Density), as specified in the IR Drop Analysis Parameters Setup window.

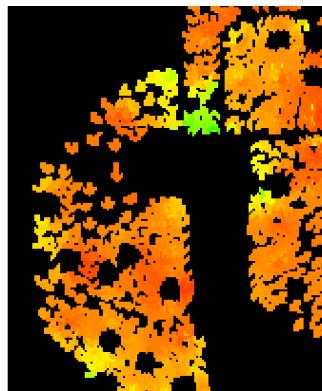
IR Drop Vision

Depending on the View Modes selection, color coded IR Drop, Voltage, or Current Density is displayed in the canvas.

Select a layer from *Vision Layer* to display information specific to the layer.

Select *Log Scale* to display the color gradient in log scale instead of a linear scale.

Select *Show Vectors* to display vector field for Current Density. Colored arrows show you the current flow.



Hover over pins and cline segments to see datatips.

Performing Power Inductance Analysis

Use the Power Inductance workflow to analyze loop inductance from capacitor to IC device or for power pins.

Evaluate the effectiveness of capacitor locations by quickly extracting the loop inductance of each capacitor observed from the given devices on a power rail.

Identify weak pins and meet the different per-pin PDN impedance requirements due to different current of various pins of device by quickly calculating the loop inductance of each power pin for IC devices.

1. Choose *Analyze – Workflow Manager*.
2. Choose *Power Inductance Workflow*.
3. Click *Design Setup* to verify and set up the design. Under Analysis Setup, click *Select Nets*.
4. Under Analysis Setup, click *Select Nets*.
5. Click *Select ICs and Decoupling Capacitors* to select ICs and decaps.

Notice the green tick against this option because the ICs and capacitors related to the selected nets are included by default under Net Pair Table.

6. Click *Set up Frequency* to open the PowerSI tab of the IDA Power Inductance Analysis Parameters Setup dialog box.
7. Edit the frequency value, if needed.
By default, the value is 1MHz .
8. Set translation and simulation settings in the IDA Power Inductance Analysis Parameters Setup dialog box.
9. Click *Start Analysis*.
Progress of analysis is shown in the Analysis section of the Analysis Workflows window.
When analysis is complete, a green check appears for the *Start Analysis* task. If a previous analysis is stored, a prompt appears for overwriting the old analysis.
10. Click *Save Analysis Results* to save the results. Specify a name and location for the results file.
The extension for the result file is `.pindida` for the Power Inductance analysis.
11. Select a view from View Modes.
The available values are *Capacitor to IC* and *IC Power Pin*.
Depending on the selection, you can view the Power Inductance table or vision.

You can also create an HTML report at the end by clicking *Create Report*. The report will include all inductance tables and plots.

Related Topics

- [Setting Up Design](#)
- [Selecting Nets](#)

Performing Topology Extraction

1. Choose *Analyze – Workflow Manager*.
2. Choose *Topology Extraction Workflow*.
3. Under Analysis Setup, click Select Nets.
4. Click *Set up Analysis Options* to open Topology Extraction Parameters Setup and specify the Minimum Coupled Length for extraction.
5. Click *Start Extraction*.

Related Topics

- [Selecting Nets](#)

Performing Via Modeling using a Wizard

The Via Wizard Workflow is available when you open a layout for via modeling from Topology Workbench.

1. If you have a previously saved layout file (.vwx), click *Load Layout* and load the file saved layout file to import cross section, via structure, and analysis options.
If you do not have a layout file, define cross section and create via structure.
2. Click *Define Cross-section* to open the Cross-section Editor dialog box and define cross-section.
You must define at least one Plane layer to be able to validate cross-section and proceed.
A green tick mark is displayed when the cross-section is defined.
3. Click *Create via structure* to create the via structures using the Create Structures window.
4. Set up the Analysis Setup options by clicking the following links:
 - a. *Set up Frequencies*: Opens the Frequency Settings tab of the Clarity 3D FEM engine in the IDA Via Wizard Analysis window.
 - b. *Set up Solver Options*: Opens the Frequency Settings tab of the Clarity 3D FEM engine in the IDA Via Wizard Analysis window.
 - c. *Set up Geometry Options*: Opens the Geometry Options tab of the Clarity 3D FEM engine in the IDA Via Wizard Analysis window.
 - d. *Set up Analysis Options*: Opens the Translation tab of the IDA Via Wizard Parameters Setup window.
5. Click *Save Layout* to save the layout file.
6. You can view the 3D model by clicking *View 3D Geometry*.

Performing Interconnect Model Extraction

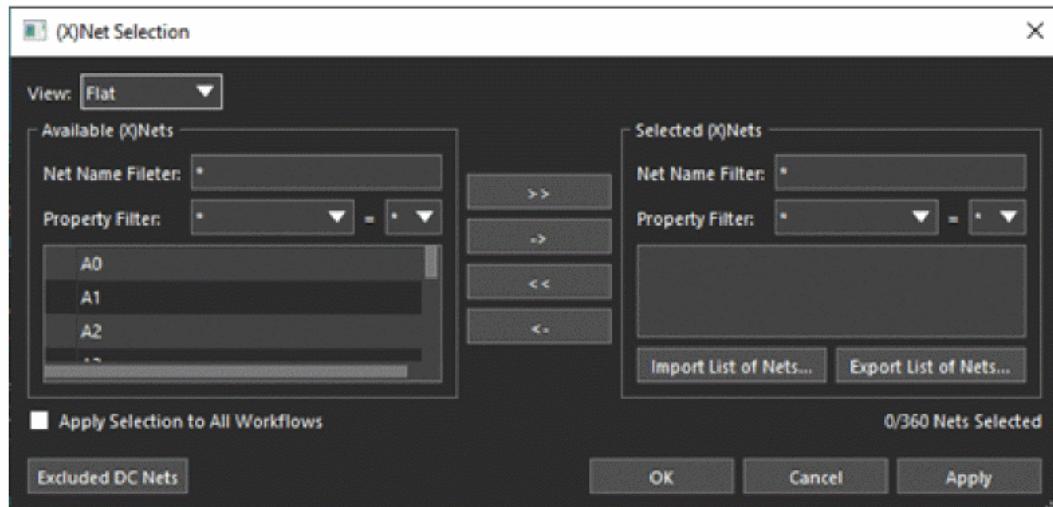
Selecting Nets

In a net-based analysis, every segment of the selected nets is simulated. Whereas, in a directed group, only net portions specified by a start component and one or more end components are simulated.

