cādence®

Allegro® PCB Router User Guide

Product Version 23.1
September 2023

Document Last Updated: October 2018

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

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

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. 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/or 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

2

3

About this Manual

Welcome

Allegro® PCB Router is a design automation tool that is used to automatically or interactively route dense (PCB, Package, and MCM) designs. The router includes a graphical user interface (GUI) and uses a routing engine that features ShapeBased technology where circuit elements are modeled as basic geometric shapes. Each shape can have rules associated with it that enforce design constraints such as component spacing and orientation, wire width and clearance, timing, noise, and crosstalk.

Audience

This manual is written for Electrical Engineers and Layout Designers who are new to the router but are already familiar with current methods and practices used to design printed circuit boards, packages, and multi-chip modules.

Where to Find Information in this Manual

This user guide contains the following chapters and appendices.

- Chapter 1, “Getting Started” outlines what you need to know to begin working with the router and includes licensing information and startup options.
- Chapter 2, “Working with the Router” provides an overview of the router user interface and the various command options.
- Chapter 3, “Setting Rules and Constraints,” describes the restrictions you can put in place in order to optimize your design.
- Chapter 4, “Placing Components,” explains how to set placement controls, the use of a floor plan, and considerations for meeting design requirements.
- Chapter 5, “Routing Connections”, covers the process of setting up the routing environment, including automatic routing, interactive routing, and managing the routing topology.

- Chapter 6, "Working with Advanced Technologies", describes the issues surrounding crosstalk, coupled noise, and high-density designs.
- Chapter 7, "Troubleshooting the Design File", explains the methods for correcting and optimizing the router Design File.
- Appendix A, "Router Startup Options," identifies switches that are available when you start the router and associated Did File considerations.
- Appendix B, "Setting Colors and Fonts," lists the default colors and fonts used by the router and how you can change them.
- Appendix C, "Understanding Symbols", illustrates and defines the router violation symbols.
- Glossary, lists and defines common routing terminology.

Conventions

The following fonts, characters, and styles have specific meanings throughout this manual.

- Courier font identifies text that you type exactly as shown, such as router command names, keywords, and other syntax elements.

  For example:

  ```
  (average_pair_length [on | off])
  ```

- Italic type identifies menu paths or dialog box buttons in the graphic user interface (GUI), titles of books, and is also used to emphasize portions of text.

  For example:

  Choose *File – Quit.*

  Click *Apply.*

  To learn how to route a design using the router, see the *Allegro PCB Router Tutorial*.

- Italicized words enclosed in angle brackets (<>) are placeholders for keywords, values, filenames, or other information that you must supply.

  For example:

  <*directory_path_name*>::= <*id*>

- References to keys on your keyboard and mouse buttons are enclosed in square brackets. [Shift] refers to the Shift key.

Where to Find Additional Information

To access additional technical documentation from within the router user interface, display the online Help page by choosing *Help – Product Help* from the main menu.

How to Contact Technical Support

If you have questions about installing or using the router, contact the Cadence Online Support

1

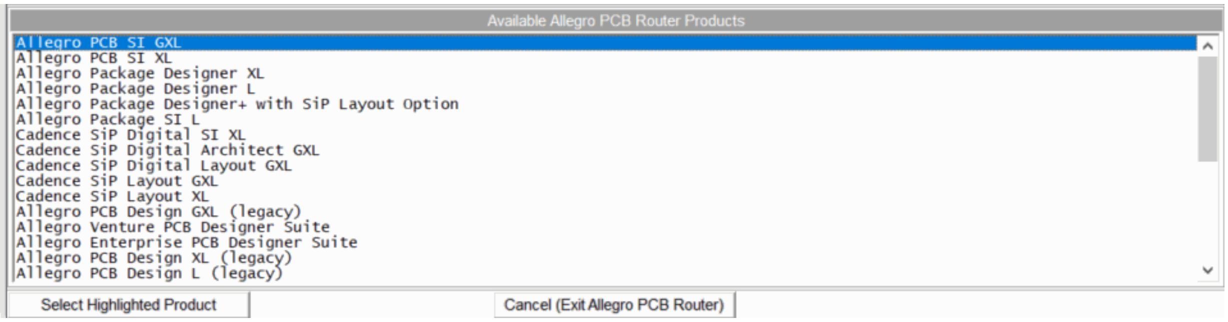
Getting Started

In this chapter . . .

Understanding Licensing

Overview

When you click the *Start Allegro PCB Router* button from the Startup dialog box, the licensing system first determines if you have multiple product choices available to you based on the feature strings in your license file. If your license file specifies only one product, the router starts immediately. Otherwise, the *Product Selection* dialog box displays the Cadence Allegro PCB and Package products available at your site.


Available Allegro PCB Router Products
Select Highlighted Product
Cancel (Exit Allegro PCB Router)

```
Allegro PCB SI GXL
Allegro PCB SI XL
Allegro Package Designer XL
Allegro Package Designer L
Allegro Package Designer+ with SiP Layout Option
Allegro Package SI L
Cadence SiP Digital SI XL
Cadence SiP Digital Architect GXL
Cadence SiP Digital Layout GXL
Cadence SiP Layout GXL
Cadence SiP Layout XL
Allegro PCB Design GXL (legacy)
Allegro Venture PCB Designer Suite
Allegro Enterprise PCB Designer Suite
Allegro PCB Design XL (legacy)
Allegro PCB Design L (legacy)
```

Note: If a product option is not currently available at your site, contact your local Cadence sales representative for information regarding a software upgrade.

For details on starting the router in batch mode see Using the Router in the Design Process on page 25.

To start the router with the appropriate license

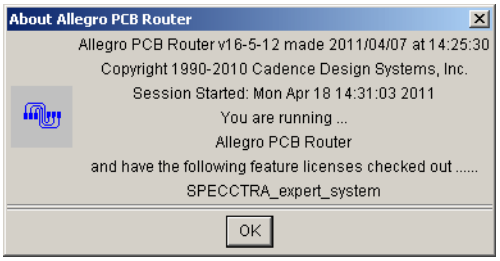
1. Select a product in the *PCB Router Product Selection* dialog box.
2. Click the *Select Highlighted Product* button.

   The appropriate license(s) are checked out and the router starts.

To verify the product you are running and feature license(s) you have checked out

1. Choose *Help – About Allegro PCB Router*

   The About Allegro PCB Router dialog box appears.



Allegro PCB Router v16-5-12 made 2011/04/07 at 14:25:30
Copyright 1990-2010 Cadence Design Systems, Inc.
Session Started: Mon Apr 18 14:31:03 2011
You are running ...
Allegro PCB Router
and have the following feature licenses checked out ......
SPECCTRA_expert_system

2. Within the dialog box, read the information that follows the string:

   ```
   You are running...
   ```

3. Click *OK*.

Router Features

The following table provides a brief description of router features.

Table 1-1 Features

Feature Name	Description
ViewBase	Base option
EditBase	Base EditRoute option
RouteBase	Base AutoRoute option
IPlaceBase	Base EditPlace option
PlaceBase	Automatic Placement option
RouteADV	AutoRoute ADV option
RouteMVIA	MicroVia option
RouteDFM	AutoRoute DFM option
RouteFST	HighPerformance option for AutoRoute
EditFST	EditHighPerformance option for AutoRoute

Router Features and Enabled Commands

The following tables lists the router commands that are enabled by each feature.

Note: Items indented in the table columns are keywords and represent specific variations of the commands (left justified) in each column.

Table 1-2 Commands Enabled by ViewBase

assign_pin	smart	limit_crossing
assign_supply	snap	limit_vias
associate	via	limit_way
autosave	via_keepout	max_stagger
bestsave	wire	max_stub
bind_mouse_button	highlight	max_total_vias

- cct_mode
- change smd_escape
- change_width_by_rule
- check
  - all
  - place
  - route
- check_area
- checkmode
- component_pin_property
- component_property
- define
  - bundle
  - class
  - cluster
  - component
  - group
  - group set
  - keepout
  - net
  - net pins
  - noise_table
  - padstack
  - poly_wire
  - room
- defkey
- delete
- delete component
- if
- image_property
- image_pin_property
- keep_selected
- layer_property
- license usage
- lock
- mode
  - measure
  - delete keepout
  - edit fence
  - edit keepout
  - edit region
  - edit ruler
  - highlight component nets
  - merge keepout
  - select
- net_property
- order
- place_rule
- place_status_file
- plc_post_process
- protect
- quit
- read
  - colormap
  - floor_plan
  - keepout

  - pin_width_taper
  - power_fanout
  - redundant_wiring
  - reorder
  - spiral_via
  - staired_via
  - time_length_factor
  - tjunction
  - width
- select
- set
- setexpr
- set_focus
- setup_check
- sh
- show
  - component_labels
  - interchange
  - unroutes
- skill_cmd
- skill_mode
- splash
- status_fileViewBase
- stop
- undo
- unfix
- unit
- unprotect

delete wirebond	placement	unselect
did_file	routes	view
direction	wire	vset
disassociate	rebuild_power_net	while
do	redo	wildcard
edit_wires_for_placement	repaint	write
evaluate	report	write colormap
fence	room_rule	write environment
fix	rule	write keys
forget	allow_redundant_wiring	
grid	clearance	
place	inter_layer_clearance	
place_major_factor	junction_type	
route_major_factor	limit_bends	

Table 1-3 Commands Enabled by EditBase

mode (other routing modes)	relocate	trade
place	sequence	unlock
push	swap	unplace

Table 1-4 Commands Enabled by RouteBase

bus	recorner	restricted_layer_length_factor
center	reduce_padstack	seedvia
clean	route	smart_route
cost	rule	sort
critic	effective_via_length	spread
fanout	length_amplitude	tax
filter	length_factor	testpoint

limit	length_gap	

Table 1-5 Commands Enabled by IPlaceBase

align	work_origin	mode (other placement modes)
change	define place_boundary	
component_image	density_analysis	

Table 1-6 Commands Enabled by PlaceBase

apply_small_comp_pattern	change align_base	interchange
assign_small_comp_pattern	form_cluster	mode swap
autodiscrete	high_speed	secondary_connection
autorotate	initplace	site

Table 1-7 Commands Enabled by RouteADV

check	define (with virtual_pin)	rule
use_layer	define	region
use_via	class_class	via_at_smd
circuit	layer_noise_weight	clearance buried_via_gap
use_layer	pair	layer
use_via	region	wirebond
define (with layer_rule)		

Table 1-8 Commands Enabled by RouteMVIA

mode	report	stack_via
change_via	stack_via_depth	stack_via_depth
rotate_via	rule	set microvia on

Table 1-9 Commands Enabled by RouteDFM

miter	rule	smart_route
	clearance testpoint	auto_miter
	testpoint_rule	auto_testpoint

Table 1-10 Commands Enabled by RouteFST

change min_shield	sample_window	max_segment
check (with include/exclude miter)	shield	max_noise
circuit	switch_window	parallel_noise
max_delay	relative_delay	parallel_segment
match_fromto_delay	relative_group_delay	saturation_length
match_fromto_length	relative_group_length	shield_gap
match_group_delay	relative_length	shield_loop
match_group_length	define	shield_tie_down_interval
match_net_delay	class_class	shield_width
match_net_length	layer_noise_weight	tandem_noise
max_length	pair	tandem_segment
min_length	region	tandem_shield_overhang
max_restricted_layer_length	recorner	set
max_total_delay	round	tandem_depth
max_total_length	rule	shield
min_delay	defer_shield	unmiter
min_total_delay	length_amplitude	unprotect
min_total_length	length_factor	all testpoints
priority	length_gap	

Using the Router in the Design Process

How it Works

The router uses ShapeBased technology to automatically or interactively place and route dense (PCB, package, and MCM) designs. Circuit elements are modeled as basic geometric shapes. Each shape can have rules associated with it that enforce design constraints, such as component spacing and orientation, wire width and clearance, timing, noise, and crosstalk.

Following are some benefits of ShapeBased technology.

- Accurately models pads, pins, wires, and vias in a shape database instead of a memory consuming grid map by minimizing memory requirements.
- Uses the exact dimensions of shapes by maximizing the use of available space and accommodates mixed pin pitches and mixed size components.
- Supports complex hierarchical design rules, thus improving manufacturability.
- Routes gridless or with submil wire and via grids by maximizing the use of the design routing area, which can reduce signal layers.

Because the router associates design rules with geometric shapes, you do not have to manage and apply rules through traditional grid-mapping techniques. The router places and routes without grids or with sub-mil grids, avoiding the massive memory requirements of traditional grid-mapped tools.

Many grid-based autorouters attempt to complete all connections in each routing pass until the design is routed completely. These autorouters prohibit crossover and clearance conflicts.

The router uses a different approach, called *adaptive routing*. The autorouter attempts to connect all wires in the first routing pass by allowing crossover and clearance conflicts. During each additional pass, the autorouter reduces conflicts by using its intelligent rip-up-and-retry and push-and-shove algorithms.

With each pass, the router gathers information and “learns” about the problem areas where conflicts exist to eliminate them and completely route the design.

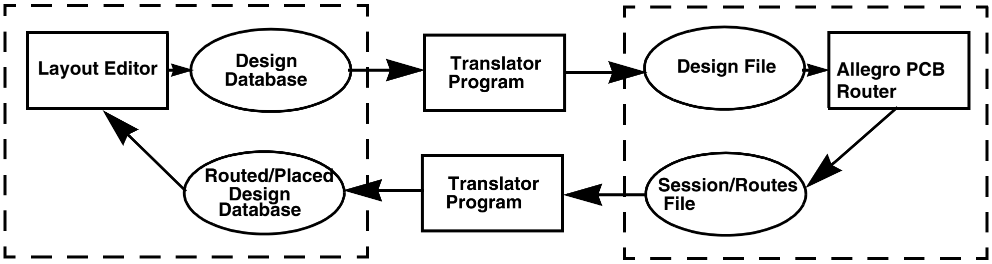
Although it sometimes uses a large number of routing passes, the autorouter usually achieves a high completion rate in a short period of time because it uses the conflict information from each pass to achieve an overall solution.

Understanding the Workflow

The router works with your layout editor to produce a placed and routed design. After laying out your physical design, you translate the data to a router Design file. This file contains all the information router needs to place and route the design.
Any design rules you set in your layout editor are transferred to the Design file.

After placing and routing, you translate a Routes file or a Session file.
A translator merges the routing information with your original design layout files, creating a placed and routed design that updates the Allegro database. The flow from your layout editor to the router, and back again, is illustrated in the following diagram.

Figure 1-1 The Router Workflow



The router Design file is an ASCII text file that contains a netlist, boundary outlines, keepout areas, and all component libraries required to place and route your design. It also contains any rules you set in your layout editor to constrain placement and routing.

Note: If you don’t plan to place components using the router, you must create a design layout with placed components using your layout editor before translating the design to a router Design file.

Design Data Files

Before you can begin work in the router, you must read in design data generated from either a host CAD system or from a previous router session. This design data is comprised of one or more of the files described in the following table.

Table 1-11 Design Data Files

Input File	Description	Read during . . .
Design file	A Design file (.dsn) contains design data generated by your host CAD system. It includes complete information about components, connectivity (netlist) and design constraints. The Design File is the starting point for all work in the router.	Startup
Floor Plan file	A Floor Plan file (.pln) contains design data generated by the router. It includes information about how components should be grouped together for optimal placement. To create floor plan data in the router, you must first read in a Design File generated from your host CAD system.	Session
Placement file	A Placement file (.plc) contains design data generated by the router. It includes information about component placement, such as x,y coordinate location, rotation, layer association, etc. To create placement data in the router, you must first read in a Design File generated from your host CAD system.	Startup -or- Session
Routes file	A Routes file (.rte) contains design data generated by the router. It includes information about wiring that has been routed in the router. To create routing data in the router, you must first read in a Design File generated from your host CAD system.	Startup -or- Session

Input File	Description	Read during . . .
Session file	A Session file (.ses) contains design data generated by the router. It includes a nearly complete record of all commands and activities that were executed during a design session.	Startup
Wires file	A Wires file (.w) includes wire and via information that you saved from a previous router session.	Startup -or- Session

Note: You can also read a Do file to execute a sequence of commands, or a Did file that was used to automatically record commands used during a previous router session. However, you cannot perform meaningful design work with these unless you also read in their associated Design files.

For further details, see Understanding the Batch Script on page 34 and Using Did Files on page 59)

Reading Data Files at Startup

To read data files at startup

1. Launch the router. See Starting an Interactive Router Session on page 30 for details.

   The Startup dialog box appears as shown in Figure 2-2 on page 31.

2. If you know the names of the data files you want to load, enter them in the appropriate text boxes and proceed to step 3.

   -or-

   a. Click the *Browse* button in a text box.

      A File Browser appears.

   b. Choose a file of the appropriate type in the File Browser window, then click *Open*.

      -or-

      Specify the file by entering its path and filename, then click *Open*.

   c. Repeat these steps until the names of all desired data files are entered in the dialog box.

3. Click *Start Allegro PCB Router*.

   The router starts and loads the specified data files.

Reading Data Files during a Session

To read data files during a router session

1. Choose *File – <mode>* to switch to the appropriate (Place or Route) mode.
2. Choose *File – Read – <data file type>.*

   A Read dialog box appears.
3. If you know the file/pathname, enter it in the text box and proceed to step 4.

   -or-

   a. Click the *Browse* button.

      A File Browser appears.

   b. Choose a data file of the appropriate type in the File Browser window, then click *Open*.
4. Click *Apply* or *OK* in the Read dialog box.

   The data file is loaded into the router.

Starting an Interactive Router Session

Before you start the router, you need to translate the layout design from your CAD system to a router Design file. See the router translator manual provided with your layout system for information about translating layout data from your CAD system.

Note: Before you can run the router on a Windows system, you must attach a software license key to a parallel port or install the FLEXlm license server.

To start the router, you can do one of the following:

- Open the Startup dialog box
- On UNIX-based systems, use the `specctra` or `allegro_pcb_router` command and switches. These are useful if you know the options you want to use and the paths and filenames you want to specify.
- Run a batch script. This is useful if you want to run several router sessions unattended.

Starting PCB Router

On a UNIX system

➡ Enter `allegro_pcb_router &` in a shell window

or

➡ Enter `specctra &` in a shell window

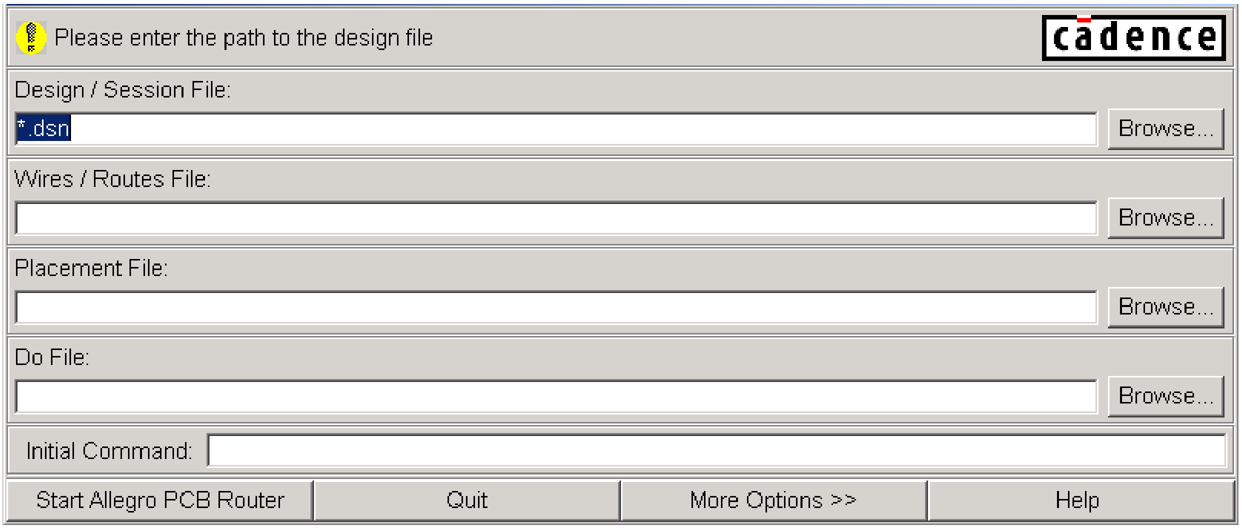
On a Windows system

➡ Use the Start menu, for example:

Click *Start – All Programs –* Cadence *<release number> – Allegro X Products – PCB Router*

When you start a session, the Startup dialog box, shown in the following figure, is displayed. From here you can specify startup options and files.

Figure 2-2 Startup Dialog Box


Please enter the path to the design file

Design / Session File: *.dsn

In the Startup dialog box, enter a filename in the filename data entry box, or *Browse* to locate a file. If you enter a filename, include the full path if the file is not located in the current directory. See File Naming Conventions on page 37 for further details.

The Startup dialog box initially displays only basic startup options. To see additional options, click *More Options*. The dialog box expands to display the additional Startup options.

In the *Startup* dialog box:

1. Enter the name of the Design file or Session file in the Design / Session data entry box.
2. Set the startup options you want to use. Click the *More Options* button if you want to use additional startup options. See Startup Options on page 326 for further details on router startup options.
3. Click the *Start Allegro PCB Router* button.

Using Command-line Switches

You can use command line switches to start a router session. Start the router from the command line to bypass the Startup dialog box. To use this method, you must know the paths and filenames for all the files you want to load, and you must know the startup switches you want to use.

Router Command Line Syntax

The PCB Router command line syntax is:

```
specctra [<design_or_session_file>] {[< switch>]}
```

where

`<design_or_session_file>` is the name of the Design file if you are starting a new session, or the name of the Session file if you are restarting with data from a previous session.

`and`

`<switch>` is one or more of the optional router command line switches.

You can use switches to specify a Routes, Session, or Wires file, a Placement file, a Do file, and to set other start up options. See Startup Options on page 326 for further details on router command line switches.

If

- A file is not located in the current directory, you must include a complete path with the filename. See “File Naming Conventions” on page 37 for further details.
- The router `bin` directory is not included in your path variable, you must include the full path name with the `specctra` command.

Running the Router with a Batch Script

You can use a batch script to run several router sessions in sequence, unattended. At the start of each session, the router reads a Design file and a Do file (see “Understanding the Do File” for more information). To run the router batch sessions, prepare the following files:

- Design file for each session
- Do file for each session
- Batch script

Understanding the Do File

A Do file is a text file that contains a list of autorouter commands. It is a script that controls the autorouter.

Following is an example of a Do file.

```
# Lines beginning with '#' are comments

# General purpose do file

# Initial Commands

bestsave on bestsave.wre

status_file route.sts

unit mil

grid smart (wire 1) (via 1)

# Standard Routing Commands

smart_route
```

The order of commands in a Do file is important because the autorouter executes each command *sequentially*. For example, you would not want to route the design before you set rules, such as clearance and width rules, that you want the autorouter to follow.

See Editing the Do File on page 193 for details on creating a Do file.

See Using Do Files on page 63 for details on running Do files.

Understanding the Batch Script

The batch script controls each session with a `specctra` command that includes a Design filename, a Do filename, and any switches you want to use.

Creating a Batch Script

You use a text editor to create a batch script. A batch script contains a series of allegro_pcb_router commands. Each line in a batch script starts a new session. The `-do` switch specifies the Do file that contains commands to control the session.

The `-quit` switch is included with each `specctra` command to end that session after the last command executes in the Do file. Place the `-quit` switch at the end of each command line in the script to exit that session and start the next session.

If a file is not located in the current directory, include the correct path with the filename. The following example shows the contents of a batch script. It works for both UNIX and Windows systems.

```
specctra design1.dsn -do des1.do -nog -quit

specctra design2.dsn -do des2.do -nog -quit

spectra design3.dsn -do des3.do -nog -quit

specctra sample.dsn -do sample.do -nog -quit
```

The equivalent batch script for Windows systems is shown in the next example:

```
start /wait specctra design1.dsn -do des1.do -nog -quit

start /wait specctra design2.dsn -do des2.do -nog -quit

start /wait specctra design3.dsn -do des3.do -nog -quit

start /wait specctra sample.dsn -do sample.do -nog -quit
```

This example runs four sessions in sequence. The Design files and Do files are located in the current directory, and the bin directory is set in the path variable.

If the Design file and Do file are in the same directory, you can use the `$/` symbols on UNIX systems (`\$/` in a C shell) instead of retyping the entire path for the Do file. For example:

```
allegro_pcb_router my_project/project1.dsn -do \$/project1.do -quit
```

Use the $\ symbols on Windows systems. For example:

```
specctra my_project\project1.dsn -do $\project1.do -quit
```

If the location of the specctra executable is not included in your path variable, you must include the full path name with the `specctra` command.

Creating a specctra .ini File

If you are running Windows, you must create an ASCII file called `specctra.ini` in the `\Windows` directory to run the router in batch mode. This file must contain the following entries:

```
    [QA]

    ExitWindows = 1
```

`ExitWindows` is a variable that controls whether Windows exits to DOS when you quit the router. The following values can be used.

Value	Result
1	Windows exits to DOS after the router quits.
0	Windows does not to exit to DOS after the router quits.

Important

> If you create a router.ini file and you use the router interactively, you will exit Windows when you quit the router. If you want to use the router without running in batch mode, rename the `specctra.ini` file, or edit this file and set `ExitWindows = 0`.

Running a Batch Script

Before running a batch script on a UNIX system, use `chmod` to make the script executable.

```
chmod +x <filename>
```

To start a batch script on a UNIX system

➤ Enter the script name at the command prompt

To start a batch script on a Windows system

1. Open a Command Prompt window.
2. Change to the directory where the batch script is located and enter the script name.

File Naming Conventions

You must observe standard file naming conventions for the system you are using. By default, the router looks for files you specify in the current directory (the directory where you started the router). To specify a file in a different directory, you must include its directory path and filename.

Filenames used in a router command can be preceeded by `$/` for UNIX files or `$\` for Windows files as a shorthand way to denote the design directory path as a prefix to a filename. For example, to save the current routed wires in a Wires File:

- On a UNIX platform, enter

  ```
  write wire $/ mydesign.w.
  ```

  If the `design` directory is `/home/user/test`, you write the file as

  ```
  /home/user/test/mydesign.w.
  ```

- On a Windows platform, enter

  ```
  write wire $\mydesign.w.
  ```

  If the `design` directory is `\home\user\test`, you write the file as

  ```
  \home\user\test\mydesign.w.
  ```

In addition, you can use the following variables in commands to denote the design file base name, filename, and title:

- $dsn, which denotes the base name of the design file
- $designFile, which denotes the filename of the design file
- $designID, which denotes the design identifier that follows the PCB keyword at the beginning of the design file.

Note: When you use the `allegro_pcb_router` command to start a session in a UNIX C shell (on the command line or in a batch script), you must use \$/ to denote the design directory.

2

Working with the Router

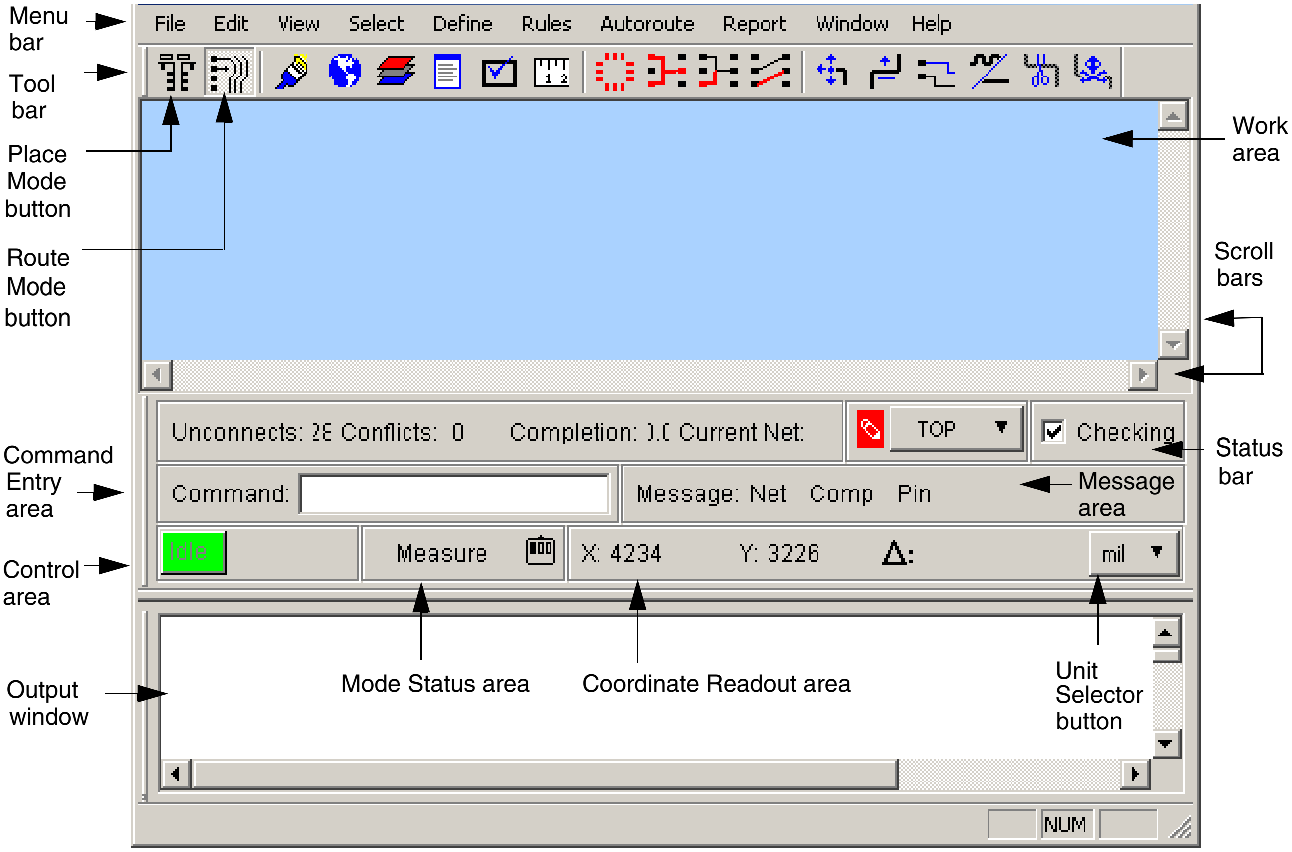
In this chapter . . .

Understanding the Graphic User Interface

The major features of the graphic user interface are labeled in the following figure and described in Table 2-1 on page 40.

Note: The areas of this GUI image have been re-sized for illustrative purposes and do not reflect the actual (default) display sizes of these areas in the graphic user interface.

Figure 2-1 The Router GUI



Unconnects: 2E Conflicts: 0 Completion: 0.0 Current Net:

Command:

Message: Net Comp Pin

X: 4234 Y: 3226 Δ:

Table 2-1 Graphic User Interface Features

Feature	Purpose
Command Entry area	Enables command entry from the keyboard.
Control area	Enables monitoring of the router operating status and control or interrupt of commands.

Feature	Purpose
Coordinate Readout area	Displays X, Y coordinates as you move the pointer in the work area.
Menu bar	Accesses menus.
Message area	Displays warning and error messages as well as prompts and other information about current operations.
Mode Status area	Displays the current left mouse button mode.
Output window	Displays a transcript of your work session. For further details, see “Command Echoing” on page 57.
Place button	Accesses the place environment tool bar.
Route button	Accesses the route environment tool bar.
Scroll bars	Pans the work area left, right, up and down.
Status bar	Displays information such as number of passes completed, number of unroutes, number of violations, and the percentage of routes completed.
Title bar	Displays the router title and version number and the design path and filename.
Tool bar	Executes frequently used commands.
Unit Selector button	Indicates and changes the current measure unit.
Work area	Displays the design.

Switching Between Routing and Placement

A design session starts in the Routing environment. You can switch to the Placement environment by:

- clicking the Place Mode icon on the Tool bar.
- choosing *File – Placement Mode* from the menu bar.

From the Placement environment, you can switch to the Routing environment by:

- clicking the *Route Mode* icon on the Tool bar.
- choosing *File – Routing Mode* from the Menu bar.

In a Do file, you can use the `application_mode` command to switch between the routing and placement environments (see Understanding the Batch Script on page 34 for more information).

For example:

```
application_mode placement
application_mode routing
```

Using the Mouse

The router GUI uses all three mouse buttons. Throughout this book, the mouse buttons are referred to as:

- [LB] left mouse button
- [MB]middle mouse button
- [RB] right mouse button

Note: If you have a two-button mouse, press `[ALT]` and `[RB]` simultaneously when you see `[MB]`.

You can use the mouse for:

- Zooming
- Panning
- Viewing the entire design
- Setting the left-button mode
- Measuring distance
- Changing the measurement unit

Zooming

Using the middle mouse button, you can zoom in to magnify a small region of the design or zoom out to see more of the design. You can zoom in or out without interrupting a router operation.

Note: Another way to zoom in and out is with menu commands. Click *View – Zoom In* or *View – Zoom Out*. If you accidentally zoom in or zoom out, you can return to your previous view by clicking *View – Zoom – Previous*.

To zoom in

1. Move the pointer to the lower left corner of the region you want to magnify.
2. Drag [MB] diagonally in an upward direction to enclose the region.

   A bounding rectangle shows the zoom region. Before you release `[MB]`, you can move the pointer and adjust the rectangle to enclose the exact region you want to magnify.

3. Release [MB] to zoom in.

To zoom out

1. Move the pointer to any location in the work area.
2. Drag [MB] diagonally in a downward direction.

   As you drag [MB] to zoom out, you'll see two rectangles. An inner rectangle represents the current view. It is surrounded by an outer rectangle that changes size as you drag the pointer.

   The relative size of the outer rectangle to the inner rectangle indicates how far you zoom out when you release [MB]. For example, if you adjust the size of the outer rectangle so that it is about twice the size of the inner rectangle, you zoom out by a factor of about two to one.

3. Release [MB] to zoom out.

Panning

You pan your view of the work by clicking [MB] at the location you want as the new center-of-view. You must click [MB] without dragging the pointer. You can pan without interrupting a router operation.

To pan the work area

1. Move the pointer to the location you want as the new view center.
2. Click [MB] to change the view center to the location under the pointer.

   You can also pan your view of the work area by dragging the screen image to a new center-of-view.

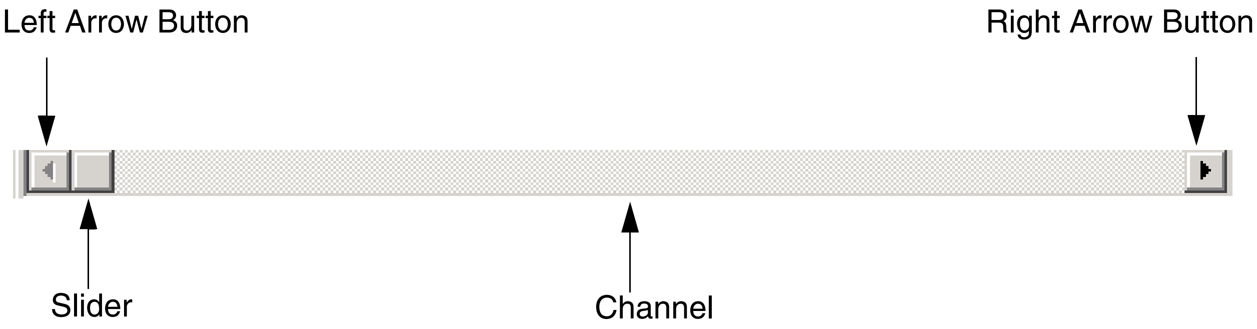
3. Move the pointer into the screen image.
4. Press [MB] with [Shift] and move the pointer to new destination point.
5. Release both [MB] and [Shift].

   You can use scroll bars instead of [MB] to pan. The horizontal scroll bar pans the work area view left and right, and the vertical scroll bar pans the work area view up and down.

   Each scroll bar consists of two arrow buttons, a slider, and a channel through which the slider moves.

You can use the slider button to quickly pan the work area view. When you drag the slider, the work area pans in the same direction that you move.

Figure 2-2 Scroll Bar Components



The arrow buttons at either end of a scroll bar pan the work area in small increments. Click the button to pan one increment in the direction of the arrow, or click and hold to continually pan the view.

You can also click anywhere in the channel to pan one screen at a time.

Viewing the Entire Design

You view the entire design by dragging [MB] horizontally. When you drag the pointer horizontally, a single fixed-size rectangle displays before you release [MB]. If you don't drag the pointer horizontally, you might see double rectangles that indicate you are about to zoom out, or you might see a single rectangle that stretches as you move the pointer and indicates you are about to zoom in.

To view the entire design

1. Move the pointer into the work area.
2. Drag [MB] horizontally.
3. Release [MB] to view the entire design.

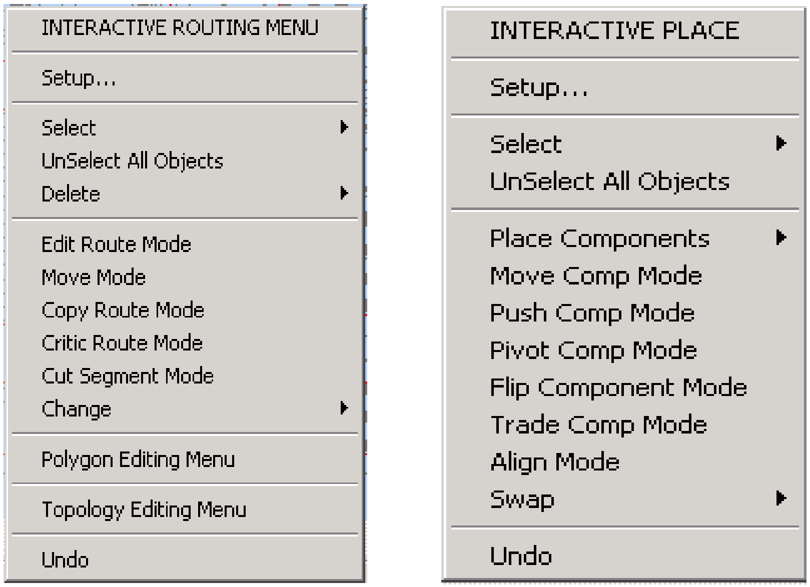
Note: You can also view the entire design by clicking the globe icon in the toolbar, or by clicking *View – Zoom – All.*

Setting the Left-button Mode

You can set the left mouse button to one of several modes. The [LB] modes that are available to you depend on whether you select the Place or Route environment. The [LB] mode determines the action that results when you use the [LB] in the work area.

You can select a [LB] mode by using the right button menu. If you move the pointer into the work area and press the right mouse button, a popup menu includes several [LB] mode choices. You can also set some common [LB] modes, shown in Figure 3-3, by clicking a button on the tool bar.

Figure 2-3 The Place and Route Options



To set the [LB] mode

1. Move the pointer to any location in the work area.
2. Press [RB] to display the right-button menu.
3. Slide the pointer to your choice on the menu and release [RB].

Measuring Distance

The router starts with the [LB] mode set to Measure. You use the measure mode to extract coordinate and keepout information about component pads or pins, determine the X,Y coordinate of a single point, and measure distance between two points.

When you click a single point in the work area, the X,Y coordinates display in the coordinate readout area. If you click a wire, keepout, component pad or pin, information about that object displays in the router Output window.

When the [LB] mode is measure, and you drag `[LB]` between two points in the work area, the distance between the start point and end point displays in the coordinate readout area to the right of the delta symbol and in the output window.

To measure distance

1. Check the mode status area.

   If Measure is not displayed, use the tool bar to set the Measure mode.

2. Move the pointer to the start point.
3. Click, hold, and drag the pointer to the end point and release [LB].
4. Read the measured distance displayed beside the delta symbol.

Changing the Measurement Unit

The unit button displays your current measurement unit. If you click the unit button, a popup menu displays five choices:

- inch
- mil (1/1000th of an inch)
- cm (centimeter)
- mm (millimeter)
- um (micron)

When you change the measurement unit, you change the unit in which dimensions and coordinates display, the units used in reports, and the units used when you enter commands from the keyboard or from a Do File (see Understanding the Batch Script on page 34 for more information).

Using the Toolbars

The buttons on the toolbars provide quick, easy access to the most commonly used commands.The toolbar display is unique to the mode that you are working in. The buttons themselves are organized into the following functional groups:

- Viewing operations
- Selection operations
- Edit operations

Figure 2-4 Route Toolbar

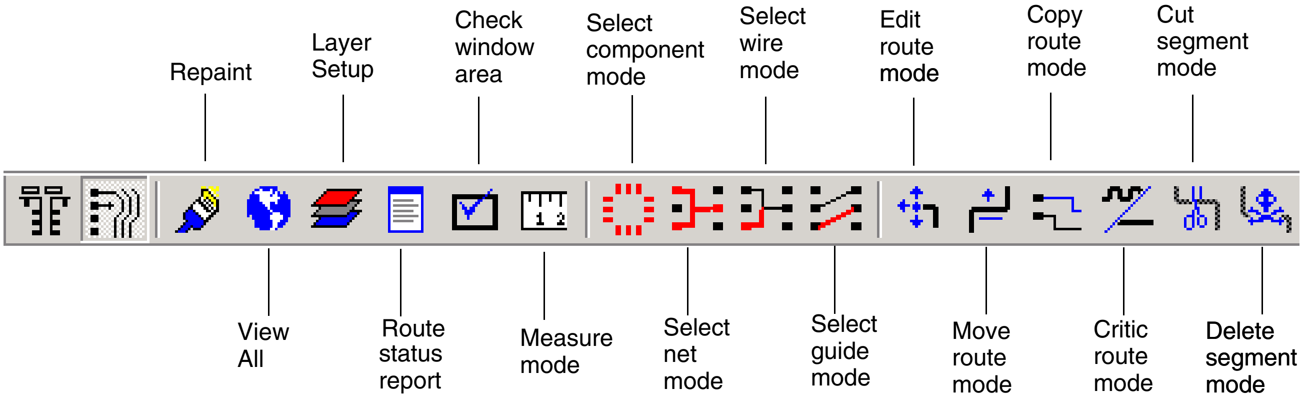
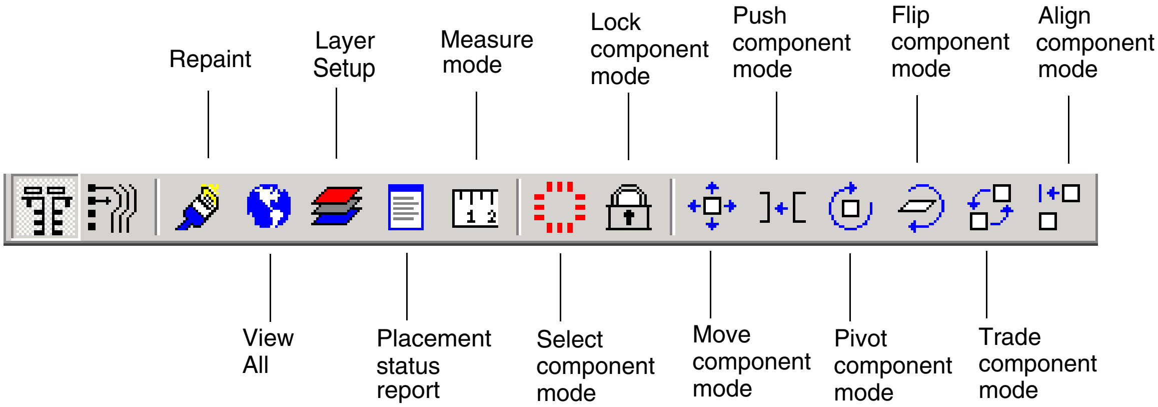


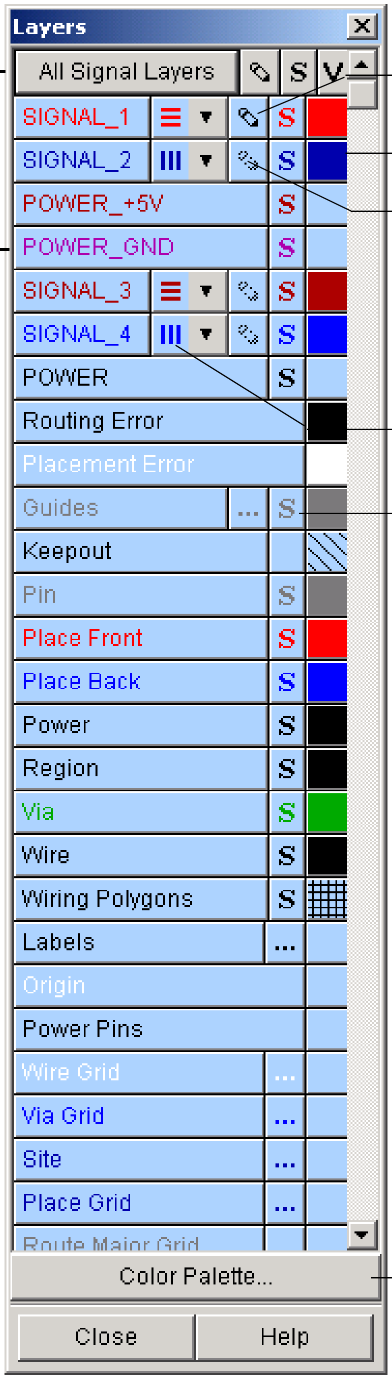
Figure 2-5 Place Toolbar



Using the Layer Dialog Box

You choose *View – Layers* to open the Layer dialog box as shown in Figure 3-5. The Layer dialog box controls layer selection, layer routing direction, layer visibility, and interactive editing between primary and secondary layers.

Figure 2-6



View all signal layers button

Layer ID button

Primary layer button (pencil bold)

Layer view button

Secondary layer button (pencil dimmed)

Selection and routing direction button

Object selection control button

Open Color Palette Dialog Box

Each row in the Layer dialog box represents a physical layer in the design or a graphic layer. Physical layers appear in the structure section of the Design File. Graphic layers are created by the router.

Layer Dialog Box Buttons

The router controls layer visibility in three ways. You can:

- Click the All Signal Layers button to display all of the signal layers.
- Click the individual layer ID buttons to show or hide a layer.
- Click the individual layer view buttons to display a specific layer.

When you activate a layer, the *layer view* button changes to the layer color. When you deactivate a layer, the color of the *layer view* button for that layer changes to the background color of the dialog box.

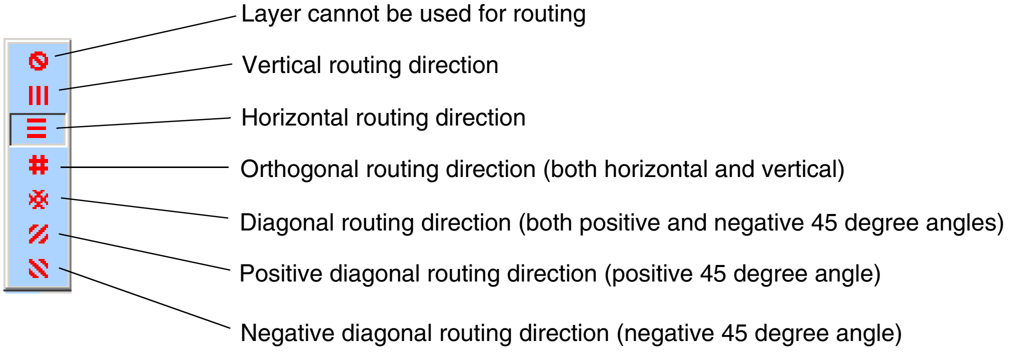
The object selection control button (S) controls selection by layer. If the S button is disabled, you cannot select objects on that layer. This is useful when shapes are overlapping.

The pencil icon sets the primary and secondary editing layers. The primary layer (pencil bold) is the active signal layer for interactive routing and editing. The secondary layer (pencil dimmed) is the layer that EditRoute switches to if you add a via.

Layer Selection and Direction Status Button

The *Selection and Routing Direction* button controls a layer's routing direction and whether a layer is selected. When you click it, a menu of icons is presented that enables you to change the layer's status, as shown in Figure 3-6.

Figure 2-7 Layer Selection and Direction Status


Layer cannot be used for routing
Vertical routing direction
Horizontal routing direction
Orthogonal routing direction (both horizontal and vertical)
Diagonal routing direction (both positive and negative 45 degree angles)
Positive diagonal routing direction (positive 45 degree angle)
Negative diagonal routing direction (negative 45 degree angle)

When a layer is unselected, the router does not use that layer unless you override the unselected status by a `use_layer` rule.

Color Palette Button

The *Color Palette* button opens the Color Palette dialog box, which includes a list of objects with their current patterns and colors, a color chip display, and a pattern chip display. The objects list contains the names of the signal and power layers defined in the Design File and the names of system layers. Each system layer provides a visual feature, such as guides, labels, or conflicts and rule violations.

To change the color assigned to an object, click the object name and click a color chip. To change the pattern assigned to an object, click the object name and click a pattern chip.

Reading the Status Bar

The status bar beneath the work area displays status information for the current routing or placement operations. At the beginning of a session, the status bar contains current routing status information and the Checking check box for interactive routing. Checking controls whether Design Rule Checking is enabled or disabled.

The status information changes depending on which environment you are using.

Autorouting Status

During autorouting operations, the status bar displays current routing status information and a convergence meter.

The Autorouting Status bar contains the information listed in the following table.

Table 2-2 Autorouting Status Bar Fields

This field . . .	Indicates . . .
Attempts	the total number of connection reroutes attempted.
Completion	the percentage of connections that are routed and not involved in conflicts.
Conflicts	the total number of crossing and clearance conflicts.
Pass	the number of the current or last routing pass completed and the total number of passes scheduled. The information is presented in an n/t format, where n equals the number of the current pass, and t equals the total number of passes scheduled.
Reroutes	the total number of connections scheduled for rip up and reroute.
Unconnects	the number of unrouted connections. Partially routed connections are included in the count, but connections that are attached to fixed nets are not included.

Viewing the Interactive Routing Status

During interactive routing operations, the status bar displays current routing status information and several interactive routing controls. Click *Rules – Check Rules* to update the number of conflicts.

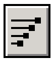
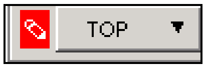
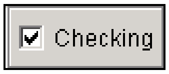
The Interactive Routing status bar contains the information listed in the following table.

Table 2-3 Interactive Routing Status Bar Fields

This field	Indicates . . .
Completion	the percentage of connections that are routed and not involved in conflicts.
Conflicts	the total number of crossing and clearance conflicts.
Current Net	the name of the current net being routed or the name of the last net you routed. Current Net also displays the net name associated with a shape, when the cross-hairs pass over a shape in Measure mode.
Unconnects	the number of unrouted connections. Partially routed connections are included in the count, but connections that are attached to fixed nets are not included.

The Interactive Routing status bar also contains the controls listed in the following table.

Table 2-4 Interactive Routing Status Bar Controls

Button	Function
	Displays the Set Via Pattern dialog box, which controls the via patterns used when you route multiple connections simultaneously
	Identifies the name and color of the current primary routing layer. Clicking on this button enables you to change the primary routing layer.
	Controls whether you can create width and clearance violations while you route connections. If this control is enabled, you cannot create these types of violations.

Viewing the Placement Status

During interactive or automatic placement operations, the status bar displays current placement status information.

In interactive modes, when you move, pivot, flip, or trade components, the status bar displays the component reference designator (Ref) and the change in Manhattan length. When you pivot a component, the status bar also displays the change in rotation (DR).

The Placement Status bar contains the information listed in the following table.

Table 2-5 Interactive Placement Status Bar Fields

This field	Indicates . . .
DR	the angle of rotation, in degrees, when you pivot a component during interactive placement operations.
Locked	the number of locked components.
Placed	the number of both large and small components placed inside the design boundary compared to the total number of components. The information displays in an n/t format, where n equals the number of placed components and t equals the total number of components.
Ref	the component ID (reference designator) of the component currently attached to the pointer during interactive placement operations.
Selected	the number of selected components.
Violations	the number of components that violate placement rules.

The Interactive Placement status bar also contains the controls listed in the following table.

Table 2-6 Interactive Placement Status Bar Controls

Button	Function
 ☑ Checking	Controls whether you can create placement rule violations while you place or relocate components. If this control is enabled, you cannot create placement rule violations.

Using Commands

You control the router and other features by using commands. You initiate these by:

- Clicking toolbar icons
- Choosing commands from pulldown or popup menus
- Entering commands from the keyboard or by running a Do file.

Some menu commands open dialog boxes. When a dialog box opens, edit it as needed and click *Apply* or *OK* to run the command. Dialog boxes with *Close* buttons perform immediate actions when you use their controls. When you are ready to close the dialog box, click *Close*. Click *Help* if you want information about the dialog box.

You can enter keyboard commands interactively by typing them in the command entry area or by pressing predefined shortcut keys in the work area. You can also use a text editor to enter commands in a Do File, and run the Do File during a session. You also use Do Files when you run in batch mode.

Pausing and Stopping Commands

The button in the *control area* at the bottom left corner of the session window indicates the status of automatic command operations. When the tool is waiting for a command, the word *Idle* appears on this button.

The word *Form* appears on the button when you open a dialog box that must be closed before you can perform other operations.

When the tool is performing an operation that you cannot interrupt, the word *Busy* appears on the button.

When the tool is performing an operation that you can interrupt, the word *Pause* appears on the button. If you click *Pause*, the tool pauses the operation, the *Pause* button disappears, and the *Continue* and *Stop* buttons appear.

- Click *Continue* to proceed with the current operation
- Click *Stop* to cancel the current operation

If you click *Stop* while running commands from a Do File, the tool cancels the current operation and exits the Do File without running the remaining commands.

Note: You can perform viewing tasks such as zooming, panning, highlighting, repainting the work area, and displaying reports without clicking *Pause* and interrupting the current operation. However some commands, such as `define`, pause the operation and ask if you want to proceed. You must click *Continue* or *Stop* before you can perform one of these commands.

Command Echoing

The router uses two modes to control command echoing to the Output window.

- Verbose mode
- Quiet mode

As you execute menu or keyboard commands within the Graphic User Interface, *verbose mode* is used to echo (display) the command syntax in the Output window. However, upon detecting a Do file execution, the router temporarily switches to *quiet mode* until the Do file completes, then exits back to verbose mode.

Quiet mode simply means that commands are not echoed in the Output window. By default, this mode is in effect under the following conditions:

- Starting the router from the command line with a supplied Do file parameter (Unix only). For example:

  ```
  allegro_pcb_router design1.dsn -do des1.do
  ```

- Starting the router and entering a Do file name in the startup form.
- Running a Do file from the router command line using the `do` command. For example:

  ```
  do des1.do
  ```

Tip

It is possible to override quiet mode and use verbose mode for all future Do file executions by entering the following command at the router command line.

```
vdo -filename
```

Note: This action is irreversible and initially displays an error message stating that *filename* does not exist. However, it will safely turn on verbose mode for the duration of the session.

Specifying Verbose Mode for Do File Executions

For specific cases, you can specify that commands in a Do file be echoed to the Output window.

Starting the Router on Unix

You can specify that verbose mode be used when starting the router from the command line with a supplied Do file parameter by using the `-vdo` switch.

For example:

```
allegro_pcb_router design2.dsn -vdo des2.do
```

Note: Verbose mode is not available when invoking the router through the startup form.

Running a Do file from the router Command Line

You can specify that verbose mode be used when running a Do file from the router command line using the `vdo` command.

For example:

```
vdo des3.do
```

Note: When running nested Do files (one Do file containing a call to run another Do file), *each file* runs according the documented use of the `do` or `vdo` command. This is also true for router startup Do files (Unix only).

Using Did Files

You can use a Did File to document the rules and other settings you assigned during a session. There are two types of Did Files you can create:

- A Session Did file, which is a record of commands you use during a session.
- A Rules Did file, which you can interactively control and edit using the Rules Did File Editor.

The following discussion explains how to use Session Did files. For information on using Rules Did files and the Rules Did File Editor, click *Edit – Rules Did File* in the Menu Bar, and click the *Help* button when the Rules Did File window opens.

By default, the tool creates a Session Did file in your `design` directory when you start a session. The default is a 10-character filename that consists of the month, day, hour, minute, and second when you started the session, with the .did extension. For example, 0608153428.did is a did file created June 8th at 15:34:28.

You can use the `-did` startup option to specify a different did filename or save the file in a different directory. For example

```
specctra -did design.did -design design.dsn
```

If you do not want to create a Did file for the session, you can use the `-nodid` startup option. For example

```
specctra -nodid design.dsn
```

You can suspend or resume Did file recording, close the current Did file, or open a new file, at any time during a session.

To see the current status of the Session Did file, click *File – Did File* to open the Did File dialog box.

- If no Did file is currently open, the message "Note: No Did file has been specified" appears in the message area below the Action controls.
- If a Did file is currently open, the message tells you the filename and whether it is active (the tool is currently recording commands) or inactive (recording has been suspended).

Only one Session Did file can be open for recording at a given time. If you open an existing file, the tool overwrites the file.

When you resume recording in an inactive file, the tool appends the commands in the file. However, if you open another file, the tool closes the active or inactive file.

To open a Did File for recording

1. Click *File – Did File*

   The Did file dialog box opens. If a Did file is currently open (active or inactive), its filename appears in the data entry box. Otherwise, a default filename appears. If this is the filename you want, proceed to step 4.

2. Click the *Browse* button

   A Browse dialog box opens.

3. Choose a file from the dialog box and click *Open*.
4. Click *Apply* or *OK*.

To suspend Did File recording

1. Click *File – Did File*

   The Did File dialog box opens.

2. Click *Suspend* if it is not already enabled.
3. Click *Apply* or *OK*.

To resume Did File recording

1. Click *File – Did File*

   The Did File dialog box opens.

2. Click *Resume* if it is not already enabled.
3. Click *Apply* or *OK*.

To end Did file recording and close the file

1. Click *File – Did File.*

   The Did File dialog box opens.

2. Click *Close*.
3. Click *Apply* or *OK*.

You can edit a Did file for use as a Do file. Retain only the commands that set rules or grids, define groupings such as classes or clusters, or perform other tasks you want to repeat in another session.

To edit a Did file during the session, close the file and open it in the Rules Did File editor. To edit a Did file after ending the session, use a text editor.

You can use *File – Execute Do File* or the `do` command to run a Do file anytime during a session (see Using Do Files on page 63 for more information).

Additionally,

- You can delete the current (active or inactive) Did file when you use *File – Quit* or the `quit` command to end the session.
- The Did file, along with the Design file, is also useful if you want to report a problem to Cadence.
- For general information about specifying filenames, see File Naming Conventions on page 37

Did File Considerations

The router observes the following rules when recording each command in your Did file:

- All correct command line and Command file entries are placed in the Did file exactly as they are entered. Incorrect commands are recorded as entered, followed by an error message that appears as a comment.
- All display commands (zoom, pan, view) except measure are translated to command line form and written to the Did File.
- All commands executed during pause mode are recorded as comments in the Did file.
- If a `do` command is executed, it is recorded as a comment in the Did file, since all commands in a nested Do file are recorded as that file is executed (see Using Do Files on page 63 for more information).
- Menu selections that result in a mode change generate an entry in the Did file of the form:

  ```
  mode<mode_type>
  ```

  where

  ```
  <mode_type> ::= [ sel net | sel comp | sel wire | measure | fence ]
  ```

- Zoom and pan actions generate the following entries in the Did file:

```
zoom coord <vertex> <vertex>
```

- When layers are turned on or off from the layer panel, the `vset` and `repaint` commands are generated in this form:

  ```
  vset <layer_name> [on / off]

  repaint
  ```

  For example, the following command sequence in a Do file turns layer L1 on and layers L2, L3, and L4 off:

  ```
  vset L1 on

  vset L2 off

  vset L3 off

  vset L4 off

  repaint
  ```

  The screen isn't updated until the `repaint` command is executed.

- The `if` and `while` commands generate an entry in the Did file of the form:

  ```
  #if (<expression>) (

  (<command group>)

  #) endif

  # while (<expression>) (

  (<command_group)

  #) end while
  ```

Using Do Files

A Do file is a text file that contains a list autorouter commands that are executed sequentially. You can use a Do file to automate part or all of your session. You can run a Do file at any time during a session by clicking *File – Execute Do File* and specifying the filename. You can also use the `-do` startup option to specify a Do file when you start a session.

See Editing the Do File on page 193 for details on creating an autorouting Do file.

See also Appendix C in the *Allegro PCB Router Command Reference* for listings of sample Do files that you can customize.

To run a Do file during a session

1. Choose *File – Execute Do File*.

   The Execute Do file dialog box opens. If you know the name of the file, enter it in the dialog box and proceed to step 4.

2. Click the *Browse* button.

   A Browse dialog box opens.

3. Select a file in the dialog box and click *Open*.

   To specify a file in a directory other than the current directory, enter its path and filename.

4. Click *Apply* or *OK*.

You can also use the `do` command to run a Do file during a session. For example:

```
do rt_rules.do
```

For more information about running Do files during a session, see the `Do` command.

To run a Do file at the beginning of a session, use the `-do` startup option to specify the filename. For example:

```
specctra design.dsn -do rt_rules
```

Note that:

- You can use the `dofile_auto_repaint` option in the `set` command (or click *View – Do file Repaints*) to control whether the tool repaints the work area after operations performed by the commands in a Do file.

- By default, Do files are run in *quiet mode* in the router. This means that the commands in the Do file are not displayed in the Output window as it runs. For information about running Do files in *verbose mode* (commands display in the Output window), see Command Echoing on page 57.
- For general information about specifying filenames, see File Naming Conventions on page 37

Understanding File Saving

You can save the work you do in the router using various output files as described in the following table. See File Naming Conventions on page 37 for details about specifying filenames and directory paths for output files on your platform.

Table 2-7 Output Files

File	Description
Placement	A Placement file (.plc) contains design data generated by the router. It includes information about component placement, such as x,y coordinate location, rotation, layer association, etc.
Floor Plan	A Floor Plan file (.pln) contains design data generated by the router. It includes information about how components should be grouped together for optimal placement.
Routes	A Routes file (.rte) contains design data generated by the router. It includes information about wiring that has been routed in the router.
Wires	A Wires file (.w) contains wire and via information. Note: A Routes file is the suggested means for saving wiring for subsequent autorouting sessions.
Session	A Session file (.ses) contains design data generated by the router. It includes a nearly complete record of all commands and activities that were executed during a design session.
Rules	A Rules file (.rul) contains rules and user properties not included in the Session file. You can use this file as a Do file in the current session or in a future session to load these rules back into the router.

Saving a Placement File

Use a placement file to save the current component placement information for use in a subsequent session.

