SPECCTRA® Tutorial

Product Version 10.0
May 2000

Cadence Design Systems, Inc., 555 River Oaks Parkway, San Jose, CA 95134, USA

Table of Contents

About This Book

This Tutorial consists of this book and several accompanying files. The tutorial files are copied to the *tutorial* directory when you install the tool on your computer or network.

The autorouter runs on Windows NT and on many UNIX platforms. Separate instructions are provided for Windows and UNIX platforms when steps or procedures differ between these platforms. The majority of the screen shots in this tutorial are from the Windows environment unless otherwise noted.

Audience

This tutorial is written for users who are familiar with the current methods and practices used to design printed circuit boards.

Conventions Used in This Tutorial

The following fonts, characters, and styles are used to identify or represent different types of information.

- **Boldface** type identifies text that you type exactly as shown, such as commands, keywords, and other syntax elements. In the following example, connect, on, and off are keywords.

  (**connect** [**on | off**])

  Syntax examples and command examples that are not entered by you from the keyboard are not bold. For example:

  (boundary (rect pcb 0 0 9000 4000))

- *Italic* type identifies titles of books and emphasizes portions of text. For example:

  > See the *Installation and Configuration Guide* for information about installing the autorouter.

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

  > <*directory_path_name*>

- `Courier` type identifies prompts, messages, and other output text that appears on your screen. For example, if you misspell the command *define* as *defin*, an error message displays in the output window.

  ```
  ‘Syntax error in command: token 1 =
  defin’
  ```

  Courier type also identifies operating system commands and switches.

  Courier type enclosed in brackets ([ ]) identifies keys on your keyboard and mouse buttons.

  For example, `[Shift]` means the shift key. The carriage return key is labeled Enter on some keyboards and Return on others. This manual uses `[Enter]`.

  Mouse buttons are identified by two uppercase letters enclosed in brackets.

  `[LB]` left button
  `[MB]` middle button
  `[RB]` right button

  If you have a 2-button mouse, press `[Alt] and [RB]` simultaneously when you see `[MB]`.

If a task requires a series of steps that are different on different computer platforms, separate procedures are provided. For example:

Start the tool by using the Startup dialog box (Windows)

1. Click Start on the task bar, slide the pointer to Programs, and choose SPECCTRA.

   .

   .

Start the tool by using the Startup dialog box (UNIX)

1. Change your current directory to the SPECCTRA tutorial directory.

   .

   .

Special Terms

The following special terms are used in this manual.

- The word *enter* used with commands means type the command and press `[Enter]`. For example:

  "Enter the command **grid wire 1**" means

  1. Type **grid wire 1**.

  2. Press `[Enter]`.

- *Click* means press and release the left mouse button.
- *Click-middle* means press and release the middle mouse button.
- *Click-right* means press and release the right mouse button.
- *Drag* means press and hold the left mouse button while you move the pointer.
- *Drag* `[MB]`means press and hold the middle mouse button while you move the pointer.
- *Double-click* means press and release the left mouse button twice in rapid succession.

- *Click twice* means click two times at the same location in the work area.
- *Switch* refers to one or more characters preceded by a dash (-). You can use a switch when you start SPECCTRA.

Where to Find Additional Information

For information about the tool, and an overview of new features and enhancements, see the document entitled *What's New in 10.0*.

For installation information, see the *SPECCTRA Installation and Configuration Guide*.

For descriptions of known problems and limitations in SPECCTRA Version 10.0, refer to the *Release Notes*.

If you need information about design file syntax, use Acrobat™ Reader from Adobe Systems, Incorporated to view the *SPECCTRA Design Language Reference* manual.

For information about starting the autorouter, command syntax, menu help, and design methodology, choose Contents from the SPECCTRA Help menu and select from the list of topics.

For information about starting the tool, command syntax, menu help, and design methodology, open the online help file *SPECCTRA.hlp* in your help viewer (WinHelp for Windows 95/98/2000 and Windows NT or HyperHelp for UNIX) and choose from the list of topics. You can also access online help from the SPECCTRA Help menu.

Your Comments About This Tutorial

We are interested in your comments and opinions about this tutorial. Please consider the following questions when you form your comments.

- Is the information organized logically? If your answer is no, how could we better organize the information?
- Did you find any inaccuracies or omissions? If your answer is yes, what are the inacurracies or omissions? Please include page numbers.
- What suggestions do you have for improving this tutorial?

Please send your comments via the Internet by email to **cct_pubs@cadence.com**. Remember to include the document title with your comments.

How to Contact Technical Support

If you have questions about installing or using SPECCTRA, you can visit the Cadence Web site at

http://sourcelink.cadence.com /supportcontacts.htm

Introducing the Autorouter

This tutorial teaches you how to use autorouter to place and route printed circuit board designs.