If analysis mode is set to *Net Based*, the *Select Nets* flow step is presented and marked with a cross to indicate that the step must be completed.

To select nets or modify a selection, do the following steps:

1. Click *Select Nets* to open the *(X)Net Selection* window.



The available nets are listed in the left pane and the selected nets in the right pane.

2. Select either *Flat* or *Hierarchical* from View.

The *Flat* view lists all nets in the design in alphabetical order.

The *Hierarchical* view lists the nets under defined hierarchy objects, such as differential pairs, net groups, bus, XNets, or nets.

3. You can add one or more nets directly from the canvas by clicking or windowing around nets.

Click nets while pressing `ctrl` to remove them from selection. When done, right-click and choose *Done* or *Clear All*.

Or, you can select available nets or XNets from the left pane and double-click to add to the selected list on the right. Similarly, select nets or XNets on the right plane and double-click to remove them from the selected list.

Use the *Available XNets/nets* and *Selected XNets/nets* fields to filter out specific XNets/nets. You can also click the buttons in the middle to select nets. The >> button selects all available nets whereas << removes all selections.

Voltage nets are excluded from the list of available nets. To see a list of the excluded nets, click *Voltage Nets Excluded from List*.

You can also import from a list of nets by clicking *Import List of Nets*. You can click *Export List of Nets* to export the list of selected nets to a .lst file.

4. Click *Apply* to set the changes or click *OK* to set the changes and exit the dialog.

The Select Nets step has a green check () and you can proceed with the rest of the flow steps.

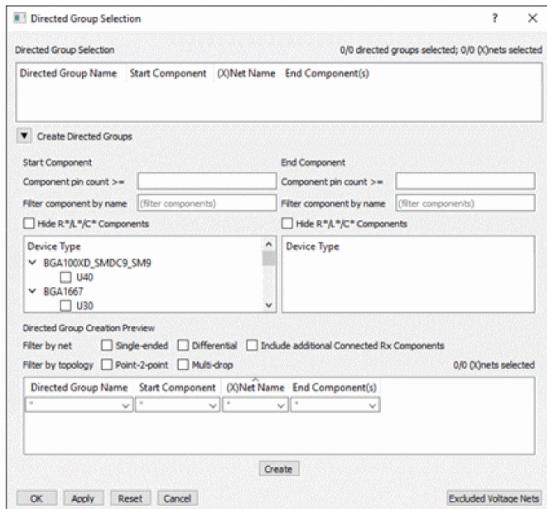
Specifying Directed Groups

In directed groups or component-based net selection, you can select an entire interface based on component connectivity rather than each net name.

Directed groups help prioritize issues; for example, issues found near the start point as compared to issues found near the end point. In addition, directed groups let you associate specific analysis results for a collection of nets as encountered along the same *time-line* as progressing from the start to the end point.

If analysis mode is set to *Directed Group*, the Select Directed Group flow step is presented and marked with a cross to indicate that the step must be completed.

To select, create, or modify directed groups or modify a selection; click *Select Directed Group* to open the Directed Group Selection dialog box.



The top section, *Directed Group Selection*, lists the selected directed groups. A directed group is the portion of a net between a starting component and one or more ending components.

Use the expandable *Create Directed Groups* section to create directed groups.

To create a directed group, do the following steps:

1. Select a start component either on the canvas or from the list of components in the Start Component box or on the canvas.

When you click a start component, all possible end components are visible while the other components are dimmed on canvas.

The valid end components are listed under *End Component*.

Hide the passive components by selecting *Hide R*/L*?/C* Components*. You can also filter the list for ease of use.

2. Select one or more end components.

A preview of the directed group is shown in the bottom section.

Directed Group Name	Start Component	Net Name	End Component(s)
*	U11	JUNK	U12
*	U11	NET_ALL_11	U12
*	U11	NET_PU_21	U12
*	U11	NET_UU_7	U12
*	U11	NET_UU_8	U12

All nets are selected by default. You can filter the list to display selected nets.

3. If needed, right-click in the preview area and choose any one of the options to select nets based on any filters.

Directed Group Name	Start Component	Net Name	End Component(s)
*	U11	JUNK	U12
*	U11	NET_ALL_11	U12
*	U11	NET_PU_21	
*	U11	NET_UU_7	
*	U11	NET_UU_8	

- Add All Visible Nets to the Selection
- Deselect All Visible Nets
- Deselect All Nets
- Select All Visible Nets Only

The nets that are displayed based on the filter settings are the visible nets. In the example image, the filter is set to * to select all nets.

4. Click *Create* to create the directed group.

The group is created and selected by default.

To select available directed groups or modify a section, do the following steps:

1. Select listed directed groups from the *Directed Group Selection* section.

The selected groups are marked by a tick in the box on the left.

2. Click *Apply* to update the design or click *OK* to update the design and close the dialog box.

The Select Directed Groups step has a green check and you can proceed with the rest of the flow steps.