The placement file includes the X,Y coordinate location, and the side (mounting surface), rotation, physical and electrical properties, and lock status for each component in the design. This file contains the placement information that gets merged with your original design layout. Placement information is written for each instance of a component image.

Note: When you are placing a difficult design, you can apply different rules and constraints over several placement trials and save the results of each trial in a separate placement file.

See the procedure for *File – Write – Placement* in the *Allegro PCB Router Command Reference* for details on saving placement information to a file.

You can also use the `write` command to save a Placement file.

For example:

```
write placement

write placement place3.plc
```

Saving a Floor Plan File

Use a Floor Plan File to save cluster and room definitions for use later in the current session or a subsequent session.

The floor plan file is a text file that contains the data needed to define clusters and rooms. Room definitions include room rules, which set cluster and component assignments, and can also specify power net connections, height restrictions, and power dissipation limits.

See the procedure for *File – Write – Floor Plan* in the *Allegro PCB Router Command Reference* for details on saving floor plan information to a file.

You can also use the `write` command to save a Floor Plan file.

For example:

```
write floor_plan

write floor_plan fplan3.pln
```

Saving a Routes File

Use a Routes file to save routing information for use later in the current session, during a subsequent session, or when you translate routing information back to your layout system. The routes file contains the wiring data that you merge with your original design layout.

See the procedure for *File – Write – Routes* in the *Allegro PCB Router Command Reference* for details on saving routing information to a file.

You can also use the `write` command to save a Routes file.

For example:

```
write routes (include testpoints)
write routes (type protect)
```

Saving a Session File

Use a Session file to save placement, floor plan, and routing information from the current session. You can use the file to resume the session later or to translate the information to your layout system.

When you achieve satisfactory results, it is recommended to save the routing and placement information in a Session file before you exit the session. If you are saving only routing information, save it in a Routes file as an alternative to the Session file. You can also save just the session rules and processing commands separately in a Rules file.

See the procedure for *File – Write – Session* in the *Allegro PCB Router Command Reference* for details on saving session information to a file.

```
write session
write session session3.ses
write session session3.ses (comment my third session)
```

Saving a Rules File

The router does not save rules or user properties in the Session file or Routes files.
By default, the tool begins recording commands in the Rules Did File Editor when you start a session.

You can use *Edit –Rules Did File* to open the Rules Did File Editor window. Within this window you can edit recorded commands and control which types of commands are recorded. The tool stores the edits in a *temporary* file. You can then save the contents of this temporary file to a Rules file and load it back in later during the same session or a subsequent session as a Do file (See Understanding the Batch Script for more information).

You can also edit the Session Did file, using a text editor, and use it as a Do file. For more information about how to use Did files, see *File – Did File*.

See the procedure for *Edit– Rules Did File* in the *Allegro PCB Router Command Reference* for details on saving rules information to a file.

3

Setting Rules and Constraints

In this chapter . . .

Design Rules Hierarchy

You meet electrical and other design requirements by setting rules. These rules are often defined in your layout tool and translated for the router. You can override rules from your layout tool and define additional rules in a router session. You set electrical rules by using the `circuit` and `rule` commands. You can also set many of these rules using one of the `define` commands.

Note: Rules you set during a session override rules set in the Design file.

You can set global rules for the design or specific rules for objects or collections of objects. Rules are *hierarchical*. In other words, higher precedence rules always override lower precedence rules that apply to the same object. When multiple rules apply to the same connection, the tool follows the rule with the highest priority, called its *precedence level*. PCB rules (global rules for the entire design) have the lowest precedence level in the hierarchy.

For routing, you can set specific rules for the following:

- Layers
- Classes of nets
- Selected nets
- Individual nets
- Groups of nets
- Sets of groups
- Individual fromtos
- Padstacks
- Regions

For more information, see Routing Rules Hierarchy on page 78.

For placement, you can set specific rules for the following:

- Image sets
- Images
- Components
- Super clusters
- Rooms

- Image sets within rooms.

You can also set special edge to edge spacing rules for image families and individual images.

For more information, see Placement Rules Hierarchy on page 72.

Setting Placement Rules

You apply placement rules using the Rules menu or the `place_rule` command.

Note: Placement rules that you set in the session override corresponding rules assigned in the Design file.

In general, the placement rules you can set are:

- Minimum permitted spacing
- Permitted orientations
- Permitted placement sides
- Opposite sides constraints

Rule Checking

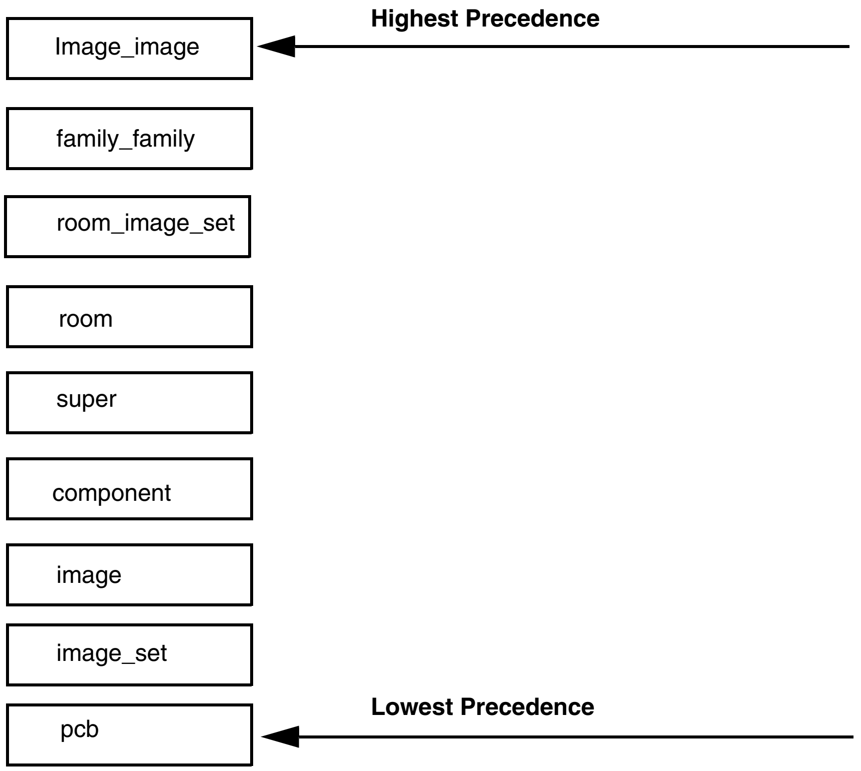
By default, the tool automatically performs design rule checking, and prohibits rule violations during automatic and interactive placement operations. You can disable rule checking for some interactive tasks, such as placing a connector over the design boundary. When you enable rule checking again, you should run a rules check to find any violations that occurred while checking was disabled.

Placement Rules Hierarchy

The tool applies placement rules according to the Placement Rules Hierarchy figure on page 73.

A pcb rule (global rule for the design) has the lowest precedence in the hierarchy. An image-to-image spacing rule has the highest precedence. Rules set at one level of the hierarchy override conflicting rules set at lower levels.

Figure 3-1 Placement Rules Hierarchy



You can apply rules to all the levels shown in the rule precedence. However, some rules only apply at specific levels, as shown in the following table.

Table 3-1 Placement Rules Application

Level	Rules that Apply
component	spacing, permitted orientations, permitted sides, opposite side
family_family	spacing
image	spacing, permitted orientations, permitted sides, opposite side
image_image	spacing
image_set	spacing, permitted orientations, permitted sides, opposite side
pcb	spacing, permitted orientations, permitted sides, opposite side
room	spacing, permitted orientations, permitted sides, opposite side
room_image_set	spacing, permitted orientations, permitted sides, opposite side

Level	Rules that Apply
super	spacing, permitted orientations, permitted sides, opposite side

Setting Spacing Rules

During automatic or interactive placement, spacing rules determine the minimum amount of space required between components, or between components and keepout areas. You can set spacing rules in the Design file or in the tool. Rules set in the tool override rules set in the Design file.

The following procedures in the *Allegro PCB Router Command Reference* explain how to set spacing rules.

- Setting global spacing rules
- Setting spacing rules for images sets
- Setting spacing rules for individual images and components
- Setting spacing rules for rooms and image sets in rooms
- Setting pad and body edge spacing rules for image families
- Setting pad and body edge spacing rules for individual images

You can set a single rule or separate object-to-object rules. You can set different object-to-object rules for the front and back sides of the design. For image set and room image set spacing, you can also set different object-to-object rules for SMD and through-pin components.

You can set a spacing preference for an automatic placement operation. The autoplacer follows the preference except when it conflicts with current spacing rules.

Use the `define cluster` command to set spacing rules for a super cluster.

Setting Orientation Rules

Orientation rules determine the component rotations permitted during automatic or interactive placement. You define component orientations in your layout system. They are translated into the library definitions of components in the Design file.

Use orientation rules when you are not sure how component orientations are defined. You should also use orientation rules if zero degree rotation is not the same for all components. You can set orientation rules in the Design file or in the tool. Rules set in the tool override rules set in the Design file.

The following procedures in the *Allegro PCB Router Command Reference* explain how to set permitted orientation rules in the tool:

- Setting global permitted orientation rules
- Setting permitted orientation rules for image sets
- Setting permitted image orientation rules for individual images and components
- Setting permitted orientation rules for rooms and image sets in rooms

You can set different orientation rules for the front and back sides of the design. For image set and room image set orientation, you can also set different rules for SMD and through-pin components.

You can set orientation preferences for an automatic placement operation. The autoplacer follows your preferences except when they conflict with current permitted orientation rules.

Use the `define cluster` command to set orientation rules for a super cluster.

Setting Permitted Side Rules

Permitted side rules determine which sides of the design you can use for automatic or interactive placement. You can set general rules or rules for specific components or areas of the design.

The following procedures in the *Allegro PCB Router Command Reference* explain how to set permitted side rules:

- Setting global permitted side rules
- Setting permitted side rules for image sets
- Setting permitted side rules for individual components or images

- Setting permitted side rules for rooms and image sets in rooms

You can permit placement on the front, back, or both sides of the design. For image sets and room image sets, you can set different rules for SMD and through-pin components.

You can set side preferences for an automatic placement operation.

Use the `define_cluster` command to set permitted side rules for a super cluster.

Setting Opposite Side Rules

Opposite side rules determine which components you can place front to back (at the same location on opposite sides of the design) during automatic or interactive placement operations. You can set general rules or rules for specific components or areas of the design.

The following procedures in the *Allegro PCB Router Command Reference* explain how to set opposite side rules:

- Setting global opposite side rules
- Setting opposite side rules for image sets
- Setting opposite rules for individual components or images
- Setting opposite rules for rooms and image sets in rooms

You can permit opposite side placement of large components, small components, or both. For image sets and room image sets, you can set different rules for SMD and through-pin components.

Note: Use the `define_cluster` command to set opposite side rules for a super cluster.

Setting Multiple Placement Rules

You can use the `place_rule` command to apply more than multiple rules. The following example applies spacing, side, and orientation rules to an image.

```
place_rule image PLCC144 (spacing 0.5 (type smd_pin)) (permit_orient
0 180)
```

Setting Image Pad Edge to Body Edge Spacing Rules

Pin to body spacing rules determine the minimum amount of space required between the pad or body edges of specific images during automatic or interactive placement.

You can set pad to body spacing rules in the Design file or in the tool. Rules set in the tool override rules set in the Design file.

The following procedures in the *Allegro PCB Router Command Reference* explain how to set pad to body spacing rules.

- Setting image to image spacing rules
- Setting image family to image family spacing rules

You can set a single rule or separate edge-to-edge rules. You can set different object-to-object rules for the front and back sides of the design.

Note: Use the `define cluster` command to set spacing rules for a super cluster.

Setting Routing Rules

You set routing, electrical, and design rules by using the `circuit` , `rule` and define commands. You can set specific rules for layers, classes of nets, individual nets, groups of nets, sets of groups, individual fromtos, padstacks, and regions. When multiple rules apply to the same connection, the router applies the rule with the highest precedence.

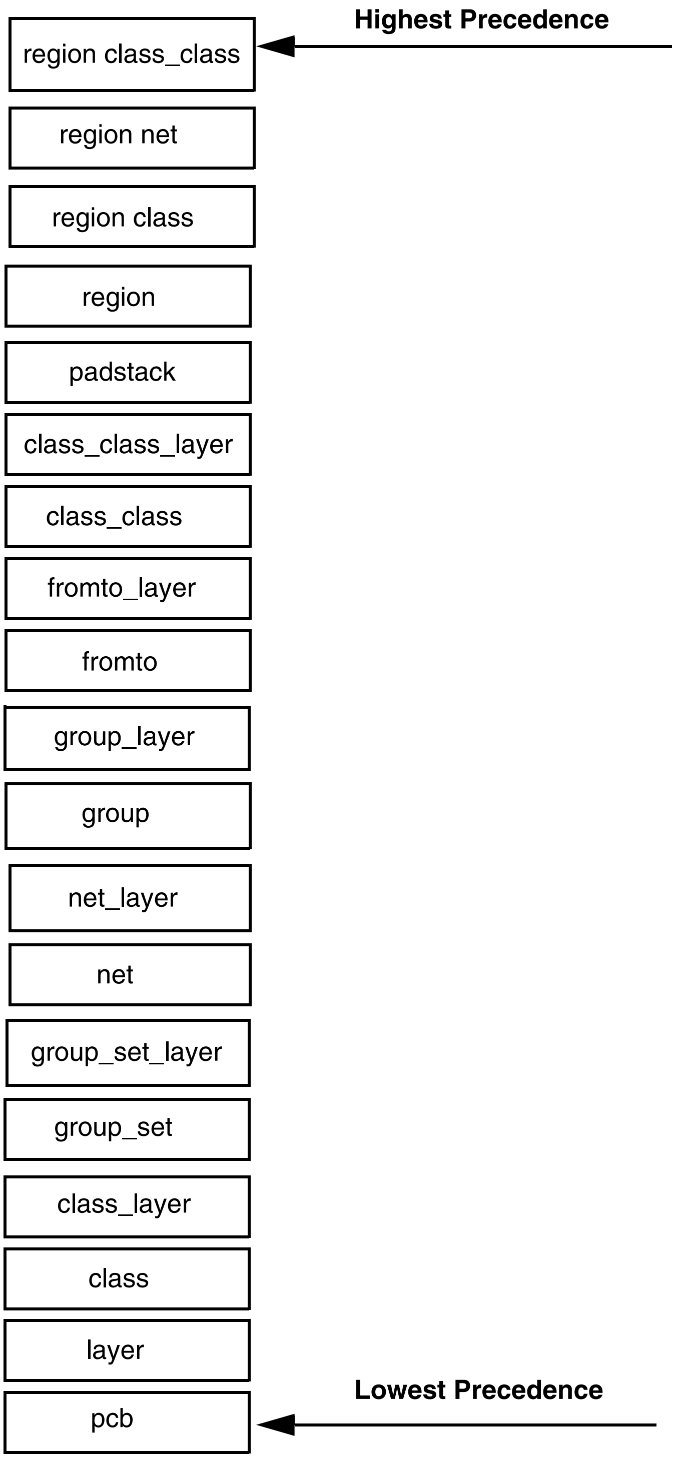
For interactive routing, you set width, clearance, and length (timing) rules. You can set these rules for layers, classes of nets, individual nets, groups of fromtos, sets of groups, individual fromtos, padstacks, and regions. When multiple rules apply to the same connection, the interactive router applies the rule with the highest precedence.

Routing Rules Hierarchy

The tool applies routing rules according to the hierarchy shown in the Routing Rules Hierarchy figure on page 79

A pcb rule (global rule for the design) has the lowest precedence in the hierarchy. A region class_class rule has the highest precedence. Rules set at one level of the hierarchy override conflicting rules set at lower levels.

Figure 3-2 Routing Rules Hierarchy



You can apply rules to all the levels shown in the rule precedence. However, some rules only apply at specific levels, as shown in the following table.

Table 3-2 Routing Rules Application

Level	Rules that Apply
class	clearance, wiring, timing, shielding, crosstalk, noise
class_class	clearance [1], crosstalk, noise
class_class_layer	clearance, crosstalk, noise
class_layer	clearance, wiring, timing, crosstalk, noise
fromto	clearance, wiring, timing, shielding, crosstalk, noise
fromto_layer	clearance, timing, crosstalk, noise
group	clearance, wiring, timing, shielding, crosstalk, noise
group_layer	clearance, timing, crosstalk, noise
group_set	clearance, width, timing
group_set_layer	clearance, width
layer	clearance [2], wiring, time_factor, crosstalk, noise, noise_weight, costing
net	clearance [2], timing, shielding, crosstalk, noise
net_layer	clearance, wiring, timing, crosstalk, noise
padstack	clearance
pcb	clearance, wiring, timing, crosstalk, noise
region class_class	clearance
region net	clearance, width, diffpair_line_width, neck_down_width, edge_couple_tolerance_plus, edge_couple_tolerance_minus
region class	clearance, width, diffpair_line_width, neck_down_width, edge_couple_tolerance_plus, edge_couple_tolerance_minus
region	clearance, width, diffpair_line_width, neck_down_width, edge_couple_tolerance_plus, edge_couple_tolerance_minus

[1] Except for all mechanical drill hole (mhole) clearances.
[2] Except for mechanical drill hole to mechanical drill hole (mhole_mhole) clearances.

Setting Global Width Rules

Global width rules are applied at the pcb level, which has the lowest priority in the Routing Rules Hierarchy on page 78. To set a global width rule of 10 mils between all object types, add the following command to your Do file (see Editing the Do File on page 193 for more information).

```
rule pcb (width 10)
```

Setting Wire Widths by Group, Net, and Class

You can assign width rules at group, net, and class levels and rely on the autorouter to determine which level has precedence during routing. Before you can assign rules to a group or class, you must define the group or class.

For example, to assign rules at group, net, and class levels, determine the group and class names, and the wire widths you want to use, and add the following commands to your Do file.

```
define (group G1 (fromto U8-7 U12-3)(fromto U10-4 U15-11))
rule group G1 (width 8)
define (class C1 AR0 AR3 RT3 S0)
rule class C1 (width 10)
rule net NETA (width 6)
```

Controlling Wire Width by Layer

When you set a width rule for a signal layer, all wires routed on that layer follow the rule unless they are overridden by a higher precedence rule.

For example, to control the width of wires on layers s2 and s3, add the following commands to your Do file.

```
rule layer s2 (width 1)
rule layer s3 (width 2)
```

Note: Layer rules are employed only when the appropriate license is installed (see Understanding Licensing on page 18 for more information).

Controlling Wire Widths of Net Classes by Layer

You can control the impedance of all nets in a class on a particular layer by setting a wire width rule for that layer. Compute the wire width that satisfies the impedance requirement and set the layer rule.

For example, you define a class called ECL, which is made up of nets ECL1, ECL2, and ECL3. To control the wire width of wires in the class ECL on layers s2 and s3, add the following commands to your Do file .

```
define (class ECL ECL1 ECL2 ECL3 (layer_rule s2 (rule (width 8))) (layer_rule s3
(rule (width 6))))
```

Note: Layer rules are followed only when the appropriate license is installed (see Understanding Licensing on page 18 for more information).

Controlling Differential Pairs by Region

You can constrain differential pairs within specific regions of a layer to facilitate routing through congested areas of the design such as BGA fields. Narrower line widths and smaller gaps may be required, not only for routing differential pairs through dense pin fields but also to adjust for changes in material as with rigid-flex designs. Since Allegro only allows for one material per layer, the only way to specify different constraints due to a change in dielectric constant is to use a constraint region. For example, you could add the following commands to a Do file to define a region and set region routing rules. The differential pair rules are shown in bold.

```
define (region BGA0 (polygon TOP 0 79 66.5000 81 66.5000 81 64.5000 79 64.5000 79
66.5000 ))
```

```
rule region BGA0(width 0.004)
rule region BGA0(clearance 0.004)
rule region BGA0 (clearance 0.004 (type via_via_same_net))
rule region BGA0 (clearance 0.004 (type smd_via_same_net))
rule region BGA0 (diffpair_line_width 0.005)
rule region BGA0 (neck_down_width 0.003)
rule region BGA0 (edge_primary_gap 0.005)
rule region BGA0 (neck_down_gap 0.003)
rule region BGA0 (edge_couple_tolerance_plus -1)
rule region BGA0 (edge_couple_tolerance_minus -1)
```

Special Considerations

Rule Conflicts:

Conflicting rules can occur. For example, if NET1 is in CLASS1, and NET2 is in CLASS2, the nets can be paired, and there may be different pair rules specified for the two classes. In these cases, the *most conservative* rule is applied.

Overlapping Regions:

Rules assigned to a region that have the same coordinates and layer range as an existing region are merged. Overlapping regions are allowed, but if rules conflict, the rules of the *last defined* region are used.

Controlling Fromtos Connected to Virtual Pins

You control the width of a wire between a physical pin and a virtual pin by using the `define net` command. The `define net` command creates the virtual pin and assigns a fromto rule.

For example, to control the wire width of fromto U1-1 connected to virtual pin VP1 on the net CLK1, add the following commands to your Do file (see Editing the Do File on page 193 for more information).

```
define (net CLK1 (fromto U1-1 (virtual_pin VP1) (rule (width 6))) (fromto
(virtual_pin VP1) U2-1) (fromto (virtual_pin VP1) U3-1))
```

Note:

- You can specify the exact location of a virtual pin (see Defining a Branch Topology on page 177 for more information).
- Virtual pin operations are followed only when the appropriate license is installed (see Understanding Licensing on page 18 for more information).

Controlling Clearance

Clearance rules can be set at any level of the routing rules hierarchy to control the minimum edge-to-edge distance allowed between conductive objects such as pins and vias. These rules are used during routing operations to evaluate spacing.

For example, you could add the following commands to a Do file to set clearance rules between smds and wires on a layer and bbvias and thruvias of a padstack respectively.

```
rule layer S1 (clearance 50 (type smd_wire))
rule padstack V25 (clearance 20 (type bbvia_thruvia))
```

Same Net Clearance Rules

Clearance rules can also be exclusively applied to objects that reside on the same net. For example, you could add the following commands to a Do file that would enable same net checking in the design, define a minimum distance of 5 units between the edges of conducting objects on different nets and 2 units between the edges of conducting objects on the same net.

```
set same_net checking on
rule pcb (clearance 5)
rule pcb (clearance 2 same_net)
```

High Density Interconnect Clearance Rules

In addition to setting clearance rules for typical routing objects, you can also set clearance rules for HDI-related objects such as mechanical and net-based drill holes as well as the different types of vias (bbvias, microvias, thruvias and testvias). This implies that PCB Router allows you to define same-net and net-to-net clearance rules for specific types of vias. This is because of the uniqueness of the spacing requirements for each specific type of via used in a design.

Once defined, individual HDI clearance rules can be enabled or disabled in a design using the `set` command.

For example, you could add the following commands to a Do file to enable and define a microvia_wire clearance rule at the pcb level.

```
set microvia_wire on
rule pcb (clearance 35 (type microvia_wire))
```

Or, you could add the following commands to a Do file to enable and define a microvia_bbvia clearance rule at the pcb level.

```
set microvia_wire on
rule pcb (clearance 35 (type microvia_bbvia))
```

Also, the full set of clearance rules can be enabled or disabled in a design simultaneously using the following `set` command

```
set <object|paired_object> <on | off>
```

Examples:

```
set bbvia on
```

enables clearance rules for all bbvias in the design.

```
set bbvia_bbvia off
```

disables clearance rules for all bbvias - to - bbvias in the design.

```
set micorvia_bbvia samenet on
```

enables clearance rules for all microvias - to - bbvias on the same nets in the design.

```
set drill_holes on
```

enables clearance rules for all drill holes in the design.

Important

By default, all HDI clearance rules are disabled (off).

Note: You must be licensed to run the PCB Router XL product (or greater) in order to use HDI clearance rules in your design.

If a conflict occurs between the clearance rules defined for the same level of the rule hierarchy, it is resolved using the approach that a more detailed clearance rule has higher priority.

Clearance rules on blind/buried vias are valid at all levels of rules hierarchy (ref. to hierarchy diagram) with the established precedence.

Drill Hole Examples

The following commands can be added either to a Do file or entered directly through command line interface of Specctra GUI window. However, the later enter has higher priority.

```
set drill_holes on
rule pcb (clearance 85 (type mhole_wire))
rule pcb (clearance 85 (type mhole_nhole))
rule pcb (clearance 85 (type mhole_pin))
rule pcb (clearance 55 (type mhole_via))
rule pcb (clearance 85 (type mhole_wire))
rule pcb (clearance 20 (type nhole_nhole))
rule pcb (clearance 80 (type nhole_area))
rule pcb (clearance 10 (type nhole_pin))
rule pcb (clearance 70 (type nhole_via))
```

You could also add the following commands to define drill hole clearance rules at the padstack level. Note that these would override some rules defined in the previous example due to rule precedence described in Figure 3-2 on page 79.

```
rule padstack v25 (clearance 20 (type mhole_mhole))
rule padstack v25 (clearance 20 (type mhole_nhole))
rule padstack v25 (clearance 15 (type nhole_area))
rule padstack v25 (clearance 15 (type nhole_via))
rule padstack v25 (clearance 15 (type nhole_wire))
```

Note:

- Mechanical drill hole (mhole) clearance constraints are unavailable at certain levels in the routing rules hierarchy. Refer to the footnotes in Table 3-2 on page 80 for details.
- `mhole` = mechanical drill hole
- `nhole` = net-based drill hole
- Drill hole and microvia object names are case insensitive. For example, `mhole_wire and Mhole_Wire` are both acceptable as are microvia_thrupin and MicroVia_ThruPin.
- Drill hole and microvia object pairs can be listed in reverse order. For example, `mhole_wire` and `wire_mhole` are both acceptable as are microvia_smd and smd_microvia.

- Antipads related to mechanical pins act as keepouts during routing operations. Also, antipad overlaps are checked and reported as violations. These actions are independent of the current settings (enabled/disabled) for drill hole rules.

Mircrovia Examples

The following commands can be added either to a Do file or entered directly through command line interface of Specctra GUI window. However, the later enter has higher priority.

```
set microvia on
rule pcb (clearance 35 (type microvia_wire))
rule pcb (clearance 37 (type microvia_microvia))
rule pcb (clearance 32 same_net (type microvia_microvia))
rule pcb (clearance 30 (type microvia_bbvia))
rule pcb (clearance 30 (type microvia_testvia))
rule pcb (clearance 20 same_net (type microvia_testvia))
rule pcb (clearance 45 (type microvia_thrupin))
rule pcb (clearance 45 same_net (type thrupin_microvia))
rule pcb (clearance 15 (type microvia_thruvia))
rule pcb (clearance 15 same_net (type microvia_thruvia))
rule pcb (clearance 20 (type smdpin_microvia))
rule pcb (clearance 20 same_net (type microvia_smdpin))
```

Bbvia Examples

The following commands can be added either to a Do file or entered directly through command line interface of Specctra GUI window. However, the later enter has higher priority.

```
set bbvia on
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 PCB (clearance 0.12 (type bbvia_thruvia))
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))
```

And for same net clearance:

```
rule PCB (clearance 0.1 same_net (type bbvia_bbvia))
rule PCB (clearance 0.1 same_net (type bbvia_smdpin))
rule PCB (clearance 0.1 same_net (type bbvia_thruvia))
rule PCB (clearance 0.1 same_net (type bbvia_testvia))
rule PCB (clearance 0.1 same_net (type bbvia_testpin))
rule PCB (clearance 0.1 same_net (type bbvia_bondpad))
rule PCB (clearance 0.1 same_net (type bbvia_area))
```

Note:

- The object names are case insensitive. This implies the routing algorithm and the DRC engine do not differentiate between say bbvia or BbVia.

- bbvia is a generalization for buried / blind vias
- The `same_net` keyword attaches the clearance rule for typed objects on the same net. If this is not present (default) in the rule definition, the rule is attached to typed objects on different nets.
- Clearance defines the distance between the nearest edges of two object. It does not define the distance between their centers.

For further details, see the `rule` and `set` commands.

Getting Placement and Routing Reports

After placing components or routing wires, you can display placement or routing reports in a report window. You can also save reports in text files. Reports give you detailed information about the placement or routing characteristics of your design, including rules and rule violations.

The reports you can access using the Report menu depend on whether you are in the routing or placement environment. Some reports are available in both environments, but in general you should be in the routing environment to get routing reports and the placement environment to get placement reports. See Switching Between Routing and Placement on page 41 for details.

To display placement or routing reports in a report window

1. Choose *Report* to display the Report menu.
2. Look for the report you want to display and do one of the following:
    - ❑ If you see the report name in the menu, click the report name. Some reports require additional information, such as an object ID or filename. If a dialog box appears, enter the information or set the controls, and click Apply or OK.
    - ❑ If you do not see the report name in the menu, choose *Specify* to open the Report Specify dialog box, and click the report name in the Reports list.

    The report opens in a report window. You can search the report for specific words or regular expressions. You can also save the report in a text file. When you finish reading the report, click *Close* to dismiss the Report window.

To search a report

1. Open the report in a Report window.
2. Enter the word or character string you want to find in the Search data entry box.

    You can enable or disable the Case Sensitive control depending on whether you want to search only for the exact uppercase and lowercase letters you entered in the data entry box.
3. Click the *up arrow* button to search forward through the report.

    -or-

    Click the *down arrow* button to search backward through the report.

To save a report in a text file

1. Open the report in a Report window.

   If you know the name of the file, enter it in the Save data entry box and proceed to step 3.

2. Click the *Browse* button.

   A File browser opens. Select a file from the browser.

3. Click the *Save* button.

If you have opened several report windows, you can use the Report menu to close all windows at the same time.

To close all report windows

➤ Click *Report – Close All*.

You can also use the `report` command to open a report in a report window or save a report in a text file.

For example:

```
report status
report status trial1.sts
report status window
```

If you do not specify a filename, the tool applies the default filename for the report. For general information about specifying filenames, see File Naming Conventions on page 37.

Checking Design Rule Violations

The tool automatically checks for conflicts and rule violations at the beginning of a session and after every routing or placement operation. You can disable rule checking for some interactive routing or placement operations. When rule checking is enabled, you cannot place a component or route a wire anywhere that violates placement or routing design rules.

To disable or enable rule checking

➤ Click *Checking* at the right end of the Status bar.

A check mark indicates that rule checking is enabled.

If you choose to disable rule checking during interactive routing or placement operations, you should enable checking and run a rules check after performing the interactive tasks to look for new conflicts or rule violations.

You should also run a rules check if you add or change a rule during the session. You can check for routing conflicts and rule violations, placement rule violations, or both.

Note: If you select one or more design objects, the tool checks only those objects for violations. Otherwise, the tool checks the entire design.

To check for conflicts and rule violations in the routing environment

Do either of the following:

- Choose *Rules – Check Rules – Routing.*

  This checks for routing rule violations.

- Choose *Rules – Check Rules – All*.

  This checks for routing conflicts and both routing and placement rule violations

To check for conflicts and rule violations in the placement environment

Do either of the following:

- Choose *Rules – Check Rules – Placement*

  This checks for placement rule violations.

- Choose *Rules – Check Rules – All*

This checks for routing conflicts and both routing and placement rule violations.

To check for conflicts and rule violations in the current viewing area

➤ Click the *Check Window Area* icon in the tool bar to check for routing conflicts and both routing and placement rule violations within the area of the design *that is currently visible* in the work area.

The tool displays graphic symbols where conflicts or violations have occurred. See Router Symbols on page 339 for details.

- When you check for routing conflicts and rule violations, the checker examines the current wiring based on current routing rules.
- When you check for placement rule violations, the checker examines the current component positions and orientations based on current placement and routing rules.

By default, the checker checks for clearance and crossover conflicts, length and crosstalk rule violations, and placement rule violations. You can set rule checking controls to ignore some or all of these, or to check for additional rule violations.

You can generate routing or placement reports that list all current conflicts or rule violations.

To analyze rule violations

1. Select the objects involved in the violation.
2. Choose *Report – Specify*

   The Report Specify dialog box appears.
3. Do either of the following:
   - ❑ Choose *Conflicts – Placement* to generate a report that lists rule violations.
   - ❑ Choose *Conflicts – Routing* to generate a report that lists routing conflicts and rule violations.

   The checker displays the report in a Report window.
4. Review the violations listed in the Report window. When you have finished reviewing the report, you can click *Close* to dismiss the report window.

Creating Keepout Areas

Keepouts are areas of the design where you want to prohibit routing or placement objects. You can define a general keepout for all objects, or specific keepouts for components, wires, vias, wire bends, or wire elongations. A keepout area can apply to a single layer or to all layers of the design.

You can define new keepout areas, and you can replace keepout areas defined in the tool or in the structure section of the Design file. To replace a keepout area, the tool disbands the old area and defines a new one. You cannot disband or replace a keepout area defined for an image in the library section of the Design file.

Use one of the following methods to define or replace a keepout area.

- Draw Keepout mode to draw keepout areas interactively.
- Define Keepout to define the exact dimensions and locations of keepout areas by specifying X and Y coordinates.

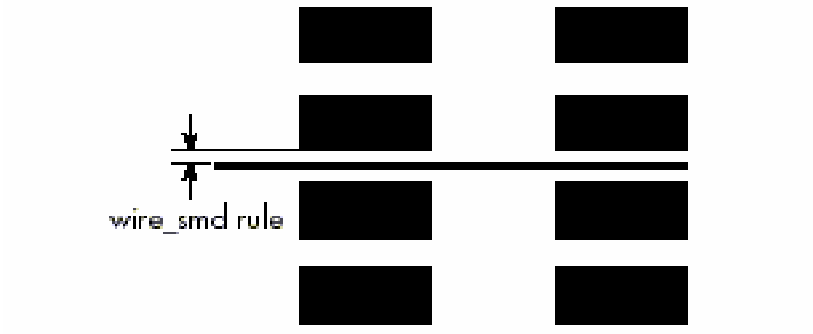
Note: You can modify keepout area boundaries in Add/Edit Polygon mode (see Changing existing vias for details), and you can disband keepout areas by deleting them within Delete Keepout mode.

Optimizing Design Rules

The router offers more flexibility for setting rules by allowing separate control of rules such as wire_pin clearance, wire_smd clearance, and wire_via clearance. You can also control widths and clearances by layer, by net, and by individual connection.

If manufacturing rules permit, set the wire_smd clearance rule to allow one wire between SMD pads as shown in the following figure.

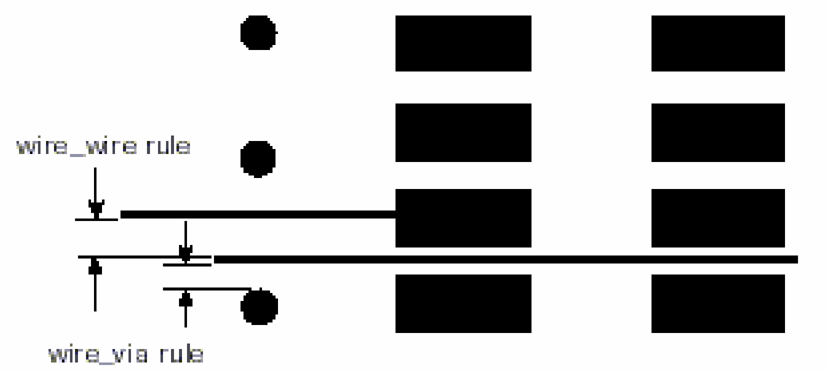
Figure 3-3 Allow Wire



Wire-to-SMD rule should allow one wire between pads.

You should set the wire_pin and wire_via rules to allow two wires between adjacent vias where possible. The following figure shows a good wiring pattern that results when the wire_via rule is properly set.

Figure 3-4 Wire_Via Rule Set Properly



Wire-to-wire and wire-to-via rules should allow two wires between adjacent vias.

Using Pin Delay

A schematic or board-driven flow supports pin-delay values, by associating the PIN_DELAY property with particular component instance or definition pins. You can include pin delay in DRC calculations for DIFFERENTIAL PAIR PHASE TOLERANCE, PROPAGATION DELAY, and RELATIVE PROPAGATION DELAY by assigning the PIN_ DELAY property to component instance or definition pins. The PIN_ DELAY property specifies the time delay or length from the internal package connection to the pin's mounting layer. Use the PIN_DELAY property to manage interchip delay or die-to-die timing paths across a board and thereby factor inter-package delay into timing requirements.

Pin delay specifies the length of a through pin multiplied by router's **pindelay_prop_velocity _factor factor**, which is a constant used to convert from time to length units if you defined DIFFERENTIAL PAIR PHASE TOLERANCE, PROPAGATION DELAY, and RELATIVE PROPAGATION DELAY in time units. To factor pin delay into these DRC calculations, you enable the *Pin Delay Include in All Propagation Delays and in DiffPair Phase Checks* field in the *Options tab* of the *Electrical Constraints Dialog Box*, accessible by running the `cns electrical` command in PCB Editor, described in the *Allegro PCB and Package Physical Layout Command Reference*.

Schematic-driven pin delay flow

In a schematic-driven flow, these pin-level delay values are defined as library properties that can be written to the Design Entry HDL or Allegro Design Entry CIS library files sent to PCB Editor. You can use Allegro PCB System Librarian 610 to manually assign the PIN_DELAY property and values to symbol pins or automatically import the PIN_DELAY values through its *Import Wizard*, which supports Comma Separated Value (`.csv`) and Excel (`.xls`) file format options.

Packager-XL packages the design into `pst*.dat` files. The `pstchip.dat` file contains the default values of the PIN_DELA Y property, subsequently imported into PCB Editor using the latter's *File – Import – Logic* (`netrev` command).

You can use the PIN_DELAY property values in Constraint Manager, interactive routing in PCB Editor or PCB Router, automatic routing in PCB Router, and DRC verification in both PCB Editor and PCB Router.

In PCB Editor, you can then optionally edit and override the PIN_DELAY values from the `pstchip.dat` file on component-instance pins. Only overrides are backannotated to the schematic.

Board-driven pin delay flow

In a board-driven, pin-delay flow, you can export pin delay values from an external source using PCB Editor's *File – Export – Pin Delay* (`pin_delay out` command described in the *Allegro PCB and Package Physical Layout Command Reference*) and then import them to another design and annotate them to component instance pins using *File – Import – Pin Delay* (`pin_delay in` command) in the physical layout editor. You can also use *Edit – Properties* (`property edit` command) to assign the PIN_DELAY property.

You can also derive pin delays from an external source or from APD with its Package Pin Delay Report, generated by using the *File – Export – Board Level Component* (`allegro_component` command described in the *Allegro PCB and Package Physical Layout Command Reference*).

When you change the value of the PIN_DELAY property in PCB Editor and use *File – Export – Logic* (`feedback` command described in the *Allegro PCB and Package Physical Layout Command Reference*) to export modifications to Design Entry HDL, both PCB Editor-modified (instance) and Design Entry HDL-generated (definition) PIN_DELAY values come across in the `cmdbview.dat` file if you are using a Constraint-Manager-enabled flow. If you are using a flow without Constraint Manager, manual edits to the value of the PIN_DELAY property pass in the `pstxprt.dat` file.

Constraint Manager is an optional point to enter PIN_DELAY values and to edit those propagated from the `chip.prt` files. The Constraint Manager flow maintains any overrides of the PIN_DELA Y property made in Constraint Manager or in PCB Editor, but does not backannotate them to the Design Entry HDL schematic.

In the router, you can control pin-delay functionality by specifying:

- pin delay on a pin for a component using:

  image <image_id> (pin <padstack_id> [(rotate <rotation>)] [pin_delay <delay_value>]))

- pin length on a pin for a component using:

  (image <image_id> (pin <padstack_id> [(rotate <rotation>)] [pin_length [<positive_dimension>| 0]]))

- pin delay on a pin instance using:

  (network (net <net_id> (pin_delay <delay_value> {<pin_reference>}))

  define (net <net_id> (pin_delay <delay_value> {<pin_reference>}))

- pin length on a pin instance using:

  (network (net <net_id> (pin_length [<positive_dimension >|0] {<pin_reference>})))

  define (net <net_id> (pin_length [<positive_dimension >|0] {<pin_reference>}))

- pin delay propagation velocity factor using:

  (structure (control [pindelay_prop_velocity_factor factor)]))

More information on these command and design syntaxes is described in the *Allegro PCB Router Command Reference* and the *Allegro PCB Router Design Language Reference,* respectively*.*

You can include pin delay in calculations involving automatic or interactive routing of connections that use length or delay; reports that detail length or delay diagnostics; and DRC calculations for DIFFERENTIAL PAIR PHASE TOLERANCE, PROPAGATION DELAY, and RELATIVE PROPAGATION DELAY.

PCB Editor or any other host CAD system interfacing to the router must supply the required design or `.do` file syntax to enable the pin-delay functionality. SPIF only translates pin-delay values to the router, however, if the *Pin Delay Include in All Propagation Delays and in DiffPair Phase Checks* field is enabled in the *Options tab* of the *Electrical Constraints Dialog Box*, accessible by running the `cns electrical` command in PCB Editor. Otherwise, these DRCs are verified only against etch in the router.

Even if pin delays exist on component instance or definition pins in the router, PCB Editor's *Pin Delay Include in All Propagation Delays and in DiffPair Phase Checks* field must be enabled to include pin delay in the router functions utilizing length or delay for connection length in their calculations. The default for each pin on the design in the router is zero; that is, no pin delay.

Values specified in the *Max Length* field on the *Fanout Dialog Box,* available by running *Auto – PreRoute – Fanout,* are excluded from pin-delay calculations as is stub length checking.

The *Length and Delay Rule* Report's *Total Len* and *Status* fields include length or delay values when pin-delay functionality is enabled. A *Pin Delay* length column summarizes pin-delay length contributions.

Using Z Axis Delay

To more accurately account for delay in your designs, you can include the conducting portion of a via/pin (also known as Z Axis Delay) in DRC calculations for DIFFERENTIAL PAIR PHASE TOLERANCE, PROPAGATION DELAY, and RELATIVE PROPAGATION DELAY. Z Axis Delay includes any through-hole component or any hole with a depth value. The conducting portion of a via/pin comprises the thickness through the design from the placed symbol layer where a net enters a padstack, which may be a via or a through-hole pin, to the layer from which it exits.

All layer dielectric and copper thickness lengths between the entry and exit layers are calculated for the conducting portion of a via/pin and are added to the overall net or pin pair length. Copper thickness for the entry and exit layers are excluded from the calculations. Surface mount vias, such as testpoints, have no effect on the total calculations.

To factor the conducting portion of a via/pin into these DRC calculations in the router, you can control Z Axis Delay functionality by specifying:

- whether to enable Z Axis Delay functionality using either:

  (structure (control (via_pin_length [on | off]))

  set via_pin_length [on | off]

- copper thickness for a conductive layer using:

  (structure (layer (layer_name (type signal) (copper_thickness thickness))))

- layer-to-layer thickness between conductive layers using:

  (structure (layer (layer_name (type signal) (thickness thickness))))

- via delay propagation velocity factor using:

  (structure (control [(viadelay_prop_velocity_factor factor)]))

For more information on controlling Z Axis Delay functionality in the router, see the appropriate command and design syntax, which is described in the *Allegro PCB Router Command Reference* and the *Allegro PCB Router Design Language Reference,* respectively.

PCB Editor or any other host CAD system interfacing to the router must supply the required design or `.do` file syntax to enable Z Axis Delay functionality. SPIF only translates Z Axis Delay values to the router, however, if the *Z Axis Delay Include in All Propagation Delays and in DiffPair Phase Checks* field is enabled in the *Options tab* of the *Electrical Constraints Dialog Box*, accessible by running the `cns electrical` command, in PCB

Editor, described in the *Allegro PCB and Package Physical Layout Command Reference*.

If Z Axis Delay functionality is enabled in PCB Editor and PCB Router, then Z Axis Delay is included in router's delay calculations for auto and interactive routing of connections that use length or delay; reports that detail length or delay diagnostics; and DRC calculations for DIFFERENTIAL PAIR PHASE TOLERANCE, PROPAGATION DELAY, and RELATIVE PROPAGATION DELAY. The conducting portion of a via/pin does not contribute to total net length or delay rules. By default, Z-Axis Delay is disabled in the router, in which case these DRCs are verified only against etch. For more information about using Z Axis Delay in PCB Editor, see Chapter 3, Working with Constraints, in the *Allegro PCB and Package User Guide: Creating Design Rules*.

> **Note:** If **via_pin_length** is on, DRC length and delay checking include the conducting portion of a via/pin only if the *Effective Via Length* parameter on the *PCB Wiring Rules Dialog Box,* available by running *Rules – PCB – Wiring – General*, does not apply to the calculation.

Values specified in the *Max Length* parameter on the *Fanout Dialog Box***,** available by running *Auto – PreRoute – Fanout,* do not apply to any Z Axis Delay calculation, which might create potentially undesirable effects for autorouting. The *Max Length* parameter ensures the fanout via for middle pins on ordered nets does not exceed the stub length, permitting the router to create a junction at the fanout via without violating the stub length. Given Z Axis Delay functionality, the conducting portion of a via/pin affects the stub length calculation, which means that the *Max Length* parameter might be insufficient to prevent routing problems due to stub-length issues resulting from fanout lengths. The autorouter applies no algorithm to resolve length or delay issues (for example, elongation) by adding vias solely to meet length or delay constraints.

The *Length and Delay Rule* report's *Via Len, Total Len,* and *Status* fields include Z Axis Delay length or delay values when **via_pin_length** is **on** only if the *Effective Via Length* parameter does not apply to the connection. If the *Effective Via Length* field applies, it overrides and excludes Z Axis Delay from delay calculations. For more information on **via_pin_length,** see the *Allegro PCB Router Design Language Reference*.

4

Placing Components

In this chapter . . .

Understanding Component Placement

Note: PlaceBase licenses are required for autoplacement procedures.

A placement session can involve some or all of the following procedures:

- Setting Up the Placement Environment
- Setting Placement Rules
- Interactively Placing Critical Components
- Automatically Placing Large Components
- Automatically Placing Small Components

After automatically placing large or small components, you can use interactive placement tools to meet special design requirements or optimize routability.

Setting Up the Placement Environment

You can use the Placement Setup dialog box to set some global placement options and constraints. In addition, you can display or hide component labels and routing guides. To meet specific design requirements, you can set separate placement grids for SMD and through-pin components, define or modify keepout areas, and modify the placement boundary. If necessary before you begin automatic placement, you should lock preplaced components and unplace any other components that are inside the placement boundary.

Setting Placement Rules

You set rules to control the placement process. The rules you set can include spacing, permitted orientations, permitted sides, and opposite sides. The tool obeys rules during interactive placement operations, and prevents design rule violations while rule checking is enabled.

Interactively Placing Critical Components

You can interactively preplace and lock connectors and other position-critical components. The autoplacer uses preplaced component connectivity to determine the order and locations for other large components.

Performing Basic Placement Procedures

Basic component placement involves placing large and small components, checking the placement status report for rule violations, and saving your results. For a basic introduction to component placement, try performing these basic steps on a practice design.

1. Choose *Autoplace – Unplace – All*.

   This removes components from within the design boundary. The autoplacer automatically places only components that are completely outside the boundary.

2. Choose *Autoplace – InitPlace Large Components*.

   In the InitPlace Large Components dialog box, click *All* and click *OK*.

   This places all large components using rules defined in the Design File.

3. Choose *Autoplace – Interchange*.

   In the Interchange Components dialog box, click *Large* and click *OK*.

   This interchanges large components to improve routability by reducing Manhattan lengths.

4. Choose *Autoplace – InitPlace Small Components – All*.

   In the InitPlace All Small Components dialog box, click *OK*.

   This places all large components using rules defined in the Design File.

5. Choose *Report – Specify*.

   In the Report Specify dialog box, click *Place Status*. Close the report window when you finish reading the report.

   This opens the placement status report to check the results.

6. Choose *File – Write – Session*

   Click *OK*.

   This saves the results in a Session file. You can use this file to restart the session later or to translate the results to your layout system.

Evaluating and Optimizing Placement

During component placement, the tool indicates Placement Rule violations graphically and lists them by category in a conflicts report. Other reports provide information about design objects, current rules, room definitions and constraints, as well as the results of different placement operations.

When you generate a report, you can either view it in a report window or save it in a text file. For more information, see Getting Placement and Routing Reports on page 88.

In addition, two graphic reports help you see both the congested and under-utilized areas of the design.

- The Histogram report provides a visual representation of crossing congestion across the design.
- The Density Analysis report shows congestion by area to help you localize placement problems.

To correct rule violations and optimize placements, you can place components interactively. For more information, see Using Interactive Placement Tools on page 138.

For more information, see the following sections:

Understanding Component Placement

Designing and Using a Floor Plan

Additional Placement Procedures

Additional Placement Procedures

The tool provides additional capabilities you can use to satisfy specialized design or manufacturing requirements. These include:

- Creating and applying small component patterns for repetitively placing one or more small components, such as decoupling capacitors, near a large component
- Creating component associations between a large component and one or more small components such as resistors
- Assigning physical and electrical properties to components
- Controlling opposite side (front-to-back) placement
- Controlling placement according to power dissipation limits and height constraints
- Forming clusters by power net connections for split power planes
- Adding, modifying, or removing keepout areas
- Modifying the placement boundary

For more information, see the following sections:

Understanding Component Placement

Evaluating and Optimizing Placement

Designing and Using a Floor Plan

Setting Placement Controls

Using Placement Setup

Use the Placement Setup dialog box to set some global interactive and automatic placement constraints.

Through the Placement Setup dialog box, you can:

- Set global placement grid and spacing values.
- Control whether the autoplacer considers high-speed routing constraints during placement operations.

For interactive placement you can also:

- Change the pointer style of the screen pointer.
- Control whether the tool retains optimal power net routing of wires connected to components you interactively place or relocate.
- Change the alignment reference point for interactive component alignment.
- Choose the direction the tool shoves aside components to provide space for components you interactively move, flip, or pivot.
- Control whether the tool displays a force vector during component moves.
- Change the move component reference point and the component working origin for interactive place, move, pivot, and push operations.
- Change the move direction for component moves.
- Choose an option for deleting, extending, reconnecting, or attaching guides to wires after a component move.
- Choose to use alignment marks in Measure mode.
- Choose to restrict Measure mode mouse movements to 45 degree increments.
- Choose where to display measurement information.

For automatic placement, you can also:

- Control whether the autoplacer considers secondary net connections of small components connected to two large components on different nets during automatic placement and interchange operations.

The Placement Setup dialog box is divided into four panels: *General*, *Alignment*, *Move*, and *Measure*. Click the tab to choose a panel. The default Setup panel when you start a session is *General*. If you change the Setup panel, the dialog box changes to display the appropriate controls.

You can return to the Placement Setup dialog box and alter your settings as needed during a session.

To set up for automatic or interactive placement

1. Do one of the following to open the Placement Setup dialog box:
   - ❑ Choose *Autoplace – Setup*.
   - ❑ Press [RB] and choose *Setup*.
2. To change the Setup panel, click the tab to choose the panel you want.
3. Change settings by using any or all of the following, depending on the placement tasks you want to perform.

   PCB Placement Grid data entry box

   PCB Placement Spacing data entry box

   In General:

   Pointer Style option

   High Speed option

   Consider Secondary Connection option

   Rebuild Power Net option

   In Alignment:

   Align Reference option

   In Move:

   Shove for Move menu

   Show Move Vector option

   Move Component option

   Move Direction option

   Move With Wires option

In Measure:

Alignment marks menu

Snap angle option

Output option

4. Click *Apply* or *OK*.

Setting Placement Grids

A placement grid is not required to place components. If your design or manufacturing rules require a placement grid, you can:

- Set a single PCB grid for all components
- Set separate SMD and through-pin (PTH) grids

You can set a grid for SMD components only. If you set a grid for through-pin components, you must also set an SMD grid. The through-pin grid value must be a multiple of the SMD grid value.

You use placement grids when you place or relocate components during both automatic and interactive operations. The default grid values are all zero, which means no placement grids are defined. If you set a placement grid and later want to work without the grid, reset the value to zero.

You can change the placement grids at any time during the session. The new grids are used for subsequent placement operations, but current placements are not affected.

You can display major grid points on the placement grids. The major grid spacing is a multiple of the placement grid spacing. You can display major grid points in a different color from the placement grid color.

You can also control the visibility of the placement and major grids, and control whether the grids are displayed as lines or dots.

To set a global placement grid for all components

1. Open the Placement Setup dialog box by doing one of the following:
   - ❑ Choose *Autoplace – Setup*.
   - ❑ Press [RB] and choose *Setup*.
2. Enter a value in the PCB Placement Grid data entry box.
3. Click *Apply* or *OK*.

To set separate placement grids for SMD and through-pin components

1. Choose *Define – Design Grids.*

   The Design Grids dialog box appears.

2. Click the *Placement* tab.
3. Set the grid spacing values for SMD components, through-pin components, or both. You can:
   - ❑ Enter a value in the Place Grid data entry box to set a placement grid for all components.
   - ❑ Enter a value in the SMD Grid data entry box to set a placement grid for SMD components. This value must be a multiple of the value in the Place Grid data entry box if the Place Grid value is not 0.
   - ❑ Enter a value in the PTH Grid data entry box to set a placement grid for through-pin components. This value must be a multiple of the value in the SMD Grid data entry box.
4. Click *Apply* or *OK*.

To set the major grid and control how placement grids are displayed

1. Choose *View – Display Grids*

   The Display Grids dialog box appears.
2. On the Placement Major Grid panel:
   - ❑ Enter a value in the Factor data entry box if you want to display major grid marks on the placement grids. This value is the number of placement grid points between each major grid point.
   - ❑ Enable or disable Grid to control whether the placement grids are visible.
   - ❑ Enable or disable Factor to control whether the major grid marks are visible on the placement grids.
   - ❑ Click the As popup menu and choose either Lines or Dots to display the placement grids as lines or dot. The default is Lines.
3. Click *Apply* or *OK*.

   When you use a placement grid during interactive placement, components snap vertically or horizontally to the nearest grid point based on the current Move Component option setting in the Interactive Placement Setup dialog box.

   Grid values set in the Design Grids dialog box take precedence over the grid value set in the Placement Setup dialog box.

   - ❑ Make sure you enter grid values that are scaled for the units you are using.

- ❑ You can also display or hide wire grids using the Layers panel.
- ❑ In addition to the placement grids, you can set site grids for individual images. Choose *Define – Design Grids* to open the Design Grids dialog box.

Using High-Speed Rules

When you set maximum and matched net length or delay rules for routing, the autoplacer can use these rules to guide component placement.

The autoplacer assigns a higher placement priority to components on nets with length or delay rules. These rules assure that the design can meet wire length limits during autorouting.

You can use the high-speed control in the Placement Setup dialog box to determine whether the autoplacer observes high speed routing rules during component placement operations. By default, the high-speed control is enabled.

To enable or disable high speed control for placement

1. Open the Placement Setup dialog box by doing one of the following:
   - ❑ Choose *Autoplace – Setup*.
   - ❑ Right-click and choose *Setup*.
2. Click *High Speed* to enable it or disable it.
3. Click *Apply* or *OK*.

You can also use the `high_speed` command to enable or disable high speed constraints for placement.

For example:

```
high_speed off
```

Locking and Unlocking Components

Use the [Un]Lock Components dialog box to lock or unlock components. You can lock or unlock all placed components, components you select, or components you choose from a list. You can also specify whether to lock or unlock component position or swappable gates, subgates, or pins.

To lock or unlock selected components, you must select the components before opening the [Un]Lock dialog box.

To lock or unlock components from a list

1. Choose *Edit – [Un] Lock Components*

   The [Un]Lock Components dialog box opens.

2. Specify the components you want to lock or unlock by doing one of the following:
   - ❑ Click the *All* button.
   - ❑ Click the *Selected* button.
   - ❑ Click the *By List* button, and choose one or more components in the Components List.
3. Choose an Action by clicking the *Lock* button to lock components or the *Unlock* button to unlock components.
4. Choose one or more Lock Type options. You can:
   - ❑ Click Position to lock or unlock component positions.
   - ❑ Click Gate to lock or unlock swappable gates.
   - ❑ Click Subgate to lock or unlock swappable subgates.
   - ❑ Click Pin to lock or unlock swappable pins.
5. Click *Apply* or *OK*.

You can also use the `lock` and `unlock` commands to lock and unlock component positions, gates, subgates, or pins.

For example:

```
lock selected (type position)
lock U11 U12 U13 (type gate)
unlock selected (type position gate)
```

Unplacing Components or Clusters

If you want to move components outside the design boundary, you can have the tool unplace them for you. Unplacing components is sometimes necessary before automatic placement.

During automatic placement, the autoplacer places only those components that are located completely outside the design boundary. If components you want to place are located partially or completely within the boundary, you must unplace them.

You can unplace all placed components, components of a particular property type, components you select, components you choose from a list, or component clusters. Before unplacing components, lock any components that you want to remain inside the boundary.

To unplace all placed components

➤ Choose *Autoplace – UnPlace – All Components*.

To unplace components of a particular property type

Do one of the following:

- Choose *Autoplace – UnPlace – Small Components*.

  This unplaces all small components, including small capacitors, small resistors, and small discretes. Large components, including large capacitors, resistors, and discretes are not unplaced.

- Choose *Autoplace – UnPlace – Discrete Components*.

  This unplaces small discrete components. It does not unplace large discretes.

- Choose *Autoplace – UnPlace – Capacitors*.

  This unplaces small decoupling capacitors. It does not unplace large capacitors.

- Choose *Autoplace – UnPlace – Resistors*.

  This unplaces small resistors. It does not unplace large resistors.

To unplace clustered components

➤ Choose *Autoplace – UnPlace – Clusters*

This unplaces all components grouped into component clusters.

To unplace selected components

➤ Choose *Autoplace – UnPlace – Selected Components*.

This unplaces all currently selected components.

To unplace components you choose from a list

1. Choose *Autoplace – UnPlace – Components By List*.

   This opens the Unplace Components dialog box.

2. Choose one or more components from the Components list.
3. Click *Apply* or *OK*.

   If any components have attached wires, a message popup appears with the prompt:

   ```
   Delete the Wires on the Components and then Unplace?
   ```

   Click *Yes* to delete the wires. Click *No* to keep the wires. If you click `No`, the tool does not unplace the components except those where all the wires have the pin escape attribute. These wires are moved with the components.

Using the Basic Placement Process

Preplacing Connectors and Critical Components

After setting up your placement environment and assigning the rules you need, you are ready to begin placing components. The first step is to interactively preplace connectors and other critical components.

Before you begin preplacing components, you might need to disable rule checking. For example, rule checking must be disabled when you place an edge connector with its plating bar outside the design boundary. When rule checking is enabled, you cannot interactively place any part of a component outside the boundary.

You use the interactive [LB] modes to preplace components. With these mode, you can:

- Place components that are outside the design boundary by using Move Component Mode or one of the Place Component modes.
- Relocate components that are inside the design boundary by using Move Component mode, Pivot Component Mode, or Flip Component Mode
- Lock components at their current positions inside the design boundary by using the Lock Position mode. This prevents the tool from adjusting their positions during automatic placement.

You can also use the following modes:

- Push Component Mode to push other components out of the way while moving a component within the design boundary
- Trade Component Mode to switch the positions of pairs of components
- Align Component Mode to align components with respect to a particular reference pin

After preplacing and locking critical components, enable rule checking and check for Placement Rule violations.

You should also unplace any other components that are inside the design boundary. During automatic placement, the tool places only those components that are outside the boundary.

Automatically Placing Large Components

After preplacing connectors and critical components interactively, you are ready to automatically place large components. This generally involves the following steps:

1. Performing an initial constructive placement
2. Interchanging placed components
3. Rotating placed components
4. Swapping gates and pins interactively

The first step is to take each large component that is outside the placement boundary and placing it inside the boundary. The placer starts by placing larger components with the highest connectivity.

The second, third, and fourth steps optimize the initial placements, reducing Manhattan lengths and improving routability.

Automatically Placing Small Components

Generally, you place small components after placing large components.

You can place all small components at once, or you can place small capacitors, small resistors, and small discretes in separate operations using different preferences.

For instance, you can place small capacitors first, then place small discretes, and finally place all remaining small components (in this instance including small resistors). However, if you do not need to treat some small components differently from others, you can place all small components in a single operation.

Placing small components usually involves an initial placement operation followed by one or more operations designed to optimize the placements. To achieve this, you can:

- Interchange components
- Rotate components
- Swap equivalent gates and pins

In the initial placement, the autoplacer takes components that are entirely outside the design boundary and places them inside the boundary. After placing small components, you should optimize the placements to reduce Manhattan lengths and improve routability.

You usually place decoupling capacitors before other small components. During automatic placement, the autoplacer attempts to place them near the power pins of large components. If space is available, the autoplacer can place capacitors on the other side of the design directly behind SMD power pins.

Evaluating Component Placement

This application provides reports and graphical analysis tools to help you evaluate your placement results. These tools can help you to detect and correct placement violations and to achieve optimal placement results. You can

- Click *Report – Place Status* to read the place status report to review the placement process and see the results of each placement operation.
- View graphical marks and read the placement conflicts report to identify and correct placement rule violations (see Checking Design Rule Violations on page 90 for details).
- Use the autorouter to evaluate placement results.
- Click *Report – Specify – Place Lengths* to evaluate the Place Lengths report.

See Viewing a Density Analysis on page 119 for details on evaluating placement in congested areas of the design.

Viewing a Histogram

The histogram is a graphic analysis tool that uses bar graphs to represent relative crossing congestion across the design. The graphs along the bottom and right side edges of the design boundary indicate crossing congestion areas by the height of each bar in the display.

Use the histogram to identify crossing congestion areas of the design. Smooth histogram curves indicate an even distribution of connections across the design. Any large peaks in the histogram indicate extremely congested areas that you should correct, if possible.

To show or hide the histogram display

➤ Click *Autoplace – Crossing Histogram.*

Each bar in the histogram represents an invisible cut-line that extends across the design. The display along the bottom edge of the design boundary represents normalized crossing cuts that intersect the vertical cut-lines. The display along the right edge of the design boundary represents normalized crossing cuts that intersect the horizontal cut-lines. A small dot above a bar indicates the most highly congested cut-line.

You can also use the view or vset command to show or hide the crossing histogram display.

- To view the histogram, enter:

  ```
  view histogram on
  ```

  or

  ```
  vset histogram on
  repaint
  ```

- To hide the histogram, enter:

  ```
  view histogram off
  ```

  or

  ```
  vset histogram off
  repaint
  ```

Viewing a Density Analysis

You can generate a density map that shows current circuit congestion by overlapping the design with an array of colored cells. You can also show or hide the display without updating the density map.