What Your Prior Experience Should Be

The Tutorial is for printed circuit board designers who know current design methods and practices but have little or no experience using the autorouter. You are expected to know the common methods for starting an application, using the mouse, and opening and closing windows on a Windows or UNIX platform.

What You Will Learn

Each lesson in this tutorial covers a set of topics that are important to understanding the basic use and operation of the routing and placement tools. The tutorial includes this introductory chapter and five lessons that cover

- Basic concepts
- Placing components
- Autorouting
- Setting rules and controlling the autorouter
- Interactively routing and editing

How to Use This Tutorial

This tutorial is designed as a step-by-step process for learning how to use autorouter. The information you learn in a lesson builds upon the previous lessons. However, you can do the lessons as separate tutorials.

Lesson	Teaches you to...
Lesson 1, Learning Basic Components	Use the autorouter’s graphical user interface (GUI)
Lesson 2, Placing Components	Place components automatically and interactively
Lesson 3, Autorouting a PCB Design	Automatically route a PCB design
Lesson 4, Setting Rules and Controlling the Autorouter	Set rules and routing options to control the autorouter
Lesson 5, Interactive Routing and Editing	Route interactively and edit the routing

You must have the Placement option to complete the work in Lesson 2. If you are only interested in learning about interactive routing, you can skip the first three lessons, but you need to complete Lesson 4. This lesson includes information on setting rules, which is relevant to interactive routing and editing. You must have the EditRoute option to complete the work in Lesson 5.

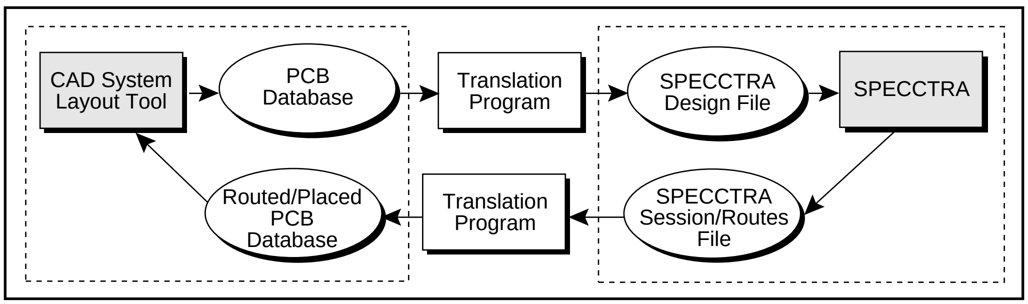
This book accompanies a series of lesson files that are supplied with the autorouter. The *tutorial* directory contains the lesson files, which are installed with the software. You use the files with the software to learn by doing.

If you are using the AR2U (two layer) version of the autorouter, use the lesson files in the *tutorial\ar2u files* directory. The files in this directory are modified. If you copy the files from the AR2U files directory to the tutorial directory, the instructions to load files in the lessons work as written and the lesson icons in the autorouter group link to the AR2U files.

How the Autorouter Fits in the PCB Design Process

The autorouter extends your PCB CAD system by adding automatic and interactive placement and routing tools. You use the autorouter to place components and route your printed circuit board design.

After you create a PCB database in your layout system, you translate the design data to a design file. You place and route the design in the autorouter, save the results, and then merge the placement and routing data with your original PCB database.



The Autorouter in the PCB Design Process

Transferring Designs Between the Autorouter and the Layout System

Each PCB layout system stores design information in a unique format, but the files used by the autorouter to store design data are the same regardless of your layout system.

You transfer your printed circuit board design between the autorouter and the layout system by translating the design data from one format to the other. All files that are read and written by the autorouter are plain text files, which are described in the following table.

File Type	**Naming Convention**	**Description**
design	<filename>.dsn	Created by translating design information from the layout system. Contains PCB boundary data, layer definitions, padstack definitions, component data, netlist, preroutes, and design rules.
session	<filename>.ses	Created by the autorouter. Contains a pointer to the original design file, placement and route data, gate, subgate, pin, and terminator information.
routes	<filename>.rte	Created by the autorouter. Contains route data that can be translated to your layout system and read by the autorouter.
wires	<filename>.w	Created by the autorouter. Contains route data that can be read by the autorouter only.

Some layout systems require intermediate text files to transfer a design to and from the autorouter. Other systems read and write the autorouter's files directly without intermediate files. The files that are needed to transfer designs between the autorouter and several popular layout systems are described in the following table.