Setting Up Component Models

You can choose to use:

- Default pin models for all pins in simulation
Requires no component models and assignment to achieve simulation.
Accept the default set up and proceed to select nets and set up analysis.
- Only models assigned in the design
Requires full setup to achieve accurate simulation and requires all pins as part of the simulation to have assigned models through existing component models.
Ensure that *Use defaults for missing models* is not set and then specify the default models and set up the libraries.
- Models assigned in the design, but apply default pin models for any missing models
Allows simulation to proceed even if there are some missing model by using the default pin models.
Ensure that *Use defaults for missing models* is set and then specify the default models and set up the libraries.

 Only IBIS buffer models are considered for crosstalk.

Setting Up Default Models

To set up the default models:

1. Click *Default Model Setup* to open Crosstalk Default Model Setup.
Crosstalk Default Setup lists the default IO cell models for single-ended and differential pin use: *IN*, *OUT*, and *BI/Other*. By default, models are picked from the Cadence installation. Set `IDA_MODEL_PATH` to specify a different path.
2. Click *Browse* button for a model type to open *Default Model Browser* and change the default model.
3. Set a search path.
4. Select a model and click *OK*.

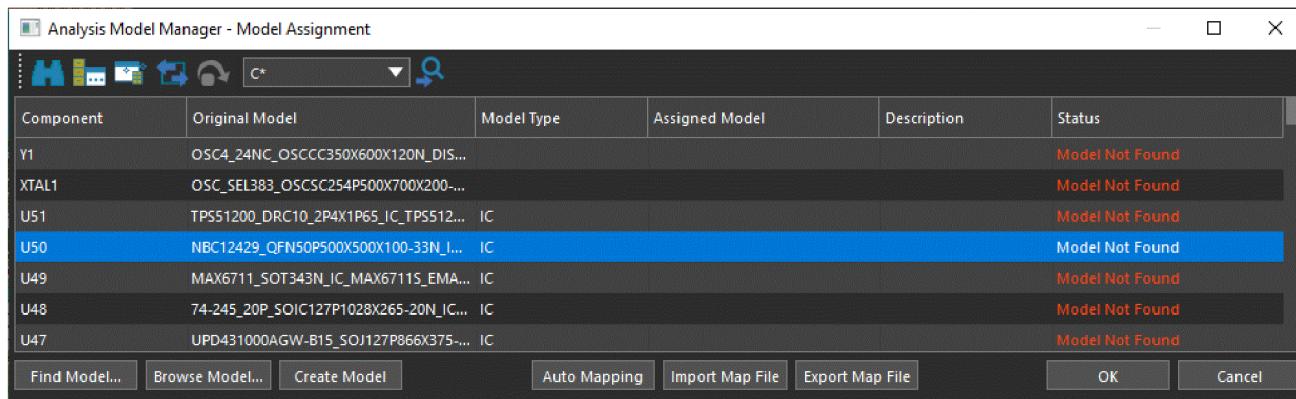
The default model for the selected model type is updated. The search path for all models is also changed if it is set to a new path. However, the other default models are not changed.

Using Analysis Model Management

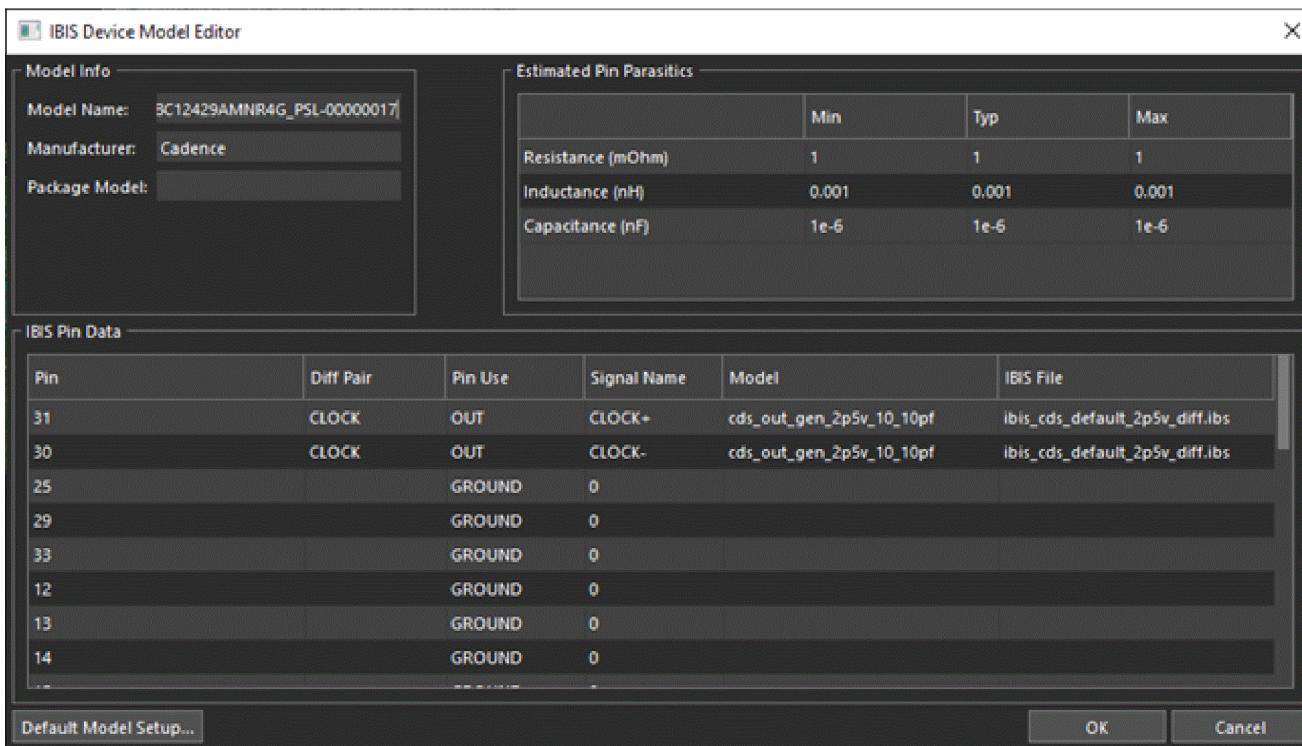
Use Analysis Model Manager (AMM) to manage libraries and assign models.