When you perform a density analysis, the tool calculates circuit congestion (wire channel demand based on the Manhattan tree) at the center of each cell, computes or updates the density map, and automatically sizes the cells. Wire channel supply is based on the number of signal layers and the projected number of required vias.

This tool creates an array of rectangular cells that overlay the entire design. Each cell is filled with a color to represent its relative congestion.

To generate (or update) and display the density map

➤ Click *Autoplace – Density Analysis*.

A color index below the design shows the colors used for different degrees of congestion. The tool uses basic color codes to represent relative cell congestion. The default colors are red, yellow, and green.

- Red indicates high congestion
- Yellow indicates moderate congestion

- Green indicates low congestion

Some of these colors might be different depending on your current color map.

To hide the density map

➤ Click *Autoplace – View Off*.

You can also use the `density_analysis` command to generate or update the density map, and use the `view` or `vset` command to show or hide the density map without updating it.

To calculate the current circuit congestion, enter:

```
density_analysis
```

To view the current density map, enter:

```
view density on
```

or

```
vset density on
```

```
repaint
```

To hide the density map, enter:

```
view density off
```

or

```
vset density off
repaint
```

Designing and Using a Floor Plan

Designing a Floor Plan

Use a floor plan to control how and where you place components. Floor planning helps you restrict components to certain areas on a crowded design. In a floor plan, you divide the design into rooms and assign room content according to power supply connections, component height, power dissipation, or other criteria.

You can define room boundaries for different areas of the design, and assign room contents by specifying conditions for including or excluding components, or clusters of components. You can also assign Placement Rules to individual rooms.

To design your floor plan, group the components into floor plan clusters. This initial step enables you to place components together based on signal net or power supply connections, or to satisfy other criteria that you define. You can:

- Define clusters by choosing individual components
- Form clusters automatically by signal or power net connections, by large component connectivity, or by connections to a seed component

Although you do not need clusters to define rooms and assign components to them, clusters make it easier to determine how to partition the design and size the rooms of your floor plan.

To complete your floor plan, you define rooms that control where and how the autoplacer places components or clusters during automatic placement. A floor plan consists of one or more rooms. A room describes an area on one or both sides of the design. You can:

- Draw rooms interactively
- Define rooms by specifying exact dimensions

After defining a room, you assign room contents by including or excluding components and clusters, specifying power nets, or constraining component height or power dissipation. You can also assign Placement Rules to rooms or to image sets in a room.

For more information, see the following sections:

Understanding Component Placement

Evaluating and Optimizing Placement

Additional Placement Procedures

Grouping Components into Clusters

You can group components into clusters based on criteria such as connectivity, net connections, or desired locations. You can also use clusters to fix component positions and rotations with respect to each other or to overlap components in a densely packed area of the design.

You can define clusters by choosing individual components and a cluster type. The general types of clusters you can define include:

- Floor plan
- Super
- Piggyback
- Super Piggyback

You can also form clusters automatically. The tool can form floor plan clusters based on signal or power net connections, or on large component connectivity.

Creating Rooms

Rooms are areas of the design that you define. You can define a room for one or both sides of the design. Use rooms to control where you want the tool to place components or clusters of components during automatic placement operations.

To define a room, you can

- Use Draw Room mode to draw rooms interactively.
- Use Define Room to define the exact dimensions and locations of rooms by specifying X and Y coordinates.

You assign components to rooms by including and excluding specific components or clusters and by specifying priority nets. You can also assign Placement Rules to rooms or to image sets within a room.

A floor plan can consist of one or more rooms.

Assigning Components to Rooms

After you finish Creating Rooms, you can set conditions that guide or constrain component placement in the rooms. These conditions control the following:

- Including or excluding specific components or clusters
- Permitting components on specific power nets
- Setting component height or power dissipation limits

If you include components in a room, or exclude components from a room, you classify them as either hard bound or soft bound.

If you assign power nets to a room, the tool includes components with connections on these nets and excludes all other components. If you do not assign power nets, the tool does not use power net connections as a criterion for including or excluding components.

If you set maximum height, minimum height, or power dissipation limits for a room, every component must have the appropriate physical properties assigned in the tool or in the Design file. The power dissipation limit is the maximum total power dissipation permitted for all components in the room.

For more information about defining properties, see `component_property` and `image_property` commands.

To assign contents for a room

1. Choose *Rules – Room – Contents*.

   The Room Contents Rules dialog box opens.

2. Click *Pick Room*.

   The Pick Active Room dialog box opens.

3. Choose a room name in the Items list, and click *OK*.
4. Disable all room content controls you do not want to use to assign contents to the room.
5. Set one or more room content controls to assign contents to the room. You can
   - ❑ Set any or all *Incl Comp List* and *Excl Comp List* controls
   - ❑ Set the *Room Height* control
   - ❑ Set the *Max Power Dissipation* control

- ❑ Set the *Room Power Net List* control

6. Click *Apply* and repeat steps 2 through 5 to assign contents to another room.

   -or-

   Click *OK* to dismiss the Room Contents Rules dialog box.

After assigning room contents, you can set placement rules for a room and for image sets within a room. You can also use the `room_rule` command to assign room contents.

For example:

```
room_rule memory
(include U1 U2 U16 (type hard))
(exclude remain (type hard))
```

```
room_rule ANALOG (power AGND)
```

```
room_rule height_2 (height .2 -1)
```

You can assign contents when you use the `define room` command to define a room. The following example defines a room and sets a power dissipation limit of 5000.

```
define (room P2W
(rect S4 0.235 0.875 3.135 0.875)
(power_dissipation 5000))
```

Height and power dissipation values must be consistent throughout the design for the system of units you are using.

Setting Placement Rules for Rooms

After defining a room and assigning its contents, you can set Placement Rules for rooms and for image sets in the rooms to guide automatic placement. The rules you can set include:

- Spacing
- Permitted Orientations
- Permitted Sides
- Opposite Sides

Using Additional Placement Procedures

Modifying the Placement Boundary

The placement boundary encloses the area of the design where component placement is permitted. This boundary applies to all signal layers. If you do not define a placement boundary, it uses the routing boundary to enclose the area for interactive and automatic placement.

You can define a new placement boundary or replace the existing placement boundary. When you replace an existing boundary, the tool automatically disbands the old boundary for you.

Defining the Placement Boundary

Either of the following methods can be used to define a placement boundary.

- Use Draw Place Boundary mode to draw the boundary interactively.

  Choose *Define – Place Boundary – Draw Mode* to set [LB] to Draw Mode.

- Use Define Place Boundary to define the exact dimensions and locations of the boundary by specifying X and Y coordinates.

  Choose *Define – Place Boundary – By Coordinates* to display the Define Boundary dialog box.

Using Physical Properties

You can assign physical properties to components and images. Properties assigned to a component override conflicting properties assigned to its image. Physical properties consist of the following:

- Type
- Height
- Power dissipation

You can assign properties in the tool or in the Design file. For information about assigning properties, see the following sections.

- Defining component properties
- Defining image properties

Use physical properties when you want to:

- Assign Placement Rules to image sets on the design or in rooms (see also Setting Placement Rules)
- Select or specify components for exclusive processing in automatic placement operations (see also `select component` command).
- Constrain component height or power dissipation in rooms.

You can assign both known properties and user properties to components and images. If you add, change, or remove known component or image properties, the tool records the change when you save the Placement file or the Session file. The tool does not record user properties in the Placement file or Session file.

Using Separate Front and Back Image Definitions

If you define separate front and back footprints for images in your layout system, the tool uses the correct image definition for the side on which you place a component. All front or back side Placement rules apply.

For more information, see Setting Placement Rules on page 72.

Using Image Site Grids

You can use image site grids to establish a regular spacing pattern between like components. When you define an image site, it supersedes any placement grid you have defined.

To define an image site, choose an image and specify horizontal and vertical spacing values. An image site applies to both the front and back sides of the design.

To set site grids for images

1. Choose *Define – Design Grids*

   The Design Grids dialog box appears.

2. Click the Placement tab.

3. Enter horizontal and vertical grid spacing values in the table on the Grid By Image panel. For each image that you want to assign to a site grid, find the image name in the Image column and, in the same row

   ❑ Enter the horizontal spacing value in the X Grid data entry box.

- ❑ Enter the vertical spacing value in the Y Grid data entry box.

4. Click *Apply* or *OK*.

Note: If you are using both placement grids and site grids, the site grids for SMD images must be multiples of the SMD placement grid, and the site grids for through-pin images must be multiples of the through-pin placement grid.

Using Thermal Constraints

Thermal constraints limit power dissipation in critical areas of the design. You set thermal constraints by:

- Assigning maximum power dissipation values to all components or images in the design.
- Defining rooms for the critical areas of the design.
- Assigning a power dissipation limit to each room.

To set power dissipation values in the tool, you can assign component properties for an individual component or assign image properties for the instances of an image. Component properties override image properties. You can change properties assigned in the Design file from within the tool.

To set a thermal constraint for an area of the design, define a room and specify a maximum total power dissipation limit when you assign components to rooms.

During automatic placement, the tool calculates the total power dissipation of all the components placed in a room.

- If the total power dissipation exceeds the power dissipation limit assigned to the room, the tool displays a warning in the Output window.
- If the total dissipation is less than the assigned limit, the tool can add components to the room until the maximum power dissipation is reached.

Using Height Constraints

Height constraints limit component height in certain areas of the design. You set height constraints by:

- Assigning maximum height value to all components or images in the design.
- Defining rooms for the critical areas of the design.
- Assigning maximum and minimum height a power limits for components in each room.

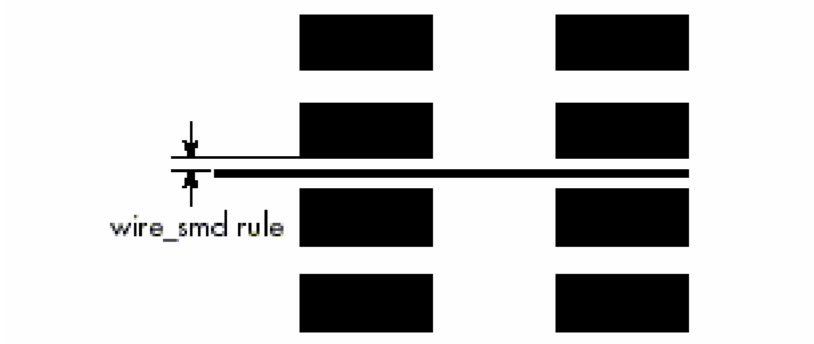
To set maximum height values in the tool, you can assign component properties for an individual component or assign image properties for the instances of an image. Component properties override image properties. Properties assigned in the Design file can be changed from within the tool.

To set height constraints for an area of the design, define a room and specify a maximum height limit when you assign components to rooms.

Placing Components in a Surface Mount Device Design

One of the most important aspects of successful SMD design is choosing a component placement that effectively manages the via space needed to escape SMD pads. If adequate via space between components is not provided, you can expect high conflict counts and unroutes. The following figure shows how the space between the pads of adjacent SMD components should allow at least one via site. For SMD components having a large pin count, more space is required to escape into the design. A minimum of three to four via sites should be allowed between the pads of adjacent components that have large pin counts.

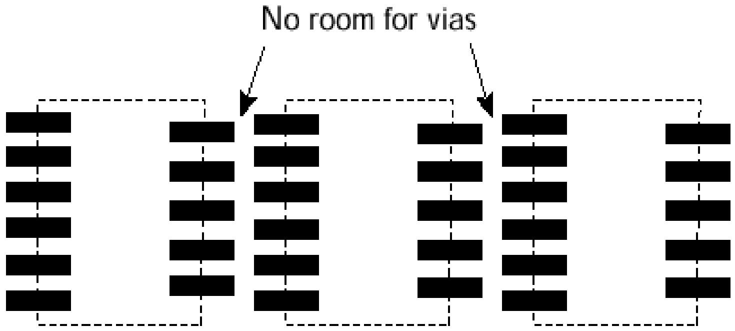
Figure 4-1 Spacing Adjacent SMD Components



Allow for at Least One Via Between Components During Placement.

Do not brickwall SMD components by placing components too close together. The following figure shows an example of brickwalling, where the autorouter has no chance to access the SMD pads from another layer by using vias between components.

Figure 4-2 Brickwall-ing

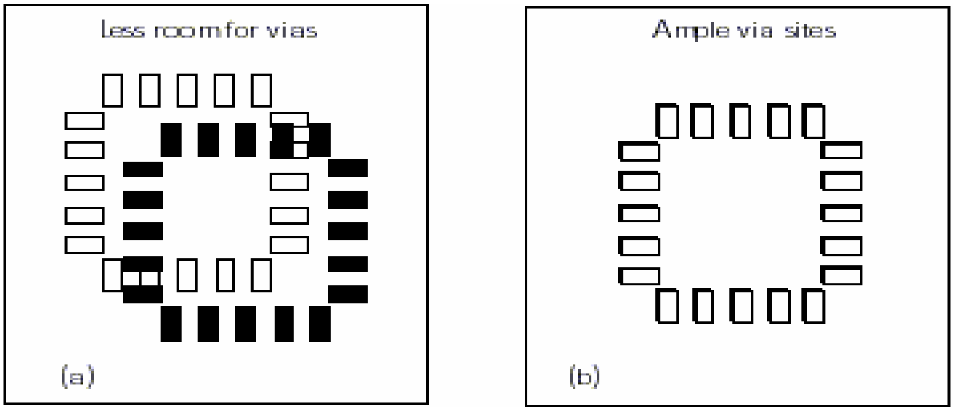


Note: Brickwall-ing SMDs leaves no room for vias between components.

Mounting SMDs Back-to-Back

When SMD components are mounted on both sides of the design, the pads should be opposite each other as shown in the following figures. The goal is to create as many potential via sites as possible.

Figure 4-3 SMDs Back-to-Back



Controlling SMD-to-Via Escapes

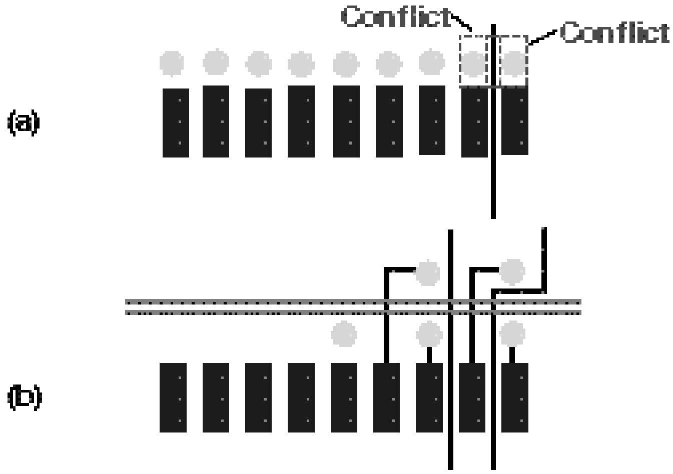
Following are three methods for creating SMD-to-via escapes.

- Prerouting wire-to-via escapes
- Using the Fanout command
- Using no predefined escape wiring

Managing the Via Grid

Avoid creating via barriers. A via barrier occurs when vias are placed so that no wires can pass between them. When a via barrier is created, the autorouter has to work harder, routing is slower, and a poor completion rate results. A via grid that is too fine can create a via barrier. Use the `grid smart` command or choose a via grid spacing that allows two wires between vias. In the following figure, the first illustration shows a via barrier and the second illustration shows the same component with improved via fanout.

Figure 4-4 Improving Via Fanout



Escaping SMD Pads

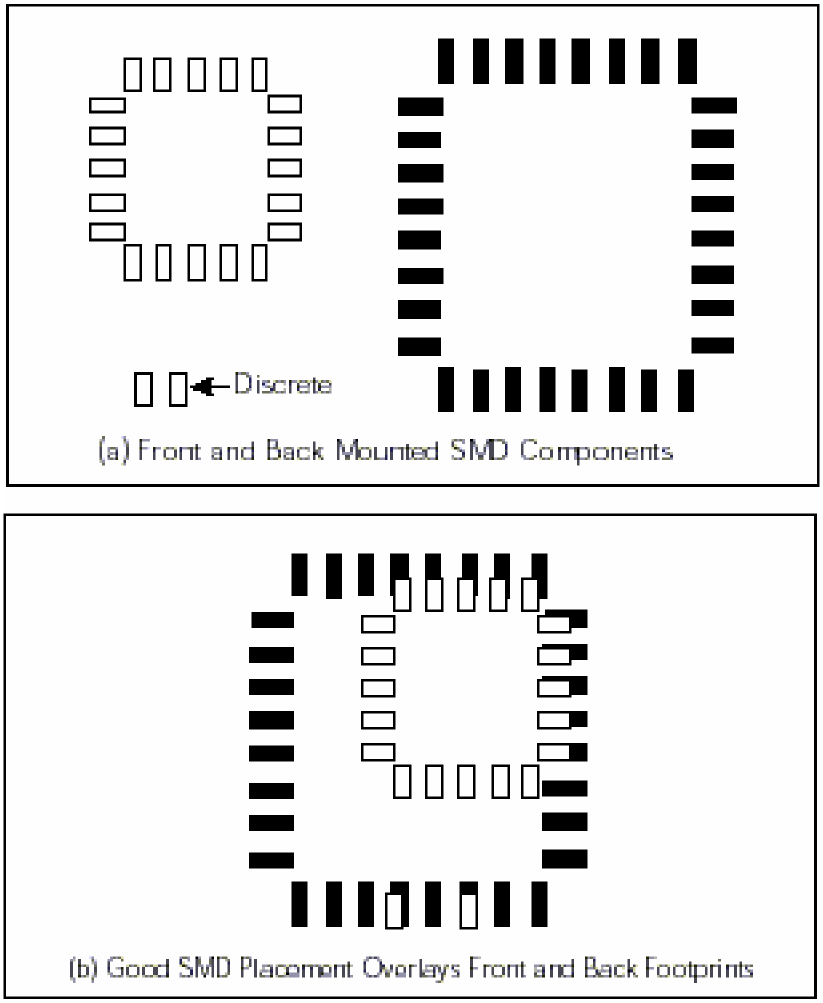
To set a via grid spacing for routing, consider the wire grid spacing and the final via grid spacing, and use the grid smart command.

For example:

```
unit mil
grid smart (wire 1) (via 25)
fanout 5
route 25
```

When you use the `grid smart` command at the beginning of a Do file, it allows two wires between vias, during the initial route passes. If manufacturing rules prohibit setting the via grid to 1 mil, use the smallest grid spacing that is an even multiple of the wire grid and satisfies your manufacturing rules. See Understanding the Batch Script on page 34 for more information.

Figure 4-5 Identical Back-to-Back SMDs



(a) Front and Back Mounted SMD Components

(b) Good SMD Placement Overlays Front and Back Footprints

SMD pad escape perimeters are used when layers are unselected.

All sides of the escape area are equal in length to twice the smd_escape distance. The default smd_escape distance is 0.25 inches (0.635 cm). You can set the smd_escape distance to a different value by using the change command.

For example, if you wanted to change the escape distance to 0.125 inches, you would use the following command:

```
change smd_escape 0.125
```

Meeting Electrical Design Requirements

This application provides some placement tools you can use to satisfy electrical design requirements. You can also use some routing rules to help guide automatic placement.

Using High Speed Constraints to Control Placement

Using Net Priority to Influence Component Placement

Using Split Power and Ground Planes

Associating Components

Placing Decoupling Capacitors

Using High Speed Constraints to Control Placement

If your design includes high-speed circuits with maximum or matched routed length or delay limitations, the tool can observe these restrictions during automatic placement operations. Before placing components, you must use the autorouter `circuit` command to apply the required rules. Examples of autorouter (fast-circuit) commands that apply maximum and minimum length rules are

```
circuit net CLOCKA (length 2000 -1)

circuit net CLOCKB (length 4000 3000)

define (class C1 CLK1 CLK2 (circuit (length 1500 1200)))
```

You can use the Placement Setup dialog box to control compliance with high speed routing rules during placement operations. When High Speed is enabled, the tool follows the high speed rules and places the affected components first. At the beginning of a session, High Speed is enabled by default.

To control placement for high speed routing rules

1. Click *Autoplace – Setup*

   The Placement Setup dialog box opens.

2. In the General tab, click the checkbox to enable or disable High Speed.

   You can also use the `high_speed` command. For example, `high_speed on`.

Using Net Priority to Influence Component Placement

You can influence the automatic placement of components by assigning routing priorities to nets. Components with priority nets are placed earlier during automatic placement. Components that share a priority net are placed closer together.

The tool orders components for placement based on their connectivity, and places the most highly connected components first. When you assign a priority to a net, the tool adds a priority factor to the connectivity. The range of priority values, from lowest to highest, is 1 to 255. The default priority for a net is 10.

If you want to move a component up in the placement order, assign a priority value of 100 or more to one or more of the attached nets. Use a value of 255 for maximum effect.

To apply priority to a net

1. Click the *Routing mode* icon to change to the routing environment.
2. Choose *Rules – Net – Wiring*

   The Net Wiring Rules dialog box opens.
3. Click the *Pick Net* button to open the Pick Active Net for Rules dialog box, choose a net, and click *OK*.
4. Enter a value of `100` or more in the Net Priority data entry box.
5. Click *Apply* or *OK*.
6. Click the *Placement mode* icon to change to the placement environment.

Using Split Power and Ground Planes

Usually, when a design has split power planes, you want to place each component over the plane that supplies its ground or power. The tool enables you to automatically cluster and place components above the region of the plane to which they are connected.

To automatically form clusters for split power planes, you form a cluster for each plane by specifying the power nets for the plane. The tool automatically clusters all the components connected to the power plane, and assigns a name to the cluster.

The tool also automatically defines rooms for the power planes, and assigns each cluster to the appropriate room. The boundaries of each room match the boundaries of the corresponding power plane region.

During automatic placement, the autoplacer places components over the correct region.

Note: When your design uses split power or ground planes, you must define these as split layers in the design layout system so they are included in the Design File.

Associating Components

Use component associations to guide small component placement. You can associate several small components with a large component when you want to place the small components in close proximity to the large component. When small components are not associated with a large component, the autoplacer places them sequentially.

To define component associations

1. Click *Define – Associate Components – By List.*

   The Associate Components dialog box opens.

2. Choose one large component name in the Component list on the left side of the dialog box.
3. Choose one or more small component names in the Component list on the right side of the dialog box.
4. Click *Apply* or *OK*.

You can also use the `associate` command.

For example, if you want to automatically place small components C6 and R3 with large component U1, enter the command:

```
associate U1 C6 R3
```

Note: You can interactively associate components using Associate Components mode. Choose *Define – Associate Components – Mode.*

A component can belong to only one component association. You must disband an association before you can use its components in another association. To disband an association, select its large component.

Component associations work well for automatically placing a large number of small components in close proximity to large components. You can also create and apply small component patterns to place small components near large components together when their relative positions and rotations are important.

Placing Decoupling Capacitors

Usually, you place decoupling capacitors separately from other small components. You can set automatic placement options to place capacitors close to the power pins on large components, and when possible, under SMD pads on the opposite side of the design.

You can associate components when you want to place small capacitors near a particular large component. Alternatively, after placing large components, you can use small component patterns to place capacitors in specific locations and with specific rotations relative to the large components.

Meeting Manufacturing Design Requirements

You can meet manufacturing requirements by grouping components and specifying component placement side, orientation, and spacing constraints. When two or more components require fixed locations and orientations with respect to each other, you can define a super cluster.

You can control placement side, orientation, and spacing by setting spacing rules, permitted orientation rules, and permitted side rules for individual components or images. You can also set individual pad edge and body edge spacing rules.

Additionally, you can set rules that constrain opposite side placements of large or small components.

In addition to the placecement grids for SMD and through-pin components, you can set site grids for specific images.

Meeting Thermal Design Requirements

You can observe thermal constraints by setting power dissipation limits in critical areas of the design. You can limit total power dissipation in these areas by:

- Assigning power dissipation values to all components using *Define – Properties – Component* or to images using *Define – Properties – Image*.
- Defining rooms
- Applying power dissipation constraints to the rooms

When you place components automatically, the autoplacer calculates the total power dissipation of all the components placed in a room. If this value exceeds the power dissipation limit assigned to the room, the autoplacer displays a warning in the output window. If the total dissipation is less than this limit, you can add components to the room until you reach the limit.

Meeting Mechanical Design Requirements

Mechanical requirements are dictated by physical constraints or limitations of the design. Before you begin automatic placement, you should preplace connectors and other critical components that must occupy specific locations of the design.

After preplacing components, you should lock them to prevent the tool from moving them during automatic placement operations. You can also lock gates, subgates, and pins that you do not want the tool to swap during automatic swapping operations.

If you place capacitors or discrete components at specific locations, you should also lock them before placing other small components.

You can observe height constraints by setting height limits in critical areas of the design. You can also limit the minimum and maximum component heights in these areas by:

- Assigning maximum height values to all components using *Define – Properties – Component*) or to images using *Define – Properties – Image*.
- Defining rooms
- Applying component height constraints to the rooms.

If you need to place two or more components in overlapping positions, you can define piggyback clusters with overlapping components.

Using Interactive Placement Tools

About Interactive Placement

You use interactive placement tools to:

- place or move components that are outside the placement boundary to positions inside the boundary.
- relocate components that are inside the placement boundary.
- push components into densely packed areas of the design.
- trade locations and align components to optimize placements and improve routability.
- swap nets on gates, subgates, and pins to optimize placements and improve routability.

Before you begin interactively placing components, correcting rule violations, or changing placements to improve routability, you should set up your placement environment.

You set interactive placement modes by using the Interactive Place menu. You can also set some frequently used modes by clicking icons on the tool bar.

Note: If rule checking is enabled, you cannot violate any placement or routing design rule during interactive placement. If you disable rule checking, you are allowed to violate design rules. After placing components, you should enable rule checking once again to check for any rule violations.

Placing Components Interactively

You can place components interactively at locations you choose by using one of the Place Components [LB] modes. The Place Components modes include:

- Place Connect Mode, which enables you to place components by connectivity *without* assistance from the tool. You click on the location where you want to place the component

  Press [RB] and choose *Place Components – Place Connect Mode*

- Guided Place Connect Mode, which enables you to place components by connectivity *with* assistance from the tool. The interactive placer recommends a location. You can choose this location or click to place the component at a different location.

  Press [RB] and choose *Place Components – Guided Place Connect Mode*.

- Place List Mode, which enables you to place components from a list. You specify the components you want to place. You can define the placement order, or let the tool choose the placement sequence based on connectivity.

  Press [RB] and choose *Place Components – Place List Mode*

You can also interactively place components at exact locations by specifying X and Y coordinates.

Relocating Components

After placing components inside the design placement boundary, you can relocate components to optimize placements and improve routability.

You can use the following modes:

- *Move Component mode* to move components from one location to another within the boundary
- *Pivot Component mode* to rotate components relative their current orientations
- *Flip Component mode* to flip components from one side of the design to the other.

By default, there must be enough space to relocate the components without causing rule violations. However, you can enable the Shove for Move setup option that enables you to relocate components.

Pushing Components

You can use *Push Component mode* to push components out of the way while sliding a component or group of components from one location to another within the placement boundary. You cannot push components into a keepout, via, or wire.

You can move a single component, a group of components, or components within rectangular areas.

When rule checking is enabled, you cannot push components over the placement boundary or to any locations that violates current placement or clearance rules. However, Push Component mode does not recognize spacing rules assigned to rooms or image sets in rooms.

To push components

1. Click to Push Component mode by doing one of the following:
   - ❑ Click the Push Components icon on the tool bar.
   - ❑ Right-click and choose Push Comp Mode.
2. Click the component you want to slide.
   - ❑ To slide a single component, click the object.
   - ❑ To slide a group of adjacent components, drag the pointer to draw a rectangle around the components. A component does not have to be completely enclosed by the rectangle.
   - ❑ To slide multiple components or groups of components, press the [Shift] key, click on one or more components, drag the pointer around components in one or more rectangular areas, or both, and release [Shift].

   Each of these actions attaches a ghost image of the component or components to the pointer, and display component reference designators, X-shapes to mark component working origins, and guides to show connectivity to other components.
3. If you want to choose additional components or exclude components from the slide, press the [Shift] key and either:
   - ❑ Click on or drag the pointer around unselected components you want to include in the slide.

   or

   - ❑ Click on or drag the pointer around selected components you want to unselect and exclude from the slide.

   The ghost image disappears until you release [Shift].
4. Release [Shift] if pressed, move the pointer into an area occupied by one or more components.

   As you move the pointer, the tool slides the ghost image and pushes aside ghost images of other components that are in the way. You can slide components horizontally or vertically, but not both at the same time
5. Click to replace the components.
6. Repeat steps 2 through 5 to slide other components or groups of components.
7. To cancel a push operation without moving the objects, press [RB] and choose *Cancel*.

Note: You can reverse push component operations using *Edit – Undo*. Use *Edit – Redo* if you want to reapply operations that you reversed with *Edit – Undo*.

Trading Components

You can use Trade Component Mode to trade positions between two components. you can trade like or unlike components.

The tool trades components by switching the positions of their working origins. When Rule Checking is enabled, the tool trades component positions unless the trade violates a Placement Rule. Disable Rule Checking if you want to trade components despite placement rule violations.

To trade component positions

1. Set the Trade Component mode by doing one of the following:
   - ❑ Click the Trade Component icon on the tool bar
   - ❑ Right-click and choose Trade Comp Mode.
2. Click the two components you want to trade.
3. Repeat step 2 for each additional trade you want to perform.

You can trade adjacent components with a single action by dragging the pointer over them. If you attempt to trade more than two components, the tool terminates the operation and displays an error message in the output window.

You can reverse trade component operations using *Edit – Undo*. Use *Edit – Redo* if you want to reapply operations that you reversed with *Edit – Undo*.

Aligning Components

You can use Align Component mode to align components within the design boundary. The tool aligns the components with respect to an alignment reference that you can change as a setup option.

Component and pin origins are defined in the layout system and can be different for each image or component type in a design. By default, the tool aligns components with respect to their upper left pins.

You align components to a reference component, which does not move during the alignment operation. The tool first attempts to align the components vertically. If this fails, the tool attempts to align the components horizontally.

The tool cannot complete an alignment if it causes a Placement Rule violation.

To align components

1. Align Components by doing one of the following:
   - ❑ Click the Align Components icon on the tool bar.
   - ❑ Right-click and choose Align Comp Mode.
2. Click the component you want to align, or drag the pointer to enclose several components.
3. Click the component you want to use as the alignment reference.

You can also use the `align` command to align components. First, select the components you want to align. Then, enter the align with the reference designator of the reference component.

For example:

```
select U1 U2 U4
align U3
```

Note: Alignment by pin reference uses the origin of the specified pin. If different components have different locations for their pin origins, alignment by pin reference can produce unexpected results.

You can reverse alignment operations using *Edit – Undo*. Use *Edit – Redo* if you want to reapply operations that you reversed with *Edit – Undo*.

Associating Components Interactively

You can use the Associate Component mode to interactively associate a large component with one or more small components. Associations are usually used when you want to place several small components close to a large component with which they share a specific function.

For instance, you can create an association of resistors with a large integrated circuit component. Then, when you automatically place large components, the tool also places the resistors next to their large component associates. Component associations do not affect interactive placement or other automatic operations.

To associate components

1. Choose *Define – Associate Components – Mode*.
2. Click a large component.

3. Click one or more small or discrete components.
4. Press [RB] and choose *Associate* in the Associate Comp popup menu.
5. Do one of the following:

    ❑ Repeat steps 2 through 4 if you want to form another component association.

    ❑ Right-click and choose *Reset* to exit Associate Components mode and return to Measure mode.

    You can also:

    ❍ Define component associations by specifying component reference designators.

    ❍ Use a component in only one association at a time. If you want to use an associated component in a different association, you must first disassociate the current association.

Swapping Gates and Pins Interactively

You can swap net connections between functionally equivalent gates or pins to reduce circuit congestion, Manhattan lengths, and the number of vias required for routing.

You can also set [LB] to interactively select, swap, and unselect connections on equivalent gates, subgates, pins, or terminators.

To interactively swap gate, subgate, pin, or terminator connections

1. Press [RB], choose *Swap*

    The Swap menu appears.

2. Choose the Swap mode for the connections you want to swap. You can choose *Gate Mode*, *Sub-Gate Mode*, *Pin Mode*, or *Terminator Mode*.
3. Drag the pointer over an area that contains the gates where you want to swap net connections.

    This selects, swaps, and unselects connections within the area.

4. Repeat step 2 to swap connections in a different area of the design.
5. Repeat steps 1 through 3 to change the swap mode and continue swapping net connections.

You can exit a swap mode by clicking *View – Measure Mode*.

You can also set [LB] to swap gate or pin connections by using the mode `swap` command.

For example:

```
mode swap_gate
mode swap_subgate
mode swap_pin
mode swap_terminator
```

Generally, the tool swaps connections when the swap results in reduced Manhattan lengths. However, if you select only two swappable objects, the tool performs a swap even if it increases the Manhattan lengths.

You cannot swap locked gates, subgates, pins, or terminators.

The necessary package swap information (`<gate_pin_swap_code>`) for each swappable pin must be translated from the component libraries on your layout system and included in the network section of the Design File.

5

Routing Connections

In this chapter . . .

Understanding Routing

In general, routing a design can involve some or all of the following procedures:

Setting the Routing Controls

Using Smart Route

Running the Autorouter

Evaluating the Progress of Autorouter

Editing the Do File

Controlling Routing Topologies

Applying Convergence Techniques

Layer Set Rule Checking

Differential Pair Checking

Using Interactive Routing Tools

Setting the Routing Environment

You can use the *Autoroute – Setup* dialog box to set global routing options and constraints, such as wire and via grids, wire widths and clearances, routing direction, and fences. You can use various commands from the Define menu to define routing parameters, groups of objects to be routed, routing regions, keepouts, and fences.

Setting Routing Rules

You set rules to control the autorouting process. The rules you use can include wire width, clearance, impedance control, cross-talk control, and timing control. You can apply most of these rules globally to the entire design, or to specific layers, nets, or groups of objects, based on the Routing Rules Hierarchy. The tool obeys the rules you have set during interactive routing operations, and prevents design rule violations provided that Rule Checking is enabled.

Routing Interactively

You can route connections interactively using the Interactive Routing menu. The various interactive routing modes provide efficient ways to manually route connections in order to meet critical design constraints.

Setting the Routing Controls

You can control the autorouting process in a number of ways to achieve optimal routing results for a wide variety of design requirements. The following general controls are available.

Grid Controls

Setting Wire and Via Grids

Layer Controls

Controlling the Routing Layers

Layer Set Routing

Preventing Routing on External Layers

Net Controls

Defining a Class of Nets

Defining a Group of Fromtos

Defining a Group Set

Routing Differential Pairs

Routing Several Nets as a Bundle

Separating Analog and Digital Signals

Separating Clock Lines

Enabling Same Net Checking

Prioritizing How Nets are Routed

Controlling Virtual Pin Interconnects

Routing Topology Controls

Dynamic Virtual Pins

Defining a Branch Topology

Via Controls

Controlling Via Use

Setting Wire and Via Grids

You can set PCB wire grids and PCB via grids that apply to both autorouting and interactive routing operations. Each grid can be uniform or nonuniform and you can set both grid spacings as well as grid offsets. Grid offsets are measured from the design origin.

For wires, you can also set major grid marks on the wire grid and control how the grid displays. For vias, you can also set separate grids for individual vias.

During interactive routing, you can change PCB wire and via grids by resetting grid values. After you change a grid value, the interactive router follows the new grids during routing. Wires and vias digitized before the current grid change are not affected.

Grids are not required. If you set the wire grid spacing to zero, you can edit and route the wires in a gridless mode. If you set the via grid value to zero, you can position vias without using a grid.

You can set a global wire grid for the design or separate wire grids for individual routing layers. You can also set a major grid for the wire grid and control how the grids are displayed. Grids set for individual layers override the PCB wire grid.

To set the PCB wire grid

1. Choose *Define – Design Grids*

   The Design Grids dialog box appears.

2. On the Wire tab, enter grid spacing and grid offset values on the Routing Wire Grid panel.

   To set the grid spacing

   - ❑ Enter the X direction grid spacing value in the X Grid data entry box.
   - ❑ Enter the Y direction grid spacing value in the Y Grid data entry box.

   A value of -1 means the grid spacing in that direction is undefined.

   If you want to set optional grid offsets, you can

   - ❑ Enter an X direction offset value in the X Offset data entry box.
   - ❑ Enter a Y direction offset value in the Y Offset data entry box.

   A value of 0 means no offset for that direction is defined.

3. Click *Apply* or *OK*.

You can also set wire grids for individual routing layers by using the `grid wire` command. Wire grids set for individual layers override the global wire grid.

You can display major grid points on the wire grids. The major grid spacing is a multiple of the wire grid spacing. You can also display major grid points in a different color from the wire grid color.

You can also control the visibility of the wire and major grids, as well as whether the grids are displayed as lines or dots.

To set the major grid and control how wire grids are displayed

1. Choose *View – Display Grids*

   The Display Grids dialog box appears.

2. Do the following on the Routing Major Grid panel.

   a. Enter a value in the Factor data entry box if you want to display major grid marks on the wire grid. This value is the number of wire grid points between each major grid point.

   b. Enable or disable Grid to control whether the wire grids are visible.

   c. Enable or disable Factor to control whether the major grid marks are visible on the wire grids.

   d. Click the As popup menu and choose either Lines or Dots to display the wire grids as lines or dot. The default is Lines.

3. Click *Apply* or *OK*.

Note: You can set a global via grid for the design or separate via grids for individual vias. Grids set for individual vias override the PCB via grid.

To set PCB via grids

1. Choose *Define – Design Grids*

   The Design Grids dialog box appears.

2. Click the Via tab.

3. Enter grid spacing and grid offset values on the Grid For All Vias panel.

4. Do the following to set the grid spacing.

a. Enter the X direction grid spacing value in the X Grid data entry box.

b. Enter the Y direction grid spacing value in the Y Grid data entry box.

A value of -1 means the grid spacing in that direction is undefined.

c. If you want to set optional grid offsets, do the following.

❍ Enter an X direction offset value in the X Offset data entry box.

❍ Enter a Y direction offset value in the Y Offset data entry box.

Note: A value of 0 means no offset for that direction is defined.

5. Click *Apply* or *OK*.

To set grids for individual vias

1. Choose *Define – Design Grids*

The Design Grids dialog box appears.

2. Click the Via tab.

3. Enter grid spacing and grid offset values in the table on the Grid For Via Name panel. For each via that you want to define a via grid for, find the via ID in the Via column. In the same row, you can:

❑ Enter the X direction grid spacing value in the X Grid data entry box.

❑ Enter the Y direction grid spacing value in the Y Grid data entry box.

❑ Enter an X direction offset value in the X Offset data entry box.

❑ Enter a Y direction offset value in the Y Offset data entry box.

Note: A grid spacing value of -1 means the grid spacing in that direction is undefined. A grid offset value of 0 means no offset for that direction is defined.

4. Click *Apply* or *OK*.

Note:

❑ Make sure you enter grid values that are scaled for the units you are using.

❑ You can display wire and via grids by using the Layers panel.

Defining a Class of Nets

A class is a convenient way of managing more than one net at a time. After defining a class, you can select the class of nets for routing and assign the same rules to every net member of the class.

You associate nets by assigning them a class name. For example, to associate nets and assign them the class name C1, you could add the following command to your DO file substituting net names in your design for `<net id>` (see Editing the Do File on page 193).

```
define (class C1 <net id> <net id>)
```

Points to consider when working with class of nets:

- A class is not limited to two nets. You can include as many nets in the class as required.
- A net can be a member of one or more classes.
- You can apply the same rules to several nets by assigning them to the same class.

You use the `define` command to add or remove nets in a class *without disbanding the class*. Use the `forget` command if you want to disband the class and any rules you applied to the class.

For example, to disband the class C1, you would enter the following command:

```
forget class C1
```

Defining a Group of Fromtos

You associate fromtos (single pin-to-pin connections on a net) by assigning them a group name. For example, to associate fromtos and assign them the group name `G1`, add the following command to your Do File (see Editing the Do File on page 193).

```
define (group G1 (fromto U1-6 U3-4) (fromto U1-6 U5-13))
```

Points to consider when working with class of nets:

- A group is not limited to two fromtos. Include as many fromtos as you require.
- A fromto can be in more than one group.
- After you define a group, you can assign rules to the group by using the commands of the *Rules – Group* menu.
- You can apply rules to single fromtos.

To disband a group, use *Define – Group – Define/Forget By List* to open the Define/Forget Group by List dialog box.

Defining a Group Set

You can associate groups of fromtos by assigning them a group set name.
For example, to associate groups `G1` and `G2` to a group set called `SET1`, add the following commands to your Do file.

```
define (group G1 (fromto U1-6 U3-4) (fromto U1-6 U5-13))
define (group G2 (fromto U1-4 U3-9) (fromto U1-16 U3-7))
define (group_set SET1 G1 G2)
```

Note: You can have multiple groups in a group set. Include as many groups as you require.

To disband a group set, use the command `forget group_set`.

Applying Standard Routing Passes

The `route` command runs basic autorouting passes. With each routing pass, the autorouter attempts to route connections that are not yet routed and reroute connections that are involved in conflicts or are close to wires involved in conflicts.

If you did not select any connections, the autorouter attempts to route all connections defined in the network, except those that are fixed or protected. If you select any connections, the autorouter attempts to route only those connections that you selected.

A minimum of 25 passes is suggested for the initial series of routing passes. After these initial 25 routing passes, you should run two clean passes by using the `clean` command. The `clean` command rips-up and reroutes every connection, removes unnecessary vias and bends, and alters the routing problem by making new or different routing channels available for the next series of route passes. You will see a noticeable improvement in the routing quality after the clean passes.

Place the following `route` and `clean` commands in the Standard Routing Commands section of your Do file (see Editing the Do File on page 193).

```
route 25
clean 2
route 50 16
clean 4
```

During the second series of routing passes, the autorouter often achieves routing completion. The `route` command applies 50 additional passes and sets the pass counter to 16. A value of 16 for the pass counter decreases conflict cost and is usually used when you apply additional routing passes after the initial series of 25 routing passes.

If the autorouter completes the wiring in fewer than the number of passes you specified with the `route` command, the remaining router passes are not executed. Instead, control passes to the next command in the Do file.

The four clean passes near the end of an autorouting session help to minimize the number of vias used and improve the overall quality of routing.

Note: After running the `route` and `clean` commands, you should use the routing status report to monitor and analyze the autorouting progress and determine when you need to adjust your routing strategy. See Evaluating the Progress of Autorouter on page 207 for more details.

You can read the status report any time after running the autorouter. You can also read the status report during a run by pausing the autorouter (click *Pause* in the control area) any time after the first routing pass.

Routing Differential Pairs

Generally, differential pair routing spans the entire task range. You perform many of the same tasks as when routing other nets, such as fanout, general routing and cleanup.

Typical steps used in routing differential pairs are:

1. Defining the Differential Pair
2. Setting Differential Pair Rules
3. Fanout
4. Gathering the Nets Together
5. Making adjustments to ensure the best possible route solution.
    a. Coupling and Uncoupling Events
    b. Phase Control
    c. Getting through Other Pins
    d. Turning Corners Differentially

Defining the Differential Pair

You define a differential or balanced pair by choosing *Define – Net Pair – Define/Forget By List*, select two nets in the Define/Forget Net Pairs dialog box and clicking the *Create Pair* button. See the procedure for *Define – Net Pair – Define/Forget By List* in the *Allegro PCB Router Command Reference* for further details.

You can also use the `define_pair` command to do this. For example, to define a differential pair that consists of nets CLK1 and CLK2 with a gap of 5 mils and a differential pair that consists of nets CLK4 and CLK5 with a gap of 7 mils, add the following commands to your Do file.

```
define (pair (nets CLK1 CLK 2 (gap 5)))
define (pair (nets CLK4 CLK5 (gap 7)))
```

See the `define_pair` command in the *Allegro PCB Router Command Reference* for further details.

You can use a ? as a wildcard character to define nets in a pair. For example, the following command will pair nets A1+ with A1-, A2+ with A2-,A3+ with A3-, and so forth.

```
        define (pair (nets A?+ A?-))
```

Differential pair routing is followed only when the appropriate license is installed (see Understanding Licensing on page 18 for more information).

Rules

The following rules represent variables that may be adjusted in a design with differential pairs. See the `routing_rules` command in the *Allegro PCB Router Command Reference* for details on each rule.

- edge_primary_gap
- diffpair_line_width
- neck_down_gap
- neck_down_width
- edge_coupling_tolerance_plus
- edge_coupling_tolerance_minus
- ignore_gather_length
- max_uncoupled_length

- phase_control
- phase_tolerance
- diffpair_group_level

Rules Hierarchy

Differential pair rules are implemented in a hierarchical fashion, the same as timing and shielding rules. Therefore, you can specify different diff pair rules at different rule levels (where allowed) that are governed by rule precedence.

Rule levels where diff pair rules are not allowed are:

- Class to Class
- Class to Class Layer
- Padstack
- Region

For example, specifying each of the following rules:

```
rule pcb (edge_primary_gap 8)
rule class CLASS1 (edge_primary_gap 10)
rule net NET1 (edge_primary_gap 5)
```

would mean that every diff pair on the design would get a primary gap of 8. However, due to rule precedence, any diff pair that has a net in CLASS1 would get a primary gap of 10, and any diff pair with NET1 as one of its members would get a primary gap of 5.

Also, values for *wire gap* defined in the rule levels take precedence over any of the *wire-to-wire spacing rules* when routing the diff pair. Therefore, a diff pair primary gap defined at the pcb level would be used instead of a wire-to-wire clearance defined at the fromto level. If no diff pair primary gap values are defined at any level of the rule hierarchy, then the wire-to-wire clearance values are used.

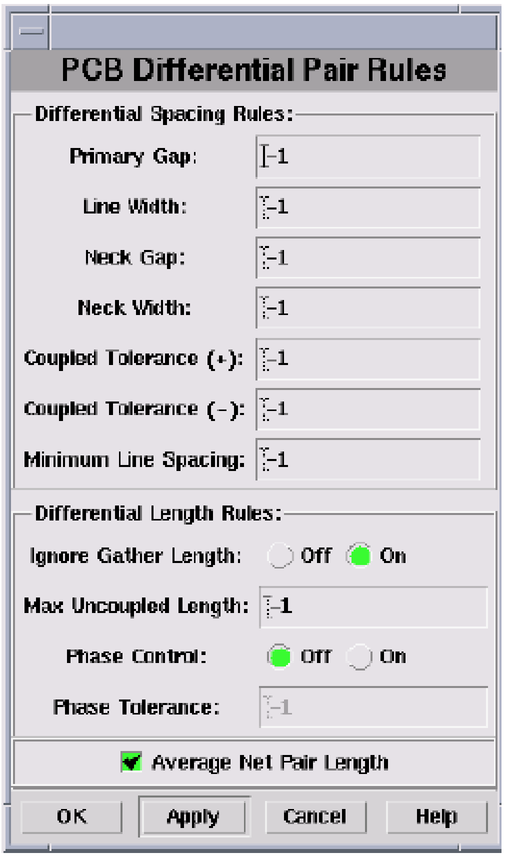
Rule Conflicts

Conflicting rules can occur. For example, if NET1 is in CLASS1, and NET2 is in CLASS2, the nets can be paired, and there may be different pair rules specified for the two classes. In these cases, the “most conservative” rule is applied.

Setting Differential Pair Rules

You set diff pair rules at the different rule levels by using Differential Pair Rules dialog boxes. You access these dialog boxes by choosing *Rules – <rule level> – Differential Pair* from the router main menu. An example of one of these dialog boxes is shown in the following figure.

Figure 5-1 PCB Differential Pair Rules Dialog Box



PCB Differential Pair Rules

Differential Spacing Rules:

Primary Gap: -1

Line Width: -1

Neck Gap: -1

Neck Width: -1

Coupled Tolerance (+): -1

Coupled Tolerance (-): -1

Minimum Line Spacing: -1

Differential Length Rules:

Ignore Gather Length: ○ Off ● On

Max Uncoupled Length: -1

Phase Control: ● Off ○ On

Phase Tolerance: -1

☑ Average Net Pair Length

See the procedures for *Rules – PCB – Differential Pair* in the *Allegro PCB Router Command Reference* for further information on setting diffpair rules.

Applying Layer and Length Rules

You can set layer and length rules for a differential pair by assigning the rules to one net of the pair or by including the net pair in a class and assigning rules to the class. For example, to set layer rules for a pair that consists of nets CLK1 and CLK2, and to assign length rules to a class that consists of net pair CLK1 and CLK2 and net pair CLK4 and CLK5, add the following commands to your Do file.

```
define (pair (nets CLK1 CLK2 (gap 5)))
define (pair (nets CLK4 CLK5 (gap 7)))
circuit net CLK1 (use_layer INT3)
define (class DIF_PR CLK1 CLK2 CLK4 CLK5)
circuit class DIF_PR (length 6100 6000 (type actual))
rule class DIF_PR (length_amplitude 200)
rule class DIF_PR (length_gap 30)
```

Disbanding Differential Pairs

You disband a differential or balanced pair by choosing *Define – Net Pair – Define/Forget By List*, selecting the pair in the Define/Forget Net Pairs dialog box and clicking the *Forget Pair* button. See the procedure for Define – Net Pair – Define/Forget By List in the *Allegro PCB Router Command Reference* for further details.

You can also use the `forget_pair` command to do this. See the `forget_pair` command in the *Allegro PCB Router Command Reference* for further details.

Routing Several Nets as a Bundle

When you want to route a bus or a topologically compatible set of nets together, you can define a bundle, assign nets to the bundle, and specify gaps and routing layers for the bundle. The autorouter routes all the nets of the bundle over the specified layers with the same topology and maintains the gap between the nets, except to connect to pins or to diverge to avoid obstacles.

Bundles behave like differential pairs, except they are not restricted to two nets. Bundles follow the same rule hierarchy as differential pairs, and you define rules similarly.

To create bundles, use the `define bundle` command.

Note: Use the `forget bundle` command to disband a bundle.

Other commands you can use are `select bundle`, and `unselect all bundle`, and `report bundle`.

You can employ bundle routing only when the appropriate license is installed (see Understanding Licensing on page 18 for more information).

Reducing or Enlarging Wire Width on a Segment

You can reduce or enlarge a wire segment connecting to a pin so that it is the same width as the pin by using the `pin_width_taper` rule. By default, the autorouter reduces the wire segment connecting to a pin. The modified wire segment is from the pin to the first bend. The pin with the narrowest width is used.

For example, to enlarge all wire segment widths to the same width as the connecting pin, add the following command to your Do file (see Editing the Do File on page 193 for more information).

```
rule pcb (pin_width_taper up)
```

The autorouter enlarges the wire segment if no clearance violations to adjacent pins occur. Pin width tapering occurs during the autorouting phase by default when you define a `pin_width_taper` rule. However, you can defer pin width tapering to the post-processing phase by using the `set search_tapering off` command.

Important

> If the pin width is smaller than the wire width assigned by a pcb or layer rule, tapering does not occur.

Separating Analog and Digital Signals

You can define an area of the design called a fence, and use the fence to force the autorouter to route only the connections within the fence and to route other connections that do not cross the fence boundary.

There are two types of fences, hard and soft. To separate analog and digital signals use a soft fence. A soft fence is transparent to connections that have one pin inside and one pin outside the boundary. In addition to defining fences to enclose analog and digital connections, you must enable the `soft_fence` control. For example, to define two fences, replace the <X Y> coordinates with the correct values for your design, and add the following commands to your Do file.

```
set soft_fence on
fence <X Y> <X Y>
fence <X Y> <X Y>
```

where:

- The <X Y> values are the coordinates for diagonally opposite corners of a fence boundary.

- Fences that overlap merge to form a polygonal fence region.

Separating Clock Lines

You can control wire-to-wire clearances between nets in a class and all other nets, and between the nets in one class versus the nets in other classes. A class clearance rule sets a minimum clearance between the nets in the class and all other nets in the design. A class-to-class clearance rule sets the minimum clearance between the nets in one class versus the nets in the other named classes.

For example, to define two classes CLK1 and CLK2, set a clearance rule of 20 between CLK1 and all other nets, set a clearance rule of 25 between CLK2 and all other nets, and set a clearance rule of 30 between the nets in class CLK1 and the nets in class CLK2, add the following commands to your Do file (see Editing the Do File on page 193 for more information).

```
define (class CLK1 ck0 ck1)
define (class CLK2 ck2 ck3)
rule class CLK1 (clearance 20 (type wire_wire))
rule class CLK2 (clearance 25 (type wire_wire))
rule class_class CLK1 CLK2 (clearance 30 (type wire_wire))
```

Note: To set a clearance rule between nets of the same class, use a `class_class` rule and include the class name twice. For example:

```
rule class_class CLK1 CLK1 (clearance 15 (type wire_wire))
```

Controlling the Routing Layers

You can control which layers you use to route certain nets. If you unselect a layer, that layer is not used for routing.

Use the `direction` command to control routing layers:

See the `select` command to specify the routing layer for groups, nets, and classes of nets. See also the `unselect` command to route with fewer routing layers.

Note: If SMD components are mounted on a layer that is unselected, the autorouter still escapes the SMD pins to access internal layers. The autorouter uses a default SMD escape distance of 0.25 inches unless you override it by using the `change smd_escape` command.

Enabling Same Net Checking

A same net violation occurs when two objects in the same net are too close to each other. The route, clean, critic, and check commands can check rule violations on the same net. A yellow box appears around the same net violation. To enable same net checking, enter the following command:

```
set same_net_checking on
```

The default is `off`.

- The `via_via` and `via_via_same_net` clearance rules are always checked and are not affected by this control.
- You can also enable or disable same net checking in the Interactive Routing Setup dialog box. During interactive routing, same net checking applies only in Edit Rout mode.

Limiting the Wrong Way Routing Distance

When you apply the `limit_way` rule, the autorouter uses a via to change the routing direction if the wrong way distance exceeds the `limit_way` rule. You can route critical nets and net classes with the shortest possible paths by limiting wrong way routing and assigning a high routing priority. This method can result in an increase in vias but reduces wrong way routing, which can add extra wire length. For example, to route nets in class CRITICAL with the shortest possible paths, add the following commands to your Do File (see Editing the Do File on page 193).

```
define (class CRITICAL RST CLK)
rule class CRITICAL (limit_way 200)
circuit class CRITICAL (priority 255)
```

Prioritizing How Nets are Routed

You can assign priority to nets and classes of nets to control when they are routed relative to other nets in the design. Priority values can range from 1 to 255. A net with a priority value of 255 has the highest routing priority. The default priority value for all nets in a design is 10.

For example, to route net NET2 before all others, and then route the net class CLK before any remaining nets, add the following commands to your Do file.

```
circuit net NET2 (priority 255)

circuit class CLK (priority 235)
```

Note: If you assign priorities to several nets, separate each entry by 10 or more. Otherwise, the autorouter might consider the nets to have the same priority when it computes its internal priority based on the rules assigned to the net.

Controlling Virtual Pin Interconnects

Virtual pin interconnects are biased toward source-virtual pin branches as specified by the *source_seg_ratio* rule (a.k.a. 80% rule). This improves overall results by allocating the greater proportion of a net delay/length to the net portion common to all loads on the net and also conforms to current best-practices in electrical engineering. See Controlling Routing Topologies on page 172 for more information on using virtual pins.

Setting the source_seg_ratio Rule

The source_seg_ratio rule is set at a default of 80%. However, you can change it to whatever value you think works best for your design. Generally, you set this rule before defining the virtual pins in your net. As an alternative, you can experiment with different settings during a session.

To experiment with source_seg_ratio during a session, you must:

1. Forget the nets.
2. Set the `source_seg_ratio` rule.
3. Define the virtual pins before routing.

To set the source_seg_ratio rule:

➤ Enter the following command in the router console window.

```
rule source_seg_ratio <ratio>
```

`where <ratio>` is a percentage between 1 and 99.

Layer Set Routing

During automatic routing, a layer change may be necessary in order to complete the route of certain traces. These layer changes, if uncontrolled, can have an adverse affect upon impedance depending on where the new layer is located in the stack-up. Furthermore, discontinuities caused by the impedance changes in these traces can result in reflections and other undesirable side effects that may lead to instabilities in the system.

To eliminate this problem, the router enables you to constrain these traces to route on specific layers that share the same characteristic impedance. This technique is known as *layer set routing*.

Note: Layer set routing is not currently supported in PCB Editor.

Setup

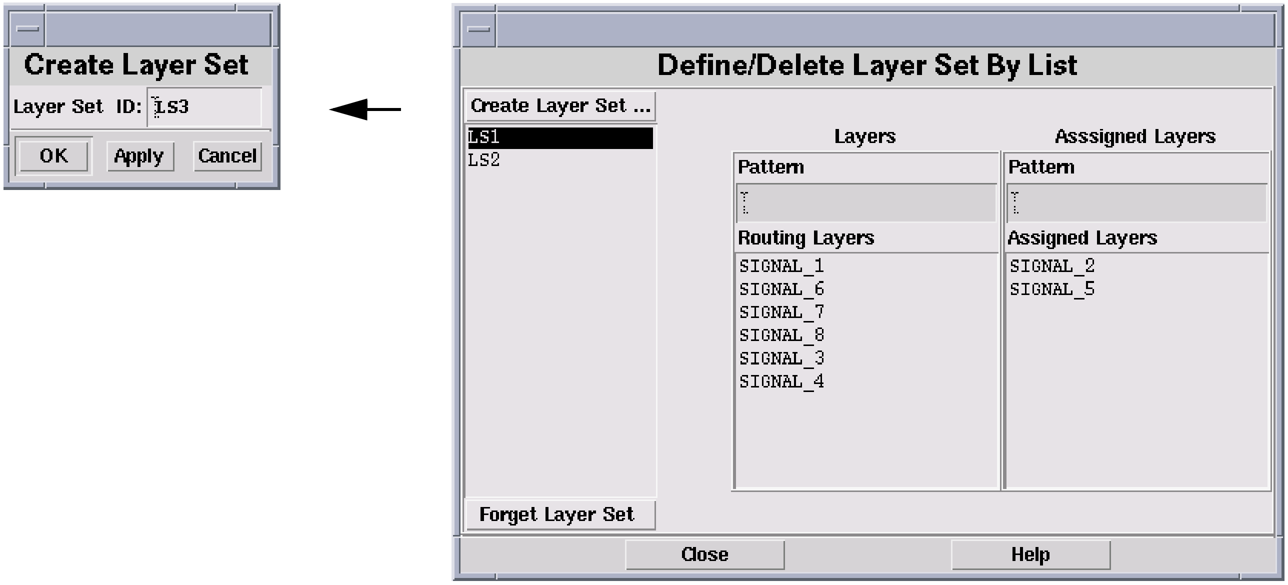
The following general steps are required to set up a design for layer set routing.

- Define one or more layer sets with associated signal layers.
- Apply layer set rules to the target nets at the appropriate rule level.

Defining the Layer Set

Before layer set rules can be applied to nets, the layer set itself must be created. During this task, you assign an ID to the layer set and can also select its associated signal layers using the dialog boxes shown in the following figure.

Figure 5-2 Define Layer Set Dialog Boxes



Layer Set ID: LS3

LS1
LS2

Routing Layers: SIGNAL_1, SIGNAL_6, SIGNAL_7, SIGNAL_8, SIGNAL_3, SIGNAL_4

Assigned Layers: SIGNAL_2, SIGNAL_5

See the procedures for *Define – Layer Set – Define/Delete Layer Set By List* in the *Allegro PCB Router Command Reference* for further information on creating and deleting layer sets as well as assigning signal layers to a layer set.

Applying Layer Set Rules

You apply layer set rules to nets by accessing the appropriate Wiring Rules dialog box from the Rules menu. Choosing the right Wiring Rules dialog box enables you to apply layer set rules at the correct level within the routing rule hierarchy.

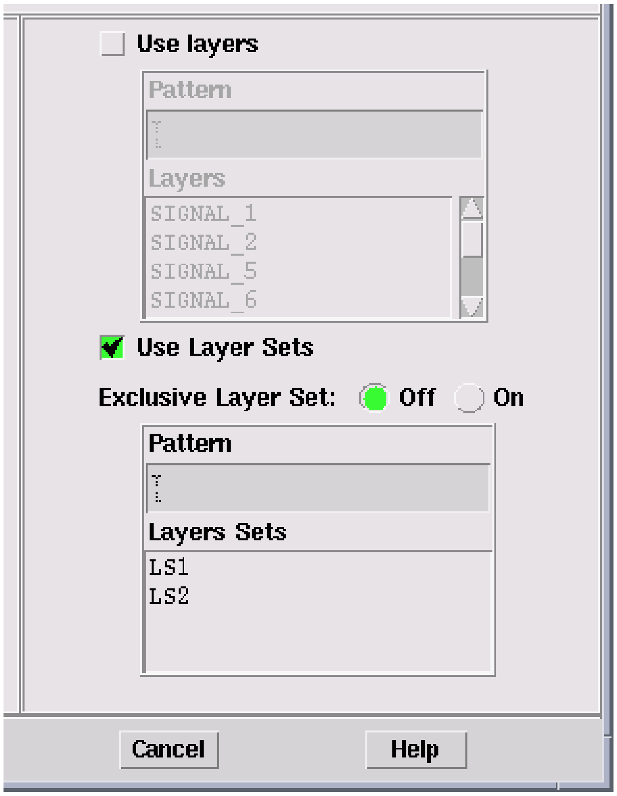
Rule Levels

Layer set rules can be applied at the following routing rule levels.

- Net
- Class
- Group
- Group Set

During this task, you select the rule level object as well as the associated layer set(s) containing the signal layers where its routing is to be restricted. The Layers panel from a Wiring Rules dialog box is shown in the following figure.

Figure 5-3 Group Wiring Rules Dialog Box - Layers Panel



□ Use layers

Pattern

Layers

SIGNAL_1
SIGNAL_2
SIGNAL_5
SIGNAL_6

☑ Use Layer Sets

Exclusive Layer Set: ● Off ○ On

Pattern

Layers Sets

LS1
LS2

To apply layer set rules to nets

1. Choose *Rules – <rule level> – Wiring – General*.

   The Wiring Rules dialog box for the chosen rule level appears.

2. Click the *Pick* button and select the rule level object to apply the rules to.

3. Click to enable *Use Layer Sets*.

   Use Layers (if previously selected) is disabled.