Layout System	**Intermediate File**	**What's in the File**
Allegro	board (brd) file	All PCB design data, including nets, properties, components, padstacks, preroutes, PCB boundary, and rules
Board Station	tech.tech	Layer definitions and rules
	geoms_ascii	Image definitions, PCB outline, keepout and keepin areas, and rules
	nets.nets	Netlist
	traces.traces (Optional)	Preroutes, area fill information, and high speed topology specifications
	gates.gates (Optional)	Gate and pin swap information
	pins.pins (Optional)	Properties attached to pins
	testpoints.testpoints (Optional)	Testpoint information
	mfg/neutral_file (Optional)	Pin X,Y coordinates

Layout System	Intermediate File	What's in the File
PADS	ASCII output	All PCB design data, including nets, components, padstacks, preroutes, PCB boundary, and rules
PCAD	PDIF	All PCB design data, including nets, components, padstacks, preroutes, PCB boundary, and rules
Protel	Protel text file	Writes the autorouting design file and reads routes and session files directly

Many PCB layout systems have built-in features to transfer designs to and from the autorouter. Some layout systems include a choice on a menu or a separate GUI to simplify the transfer process. Refer to the documentation for your layout system or the documentation that was included with your autorouter translator to determine how to transfer designs between your layout system and the autorouter.

Understanding the Autorouting Design File

The autorouting design file is a text file that contains the information needed to represent a printed circuit board in the autorouter. The PCB outline, layers, components, padstacks, nets, and preroutes are represented in the design file in five sections. The five sections of the design file are described in the following table.

Design file section	Information it includes
structure	Working units, layer definitions, PCB boundary, power planes, region rules, PCB keepouts, via ids, global rules, grid definitions
placement	Component instances that consist of image names, reference designators, X,Y locations, PCB side, and rotation
library	Image definitions that include pin names and pin locations, pin definitions, and padstack definitions
network	Net list (net names and pin lists), class definitions, class to class definitions, group definitions, differential pair definitions, and net, class, or group rules
wiring	Preroute information

Because the autorouting design file is a text file, you can view it in a report window by using **Report > Design** or in a text editor.

Caution: Do not edit a design file in a text editor. Most translators use the design file to merge the routing data with the original layout system database. If you change the design file and it's no longer synchronized with the layout system database, the translation of route data to your layout system could fail. If you need to make a change to the design data, make the change in your layout system and translate the revised design to the autorouting design file.

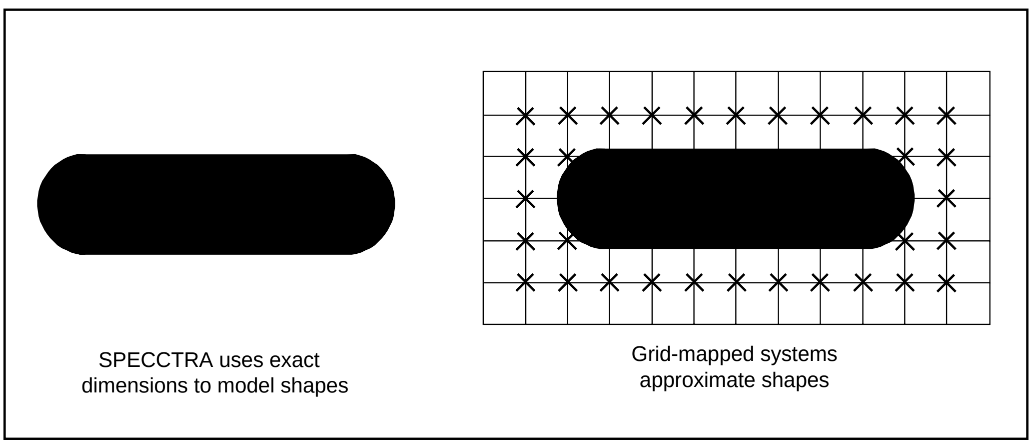
Using a text editor, you can search for keywords. In UNIX, you can use vi, emacs, or textedit. In Windows, you can use Notepad or Write. If the design file is too large for Notepad, use Write with the *no conversion* option.

Note: In Windows, you cannot view a design file in a text editor while the file is loaded in the autorouter. You can copy and rename the design file, then view the design information while the design is loaded in the autorouter.