Click *Manage Libraries* to open AMM Library Management to quickly view the libraries that are loaded and to manage libraries without opening the entire AMM dialog.

To assign IBIS models, click *Model Assignment* to open Analysis Model Management - Model Assignment. Use this to create models or browse for models. You can browse IBIS libraries directly.



Click *Create Model* to generate an IBIS model from an Allegro component with Model Type *IC*. This will open the IBIS Device Model Editor window that you can use to browse and select IO models.



You can specify the IBIS model, manufacturer, and global package parasitic under Model Info.

⚠ Analysis Model Management - Model Assignment uses the ASI_MODEL property to pass model assignments to AMM. The ASI_MODEL property is applied to component instance and definition.

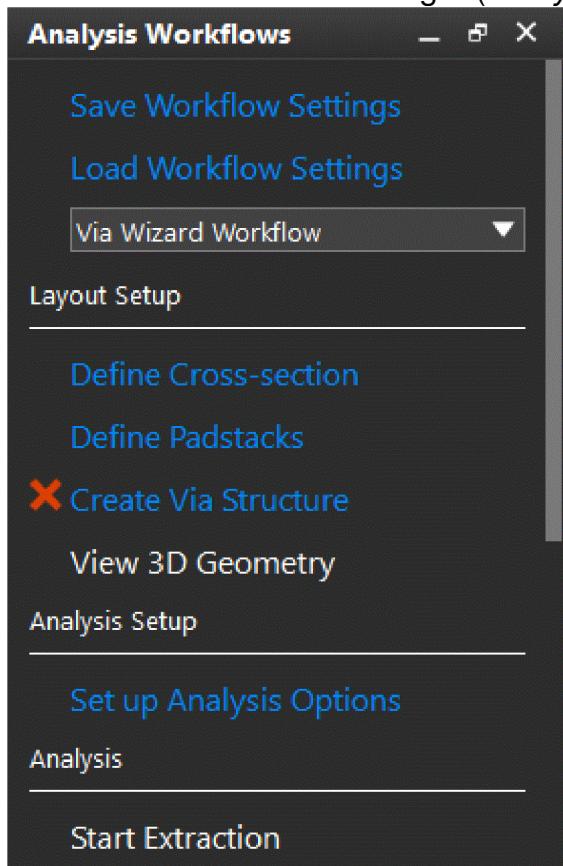
⚠ Set up component models for the following workflows:

- [Performing Crosstalk Analysis](#)
- [Performing Reflection Analysis](#)

IDA Via Wizard Workflow

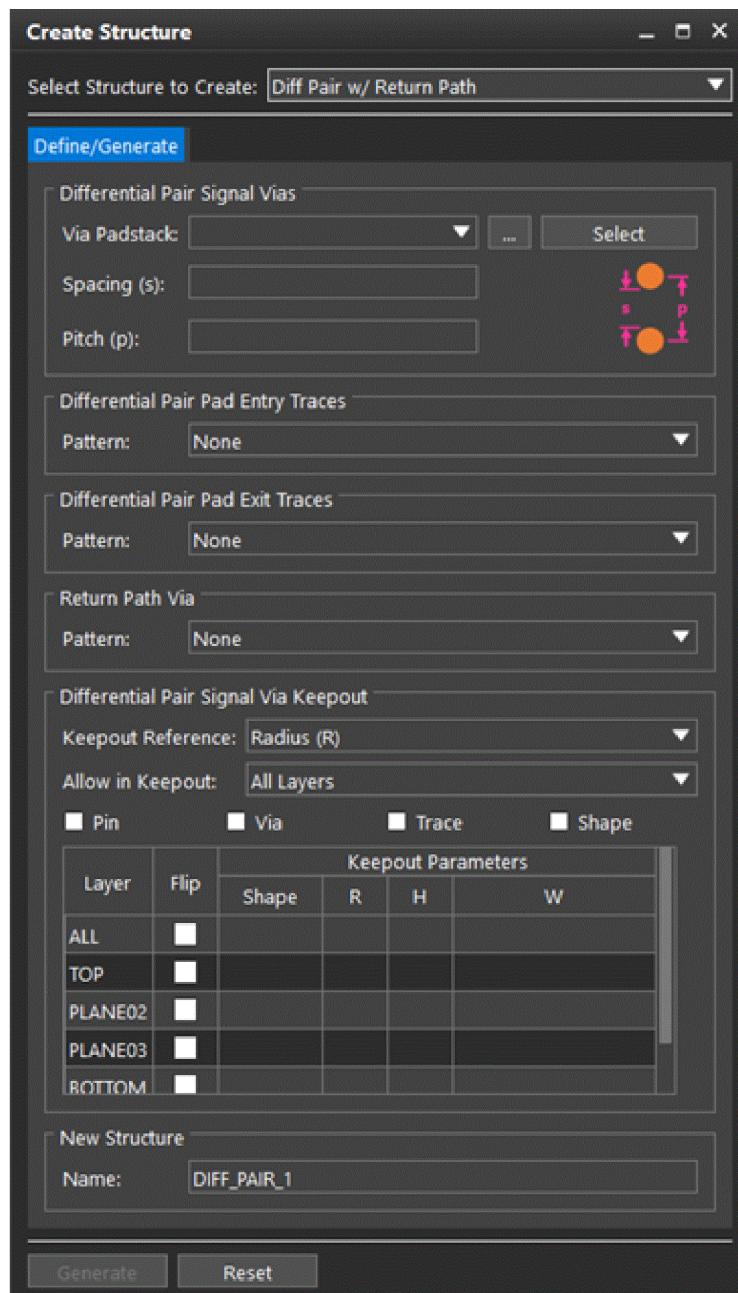
Use the Via Wizard to quickly and accurately build a 3D via model. Vias are a critical structure in high-speed signal and power nets. The accuracy of signal simulations is affected by a via is modeled.