4. Enable or disable *Exclusive Layer Set*.

   Note: When on, all net members route to the *same* layer set as determined by the autorouter.See Exclusive Layer Set on page 169 for further details.

5. Select one or more layer sets to be assigned to the selected rule level object by entering a layer set name or name pattern in the *Layer Sets* Pattern data entry box or click a layer set ID(s) in the *Layer Sets* List box.

6. Click *Apply* and repeat steps 2 through 5 to apply layer set rules to other nets.

   or

   Click *OK* to dismiss the Wiring Rules dialog box.

Exclusive Layer Set

Exclusive Layer Set is a switch that enables you to control whether the autorouter has the freedom to choose an exclusive layer set (from among the assigned layer sets) for routing all *members* in the selected class, group or group set.

Note: The meaning of the previously used term *members* is context-sensitive. It is dependent upon the rule object type being referred to. For example, members of a class are nets and members of a group set are groups.

The exclusive layer set decision is based on the amount of total routed length on each assigned layer set after five passes. The layer set with the *least amount* of total routed length is chosen as the exclusive layer set for all members of the selected rule object. Turning this switch on enables this behavior and allows the autorouter to check and converge on an exclusive layer set.

For example, consider the following scenario.

Class1, consisting of sixteen nets, is set to route on layer sets LS1, LS2 and LS3.

With Exclusive Layer Set *on*, the autorouter checks, converges and ultimately routes *all* the members on either LS1, LS2 or LS3.

With Exclusive Layer Set *off*, all members in Class1 are allowed to route independently on one or more of the assigned layer sets. For example, four nets on LS1, ten nets on LS2 and two nets on LS3.

Preventing Routing on External Layers

You prevent the autorouter from routing on external layers by unselecting those layers. When external layers are unselected, SMD pins can still escape, but all other routing to and from escape vias is performed on internal layers. For example, to prevent routing on external layers TOP and BOT and set the SMD escape distance to 200 mils, add the following commands to your Do file (see Editing the Do File on page 193 for more information).

```
unselect layer TOP
unselect layer BOT
change smd_escape 200
```

Max Restricted Layer Length

Max Restricted Layer Length sets a circuit rule that limits routed length on restricted layers. The rule applies to nets, classes of nets, fromtos, groups, and group sets. This rule is provided to limit routing on external layers. It works in conjunction with the Restricted Layer Length Factor which must be set to mark a layer as restricted.

Note: At the class and group set levels this rule applies to individual nets and groups, respectively.

Max Total restricted Layer Length

Max Total Restricted Layer Length determines the maximum value of the total routed length of all source-to-load paths in the group. The sum of the lengths of the routed source-to-load paths in the group must be less than or equal to the Max Total Restricted Layer Length.

Ignoring Nets During Autorouting

You prevent the autorouter from routing a net or class of nets by fixing the net or class before you begin routing. The unrouted fromtos of fixed nets are not attempted by the autorouter. The routed wires of fixed nets cannot be altered by the autorouter. To ignore a net during autorouting, use the `fix net` command.

For example:

```
fix net NETX
```

Note: The difference between the `protect` command and the `fix` command is that the autorouter will route to protected wires, but will not route to fixed wires.

Preventing Rerouting

You can use the `protect` command to prevent rerouting of existing wires and vias. The `protect` command applies only to wires and does not apply to unrouted fromtos. The autorouter can complete any unrouted portions of nets that have protected wires.

The autorouter treats protected wires as routing barriers. When a net is protected, pins on the net are marked with white circles at the center, and any attached wires are marked with a solid white line through the center. To see the solid white circles in pins that have wires attached, disable the wires display in the Layers panel.

To protect all wires, insert the following command before the initial `route` commands in your Do file (see Editing the Do File on page 193 for more information).

```
protect all wires
```

Note: The difference between the `protect` command and the `fix` command is that the autorouter will route to protected wires, but will not route to fixed wires. The `fix` command is discussed in Ignoring Nets During Autorouting on page 170.

Autorouting Two-layer Designs

The autorouter's built-in strategy is excellent for two-layer designs. Therefore, improvement of autorouting results for two-layer designs depends on good placement techniques, particularly for layouts with a large number of SMD devices.

The `fanout` command should not be used on two-layer designs. You usually have better results when you allow the autorouter to use vias as required. On two-layer designs a large number of routing passes should be used as long as conflicts follow a downward trend. The typical number of passes for a two-layer design is 200 to 300. The autorouter is fast on two-layer designs, so the time to complete each pass is short.

Controlling Routing Topologies

What is a Topology?

A routing topology defines how the pins in a net are routed. The two topologies the autorouter uses are starburst (default) and daisy-chain. The daisy-chain topology is further divided into simple (default for daisy-chain), balanced, and mid-driven.

You can use the `define net` command to:

- Order pin-to-pin connections
- Define a branch topology
- Sequentially order how pins are routed
- Reorder nets for daisy chain routing

Use the `define class` command to control junctions on starburst nets.

Use the `order` command to order nets for daisy-chain routing.

Use the `rule` command to control stub length on daisy-chained nets

Use the `assign_pin` command to assign source, load, and terminator properties

For more information, see Using the Topology Editing Menu on page 239

Topologies are geometrically and electrically developed to:

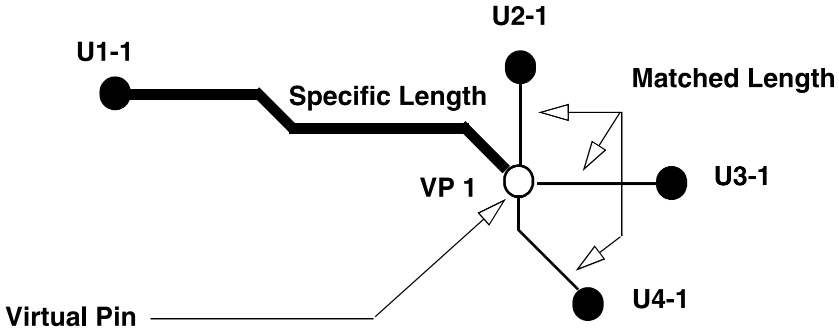
- achieve accurate signal distribution to all nodes in the net per manufacturers spec.
- develop consistent equalized timing to all legs or nodes in the net.

Due to these requirements, there is a need to break up the net into smaller pieces with controlled branches. These branches require the system to create some sort of tee points with all routing between these tee points being equalized. These tee points are known as *virtual pins*.

What is a Virtual Pin?

A virtual pin is simply a balance point between the pins of a net. It is not a real pin, and may not even be a drilled via when the routing task is complete but is merely a location to hold the place of branching interconnect points. Therefore, you use a virtual pin as a terminal to define a fromto tree so that delay, length, width, and clearance rules may be set to control impedance and minimize skew.

Figure 5-4 Virtual Pin Topology Example



Dynamic Virtual Pins

Virtual pins (those without an PCB Editor fixed tee tolerance) are free to move before, during and after automatic routing. This capability enables connection percentage, connection length, via count and adherence to length and delay constraints to be optimized while observing user-specified virtual pin position/radius and etch/via keepouts. This is especially important where you must complete multiple H-Tree solutions, as is generally the case with memory buses.

Additionally, during interactive routing, you can manipulate a routed virtual pin as easily as an unrouted pin using full push and shove capabilities. If length checking is activated, timing is automatically re-evaluated after manual virtual pin movement. However, no automatic adjustment of tuning lengths occurs (as with the autoroute process). For more information on interactive routing, see Using Interactive Routing Tools on page 236.

Automatic Routing and Dynamic Virtual Pins

To achieve dynamic movement of virtual pins during automatic routing, the router observes the following virtual pin guidelines.

- Physical Configuration

- Placement
- 80% Rule
- Shoving
- Tuning Compensation
- Maximum Via Count warning

Physical Configuration

The router chooses the form for a virtual pin that occupies the least real estate and minimum layer span while satisfying electrical constraints.

Therefore, a virtual pin can exist in any of the following forms.

- copper tee
- through via
- fanout via
- blind or buried via
- component pin

Placement

Virtual pins are placed to coincide with pre-existing same-net vias or wire T-junctions that satisfy constraints. Timing is also used to help locate the pin relative to its target. The following placement guidelines are also observed.

- Virtual pins cannot be placed on top of the following:
  - ❑ keepout
  - ❑ hole
  - ❑ cutout
  - ❑ another virtual pin
  - ❑ a pin of a different net

  Note: Two virtual pins can occupy the same location *on different layers* if they are either blind or buried vias or copper tees*.*

- Virtual pins are not allowed to "eat up" routing channels by blocking a trace between it and nearby fixed objects such as pins.

80% Rule

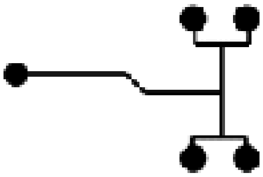
Virtual pin interconnects are biased toward source-virtual pin branches as specified by the *source_seg_ratio* rule (a.k.a. 80% rule). This improves overall results by allocating the greater proportion of a net delay/length to the net portion common to all loads on the net and also conforms to current best-practices in electrical engineering.

To distinguish source from load, the router does the following:

- First treats pins with their PINUSE property set to OUT as sources.

  Failing to find a source, it groups pins that share the same match-delay and virtual pin as loads, and treats pins without match-delay as sources.
- In the absence of both PINUSE and match-delay, pins in close proximity are heuristically clustered as loads to minimize overall etch length.
- For topologies that contain multiple virtual pins (such as H-Trees), the 80% rule is applied recursively. For example, for the H-Tree in the following figure, the ratios used are 80% for the longest wire, 16% for the two shorter wires, and 4% for the remaining wires

Figure 5-5 H-Tree Topology Example



For details on setting and experimenting with the *source_seg_ratio* rule, see Setting the source_seg_ratio Rule on page 165.

Shoving

Virtual pins behave the same as ordinary vias or copper tees in terms of movement allowed. Specifically, they can shove adjacent etch during automatic routing. Note that virtual pins can also be moved manually when routing interactively.

Tuning Compensation

When movement of a virtual pin changes the lengths between the virtual pin and other pins, the router re-adjusts the tuning lengths on each of the legs to compensate for the altered topology.

Note: Tuning also recognizes when a branch cannot be adjusted to meet constraints because of sub-optimal virtual pin placement or unduly restrictive timing requirements. This avoids triggering unnecessary cycles of adjustment to the other branches.

Maximum Via Count Warning

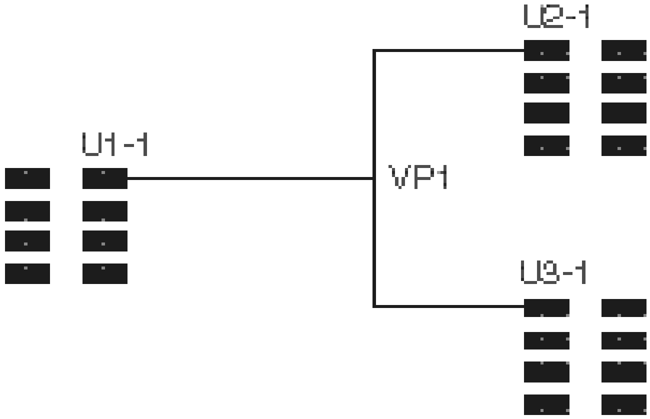
If virtual pins are restricted to single-layer solutions because of a maximum via count rule, a warning message is displayed in the router Output window.

Defining a Branch Topology

You create a branch or tree topology by adding virtual pins to a fromto definition. You can also set delay, length, width, and clearance rules to control impedance and minimize skew. For example, to add a virtual pin to net CLK1 that includes pins U1-1, U2-1, and U3-1, and set a length between 300 and 350 mils for the common path, add the following command to your Do file (see Editing the Do File on page 193 for more information).

```
define (net CLK1 (fromto U1-1 (virtual_pin VP1) (circuit (length 350
300))) (fromto (virtual_pin VP1) U2-1) (fromto (virtual_pin VP1) U3-
1))
```

Figure 5-6 Branch Topology



Note: You can specify the position of the virtual pin. For example, to place VP1 at X,Y coordinates .875 1.05, add the following command to your Do file (see Editing the Do File on page 193 for more information).

```
define (net CLK1 (fromto U1-1 (virtual_pin VP1 (position .875
1.05))) (fromto (virtual_pin VP1) U2-1) (fromto (virtual_pin VP1)
U3-1))
```

You can specify multiple levels of virtual pins to construct a big tree topology. Virtual pins are typically used for clock nets.

You can use the `junction_type term_only` rule to force virtual pins to be vias instead of wire-to-wire tjunctions. If the virtual pin is a via, it will snap to the specified via grid, which might not be the same as the X,Y coordinate position that you specified.

Note: The rules for virtual pins are employed only when the appropriate license is installed (see Understanding Licensing on page 18 for more information).

Assigning Source, Load, and Terminator Properties

The autorouter employs the daisy chain rules that are set in your layout system. If the rules are not set in your layout system and you need to route a net in daisy chain fashion to control the routing order of source, load, and terminator pins, you can assign pin properties and the daisy order rule. Use the `assign_pin` command when you want to assign a large number of source and terminator properties. You can use `assign_pin` to change the status of all pins of a component or only specific pins.

When you assign a pin type, the assignment has no effect until the `order daisy` command is applied. If you assign all pins of a component as source or terminator types and then reorder a single net attached to one pin of that component, immediately reassign the other component pins to load status to avoid possible errors during subsequent order operations. For example, to assign source, load, and terminator pin properties to pins on net SIG5 and order the net for daisy chain routing, add the following commands to your Do file (see Editing the Do File on page 193).

```
assign_pin load U23 (pins 7 8 9 22 24 26)
assign_pin terminator RN75
order daisy net SIG5
```

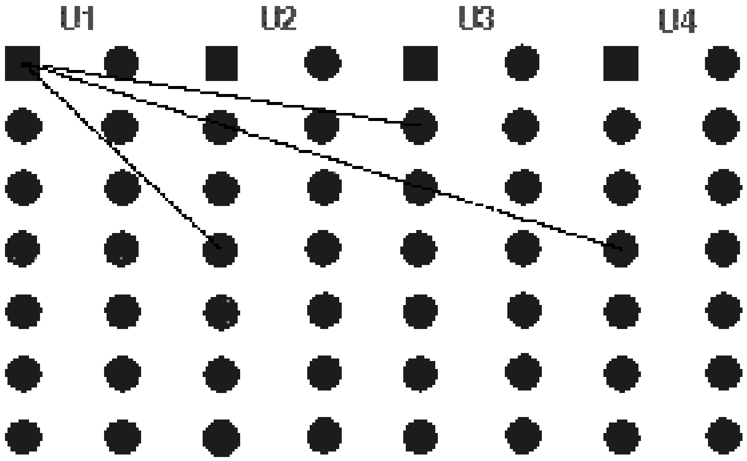
Ordering Pin-to-Pin Connections

The autorouter follows net ordering rules that are explicitly set in your layout system. If net ordering rules are not set in your layout system, and you need to define pin-to-pin routing to satisfy timing requirements, you can define the fromto order in the autorouter. You can explicitly order a starburst pattern or create any routing pattern you want by using the fromto rule with the `define_net` command.

For example, to define how the pins in net NET1 are connected, add the following command to your Do file.

```
define (net NET1 (fromto U1-1 U2-4) (fromto U1-1 U3-2) (fromto U1-1
U4-4))
```

Figure 5-7 Defining Net Pin Connections

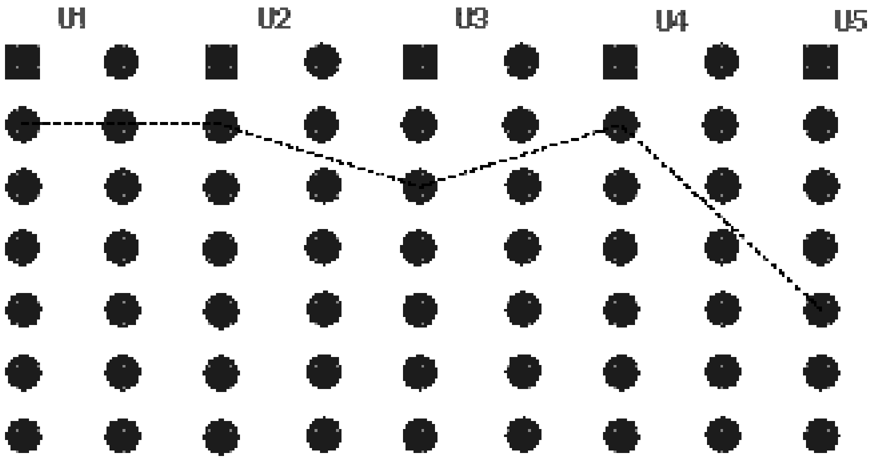


Sequentially Ordering How Pins are Routed

To specify a sequential routing order of pins, use the order rule with the `define net` command. Use the `comp_order` rule to specify the order by component ID. For example, to define that pins U1-2, U2-2, U3-3, U4-2, and U5-5 in net NET1 are routed in order, add one of the following commands to your Do file (see Editing the Do File on page 193).

```
define (net NET1 (order U1-2 U2-2 U3-3 U4-2 U5-5))
define (net NET1 (comp_order U1 U2 U3 U4 U5))
```

Figure 5-8 Specifying the Routing Order of Pins



Note: You can order how nets in a class are routed. For example, to define the routing order of nets in class C1 as connected from U1 to U2 to U3 to U4 add the following command to your Do file.

```
define (class C1 sig12 sig21 sig19 (topology (comp_order U1 U2 U3
U4)))
```

Controlling tjunctions on Starburst Nets

To control tjunctions on starburst nets, use the tjunction and junction_type rules. The tjunction rule applies only to nets ordered for starburst routing. When the tjunction rule is enabled (default), you control where tjunctions occur by using the junction_type rule.
If junction_type is term_only, tjunctions can occur only at pins or vias. If junction_type is supply_only, tjunctions can occur only at pins, vias, and SMD pads connected to a supply net. If junction_type is all (default), tjunctions can occur at pins, vias, and wire segments.

For example, to set tjunctions to occur only at pins or vias for all nets in CLASS1, add the following commands to your Do file (see Editing the Do File on page 193).

```
define (class CLASS1 clka clkb clkc)
rule class CLASS1 (tjunction on) (junction_type term_only)
```

Ordering Nets for Daisy-Chain Routing

You can order nets for simple daisy-chain routing by using the `order daisy` command. You can only apply the `order daisy` command to unrouted nets. You can order daisy-chain routing for all nets, by class, or by individual net.

For example, to order daisy-chain routing for all nets, add the following command to your Do file.

```
order daisy all_net
```

Note: You can order an individual net by using the reorder daisy rule. For example, to order daisy-chain routing for NETA, add the following command to your Do file.

```
rule net NETA (reorder daisy)
```

When you order nets for daisy-chain routing, tjunctions cannot occur except within the stub distance specified by the max_stub rule. If you do not specify the max_stub rule, no stub is allowed.

Controlling Stub Length on Daisy-Chained Nets

Daisy-chain routing does not allow tjunctions unless you specify a max_stub rule. The autorouter follows the daisy chain and maximum stub length rules that are set in your layout system. If you do not set the rules in your layout system, and you need to route a net in daisy chain fashion and control tjunctions at pins and vias, you can order the net for daisy chain routing and set a max_stub rule.

If you do not specify a `max_stub` rule, the default is zero or no stub. If the `max_stub` rule is greater than zero, the junction_type rule determines where stubbing can occur. When junction type is term_only, tjunctions can only occur at pins or vias. When junction type is all (default), tjunctions can occur at pins, vias, and wire segments.

For example, to specify a maximum stub length of 250 mils for all daisy-chained nets in CLASS1 and limit tjunctions at pins and vias, add the following commands to your Do file (see Editing the Do File on page 193).

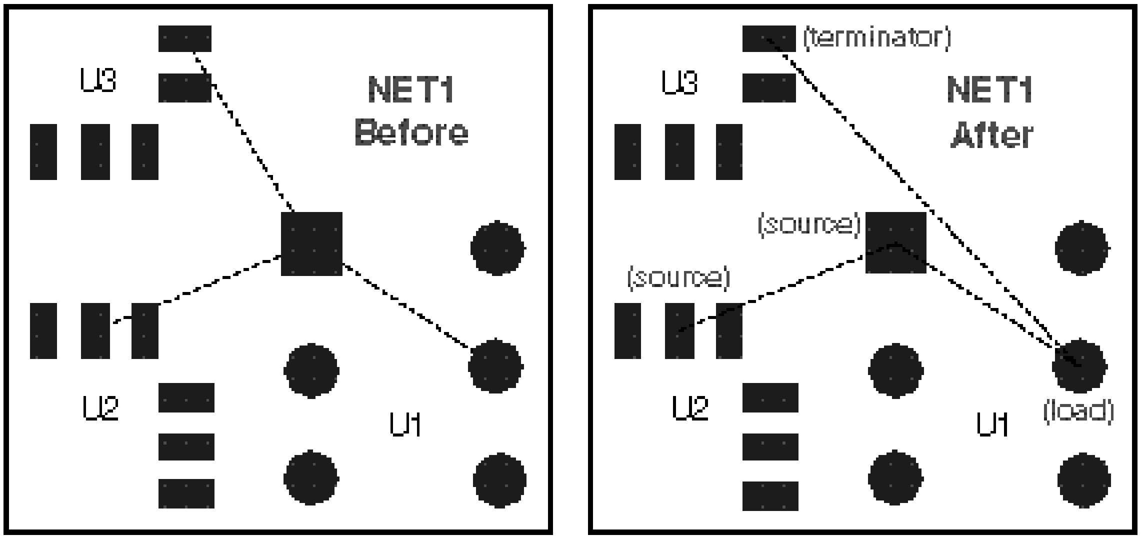
```
define class CLASS1 (clka clkb clkc)
order daisy class CLASS1
rule class CLASS1 (max_stub 250) (junction_type term_only)
```

Reordering Nets for Daisy-Chain Routing

You can set the pin type to source, terminator, or load (default) status before you use the reorder daisy rule. The pin type is not evaluated by the autorouter until the reorder daisy rule is applied. If multiple source pins are defined, they are chained first, followed by load pins, and then all terminator pins are chained.

The following figure shows a net before and after daisy-chain reordering.

Figure 5-9 Before and After Daisy-Chain Reordering



To control source and terminator pin types, use the source and terminator attributes with the `define net` command.

For example, to specify U1-1 and U2-2 as the source pins and U3-3 as a terminator pin, add the following command to your Do file (see Editing the Do File on page 193).

```
define (net NET1 (source U1-1 U2-2) (terminator U3-3) (rule (reorder
daisy)))
```

Note: You can use the `?` and `*` wildcards to assign pin types to multiple nets. For example, to specify that all nets beginning with NET that have pins on U1 and U2 be treated as source and pins on U3 be treated as terminator, add the following command to your Do file.

```
define (net NET* (source U1 U2) (terminator U3) (rule (reorder
daisy)))
```

Controlling Via Use

Using Specific Vias

Specifying a Via to Use

Controlling which Via is Used

Using Larger Diameter Vias

Controlling the Maximum Number of Vias on a Net

Limiting the Number of Vias per Fromto

Preventing Via Use for a Net or Class

Controlling the Maximum Number of Vias on a Net

Disallowing Vias for the Entire Design

Other Operations Involving Vias

Controlling Via-to-Via Clearances

Setting Via-to-Via Clearance

Fanning Out Power Pins

Setting the Maximum Escape Distance for Fanout

Sharing Power and Ground Vias

Controlling Via-to-Via Clearances

You can control the clearance between vias that are on different layers by using the `buried_via_gap` rule. You control the clearance between vias on the same layer by using the `via_via` rule.

For example, to set a minimum clearance of .015 mils between buried vias on different layers and a minimum clearance of .025 mils between vias on the same layer, add the following commands to your Do file (see Editing the Do File on page 193).

```
rule NETB (clearance .015 (type buried_via_gap))
rule NETB (clearance .025 (type via_via))
```

Note: All via-to-via clearances are measured from the edge of one via to the edge of the other.

When manufacturing rules permit blind and buried vias to stack up on different layer pairs, use the `layer_depth` option to the `buried_via_gap` rule to limit the layers checked for buried via clearances.

Setting Via-to-Via Clearance

When the clearance between vias is too small, islands of copper can be created on power and ground planes due to antipad overlaps, and access to SMD pads can be blocked by a wall of vias. You can set a pcb level (global) via-to-via clearance rule to eliminate or minimize these problems. For example, add the following rule to your Do file (see Editing the Do File on page 193).

```
rule pcb (clearance 50 (type via_via))
```

Using Specific Types of Vias

Specifying a Via to Use

You define the vias that are available for routing in the via descriptor of the Design File. Vias that you define as spares are not available for routing, except if the spare via is associated with a net by a `use_via` rule or specified in the `testpoint` or `testpoint rule` commands.

The following shows the via descriptor in the Design File. In this example, the vias v25, bbv1_2, and bbv2_3 are available for routing. The spare vias are `testpt1` and `testpt2`.

```
(Via v25 bbv1_2 bbv2_3 (spare testpt1 testpt2))
```

The autorouter chooses the available via with the smallest diameter to make a wiring transition between layers. In the example above, `bbv1_2` has the smallest diameter and is, therefore, used by the autorouter. If you want the autorouter to use via `v25`, you must unselect all available vias, then select via `v25` by entering the following commands.

```
unselect all vias

select via v25
```

Note: If you select a via without unselecting all other available vias, the via that you selected becomes one of the available vias. Other vias might also be available for routing. If the via that you selected does not have the smallest diameter, it will not be used by the autorouter.

To make sure that a specific via is the one that is used for routing, you should define only that via in your Design File. For example, if you want to make sure that v25 is the only available via, you can edit the via descriptor in the Design File to the following:

```
(Via v25 (spare bbv1_2 bbv2_3 testpt1 testpt2))
```

In general, you use spare vias for test point and fanout operations.

Controlling which Via is Used

You can control which via is used to route a group, net, class of nets or region by setting a `use_via` rule. The via you choose must have a padstack shape on the layers used for routing.

For example, to route net CLK4 with via Via25, group G1 with via V35, class C1 with via V30, and region R1 with via V40 add the following commands to your Do file (see Editing the Do File on page 193).

```
circuit net CLK4 (use_via Via25)
define (group G1 (fromto U4- U4-8) (fromto U4-20 U2-17))
circuit group G1 (use_via V35)
define (class C1 DATA1 DATA2 DATA3 DATA4 DATA5)
circuit class C1 (use_via V30)
circuit region R1 (use_via V40)
```

Note: Vias specified by the use_via rule are used even if you unselect all vias.

In the following example, a special via (via_power) is used for the power net GND.

```
circuit net GND (use_via via_power)
```

The autorouter does not require special vias to connect power nets on power planes. When a power net is routed with a via padstack that passes through the power plane, the autorouter assumes that the via connects to the corresponding power plane with an isothermal shape.

Using Larger Diameter Vias

You control which vias are used to connect to power and ground planes by using a `use_via` rule. For example, to route VCC and GND nets with via V40, add the following commands to your Do file (see Editing the Do File on page 193).

```
circuit net VCC (use_via V40)
circuit net GND (use_via V40)
```

For more information, see Controlling which Via is Used on page 185.

The use_via rule can be defined at the following levels of the rules hierarchy:

- class
- group_set
- net
- group
- fromto
- region
- region_class
- region_net
- The use_via rule is inherited down this hierarchy (*from class to region_net*) and is overridden up the hierarchy (*from region_net to class*).

Controlling the Number of Vias

Limiting the Number of Vias per Fromto

You can limit the number of vias used for each connection by using the `limit_via` command or the `limit_vias` rule. For example, to restrict the autorouter to a maximum of two vias per each connection, add the following command to your Do file.

```
limit via 2
```

Note: The `limit_via` command operates with the same scope as a rule.

Another way to restrict the autorouter to a maximum of two vias per each connection is by using the following command:

```
rule pcb (limit_vias 2)
```

Preventing Via Use for a Net or Class

To prevent via use for a net or class of nets use a net or class `limit_vias` rule with a limit of zero. For example, to prevent any via use for CLASSX, add the following command to your Do file (see Editing the Do File on page 193).

```
rule class CLASSX (limit_vias 0)
```

Controlling the Maximum Number of Vias on a Net

You control the maximum number of vias used to route a net by assigning a `max_total_vias` rule. The autorouter cannot exceed the maximum number of vias assigned when routing the net. For example, to limit the maximum number of vias used to route net SIG1, add the following command to your Do file (see Editing the Do File on page 193).

```
rule net SIG1 (max_total_vias 3)
```

Disallowing Vias for the Entire Design

To disallow vias for the entire design, use the `unselect_all_vias` command. This command leaves the autorouter with no vias to use and is an efficient way to disallow vias.

Note: The use_via rule overrides the `unselect_via` command.

Satisfying Buried Via Requirements

Using Buried Vias

When your design includes buried vias, the autorouter chooses the correct via to transition between layers.

To use buried vias

1. Make sure you define blind and buried via padstacks in the Design File.
2. Make the blind and buried vias available for routing by using the following command:

   ```
   select via <id>
   ```

Note: For additional information on via definitions in the Design file, see Specifying a Via to Use on page 184

Controlling Via-to-Via Clearances

You can control the clearance between vias that are on different layers by using the `buried_via_gap` rule. You control the clearance between vias on the same layer by using the `via_via` rule.

For example, to set a minimum clearance of .015 mils between buried vias on different layers and a minimum clearance of .025 mils between vias on the same layer, add the following commands to your Do file (see Editing the Do File on page 193).

```
rule NETB (clearance .015 (type buried_via_gap))
rule NETB (clearance .025 (type via_via))
```

Note: All via-to-via clearances are measured from the edge of one via to the edge of the other.

When manufacturing rules permit blind and buried vias to stack up on different layer pairs, use the `layer_depth` option to the buried_via_gap rule to limit the layers checked for buried via clearances.

Placing Vias Under SMD Pads

You can control whether vias are allowed under SMD pads by using the via_at_smd rule.

```
rule pcb (via_at_smd [off | on [(grid [off | on])] | [(fit [off | on])] | [(thru
[off | on])]])
```

When the via_at_smd rule is `on`, the autorouter can place vias under SMD pads. The default setting is `off`.

Example:

```
rule pcb (via_at_smd on)
```

Options

The `grid on` option allows vias under SMD pads only if a via grid point lies within the SMD pad. The default setting is `off`. If the grid option is not specified or is disabled, the via is placed at the pad origin, even if the origin is off-grid.

Example:

```
rule pcb (via_at_smd on (grid on))
```

The `fit on` option allows vias under SMD pads only where the via fits entirely within the SMD pad perimeter. The default setting is `off`.

Example:

```
rule pcb (via_at_smd on (fit on))
```

The `thru on` option allows thru vias to be placed under SMD pads. The default setting is off. If the thru option is not specified or is disabled, only blind vias are allowed under SMD pads.

Example:

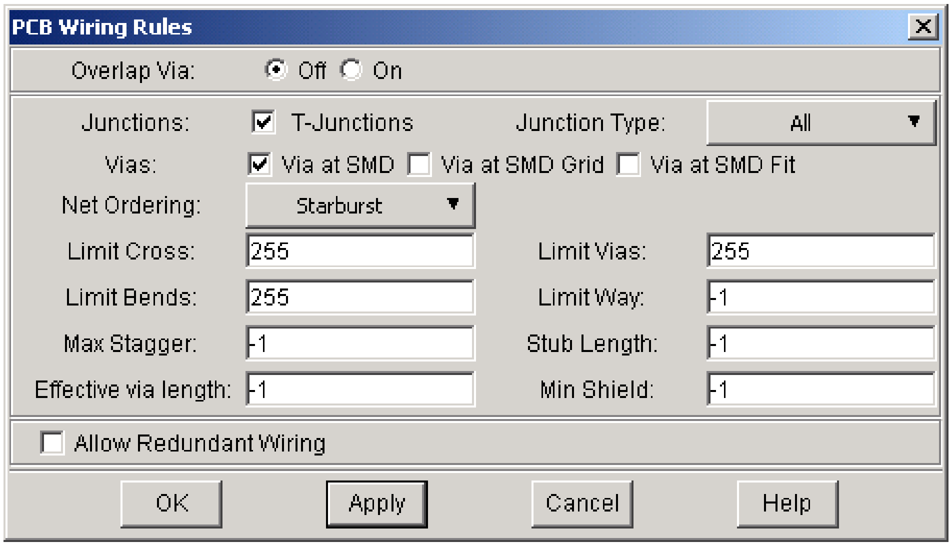
```
rule pcb (via_at_smd on (thru on))
```

The following complete example allows thru vias under SMD pads only if a via grid point lies within the SMD pad, and only if the via fits within the SMD pad.

```
rule pcb (via_at_smd on (grid on) (fit on) (thru on))
```

You can also set these options in the PCB Wiring Rules dialog box. To access it, choose *Rules – PCB – Wiring – General*.

Figure 5-10 PCB Wiring Rules Dialog Box



PCB Wiring Rules

Overlap Via: ● Off ○ On

Junctions: ☑ T-Junctions   Junction Type: All

Vias: ☑ Via at SMD □ Via at SMD Grid □ Via at SMD Fit

Net Ordering: Starburst

Limit Cross: 255   Limit Vias: 255

Limit Bends: 255   Limit Way: -1

Max Stagger: -1   Stub Length: -1

Effective via length: -1   Min Shield: -1

□ Allow Redundant Wiring

- If there is an attach rule in the padstack definition section of the Design File, it must be set to on to permit vias under SMD pads. If unspecified, the attach rule defaults to `enable`.
- To avoid routing conflicts, when using thru on, be sure there are no SMD pads superimposed on the opposite side of the design.

Autorouting with Do Files

Do files are an invaluable tool in the autorouting process. Using Do files you can:

- recreate a previous routing session (by editing a session Did file and running as a Do file)
- automate redundant activities (such as putting nets in classes).
- simplify routing work and increase your productivity.

A Do file is a text file that contains a sequence of autorouter commands.The selection and ordering of commands in this file is very important as they are executed sequentially when you run the file. Following are some commands that are commonly used in an autorouting Do files.

```
bestsave on $\best.wir
status_file $\route.sts
.
.
.
route 25
clean 2
route 50 16
clean 3
write routes $\final.rte
report status $\final.sts
```

Special Purpose Do Files

It is recommended that you create and run several special purpose Do files to accomplish an autorouting task as opposed to running one large Do file. For example, most designers use separate Do files to:

- set up the routing environment.
- set up the design rules for a specific design.
- set up fences and keepout areas.
- etc.

You can then use and reuse these Do files for similar designs that must meet the same type of constraints. You can also call one or more of these files from a single *master* Do file to accomplish a specific autorouting task.

Nesting

The number of nested Do file levels is unlimited. For example, your master Do file may call a rules Do file named `rules.do`, that in turn may call other Do files such as `classes.do`, `topology.do`, and `timing.do` and so on.

Do file Creation

Rules Did File Editor

After you are familiar with router commands and command syntax, you can often write a Do file from memory. However, while you are learning, you can use the Rules Did File editor to record the commands that are generated as you use the menus and dialog boxes. Do files created by this method are syntactically correct.

The editor can record rule setting and other commands. You can control which types of commands are recorded and turn recording on and off at any time during the session. You can also use the editor to add and modify commands.

Note: The Rules Did File editor records only rule setting commands if *Rules Only* is checked. Remove the check from *Rules Only* to record other commands such as autorouting commands. When you close the editor, the settings are retained for the session. However, Rules Did File editor settings are not retained after you exit the router.

Session Did File

You can edit your session Did file (created by default) for use as a Do file. Retain only the commands that set rules or grids, define groupings such as classes or clusters, or perform other tasks you want to repeat in another session.

Existing Do File Customization

There are many sample Do files that are already written that you can customize to suit your specific design needs. See Appendix C in the *Allegro PCB Router Command Reference* for listings of sample Do files.

Editing the Do File

To edit a Do file

1. Launch a standard ASCII text editor, such as Microsoft Windows Notepad (Windows) or vi (Unix).
2. Open an existing Do file to customize or create a new one by typing a list of autorouting commands in the order that you expect them to be executed.
3. Insert additional commands, as needed, to meet your specific design requirements.
4. Save the file with the extension .do.

For information on running Do files, see Using Do Files on page 63.

Programmable Do Files

Your Do files can include conditional expressions, iterative macros and system variables to allow for decision branches and loops.

Expressions

An `if` expression can be used to execute one of two groups of autorouter commands based on the value of an expression. The `while` expression executes a group of commands repeatedly until an expression is no longer true (zero). Expressions can include the use of system variables. See Table 5-2 on page 194 for details.

Operators

When more than one operator is used in an expression, rules of precedence determine the order of evaluation. Evaluation of operators at the same precedence level in a single expression is from left to right.

The following table lists the numeric binary operators in descending order of precedence, with the highest precedence operator at the top. Operators that have the same precedence level are grouped together. For example, multiply, divide, and modulo have the same precedence level.

Table 5-1 Expression Operators

Operator	Function
( )	grouping
-	negation
!	logical NOT
*	multiply
/	divide
%	modulo
+	add or concatenate
-	subtract
<	less than
>	greater than
<=	less than or equal to
>=	greater than or equal to
==	equal to
!=	not equal to
&&	logical AND
||	logical OR

Table 5-2 System Variables

Variable Name	Data Type	Description
bottom_layer_sel	Integer	Value is 1 if bottom layer is selected, 0 if unselected.
complete_wire	Real	Completion ratio expressed as a percentage.
conflict_clearance	Integer	Number of clearance rule violations.
conflict_crossing	Integer	Number of crossing conflicts.
conflict_wire	Integer	Number of crossing and clearance conflicts.
conflict_xtalk	Integer	Number of crosstalk rule violations.
connections	Integer	Total number of connections to be routed.
current_wire	Integer	Current wire being routed or unrouted.

Variable Name	Data Type	Description
locked_comp	Integer	Number of locked components.
partial_selection	Integer	Value is 0 if no nets or all nets are selected and 1 if one or more nets (but fewer that all nets) are selected.
placedcomp	Integer	Number of placed components.
power_layers	Integer	Number of power layers.
reduction_ratio	Integer	Conflicts reduction ratio from the last completed routing pass.
reroute_wire	Integer	Number of wires and wire segments to be rerouted in the current pass.
route_pass	Integer	Current routing pass or last pass.
sel_comps_list	String	Names of all selected components
sel_nets_list	String	Names of all selected nets.
sel_signal_layers	Integer	Number of selected signal layers.
selectedcomp	Integer	Number of selected components.
selectednet	Integer	Number of selected nets.
signal_layers	Integer	Number of signal layers.
smd_pins	Integer	Number of SMD pads.
thru_pins	Integer	Number of through-hole pins.
top_layer_sel	Integer	Value is 1 if top layer is selected, 0 if unselected.
total_pass	String	Total passes for the current command.
total_pins	Integer	Total number of pins.
total_vias	Integer	Total number of vias.
totalcomp	Integer	Total number of components on the PCB.
unconnect_wire	String	Unconnected wires (unconnects)
units	String	Unit of measure set by user.
unplaced_comp	Integer	Number of unplaced components outside the placement boundary.

Variable Name	Data Type	Description
unplaced_large	Integer	Number of large components outside the placement boundary.
unplaced_small	Integer	Number of small components outside the placement boundary.

Using Fanout

Fanning Out Signal Pins

You can route short escape wires from pins to vias by using the `fanout` command. With the `fanout` command, you can control pin and via sharing, specify the layer depth, control the escape direction, and set a temporary via grid for this command to use. For example, to route escape wires from pins to vias outside the component footprint, add the following command to your Do file (see Editing the Do File on page 193).

```
fanout (direction out)
```

Fanning Out Power Pins

You can fanout power pins by using the `fanout` command. With the `fanout` command, you can control pin and via sharing, specify the depth of blind and buried vias, and control the via grid. For example, to force the autorouter to fanout power pins only inside component bodies, add the following command to your Do file.

```
fanout (pin_type power) (direction in)
```

Note: You should place the `fanout` command near the beginning of your Do file after you have defined all rules and before any autorouting.

You can specify a via grid when you use the fanout command by using the `via_grid` option. For example, to force the autoroute to fanout power pins on a 250 mil via grid, add the following command to your Do file.

```
fanout (pin_type power) (via grid 250)
```

If you use the `smart_via_grid` option with the `fanout` command, the autorouter automatically calculates a via grid based on whether you specify that one wire or two wires can be placed between a via. This via grid is used only during the `fanout` command.

Note: You can use `power fanout` rules to control the order in which fanout connects power pins of large components to decoupling capacitors and vias at the PCB, NET, and CLASS levels. For details see the <power_fanout_descriptor> in the `rule` command.

Setting the Maximum Escape Distance for Fanout

You set maximum SMD escape distance by using the `max_len` control with the `fanout` command. For example, to set a maximum escape distance of 200 mils for power and ground nets, add the following commands to your Do file (see Editing the Do File on page 193).

```
unit mil

fanout 5 (max_len 200) (pin_type power) (direction in_out)
```

Sharing Power and Ground Vias

You can set the autorouter to share escape vias attached to SMD power or ground pins with decoupling capacitor pins when the distance between vias and pins is within a certain range. You can also allow SMD pins to share power through-pins and control the maximum power and ground escape distance. For example, to fanout SMD power pins using five passes, with a maximum escape length of 100, with pin and via sharing enabled for shared pins or vias within 75 mils of a decoupling capacitor pin, add the following command to your Do file.

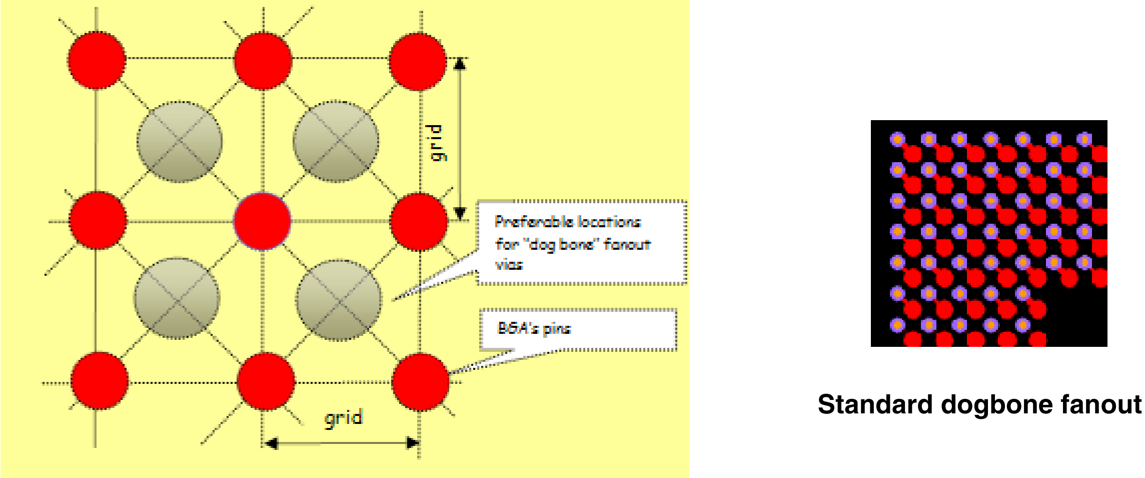
```
fanout 5 (pin_type power) (max_len 100) (via_share on) (pin_share on)
(share_len 75)
```

Fanout on BGA Components

Symmetric Pin Arrays

Generally, BGA pin patterns are based on symmetric arrays. When a differential pair is inside a symmetric pin array, the autorouter places the fanout vias as close as possible using a standard dogbone fanout pattern. In these cases, the location the fanout vias are at the geometric centers of the virtual cells using a fixed 45 degree fanout angle as seen in Figure 5-11.

Figure 5-11 Fixed (45 degree) fanout angle used for symmetric pin arrays



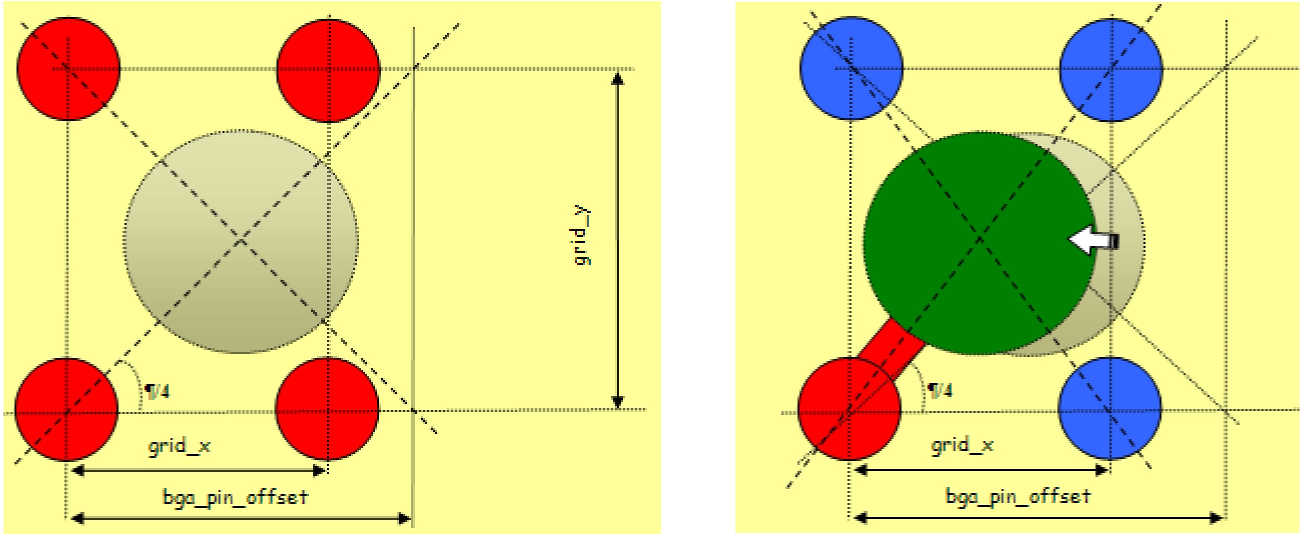
Standard dogbone fanout

This approach works well until non-symmetric pin arrays are encountered. For these cases a variable fanout angle is required as shown in Figure 5-12.

Non-symmetric Pin Arrays

Prior to running fanout, you can set the fanout angle to be variable using the command `set bga_fanout_any_angle on`. For further details, see the `set` command in the *Allegro PCB Router Command Reference*.

Figure 5-12 Variable fanout angle required for non-symmetric pin arrays



Enabling Dogbone BGA fanout

Some BGAs may have very specific pattern where some rows, columns, or area inside the component can be removed. In this case, internal router BGA detection algorithm may fail to classify the component as BGA and doesn't apply the dogbone pattern to it. For such instances, a component property `dogbone_fanout` has been added to force dogbone fanout pattern on a component.

```
component_property U1 (dogbone_fanout on)
```

You can specify this property in Allegro X Editor for a symbol or symbol instance.

BGA fanout with HDI via patterns

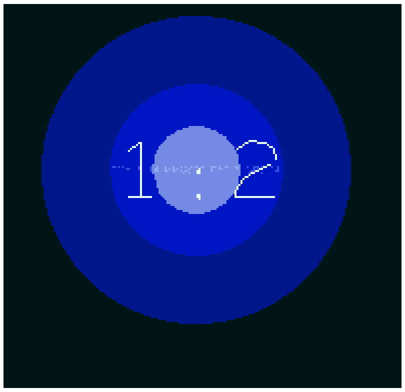
There is a constant increase in the BGA density and pin count, consequently, functionality to handle new requirements is getting added. Recent PCB fabrication technologies have enabled further miniaturization in the manufacturing process. These improvements along with advanced software solutions specifically for BGA allow successful design for these devices.

Introducing blind, buried, and micro via technologies, and HDI layers stack-ups for router has extended this technology and enables creating compact and optimized BGA fanout. Designers can combine the following fanout options to create optimal fanout.

Via at SMD

The functionality is enabled by `via_at_smd` rule defined at PCB level and can be specified in Allegro Constraint Manager. The rule allows exiting to inner layer using a blind or micro via.

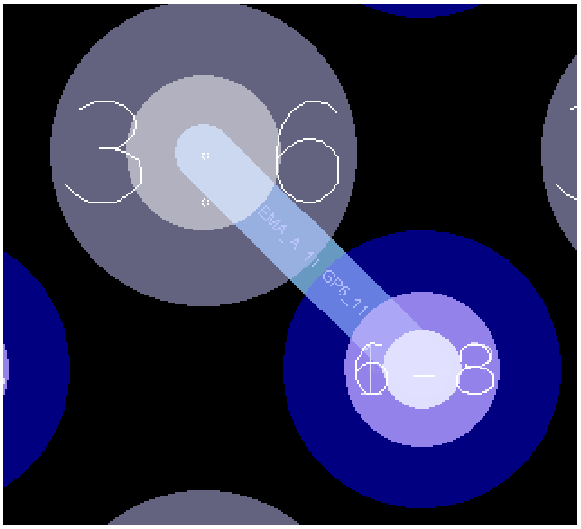
Figure 5-13 Via in pad fanout for BGA



Via stack

Via stacking is controlled by the `stack_via` rule and can be specified through the entire rule hierarchy including a particular layer. Pad-pad Connect constraint from Allegro allows defining stack for micro via objects and in coincident configurations. Router supports coincident and offset stack for blind, buried, and micro via objects. The following figure shows fanout patterns with micro via stacked for both sides and core via staggered.

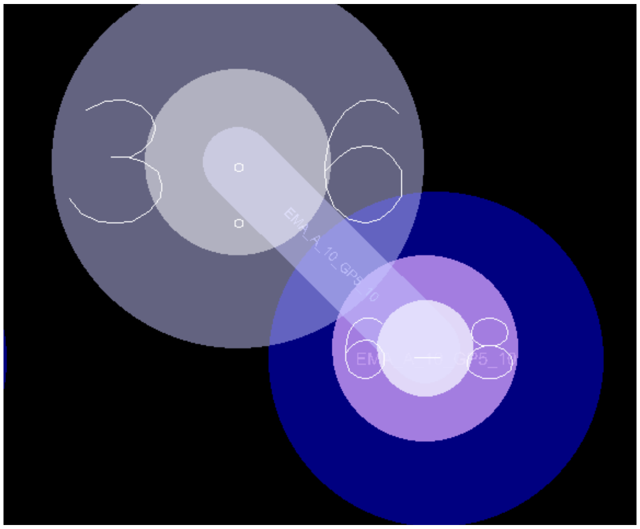
Figure 5-14 Stacked via pattern for BGA fanout



Tangent via

Tangent via configuration allows via in stack pattern to touch but not overlap each other. The case is initiated by special tangent keyword in same net via-via spacing constraint. In Allegro CM the rule is identified by 0 value for same net spacing value. Router creates wire segment that connects tangent via to ensure proper electrical connection. Same net spacing applies for the whole rules stack and can be specified by the layer to define a connection on it. The tangent case can be specified between particular via types, for example, between micro via and for specified layer that creates a compact fanout configurations. The following figure shows tangent micro and core via on sixth signal layer.

Figure 5-15 Tangent via pattern for BGA fanout

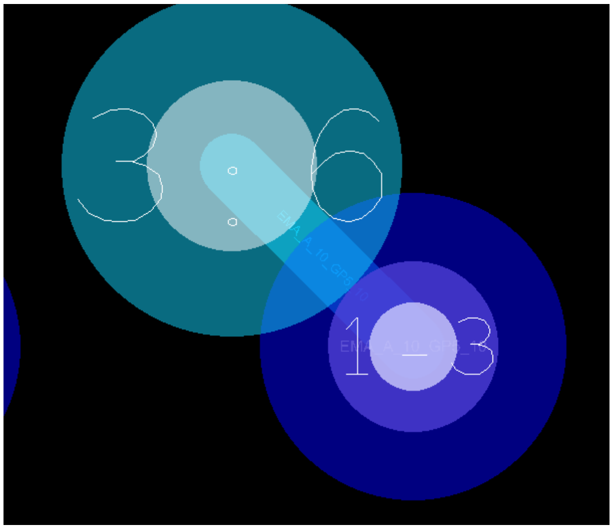


Inset via

Inset via configurations allow creating via patterns where via pads can overlap and actual pads locations are controlled by same net clearance rule for net-based hole to via pad. For PCRB Router this case is defined by a special inset keyword in the same net via-via spacing constraint.

Router creates a wire segment that connects tangent via to ensure proper electrical connection. Same net spacing applies for the whole rules stack and can be specified by layer to define a connection on it. The tangent case can be specified between particular via types, for example, between micro via and for specified layer that allows creating compact fanout configurations. The following figure shows inset micro and core via on second internal layer.

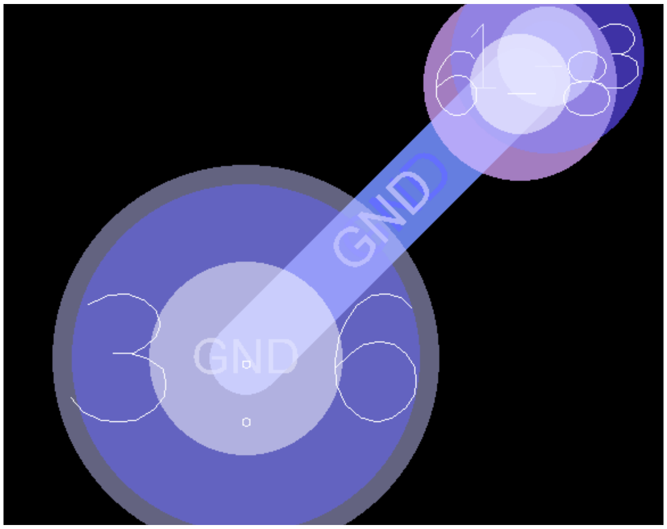
Figure 5-16 Inset via pattern for BGA fanout



Staggered via

Staggered via configuration applies when none of the previously listed configurations are used. Via in the pattern are staggered one under another into the appropriate diagonal direction considering the via same net clearance rule and via staggering minimal gap. If same net rule is disabled, via net to net clearance is considered. Same net spacing applies for the whole rules stack and can be specified by layer to define a connection on it. The tangent case can be specified between particular via types, for example, between micro via and for specified layer that allows creating compact fanout configurations. The following figure shows micro via and core via staggered to form fanout pattern.

Figure 5-17 Staggered via pattern for BGA fanout



Using Smart Route

The `smart_route` command is a simple way of routing most designs. It automatically executes autorouter commands, based its evaluation of your design. The `smart_route` command does the following:

- Sets the wire and via grids to the grid spacing used in your layout system or to the grid spacing that you specify.
- Performs fanout if there are several signal layers or if the top or bottom signal layer is unselected.
- Calculates a via grid to achieve optimum fanout, and adjusts the via grid accordingly.
- Applies bus routing.
- Runs standard routing passes.
- Provides warning and error messages to help analyze difficult designs.
- Saves routing information to a Wires File if routing improved as compared to the previous save.
- Runs clean passes after the routing completes.
- Stops autorouting if the routing will not converge.

You can set the minimum wire and via grids and control whether `smart_route` performs the following operations.

- Routes short escape wires from SMD pads (fanout). During fanout, you control whether the autorouter does the following.
- Shares vias between SMD pads on the same net
- Can escape to through-pins
- Generates test points.
- Miters corners.

Place the `smart_route` command in the Routing Commands section of your Do file (see Editing the Do File on page 193).

Running the Autorouter

You can use any combination of the following methods to apply commands and control the autorouter.

- Use GUI menus and dialog boxes
- Enter commands from the keyboard
- Use a Do file (see Using Do Files on page 63).

The preferred method for controlling the autorouter is with a Do file. A Do file is a text file that contains a list of one or more commands. Each command occupies a separate line in the Do file. Commands are organized in the order you want them to run from the beginning of the file to the end.

The Do file also serves another purpose. It is a record of the rules and other commands you use during an autorouting session. If you need to revise a finished design, you can edit the original Do file and reuse it.

You can run the autorouter using either of the following methods:

- Use the `smart_route` command, which automatically evaluates your design and runs the appropriate autorouting commands.
- Use standard autorouting commands to preroute, route, and postroute the design in separate stages.

The `smart_route` command evaluates your design, performs calculations, starts the autorouter, and continues to apply routing passes until either all connections are routed or further improvement is unlikely. See about smart routing for more details.

The standard autorouting commands (`bus`, `fanout`, `route`, and `clean`) let you route your design in stages while monitoring autorouter progress.

You can route the entire design at once, or you can

- Select one or more connections and route only those connections.
- Define a routing keepin area (fence) and route only connections that are completely within the area.

You can also select which layers are available for routing and select the vias to use when changing layers.

Note: Generally, you must set up the routing environment and assign all rules before running the autorouter.

Evaluating the Progress of Autorouter

You can evaluate the progress of the autorouter by inspecting the design to identify areas where unexpectedly large numbers of conflicts or unroutes may exist. If you find areas like this, you may want to rearrange the component placement or reorder the netlist to improve the autorouter's ability to converge.

Another, more effective, way to evaluate the autorouter's progress is to study the Status File to determine whether the autorouter will likely reach 100% completion. The Status File contains valuable information that you can use to monitor the routing and predict whether problems will occur.

For more information, see the following sections:

Applying Convergence Techniques

Layer Set Rule Checking

Differential Pair Checking

Controlling Routing Topologies

Analyzing the Status File

The Status File provides valuable information about how well the autorouter is progressing. You can identify potential problems and make appropriate changes to the design constraints or parameters, to assure 100% completion.

The following table offers guidelines for analyzing the Status File information. These guidelines indicate where you can look for potential problems.

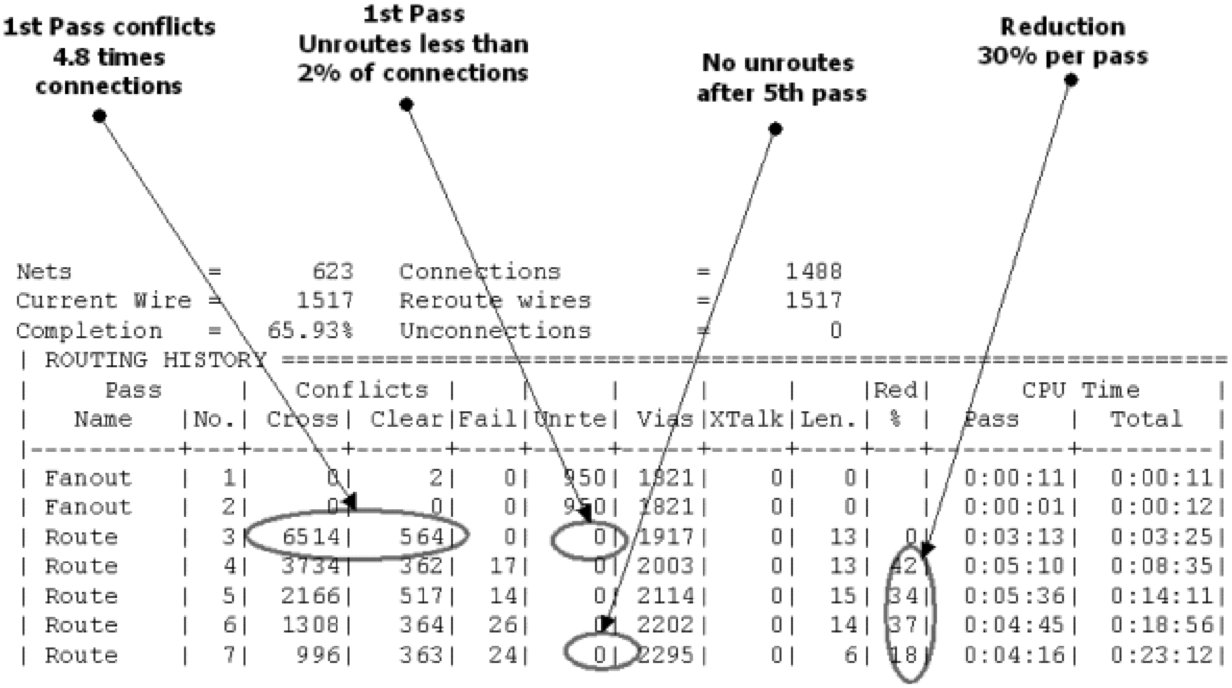
Table 5-3 Status File Guidelines

Routing phase	Routing pass	Problem indicator
Initial phase	1	Unroutes exceed 2% of connections
		Conflicts exceed 5 per connection (7 max)
	2	Conflict reduction is less than 30% per pass (Reduction rate should be at least in the double digits.)
	3	

Routing phase	Routing pass	Problem indicator
	4	
	5	Unroutes exist (Some difficult designs may still have a few unroutes.)
Converge phase	6	Conflicts not reducing over 10 or more passes
	.	
	.	
	74	
	75	Conflicts remain

The Status File shown below reflects normal routing progress with a good likelihood of converging to 100% completion.