The Via Wizard is available in the IDA Workflow Manager (*Analyze – Workflow Manager*).

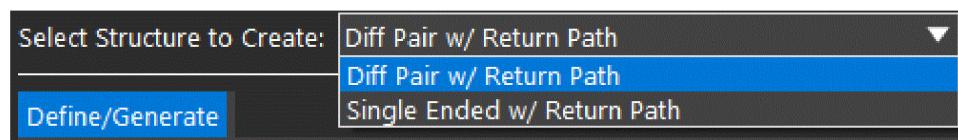


To define a via using the IDA Via wizard, do the following steps:

1. In the Analysis Workflows window, ensure *via wizard workflow* is selected.
2. Define the cross-section using the Cross-section Editor.
3. Define padstacks using the Padstack Editor.
You can edit an existing padstack from Options pane for an instance or change the definition.
4. Click *Create Via Structure* to open the Create Via Structure window.



The parameters you specify in the Define/Generate section of the window changes depending on the value you select in Select Structure to Create field. The valid values are Diff Pair w/Return Path and Single Ended w/Return Path, for a differential pair or single-ended via respectively.



5. Select the structure to create.

You can either choose to create a differential pair or a single-ended via, both with return path.

6. Specify the via padstacks.

Select from the Via Padstacks list or select from the Library Padstack Browser that opens on clicking the more button.

Click *Select*.

7. For a differential pair vias, specify the spacing and the pitch.

8. Specify the pad entry and exit trace patterns.

For a differential pair, the values can be *None* (default), *Rectangular*, *Offset*, or *Tandem*. For a single-ended via, the values can be *None* (default) or *Single*.

9. Specify the pattern for the return path via.

For a differential pair, the values can be *None* (default), *Inline*, *Equidistant*, *Diamond*, or *Single*. For a single-ended via, the values can be *None* (default) or *Single*.

10. For a differential pair, specify the keepout.

The reference can be a *Radius (R)* or *Spacing (S)*.

11. For a single-ended via, specify the via void.

12. Verify the name of the structure.

13. Click *Generate*.

A green tick will appear next to Create Via Structure to indicate that the structure has been successfully created.

The View 3D Geometry option will be listed in the wizard flow.

To open the 3D model in Allegro 3D Canvas, click *View 3D Geometry*.

14. Set up analysis by clicking *Set up Analysis Options*.

The IDA Via Wizard Analysis Parameters Setup window appears.

15. Set the S-parameter model format in the Simulation tab by specifying the Port Reference Impedance and the Output Format for Network Parameters.

16. If needed, specify the frequency settings for 3D Clarity Layout.



Refer to the 3D Clarity Layout user guide for more detail on parameters.

17. Click *Apply* to save the changes.

18. Click *OK* to close the window.

The View

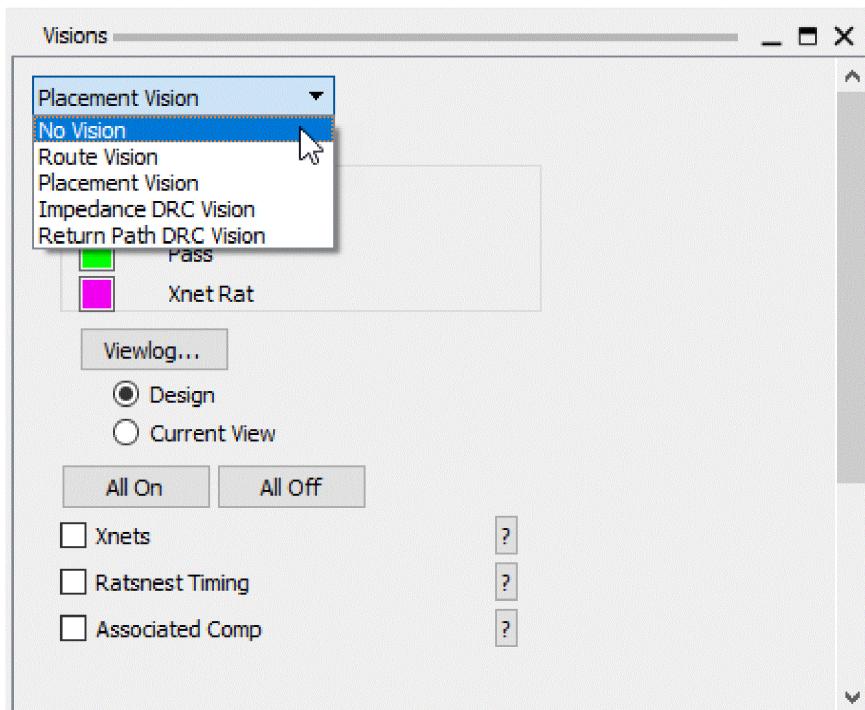
19. Click *Start Extraction* to start the simulation and extract the model.
The extracted via model is refreshed in Topology Explorer.

Using Vision Manager: Visual Results for Analysis and DRCs

Use the Vision feature to display selected nets or segments of an analysis or DRC result using color codes. The nets or segments that are not part of the analysis or DRC are dimmed, by default.

Access the different visions using *Vision Manager* (*View – Vision Manager*).

 Vision Manager is available from Allegro X PCB Editor when the PCB *High-Speed* option is chosen.