Figure 5-18 Normal Routing Progress



1st Pass conflicts 4.8 times connections

1st Pass Unroutes less than 2% of connections

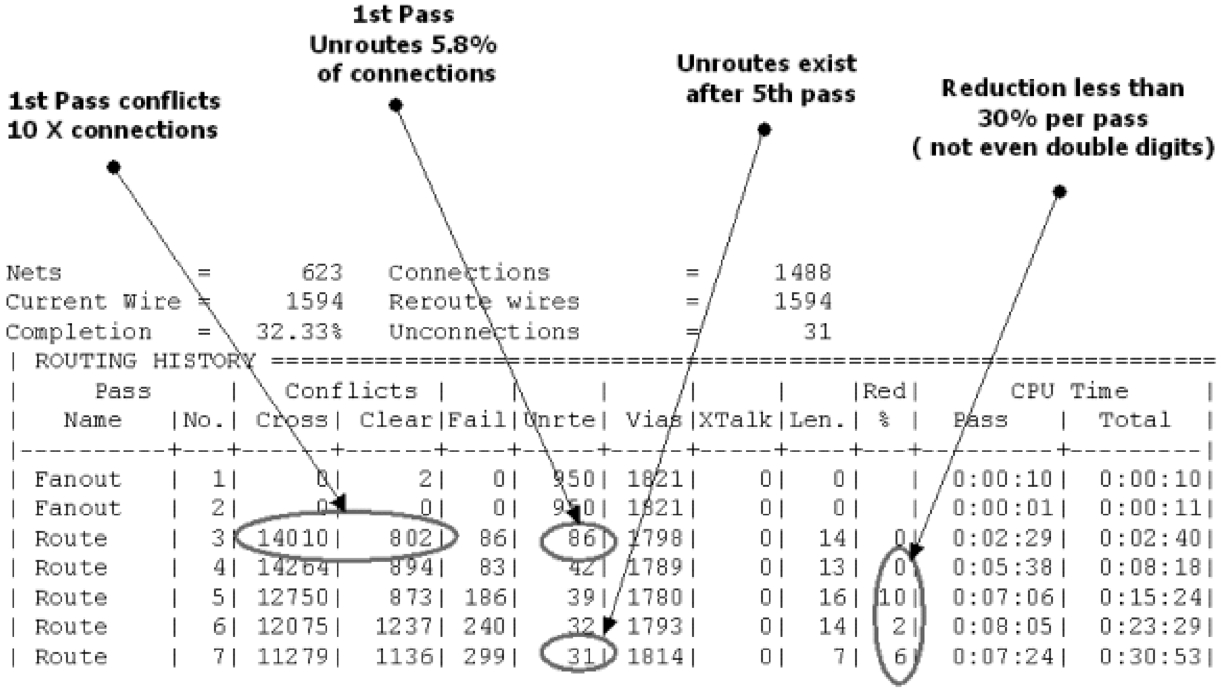
No unroutes after 5th pass

Reduction 30% per pass

```
Nets          =      623   Connections          =      1488
Current Wire =      1517   Reroute wires        =      1517
Completion    =   65.93%   Unconnections        =         0
| ROUTING HISTORY ==========================================================
|      Pass      |  Conflicts |    |     |     |     |    |Red|     CPU Time      |
|   Name   |No.| Cross| Clear|Fail|Unrte| Vias|XTalk|Len.| % |  Pass    |  Total  |
|----------+---+------+------+----+-----+-----+-----+----+---+----------+---------|
| Fanout   |  1|     0|     2|   0|  950| 1821|    0|   0|   | 0:00:11| 0:00:11|
| Fanout   |  2|     0|     0|   0|  950| 1821|    0|   0|   | 0:00:01| 0:00:12|
| Route    |  3|  6514|   564|   0|    0| 1917|    0|  13|  0| 0:03:13| 0:03:25|
| Route    |  4|  3734|   362|  17|    0| 2003|    0|  13| 42| 0:05:10| 0:08:35|
| Route    |  5|  2166|   517|  14|    0| 2114|    0|  15| 34| 0:05:36| 0:14:11|
| Route    |  6|  1308|   364|  26|    0| 2202|    0|  14| 37| 0:04:45| 0:18:56|
| Route    |  7|   996|   363|  24|    0| 2295|    0|   6| 18| 0:04:16| 0:23:12|
```

The next example shows a Status File that indicates a number of problem areas, based on the guidelines for routing progress.

Figure 5-19 Status File Indicating Problem Areas



1st Pass conflicts 10 X connections

1st Pass Unroutes 5.8% of connections

Unroutes exist after 5th pass

Reduction less than 30% per pass (not even double digits)

```
Nets          =      623   Connections          =      1488
Current Wire =     1594   Reroute wires        =      1594
Completion    =   32.33%   Unconnections        =        31
| ROUTING HISTORY ==================================================================
|      Pass      |  Conflicts |     |     |     |     |    |Red|      CPU Time     |
|    Name    |No.| Cross| Clear|Fail|Unrte| Vias|XTalk|Len.| % |   Pass    |  Total  |
|----------+---+-----+------+----+-----+-----+-----+----+---+---------+---------|
| Fanout    |  1|     0|     2|   0|  950| 1821|    0|   0|   | 0:00:10| 0:00:10|
| Fanout    |  2|     0|     0|   0|  950| 1821|    0|   0|   | 0:00:01| 0:00:11|
| Route     |  3| 14010|   802|  86|   86| 1798|    0|  14|  0| 0:02:29| 0:02:40|
| Route     |  4| 14264|   894|  83|   42| 1789|    0|  13|  0| 0:05:38| 0:08:18|
| Route     |  5| 12750|   873| 186|   39| 1780|    0|  16| 10| 0:07:06| 0:15:24|
| Route     |  6| 12075|  1237| 240|   32| 1793|    0|  14|  2| 0:08:05| 0:23:29|
| Route     |  7| 11279|  1136| 299|   31| 1814|    0|   7|  6| 0:07:24| 0:30:53|
```

Pass 1 Indicators

The following factors can be causes for excessive conflict generation during the first routing pass:

- Design rule errors
- Voltage and ground pins are routed as signals
- Incorrect routing direction(s) are set
- Clearance rules are too big
- Protected wires prevent rip up and reroute

Passes 2 - 5 Indicators

The following factors can be causes for conflict reduction of less than 30% during passes 2 - 5:

- Router is via-starved.
- Not enough signal layers are defined.
- The design is overconstrained with too many rules.
- The design is very complex and dense.

Passes 6 and Beyond Indicators

The following factors can be causes for unroutes after pass 5:

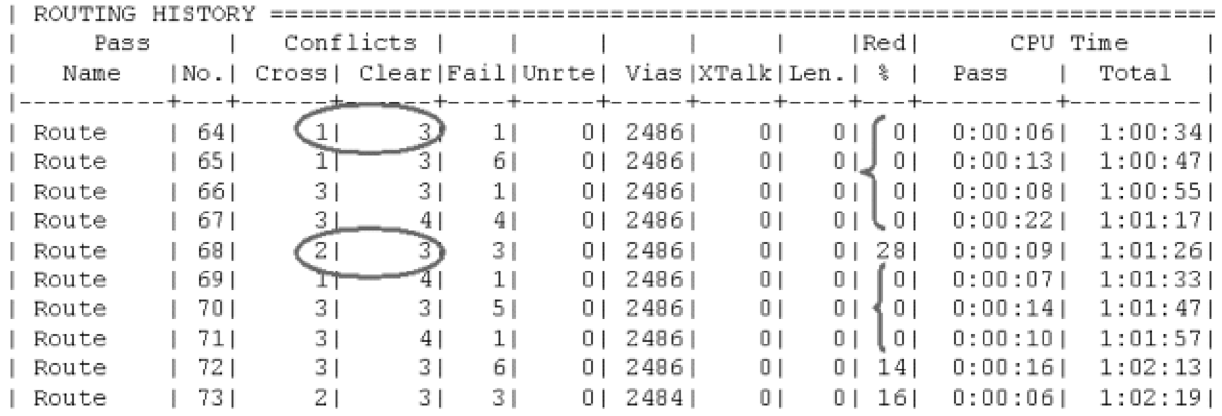
- Keepout regions are blocking pin exits.
- SMD component pins overlap front-to-back, preventing via placement.
- Component pins are placed outside route a boundary or fence.

Consider Halting the Routing Process if the Following Conditions Occur (as shown in the sample Status File)

During the last 10 passes, the conflict reduction is very low.

There are batches of 3 or more passes in a row where the reduction rate is 0 (see below).

Figure 5-20 Halt Routing Progress



```
| ROUTING HISTORY ==============================================================
|     Pass     |   Conflicts |     |     |     |     |    |Red|      CPU Time     |
|   Name    |No.| Cross|  Clear|Fail|Unrte| Vias|XTalk|Len.| % |  Pass    |  Total  |
|----------+---+------+------+----+-----+-----+-----+----+---+---------+---------|
| Route     | 64|     1|     3|   1|    0| 2486|    0|   0|  0|  0:00:06|  1:00:34|
| Route     | 65|     1|     3|   6|    0| 2486|    0|   0|  0|  0:00:13|  1:00:47|
| Route     | 66|     3|     3|   1|    0| 2486|    0|   0|  0|  0:00:08|  1:00:55|
| Route     | 67|     3|     4|   4|    0| 2486|    0|   0|  0|  0:00:22|  1:01:17|
| Route     | 68|     2|     3|   3|    0| 2486|    0|   0| 28|  0:00:09|  1:01:26|
| Route     | 69|     1|     4|   1|    0| 2486|    0|   0|  0|  0:00:07|  1:01:33|
| Route     | 70|     3|     3|   5|    0| 2486|    0|   0|  0|  0:00:14|  1:01:47|
| Route     | 71|     3|     4|   1|    0| 2486|    0|   0|  0|  0:00:10|  1:01:57|
| Route     | 72|     3|     3|   6|    0| 2486|    0|   0| 14|  0:00:16|  1:02:13|
| Route     | 73|     2|     3|   3|    0| 2484|    0|   0| 16|  0:00:06|  1:02:19|
```

Applying Convergence Techniques

During the convergence phase of autorouting, you may be able to apply various techniques to improve the completion rate. These techniques are especially useful if the autorouter is having difficulty resolving conflicts and achieving convergence to 100% completion. They may be essential to completing your design if the autorouter stalls during the convergence phase.

In general, you can improve convergence by performing the following tasks:

- Run 2 clean passes and then perform additional route passes.
    - Decrease the grids.
        - Use grid wire 1
        - Use grid via 1; rule pcb (clearance 15 (type via_via))
    - Relax some rules, if possible.

The main goal in achieving convergence is to change the problem for the autorouter. Whenever you change design constraints and parameters, you force the autorouter to re-evaluate the design and search for new means to complete connections.

Click any of the following links to learn more details about advanced convergence techniques.

Table 5-4 Advanced Convergence Techniques

This technique . . .	Achieves this result . . .
Converging with Autoremove	Eliminates problem routes and opens up new routing channels; forces the autorouter to reroute with new channels.
Converging with Delete Conflicts	Eliminates all wires with conflicts; frees up routing space to allow different routing solutions.
Converging with Filter	Eliminates wires with conflicts when convergence has stopped; allows more efficient manual clean up and routing to complete final connections.

For more information, see the following sections:

Setting the Routing Controls

Using Interactive Routing Tools

Converging with Autoremove

Autoremove – Manual

You can use the Remove Mode to force the autorouter to create an unroute rather than allowing it to restore a wire to its original location. The autorouter would normally restore the wire to its original location if an attempt to reroute the wire fails because a new path cannot be found.

The following command example shows how to set the `Remove` Mode:

```
route 50 16 -remove
```

The Remove mode is applied only to nets with internal priorities less than 200.

Note: This option is best used only if the number of fails is greater than 100 and there are hundreds or thousands of conflicts over ten or more routing passes.

Autoremove – Automatic

By default, `autoremove` is implemented automatically when the autorouter finds that, during the last five passes, the reduction ratio is less than 5 and the fails are greater than 50.

If you set priorities >= 200 on all critical nets, then autoremove will not rip up those nets during subsequent routing passes.

Autoremove – Disable

You can disable `autoremove` entirely by using the following `set` command:

```
set autoremove off
```

Note: Although disabling `autoremove` will prevent the rip up and rerouting of wires, by disabling it, you may end up preventing the router from converging towards 100% completion. If you disable `autoremove`, you should monitor the progress of the autorouter closely, and expect to complete the routing manually.

Converging with Delete Conflicts

You can use the `delete_conflicts` command to eliminate all wires with conflicts. This will force the autorouter to deal with an entirely new routing problem, and may improve the likelihood of converging to 100% completion. To set this condition, enter:

```
delete conflicts
```

You can also use the -`segment` option to delete only wire segments that are in violation. To set this condition, enter:

```
delete conflicts -segment
```

If you apply the `delete_conflicts` command selectively, you can free up routing space and force the autorouter to attempt a different routing solution.

Converging with Filter

The `filter` command removes final routing conflicts and leaves unconnects in their place. You can then complete the remaining connections manually. This is typically done to finish routing a design that has stopped converging. This should be your last resort when trying to achieve routing completion.

To set this condition, enter:

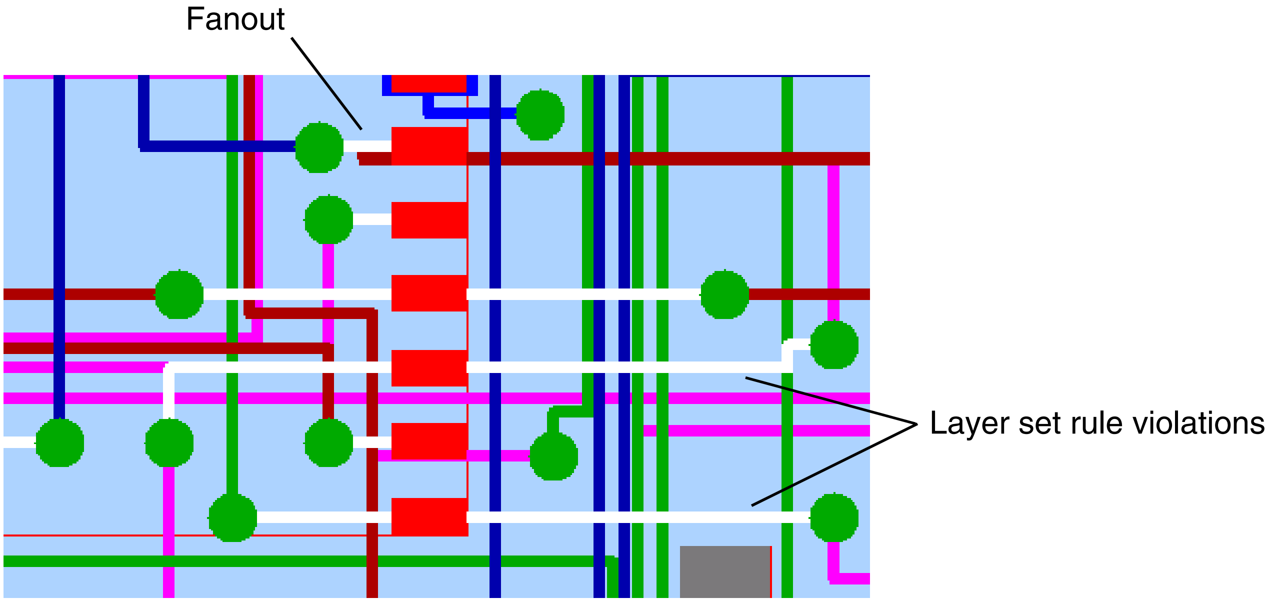
```
filter 5
```

Some layout designers would prefer to clean up the conflicts, rather than running filter and then routing the entire unroutes manually. You should experiment with this technique, and be careful to save intermediate versions of the routed design, to determine your individual preferences.

Layer Set Rule Checking

When enabled, the router performs layer set rule checking. During interactive routing, error messages are displayed in the router Message panel as violations occur. The autorouter displays layer set rule violations graphically by highlighting trace on illegal layers in the design using the current highlight color. An example of this is shown (white trace) in the following figure.

Figure 5-21 Graphical Layer Set Rule Check



To control layer set rule checking

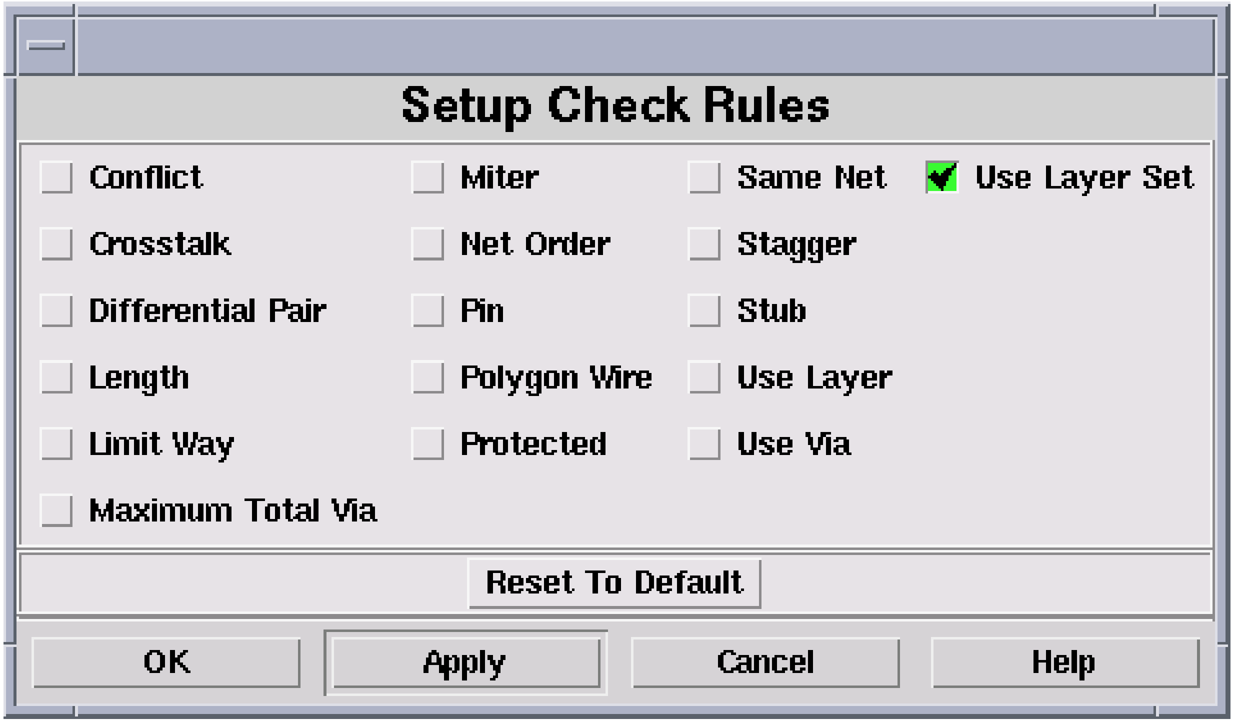
1. Choose *Rules – Check Rules – Setup.*

   The Setup Check Rules dialog box appears.

2. Click to enabling or disabling the *Use Layer Set* check rule option.

3. Click Apply or OK.

Figure 5-22 Setup Check Rules Dialog Box



Setup Check Rules

☐ Conflict
☐ Crosstalk
☐ Differential Pair
☐ Length
☐ Limit Way
☐ Maximum Total Via

☐ Miter
☐ Net Order
☐ Pin
☐ Polygon Wire
☐ Protected

☐ Same Net
☐ Stagger
☐ Stub
☐ Use Layer
☐ Use Via

☑ Use Layer Set

Layer Set Reports

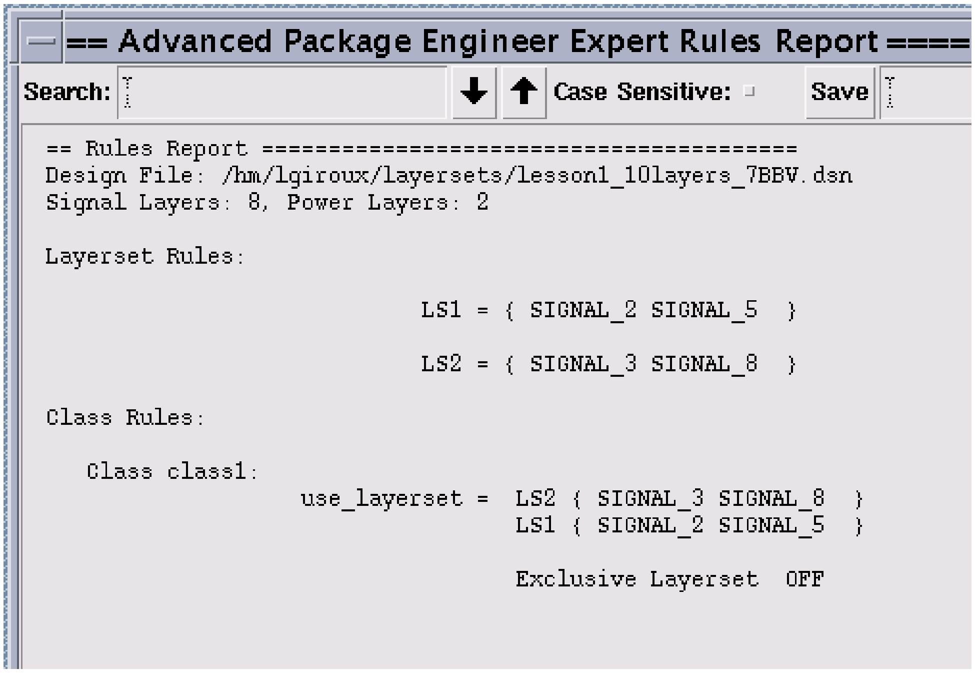
You can generate the following report types that contain layer set information.

- Layer Set Rules
- Net, Class, Group and Group Set Rules
- Layer Set Exception (nets that have routing which do not adhere to the layer set rule and are not reported as a DRC)
- Route Status

To generate a rules report

➤ Choose *Report – Rules – Specify – <report type>*

Figure 5-23 Composite Layer Set and Class Rules Report Examples

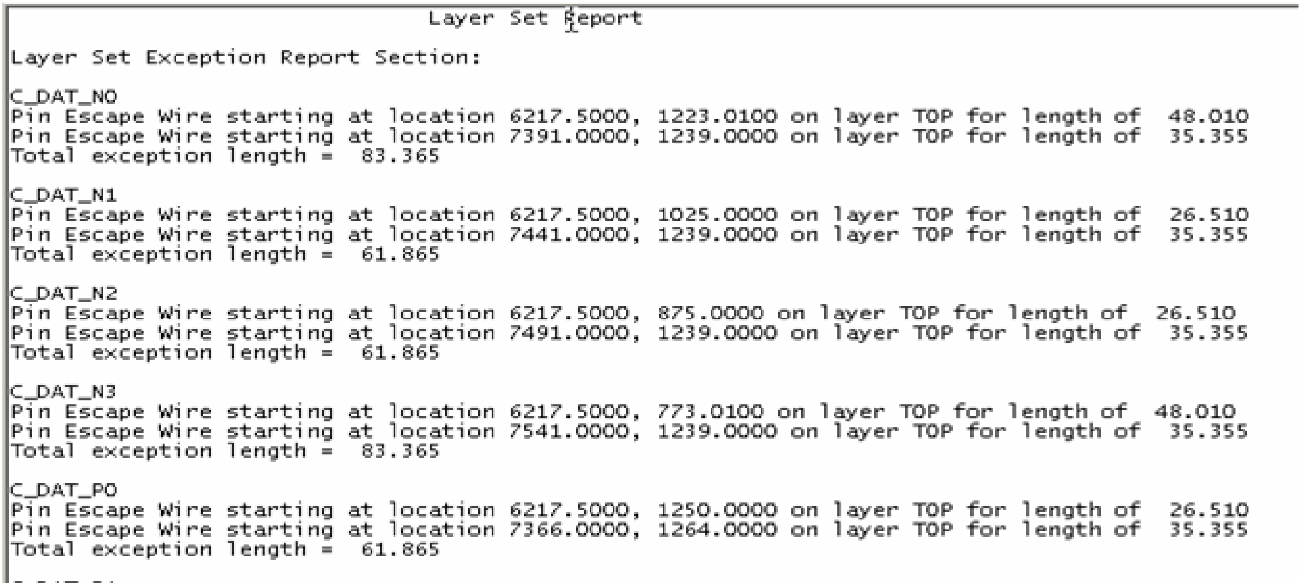


```
== Rules Report ======================================
Design File: /hm/lgiroux/layersets/lesson1_10layers_7BBV.dsn
Signal Layers: 8, Power Layers: 2

Layerset Rules:

                            LS1 = { SIGNAL_2 SIGNAL_5  }

                            LS2 = { SIGNAL_3 SIGNAL_8  }

Class Rules:

   Class class1:
                   use_layerset =  LS2 { SIGNAL_3 SIGNAL_8  }
                                   LS1 { SIGNAL_2 SIGNAL_5  }

                                   Exclusive Layerset  OFF
```

To generate a Layer Set Exception report

➤ Choose *Report – Specify – Layer Set*

Figure 5-24 Layer Set Exception Report Example



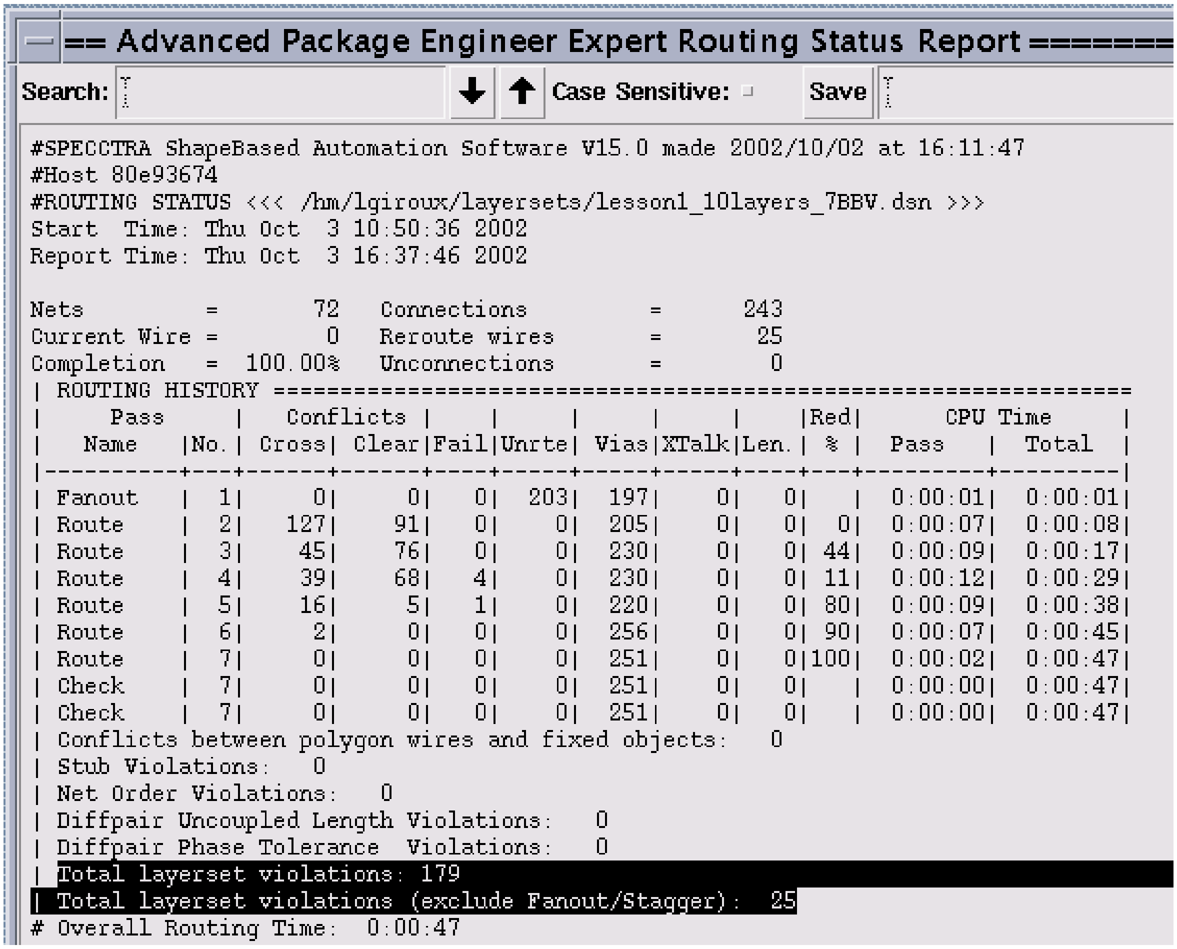
```
                               Layer Set Report

Layer Set Exception Report Section:

C_DAT_N0
Pin Escape Wire starting at location 6217.5000, 1223.0100 on layer TOP for length of  48.010
Pin Escape Wire starting at location 7391.0000, 1239.0000 on layer TOP for length of  35.355
Total exception length =  83.365

C_DAT_N1
Pin Escape Wire starting at location 6217.5000, 1025.0000 on layer TOP for length of  26.510
Pin Escape Wire starting at location 7441.0000, 1239.0000 on layer TOP for length of  35.355
Total exception length =  61.865

C_DAT_N2
Pin Escape Wire starting at location 6217.5000, 875.0000 on layer TOP for length of  26.510
Pin Escape Wire starting at location 7491.0000, 1239.0000 on layer TOP for length of  35.355
Total exception length =  61.865

C_DAT_N3
Pin Escape Wire starting at location 6217.5000, 773.0100 on layer TOP for length of  48.010
Pin Escape Wire starting at location 7541.0000, 1239.0000 on layer TOP for length of  35.355
Total exception length =  83.365

C_DAT_P0
Pin Escape Wire starting at location 6217.5000, 1250.0000 on layer TOP for length of  26.510
Pin Escape Wire starting at location 7366.0000, 1264.0000 on layer TOP for length of  35.355
Total exception length =  61.865
```

Route Status Report

To generate a route status report

➤ Choose *Report – Route Status*

Figure 5-25 Route Status Report Example


== Advanced Package Engineer Expert Routing Status Report =======
Search:
Case Sensitive: □
Save

```
#SPECCTRA ShapeBased Automation Software V15.0 made 2002/10/02 at 16:11:47
#Host 80e93674
#ROUTING STATUS <<< /hm/lgiroux/layersets/lesson1_10layers_7BBV.dsn >>>
Start  Time: Thu Oct  3 10:50:36 2002
Report Time: Thu Oct  3 16:37:46 2002

Nets         =      72   Connections        =     243
Current Wire =       0   Reroute wires      =      25
Completion   =  100.00%  Unconnections      =       0
| ROUTING HISTORY ===============================================================
|     Pass     |   Conflicts |    |     |     |     |    |Red|      CPU Time     |
|   Name   |No.| Cross| Clear|Fail|Unrte| Vias|XTalk|Len.| % |  Pass    |  Total  |
|----------+---+------+------+----+-----+-----+-----+----+---+---------+---------|
| Fanout   |  1|     0|     0|   0|  203|  197|    0|   0|   |  0:00:01|  0:00:01|
| Route    |  2|   127|    91|   0|    0|  205|    0|   0|  0|  0:00:07|  0:00:08|
| Route    |  3|    45|    76|   0|    0|  230|    0|   0| 44|  0:00:09|  0:00:17|
| Route    |  4|    39|    68|   4|    0|  230|    0|   0| 11|  0:00:12|  0:00:29|
| Route    |  5|    16|     5|   1|    0|  220|    0|   0| 80|  0:00:09|  0:00:38|
| Route    |  6|     2|     0|   0|    0|  256|    0|   0| 90|  0:00:07|  0:00:45|
| Route    |  7|     0|     0|   0|    0|  251|    0|   0|100|  0:00:02|  0:00:47|
| Check    |  7|     0|     0|   0|    0|  251|    0|   0|   |  0:00:00|  0:00:47|
| Check    |  7|     0|     0|   0|    0|  251|    0|   0|   |  0:00:00|  0:00:47|
| Conflicts between polygon wires and fixed objects:   0
| Stub Violations:   0
| Net Order Violations:   0
| Diffpair Uncoupled Length Violations:   0
| Diffpair Phase Tolerance  Violations:   0
| Total layerset violations: 179
| Total layerset violations (exclude Fanout/Stagger):  25
# Overall Routing Time:  0:00:47
```

Note: In the report, the line item *Total layerset violations* includes all trace highlighted in the design by layer set rule checking. Although highlighted, trace that falls within an smd_escape distance is not considered a layerset violation. Therefore, actual layerset violations are reflected in the line item *Total layerset violations (exclude Fanout/Stagger)*.

Differential Pair Checking

The router provides tools to check for edge coupling and phase violations on differential pairs. After violations are detected, you can interactively examine the pair (at the segment level) by generating graphical feedback in the design to show:

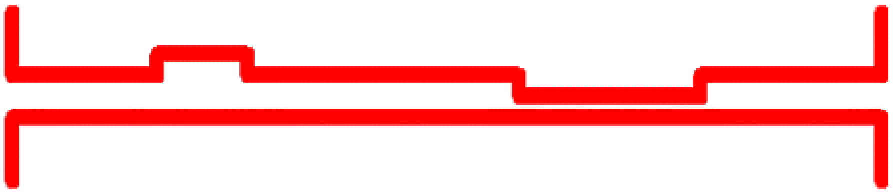
- tolerance contours
- uncoupled segments
- coupling points
- phase violation penalty boxes

Additionally, edge coupling and phase violation information is included in the Pairs report. You can view these results by choosing *Report – Specify – Pairs*.

Coupling Checks

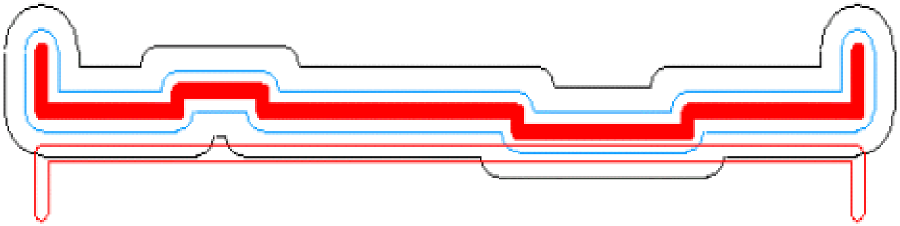
Using primary gap and gap tolerances (plus / minus), each differential net defines a special *coupling area* that should totally encompass its partner net. If any edge of its partner falls beyond this area, a violation occurs. The following figures illustrate the checking mechanism that is used.

Figure 5-26 Net Pair



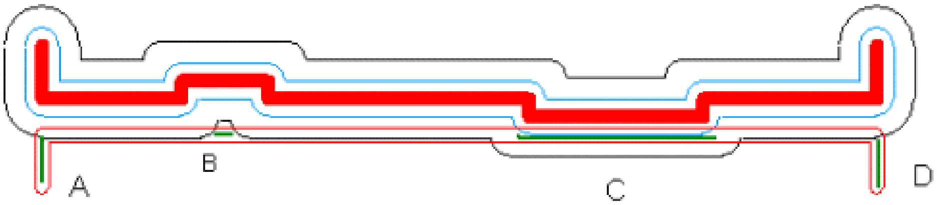
In the following figure, the top net of the pair in the previous figure is shown with its gap tolerance contours displayed. The blue line represents (primary gap - tolerance minus) with the black line representing (primary gap + tolerance plus).

Figure 5-27 Top Net with Gap Tolerance Contours Displayed



According to the coupling check, there are four violations as indicated in green with A, B, C, D in the following figure.

Figure 5-28 Coupling Check and Partner Net Violation Feedback



A, B, D are violations because both edges of that part of the net are outside the black contour. C is a violation because one edge of that part of net is inside the blue contour.

Note that the net between A and B does not cause a violation even though part of its edge falls outside of black contour. This is because one edge of the central line of the net is still inside the black contour.

Similarly, the following two figures show how checking is applied to the partner net.

Figure 5-29 Bottom Net with Gap Tolerance Contours Displayed

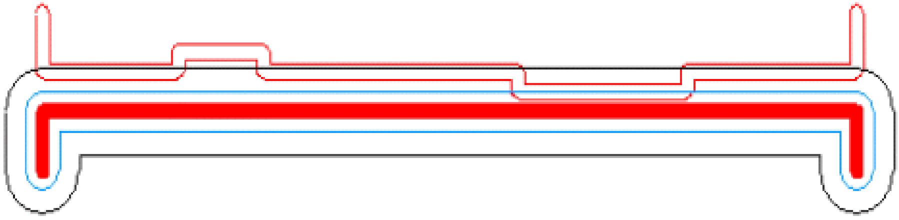
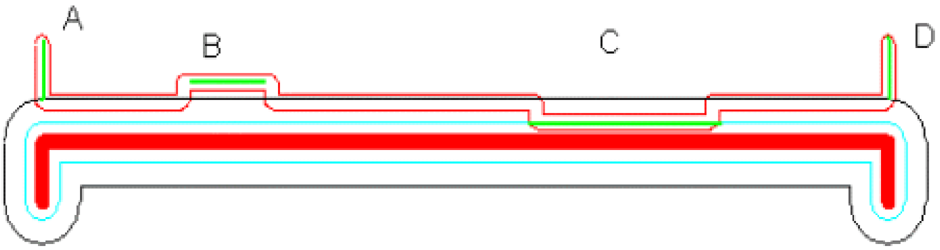


Figure 5-30 Coupling Check and Partner Net Violation Feedback



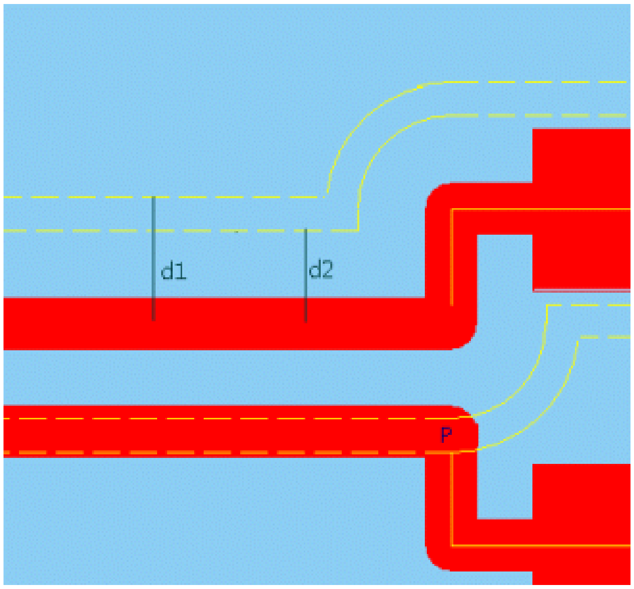
Coupling Violation Area

Wire width is defined by two edges. An uncoupled segment occurs when one of the following conditions occur:

- at least one edge of the wire falls outside of tolerance(-) contour.
- both edges of the wire fall outside of tolerance (+) contour.

However, all coupling checks are performed using the tolerance(+/-) contours and central line of wires. The example in the following figure illustrates this.

Figure 5-31 Coupling Check Example



Note that at point P, the central line is crossing the tolerance(+) contour and creates an uncoupled segment.

The distance between the central line and the tolerance(+) contour is d1.

The distance between the central line and the tolerance(-) contour is d2.

The pair of nets have the following constraints:

- line width of 8
- primary gap of 8
- tolerance(+) 3
- tolerance(-) 2

Therefore:

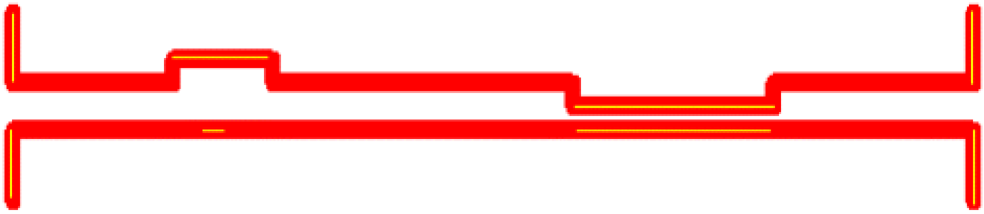
d1 = 4(half line width) + 8(primary gap) + 3(tol +) + 4(half line width) = 19

d2 = 4(half line width) + 8(primary gap) - 2(tol -) + 4(half line width) = 14

Examining Coupling Violations

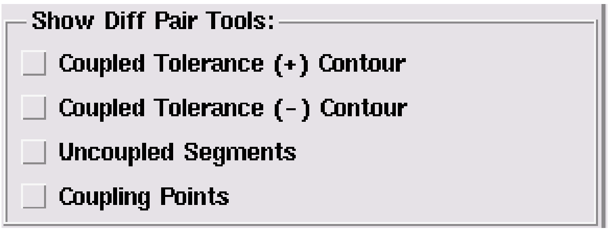
Once the router checks for differential pair coupling violations, all uncoupling parts on both nets in the pair(s) are shown with a solid line in the design to indicate the segments of violation on the nets. An example of this is shown in the following figure.

Figure 5-32 Uncoupled Net Segments



At this point, you can examine each of these segments interactively to determine a solution using the graphical feedback provided by the differential pair coupling tools in the Interactive Routing Setup dialog box. These tools are shown in the following figure.

Figure 5-33 Diff Pair Tools Setup



Show Diff Pair Tools:

□ Coupled Tolerance (+) Contour

□ Coupled Tolerance (-) Contour

□ Uncoupled Segments

□ Coupling Points

To set up and use the interactive differential pair coupling tools

1. Choose *[RMB] – Setup.*

   The Interactive Routing Setup dialog box appears.

2. In the Show Diff Pair Tools section of the General tab, click the checkbox of the tools that you want to enable.

3. Click *Apply* or *OK*.

4. Click on a differential net wire that you want to check.

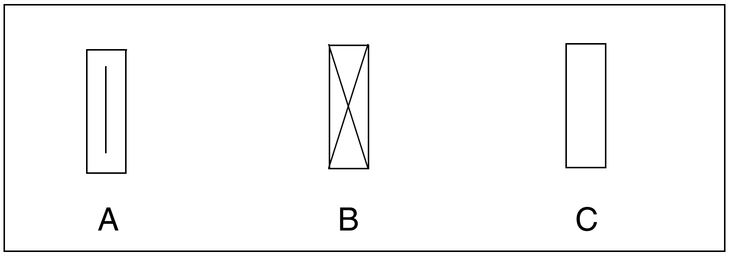
   Graphical feedback is displayed for the selected wire.

5. If desired, click-right on the wire to change the tool setup.

Coupling Point Feedback

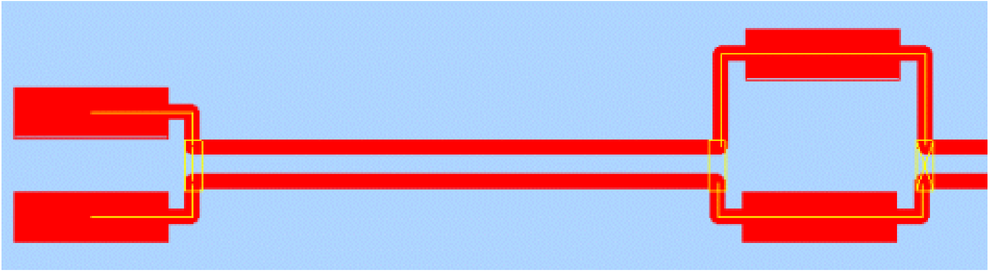
There are three different coupling points that can be displayed when you enable the Coupling Points tool. These are shown in the following figure.

Figure 5-34 Coupling Points



- A box with a line inside indicates coupling points with no phase control violation (A).
- A box with a cross inside indicates coupling points with phase control violation (B).
- An empty box indicates uncoupling points (C).

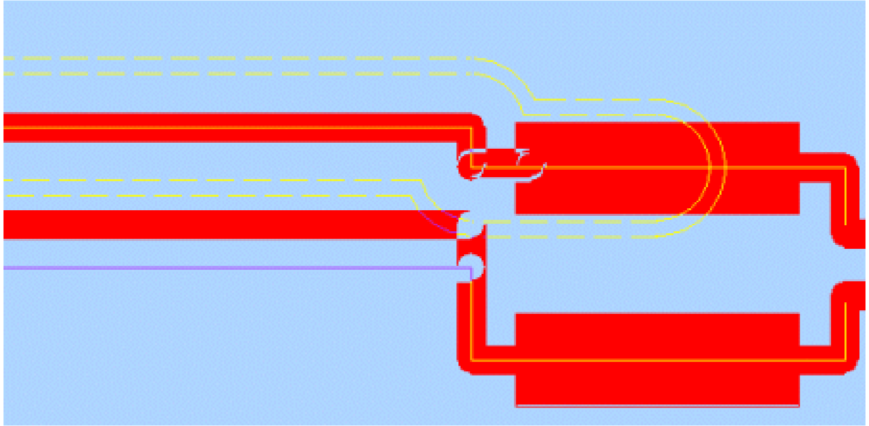
Figure 5-35 Coupling Points Displayed on a Pair



Contour Display in Interactive Route Mode

Within interactive route mode, the contour of the *partner ne*t is displayed as you interactively edit a wire to correct coupling violations. An example of this is shown in the following figure. This feedback enables you to easily place the wire within the valid area.

Figure 5-36 Partner Net Contour Display



Differential Pair Tool Color

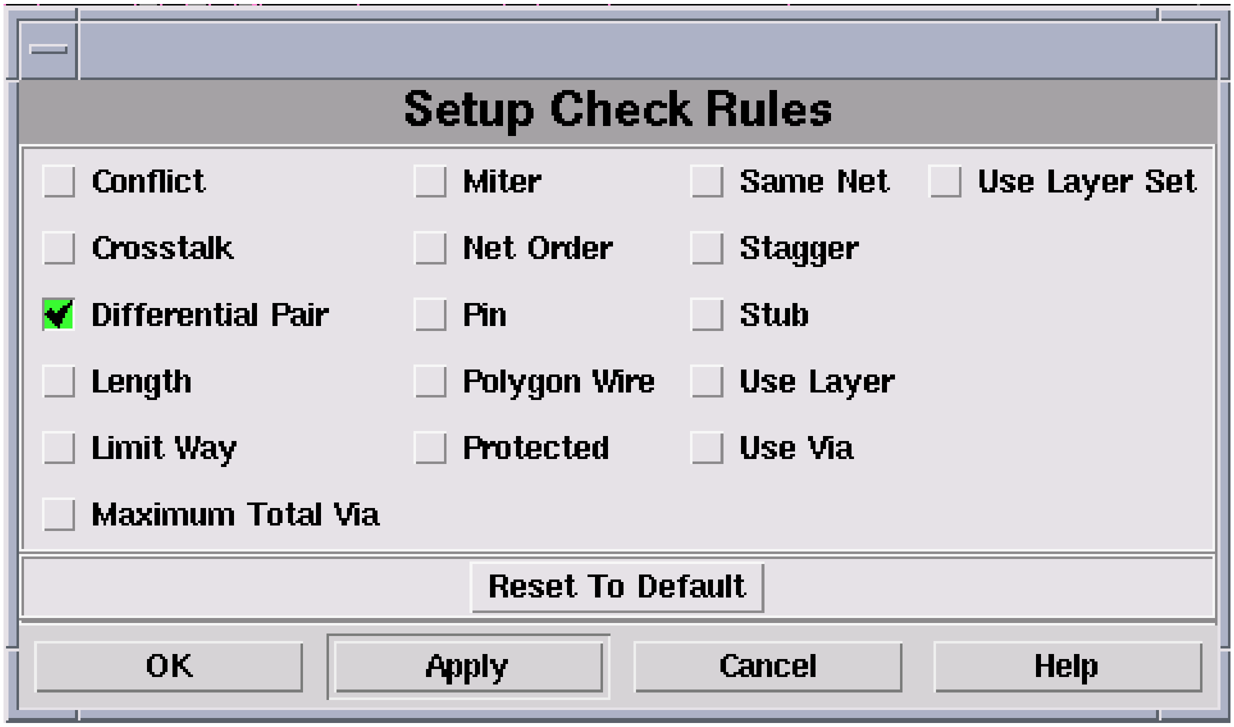
Differential pair tools are displayed in the color chosen for *Select*. In other words, no specific color is assigned for drawing tolerance plus/minus contours, uncoupled segments and coupling points. See View – Color Palette for details on changing the *Select* color.

Differential Pair Rule Checking

Rule Setup and Check Start

You set up differential pair rule checking using the Setup Check Rules dialog box shown in the following figure. By default, differential pair checking is enabled.

Figure 5-37 Setup Check Rules Dialog Box



Setup Check Rules

☐ Conflict
☐ Crosstalk
☑ Differential Pair
☐ Length
☐ Limit Way
☐ Maximum Total Via
☐ Miter
☐ Net Order
☐ Pin
☐ Polygon Wire
☐ Protected
☐ Same Net
☐ Stagger
☐ Stub
☐ Use Layer
☐ Use Via
☐ Use Layer Set

To set up differential pair rule checking

1. Choose *Rules – Check Rule – Setup*.

   The Setup Check Rules dialog box appears.

2. Either check (enable) or uncheck (disable) the checkbox next to *Differential Pair*.
3. Click *Apply* or *OK*.

To start differential pair rule checking on a design

➤ Choose *Rules – Check Rules – Routing*.

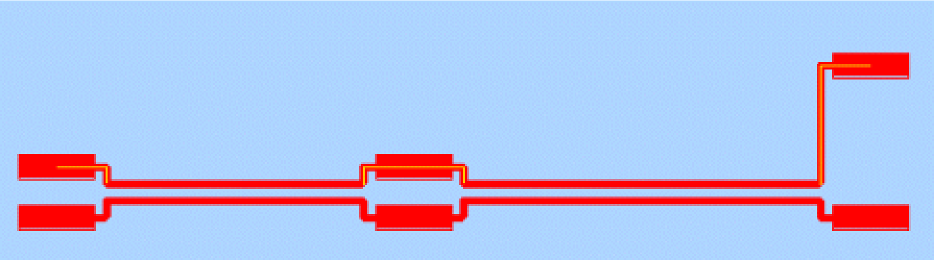
or

Choose *Rules – Check Rules – All*.

Maximum Uncoupled Length Check

The router calculates uncoupled length on differential nets. If the summation of uncoupled length on one net exceeds `maximum_uncoupled_length`, then all uncoupled segments on that net are displayed. In the following figure, the upper net has an uncoupled length violation, but not the lower one.

Figure 5-38 Uncoupled Length Violation Example



Note: At the group level, the max uncoupled length check can be based on individual from-to in the group or based on the total from-to in the group (XNets). See Handling Extended Nets for further details. However, at the class level, uncoupled length rule applies to individual net only.

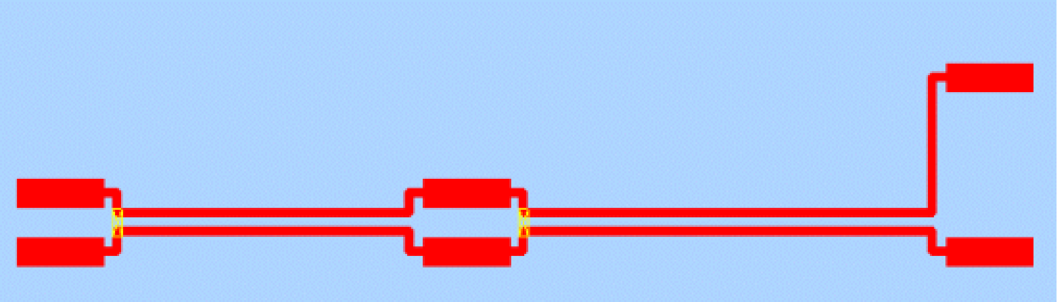
Phase Control Check

Phase control checking is done from pin to pin. Gather length is always taken into account during this check. In other words, the `ignore_gather_length` rule does not apply to phase control.

Uncoupled segment length is checked from the first pin to the last pin. If the length difference at the pins between two nets exceeds the phase tolerance, a phase violation occurs. To indicate a phase violation, a penalty box (a box with cross inside) is displayed on the first coupling point of that wire.

If a net has multiple wires, then a penalty box is displayed on every wire in that net as shown in the following figure.

Figure 5-39 Phase Violation



To set up phase checking for differential pairs

1. Choose *Rules – <rule level> – Differential Pairs*.

   The Differential Pair Rules dialog box appears.

2. Enter a positive tolerance value in the *Phase Tolerance* data entry box.

   Note: If the tolerance is left unspecified (-1), phase checking is disabled.

3. Click *Apply* or *OK*.

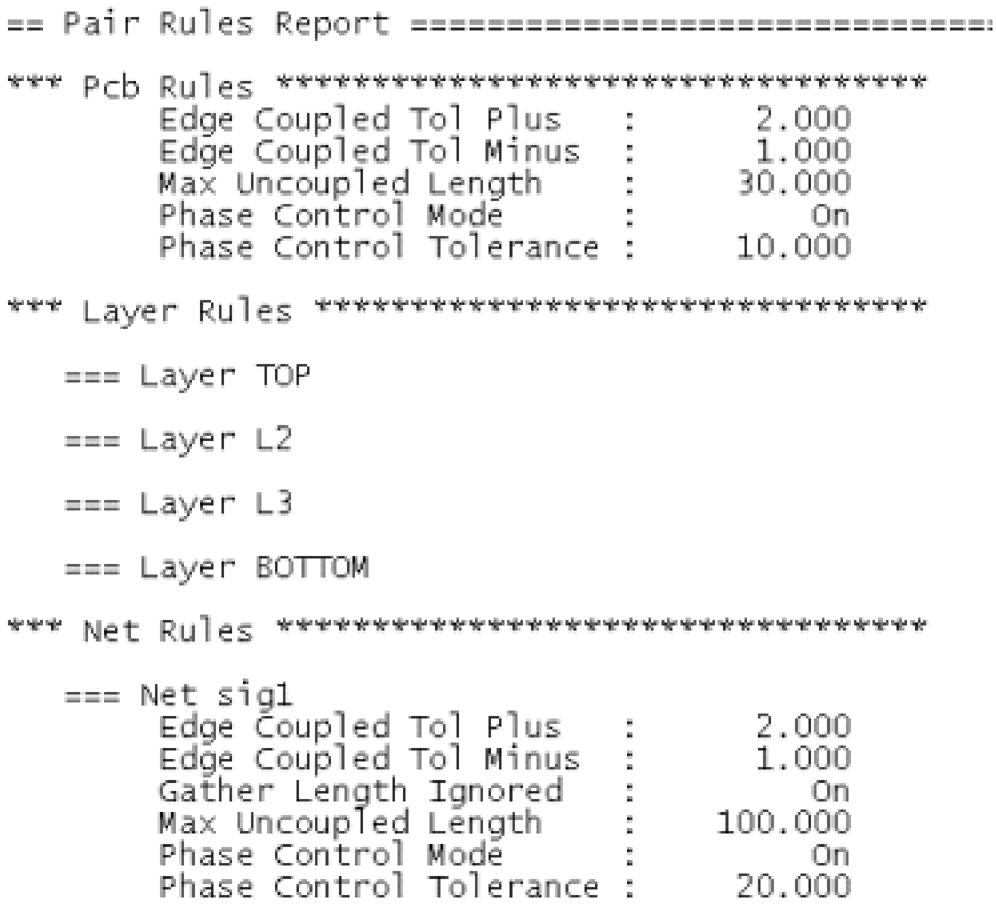
Differential Pair Rules and Check Results Reporting

To report differential pair rules

1. Choose *Report – Specify – Pairs Rules*.

   A rule report is displayed in the report window.

Figure 5-40 Differential Pair Rule Report



```
== Pair Rules Report ==================================

*** Pcb Rules **********************************
        Edge Coupled Tol Plus   :      2.000
        Edge Coupled Tol Minus  :      1.000
        Max Uncoupled Length    :     30.000
        Phase Control Mode      :         On
        Phase Control Tolerance :     10.000

*** Layer Rules ********************************

   === Layer TOP

   === Layer L2

   === Layer L3

   === Layer BOTTOM

*** Net Rules **********************************

   === Net sig1
        Edge Coupled Tol Plus   :      2.000
        Edge Coupled Tol Minus  :      1.000
        Gather Length Ignored   :         On
        Max Uncoupled Length    :    100.000
        Phase Control Mode      :         On
        Phase Control Tolerance :     20.000
```

To report differential pair check results

1. Choose *Report – Specify – Pairs*.

   A check report is displayed in a report window.

The following figures show a differential pair test case and it’s report.

Figure 5-41 Differential Pair Test Case

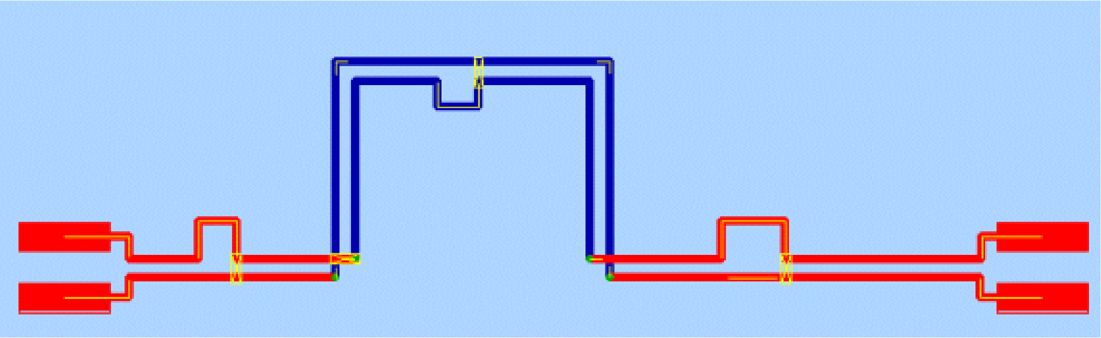
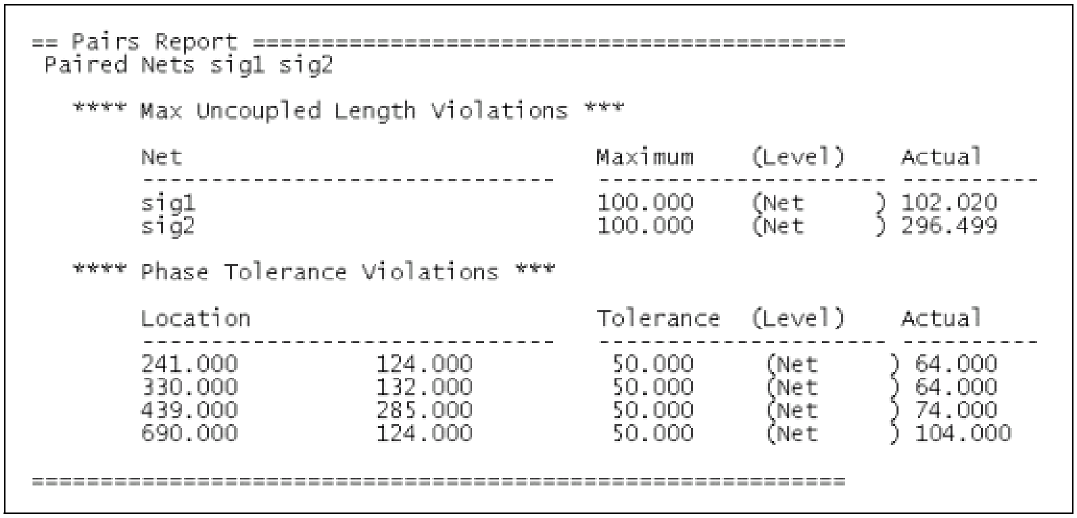


Figure 5-42 Check Results Report



```
== Pairs Report ============================================
 Paired Nets sig1 sig2

   **** Max Uncoupled Length Violations ***

        Net                                Maximum    (Level)    Actual
        -------------------------------    ---------------------  ----------
        sig1                               100.000    (Net      ) 102.020
        sig2                               100.000    (Net      ) 296.499

   **** Phase Tolerance Violations ***

        Location                           Tolerance  (Level)    Actual
        -------------------------------    ---------------------  ----------
        241.000          124.000            50.000     (Net      ) 64.000
        330.000          132.000            50.000     (Net      ) 64.000
        439.000          285.000            50.000     (Net      ) 74.000
        690.000          124.000            50.000     (Net      ) 104.000

============================================================
```

Meeting DFM Requirements

The router provides various tools to help you optimize the manufacturability of the finished design. This process, known as Design For Manufacturing (DFM), can require various "finishing" tasks that will yield higher quality designs with fewer manufacturing and assembly defects.

The links listed below provide more detailed information about how to use the router to meet DFM requirements.

- Generating and Controlling Test Points
- Reducing unnecessary wrong way routes and jogs with the `critic` command
- Mitering 90 Degree Corners
- Optimizing trace spacing with the `spread` command
- Performing full design DRC with the *Rules – Check Rules – Setup* dialog box

Generating and Controlling Test Points

After all routing is completed, use the `testpoint` command to add test points to your design and to set the test point mode for the current autorouting session. Test points can be existing vias as well as through pins. You can assign test points to the front, back, or both sides of the design, and you can set a test via grid and minimum center-to-center test point spacing. You can identify a particular via for test points, or allow any via to be used.

Adding Test Points to All Nets

Adding Test Points to Specific Nets

Specifying a Probing Layer and Test Via Type

Using Through Pins as Test Points

Protecting Test Vias

Setting a Testpoint to Outline Clearance

Note: The `testpoint` command overrides the pcb `testpoint_rule`. For example, if you enter `testpoint` without options, the operation proceeds with the `testpoint` command default settings, and ignores any rules set at the pcb level with the `testpoint_rule`. Rules set at the higher levels are not affected. To assign test point rules by net, class, or for the entire design, after you set the rule, and add them during the next route, clean or filter pass, see the testpoint rule.

You should assign test points by using the `testpoint` command after routing is complete but before using the `clean`, `spread`, and `miter` commands.

The `testpoint` command usually is not entered in Do files. Routing should be completed before you generate test points.

Rather than using the `testpoint` command, you can choose *Autoroute – Post Route – Testpoints*.

Adding Test Points to All Nets

You should use the `testpoint` command after routing is completed so the vias do not contribute to routing channel congestion during rip-up and reroute operations. Use the `clean` command after the `testpoint` command to minimize tjunctions and remove excess vias.

For example, to create test points for all nets by using via TSTVIA on the back side of the design with a minimum center-to-center spacing of 50 mils, enter the following commands:

```
testpoint (side back) (center_center 50) (use_via TSTVIA)
clean 2
```

Adding Test Points to Specific Nets

You add test points to specific nets by selecting the nets and using the `testpoint` command. For example, to add test points to all nets in class CLASSX, enter the following commands:

```
select class CLASSX
testpoint
unselect all nets
clean 2
```

Specifying a Probing Layer and Test Via Type

By default, the bottom layer is the probing layer. You can specify a probing layer and a test via type by using the `testpoint` command. For example, to specify the top layer for test probing and the test via type as a single layer pad, enter the following commands:

```
testpoint (side front) (use_via via_1_1)
```

```
clean 2
```

Using Through Pins as Test Points

If you set the `pin_allow` rule, the autorouter attempts to use through-pins as test points. If a net does not contain through-pins that satisfy all the testpoint requirements, the autorouter attempts to use vias.

For example, to use through-pins as test points on a grid of .1 inches, enter the following commands:

```
unit inch
testpoint (pin_allow on) (grid 0.1)
```

Protecting Test Vias

Test vias are usually protected if a test fixture for the design already exists. Use the Wires, Routes, or Session File from a previous autorouting session and protect the test vias so that they will not be changed by subsequent autorouting operations.

For example, to load the Routes File `rev2.rte` and protect the test points for all nets, enter the following commands:

```
read routes rev2.rte
protect all testpoints
```

Setting a Testpoint to Outline Clearance

You can control the amount of space between the edge of the component outline and the edge of a testpoint. For example, to set the clearance between test points and the component outlines to .1 inch, enter the following commands:

```
unit inch
testpoint (image_outline_clearance 0.1)
```

You can also control the distance permitted between a component outline and the center of a testpoint. For example:

```
unit inch
testpoint (comp_edge_center 0.15)
```

Mitering 90 Degree Corners

You change 90 degree corners to 135 degrees by using the `miter` command. You should complete all routing, add test points, and complete four clean passes before you use `miter`.

Note: The `miter` command is usually not included in a Do file because all routing should be completed before wiring is mitered.