You can access any one of the following visions:

- *Route Vision*
- *Placement Vision*

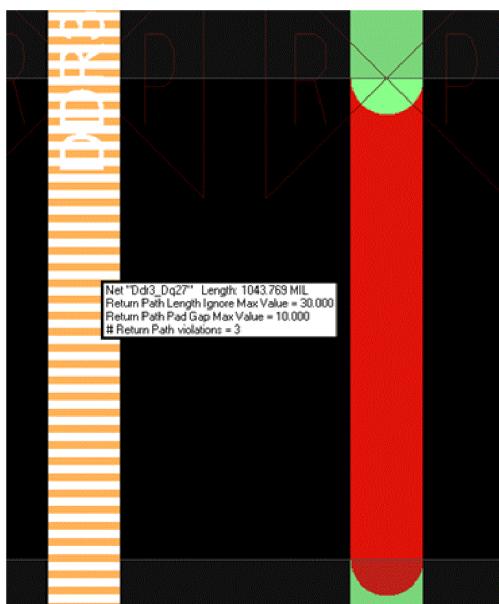
- *Impedance DRC Vision*
- *Return Path DRC Vision*

 The Visions available depends on the license you are using. Depending on your license, you may not be able to access all the Visions.

To view results for an analysis, do the following steps:

1. Select the analysis for which you want to see the color-coded results.
The available Visions depend on the license you use.
2. Depending on the vision specify the color code.
3. For Route Vision, click *Configure* and select the rules.
The results are displayed in the Vision Report window.
4. For Placement Vision, set any or all the rules.
The violations are displayed in the selected color in the canvas.
5. For Impedance and Return Path. select the nets for which you want to view the vision.
 - a. To modify selected nets, click *Modify Selection*.
 - b. Right-click and choose *Done* when finished.
 - c. Hover over a net to view details of the violations.

The violations are displayed in the selected color. For example, in the following image, segments with violation are shown in red and on hovering over a net shows details of the return path status.



Route Vision

<i>Parallel Gap Less Than Preferred</i>	<p>Set this rule if you specify a gap value larger than the DRC minimum. The segments with a gap value less than what you specified are highlighted by showing them in the Route Vision color, red by default. The regular color is shown for segments that meet or exceed the gap value you specified.</p> <p>For example, if you set a gap value of 10 and the Route Vision color is red, all segments with gap value less than 10 will be shown in red.</p>
<i>Non-Optimized Segs</i>	<p>Set to highlight non-optimized segments by showing these segments in the Route Vision color, red by default. Optimized segments have minimum line-to-line spacing and maximum pad-to-line spacing. After setting this rule and closing the Route Vision Configure window, click <i>Channel</i> in the Vision window to define the air gap for the required channel size.</p>
<i>Uncoupled Diff-Pair Segs</i>	<p>Set to highlight uncoupled differential-pair segments by showing these segments in the Route Vision color, red by default.</p> <p>After setting this rule and closing the Route Vision Configure window, set <i>Ignore uncoupling at gathers</i>' so uncoupled differential-pair segments exiting the pad will not be indicated, until after cline coupling has occurred.</p>

<i>Non-ideal Pad Entry</i>	<p>Set to highlight segments having Non-Ideal Pad Entry by showing these segments in the Route vision color, red by default.</p> <p>Ideal Pad Entry has proper pad entry direction, and the first segment completely outside the pad must honor the same-net line-to-pad spacing constraint.</p>
<i>90 Degree Corners</i>	<p>Set to highlight segments at 90 degree corners by showing these segments in the Route Vision color, red by default. The shorter segment of the two is the one indicated.</p>
<i>Min Miter/Corner Size</i>	<p>Set to highlight corner segments having miter/corner size less than the minimum value specified by showing these segments in the Route Vision color, red by default.</p> <p>Use this rule to verify that segments are honoring the <i>Miter</i> option of Add Connect and the <i>Min Corner Size</i> option of Slide.</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> ⚠ This check is for segments only and not for arc corners. </div>
<i>Min Seg/Arc Length</i>	<p>Set to highlight segments with length less than the minimum value specified by showing these segments in the Route Vision color, red by default.</p>
<i>Min Arc Radius</i>	<p>Set to highlight arcs with radius less than the minimum value specified by showing them in the Route Vision color, red by default.</p>
<i>Non Arc Corners</i>	<p>Set to highlight non-arced corners by showing them in the Route Vision color, red by default. The shorter segment of the two (either arc or straight) is the one indicated.</p>

Placement Vision

Xnet Vision	<p>Set to see all the XNet rats of a design. For each XNet, a separate line is shown connecting the start and the end pins of the XNet. The line segments of the XNet that connect to the discrete component are ignored in the vision.</p> <p>XNet rats are shown in the Placement Vision XNet rat color, magenta by default.</p>
Ratsnest Timing Vision	<p>Set to compare the Manhattan distance of the ratsnest with the DRC Timing constraints defined in Constrained Manager. Ratsnests with Manhattan distance more than the distance calculated from its constraints are shown in the Placement Vision Pass color, green by default.</p>
Associated Component Vision	<p>Set to check the spacing of associated components from their parent component. Associated components placed outside the allowed distance are shown in the Placement Vision Fail color, red by default. The associated components that meet the spacing constraints are shown in the Placement Vision Pass color, green by default.</p>

APD: Working with a Plating Bar

A plating bar is used for electroplating the exposed conductors. If you plan to electroplate items on your component, such as component pins and bond wire pads, you must include a plating bar in your design. Not all manufacturing processes use a plating bar technique. In these situations, the plating bar symbol is not required.