Use the `unmiter` command to restore 135 degree corners to 90 degrees.

Creating Round Corners

You can create round corners by using the `miter` command. You should complete all routing, add test points, and complete four clean passes, before you use `miter`. For example, to change 90 degree corners to round corners, enter the following command:

```
miter (style round)
```

The `unmiter` command does not remove round corners. You should save the routing to a wires or Routes File before you use `miter` to create round corners.

Adding Extra Wire-to-Object Clearances

After you run all route and clean passes, but before you use the `miter` command, you can use the `spread` command to add space between wires, and between wires and pins. For example, if you want to add six mils of extra clearance between wires and through-pins and eight mils of extra clearance between wires, enter the following command:

```
spread (extra 6 (type wire_pin)) (extra 8 (type wire_wire))
```

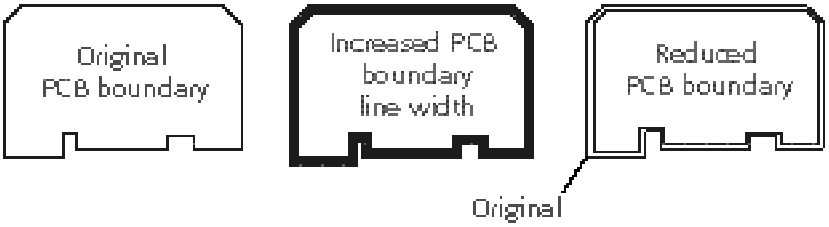
Note: Instead of entering the `spread` keyboard command, you can choose *Autoroute – Post Route – Spread Wires*.

The default is to add one-half the default clearance value between adjacent wires, between wires and through-pins, and between wires and SMD pads.

Creating a Special Wire – Boundary Clearance

The autorouter uses wire-to-area clearance rules to determine the wire-to-design boundary clearance. If you want to create a special wire-to-design boundary clearance, adjust the design boundary in your layout system by changing the line width of your design boundary to a value that is double the clearance you want or by reducing the design boundary so that it corresponds to the actual clearance boundary.

Figure 5-43 Wire – Design Boundary Clearance



Applying Engineering Changes

You can apply engineering changes to completed designs. You should follow the guidelines listed below when you use the autorouter to apply engineering changes to a completed design.

- Make sure the autorouter environment settings such as grids, rules, layer selection, and layer direction are identical to those settings used during the previous autorouting session.
- Read the rules from the previous Do or Did File. (If you use the Did File, you must edit the file.) For more information, see
    - ❑ Understanding the Batch Script
    - ❑ `did_file` command
- Protect wires for critical nets and test vias.

To apply engineering changes to completed designs

1. Re-extract the Design File after the netlist changed.
2. Load the design in the autorouter with the Wires or Routes File from the previous session. If you load a Routes File, use the `ignore_net` option with the `read routes` command.

   Incomplete wires and extra wires remain until the first routing pass.

Note: If the `miter` command was used in the previous session, you should use the Routes File you saved before these operations were performed or use the `unmiter` command to remove the 135 degree corners.

To examine incomplete wires, use *View – Highlight – Incomplete Wires*.

To delete incomplete wires, use *Edit – Delete Wires – Incomplete Wires*.

Using Interactive Routing Tools

You begin an interactive routing session with preparation steps, such as setting interactive routing environment options, routing rules, routing layers, and other parameters. To initiate some preparation steps, such as setting rules or grids, you use commands in the routing menu bar pulldown menus. Other preparation steps you initiate by using commands in the right mouse button ([RB]) popup menus.

After the preparation steps, you use various left mouse button ([LB]) modes, which you can set using tool bar icons or commands on the [RB] popup menus. You can return to any preparation step during your interactive session and make appropriate changes.

Using the [LB] modes, you can perform the following tasks:

- Measure distances and query objects.
- Create, edit, move, copy, cut, and delete wire segments and vias, remove extra wire bends, and repair nets by removing wires and vias that violate fromto ordering rules.
- Add, edit, move, copy, cut or cut out, merge, and delete wiring polygons, as well as move, copy, and cut or cut out keepout areas.
- Edit net topologies.
- Change via attributes, wire segment widths, and net connectivity on wire segments and wiring polygons.
- Change layer assignments on wire segments, wires, and wiring polygons.

The interactive routing environment contains three alternate [RB] popup menus:

- Interactive Routing menu
- Polygon Editing menu
- Topology Editing menu

When you start a session, the default [RB] menu is Interactive Routing.

Preparing for Interactive Routing

Using the Interactive Routing Menu

You set interactive routing and editing modes by using the right mouse-button ([RB]) popup menus. You also use the [RB] menus to set or change your interactive routing setup, to select and unselect design objects, to delete objects, to choose options for the current editing or routing mode, and to choose an alternate interactive menu.

The interactive routing environment contains three alternate [RB] popup menus:

- Interactive Routing menu
- Polygon Editing menu
- Topology Editing menu

You can change to an alternate menu by choosing it from the current menu. When you start a session, the current [RB] menu is Interactive Routing.

After you choose a mode from an [RB] menu, you use the left mouse-button ([LB]) to interactively route and edit your design. For example, if you choose Edit Route Mode from the Interactive Routing menu, you use [LB] to interactively route or edit wires and vias.

To access the Interactive Routing menu

1. Click the Route Mode button in the tool bar.
2. Move the pointer into the design area.
3. Press [RB].

You use the Interactive Routing menu to set modes for routing, editing, moving, and copying wire segments and vias; for cutting wire segments; for changing wire widths, via properties, and wire connectivities, for repairing nets with wires that violate fromto ordering rules, and for deleting wire segments, vias, wiring polygons, and keepout areas.

The interactive Routing menu includes the following:

- Setup
- Select
- UnSelect All Objects
- Delete

- Edit Route Mode
- Move Mode
- Copy Route Mode
- Critic Route Mode
- Cut Segment Mode
- Change
- Undo

The alternate interactive menus include:

- Polygon Editing Menu
- Topology Editing Menu

Note: The right-button menu accesses three different interactive menus: Interactive Routing, Polygon Editing, and Topology Editing. The active menu is indicated at the top of the menu when you press [RB].

You can interactively route with or without wire or via grids (see the Setup dialog box).

The Merge Keepout and Delete Keepout modes are not included in the Interactive [RB] menu. You can use *Define – Keepout – Merge Mode* to set [LB] to Merge Keepout mode or use *Define – Keepout – Delete Mode* to set [LB] to Delete Keepout mode.

Using the Polygon Editing Menu

You use the Polygon Editing menu to set modes for adding, editing, moving, and copying polygons; for cutting out an area of a polygon; and for changing the net assignment and layer of a wiring polygon. You can also use the Polygon Editing menu to set the Merge Wiring Polygon mode. To access any of these options, press [RB], select the Polygon Editing Menu and choose the desired command.

- Add/ Edit Polygon Mode
- Move Mode
- CopyPolygon Mode
- Cut Polygon Mode

- Change Layer Mode
- Merge Wiring Polygon Mode

The other interactive menus are

- Interactive Routing Menu
- Topology Editing Menu

Using the Topology Editing Menu

Interactive topology editing consists of two menus: Topology Editing and Edit Topology. Initially, you use the Topology Editing menu to choose how you pick the net you want to edit. After you pick a net, you use the Edit Topology menu to assign pin attributes; add, move, and delete virtual pins; reorder pin-to-pin connections; fix and unfix pins; set fromto rules; forget fromto rules; forget net rules, and forget connection ordering for nets.

The function of the Topology Editing and Edit Topology menus are combined in the Topology Edit dialog box, which opens if you click Alternate Topology Editing on the [RB] menu.

The pick-net methods you can use are

- Pick Net Mode
- Pick Net By List
- Pick Net By Class
- Pick Net By Bundle

The topology editing modes you can use are

- Pin Attribute Mode
- Add Virtual Pin Mode
- Delete Virtual Pin Mode
- Move Virtual Pin Mode
- Reorder By Pin Mode
- Reorder By Comp Mode

- Fix/Unfix Pin Mode
- Set Fromto Rules Mode
- Forget Fromto Mode

The other interactive menus are

- Interactive Routing Menu
- Polygon Editing Menu

Setting the Interactive Routing Environment

Before you begin interactive routing, set up the routing environment to satisfy basic design constraints and editing preferences.

To set your environment for interactive routing

➤ Press [RB] and choose *Setup*.

The Interactive Routing Setup dialog box opens. You can use this dialog box to set global (pcb) width and clearance rules and to set interactive routing controls in five different categories: General, Measure, Bus, Style, and Move/Copy.

The five tabs in the dialog box contain the controls for these categories. The controls on the General tab appear by default when you open the dialog box. To display the controls on a different tab, click its label. See Setting PCB Wire Width and Clearance Rules for details on how to set your interactive routing environment by using the Interactive Routing Setup dialog box.

On the General tab:

- Setting via assistance
- Controlling jogs during push routing
- Displaying timing/length rule constraints
- Setting snap to pin origin
- Setting push routing
- Setting check region

- Controlling redundant wiring
- Allowing floating net routing
- Allowing automatic polygon merging
- Allowing automatic shielding
- Allowing automatic length adjustments
- Allowing multiple pin connections on a single wire
- Allowing same net checking

On the Measure tab

- Setting Measure mode options
- Controlling Measure mode output

On the Bus tab

- Enabling bus routing
- Enabling tandem pair routing
- Enabling fit via pattern
- Setting spacing for gathered wires

On the Style tab

- Setting the pointer style
- Setting the routing snap angle
- Setting the routing style

The Move/Copy tab contains controls that you use to move objects in Move mode or copy polygons in Copy Polygon mode.

After you enter or change settings in the dialog box, click *OK* or *Apply*.

Note: You can change interactive routing environment settings at any time during your routing session. Many of the interactive editing modes have [RB] menus that include a Setup command. Pressing [RB] brings up one of these menus or the Interactive Routing menu. Choose *Setup* and make the changes you want in the Interactive Routing Setup dialog box.

Setting PCB Wire Grids Interactively

You can set PCB wire grids and PCB via grids that apply to both autorouting and interactive routing operations. Each grid can be uniform or nonuniform, and you can set both grid spacings and grid offsets. Grid offsets are measured from the design origin.

For wires, you can also set major grid marks on the wire grid and control how the grid displays. For vias, you can also set separate grids for individual vias.

During interactive routing, you can change PCB wire and via grids by resetting grid values. After you change a grid value, the interactive router follows the new grids during routing. Wires and vias digitized before the current grid change are not affected.

Grids are not required. If you set the wire grid spacing to zero, you can edit and route the wires in gridless mode. If you set the via grid value to zero, you can position vias without using a grid.

You can set a global wire grid for the design or separate wire grids for individual routing layers. You can also set a major grid for the wire grid and control how the grids are displayed. Grids set for individual layers override the PCB wire grid.

To set the PCB wire grid

1. Click *Define – Design Grids*

   The Design Grids dialog box opens.

2. On the *Wire* tab, enter grid spacing and grid offset values on the Routing Wire Grid panel.

   To set the grid spacing:

   ❑ Enter the X direction grid spacing value in the X Grid data entry box.

   ❑ Enter the Y direction grid spacing value in the Y Grid data entry box.

   A value of -1 means the grid spacing in that direction is undefined.

   If you want to set optional grid offsets, you can:

   ❑ Enter an X direction offset value in the X Offset data entry box.

- ❑ Enter a Y direction offset value in the Y Offset data entry box.

  A value of 0 means no offset for that direction is defined.

3. Click *Apply* or *OK*.

You can also set wire grids for individual routing layers by using the `grid wire` command. Wire grids set for individual layers override the global wire grid. See Setting Wire and Via Grids on page 152 for details.

You can display major grid points on the wire grids. The major grid spacing is a multiple of the wire grid spacing. Major grid points can be displayed in a different color from the wire grid color. You can also control the visibility of the wire and major grids, and control whether the grids e display as lines or dots.

To set the major grid and control how wire grids are displayed

1. Choose *View – Display Grids*

   The Display Grids dialog box opens.

2. On the Routing Major Grid panel

   - ❑ Enter a value in the Factor data entry box if you want to display major grid marks on the wire grid. This value is the number of wire grid points between each major grid point.
   - ❑ Enable or disable Grid to control whether the wire grids are visible.
   - ❑ Enable or disable Factor to control whether the major grid marks are visible on the wire grids.
   - ❑ Click the As popup menu and choose either Lines or Dots to display the wire grids as lines or dot. The default is Lines.

3. Click *Apply* or *OK*.

You can set a global via grid for the design or separate via grids for individual vias. Grids set for individual vias override the PCB via grid.

To set PCB via grids

1. Choose *Define – Design Grids*

   The Design Grids dialog box opens.

2. Click the *Via* tab.

3. Enter grid spacing and grid offset values on the Grid For All Vias panel.

   To set the grid spacing:

   ❑ Enter the X direction grid spacing value in the X Grid data entry box.

   ❑ Enter the Y direction grid spacing value in the Y Grid data entry box.

   A value of -1 means the grid spacing in that direction is undefined.

   If you want to set optional grid offsets, you can:

   ❑ Enter an X direction offset value in the X Offset data entry box.

   ❑ Enter a Y direction offset value in the Y Offset data entry box.

   A value of 0 means no offset for that direction is defined.

4. Click *Apply* or *OK*.

To set grids for individual vias

1. Choose *Define – Design Grids*

   The Design Grids dialog box opens.

2. Click the *Via* tab.

3. Enter grid spacing and grid offset values in the table on the Grid For Via Name panel. For each via that you want to define a via grid, find the via ID in the Via column.

   In the same row, you can:

   ❑ Enter the X direction grid spacing value in the X Grid data entry box.

   ❑ Enter the Y direction grid spacing value in the Y Grid data entry box.

   ❑ Enter an X direction offset value in the X Offset data entry box.

   ❑ Enter a Y direction offset value in the Y Offset data entry box.

   A grid spacing value of -1 means the grid spacing in that direction is undefined. A grid offset value of 0 means no offset for that direction is defined.

4. Click *Apply* or *OK*.

Note: Make sure you enter grid values that are scaled for the units you are using.

Setting PCB Wire Width and Clearance Rules

Use the Interactive Routing Setup dialog box to set wire width and all clearance rules at the PCB (global) level.

PCB rules apply to the entire design and have the lowest precedence in the rule hierarchy. PCB rules are overridden by rules at all other levels of the hierarchy.

To set PCB wire width and clearance rules

1. Enter a value in the PCB Wire Width data entry box.
2. Enter a value in the PCB Clearance data entry box.

Setting the Primary and Secondary Routing Layers

Before you begin interactive routing, click the Layers icon in the tool bar to display the Layers panel, and click the pencil button area of the layers you want to enable for routing. Each pencil button toggles between:

- Primary layer, which is indicated by a bold pencil icon.
- Secondary layer, which is indicated by a dimmed pencil icon.
- Disabled layer, which is indicated by no icon.

The primary routing layer is the active routing layer. Only one layer can be the primary layer, but you can change the primary layer during the session.

The secondary layers are alternate routing layers. The interactive router should be able to reach a secondary layer by adding a via, but you can specify any signal layer as a secondary layer. The interactive router does not check for an available via until you attempt to switch layers while routing a wire.

The primary layer name appears in the status bar. The layer name displays in the layer's color.

To choose primary and secondary routing layers

1. Click the Layers icon on the tool bar

   The Layers panel opens.

2. Click one or more times the pencil button area of a layer you want to enable as a primary or secondary layer, or disable for routing.

3. Repeat step 2, above, for each layer you want to enable or disable.

To enable or disable all secondary routing layers

➤ Click the *All Signal Layers* pencil button area once or twice to enable or disable all secondary layers.

To change the primary layer designation

1. Click the *layer pencil* button areas

   When a click on a *pencil* button area changes one of the secondary or disabled layers to the primary layer, its pencil icon becomes bold. The pencil icon of the layer that was primary automatically dims to indicate secondary layer status.

2. Press the [Spacebar] one or more times with the pointer in the design window.

   Pressing the [Spacebar] lets you cycle the primary layer choice through the enabled layers from top to bottom. Be sure the pointer is in the design window.

During interactive routing, the router switches primary and secondary layers, depending on the actions you take. To learn how to control layer switching, refer to Controlling the Routing Layers.

Using Rulers

Use graphical rulers to achieve accurate positioning of wires, vias, and components. Alignment guides help you accurately position the rulers. You can draw horizontal and vertical rulers anywhere in the design where precision routing or placement is needed. You can also draw diagonal rulers if Pointer Snap is set to either 45 Degrees or All in the Style tab of the Interactive Routing Setup dialog box.

Usually, you zoom in and out when doing precision operations. The rulers automatically scale as you zoom. All rulers display until you remove them.

To draw graphical rulers

1. Choose *Define – Ruler – Draw Mode*.

   [LB] is set to the edit ruler mode.

2. Move the pointer to the design location where you want to start drawing the ruler.

Alignment marks appear as the pointer crosses edge lines and centerlines of wires, vias, and through-pins, and edge lines of SMD pads.

3. Click at the starting location.
4. Move the pointer horizontally or vertically.

   A horizontal or vertical line draws.
5. Click at an appropriate ending location.

   A graphical ruler with major and minor tick marks appears between the two end points.

   Click twice and you can start another ruler.

To remove graphical rulers

➤ Choose *Define – Ruler – Forget All.*

This remove graphical rulers from the displayed design.

Note: Every fifth tick mark is labeled with a distance number, if the zoom scale of the displayed design provides sufficient space to write the number.

Choosing an Interactive Routing Mode

The left mouse button ([LB]) performs different editing and routing operations, depending on the interactive routing mode that is active.

You set the interactive routing mode from one of the [RB] popup menus (Interactive Routing, Polygon Editing, or Topology Editing). Some modes can also be set from the routing tool bar. The current mode displays below the command line, beside the mouse icon in the mode status area.

The interactive routing modes are described in the following table.

Table 5-5 The Interactive Routing Modes

Interactive Routing Mode	Description
Add/Edit Polygon mode	Creates new wiring polygons, and edits edges of existing wiring polygons and keepout areas.

Interactive Routing Mode	Description
Change Connectivity mode	Assigns floating net wires and unassigned wiring polygons to nets. Can also change the net assignment of unattached wires and wiring polygons.
Change Layer mode	Changes the layer assignments of wire segments, wires, and wiring polygons.
Change Via mode	Changes properties, such as via type, fanout attribute, and test point attribute of existing vias.
Change Wire Width mode	Changes the widths of existing wires without resetting a rule and rerouting.
Copy Polygon mode	Copies existing wiring polygons and keepout areas to other locations on the same layer.
Copy Route mode	Copies existing wires to an unrouted connection if the connection has a similar length and path. All nets, including power and ground nets, can be copies.
Critic Route mode	Removes extra wire bends.
Cut Polygon mode	Creates rectangular voids in polygons (wiring polygons and keepout areas) or divides a polygon into multiple polygons. Cutouts can overlap, creating voids with complex shapes.
Cut Segment mode	Breaks individual wire segments into two segments, and allows pseudopin insertion at cut points.
Delete Net mode	Removes all wires and vias on the nets you choose, but does not remove wiring polygons. (Nets are not deleted from the design.)
Delete Segment mode	Removes wire segments and the attached vias.
Delete Wire mode	Removes wire segments and all attached segments and vias.
Delete Wiring Polygon mode	Removes wiring polygons (but not keepout areas).
Edit Route mode	Creates new wire paths, changes active wire segment widths, adds vias, and reroutes existing wires. Single or multiple wire paths can be routed.
Merge Wiring Polygon mode	Merges wiring polygons on the same layer with the same net assignments.

Interactive Routing Mode	Description
Move mode	Moves wire segments, vias, polygons (wiring polygons and keepout areas), and polygon edges. Can also merge overlapped wiring polygons or overlapped keepout areas on the same layer.
Repair Net mode	Removes all wires and vias that violate fromto order rules on the nets you choose.
Rotate Via mode	Rotates rectangular vias in 90 degree increments. (Available only in the MicroVia option, which is enabled by Using Set Microvia and requires a special license. See Understanding Licensing for details.)
Topology Editing modes	Various modes used to assign pin attributes to nets; add, move, and delete virtual pins; reorder pin-to-pin connections; fix and unfix pins; set fromto rules; forget fromto rules; forget net rules, and forget connection ordering for nets.

Routing and Editing Wires

You use Edit Route mode to route new wires or to reroute or replace existing wires. You can route either a single wire or multiple wires between connectivity objects terminal, routed wire, or routing guide with net assignments on the same net. You can also route a single wire as a floating net between two points in the design where no connectivity objects are located.

While you are routing, a preview wire of the active wire segment (the wire segment on which you are currently operating) stretches from the pointer to the last point you digitized to show the intended path before you actually digitize the segment. An envelope outlined by a dashed line surrounds the preview wire to indicate the wire-to-wire clearance rule. Small arrows and alignment marks appear when the pointer aligns with a nearby wire, pin, or via that is on the net you are routing.

You can use the Interactive Routing Setup dialog box to set certain routing controls. While routing, you can insert vias to change the routing layer. You can also use commands on the Edit Route popup menu to choose a layer, a via type or both, to change the active wire segment width, to undo the previous action, and to finish, interrupt, or cancel the current wire.

Routing Single Wires

To route a single wire between two objects with net connectivity, begin by clicking one of the objects. The interactive router gathers information about the object, sets up and initializes the route, and identifies the coordinates of the first point on the wire. For an explanation of how to route a single wire, see Routing Individual Wires on page 252.

For more information about routing a wire between two points without connectivity, see the procedures for *Edit Route Mode* in the *Allegro PCB Router Command Reference.*

If you want to route a single wire of a bundle or a differential pair, you must disable the *Enable Bus Routing Mode* option in the Bus tab of the Interactive Routing Setup dialog box.

As you move the pointer, the interactive router displays the preview wire. You can click at a location to end one wire segment and begin a new segment (for instance if you want to change the routing direction or the wire width).

You can change the width of the active wire segment, using *Edit Route Mode*.

If you change layers, for instance to avoid obstacles or to connect to an object on a different layer, the interactive router inserts a via and changes the color of the new wire segment to the color of the exit layer. For more details, see Using Vias During Interactive Routing on page 259.

The interactive router continues to display the preview wire until you either complete the wire or cancel it.

You can also use the single wire routing procedures to complete wires that you previously interrupted or to reroute completed wires.

If you have defined regions with width or clearance rules, you can control whether the interactive router adheres to these rules.

You can set optional controls that affect how the interactive router routes and edits wires. For details, see Setting the Routing Environment on page 148.

Routing Multiple Wires

You can interactively route arbitrary sets of wires, buses, differential pairs, and bundles. Differential pairs and bundles retain their association after routing is completed, but arbitrary wire sets do not.

Interactive routing of differential pairs is supported with the following rules and operations:

- Primary pair gap

- Line width
- Flow/bubble around objects or obstructions
- Interactive definition of the gather point
- The ability to change the lead or following net
- Tandem pair routing

Important

The following new rules are not automatically checked or adhered to during interactive routing of differential pairs.

- ❑ Neck gap
- ❑ Neck width
- ❑ Minimum line spacing
- ❑ Phase tolerance
- ❑ Uncoupled length

Note: In order to see the new rule conflicts created through the interactive routing differential pairs, you must run the `check` command after completing the net.

To route multiple wires between objects with net connectivity, you begin by choosing multiple (unrelated) objects of connectivity, one or more objects on nets defined as a bundle, or an object on a net defined as part of a differential pair.

In general, only those connections that use the same layers and vias can be routed together. The exception is tandem layer pairs, which are nets that must be routed on separate layers and that do not use vias during the pair routing phase. Vias can be used in the finish route phase.

Most of the operations described above for routing single wires also apply to routing multiple wires. In addition, there are several operations that apply only while routing multiple wires. For instance, you can use commands in the Edit Route popup menu to gather bus wires, cycle the control wire, pick up dropped wires, rotate the bus cursor, set the via pattern for changing layers, and control whether the interactive router completes as many wires as possible or only completes the wires if all routes succeed.

You can also set multiple wire routing controls on the Bus tab in the Interactive Routing Setup dialog box. For details, see Setting the Routing Environment on page 148.

Routing Individual Wires

You use Edit Route mode to route an individual wire (including a wire of a multiple wire set) or to edit (reroute) an existing wire. You can route a wire between two objects that have net assignments (connectivity) on the same net. Connectivity objects include terminals, routed wires, and guides.

Except for bundles and differential pairs, you can route individual wires of a multiple wire set by clicking a single pin, a via, or a wire of the set. You do not need to disable Enable Bus Routing option in the Bus tab of the Interactive Routing Setup dialog box, but you can disable Bus Routing if you want to avoid any possibility of picking multiple connections.

Before routing a single wire of a bundle or a differential pair (defined in the Design File or during the session), you must either disable Enable Bus Routing or forget the bundle or pair definition.

You can also route floating nets between two points in blank areas of the design where no connectivity objects are located.

You can use the Interactive Routing Setup dialog box to set optional controls that affect how the interactive router routes and edits wires.

To route a wire

1. Click the Edit Route icon, or press [RB] and choose *Edit Route Mode*.
2. Click a pin, a via, a guide, or a wire.

   If you start routing from a through-pin or via, the layer marked as primary in the Layers panel is used. If you start from an SMD pad or a wire, the primary layer (routing layer) is automatically set to the layer of the SMD pad or wire.
3. Move the pointer to the next location and click to add a wire segment.

   This "digitizes" the next location and creates a wire segment connecting the previous and current digitized locations.

   As you move the pointer, a preview wire stretches from the last point you digitized to show the intended path before you actually digitize a segment. An envelope outlined by a dashed line surrounds the preview wire to indicate the wire-to-wire clearance rule. Small arrows and alignment marks appear when the pointer aligns with a nearby wire, pin, or via that is on the net you are routing.

   When Push Routing is enabled (the default) in the General tab of the Interactive Routing Setup dialog box, the preview wire jogs around shapes that are not part of the same net, as shown below.

If you need to switch layers to avoid an obstruction or connect to a pin on a different layer, click at the location where you want to insert the via and do one of the following:

- ❑ Click again at the same location.
- ❑ Press the [Spacebar].
- ❑ Press [RB], choose Add Via from the Edit Route menu, and choose a via and a layer in the Switch to Layer dialog box.

4. Repeat step 3 as needed to add additional segments to the wire.
5. To complete the connection, do one of the following:
   - ❑ Click on a pin that is on the net you are routing.
   - ❑ Press [RB] and choose Finish Route from the Edit Route menu.

When you complete a connection between two pins, the pointer disconnects from the wire. You remain in Edit Route mode ready to route another connection.

Setting the Routing Style

You can use the Routing Style controls in the Style tab of the Interactive Routing Setup dialog box (press [RB] and choose *Setup* from the Interactive Routing menu) to control how wires are interactively routed. By default, only one wire segment is routed between two digitized point.

- ❑ Enable the Two Segment option if you want to route two wire segments when you move the pointer in two directions (up, down, left, right) from the previously digitized point.
- ❑ Enable the Route to Cursor option if you want the interactive router to route the optimal path between two digitized points.

You can press [RB] and choose Toggle Segment in the Edit Route popup menu to change between the One Segment and Two Segment options.

Routing Shield Nets

If you want to route shield wires when you route shielded nets, enable Auto Shield in the General tab of the Interactive Routing Setup dialog box (the default is disabled). When Auto Shield is enabled, connections that have shield rules are routed with shields. You can also edit existing shields by routing over them. The shield wires generate when you complete a connection or when you end a wire by using Done in the Edit Route menu.

After routing all your shielded wires, enter the `select unrouted_shield_tie_downs` command to select the guides of any unrouted shield tie downs. Then run the autorouter to route the tie downs (make sure you have finished all your interactive editing because you cannot undo it after running the autorouter).

Note: Wire segment endpoints snap to grid points on the defined grid that has the highest precedence. Segments snap to the via grid, if one is defined, or to the wire grid if a via grid is not defined. If neither a wire grid nor a via grid is defined, wire segments snap to the pointer snap grid, if one is defined, or to the manufacturing grid. Grid precedence for wire editing, from highest to lowest, is

- Pointer snap grid
- Via grid
- Wire grid
- Manufacturing grid
- Resolution (defined in the Design File)

For each grid type, a layer grid for the current active layer takes precedence over the global grid.

Often you need to use other editing methods to modify, improve, or complete the connection you are routing. Some frequently used methods are

- Changing incomplete routing paths
- Completing incomplete routing paths
- Rerouting completed connections
- Using vias during interactive routing
- Controlling the Routing Layers

  Use the layer panel to configure the enabled primary and secondary routing layers
- Changing wire segment widths

  Use Edit Route Mode to change active wire segment widths

  Use Change Wire Width Mode to change existing wire segment widths
- Routing floating nets

Operations in other interactive routing modes that can also be used to edit routed wires include

Routing Multiple Wires

You use Edit Route mode to route a set of wires simultaneously.

You can interactively route arbitrary sets of wires, buses, differential pairs, and bundles. The interactive router considers all multiple wire routing as bus routing. Differential pairs and bundles retain their association after routing is completed; but arbitrary wire sets do not.

To route multiple wires, make sure that the Enable Bus Routing option is enabled in the Bus tab of the Interactive Routing Setup dialog box. You can also use this tab to set other multiple wire routing controls.

In general, only connections that use the same layers and vias can be routed together. The exception is tandem pair routing, which requires that the two nets use separate layers and that you do not use vias during the pair routing phase, although vias can be used in the finish route phase. See Edit Route Mode for details.

To route multiple wire sets

1. Press [RB] and choose Setup in the *Interactive Routing Menu*.

   The Interactive Routing Setup dialog box opens.

2. Make sure the following options are set

- ❑ Enable the enable bus routing option
- ❑ Set the required spacing for gathered bus wires.
- ❑ Enable the enable tandem pair routing option, if you want to route tandem layer pairs

3. Enable the enable fit via pattern option only if you want to allow slight changes in via positions when clearance obstructions prevent adding a via pattern.

4. Click *OK*.

5. Click the Edit Route icon, or press [RB] and choose *Edit Route Mode*.

6. Drag over adjacent component pins or guides, or [Shift]-click non-adjacent pins or guides to start routing the multiple wire set.

When you [Shift]-click on pins or guides, you add or remove them from the set.

You can also pick the terminals of a set of wires that are incomplete to restart multiple wire routing.

For differential pairs specified in the Design File or defined through *Define – Net Pair*, clicking one pin or guide automatically picks both connections for routing.

Clicking pins or guides on different layers starts tandem pair routing when the tandem pair routing option is enabled.

7. Move the pointer away from the selected pins or terminals and click to add the first set of wire segments.

This attaches the pointer to the control wire, digitizes the first set of locations, and creates the wire segments connecting the pins or terminals to the new locations. You can cycle the control wire between the two outside wires and the center-most wire of the wire set as needed to control direction changes around obstacles. For differential pairs, cycling just switches the control wire from one wire to the other.

If the wires start from pins that are not at minimum spacing, the wires are either gathered to minimum separation or remain at the pin spacing until the first digitized change in routing direction, dependent on whether the `gather bus wires` option is enabled. Tandem layer pairs are always gathered to achieve vertical alignment.

With the Gather Bus Wires enabled, the start bus cursor appears (a faint white line perpendicular to the active wires) and indicates the minimum spacing between the wires. The bus cursor automatically rotates to stay perpendicular to the starting wire segments as you change the routing direction. You can fix the cursor's spatial orientation by using the Rotate Bus Cursor command in the Edit Route menu. After the wires are gathered, the start bus cursor disappears.

8. Move the pointer to the next location and click to add the wire segments.

As you move the mouse and click, multiple segments (one for each member of the wire set) are added. If routing space is limited to fewer than all the wire segments of the set, the wires that cannot be routed are automatically dropped, depending on whether you enabled the Route As Many As Possible or Route Only If All Succeed option in the Edit Route menu.

If you choose the route-as-many-as-possible option, guides are created from the last digitized locations of the dropped wires to pins on the nets, and you can continue routing a subset of wires. However, even with the option set, no new segments are added if there is insufficient space to route at least one wire of the set. If you choose the route-only-if-all-succeed option, no new segments are added if there is insufficient routing space for all the wires of the set.

If wires are dropped as you route a multiple wire set, you can restart the set by using Pickup Dropped Wires on the Edit Route menu. All wires that do not introduce violations at the current routing location are restarted. Guides are drawn from the start locations of the picked-up wires back to their last digitized locations.

9. To complete the multiple wire routing, repeat step 8 as needed to bring the connections near the next set of pins and do one of the following:

❑ Press [RB], choose Done, and then individually connect each segment to the proper pin.

OR

❑ Choose Finish Route from the Edit Route menu to finish the routes automatically. If some routes do not complete, you must individually connect each route to the proper pin.

If a wire of the set touches one of its net target pins during multiple wire routing, the interactive router automatically applies Finish Route.

If Auto Adjust Length is enabled in the General tab of the Interactive Routing Setup dialog box, Finish Route attempts to meet minimum length rules, and highlights the connection if any rule violations occur.

When you complete the connections between sets of pins or terminals, the pointer disconnects from the wires. You remain in Edit Route mode, ready to route another set of wires or a single connection.

Note: You can always interactively route individual wires of a multiple wire set, including differential pairs. See Routing Individual Wires on page 252 for details.

Often you need to use other editing methods to modify, improve, or complete the connection you are routing. The following multiple wire routing operations are available under the *Edit Route* menu:

- Gathering the bus wires
- Cycling the control wire
- Picking up dropped wires
- Rotating and fixing the start bus cursor orientation
- Routing as many wires as possible or only if all succeed
- Setting a via pattern
- Changing a multiple wire set during routing
- Overriding bus spacing
- Restarting bus routing for a set with dropped wires
- Adding individual vias
- Adding multiple vias
- Single wire operations that are often used during multiple wire routing include
- Changing incomplete routing paths
- Completing incomplete routing paths
- Rerouting completed connections
- Routing wires across region boundaries
- Using vias during interactive routing

Operations in other interactive routing modes that can also be used to edit multiple wires include the following:

- Moving wire segments or vias
- Moving the multiple segments of a bus
- Cutting wire segments
- Deleting routing objects

Controlling the Routing Layer

Before you begin routing or at any point during an interactive routing session, you can use the layer panel to configure the enabled primary and secondary routing layers. When you begin routing from an SMD pad or existing wire, the layer on which the pad or wire is located automatically becomes the primary layer. While you are editing a wire, you can change routing layers by double-clicking or pressing the [space bar]. The result of this action depends on which layers you enabled in the layer panel and whether the current terminal point is one of the following:

- A pseudopin
- A via
- A pin

Using Vias During Interactive Routing

Use vias to change layers as you are routing individual or multiple wires.

The following via operations are available:

- Selecting vias for routing
- Adding vias while routing connections
- Adding multiple vias while routing multiple wires
- Using via grids
- Displaying legal via site
- Changing existing vias

Creating and Editing Wiring Polygons

You can create wiring polygons interactively and change the boundaries of wiring polygons created in a session or defined in the Wiring section of the Design file. For more information, see Creating Wiring Polygons on page 261 and Changing Wiring Polygon Boundaries on page 262.

You can also change the boundaries of keepout areas created in a session or defined in the Structure section of the Design file. However, you cannot change the boundaries of polygons defined in the Image section of the Design file.

You can change the net connection of a wiring polygon or assign a net to an unassigned wiring polygon. When you create a polygon, it is assigned a net if it overlaps a pin, via, pseudopin, or another assigned wiring polygon. Otherwise, the polygon is not assigned a net. For more information, see Changing Wiring and Polygon Connectivities on page 280.

Often you must use other operations to modify, improve, or complete wiring polygons. Frequently used operations include the following:

Moving Polygons and Polygon Edges

Deleting wiring polygons

Cutting wire segments to attach wiring polygons to wires

Copying Polygons

Cutting polygons with a rectangle

Merging polygons

Moving and merging polygons by overlapping

Changing wire connectivities

Checking design rules and reporting conflicts

Creating Wiring Polygons

You create wiring polygons by setting [LB] mode to Add/Edit Polygon mode and dragging the pointer or digitizing points with mouse clicks. You create rectangles by dragging.

To create wiring rectangles by dragging the pointer

1. Press [RB] and choose *Add/Edit Polygon Mode* in the Polygon Editing menu.
2. Place the pointer where you want to start the polygon.
3. Drag the pointer diagonally to form a rectangle of the desired size.

   The rectangle's width and height display in the message area.
4. The rectangle automatically completes when you release [LB].

To create wiring polygons by digitizing points

1. Press [RB] and choose *Add/Edit Polygon Mode* in the Polygon Editing menu.
2. Place the pointer where you want to start the polygon and click to digitize the first corner.

   Once you have started digitizing, you can press [RB] and choose *Setup* to set the Pointer Snap angle to 90 degrees, 45 degrees, or All.

   If you make a mistake and click a point incorrectly, press [RB] and choose *Undo*.
3. Move (do not drag) the pointer to a new location and click to digitize the next corner.
4. Continue digitizing corners as needed.
   - ❑ Horizontal, vertical, and 45 degree alignment marks appear when the pointer aligns with an existing edge or corner. If you click, the new corner snaps to the alignment mark location.
   - ❑ Polygon edges cannot cross (intersect) each other. A digitized corner that intersects an edge is disallowed and an error message displays in the Message area and the Output window.
   - ❑ To remove incomplete portions of the polygon and detach the pointer, you can press [RB] and choose *Cancel*.
5. To finish the polygon do one of the following:
   - ❑ Click the initial starting point.
   - ❑ Press [RB] and choose *Finish Polygon*.

Some tings to remember:

- When you finish editing, checking occurs. If there is a violation, the final digitized point is disallowed. If you want to allow this digitized location, first disable checking.
- When you complete or cancel a polygon, you remain in Add/Edit Polygon mode, ready to create other polygons or to change polygon boundaries.
- You create the wiring polygon on the current layer. When it overlaps a pin, via, pseudopin, or another wiring polygon, the polygon is assigned the same net as the overlapped object. You can change a layer assignment by using the Change Layer mode.
- If the polygon overlaps a wire, an error occurs. For information about cutting a wire to insert a pseudopin so that you can connect the polygon to the wire, see Cutting Wire Segments on page 284.
- If the polygon does not overlap other shapes, the polygon is unassigned. You can assign nets to unassigned polygons by using the Change Connectivity mode.
- If a wiring polygon overlaps shapes that belong to different nets, an error occurs. If checking is enabled, the final digitized point is disallowed. If checking is disabled, the polygon is created, the polygon is unassigned, and a violation is indicated. See Checking design rules and reporting conflicts for further details.
- The maximum number of verticies for a polygon is 4,000.

Changing Wiring Polygon Boundaries

You change existing polygon (wiring polygon and keepout area) boundaries by setting [LB] to Add/Edit Polygon mode and digitizing boundary points with mouse clicks. The starting point must be on the edge of a polygon.

Note: The Keepout and Wiring Polygons “*S”* buttons in the Layers panel must be enabled, or you cannot edit the boundaries.

To change the boundary of polygons

1. Press [RB] and choose Add/Edit Polygon Mode in the Polygon Editing menu.
2. Place the pointer over the edge or corner of an existing polygon.

   A closed polygon icon attaches to the pointer to indicate that you can start editing the boundary.

   Once you have started digitizing, you can press [RB] and choose Setup to set the Pointer Snap angle to 90 degrees, 45 degrees, or All.

If you make a mistake and click a point incorrectly, press [RB] and choose *Undo*.

A warning displays if you attempt to edit a polygon of a type that is unselectable.

3. Click to digitize the starting point of your edit.
4. Move (do not drag) the pointer to a new location and click to digitize a corner.

   An open polygon icon attaches to the pointer to indicate that you are editing the boundary.

   - ❑ Horizontal, vertical, and 45 degree alignment marks appear when the pointer aligns with an existing edge or corner. If you click, the new corner snaps to the alignment mark location.
   - ❑ To remove incomplete portions of your edits and detach the pointer, you can press [RB] and choose *Cancel*.
5. To finish editing, click to an edge or corner of the original polygon or keepout area.

   A closed polygon icon attaches to the pointer to indicate you are finishing your editing and aligned with the edge or corner.

   - ❑ When you finish editing, checking occurs. If there is a violation, the final digitized point is disallowed. If you want to allow this digitized location, first disable checking.
   - ❑ When a polygon edit is finished or canceled, you remain in Add/Edit Polygon mode, ready to continue changing polygon boundaries or Creating Wiring Polygons.
   - ❑ The original polygon reconstructs according to the following criteria:
     - ❍ If all the new corners (not including the edge start and end points) are outside the original polygon, the portion of the original boundary between the start and end points is replaced by the new boundary.
     - ❍ If all the new corners (not including the edge start and end points) are inside the original polygon, the original geometry becomes two nested polygons, and the following message is displayed in the message area:
   - ❑ Click inside the polygon you are keeping
   - ❑ You must click on the polygon you want to keep. The other polygon is deleted. You can undo this operation.
     - ❍ If the new corners (not including the edge start and end points) are both outside and inside the original polygon, the portion of the original boundary between your start and end points is replaced by the new boundary.

- ❍ A reconstructed polygon can fail when existing edges are intersected by new segments in a way that does not allow a legal polygon to be formed. The following message is displayed in the message area

  ```
  Failed to reconstruct polygon (all possible polygons are
  illegal)
  ```

- ❍ Press [RB] and choose *Undo* or *Cancel* to digitize new edges.

❑ If you are editing an unassigned wiring polygon, and the new boundary overlaps a pin, via, pseudopin, or another wiring polygon, the edited polygon is assigned the same net as the overlapped object. If the new boundary overlaps a wire, an error occurs. For information about cutting the wire to insert a pseudopin so that you can connect the polygon to the wire, see Cutting Wire Segments on page 284. If the new boundary does not overlap any of these shapes, the polygon is unassigned. You can assign nets to unassigned polygons.

❑ If a wiring polygon overlaps shapes that belong to different nets, an error occurs. If checking is enabled, the final digitized point is disallowed. If checking is disabled, the polygon is created, the polygon is unassigned, and a violation is indicated. See Checking design rules and reporting conflicts for further details.

Cutting Polygons with a Rectangle

Use the Cut Polygon mode to cut pieces out of polygons (wiring polygons and keepout areas) or to divide a polygon into two or more separate polygons. You can cut polygons with any angle boundaries, but cutouts must be rectangular.

To cut out a section of a polygon, you draw a rectangle that overlaps some portion of the polygon. The overlapped area is removed from the polygon. You can also cut pieces out of several adjacent polygons in a single cut operation. You can cut several pieces from a polygon by using different cutting rectangles.

When you cut a net-assigned wiring polygon, the connectivity is recalculated. When you cut a top level keepout area defined in the Design file, the cut keepout area is written into the Session file.

To divide a polygon into two or more separate polygons, you draw a rectangle that passes all the way through the polygon, crossing the boundary on two or more edges. The overlapped area is removed from the polygon and the boundaries of the separated pieces are closed to form new polygons.

When you divide a net-assigned wiring polygon, the resulting wiring polygons are assigned to the same net. If the wiring polygon you divide is unassigned, each resulting wiring polygon is assigned the special net ID (sp_net*).

When you divide a keepout area, each resulting keepout area is assigned a unique keepout ID. When you divide a top level keepout area defined in the Design file, the new keepout areas are written into the Session file.

To cut polygons

1. Press [RB] and choose *Cut Polygon Mode* in the *Polygon Editing* menu.
2. Drag the pointer to draw a rectangle over the portion of one or more polygons you want to cut, or through a polygon you want to divide, and release the [LB].

   When you release the [LB], the rectangle-polygon overlap area is removed from a single polygon or overlap areas are removed from multiple polygons.
3. Repeat step 2 if you want to make additional polygon cuts.

Attaching Wiring Polygons to Wires

Use the Cut Segment mode to cut a wire segment and insert a pseudopin at the cut. You can then attach a wiring polygon to the wire segment at the pseudopin.

To insert a pseudopin to attach wiring polygons to wires

1. Click the Cut Segment icon or press [RB] and choose *Cut Segment Mode*.
2. Press [RB] and make sure *Insert Pseudopin At Cut* is enabled.
3. Click the wire segment where you want to make the cut and attach the wiring polygon.

   A pseudopin inserts at the cut.
4. Press [RB] and choose *Cancel* to return to Edit Route mode.
5. Press [RB] and choose *Add/Edit Polygon Mode*.
6. To attach a wiring polygon to the wire segment at the pseudopin, do one of the following:
   - ❑ Draw a new wiring polygon that overlaps the pseudopin.
   - ❑ Change the boundary of an existing wiring polygon to overlap the pseudopin.

   The attached wiring polygon is assigned to the net of the wire segment with the pseudopin.

- ❑ With checking disabled, you can also move an existing polygon or one of its edges over a pseudopin, enable checking, then enter the `check` command to attach the polygon and assign the wire segment's net.
- ❑ Use the Cut Segment mode to remove redundant pseudopins by pressing [RB] and choosing *Delete Redundant Pseudopins*. This is the same as clicking *Edit – Delete Redundant Pseudopins*.

Merging Polygons

You can merge individual wiring polygons or individual keepout areas. See the following topics for details.

Merging Polygons on page 266

Merging keepout areas in Merge Keepout mode

Moving and merging overlapping polygons in Move mode

Moving and Copying Routing Objects

You can use Move mode to interactively move objects by either of two methods.

You can:

- slide objects on a layer and push movable obstacles out of the way.
- move objects over obstacles to other locations on a layer.

When rule checking is enabled (the default), there must be enough space at the new location to move the objects without violating current routing rules.

You can move the following objects.

- Wire segments and vias
- Wire corners
- Components
- Polygons (wiring polygons and keepout areas)
- Polygon edges

Moving Methods

Polygons always move over obstacles. You can move single polygons, groups of polygons, or polygons within rectangular areas. You can also rotate and mirror the polygon(s) incrementally and set defaults for automatic rotation, mirroring or both rotation (first) and mirroring.

For other objects, the way they move and the way you pick them depends on whether Edit Slide is enabled in the Move/Copy tab of the Interactive Routing Setup dialog box.

When Edit Slide is enabled (the default):

- wire segments, vias, and polygon edges slide orthogonally on the layer.
- the interactive router shoves aside movable (unprotected and unfixed) wires in the path of the object(s) you are moving when Push Routing is enabled on in the General tab of the Interactive Routing Setup dialog box. You can use Allow Jogs in the Move menu or the in the General tab of the Interactive Routing Setup dialog box to control whether jogs are permitted in pushed wires. By default, Push Routing is enabled and Allow Jogs is set to Orthogonal.

When Edit Slide is disabled:

- Wire segments and vias move over obstacles (instead of pushing them out of the way), the same way polygons move. You can also move components. You can move single objects, groups of objects, or objects within rectangular areas, and you can move groups of objects that include any combination of wire segments, vias, components, and polygons.

You can also rotate and mirror the object(s) incrementally, and you can set defaults for automatic rotation, mirroring or both rotation (first) and mirroring. However, when you move components, or a group of objects that includes components, you can rotate them but not mirror them.

Picking Methods

When Edit Slide is enabled, the way you choose the object(s) you want to move depends on the type of object. You cannot choose more than one type of object at the same time. The tool uses the following priorities (in descending order) to determine which object(s) to pick.

- Vias
- Wire segments or wire corners
- Polygon edges
- Polygons

If you try to choose objects in a group, only the objects with the highest priority are picked. You cannot choose components when Edit Slide is enabled.

To disable Edit Slide, you can

1. Press [Shift] and click on a polygon before picking any other object.
2. Press [RB] and choose *Setup*

   The Interactive Setup dialog box opens,
3. Click the Move/Copy tab, disable *Edit Slide*, and click *OK*.

   You can also disable Edit Slide temporarily for the current move operation by pressing [Shift] and clicking on a polygon before picking any other object.

Note: When Edit Slide is disabled, there is no picking priority and you can move any combination of objects (except polygon edges) together in a group. You cannot move polygon edges when Edit Slide is disabled.

Objects

The following objects can be moved:

- Individual wire segments or segments and vias of a wire that lie in a straight line when Edit Slide is enabled. See Moving Wire Segments and Vias on page 269 for details.

  Note: You can move a single wire segment or via (including an individual segment or via of a multiple wire set), multiple wires and vias in a straight line (including the individual segments or vias of multiple wire sets), or individual wire corners.

- Multiple segments (of a bundle, bus, or pair) when Edit Slide is enabled. See Moving the multiple segments of a bus in Move mode for details.

  Note: You can move the multiple segments of a bus as a unit.

- Polygons and polygon edges. See Moving Polygons and Polygon Edges in Move mode for details.

  Note: You can move, rotate, and mirror single polygons, groups of polygons, and (when Edit Slide is enabled) single polygon edges.

- Components when Edit Slide is disabled. See Moving components in Move mode for details.

  Note: When Edit Slide is disabled, you can move and rotate a single component or groups of components.

- Any combination of routing objects (wire segments, vias, components, and polygons) when Edit Slide is disabled. See Moving Objects Over Obstacles on page 275 for details.

  You can move, rotate, and mirror a single object or groups of objects, or you can move and rotate a component or groups of objects that include one or more components.

Note: You can also use Move mode to overlap and merge wiring polygons or keepout areas on the same layer. Wiring polygons must belong to the same net in order to merge them, and keepout areas must be the same type and have the same rules assigned. See Moving and merging polygons by overlapping in Move mode for details.

Moving Wire Segments and Vias

Use Move mode to move individual wire segments and vias, including the individual segments and individual vias of multiple wire sets.

You can move the multiple segments of a bus as well as all the segments and vias of a wire that lie in a straight line as individual units. When you move wire segments or vias, other segments connected to the object you are moving stretch and adjust as needed. You can also move wire corners diagonally to add or eliminate a chamfer.

Clearance and wire width rules are maintained for all wires involved in a move operation. As you move wire segments or vias, wires in the path of the object you are moving can be pushed aside if Push Routing is enabled in the General tab of the Interactive Routing Setup dialog box. You can also control whether orthogonal or 45 degree diagonal jogs are permitted in pushed wires.

If obstacles prevent you from pushing wires aside, use Cut Segment mode to cut the blocked wires. See Cutting Wire Segments on page 284 for details.

You can also move wire segments and vias so that they jump over obstacles instead of pushing them if you disable Edit Slide in the Move/Copy tab of the Interactive Routing Setup dialog box. When you disable Edit Slide, you can move groups of objects that include any combination of wire segments, vias, components, and polygons using the procedures described in Moving Polygons and Polygon Edges.

To move a wire segment or via

1. Click the Move icon, or press [RB] and choose *Move Mode*.
2. Click the wire segment or the via you want to move.

   The wire segment or via attaches to the pointer.
3. Move the pointer to the location you want.

   As you move the pointer, the attached object moves and all connected wire segments stretch to follow.
4. Click to place the wire segment or via at the location you want.

Note: After you move and place the wire segment or via, you are ready to move another object.

To move several segments and vias of a wire that lie in a straight line

1. Click the Move icon, or press [RB] and choose *Move Mode*.
2. Choose the wire segments and vias you want to move by doing the following
   - ❑ Double-click on a wire segment to include vias attached to the wire segment ends.
   - ❑ Click on a wire segment or via and press [Shift] while you click another via or wire segment to include all wire segments and vias between them.

   The wire segments or vias attach to the pointer.
3. Move the pointer to the location you want.

   As you move the pointer, the attached objects move and all connected wire segments stretch to follow.
4. Click to place the wire segments or vias at the location you want.

Note: After you move and place the wire segments or vias, you are ready to move other objects.

To add a chamfer

1. Click the Move icon, or press [RB] and choose *Move Mode*.
2. Click on the wire corner where you want to add chamfer.
3. Drag the mouse diagonally toward the inside of the corner.
4. Release [LB] to set the chamfer at the size you want.

To remove a chamfer

➤ Click the chamfered segment and drag toward the outside until a corner appears. Then release [LB].

Moving Polygons and Polygon Edges

Use Move mode to move polygons (wiring polygons and keepout areas) or edges of polygons. See the Moving polygons and Moving polygon edges procedures in *Move Mode*.

You can move single polygons or groups of polygons. When moving polygons, they jump over obstacles. You can include wire segments, vias, or components when moving a group of polygons if you disable Edit Slide in the Move/Copy tab of the Interactive Routing Setup dialog box.

You can move single polygon edges when Edit Slide is enabled. Clearance rules are maintained when rule checking is enabled. Wires are pushed out of the way when Push Routing is enabled in the General tab of the Interactive Routing Setup dialog box. If Edit Slide is disabled, polygon edges cannot be moved.

Depending on the shape of the polygon, some edges might not be movable. The pointer changes to an open circle (UNIX) or a four-sided arrow (Windows) when you move the pointer over polygon edges. If the pointer doesn't change, try zooming in to enlarge the polygon. Also make sure wiring polygons and keepouts are selectable in the Layers panel.

When a wiring polygon has one or more vias attached to connecting wires, you can move the polygon, or its edge, so that the polygon no longer overlaps the via(s). When this occurs, the vias and wires remain in place. Guides are used to connect the wires to the wiring polygons.

You can move a net-assigned wiring polygon or its edges to overlap any wire segment belonging to the same net. In this case, same net wires are not pushed.

You can assign nets to unassigned wiring polygons by moving wiring polygons or edges of wiring polygons so that they overlap a pin, via, pseudopin, or a wiring polygon belonging to a net. You must use the following rule checking methodology:

To assign nets to unassigned wiring polygons during a move operation

1. Disable checking by clicking the Checking box in the status bar below the work area.
2. Move the unassigned polygon or an edge of the polygon so that it overlaps a pin, via, pseudopin, or assigned polygon.
3. Run the `check` or `check area` command.

   If there are no violations, the tool assigns the same net as the overlapped object to the polygon during checking.

For more information about checking, see *Rules – Check Rules*.

Moving Components

You can use Move mode to interactively move and rotate components.

When Edit Slide is disabled in the Move/Copy tab of the Interactive Routing Setup dialog box, you can move components over obstacles to any part of the same layer. You can move single components, groups of components, or components within rectangular areas. When rule checking is enabled (the default), there must be enough space at the new location to move the components without violating current routing or placement rules.

To move single components

1. Set [LB] to Move mode by doing one of the following:

    ❑ Press [RB] and choose *Move Mode* in the Interactive Routing menu or the Polygon Editing menu.

    ❑ Click the Move icon on the tool bar.

2. To disable Edit Slide if its currently enabled, press [RB] and choose *Setup* to open the Interactive Routing Setup dialog box, click the Move/Copy tab, click to disable Edit Slide, and click *Apply* or *OK*.

3. Click within the component to choose the component you want to move and to identify the move reference point.

    This attaches a ghost image of the component to the pointer.

    As you move the pointer, the attached object moves with it. Wires remain in place and guides are used to connect the wires to the component.

4. Move the pointer to the desired location, and click to move the component.

    Attached wires are not extended to the component's new location, guides attach from the end points to the component.

5. Repeat steps 3 and 4 to move additional components.

To move groups of components when Edit Slide is off

1. Set [LB] to *Move Mode* by doing one of the following:

   ❑ Press [RB] and choose Move Mode in the Interactive Routing menu or the Polygon Editing menu.

   ❑ Click the Move icon on the tool bar.

2. To disable Edit Slide if its currently enabled, press [RB] and choose *Setup* to open the Interactive Routing Setup dialog box, click the Move/Copy tab, click to disable Edit Slide, and click *Apply* or *OK*.

3. Choose the components you want to move. (You can also choose wire segments, vias, and polygons with the components.) You can

   ❑ Drag the pointer to draw a rectangle around one or more components. Components do not need to be completely enclosed by the rectangle.

   ❑ Press [Shift] and click one or more components, drag to draw a rectangle around components in one or more rectangular areas, or both. Using [Shift] allows you to toggle the picking status of the components.

   Note: When you select a component to move, it will return to the default rotation that you defined.

   The tool highlights the chosen components. While pressing [Shift], you can also click or drag the pointer around highlighted components to unselect them. If you drag the pointer around both highlighted and unhighlighted components, the tool chooses the unhighlighted components and de-selects the highlighted components.

4. After choosing the components you want to move, click in the work area to identify the move reference point.

   This attaches a ghost image of the components to the pointer.

   As you move the pointer, the attached objects move with it. Wires remain in place and guides are used to connect the wires to the component.

5. Move the pointer to the desired location, and click to move the components.

   Attached wires are not extended to the component's new location, but guides are attached from the end points to the component.

6. Repeat steps 3, 4, and 5 to move additional groups of components.

Note:

- ❑ You can terminate the current move operation, by pressing [RB] and choosing *Cancel*.
- ❑ You can undo a move operation by choosing Undo in the Interactive Routing menu or the Polygon Editing menu, or by clicking *Edit – Undo*.
- ❑ You can rotate the components you move. See *Move Mode* for information on rotating and mirroring objects.
- ❑ You can also use the interactive placement Move Component mode to move, mirror, and rotate components and wiring polygons. See Moving components for details.

Moving Objects Over Obstacles

You can use Move mode to interactively move any combination of one or more routing objects (wire segments, vias, components, wiring polygons, and keepout areas) by either:

- disabling Edit Slide in the Move/Copy tab of the Interactive Routing Setup dialog box.
- temporarily disabling Edit Slide by pressing [Shift] and clicking on a polygon (wiring polygon or keepout area) before choosing other objects.

You can move single objects, groups of objects, or objects within rectangular areas. You can move objects over obstacles to any part of the same layer. When rule checking is enabled (the default), there must be enough space at the new location to move the objects without violating current routing rules.

You can rotate the objects you move, and you can mirror any object except a component, and any group of objects that does not contain a component. See *Move Mode* for information on rotating and mirroring objects.

To move single objects

1. Set [LB] to Move mode by doing one of the following:
    - ❑ Press [RB] and choose *Move Mode* in the Interactive Routing menu or the Polygon Editing menu.
    - ❑ Click the Move icon on the tool bar.
2. Press [RB] and choose *Setup* to open the Interactive Routing Setup dialog box, click on the Move/Copy tab, make sure that Edit Slide is disabled, and click *OK*.
3. Click within an object to choose the object you want to move and to identify the move reference point.

This attaches a ghost image of the object to the pointer.

4. Move the pointer to the desired location, and click to move the object.
5. Repeat steps 3 and 4 to move additional polygons.

To move groups of objects

1. Set [LB] to Move mode by doing one of the following:
   - ❑ Press [RB] and choose *Move Mode* in the Interactive Routing menu or the Polygon Editing menu.
   - ❑ Click the Move icon on the tool bar.
2. Do one of the following to disable Edit Slide:
   - ❑ Press [RB] and choose *Setup* to open the Interactive Routing Setup dialog box, click on the Move/Copy tab, make sure that Edit Slide is disabled, and click *OK*.
   - ❑ Press [Shift] and click on a polygon, before picking any other object, to temporarily disable Edit Slide for the current move operation.
3. Choose the objects you want to move by doing one of the following:
   - ❑ Drag the pointer to draw a rectangle around one or more objects. Objects do not need to be completely enclosed by the rectangle.
   - ❑ Press [Shift] and click on one or more objects, drag to draw a rectangle around objects in one or more rectangular areas, or both. Using [Shift] enables you to toggle the picking status of the objects.

   The tool highlights the selected objects. While pressing [Shift], you can also click on or drag the pointer around highlighted objects to unselect them. If you drag the pointer around both highlighted and unhighlighted objects, the tool selects the unhighlighted objects and de-selects the highlighted objects.
4. After choosing the objects you want to move, click in the work area to identify the move reference point.

   This attaches a ghost image of the objects to the pointer.
5. Move the pointer to the desired location, and click to move the objects.
6. Repeat steps 2, 3, and 4 to move additional groups of objects.

Note:

- ❑ You can terminate the current move operation, by pressing [RB] and choosing *Cancel*.
- ❑ You can undo a move operation by choosing *Undo* in the Interactive Routing menu or the Polygon Editing menu, or by clicking *Edit – Undo*.
- ❑ You can also use the interactive placement Move Component mode to move, mirror, and rotate components and wiring polygons. See the procedures for *Move Component Mode* in the *Allegro PCB Router Command Reference* for details.

Copying Wires and Vias

Use *Copy Route Mode* to copy existing wires, including power nets, and vias to unrouted connections. Also use Copy Route mode to copy fanouts to component instances of the same component image or to component instances of different images that have the same footprint.

You can copy complete or incomplete connections (but not floating nets), single wires to one or more target pins, and multiple wires using a single target pin. You can mirror the wires in the x-, y-, or xy-planes as you copy them.

For copy route to succeed, the paths of the target unroutes and the wire or wires you want to copy must have a similar topology.

When you copy a single wire to a target pin on a net or fromto with different width or clearance rules, the interactive router applies these rules to the copied wire. If rule checking is enabled, the copy is successful only if it does not cause clearance rule violations.

See the following procedures for *Copy Route Mode* in the *Allegro PCB Router Command Reference* for details.

- Copying a single wire
- Copying multiple wires
- Copying fanout wires and vias

Note: To enable power net copying, choose the Copy Route mode and enable the Copy Power Nets option in the [RB] Copy Route popup menu.

Copying Polygons

You can use Copy Polygon mode to interactively copy any combination of polygons (wiring polygons and keepout areas).

You can copy single polygons or groups of polygons within a rectangular area. When rule checking is enabled (the default), there must be enough space at the new location to copy the polygons without violating current routing rules.

To copy single polygons

1. Press [RB] and choose *Copy Polygon Mode* in the Polygon Editing menu.
2. Click a polygon to choose the polygon you want to copy and to identify the copy reference point.

   This attaches a ghost image of the polygon to the pointer.
3. Move the pointer to the desired location, and click to copy the polygon.
4. Repeat steps 2 and 3 to copy additional polygons.

To copy groups of polygons

1. Press [RB] and choose *Copy Polygon Mode* in the Polygon Editing menu.
2. Choose the polygons you want to copy. You can
   - ❑ Drag the pointer to draw a rectangle around one or more polygons. Polygons do not need to be completely enclosed by the rectangle.
   - ❑ Press [Shift] and click on one or more polygons, drag to draw a rectangle around polygons in one or more rectangular areas, or both.

   This highlights the polygon(s) as you choose them. While pressing [Shift], you can also click or drag the pointer around highlighted polygons to unhighlight them. If you drag the pointer around both highlighted and unhighlighted polygons, the tool highlights the unhighlighted polygons and unhighlights the highlighted polygons.
3. After choosing the polygons you want to copy, click in the work area to identify the copy reference point.

   This action attaches a ghost image of the polygons to the pointer.
4. Move the pointer to the desired location, and click to copy the polygons.
5. Repeat steps 2, 3, and 4 to copy additional groups of polygons.

Note:

- ❑ You can terminate the current copy operation, by pressing [RB] and choosing Cancel.

- ❑ You can undo a copy operation by choosing Undo in the Interactive Routing menu or the Polygon Editing menu, or by clicking *Edit – Undo*.
- ❑ You can rotate and mirror the polygons you copy. See *Move Mode* for information on rotating and mirroring objects.
- ❑ A copied wiring polygon has the same layer assignment as its original, and by default, the same net assignment. You can use the Move/Copy tab of the Interactive Routing Setup dialog box to control whether net assignment is inherited.
- ❑ A copied keepout area has the same layer and type assignment as its original, but is assigned a unique name that is an iteration of the original base name. For example, if you copy a keepout area named `keepout3`, the copy is named `keepout4`.

Changing Wire Segment Widths

You can change the segment widths of existing wires and the active wire.

- Use *Change Wire Width mode* to change existing wire segment widths.
- Use *Edit Route mode* to change active wire segment widths.

Changing Existing Vias

Use the Change Via mode and Change Via Setup dialog box to change independently via type (padstack), fanout attribute, or test point attribute for vias that are part of existing wires. You set or change via type and attributes in the Change Via Setup dialog box. After making your changes, click on existing vias or select the vias in an area by dragging the pointer over the area. The via type and attributes of the selected vias change to the current settings.

To change vias and via attributes

1. Press [RB] and choose *Change Via Mode*.

   The Change Via Setup dialog box opens.

   The first time you press [RB] and choose *Change Via Mode*, the Change Via Setup dialog box displays. Once you are in the mode, you must press [RB] to display the dialog box again.

2. Enable or disable the Change Via Type, Change Fanout Attribute, or Change Testpoint Attribute.

You use the independent check boxes to enable or disable any or all of these change operations. For example, to change only the test point attribute for vias, make sure Change Testpoint Attribute is enabled, and Change Fanout Attribute and Change Via Type are disabled.

3. To enable and set the options, do one or more of the following:

❑ Enable Change Via Type and choose a via type from the Vias List.

❑ Enable Change Fanout Attribute and click Off or On.

❑ Enable Change Testpoint Attribute and click No Testpoint, Front, or Back.

4. Click *OK* or *Apply*.

5. Change existing vias by doing one of the following

❑ Click individual vias.

❑ Drag the pointer to define a rectangular area around vias.

Note:

❑ To continue changing vias using different settings, press [RB] and choose *Setup Change Via* and repeat steps 2, 3, and 4, above.

❑ When you are finished changing vias, press [RB] and choose *Cancel* to return to Edit mode.

❑ When assigning the fanout attribute to a via, the fanout wires do not receive the attribute, so they are not automatically protected.

Changing Wiring and Polygon Connectivities

Use Change Connectivity mode and the Setup Change Connectivity dialog box to change the net assignments of floating nets (unassigned wires and wiring polygons) and of wiring polygons already assigned to a net.

When you create floating nets, they are assigned net IDs composed of the `sp_net` prefix with a unique integer appended. Using Change Connectivity mode, you can assign nets to unassigned wires and unassigned wiring polygons. You can also change the net assignments of assigned wiring polygons that do not overlap a wire or terminal object. You can change these assignments as often as you require.

To change the connectivity of a wire that is connected to a terminal object of an existing net, you must first make it a floating net (disconnect it from the existing net by deleting the segments that are connected to pins or other terminal objects). See the procedures for *Delete Segment Mode* in the *Allegro PCB Router Command Reference* for details.

To change the connectivity of a wiring polygon that is attached to a net, you do not have to make it a floating net. But if the polygon overlaps a wire or a terminal object, you must first either move the polygon or edit the polygon boundary to remove the overlap before you can change the net assignment.

To change the connectivity of wires or wiring polygons

1. Press [RB] and choose *Change Connectivity Mode* in the Interactive Routing menu.

   The [LB] is set to Change Connectivity mode, and the Change Connectivity Setup net list dialog box displays.

2. Choose the net ID from the list box for the net that you want to assign to the floating nets or wiring polygons.

3. Click the wires or wiring polygons, or press [LB] and sweep across the wires or wiring polygons that you want to reassign.

   The wires and wiring polygons are assigned to the net you chose.

4. Press [RB] and choose Setup Change Connectivity in the Change Connectivity menu if you want to change to a different net assignment.

5. Repeat steps 2, 3, and 4 above as many times as needed to complete the wire and wiring polygon reassignments.

Changing Layer Assignments

Use the Change Layer mode to change the layer assignments of wire segments, wires, and wiring polygons. A wire consists of all the segments between any two terminals except vias or tjunctions.

To change layer assignments

1. Press [RB] and choose either *Change – Layer Mode* in the Interactive Routing menu or *Change Layer Mode* in the Polygon Editing menu.

   The Change Layer dialog box opens.

2. To set the layer assignment, enable Wire/Polygon and choose a layer from the Layer list. The layer you choose must be a signal layer.

3. Click *OK* or *Apply*.
4. Choose the objects to move them to the new layer by doing any of the following.
   - ❑ Click individual wire segments or wiring polygons.
   - ❑ Press [Shift] and click individual wires.
   - ❑ Drag the pointer diagonally to choose wire segments and wiring polygons that are within or cross into a rectangular area.
   - ❑ Press [Shift] and drag the pointer diagonally to choose wires and wiring polygons that are within or cross into a rectangular area.

Note:

- ❑ When layer assignments are changed for wires or wire segments, all appropriate vias are added or removed but wire segments are not rerouted. You must press [Shift] to choose entire wires instead of wire segments.
- ❑ If Wire/Polygon is disabled in the Change Layer dialog box, layer assignments are not changed.
- ❑ To continue changing layers using different layer assignments, press [RB] and choose *Setup Change Layer*, and repeat steps 2, 3, and 4.
- ❑ When you are finished changing layers, press [RB] and choose Cancel to return to *Edit Route mode*.

Editing Net Topologies

You can use the topology editing modes to modify the topology of a net. You can modify pin attributes; fix and unfix pins, edit virtual pins, change the net connectivity, set fromto rules, forget fromtos, forget net rules, and forget net order.

You can access the topology editing tools in two ways. You can choose a method for picking the net you want to edit, pick the net, and then use the interactive menu to choose the type of topology editing you want to apply. Instead, you can use the Alternate Topology Editing menu to open the Topology Edit dialog box that remains open for all editing operations.

To use the topology editing mode

1. Press [RB] and choose *Topology Editing Menu*.
2. From the Topology Editing menu choose one of the following:
   - ❑ *Pick Net Mode*

- ❑ *Pick Net By List*
- ❑ *Pick Net By Class*
- ❑ *Pick Net By Bundle*

3. Pick the net you want to edit.
4. Press [RB] and choose the topology edit mode you want to use.

Note: If you choose Alternate Topology Editing from the [RB] menu, the Topology Edit dialog box opens and remains open until you close it.

To use the Topology Edit dialog box

1. Press [RB] and choose *Topology Editing Menu.*
2. Choose Alternate Topology Editing from the [RB] menu.
3. Choose a pick net method.

   If you choose *Pick Net Mode*, move the mouse until the large white X is over any pin of the net you want to edit and click [LB] to pick the net.
4. Choose the topology editing mode you want to use.

To edit the topology of a picked net

After you *pick a net* for editing, the topology editing modes are available. You can:

- Modify pin attributes
- Add virtual pins
- Remove virtual pins
- Move virtual pins
- Reorder net connections by clicking pins
- Reorder net connections by clicking components
- Fix and unfix pins
- Set individual fromto rules
- Forget fromto
- Forget net rules

- Forget net order

Removing Wire Bends

Use the Critic Route mode to remove extra bends and acute angles from wires. *Critic Route Mode* can also improve pad and via entries and exits.

Critic Route applies to all wires that pass through the area you draw with the pointer, or you can choose a single connection by clicking a wire or pin. If you draw an area, Critic Route is applied to all unprotected wires on all layers within the area.

To remove extra bends from wires and improve pad and via entries and exits

1. Click the Critic Route icon, or press [RB] and choose *Critic Route Mode*.
2. Drag the pointer diagonally to remove extra bends and acute angles from wire segments within a rectangular area.

Note: If you enclosed a large number of wires with the pointer, you might get further improvements by doing the operation a second time.

Cutting Wire Segments

Use Cut Segment mode to create new segments in a wire you are routing. Clicking a segment cuts the wire at the location of the mouse. Drawing a cut line (by dragging the pointer) across several segments cuts the segments at the intersection points.

Alignment guides help you accurately position the cut lines. By default, you can draw horizontal and vertical cut lines. You can draw diagonal cut lines if you set the Pointer Snap option to 45 Degrees in the Style tab of the Interactive Routing Setup dialog box. To draw cut lines at any angle, set the Pointer Snap option to All Angle.

The following Cut Segment techniques are often used to route around obstacles.

- Cutting a wire to unblock a pushed wire
- Cutting a wire at two points to move the middle segment

In addition, you can insert a pseudopin at a cut so that you can attach a wiring polygon to the wire.

Deleting Routing Objects

Use the Delete modes to delete wires, portions of wires, and wiring polygons, or to repair nets.

When you choose one of the Delete modes, the pointer changes to a skull and crossbones. You can perform the following operations:

- Delete wire segments
- Delete complete wires
- Delete wires on nets
- Delete wiring polygons
- Repair nets

Repairing Nets

Use the Repair Net mode to delete wire segments that violate fromto order rules. Wire segments that comply with fromto order rules are not affected when you apply Repair Net.

Fromto order violations can result from design revisions. For example, if an engineering change to a finished design alters fromto ordering, you must translate the changes from the layout system, load the design into a new router session, and read the routes or Session File from the previous session. Then use the Repair Net mode to delete the incorrect wiring on the affected nets.

To repair nets

1. Press [RB] and choose *Delete – Repair Net Mode*.

   A warning box displays.

2. Click *OK* in the warning box to continue net repair.
3. Click a pin, via, or wire on the net you want to repair.

Using Undo and Redo

`Undo` cancels the last interactive operation and restores the previous routing and placement state. You can use `undo` successively. Each `undo` command restores a previous design state. `Redo` reverses the effects of `undo`.

All interactive router operations, except Repair Net, can be reversed with `undo`. Certain noninteractive routing commands, such as autorouting, placement, and check commands, can cause `undo` information to be lost. `Undo` is not possible is these cases. However, the `Check Area` command does not erase `undo` information.

`Undo` can be very useful during interactive routing because interactive router operations, such as moving routes, can cause many wire paths to change. `Undo` can automatically restore the wiring state that existed before these complicated path changes occurred. You can issue as many `undo` commands as needed to restore a previous wiring state.

The interactive router Undo and Redo functions are accessed as follows

- ❑ Choose *Undo* and *Redo* under Edit on the menu bar
- ❑ [Undo] function key, if present. To perform Redo, press [Shift] - [Undo]
- ❑ Function key [F3] to Undo and [Shift] - [F3] to Redo

To undo and then redo an interactive routing operation

1. Press [F3] to Undo the last interactive routing operation.
2. Press [Shift] - [F3] to Redo and reverse the undo operation.

Note: The Interactive Routing menu and the Edit Route menu contains the *Undo* command but not the *Redo* command.

Routing Designs with Positive Shapes

Positive shapes in PCB Editor designs are translated to PCB Router as wiring polygons. A wiring polygon in the router is a filled rectangular or polygon-shaped area on a signal layer that acts as a conductor. In some design flows, it is desirable to allow routing and staggering within wiring polygons.

You can route or stagger inside wiring polygons on signal layers in the router given one or more the following conditions:

- parameters for positive shapes in PCB Editor designs are set appropriately to allow routing within wiring polygons before translation to the router.
- you open a router Design file that contains wiring polygons with viapop and wireplow rules enabled in the wiring section of the file.
- you read a Wires file that uses wiring polygons with viapop and wireplow rules enabled into a design.

Automatic Routing and Fanout Behavior

Vias and Wires

The autorouter places vias and wires inside wiring polygons that have their viapop rule enabled. Vias may pass completely through these wiring polygons or stop at the same layer as the polygon.

Note: As a special case, fanout wires are allowed to pass through the wiring polygon with their respective vias when the wireplow rule is disabled.

Important

Vias are placed without regard or concern for creating slivers, isolations, or other manufacturing problems created by the voids of the vias.

Interactive Routing Behavior

Edit route, Move, and Copy Route allow vias and wires to be placed inside wiring polygons that have their viapop and wireplow rules enabled.

Important

Vias and wires are placed anywhere inside a wiring polygon. No concern is given to the possible generation of slivers or isolation. No limits are placed on the number or location of vias generated by Edit route, Move, or Copy Route. There are no controls or parameters available to limit via and wire placement within wiring polygons that have viapop and/or wireplow enabled.

Wiring Polygon Graphics

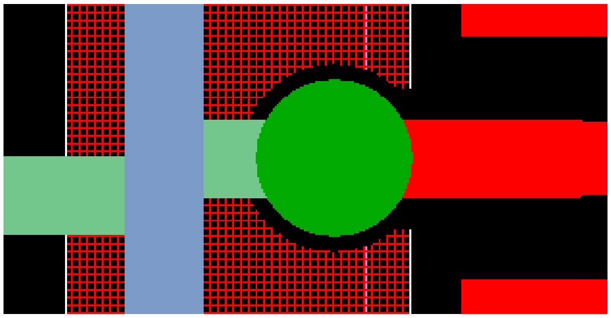
Wiring polygons, with their viapop and wireplow rules enabled, display vias and wires within their boundaries surrounded by voids.

Note: In order to view vias and wires in this way, you must go into the Layers panel by choosing *View – Layers*, and toggling Wiring Polygons off, then back on.

Vias

Vias are displayed as vias surrounded by a void in the background color representing the via to polygon clearance. This void is similar to an antipad displayed on a plane layer. The size of this void is equal to the “rough void” used in the PCB Editor design. An example of this via graphic is shown in the following figure.

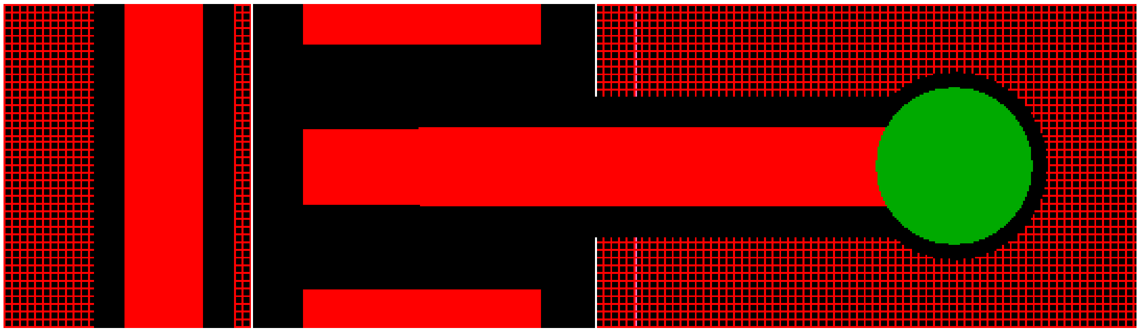
Figure 5-44 Via within Wiring Polygon with viapop enabled



Wires

Wires are displayed as wires surrounded by a void in the background color representing the wire to polygon clearance. The size of this void is equal to the “rough void” used in the PCB Editor design. An example of this wire graphic is shown in the following figure.

Figure 5-45 Wires within Wiring Polygon with wireplow enabled



Error Messages

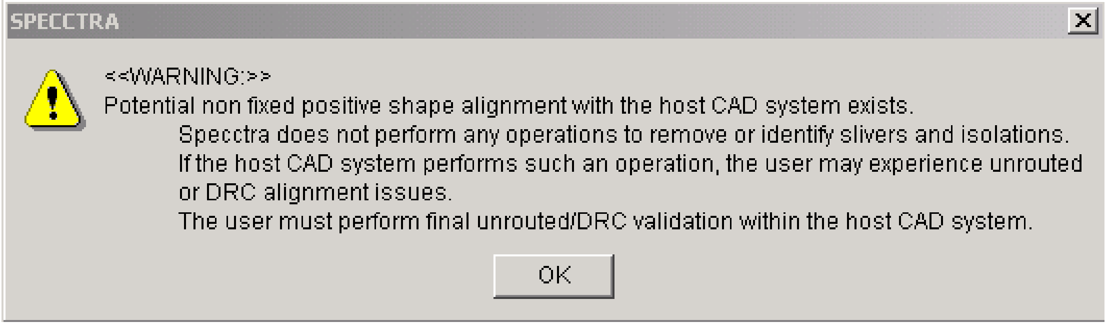
A warning message is generated when potential alignment issues exist between the router and the host cad system for polygons with the viapop or wireplow options. The message appears in either a startup error pop-up or in the message window (depending on the cause). The message also appears at the top of reports generated by *Report – Specify Conflicts – Route* and *Report – Specify – Startup Errors*.

The warning message displayed in the message window reads:

"Potential positive shape alignment with the host cad system exists."

In the case of startup errors, the following message is displayed.

Figure 5-46 Positive Shape Warning Message



<<WARNING:>>
Potential non fixed positive shape alignment with the host CAD system exists.
Specctra does not perform any operations to remove or identify slivers and isolations.
If the host CAD system performs such an operation, the user may experience unrouted or DRC alignment issues.
The user must perform final unrouted/DRC validation within the host CAD system.

These warning messages are displayed under the following conditions:

- Opening a design (with or without a session file) in the router interactive editor which contains polygons having viapop or wireplow on. A warning is displayed in a startup error pop-up and in the Routing Conflicts and Startup Errors reports.

- Opening a design (with or without a session file) in batch mode (no graphics) which contains polygons having viapop or wireplow on. A warning is displayed in a startup error pop-up and in the Routing Conflicts and Startup Errors reports.
- Reading in a Wires file or Routes file which contains polygons having viapop or wireplow on. A message is displayed in the message window and in the Routing Conflicts and Startup Errors reports.
- Writing a Session file which contains polygons having viapop and wireplow on. A warning is displayed in the message window.

6

Working with Advanced Technologies

In this chapter . . .

Understanding Crosstalk and Coupled Noise

Two types of rules control crosstalk and coupled:

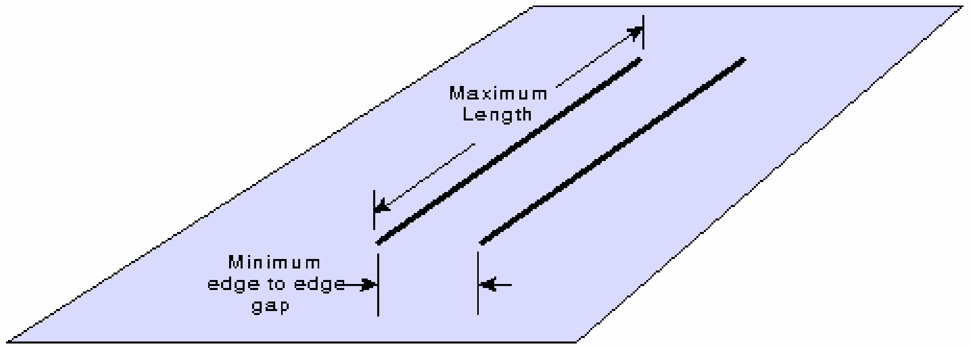
- Parallel and tandem segment
- Parallel and tandem noise

Parallel and tandem segment rules control crosstalk between the individual wire segments of nets. Parallel and tandem noise rules control the total cumulative noise that is coupled across an entire net. You can assign these rules at the layer, class, and net levels. Crosstalk at the fromto level can be controlled with parallel and tandem segment rules only.

Parallel segment crosstalk

The parallel_segment crosstalk rules control parallelism between wire segments on the same layer. You supply gap and length values as shown in the following figure.

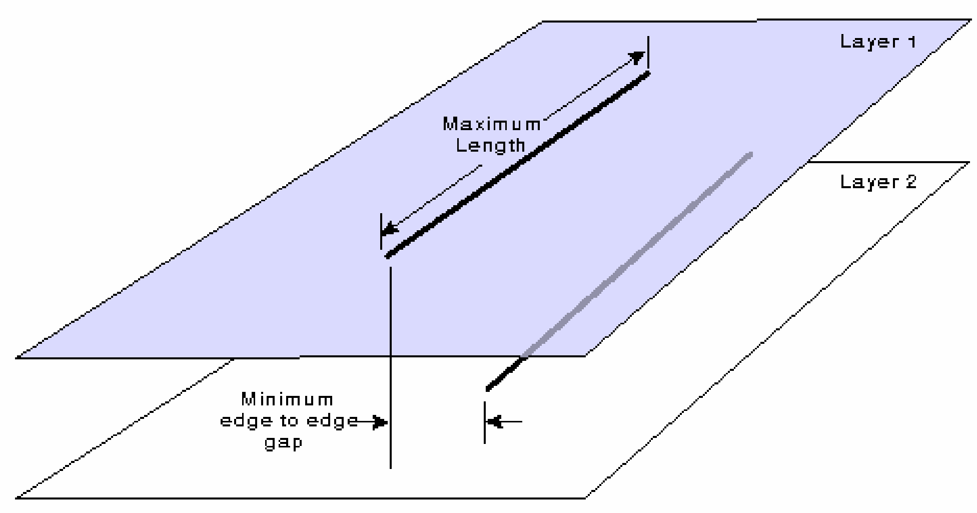
Figure 6-1 Parallel Segment Crosstalk Rules



Tandem segment crosstalk

The tandem_segment crosstalk rules control parallelism between wire segments on adjacent signal layers. There can be no intervening power layers. You supply gap and length values as shown in the following figure.

Figure 6-2 Tandem Segment Crosstalk Rules



See the following topics for more information about parallel and tandem segment crosstalk rules:

- Using Multiple Segment Crosstalk Rules
- Controlling Class-to-Class Crosstalk
- Controlling Cumulative Coupled Noise

Using Multiple Segment Crosstalk Rules

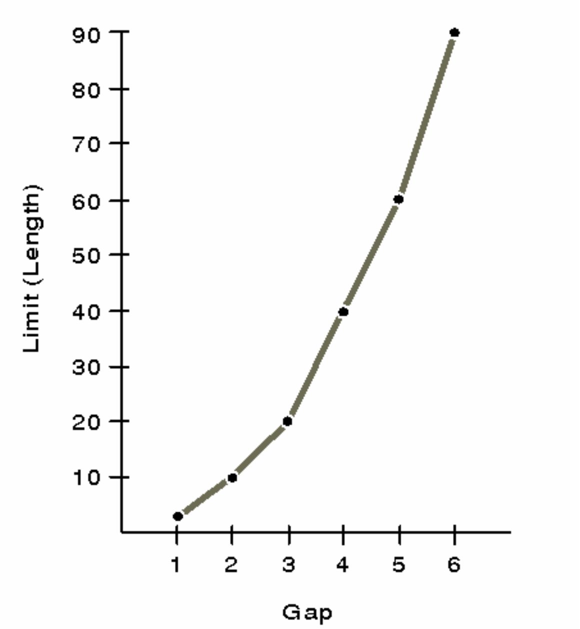
You can set different length limits for different gaps by using multiple rules. Multiple parallel_segment and tandem_segment rules can accommodate how crosstalk varies as a function of parallel length and gap. Parallel rules do not overwrite or replace previous rules but apply collectively in an autorouting session.

The following series of parallel_segment rules approximate a crosstalk characteristic that varies as a function of parallel length and gap. Multiple tandem_segment rules are applied in the same way to control interlayer crosstalk.

```
rule pcb (parallel_segment (limit 25) (gap 10))
rule pcb (parallel_segment (limit 100) (gap 20))
rule pcb (parallel_segment (limit 200) (gap 30))
rule pcb (parallel_segment (limit 400) (gap 40))
rule pcb (parallel_segment (limit 600) (gap 50))
rule pcb (parallel_segment (limit 900) (gap 60))
```

The following figure illustrates how these rules allow greater parallel lengths as gap between segments increases. Gap and length combinations to the left of the curve are rule violations.

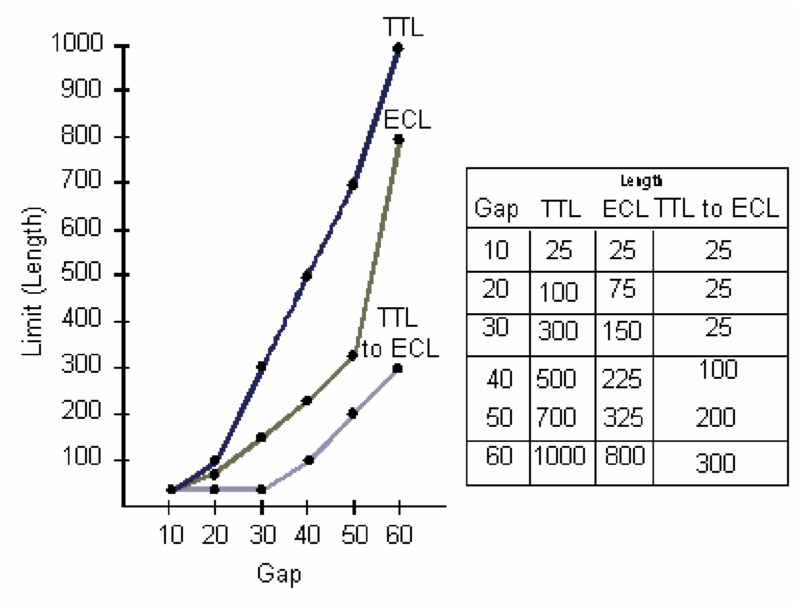
Figure 6-3 Parallel Segment and Gap Combination



Controlling Class-to-Class Crosstalk

Sometimes crosstalk is more critical between certain classes of nets. Class-to-class crosstalk rules control crosstalk between different net classes or between nets of the same class. The following figure shows how multiple class-to-class rules apply simultaneously to approximate the crosstalk characteristics for different technologies.

Figure 6-4 Multiple Class-to-Class Rules



	Length		
Gap	TTL	ECL	TTL to ECL
10	25	25	25
20	100	75	25
30	300	150	25
40	500	225	100
50	700	325	200
60	1000	800	300

Controlling Cumulative Coupled Noise

Noise coupling between wires on a design can be the cause of circuit malfunction or failure. To minimize and control noise coupling during autorouting, you must specify the maximum noise that a receiving net can tolerate. To determine whether an excessive noise condition exists, the contributions from all noise sources are accumulated. If the total exceeds the maximum noise specification, a violation exists.

Some nets are noisier than others because of the circuit technology used or due to circuit function, but any net in a design is a potential noise source. Each noise transmitting net must have one or more weights assigned that correspond to the amount of noise transmitted as a function of the gap to a parallel receiving net.

You can specify these factors in global (pcb) rules, by class, by net, and by fromto. The autorouter can reduce or eliminate both inter- and intra-layer coupled noise. The relationship between gap, wire length, layer factor, and total noise coupled onto a net is shown by the following coupled noise expression:

```
NoiseR = S (L * weight (gap) * layer_factor)
```

where:

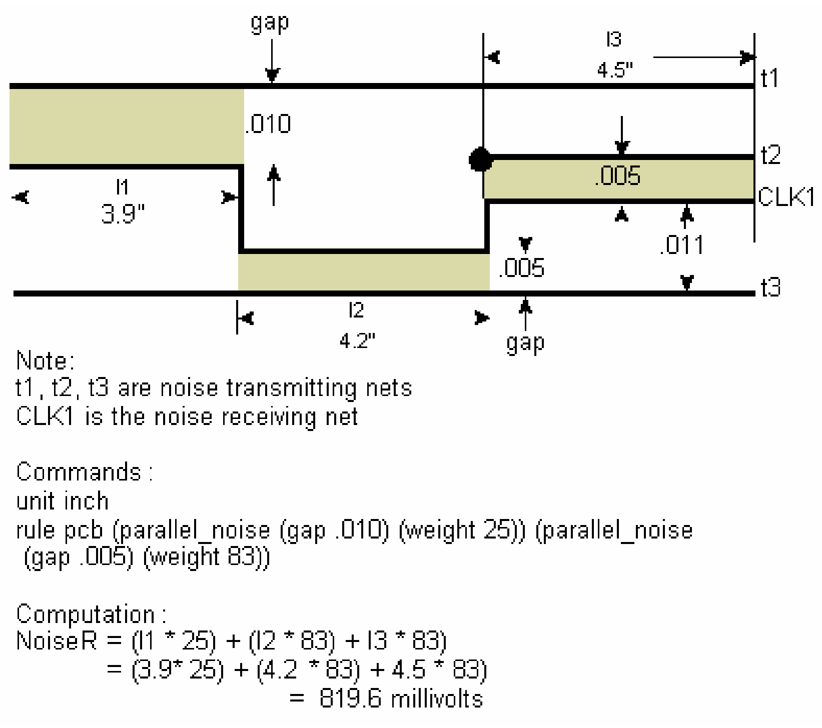
- NoiseR = the sum of all coupled noise components in a receiving net
- L = the measured wire length over which coupling occurs
- weight(gap) = the factor proportional to noise generated by the transmitting net per unit length, at a specified gap
- layer_factor = the inter or intra-layer value that depends on design characteristics; you can specify multiple values in a layer matrix.

The coupled noise expression shows how noise is computed for a receiving net where one or more noise transmitting nets exist. The additional term, not included in this expression but used to determine where routing violations occur, is the maximum noise rule (noise budget) for the receiving net.

The router computes the noise that intrudes onto a receiving net from neighboring wires. The computation is made for each transmitting wire by multiplying its noise weight by its parallel or tandem length. The weight-times-length product is multiplied by the layer_factor adjust the noise value as a function of the routing layer. Contributions from all transmitting nets are accumulated for a receiving net and the sum is compared to the maximum noise budget for the receiving net. When coupled noise accumulation on a net exceeds the maximum noise rule, the condition is a violation.

The following figure illustrates a coupled noise violation where the cumulative coupled noise on net CLK1 equals 819.6 millivolts while the max_noise rule for the net is 600 millivolts.

Figure 6-5 Coupled Noise Computation



Note:
t1, t2, t3 are noise transmitting nets
CLK1 is the noise receiving net

Commands :

```
unit inch
rule pcb (parallel_noise (gap .010) (weight 25)) (parallel_noise
 (gap .005) (weight 83))
```

Computation :

```
NoiseR = (l1 * 25) + (l2 * 83) + l3 * 83)
       = (3.9* 25) + (4.2 * 83) + 4.5 * 83)
                         =  819.6 millivolts
```

Differential Pair Routing Issues

Differential pairs are one of the more difficult topologies to get to work. The basic routing issues are:

- Fanout
- Gathering the nets together
- Controlling uncoupled length
- Controlling phase mismatches

  It is very rare that the pins are located exactly side by side where they can be parallel and not have length mismatches. Phase compensation is usually necessary.
- Getting through other pins

  This is where the geometry of the design will not let the pair remain side by side the entire way and must either split around an obstacle or split to separate layers.
- Turning corners differentially

  This is where routes must change layers and use via patterns.
- Extended nets

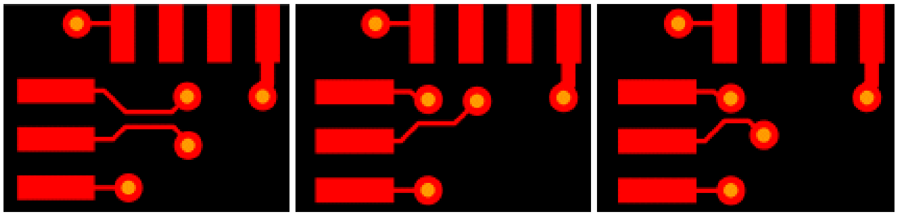
Fanout

When designs are started they inevitably need to be fanned out. This presents some interesting problems for any router. When the differential pair pins are located beside each other on a BGA, QFP, or other active type SMD device, the autorouter fans them out side by side and attempts to get the vias as close together as possible.

Fanout on standard SMD Components

The following figure shows a differential pair gathered during fanout on a PLCC type SMD component. The pair is quickly pulled together after exiting the pads. Also, note that the pair is fanned out inside the component. It could also be fanned out outside just as easily.

Figure 6-6 Differential Pair Fanout Examples



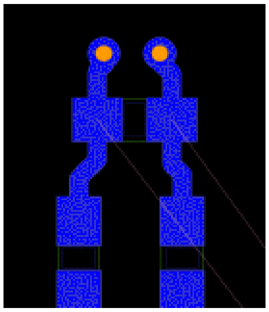
Also, note that when the system encounters a differential pair, the pair must override the normal fanout control directions if they don't allow the pair to be placed as shown. In other words, the pair must “win out” over the normal in / out technique used by the autorouter.

Fanout on Discrete SMD Components

Proper Fanout Method

During fanout the differential pair must be controlled to fanout towards each other. This rule is most critical when the pins are on separate components than if the pair is on a single component as shown in the following figure.

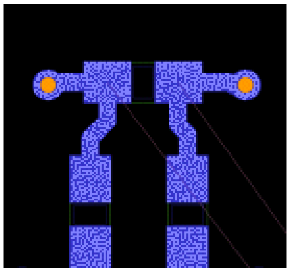
Figure 6-7 Ideal Discrete Differential Pair fanout - Two Components



Improper Fanout Method

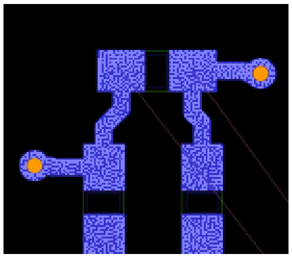
When the differential pair is on a single component the conditions in the following figures must be avoided.

Figure 6-8 Non-ideal Discrete Differential Pair Fanout



You must override the stub length constraints when fanning out differential pairs on discretes in order to avoid the phenomenon shown in the following figure. In these cases, the stub length rule must be artificially set to zero to prevent a condition where via sites are found, but are not on the same component.

Figure 6-9 Stub Length Allowed to Force Differential Pair Fanout



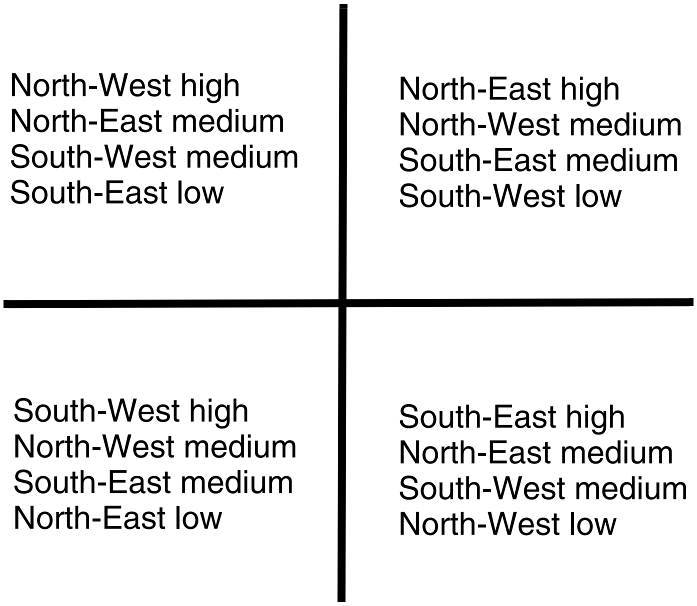
Fanout Via Location

In cases where a BGA pin is part of a net from a differential pair, a precedence rule is applied by the router during fanout to select an optimal via location and to avoid placing fanout vias for the pair in opposite directions.

The router first searches for the coupled pin of its paired net, then operates in the following way:

- If the coupled pin has no fanout via, then the current pin is routed without any preference for direction.
- Otherwise, the fanout via location for the current pin is determined using a decending order priority with respect to the quadrant where the fanout via of the coupled pin is located as shown in Figure 6-10.

Figure 6-10 Differential Pair Fanout Via Location Precedence



North-West high North-East medium South-West medium South-East low	North-East high North-West medium South-East medium South-West low
South-West high North-West medium South-East medium North-East low	South-East high North-East medium South-West medium North-West low

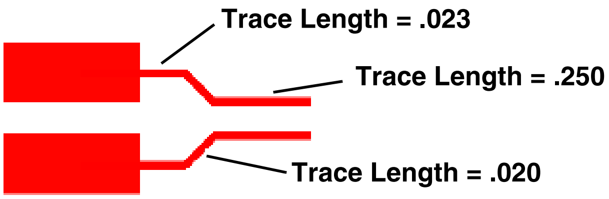
Gathering the Nets Together

Gathering is the process where the two halves of the differential pair come together and fall within the coupling band. This process starts a new *coupled event* and ends accumulation of the previous *uncoupled trace length accumulation*.

Note that a gather takes place at each pin/pad exit as well as at each and every via that the pair passes through. Therefore, this accumulation may have a significant effect on the total uncoupled length of the pair.

For example, at pins, when you consider the following trace length example:

Figure 6-11 Pin Trace Lengths for Gathering



Gather Point Location and Pin to Gather Point Routing

When routing differential pairs, the autorouter computes the location of gather points and places them as far away from their pins as needed to ensure that the net pair are phase matched. If required, small jogs are introduced in the wiring between pins and gather points for the purpose of elongation.

General Gathering Rules and Requirements

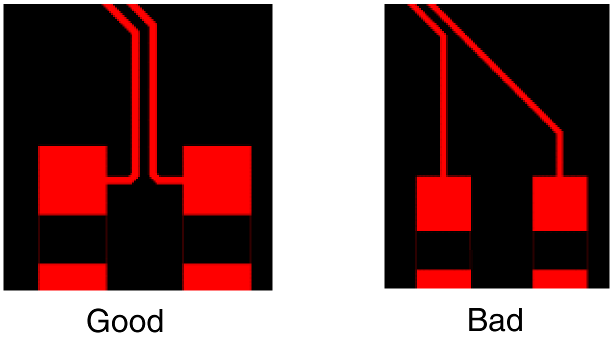
Differential pairs are gathered and split in a variety of situations and each situation has slightly different choices. These situations include:

- Entry and exit from pins to ensure that:
    - ❑ good pad entry and exit in general takes priority.

      Entering or exiting a pad that introduces a manufacturing problem in order to gather quicker is a poor trade-off. Pads should be entered and exited at the closest point that traces can come together while meeting the coupling constraint. See the following figure for an example.

- ❑ manufacturing errors are not made (acute angles or violate the pad to trace same net rules).

Figure 6-12 Good and Bad Pad Gather Examples



- ■ Adding vias during a normal route.

  Vias are generally subject to via patterns that control how the pair splits and then re-gathers.

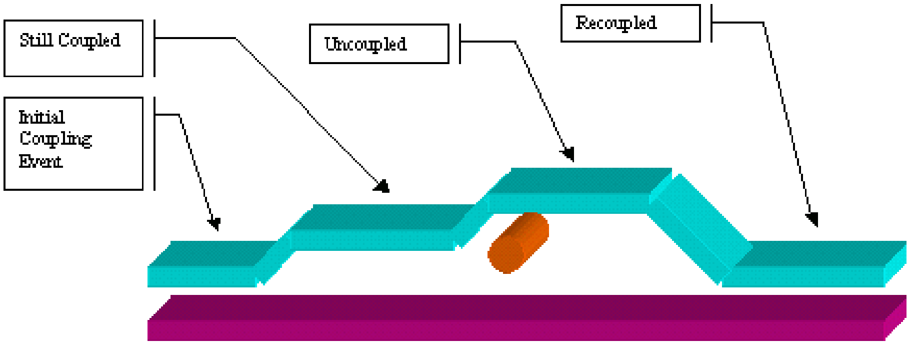
- ■ Avoiding an obstacle such as a pin, via or barrier:
  - ❑ Gathering and splitting for this purpose should occur as close to the obstacle as possible.
  - ❑ This splitting will not be done if the two nets can get by the obstacle as a coupled unit. Otherwise, no splitting shall be allowed due to an obstacle on one layer and not the other.

Coupling and Uncoupling Events

When you route a differential pair; ideally you want it to route from start to finish within the coupling band. However, this is not always possible and trace that is outside this band (uncoupled length) can have adverse effects on the pair's signal quality.

Therefore, uncoupled length must be controlled by a rule that monitors its accumulation as the differential pair is routed. The following figure illustrates coupling and uncoupling events as a (+) trace bends around a via (in red).

Figure 6-13 Coupling / Uncoupling Events Example



Rules to Control Uncoupled Length

Since this length is an accumulation of all the uncoupled lengths, you need to examine where this uncoupled length comes from.

In general it is generated in two places:

- Gathering
- Obstruction avoidance

You control the maximum amount of uncoupled length allowable for differential pairs by setting a tolerance using the `max_uncoupled_length` rule. See Routing Rules in the *Allegro PCB Router Command Reference* for details.

To set a maximum uncoupled length tolerance for differential pairs

1. Choose *Rules – <rule level> – Differential Pairs*.

   The Differential Pair Rules dialog box appears.

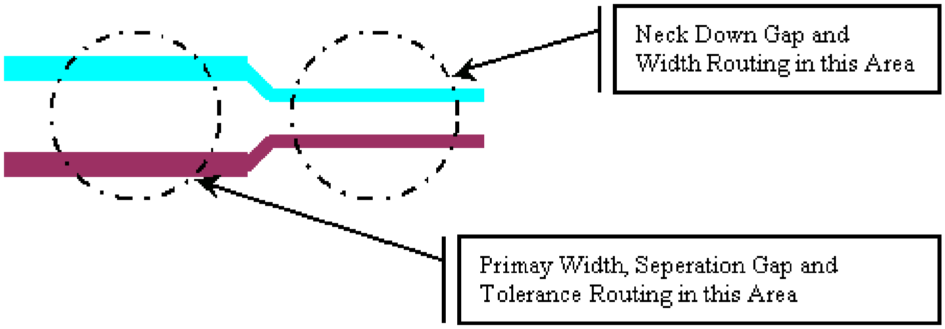
2. Enter a tolerance value in the *Max Uncoupled Length* data entry box.
3. Click *Apply* or *OK*.

Necking

With the continued shrinking of package sizes and their associated pin pitches, it is becoming more difficult to route into them or through them. To accommodate this you need to define rule areas that temporarily allow the use of smaller feature sizes.

When routing with differential pairs, the change in trace width and gap must be adjusted together to maintain the desired differential impedance. Care must also be taken to not route the pair using the smaller dimensions over it's entire length.

Figure 6-14 Neck Down Gap and Width Routing Example


Neck Down Gap and Width Routing in this Area
Primay Width, Seperation Gap and Tolerance Routing in this Area

Allowing the autorouter to *neck down* to a smaller line width and/or spacing provides better routing results than splitting the pairs around obstacles. Therefore, necking is preferred over splitting and is done within the following limitations.

- If possible, the autorouter will take advantage of the (-) coupled tolerance value. See Routing Rules in the *Allegro PCB Router Command Reference* for details.
- Where the tolerance does not allow the wires to neck sufficiently, the neck down gap and width values are used (with no gap range in between). See Routing Rules in the *Allegro PCB Router Command Reference* for details on each rule.
- Necking is only be used to route between pins or vias associated with a component such as fanout vias. It is not used to fit between a pin and a wire, two wires, or two vias that are not part of a component.

- Routing stays necked until there is enough room to go back to the primary separation. The wires stay necked while going through a pin or via field. In other words, a succession of necking and un-necking does not occur within the pin or via fields.

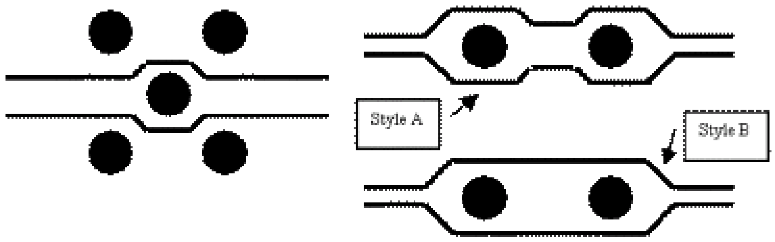
Splitting

Where necking of wires cannot be done, the autorouter splits the pair temporarily and routes each piece of the pair as a single trace. The decision on which way to split is made with respect to three values.

- The length of the segment that will exist between the splits.
- The amount of uncoupled length each method produces.
- The number of times splitting needs to be done.

If the results of two splitting methods are both visually acceptable, the method that produces the least amount of uncoupled length is used by the autorouter. Finally, in cases where there would be several iterations of splitting and re-joining, then the routes stay separated (see Style B in the following figure).

Figure 6-15 Splitting Techniques



Phase Control

As you route differential pairs, mismatches in length between the nets in the pair are common and are referred to as *phase mismatches*. If these mismatches are excessive, then the signals in each half of the pair can become distorted in relation to each other causing system problems.

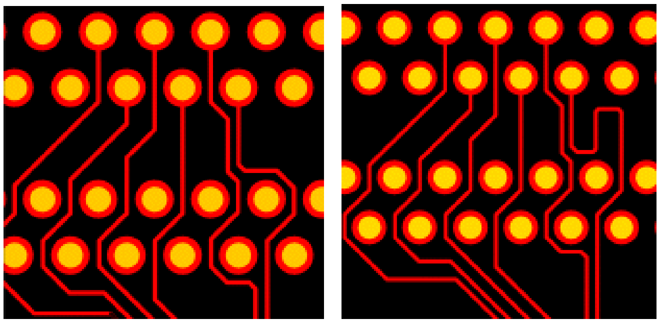
Where are phase mismatches created?

Phase mismatches are generally created in three places:

- Pin Placement on components (connectors or bad pinouts for example)
- Fanout (vias not being closely placed together)
- General Routing (vias, tuning bends and obstacle avoidance)

The following images show how phase adjustments can be made before the main coupling event to ensure that the signals are in phase when they become coupled.

Figure 6-16 Phase Compensation



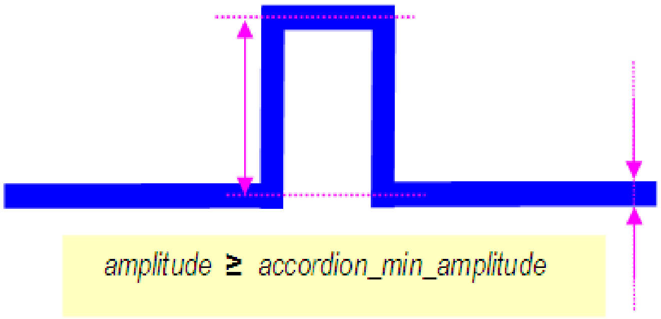
Calculating Phase Errors

In order to properly calculate a phase error, there has to be a starting and ending point for the electrical signal. The start point is a *driver* and the end point is a *receiver*. Properties can be put on a pin to designate if it is a driver or receiver. If none of the pins has a driver property, the longest pin pair is found, and the pin that is to the lower left will be the driver.

If a phase tolerance is applied to a net, then the longest driver/receiver path will be looked at. If the tolerance is applied to a fromto, then one pin is considered a driver and the other a receiver based on their topology within the net. When phase is checked, the length is compared all the way back to the driver. So this means that phase could be checked across several routed connections within a set of differentially paired nets.

Accordion Amplitude

The accordion_amplitude parameter of the rule command ensure that the Router restricts elongating accordions by their minimal amplitudes.

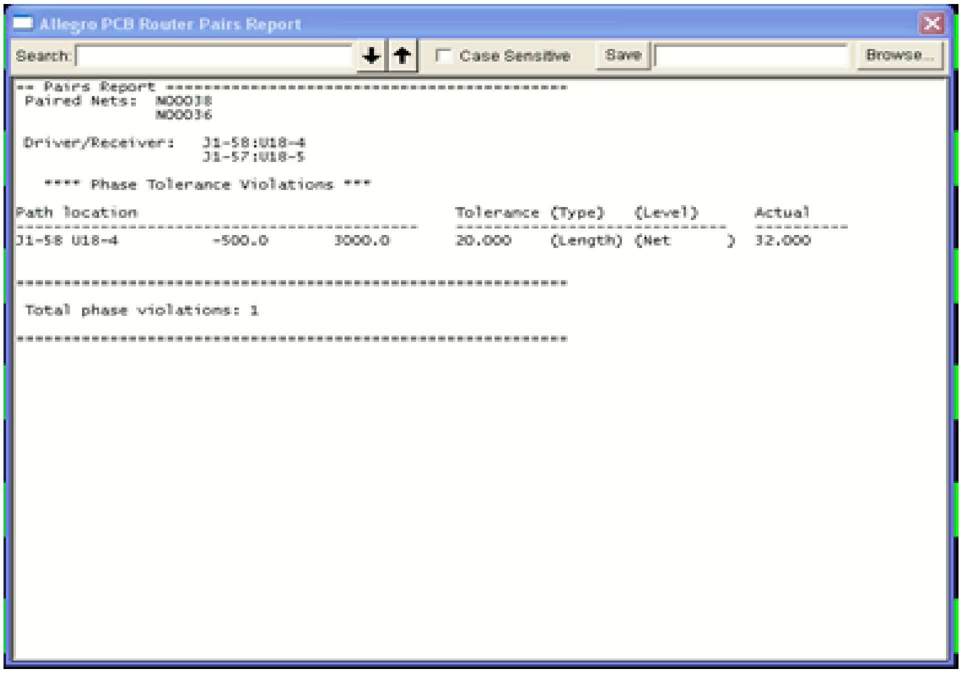


Note: If the accordion amplitude is not set, the default value used by the Router is 3 times the wire width.

An elongation accordion may be created for compensation of the ‘phase_shortage’ when:

```
phase_shortage/2 + wire_width ≥ accordion_min_amplitude
```

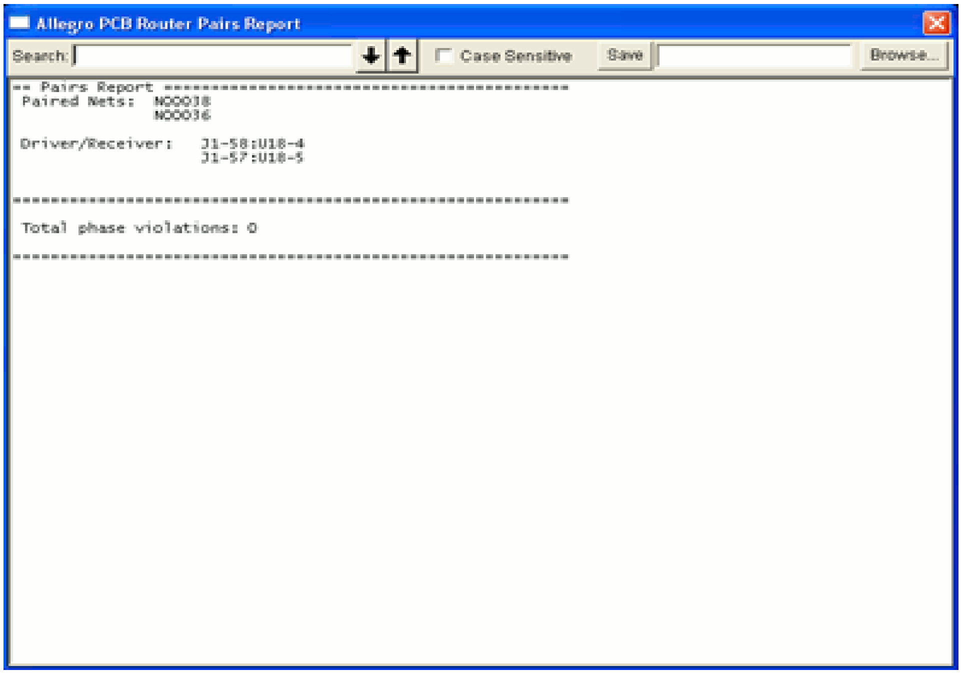
For example, if phase_shortage = 32 and wire_width = 10, the accordion is not created.



```
== Pairs Report ============================================
 Paired Nets:  N00038
               N00036

 Driver/Receiver:   J1-58:U18-4
                    J1-57:U18-5

   **** Phase Tolerance Violations ***

Path location                                Tolerance (Type)   (Level)      Actual
-------------------------------------------  -----------------------------   ----------
J1-58 U18-4          -500.0       3000.0     20.000    (Length) (Net      )  32.000


============================================================

 Total phase violations: 1

============================================================
```

Now, add the following rule and re-run the Router:

```
rule pcb (accordion_amplitude 100 15)
```



```
== Pairs Report ============================================
 Paired Nets:  N00038
               N00036

 Driver/Receiver:   J1-58:U18-4
                    J1-57:U18-5


============================================================

 Total phase violations: 0

============================================================
```

Phase Elongation Forbidden

The Phase_elongation_forbidden parameter of the rule command ensures the Router will prohibit elongation in a region.

The following example ensures that the Router will disallow any elongation in the region REG1:

```
rule region REG1 phase_elongation_forbidden
```

Cost Elongation Coupled Segments

The elongation_coupled_segments parameter of the cost command is usedby the Router to cost the elongation of coupled segments in a differential pair.

Diffpair Driver Receiver Model Translation

In PCB Editor, you can specify the driver and receiver pin pair on both nets of a differential pair. When the design is sent exported, the Router then uses this driver-receiver pin pair definition to perform phase matching and uncoupled rules calculation based on the driver-receiver pairs.

Getting through Other Pins

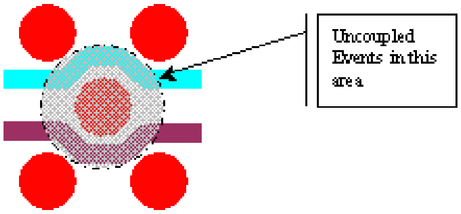
Routing through Interstitial Pins

Today's designs require the use of interstitial pin connectors to get the high pin counts necessary for backplanes and the high I/O transfers to and from theses designs.

These connectors pose a difficult problem for differential pairs. In many cases, it is impossible to get the pair through the pin fields at the default route widths. In general, the best that can be done (with standard route rules) is to split the pair and route them individually around the pins as shown in the following figure.

The cost of this technique is an uncoupled event occurring as the pair splits and passes through the pin field.

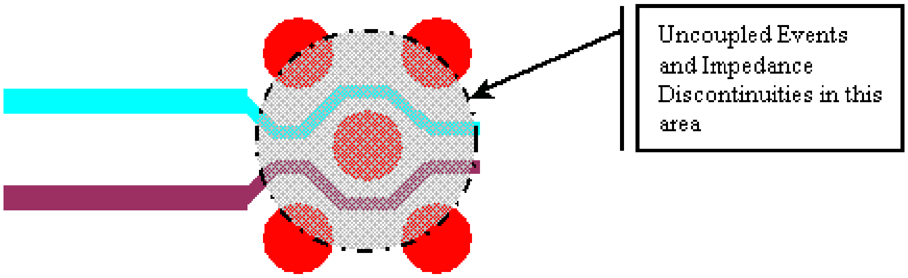
Figure 6-17 Interstitial Pin Field - Standard Route Width



Assuming that the pair can't fit together through the pin field, the autorouter uses a three-tiered approach to routing through these obstacles to minimize uncoupled events.

The first choice is to use the neck down and split method shown in the following figure.

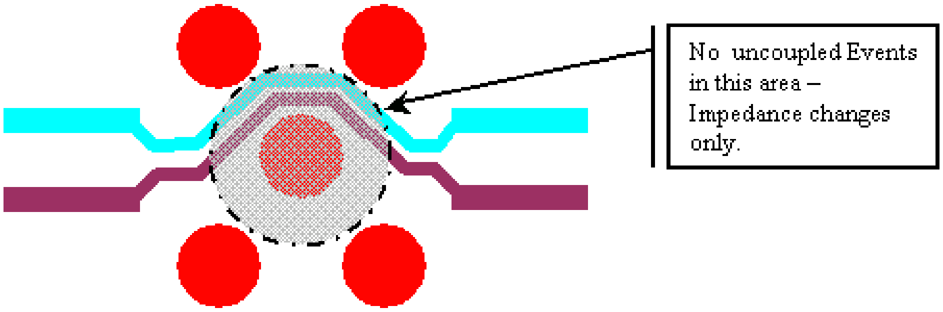
Figure 6-18 Neck down and split



The second choice is to route the pair normally but split around the pins as shown in Figure 6-17 on page 311.

If all else fails, the autorouter reduces the trace width and gap using the neck down feature set and then routes normally as shown in the following figure.

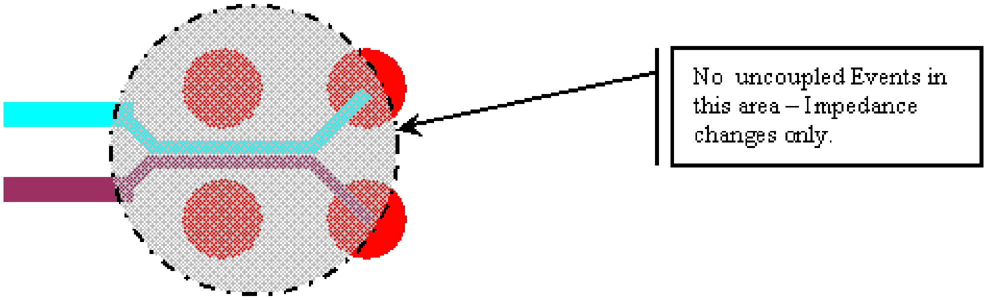
Figure 6-19 Neck down normal route



Routing into BGA Pins

BGA pin counts are climbing while the pin pitch is decreasing. This raises the need for neck down to be used in the BGA pin field. Generally, this is done through the use of a region rule. However, region rules are not necessary if necking rules are applied correctly. An example is shown below in the following figure.

Figure 6-20 Routing into BGA's Example



If the pair splits apart, then the sum of the segments getting through the pin field contributes to the uncoupled length constraint. If vias were necessary they contribute to the max_via_ count rule and so on.

Turning Corners Differentially

When a connection has enough difference between it's x and y start-endpoint coordinates to force the use of vias, it's obvious that a turn will produce some length / phase mismatch as well as potential coupling problems. The only solution to keeping them the same length will be to add length to one half of the pair.

The following figure shows an unrouted example of a sample pair from an actual design. Notice the coupling between the segments as they go from one end to the other as shown in Figure 6-22 on page 313. The only major separations are where the system negotiates a turn from horizontal to vertical. Also note that as the turns are negotiated that the segments immediately get together again as soon as possible.

Figure 6-21 Unrouted Differential Pair

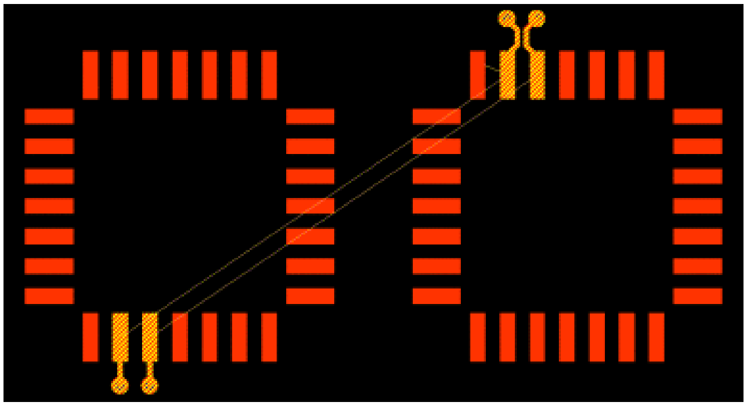
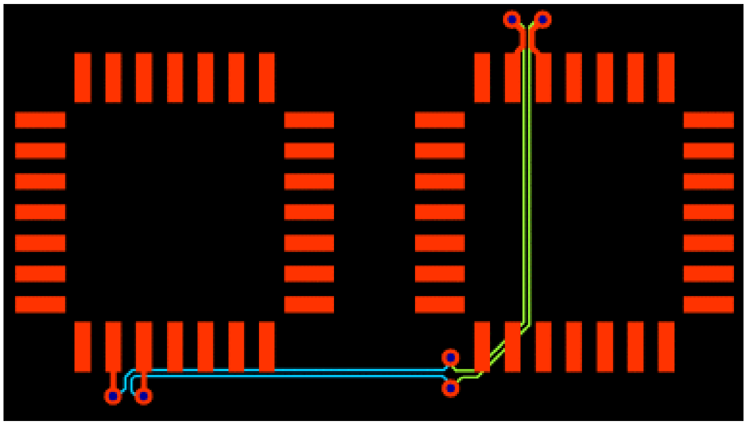
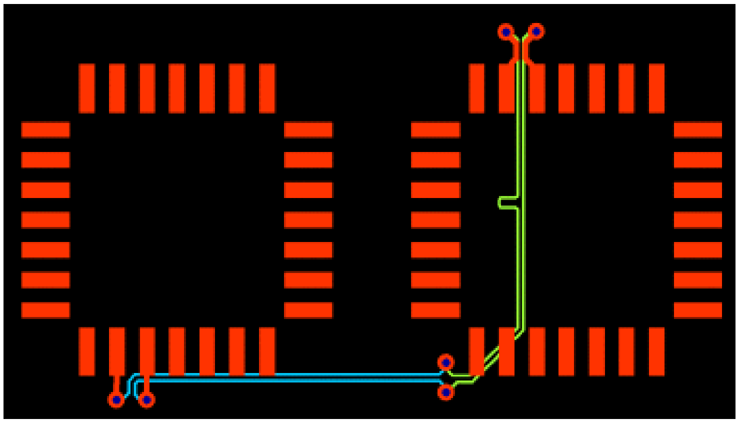


Figure 6-22 Initial Routing of the Pair



The following figure shows the pair fully routed and a tiny tuning trombone to restore the pairs phase relationship.

Figure 6-23 Length Coupling Relationships

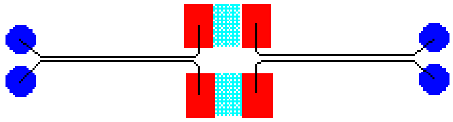


Working with Extended Nets

What are Extended Nets?

Extended nets (XNets) are nets that pass from source to a load out another pin on the same component and then continue on to more loads. They are typically found as series terminated nets. When the router receives designs that have many extended differential nets in them, it maintains the differential characteristics of each half of the net as shown in the following figure.

Figure 6-24 XNet Differential Pair Route



Handling Extended Nets

You can create a group in the router to handle XNets coming from PCB Editor. Once the group is created, you apply the `diffpair_group_level` rule to the group to specify that the `max_uncoupled_length`, and `phase_tolerance` values apply to the *total* of all of the fromtos in the group. If the group does not represent an XNet, the values apply to each fromto individually.