With APD, you can create a plating bar containing one "phantom" pin for each plating bar connection. The tool routes each net to the plating bar to ensure connectivity to all entities that require electroplating. A plating bar pin for each unconnected entity must also be embodied in the plating bar to plate for unconnected component pins, bond fingers, and die pins. Because a majority of the nets are connected to the component I/O pins, routing the plating bar pins to the component I/O pins is sufficient. In multi-die APD designs, it may be necessary to add plating bar connections for nets which are otherwise internal to the package (die-to-die nets).

This section describes the plating bar generator as a discrete feature.

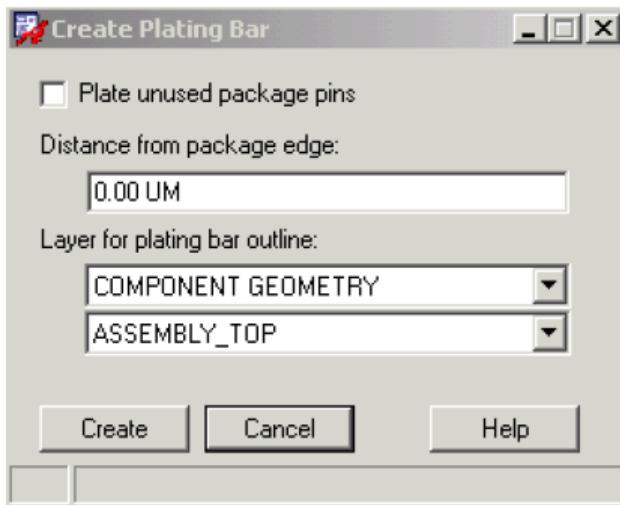
Prerequisites

You must route any nets to be plated out to the package boundary. Any routing trace that is at the package boundary is automatically trimmed or extended to intersect the plating bar when you run the `pbar create` command.

Creating a Plating Bar Symbol

The preferred method for creating a plating bar symbol is by choosing *Manufacture – Create Plating Bar* (`pbar create`) from the menu bar. When you run this command, the tool creates a plating bar component with a rectangular symbol that parallels the coordinate axes of your layout design. You can specify the distance from the component pins to the plating bar and choose whether to assign a net name to each unused component pin and route it to the plating bar in the Create Plating Bar dialog box.

Plating Bar Create Dialog Box



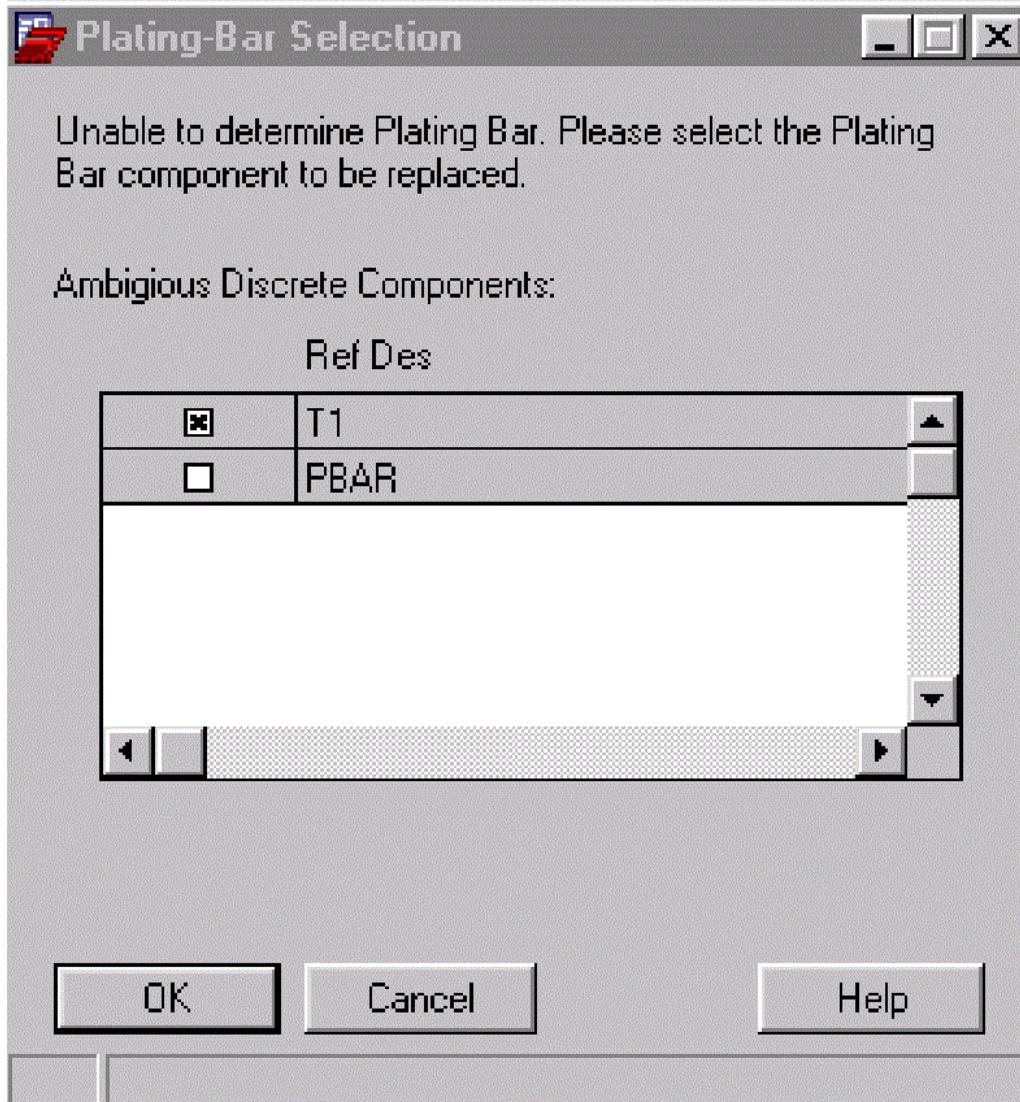
If a plating bar already exists in your design, and you are running the command to create a new plating bar, you must first delete the plating bar (`pbar delete`). The symbol and component definitions and reference designation of the old plating bar are attached to the replaced symbol. Note the two following conditions when replacing an existing plating bar with a new one:

- DRC errors can occur if minimum pin-to-pin spacing exceeds minimum line-to-line spacing of the new plating bar.
- If the existing plating bar has the **FIXED** property attached, it cannot be replaced. In this case, the following error message is generated in the console window:

The plating bar component has the **FIXED** property, so it cannot be deleted

Because your design should not contain more than one plating bar, the Plating Bar Selection dialog box appears if the tool recognizes another component instance that it cannot distinguish from a plating bar symbol. This occurs when there is more than one component in the design that meets the criteria that APD uses to recognize the plating bar; for instance, if components other than the plating bar have the **PLATING_BAR** property.

Plating Bar Selection Dialog Box



Manufacture – Create Plating Bar (`pbar create` command) treats connect lines (clines) in your design in the following manner:

- Clines at the component boundary are extended orthogonally to the plating bar boundary.
- Clines that extend past the plating bar symbol are trimmed to the plating bar boundary.
- Clines whose endpoints fall between the component boundary and plating bar are extended orthogonally from the endpoints to the plating bar.

These behaviors occur without regard for a FIXED property attached to nets or clines. However, clines with the NO_ROUTE property attached are not routed to the new plating bar.

Certain conditions may preclude the tool from creating a plating bar. When this occurs, the following

error messages may be generated:

The existing plating bar component has the FIXED property, so plating-bar creation cannot be performed.

Error: no BGA Component detected. Cannot create a plating-bar.

No clines intersect with the BGA boundary. Plating bar not created.

To view the procedure for creating a plating bar with the plating bar generator, see *Manufacture – Create Plating Bar* (`pbar create` command).

Alternative Plating Bar Creation Techniques

You also can create a plating bar using these methods:

- Graphically

Create the plating bar in the Symbol Editor by drawing one or more copper shapes and "phantom" pins. You attach the PLATING property to the copper shape to instruct the tool to short the nets. This prevents the tool from reporting net spacing violations during error checking. You place plating bars as you would place any other component. APD can semi-automatically or interactively assign nets to plating bar pins for routing. A plating bar checker verifies and reports connectivity errors.

- Padstack Editor and Symbol Editor

Use the Padstack Editor and Symbol Editor to create the pins, add the pins, add the symbol outline, and add the reference designator label. Also generate a device file for the plating bar symbol ensuring that the CLASS=PLATING_BAR property is assigned in the device file.

Logic – Auto Assign Net (`auto assign net` command) uses the CLASS property and its value to determine which symbol is the die, BGA, and plating bar. Menu selection and command entries you can use to create a plating bar symbol include:

Layout – Pins (`add pin` command)

Areas – Component Boundary (`component bound` command)

Layout – Labels – RefDes (`label refdes` command)

File – Save (`save` command)

Related Topics

- [pbar create](#)
- [pbar delete](#)
- [add pin](#)
- [auto assign net](#)
- [save](#)

Verifying Plating Bar Errors

You can verify and report plating bar connectivity errors and plating trace spacing violations, and delete DRC errors generated in previous runs of the command using the [pbar check](#) command.

As the design of IC packages has continued to evolve, the need for this command continues to change. As an example, in the past, it was necessary only to check for connectivity of the balls of the package (BGA), and optionally the bond fingers for a wire bond die. Now, discrete components are embedded into the design, etch-back techniques are used to plate dense designs, and multi-chip packages contain nets that are completely internal to the package, having no connection to the BGA component pins.

The following shows a sample log file. The actual values for pin, net, finger names, locations, and required or actual values change per design content.

```
ERROR: Minimum plating trace spacing violated (Required: 5 UM; Actual: 4 UM)
At (0.000 0.0000) (Net "ABC" and Net "XYX") .

ERROR: Minimum plating trace straight line length (Required: 5 UM; Actual: 4 UM)
At (0.000 0.0000) (Net "ABC" and Net "XYX") .

ERROR: Minimum plating trace offset violated (Required: 5 UM; Actual: 4 UM)
At (0.000 0.0000) (Net "ABC" and Net "XYX") .

ERROR: BGA.A1 not connected to plating bar

WARNING: BGA.A2 not connected to plating bar

WARNING: BF23 is plated through an etchback connection

WARNING: U1.45 has multiple (4) connections to plating bar

ERROR: No plating bar component found.
Likely cause: plating bar has wrong component class.
```

Deleting a Plating Bar

You can delete the plating bar in your design by choosing *Manufacture – Delete Plating Bar* ([pbar delete](#) command).

Delete Plating Bar Dialog Box



This action removes the plating bar's component and symbol instance and definition, unless the plating bar has the FIXED property attached. The tool generates error messages if the tool does not detect the presence of a plating bar, or it cannot distinguish the plating bar component from other components of the class DISCRETE. In circumstances where the tool detects ambiguous components, a list of such elements is displayed from which you can select the plating bar component.

If APD cannot distinguish the plating bar from another component instance, the **Plating Bar Selection** dialog box appears. This occurs when there is more than one component in the design that meets the criteria the tool uses to recognize the plating bar; for instance, if components other than the plating bar have the PLATING_BAR property.

To view the procedure for deleting a plating bar with the plating bar generator, see *Manufacture – Delete Plating Bar* ([pbar delete](#) command) in the *Allegro PCB and Package Physical Layout Command Reference*.

Generating a Plating Bar Report

Choose *Manufacture – Plating Bar Check* ([pbar check](#) command) to verify and report plating bar connectivity errors, or simply delete DRC errors generated in previous runs of the command.

Plating Bar Check Dialog Box