For example:

XNet1 is made up of NetA (pins 1, 2, 3), and NetB (pins 4, 5, 6)

XNet2 is made up of NetC (pins 7, 8, 9), and NetD (pins 10, 11, 12)

Pin1 ----- Pin2 ----- Pin3	Pin4 ----- Pin5 ----- Pin6	(XNet1)
Pin7 ----- Pin8 ----- Pin9	Pin10 ----- Pin11 ----- Pin12	(XNet2)

Nets A and C will need to be paired, as well as nets B and D:

```
define (pair (nets NetA NetC))
define (pair (nets NetB NetD))
```

Groups will need to be defined to represent the XNets:

```
    define (group XNet1 (add_fromto (fromto Pin1 Pin2)))
define (group XNet1 (add_fromto (fromto Pin2 Pin3)))
```

```
    define (group XNet1 (add_fromto (fromto Pin4 Pin5)))
    define (group XNet1 (add_fromto (fromto Pin5 Pin6)))
    define (group XNet2 (add_fromto (fromto Pin7 Pin8)))
    define (group XNet2 (add_fromto (fromto Pin8 Pin9)))
    define (group XNet2 (add_fromto (fromto Pin10 Pin11)))
    define (group XNet2 (add_fromto (fromto Pin11 Pin12)))
```

The diff pair rules will then get applied to the XNet groups:

```
    rule group XNet1 (max_uncoupled_length 1000)
    rule group XNet1 (diffpair_group_level total)
    rule group XNet2 (max_uncoupled_length 500)
    rule group XNet2 (diffpair_group_level total)
```

Since the nets are paired, each fromto within one net is paired with an appropriate fromto on the other net. Any rule applied to a fromto will also get applied to the fromto it is paired with. The most conservative rule will be used. So in the example, a maximum uncoupled length of 500 will be used, and it will get applied to the total of all fromtos in each of the groups.

Working with High Density Interconnect Designs

Using Set Microvia

Use the `set` command to control the availability of licensed MicroVia features.

The Microvia condition of the `set` command controls the availability of MicroVia features at the command line and in the Graphical User Interface.

Note: MicroVia features made available by this `set` command require a special license. See Understanding Licensing on page 18 for details.

Use `set microvia on` when you plan to incorporate microvias in your design. The following features become available.

Enhanced fanout

This feature provides improved fanout for vias under SMD pads when pads may be directly opposite each other on opposite sides of the design. For further information, see `fanout` command.

Stacked vias

This feature allows stacking of blind and buried vias at the same location on different layers, and provides enhanced support for depth control. See the `rule` command for information regarding:

stack_via rule

stack_via_depth rule

See the `report` command for information regarding the stack_via_depth report.

Via arrays

This feature provides the capability to define a template for via arrays in the Design File which works with a circuit rule to create via arrays automatically during automatic routing. Additional features enable interactive modification of via arrays.

7

Troubleshooting the Design File

In this chapter . . .

The information in this chapter is intended to be used to update and correct Design files that predate PCB Router Version 5.0 interfaces. Whenever possible, either correct the data in the design layout system or update your interface instead of using the methods described in this chapter.

Always make a backup copy of your Design file before you edit it. If you modify your Design file and add rules or data that is not supported in your layout system, the results are unpredictable when you return to the layout system.

The router Design file is a text file in ASCII format. You can edit the Design file using text editor.

Note: Before you edit a Design file, familiarize yourself with the router file structure and the design language syntax by referring to the *Allegro PCB Router Design Language Reference*.

Correcting Improper Keepout Definitions

The five types of keepouts that you can define in a Design file are described in the following table.

Table 7-1 Types of Keepouts

Keyword (type)	Description
bend_keepout	Prohibits wire corners in the keepout region
keepout	Prohibits wires, vias, and components in the keepout region
place_keepout	Prohibits components in the keepout region
via_keepout	Prohibits vias in the keepout region
wire_keepout	Prohibits wires in the keepout region

The exact location of keepouts within the file is important. For example, when you're reworking a completed design (already placed and routed) that has keepouts defined globally in the structure section of the Design File, if you unplace the components, the keepouts remain behind on the design. When you automatically place the components, the components' locations are different and the keepouts are in the wrong positions.

To correct this, you must move the keepouts from the structure section to the images in the library section of the Design File to make them part of the image definitions. This process can be a little tedious, since keepout regions defined in the structure section of a Design File use absolute coordinates; you must define a keepout region attached to an image relative to the image's origin.

The following example shows how a 0.250 square via_keepout definition (in bold type) appears in the structure section of a Design File.

```
(PCB demo8
(structure
(grid via 0.025)
(boundary (rect pcb .450 .050 1.85 1.6))
(boundary (rect signal 0.550 0.150 1.75 1.5))
(via v25 (spare testpt1 test pt2))
(rule (width .008) clearance .008))
(layer s1 (type signal)
(direction horizontal))
(layer p1 (type power)
(use_net +5V GND))
(layer s2 (type signal)
(direction vertical))
(via_keepout (rect s1 0.7 0.3 0.95 0.55))
(grid wire 0.000 s1)
(grid wire 0.005 s2)
)
```

If you transpose this keepout to the lcc20 image in the library section of the file, it appears as shown in the following example.

```
(library
(image lcc20
(pin p25x75 1 0.0000 0.1500)
(pin p25x50 (ARRAY 2 3 1 0.0500 0.1500 0.05 0.0))
(pin p25x50 (rotate 90) (ARRAY 4 8 1 0.1500 0.1000 0.0 -0.05))
(pin 25x50 (rotate 180) (ARRAY 9 13 1 0.1000 -0.1500 -0.05 0.0))
(pin 25x50 (rotate 270) (ARRAY 14 18 1 -0.1500 -0.1000 0.0 0.05))
(pin p25x50 (ARRAY 19 20 1 -0.1000 0.1500 0.05 0.0))
(via_keepout (rect s1 -0.125 -0.125 0.125 0.125))
)
```

The coordinates for the rectangular via_keepout region in the previous example are relative to the image’s origin, which in this example is the component center.

Correcting an Improper Power Layer Definition

Occasionally, a Design file is missing a power net definition. This is always the result of either an error in the design layout database or a fault in the translator. If a power net is not defined by a `use_net` statement, the router assumes it is a signal net. When you start the router, it warns you if it finds a signal net with more than 150 pins.

The router uses connectivity to prioritize components for automatic placement operations. Because power nets usually contain a large number of pins, the router considers them to be the largest nets in the design. If the router uses power nets to determine component connectivity, the placement results can be unsatisfactory.

If possible, you should always correct this type of Design file error in your design layout system. A correct power net definition, showing a single power layer that is split between +5V and GND, is shown in the next example.

```
(PCB demo8
(structure
(grid via 0.025)
(boundary (rect pcb .450 .050 1.85 1.6))
(boundary (rect signal 0.550 0.150 1.75 1.5))
(Via v25 (spare testpt1 testpt2))
(rule (width .008 (clearance .008))
(layer s1 (type signal)
(direction horizontal))
(layer p1 (type power)
(use_net +5V GND))
(layer s2 (type signal)
(direction vertical))
(grid wire 0.000 s1)
)
```

Removing a Huge Component from the Design File

Sometimes a graphic object in the design layout system gets translated to the Design file as a component. The object could be for documentation purposes, part of the design silk-screen, or some other graphic object that is not part of the network or component data.

If the object is approximately the same size as the design and is defined as a component, automatic placement is unable to place other actual components over the erroneous component. If possible, you should always correct this error in your design layout system.

An example of what this type of Design file error looks like is shown in the next example.

```
(placement
(component NO_COMP
(place U5 1.0 0.75))
(component 1cc20
(place U1 0.8000 0.4000 1 0)
(place U2 1.5000 1.1000 2 0)
(place U3 1.5000 0.4000 2 0)
(place U4 0.8000 1.1000 1 0)
)
(library
(image NO_COMP)
(image lcc20
(pin p25x75 1 0.0000 0.1500)
(pin p25x50 (ARRAY 2 3 1 0.0500 0.1500 0.05 0.0))
(pin p25x50 (rotate 90) (ARRAY 4 8 1 0.1500 0.1000 0.0 -0.05))
(pin 25x50 (rotate 180) (ARRAY 9 13 1 0.1000 -0.1500 -0.05 0.0))
(pin 25x50 (rotate 270) (ARRAY 14 18 1 -0.1500 -0.1000 0.0 0.05))
(pin p25x50 (ARRAY 19 20 1 -0.1000 0.1500 0.05 0.0))
)
```

A

Router Startup Options

Startup Options

The following tables describe the router startup options.

Table A-1 on page 326 describes general startup options you can specify from the Startup dialog box.

Table A-2 on page 327 describes additional startup options you can specify using the `specctra` command.

Table A-1 General Startup Options

Startup dialog box field	Switch	Use this option to . . .
Design or Session File		Runs the router using this design file or session file.
Wires or Routes File	-w <*file*>	Loads the named wires or routes file.
Placement File	-place <*file*>	Loads the named placement file. This data overrides the placement data in the design file.
Do File	-do <*file*>	Begins the session by running commands from this do file.
No Graphics	-nog	Runs the router in non-graphics mode (without the GUI).
Quit After Do File	-quit	Exits the router after running the last command in the do file.
No Preroutes	-nowire	Ignores prerouted wires in the design file.
Show Usage	-help	Displays help information on command line switches in the output window.
Don’t Strip Orphan Shapes	-noclean	Keeps orphan shapes (copper without net assignments). Without this option, the router removes all orphan shapes.
Simplify Polygons	-sim	Replaces small polygons with rectangles. Useful for component pads with complex shapes. A design with many polygon-shaped pads can slow performance.

Startup dialog box field	Switch	Use this option to . . .
Password File		Specifies either a network license file or a node-locked password file. Use this option only to specify a file that is not in the usual location or in your search path. You can choose Default, Node, or Network.
Default		The router uses the default search path, and looks first for a network license file.
Network	-lf <*file*>	The router uses the network license file specified by the path and filename that you enter for <*file*>.
Did File	-did <*file*>	Specifies the file where the router commands are logged during a session. The default filename is the month, day, and time with a .did extension.
Message Output File	-o <*file*>	Redirects the output transcript to the file you specify.
Status File	-s <*file*>	Specifies the autorouter status file, which contains routing status information. The default status file is monitor.sts.
Color Mapping File	-c <*file*>	Specifies the color map file that defines the colors and patterns for design objects and graphic features in the work area. If you do not use this switch, the router uses color and fill pattern definitions from the colors file in your .cct directory, from the design file, from the color.std file in the current directory, or from internal defaults. See the -noinit switch in Table A-2 on page 327.

Table A-2 Additional `specctra` Command Startup Options

Switch	Description
-allegro	Loads PCB Editor key definitions.

Switch	Description
-cct_version	The software version to use.
-docmd “ <*command*>“	Runs an initial router <*command*>. You can include multiple commands by separating them with semi-colons (;). **Note:** `All -docmd` commands are executed (left to right order) before any `-do` files specified in the command line.
-display <*host: display*>	Directs the router GUI from the host system where you are running the router to a display terminal where you want to display router graphics. This switch is available only on UNIX systems.
-fixnet	Skips processing of specified net(s) during startup. Typically used on power net(s) to speed invocation. Syntax: `-fixnet <netid>`
-geometry [Wx H] [[+**I**-] x [+**I**-] y]]	Sets the size and position of the router window. You can specify the width (W) and height (H) of the window, and the number of pixels horizontally (x) and vertically (y) between the top left corner of the window and the top left **(**+) or bottom right **(**-) corner of the screen. This switch is available only on UNIX systems.
-ignore_net	Causes the router to ignore the input net names of wires read from a wires file. Instead, the wires are identified with the net names associated with the pins they connect to.
-nocutout	Prevents cutting the keepouts of pins and conductors that belong to the same component.
-noinit	Prevents the router from reading the colors file and keys file in the .cct directory under your home directory. The router reads these files by default, if they exist, before reading any do files you specify with the -do switch. When you use the -noinit switch, the router uses color definitions from the file you specify with the -c switch, from the design file, or from internal defaults.

Switch	Description
-private_colors	Allows the tool to allocate colors, even if these colors are not available to X-Windows. Use this option if another application (such as Netscape Navigator) is already running and has allocated colors needed by your session. Note: This option is only available for UNIX platforms.
-product	Starts the session in the specified product mode. Syntax: `-product <product suite>`
-version	The MPS (Message Passing Subsystem) version number.

B

Setting Colors and Fonts

In this appendix . . .

This appendix explains how the router uses default fonts and colors to display the menu bar, dialog boxes, tool bar, and other elements of the GUI. On UNIX and Windows systems you can change the colors and fonts.

The colors used by the router menu bar, dialog boxes, tool bar, and other components are specified differently from the design objects displayed in the work area. The colors of design objects are specified in the design file or in the router color map file.

Changing Default Colors on UNIX Systems

You change the colors used for the router GUI by editing your X resources file. This file is usually named .`Xdefaults` and is usually located in your home directory.

You can set the colors of the router window by using either color names or color numbers. The color names you can use are usually listed in `/usr/lib/X11/rgb.txt` on your X server or `/usr/openwin/lib/X11/rgb.txt` for SPARC and Solaris platforms. You can specify color numbers with a hexadecimal value that represents the amount of red, green, and blue intensity on a scale from 00 to ff. Precede the hexadecimal value with the pound sign (#).

Examples of hexadecimal color entries are shown in the following table.

Table B-1 Hexadecimal Color Entries

Value	Color
#ff0000	red
#00ff00	green
#0000ff	blue
#000000	black
#ffffff	white
#888888	medium gray

The fonts that are available on your X server are determined with the `xlsfonts` command. See your X Window System documentation for further details. The default values for the color resources you can set for the router are listed in the following table.

Table B-2 Color Default Values

Resources	Color
pcb_da*idleColor:	green
pcb_da*busyColor:	red
pcb_da*interruptibleColor:	orange
pcb_da*pausedColor:	yellow
pcb_da*modalFormUpColor:	#a020f0
pcb_da*stopColor:	red

pcb_da*dofileColor:	pink

You must add the string `=ascii` to the end of the font name. The default values for the fonts you can set for the router are listed in the following table.

Table B-3 Font Default Values

Router resource	X resource	Value
Default font	pcb_da*fontList:	*-helvetica-bold-r-normal--*-120-*=ascii
Dialog Box Subtitles	pcb_da*popup*subtitle* fontList:	*-helvetica-bold-r-normal--*-120-*=ascii
Dialog Box Titles	pcb_da*popup*title* fontList:	*-helvetica-bold-r-normal--*-180-*=ascii
File Selection Boxes	pcb_da*filename* textFontList:	*-courier-bold-r-normal--*-120-*= ascii
Layer Panel	pcb_da*Layers*pop up* fontList:	*-helvetica-bold-r-normal--*-100-*=ascii
Report Window	pcb_da_fileShell*XmPushButton*	*-helvetica-bold-r-normal-
Report Window Labels	pcb_da_fileShell*XmLabel* fontList:	*-helvetica-bold-r-normal- -*-120-*=ascii
Report Window Text	pcb_da_fileShell*XmText* fontList:	*-courier-medium-r-normal--*-120-*=ascii
Scrollable Lists	pcb_da*XmList* fontList:	*-courier-medium-r-normal--*-120-*=ascii
Single Line Text	pcb_da*XmTextFiel d. fontList:	*-courier-bold-r-normal--*-120-*= ascii
Status Panel	pcb_da*Tools*popup* fontList:	*-helvetica-bold-r-normal--*-100-*=ascii

Changing Default Colors on Windows Systems

Use the Control Panel to change default colors on a Windows system.

Changing Default Fonts on UNIX Systems

You can change the fonts used for the router GUI by editing your X resources file. This file is usually named `.Xdefaults` and is usually located in your home directory.

To increase the default font size used by the router

1. Open the `.Xdefaults` file in your text editor.
2. Add the following line to the `.Xdefaults` file:

   ```
   pcb_da*fontList: -helvetica-bold-r-normal--*-140-*=ascii
   ```

3. Save the modified `.Xdefaults` file and exit the text editor.
4. Execute the UNIX command

   ```
   xrdb load .Xdefaults
   ```

When you start the router, the menu bar and other window text should be larger. If you want to change the font for other router resources, add additional lines to your `.Xdefaults` file to include the X resource and font values listed in the previous table.

Changing Default Fonts on Windows Systems

On Windows systems, the router uses the default fonts shown in the following table. The face name, height, width, and weight specifications for each font group are set for each of three graphic resolution levels: 640 by 480, 800 by 600, and 1024 by 760 or higher.

Table B-4 Windows Default Fonts

Font group	Where it's used in the router	Font used (Face name)	Height	Width	Weight	Display Resolution
MonoFont	Output windows	Courier	15	0	400	1024x768
		Courier	13	0	400	1024x768
	Report windows	Courier	11	0	400	800x600
	List box					640x480
MonoBold Font	Text field	Courier	15	0	600	1024x768
	Command entry area	Courier	13	0	600	800x600
		Courier	12	0	500	800x600
SmallText Font	Ghost text	Courier	11	7	400	1024x768
		Courier	9	7	400	800x600
		Courier	9	7	400	640x480
Default Font	All others (small)	MS Sans Serif	-11	0	400	not applicable
	All others (large)	MS Sans Serif	-13	0	400	not applicable

Note: The Default Font size depends on the font size (small or large) set in the Display Properties dialog box that you access from the Control Panel.

Cell height and width are relative values, expressed in logical units, that define a square area around the font. If the width is zero, Windows matches the aspect ratio of the physical device against the digitization aspect ratio of the available fonts to find the closest match. The match is determined by the absolute value of the difference.

A font's weight value determines its thickness. Weight values range from zero to 900. The common values for font weight are listed in the following table.

Table B-5 Common Font Weight Values

Value	Description
0	Don't care
400	Normal/Regular
700	Bold

The allegro_pcb_router.ini File

You can change any or all of the default font specifications by creating a `allegro_pcb_router.ini` file in your Windows directory. The `allegro_pcb_router.ini` file can define the fonts used for:

- Output windows, report windows, and list boxes (MonoFont)
- Text fields and text in the Command Entry area (MonoBoldFont)
- Length rule indicators (SmallTextFont)
- All other text (DefaultFont)

The following is an example of a `allegro_pcb_router.ini` file. The [GUI] section title is required. Each additional entry can appear in the file only once.The router uses internal defaults for any specification not entered in the file. All text to the right of a semicolon (;) is a comment and is ignored by the router.

```
[GUI]

; Resets Default Font specifications

DefaultFontFaceName = Arial

DefaultFontHeight = 9

DefaultFontWidth = 5

DefaultFontWeight = 400

; Resets Bold Font face name

MonoFontFaceName = Lucida Console
```

```
; Resets Small Text Font face name and weight

SmallTextFontFaceName = Lucida Console

SmallTextFontWeight = 400
```

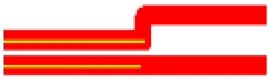
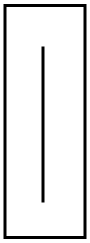
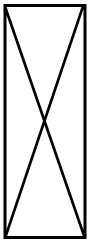
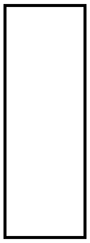
C

Understanding Symbols

Router Symbols

The router uses the following symbols to show violations:

Symbol	Default Color	Meaning
◇	Yellow	Crossover violation
□	Yellow	Clearance violation
- - - - -	Yellow	Length rule violation
▭		

Symbol	Default Color	Meaning
	White	Placement violation
Lines drawn on net segments. 	Yellow	Uncoupled diff pair net segment violation.
	Yellow	Diff pair coupling point with no phase violation.
	Yellow	Diff pair coupling point with a phase violation.
	Yellow	Diff pair uncoupling point.

Glossary

A B C D E F G H I J K L M
N O P Q R S T U V W X Y Z

A

accordion

antipads

B

balanced daisy chain topology

BBV (blind and buried vias)

balanced daisy chain topology

boundary

broadside coupled or tandem

buried resistor

bus

bus diagonal routing

C

capacitors

circuit

class

clearance

cluster

color map file

comb

companion net

complete path

component

conflicts

converge autorouting phase

cost

coupled

coupling event

coupling tolerance

crosstalk

D

delay skew

design file

DFM (design for manufacture)

did file

differential pair

discretes

do file

DRC (design rule check)

duty cycle

dynamic length mismatch

E

ECL (emitter coupled logic)

ECO (engineering change order)

electrical net

edge to edge diffpair

electrical net

EMI (electromagnetic interference)

EOU (ease of use)

extrinsic skew (loading)

F

false clocking

family

fanout

fence

fix

flight time

flood

floor plan cluster

free area

frequency domain

fromto

G

gap

gap (differential pair and bundle)

gap tolerance

gate

gather point

GND (ground)

group

group set

GUI (graphic user interface)

guide

H

h-tree

hard fence

HS (high speed)

I

IBIS

image

image set

impedance

initial autorouting phase

initial via grid

input skew

interconnect

intrinsic skew (output to output)

ISI (intersymbol interference)

J

jumper layer

K

keepouts

L

large components

layer

layers panel

logical part

LVDS (low voltage differential signaling)

M

maximum noise

MCM (multi-chip module)

meander

mid-driven daisy chain topology

minimum spacing

monotonic

N

net

noise

non-monotonic

NVP (nominal velocity of propagation)

O

orphan shapes

overshoot

P

package info

pad

padstack

parallel

part to part skew

path

PBGA (plastic ball grid array)

PCB (printed circuit board))

period

phase tolerance

physical net

physical part

piggyback cluster

pin

pin-to-pin delay

placement status report

primary spacing

priority values

propagation delay

propagation velocity

protect wire

pseudo-pin

pseudo-seg

PTH (plated through hole)

push-out/pull-in

push/shove

PWR (power)

Q

quiescent line

R

region

resistors

ring-back

ring-back high

ring-back low

ringing

rise time

room

routes file

routing status report

rule precedence

ruler

rules

S

sawtooth

seedvia

select

selected net

serpentine

settling time

shape

signal integrity

skew

skew limit

slew rate

small components

SMD (surface mount device)

SMT (surface mount technology)

split point

starburst

static length mismatch

status file

step

step limit

stub length

subgate

super cluster

super piggyback cluster

T

tandem

TBD (to be determined)

Tco (clock to output valid delay)

terminal

terminator

test point

Th (signal hold time to clock input)

threshold

through-pin

time domain

time length factor

tjunction

traceback/walkback

transmission line

transmission line skew

trombone

Tsu (signal setup time to clock input)

U

uncoupled length

undershoot

V

via

via barrier

via grid

via site

Vil/Vih (voltage input low/high)

virtual pin

Vol/Voh (voltage output low/high)

Vref

Vt (threshold voltage)

W

wire

wire grid

wires file

wiring polygon

X

Y

Z

Zo

-A-

accordion

An elongation wiring pattern that runs in rectangular steps, resembling an accordion fold.

antipads

Shapes defined on power layers in pin and via padstacks. These shapes are used to check clearances on the power layers if an antipad_gap clearance (edge-to-edge distance between adjacent antipads) is specified.

-B-

balanced daisy chain topology

A type of daisy chain routing in which the net must have at least one source pin and two or more terminator pins. Loads are evenly distributed between source and terminator pins. If more than one source pin is defined, the terminator and load branches are chained back to the closest source pin and the remaining source pins are ordered as an optimal daisy chain. If a net is ordered as daisy (type balanced) and it does not meet the minimum requirements, the net is ordered as a simple optimized daisy chain.

BGA (ball grid array)

A special type of SMD that uses a round "ball" of solder on it's underside in order to solder it to the substrate. These devices have various pin pitches and counts with current packaging as defined by IPC/JEDEC reaching into the 2500+ pin range.

BBV (blind and buried vias)

A blind via is a via that starts on the outer layer, but only reaches down part of the way into the design without totally penetrating to the other side. A buried via is a via that is totally encapsulated inside the substrate, therefore having no exposure to the outer layers of the design.

boundary

A rectangular area or closed path that defines where routing or placement can occur. The design boundary encloses all features of the design. The signal boundary encloses routing. The placement boundary encloses placed components.

broadside coupled or tandem

A method of routing differential pairs on adjacent layers. In other words an over/under style of routing. Ideally the layer pairs should produce impedance matching.

buried resistor

A technique where a standard resistor is created out of materials inside the designs substrate. They are made in two different ways. The first is where they are etched out of the existing substrate as shown in the upper figure. The other is where a via hole is filled with a resistive paste and then capped with additional metal thereby creating the resistor.

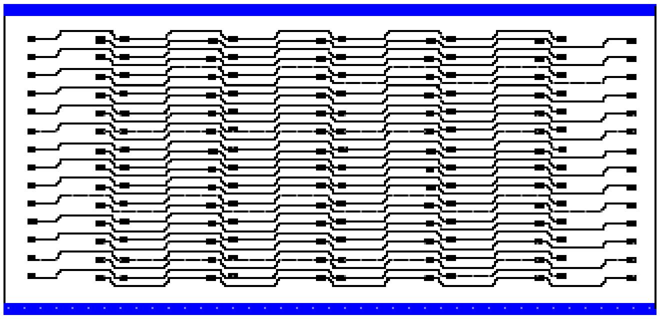
bus

A set of physical or electrical nets that should remain together during routing forming a data path through the system.

bus diagonal routing

Bus diagonal routing connects arrays of pins with diagonal wire segments, as opposed to orthogonal routing. This allows for high density routing. The following figure shows an example of bus diagonal routing.

bus diagonal routing



-C-

capacitors

A capacitor in this application is defined as a decoupling capacitor. You can assign the capacitor type property to any large or small component. Types assigned to a component take precedence over types assigned to its image.

By default, a small component is treated as a capacitor if all its pins connect to power nets and it has not been assigned the resistor or discrete type property. The autoplacer attempts to place small capacitors as close as possible to large component power pins.

circuit

A net or fromto for which electrical constraints are specified.

class

A user-defined set of nets. You define classes in the design file or by using the define class command. A net can appear in more than one class.

clearance

The distance between adjacent shapes placed within the design boundary.

cluster

A collection of components grouped together based on criteria you define.

You can define a cluster by specifying its type and the components you want to include in the cluster. Valid cluster types are floor plan, super, piggyback, and super piggyback. You can also form floor plan clusters by specifying power or signal net connections or a seed component, and letting the placer choose the components.

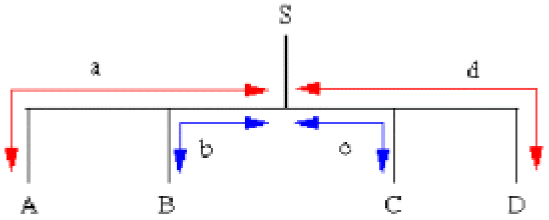
color map file

Overrides the default colors. The color map file determines the colors and fill patterns of all routing and placement objects, the background, and the design boundary.

comb

A user-defined topology that attempts to balance out placement effects with timing requirements by using a particular branching technique that appears to have a resemblance hair comb. An example is shown below.

Comb Topology



companion net

The other net in the diff pair which is modified by the program to maintain the desired diff pair spacing.

complete path

A path generated by the flood routines and via sites which extends continuously all the way from a source to a target.

component

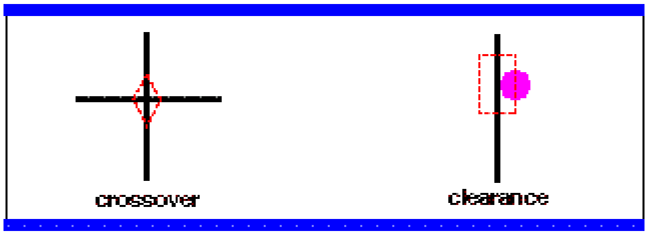
An instance of a through-pin or surface-mount package that includes the following

- Unique reference designator (component ID)
- Image name
- X,Y location
- Rotation in degrees
- Side (front or back mounting surface)

conflicts

Conflicts are crossover and clearance violations.

Conflicts



Conflicts are marked graphically with a diamond or conflict box.

Other violations include the following:

- Length rule violation is marked as a yellow dashed line
- Crosstalk violation is marked as a white rectangle
- Placement violation is marked as a thick white rectangle with diamond symbols on each corner

converge autorouting phase

The converge autorouting phase consists of the sixth and subsequent routing passes. The goal of the converge phase is to route the design completely.

Only connections that are involved with conflicts are ripped up and rerouted. You should not see unroutes during this stage. If unroutes exist, they probably cannot be completed in subsequent passes.

In the converge phase, conflict reduction proceeds at a slower pace. You typically see small percentage reductions of conflicts (less than 30%) during each of the converge passes. The overall trend during converge passes should be downward for any ten passes.

cost

A measure of the "goodness" (or, to be more accurate, "badness") of something. Reducing the cost to a single number allows a choice to be made between alternatives - pick the one with the smaller cost. Cost generally includes two factors: the quality of the proposed path (length, number of corners, via site count, adherence to design rules); and the likelihood of using that path as part of the best complete path.

coupled

Refers to segment or part of a segment that is within the band defined by primary spacing +/- tolerance.

coupling event

The point at which a segment or part of a segment passes from uncoupled to coupled status.

coupling tolerance

The allowed range that a differential pair can deviate from the primary separation gap while still be considered coupled. This value is expressed as a +/- range.

crosstalk

A condition where electrical energy is coupled from one transmission line couples into another transmission line that is physically nearby. It is produced by the mutual capacitance and inductance between the transmission lines. When severe enough it can cause false signally on components that may result in an unstable design. The one creating crosstalk is called an aggressor, and the one receiving it is called a victim. Often, a net is both an aggressor and a victim.

-D-

daisy

A net ordering method that permits only a single entry and single exit in the net on each pin and does not allow tjunctions, unless a max_stub rule is specified.

delay skew

The difference between the propagation delay on the fastest and slowest pairs in a system bus or group. For example in Ethernet cable, some cable construction employs different types of insulation materials on different pairs. This effect, in addition to unique twist ratios per pair, contributes to skew within the cable.

Skew is important because several high speed networking technologies, notably Gigabit Ethernet, uses all four pairs in the cable. If the delay on one or more pairs is significantly different from any other, then signals sent at the same time from one end of the cable may arrive at significantly different times at the receiver. While receivers are designed to accommodate some slight variations in delay, a large skew will make it impossible to recombine the original signal.

design file

A text file used as input to the autorouter. This file is created from your host layout system database by a translator that extracts the net list, component data, and rules information. The design file defines the design's size, components, netlist, design rules, preroutes, and pin and via definitions.

DFM (design for manufacture)

A method of design that tries to accommodate the needs of manufacturing thereby reducing cost and easing the product build process.

did file

Contains the commands that were executed during a routing or placement session. The default filename is the month, day, and time with a .did extension. For example, if you started a session on July 28, 1996 at 9:00 am., the default did file name would be 0728960900.did.

You can specify the name of the did file when you start the autorouter or by using the did_file command.

By default, the did file is placed in the design directory. You can change the location of the did file by using the -did switch on the command line or by specifying the path of the did file in the Startup dialog box.

differential pair

A unique topology whereby two nets are routed in close proximity to each other utilizing the self coupling effects of the copper interconnect. They are however, more difficult to route due to unique topology which requires twice as much routing space as a single standard net. The figure below illustrates a trace segment from a pair. You can see that the trace segments require 2 track channels instead of one to complete the required topology.

discretes

A discrete can be any component that you want to treat separately from other components. You can assign the discrete type property to any large or small component or image. Types assigned to a component take precedence over types assigned to its image.

do file

A text file that contains a sequence of autorouter commands. Think of the do file as a script that controls the autorouter. An example of a do file follows.

```
# Lines beginning with '#' are comments
# General purpose do file
# Initial Commands
bestsave on bestsave.wre
status_file route.sts
unit mil
grid smart (wire 1) (via 1)
# Standard Routing Commands
smart_route
```

The order of commands in a do file is very important because the autorouter executes each command in sequence. For example, you would not want to route the design before you set rules, such as clearance and width rules, that you want the autorouter to follow.

DRC (design rule check)

A process that is run on a design to ensure that it meets all required design rules. These include both electrical as well as physical requirements.

duty cycle

The difference between low-to-high and high-to-low propagation delay times when a single input causes one or more outputs to switch.

dynamic length mismatch

Checking the differential pair for length tolerance mismatches at each coupling event. Continuously monitors the user's path for phase mismatches.

-E-

ECL (emitter coupled logic)

This is a logic family that uses an "inverted voltage" scheme in order to obtain very high speed with sharp signal edges.

ECO (engineering change order)

A document used to describe a change that must be made to a design in order to guarantee functionality.

edge coupled

The standard style of differential pairs where the pair is routed on the same layer side by side.

edge to edge diffpair

A differential pair routed side by side on the same layers.

electrical net

A net that is made up of two or more physical nets that span a physical device. This is typically found in a series terminated line between two devices.

EMI (electromagnetic interference)

Results from the antenna properties of a transmission line (such as a cable, route, or package pin). Printed-circuit boards, integrated circuits, and many cables emit and are susceptible to EMI. Maximum emission levels are set by the FCC for certain frequencies (such as those used by aircraft controllers).

EOU (ease of use)

A term used to signify that a particular feature should be implemented to make a task easier for a user to perform. These techniques are usually applied to repetitive tasks.

extrinsic skew (loading)

Skew that occurs because of loading differences to the outputs of a clock driver. These can be attributed to path-to-path differences between clock trace and adjacent traces, vias, IC leads, and other signal/power planes.

-F-

fall time

The time required for a signal to change from a logic high state to a logic low state.

false clocking

A clock changing state when it crosses threshold (somewhere between VIL and VIH) unintentionally, due to excessive overshoot, undershoot or crosstalk.

family

An image family consists of one or more images assigned the same family name image property value. You can assign family-to-family spacing rules that apply to all the images of a family. An image can belong to more than one family.

Family-to-family spacing rules provide separate minimum spacing requirements between body edges, between pad edges, or from body edge to pad edge, for members of the same or different image families.

fanout

Generally, a connection made from an SMD pad to an escape via or through-pin using a short escape wire. An escape via can also be attached directly under an SMD pad. Fanouts provide access for SMD's to connections on other layers, and for through-pins assigned the expose attribute, to internal layers of the design. You can generate fanouts either during autorouting or as a pre-routing operation.

fence

A route keepin area. You can define a hard fence or soft fence.

fix

Fixing allows you to isolate nets so the autorouter can't move any part of the net (same as protect) and can't route to any point on the net. If part of the net was routed before it was fixed, that part is treated as a keepout.

flight time

The time difference between the signal at the driver reaching Vref with a reference/test load and the signal at the receiver reaching Vref. Flight time is also known as bus loss, since it historically was used to de-rate the spec Tco timing to account for the difference between the spec load and the actual system load impact on circuit timing.

flood

The first part of the routing process for a particular interconnect, in which the router begins at each end of the interconnect and steps towards the opposing end in a double ended search algorithm or from the source to target in a single ended algorithm.

floor plan cluster

A group of components, all of which you want to place either inside or outside a room.

free area

An object used to mark and fill an explored and accessible area of the design during the flood process.

frequency domain

A spectrum analyzer view of a waveform. It is used in comparing waveforms to FCC and other EMI regulatory limits. (One way to think of this is like a radio-you listen in the time domain, but you find your favorite station in the frequency domain.)

fromto

A single pin-to-pin connection on a net. A net consists of one or more fromtos (except single pin nets). A fromto is not changed when the autorouter breaks up nets.

-G-

gap

The edge-to-edge distance between parallel or tandem (parallel on adjacent layers) wires.

gap (differential pair and bundle)

The edge-to-edge distance between the wires in a pair or bundle. The autorouter maintains the gap unless the wires must diverge because of an obstacle in the routing path.

gap tolerance

Controls the amount of "deviation" +/- from the true gap that is allowed before an error is generated or considered.

gate

A set of pins that can be swapped within a component or between components. A gate consists of all the input and output pins of a functional block.

gather point

The point defined by the first time the nets come within the primary separation tolerance. It also sometimes refers to both the gather and split points.

GND (ground)

The low power net or VSS within a design. This is generally the target voltage that digital logic tries to switch to in order to generate a low logic state.

group

Fromtos of the same net or different nets that you define as a group. The same fromto can exist in multiple groups.

group set

Groups of fromtos that you define as a set. The same group can exist in multiple group sets.

GUI (graphic user interface)

Consists of menus, icons, dialog boxes, and window elements that you use to control the software program.

guide

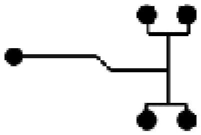
An implied connection between two points on a design. The connection points can be a combination of pins, SMD pads, and vias. A guide can connect to endpoints of wire segments, such as those created when you delete segments that have conflicts. Also called an unroute.

-H-

h-tree

A user-defined topology that attempts to balance out placement effects with timing requirements by using a particular branching technique that resembles the letter "H"

H-tree Topology



hard fence

A hard (default) fence causes the autorouter to route only connections that are completely inside the fence. If only one pin of a connection is within the fence, the connection is not routed. Connections outside the fence are not routed.

HS (high speed)

A term used to indicate that a system operates above the normal frequencies of today. These designs would then operate under a different set of design techniques and rules than a design that is not considered high-speed.

-I-

I/A (interactive)

A term used to mean that a user will interact or work with the system in some manner.

IBIS

An EIA/ANSI standard for describing an I/O buffer. It includes both DC (V/I) curves and transient (V/T) curves as tables of points. The IBIS organization home page is www.ibis.org. It has a lot of useful information about the standard as it continues to evolve.

image

A master definition of a component. All component instances are derived from an image definition. An image is defined by

- ❑ Unique name (image ID)
- ❑ Pin definition

An image can include the following

- ❑ Outline
- ❑ Side (front or back mounting surface)
- ❑ Rotation in degrees
- ❑ Associated keepouts

image set

The set of all components of a particular component or image type. The tool recognizes image sets of large, small, capacitor, resistor, and discrete components. Types assigned to a component take precedence over types assigned to its image.

- ❑ The tool classifies a component as either large or small based on pin count or assigned type property. A component with more than three pins is large. A component with three pins or less is small unless it has been assigned the large type property.
- ❑ In addition, a large or small component can be assigned the capacitor, resistor, or discrete type property. By default, the tool treats a small component as a capacitor if all its pins connect to power nets and it has not been assigned the resistor or discrete type property.

Placement rules assigned to an image set, either globally for the design or within a room, apply to all the components of that type. However, capacitor, resistor, or discrete image set rules take precedence over large or small image set rules. In addition, you can specify small capacitors, small resistors, or small discretes for exclusive processing during small component placement operations.

impedance

The ratio of input voltage to input current for a transmission line (Z0 = V/I). When a source sends a signal down a line, this is the impedance it must drive. The *Source* will not see a change in its loading *Impedance* until 2*TD, where TD is the time delay of the transmission line.

initial autorouting phase

The initial autorouting phase consists of the first five routing passes. The objective during the initial phase is to create a path for all connections by allowing conflicts and to develop the overall routing flow. The key status file indications to watch during the initial phase are fails, unroutes, and conflicts.

initial via grid

The autorouter computes an initial via grid, which might be the same as the minimum via grid if the minimum via grid is very large. The initial via grid is a multiple of the wire and via grids, and is designed to preserve routing channels by allowing two wires between vias, depending on the design rules. The initial via grid is used until the autorouter completes three routing passes or completes 50% of the routing.
Once the third routing pass or a completion rate of 50% is reached, the autorouter uses the via grid that was set in the grid smart command.

input skew

The difference between any two propagation-delay times that originate at different inputs and terminate at a single output.

interconnect

The actual metal elements that make up a connection from the source to the target.

intrinsic skew (output to output)

The difference between output edges of clock drivers that generate multiple copies from a single input clock.

ISI (intersymbol interference)

ISI refers to the interactions between the logic value/symbol from the previous switching cycle and the symbol traveling on the same channel of the current cycle. ISI occurs as a result of energy stored in the channel summing with a latter unrelated signal. It is dependent upon multi-cycle reflections and affects the rising/falling edge and settling characteristics.

-J-

jitter

The time deviation between edges of individual signals that are periodic. For example, clock jitter is the time deviation from the clock period (the clock period may be compressed or expanded). It can also affect source-synchronous circuits that have transactions spanning multiple cycles or edges, and it can also be applied to differences between rise and fall edges of a signal.

jumper layer

An imaginary layer to which you assign jumper wires. This layer is included in the normal layer stack in the design file. You can apply width and clearance rules to a jumper layer. The tool uses the actual component outline as a jumper keepout on the jumper layer. These are visible when the jumper layer is defined.

-K-

keepouts

Shapes added to a design to prevent routing or placement in specific areas. You can define keepouts to prohibit both routing and placement in an area.

You can also define specific wire keepouts, bend keepouts, elongation keepouts, via keepouts, and placement keepouts. Wire keepouts prohibit wires, but allow vias and components. Bend keepouts prohibit only wire bends. Elongation keepouts prohibit only wire elongations. Via keepouts prohibit vias, but allow wires and components. Placement keepouts prohibit components, but allow wires and vias.

-L-

large components

A large component in this application is defined as either a component with more than three pins or a component with three pins or less that has been assigned the large type property. You can assign the large type property to any component or image that you want to treat as a large component. Types assigned to a component take precedence over types assigned to its image.

layer

The autorouter uses the following layer types.

Autoroute Layers

Layer Type	**Description**
Signal	Used for routing wires
Power	Used for power distribution
Mixed	Power (plane) layer that you can use for signal routing
Jumper	Used for jumper wires
System	For internal autorouter use and to display graphics (PCB, Unroutes, and Grid are system layers)

layers panel

The Layers panel controls layer selection, layer routing direction, and layer visibility within the router user interface. To view the Layers panel, choose *View – Layers*.

logical part

A logical part is an image in your design that includes logical gate and subgate definitions. Logical parts are defined in the part library section of the design file. A logical part can have one or more instances in the design.

LVDS (low voltage differential signaling)

An acronym for Low Voltage Differential Signaling. This technique is used currently in high speed backplane and networking systems to maintain high speed and clean signaling.

-M-

manhattan length

The sum of the X and Y distances between a pin pair. The Manhattan length is the minimum wire length if a pin pair is routed orthogonally. After recornering (mitering) is done, the actual length can be less than the Manhattan length.

maximum noise

The maximum noise, controlled by a max_noise rule, that can accumulate on a net before a coupled noise violation occurs.

MCM (multi-chip module)

A ceramic substrate package that may have multiple hybrid specific components mounted within it. The MCM is then generally mounted on another substrate for incorporation into a final product.

meander

A non-optimal wiring pattern that meanders between pins in a connection. The autorouter can use a meandering pattern to add length to a connection in order to meet minimum routing length requirements, while preserving routing area that might otherwise be used up with alternative elongation patterns.

mid-driven daisy chain topology

A type of daisy chain routing in which a terminator is placed at each end of the net, and the loads are added back to a source. There must be exactly two terminators, or the net is ordered as a simple optimized daisy chain. If there is more than one source, the sources are chained together first before the rest of the net is processed.

minimum spacing

The minimum separation gap trace edge to trace edge between any set of traces.

monotonic

The phenomenon where a signal rises or falls smoothly from its high to low or low to high with no reversal of the voltage during this transition.

-N-

net

A set of pins with the same signal or voltage name. The autorouter must connect these pins with wires. Voltage can be assigned to a "power" layer. Each net is defined in the network section of the design file. Every pin of a net is identified by a component reference designator and a physical pin name.

noise

The phenomenon, where signals and switching of components generate electrical impulses that radiated off the design. These waves influence other traces around them as well as cause RF energy to be emitted. A typical source of noise is generated by the switching currents and power ground bounce caused by the high speed signal rates found on today's designs. This is where the primary emphasis from engineering of good power ground plane connections comes from.

non-monotonic

The phenomenon where the signal starts its transition, then temporarily reverses its swing and then resumes its original direction. These transition changes can cause unpredictable logic states or timing errors. They are generally caused by the crosstalk coupling of this signal's neighboring nets.

NVP (nominal velocity of propagation)

Refers to the inherent speed of signal travel relative to the speed of light in a vacuum (designated as a lower case c). NVP is expressed as a percentage of c, for example, 72%, or 0.72c. For example; CAT 5 cables will exhibit NVP values in the range of 0.6c to 0.9c. 3

-O-

orphan shapes

Copper shapes without net assignments.

overshoot

A condition where a signal transition passes above the high settling voltage level (Voh/Vih) of the circuit. It is caused by a reflected wave traveling back up the transmission line from an improperly terminated receiver. When severe enough overshoot can cause the destruction of devices due to the high peak voltages the device sees which causes the internal protection diodes to turn on, leading to early field failures.

-P-

package info

This data represents the component's body shape. It contains records for the components physical body shape, links to its pins and any alternate package information that can be used. It may also contain model, route keep outs and other mechanical information required to represent the component.

pad

A single layer copper element that represents a connection point to a component. It is generally part of a group that makes up a padstack. A component pad can be a variety of shapes and sizes including round, square, polygon etc. These shapes will all have dimensions associated with them to describe the true size of the pad.

padstack

A set of user-defined shapes that define a pin, pad, or via. These shapes can span multiple layers.

parallel

A condition where the gap between wires on the same layer is constant over some length.

part to part skew

Also known as process skew, this is the difference in output skew from package to package of the same device type.

path

A series of free areas and via sites which mark a possible path for part of a trace. The path always ends on a source or target. This results in the interconnect pattern once the routing process has been successfully concluded for this connection.

PBGA (plastic ball grid array)

A plastic body version of the BGA. See balanced daisy chain topology for a more in-depth description of these components.

PCB (printed circuit board)

Are the media in which the several different types of packaged silicon communicate with each other and the outside world. Today's designs range from 2-44 or more layers. These boards are made up of many hundreds, even thousand of electrical components all soldered onto the substrate creating an electronic product for sale to consumers.

period

For common clock circuits and multi-clock cycle transactions, period refers to a single clock or strobe cycle duration from a rising edge transition to the next rising edge transition (or falling edge to falling edge). For example, a 1GHz cycle period is 1ns duration. It is extremely useful to know the clock period when routing especially for tuning purposes. This period can be used to control the Skew between nets using realistic numbers instead of making the system work harder than necessary (for example at the minimum resolution when it is not needed).

phase tolerance

A length or delay value that defines the amount a differential pair may be mismatched in length at a coupling event.

physical net

A set of pins that form the electrical connectivity of a net.

physical part

An alternate image that lacks logical gate and subgate definitions. Physical parts map to logical parts that do include gates and subgates in their library definitions, so that a single logic definition of gates and subgates can be maintained for several equivalent library images.
Physical parts are defined in the part library section of the design file. A physical part can have one or more instances in the design.

piggyback cluster

A group of components that can overlap without violating placement rules. You must preplace and lock piggyback clusters before performing automatic placement.

pin

A terminal point that corresponds to a lead of a component. A pin is defined by

- ❑ Unique name (pin ID)
- ❑ Padstack name (padstack ID)
- ❑ X,Y location relative to the component origin
- ❑ Rotation in degrees

Pins of component instances are identified by using the component reference designator and pin ID, separated by a hyphen. For example, U2-5.

pin-to-pin delay

The time difference between the driver state change and the receiver state change. These changes are usually taken at 50% of the supply voltage. The min delay is taken when the output first crosses a defined threshold, and the max delay is taken when the output last crosses the voltage threshold, measured over all conditions.

placement status report

A report that contains a summary of placement data for the design, such as rule violations, number of placed components, Manhattan lengths, and CPU time.

For routing data, you can generate and display a routing status report.

primary spacing

The separation gap trace edge to trace edge that the user would like to maintain between the halves of a differential pair.

priority values

Priority values are assigned to nets in the range of 1 to 255. If no priority is assigned, all nets have the same default priority value of 10.

When you assign priority to multiple nets, separate each priority assignment by at least 10. If priority values are too close, the autorouter can override the priority due to other factors that determine the routing schedule.

propagation delay

All signals take some amount of time to travel or propagate down the length of the transmission line. This time is called the propagation delay of the line and is generally measured in terms of seconds/meter. Propagation delay is defined as the inverse of the propagation velocity.

Typical design delays are generally measured in nanoseconds (nS) however some of today's designs are now working in the picoseconds (pS) range. However, delay is not limited to design's or silicon. For example, in a CAT 5 cable the typical delay is less than 5 nS per meter (worst case allowed is 5.7 nS/m).

Delay is the principle reason for a length limitation in LAN cabling. In many networking applications, there is a maximum delay that can be supported without losing control of communications.

propagation velocity

The speed at which the signal travels or propagates down the length of the transmission line. It is measured in terms of meters per second. Propagation velocity is defined as the inverse of the propagation delay. Both Propagation velocity and delay are related to the time delay of the transmission line. These elements make up the majority of what a designer perceives as length matching requirements coming from engineering.

protect wire

Protecting allows you to isolate nets so the autorouter can't change any part of the net. However, the autorouter can route to a protected wire at its terminal or to a segment if tjunctions are allowed. For example, you can fanout a component, protect the fanout wires and vias, and route the design. Wires route to fanout vias when they are protected.

pseudo-pin

A vertex where three or more wire segments are connected on the same layer, or a point on a wire at which a connection to a wiring polygon is made.

pseudo-seg

A piece of a segment defined by a coupling or uncoupling event within a differential pair.

PTH (plated through hole)

PTH in the tool identifies a through-pin image or command.

push-out/pull-in

Refers to the difference in signal flight time due to signal coupling effects and signal return path discontinuities. Comparing with the delay of single-bit switching, push-out means all the drivers switching at the same direction (even mode), whereas pull-in means all the other drivers switching at the opposite direction (odd mode).

push/shove

The third part of the route process, in which space is made available for the new trace by pushing existing obstacles away from it.

PWR (power)

The supply net or VDD within a design. This is generally the target voltage that digital logic tries to switch to in order to generate a true logic state.

-Q-

quiescent line

A line that is not switching during the current clock cycle. Also called a "stuck-at" line or static line. When it is bad enough, crosstalk alone can cause a quiescent line to appear to switch during a portion of the clock cycle.

-R-

reflection

A reflection on a transmission line is a sort of an echo of the original signal. A portion of the signal power (voltage and current) transmitted down the line goes into the load, and a portion is reflected. Reflections are prevented if the load and the line have the same impedance.

region

A rectangular area of the design where you can apply routing rules. You can define a region on single or multiple signal layers.

resistors

A resistor can be any component that you want to treat separately from other components. You can assign the resistor type property to any large or small component or image. Types assigned to a component take precedence over types assigned to its image.

ring-back

When a signal rising edge crosses beyond the Vih threshold and re-crosses threshold again before settling beyond Vih. Depending upon the magnitude and duration of the re-crossing, the settling time may need to be calculated from the final crossing of Vih. This also applies to signal falling edges re-crossing Vil before settling below Vil. For a clocked signal, ring-back is typically allowed as long as the signal settles beyond the Vih/Vil threshold to satisfy the setup timing requirement.

ring-back high

Generally, a rule used to prevent a condition where the reflected energy causes the circuit to false trigger at the high voltage rail of the system. This rule allows a little ringing of the signal but does not let it pass back down thru the circuit's high threshold level. Doing so would most likely cause a false triggering of the device or a bad piece of data being present on the devices input.

ring-back low

Generally, a rule used to prevent a condition where the reflected energy causes the circuit to false trigger at the low voltage rail of the system. This rule allows a little ringing of the signal but does not let it pass back above thru the circuits low threshold level. Doing so would most likely cause a false triggering of the device or a bad piece of data being present on the devices input.

ringing

Essentially is repeated *Overshoots* and *Under-shoots*. It is what happens as all the reflections from improperly terminated transmission lines slowly start to "self-damp". The only way to get rid of this phenomenon is to properly terminate the lines to the drivers.

rise time

The time it takes for a signal to change from a logic low state to logic high state. This may also include partial transitions as well (typically specified by manufacturers as 10% ~ 90% amplitude change, or rise through specific voltage thresholds, such as 0.5V ~ 1V).

room

A rectangular or polygon-shaped area of the design that you define to control where the tool places components.

routes file

An output file generated when you use the *File – Write – Routes* menu command or the write routes keyboard command. The routes file contains the routing produced by the autorouter. Use this file to return the routing to your layout system or to restart the autorouter.

routing status report

A report that contains a summary of routing data for the design, and includes the following categories

❑ Routing status

❑ Routing history

❑ Wiring statistics

❑ Summary statistics by layer

You can see simplified routing statistics in the status file. For placement data, you can generate and display a placement status report.

rule precedence

Routing rules

The tool applies routing rules according to the following hierarchy:

pcb < layer < class < class layer < group_set < group_set layer < net < net layer < group < group layer < fromto < fromto layer < class_class < class_class layer < padstack < region < region class < region net < region class_class

A pcb rule (global rule for the design) has the lowest precedence in the hierarchy. A region class_class rule has the highest precedence. Rules set at one level of the hierarchy override conflicting rules set at lower levels.

Placement rules

The tool applies placement rules according to the following hierarchy:

pcb < image_set < image < component < super cluster < room < room_image_set < family_family < image_image

A pcb rule (global rule for the design) has the lowest precedence in the hierarchy. An image-to-image spacing rule has the highest precedence. Rules set at one level of the hierarchy override conflicting rules set at lower levels.

ruler

A graphical (horizontal or vertical) ruler that you draw anywhere in the design where precision routing or placement is needed.

rules

Geometric constraints that you assign to a net or to connections in a net. The rules are hierarchical, meaning that certain rules have precedence over others.

-S-

sawtooth

An elongation wiring pattern that runs in a diagonal pattern, resembling the teeth of a saw blade.

seedvia

A via assigned by the autorouter before routing begins if you use the seedvia command. The via is placed so that it reduces a long diagonal connection to two orthogonal connections joined by the seedvia.

select

A mechanism that lets you identify individual objects, such as wires, nets, or components, for exclusive processing by routing or placement commands. When you select wires, nets, components, or other objects before running a command, the tool operates only on the objects that you have selected.

selected net

The net of the diff pair selected by the user to being a diff pair edit. For any given edit, this could be either net of the diff pair. The user directly manipulates this net while the other (companion) net follows.

serpentine

A method used today to create delay in nets that require some form of timing adjustments to increase the time of flight for a particular signal or an entire bus. This is done to ensure that all signals arrive at close to the same time guaranteeing that a device has valid data during at the appropriate time.

session file

Contains the design filename, a history of previous session files, and component placement, floor plan, and route data from the current session.

settling time

The time required for a ringing signal to stabilize to within a specified range of its final value. This is usually seen when the signal settles above the Ring-back High or below the Ring-Back Low numbers respectively.

shape

The basic data element. Objects such as through-pins, SMD pads, vias, wires, or keepout areas are shapes. Shapes can consist of rectangles, circles, polygons, paths, and qarcs.

signal integrity

The ability of a signal to generate correct responses in a circuit. A signal with good signal integrity has digital levels at the required voltage levels at the required times. It is obtained by use or a set of electrical rules that when applied correctly shall create a design that ensures proper signal levels on all devices.

skew

The difference between two or more signals in their delay at a specified voltage threshold. For a common clock circuit, skew may be critical between a driver and receiver clock to determine setup or hold time impact. For example in a source-synchronous system this can apply to strobe vs. signal or strobe vs. strobe.

skew limit

The difference between the maximum specified values of either high-to-low or low-to-high propagation delay and the minimum values of the same.

slew rate

The signals edge rate (rate of change of a signal voltage with respect to time). 1/0 specifications (such as PCI) state the two voltages between which the slew rate is measured.

small components

A small component in this application is defined as a component with three pins or less that has not been assigned the large type property. You can assign the small type property to any component or image with three pins or less. You cannot assign the small type property to components with more than three pins. Types assigned to a component take precedence over types assigned to its image.

SMD (surface mount device)

A device that is placed on the outer layers of the design. These devices use SMT techniques to mount them on the design.

SMT (surface mount technology)

A technology where components are mounted on the outer layers of the design.

soft fence

A soft fence is useful in separating analog and digital signals. A soft fence causes the autorouter to do the following:

- ❑ All connections inside the soft fence are routed within the fence boundary.
- ❑ All connections outside the soft fence are routed outside the fence and do not cross the fence.

- ❑ All connections that cross the soft fence ignore the fence and are routed.

source/target

The source is one end of the connection, the target the other. Which end is source and which target is purely a matter of perspective; we give them different names for convenience in discussing the flood process.

The source/target may be a single object (e.g. a pin) or a collection of them (e.g. a tree of already-routed interconnects to which an additional interconnect is to be joined); but for purposes of costing, a single source or target is usually considered at any one time.

split point

Point defined by the last time the nets go outside primary spacing +/- tolerance creating an uncoupling event. This is also considered a gather point from the other direction.

starburst

A net ordering method that uses a minimum spanning tree algorithm and permits multiple entries and exits on pins.

static length mismatch

Checking the differential pair for length tolerance for a mismatch only on the overall lengths once the net is completely routed.

status file

A file that contains simplified routing statistics. Used to monitor the autorouting session. The routing statistics are saved in a default file, monitor.sts, after every 100 wires are routed or at the end of a pass, whichever occurs first.

You can control how frequently the autorouter status file updates. You can rename or redirect the status file when you start a session or anytime during the session.

To see routing statistics and additional routing data, you can generate a routing status report. For placement data, you can generate a placement status report.

step

One step of the flood, in which a small region of the design is explored for obstacles and free areas or via sites created to fill the accessible space.

step limit

A limit on the distance that can be covered by any single step.

stub length

the maximum distance between a terminal point and a wire tee junction. Stubs are generally controlled with a rule that is applied primarily control daisy chain connections.

subgate

A set of pins that can be swapped only within a gate. A subgate usually consists of only a subset of the input pins in a functional block.

super cluster

A group of components whose positions and rotations are fixed with respect to each other, forming in effect a single super component.

super piggyback cluster

A group of components, with fixed positions and rotations, that can overlap without violating placement rules. A super piggyback cluster is, in effect, a single super component.

-T-

tandem

A condition where the gap between wires on adjacent layers (instead of the same layer) is constant over some length. Tandem conditions are not calculated on mixed layers nor are they calculated between two signal layers that are separated by a power (plane) layer.

TBD (to be determined)

A Term used to signify that no decision or data exists for a particular item at this time. This item will be finished at some point in the future.

Tco (clock to output valid delay)

The delay between component clock input (at a specified input voltage threshold) and a valid signal output (at a specified reference load and output voltage threshold). This delay for system design is typically specified at component package pins or input/output pads.

terminal

A point at which a wire can connect. A terminal exists at a through-pin, SMD pad, via, tjunction, pseudo-pin, or wiring polygon.

terminator

A pin that is assigned the terminator property in the design file or in the tool. In Routing mode, you can assign the terminator property by using Define - Pin Attributes or by using the assign_pin command.

test point

A pin or via assigned to each net. The test point is used for manufacturing tests of the design. Test points must not be covered by any components and can be on a specified test grid.

Th (signal hold time to clock input)

This is the time required for the input signal to remain valid (above Vih for rising and below Vil for falling) beyond the input clock edge transition of the receiving component. Hold time is used both at receiving components for common clock and source-synchronous timing.

threshold

The maximum parallel or tandem length that can be ignored for coupled noise calculations.

through-pin

A term used to identify components that have through-pins that extend through all the layers of the design.

time domain

An oscilloscope view of a waveform. It is used for finding pin-to-pin delays, skew, overshoot and undershoot, and settling times.

time length factor

The factor, controlled by the `time_length_factor` rule, that converts the time units used in delay rules to units of length. The time length factor is a ratio of time to a unit of length and is used as a multiplier to calculate effective wire lengths from time delays. You must provide a time length factor if you are using delay rules in your design.

tjunction

An intersection of three wire segments that belong to the same net.

traceback/walkback

The second part of the interconnect routing process in which the best complete path found during the flood is analyzed and a series of trace segments and vias are inserted into the database/memory map within the constraints of that path.

transmission line

Any net (wire) AND its current return path to ground or a power supply.

transmission line skew

Skew occurring because of trace propagation differences attributed to improper layout and manufacturing tolerances involving etch, design thickness and dielectric constant.

trombone

An elongation wiring pattern that folds back against itself, resembling the slide of a trombone.

Tsu (signal setup time to clock input)

This is the time required for the input signal to be settled about Vih (rising) or below Vil (falling) at the receiving component before its input clock edge transition. Setup time is used both at receiving components for common clock and source-synchronous timing.

-U-

uncoupled length

The accumulated length of pseudo-segments that lie outside the user defined coupled band.

undershoot

A condition where a signal transition passes below the low settling voltage level (Vol/Vil) of the circuit. It is caused by a reflected wave traveling back up the transmission line from an improperly terminated receiver. When severe enough undershoot can cause the destruction of devices due to the high inverse voltages the device sees or false clocking and data errors.

-V-

velocity of propagation

Measured in terms of meters per second and is dependent on the dielectric material surrounding the interconnect pattern.

via

A shape that interconnects two or more layers.

via barrier

A via barrier occurs when vias are placed so that wires cannot pass between them. If the via grid is too fine, the autorouter can create a via barrier of fanout vias shown in (a). The following example (b), shows how you can allow routing between pads without producing conflicts by having a small wire grid and a larger via grid.

via grid

A set of equidistant points in the X and Y direction on which vias can be inserted. You can specify different X and Y grid increments and you can specify offsets.
If you define a grid of zero, the via grid is calculated internally.

via site

A quadrilateral used to mark a region of the design which has been found, during the flood to be a good place to pop the via.

Vil/Vih (voltage input low/high)

Vil and Vih refer respectively to the maximum low input voltage for a high to low input transition and minimum high input voltage for a low to high input transition. The input signal needs to remain stable beyond these voltage limits to be guaranteed latched in.

virtual pin

A terminal you use to define a fromto tree or other topology. You can use virtual pins to control delays by matching routing lengths (such as minimizing clock skew) without adding excessive wiring on each branch of a net.

Vol/Voh (voltage output low/high)

Vol and Voh are the low and high, respectively, voltage levels guaranteed at the driver output reference point for the driven signal.

Vref

A reference voltage value used to make some particular measurement.

Vt (threshold voltage)

Vt refers to the input threshold voltage which determines whether a high or low state is sensed at the receiver input. In some cases, an input threshold is specified with an additional noise margin or overdriven region specified for timing specification or signal condition requirements.

-W-

weight

A value equal to the amount of noise (usually in millivolts) per unit of parallel or tandem wire length that is coupled from a transmitting net onto a receiving net. The weight value is transmitted when the distance between parallel and tandem wires is less than the associated gap value.

wire

A physical connection between two terminals.

wire grid

A set of equidistant points in the X and Y dimensions on which the center lines of a wire must be routed. You can specify different X and Y grid increments, and you can specify offsets.

If you define a grid of zero, the via grid is calculated internally.

wires file

An output file generated when you use the *File - Write - Wires* menu command or the write wires keyboard command. It contains routing information. The wires file is used when you restart the autorouter and you want to use the routing information from a previous autorouting session.

wiring polygon

A conducting area that is not a wire segment or via.

-X-

-Y-

-Z-

Zo

The characteristic impedance of a transmission line. For more information see impedance.