Understanding Autorouter's ShapeBased™ Technology

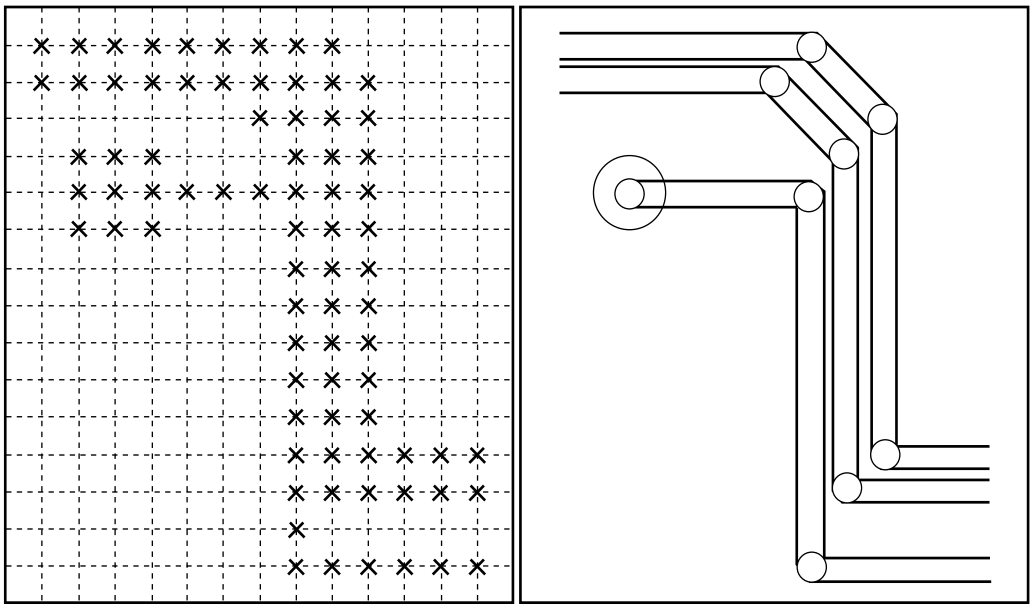
The autorouter succeeds in routing large, dense designs because of its ShapeBased technology. The autorouter differs from traditional grid-mapped systems because it models pins, pads, wires, and vias as true shapes. Grid-mapped systems define these shapes as grid points. Each pin, pad, wire, and via is defined in terms of the grid points it occupies.

The following figure shows the basic difference between the ShapeBased system and a grid-mapped approach.



Grid-Mapped Modeling of Objects Wastes Space

While grid-mapped modeling wastes space, its greater weaknesses are its excessive memory and storage requirements. The autorouter's ShapeBased approach only requires memory for storing shapes not grid points. The following figure illustrates the difference between ShapeBased and grid-mapped memory requirements.



Grid-mapped system must store 78 grid points to model 12 shapes

ShapeBased system stores 12 shapes

ShapeBased Technology Requires Less Memory than Grid-Mapped Technology

Another advantage of autorouter's ShapeBased technology is its support of complex design rules. Each shape on each layer inherits its own unique set of design rules. This means you can comply with the most complicated design requirements without resorting to tricks and work-arounds during placement and routing.

Lesson 1: Learning Basic Concepts

What You Will Learn

This lesson explains the autorouter user interface and how to perform basic design tasks. You will learn how to

- Start the autorouter and load a design
- Use the graphical user interface (GUI) for commands
- Pan and zoom
- Measure distance and get information about design objects
- Save your work
- Review a session's command history

This lesson takes about 45 minutes to complete.

What to do Before You Begin

Before you begin this lesson

- Read the chapter "About This Book," which explains the conventions, terminology, and symbols that are used throughout this tutorial.
- Make sure that the autorouter is installed properly on your computer or network.
- Make sure the tutorial files are installed in the tutorial directory.

Starting the Autorouter and Loading a Design

When you start the autorouter, you must load a design file. The design file is a text file that contains all the PCB design information that is needed by the autorouter.

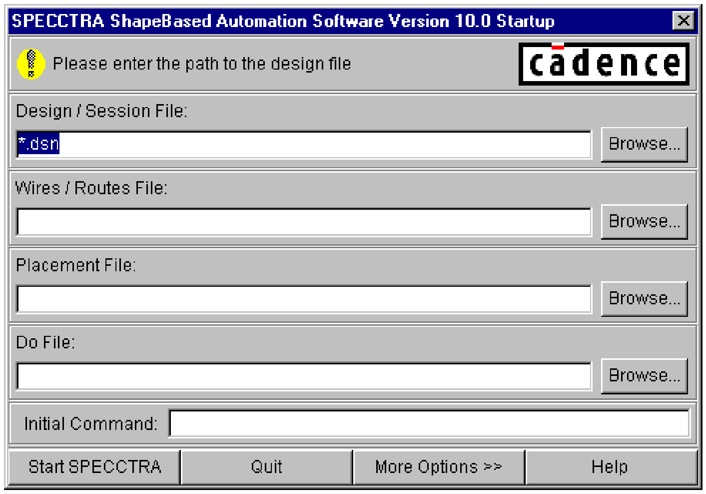
Note: You can start the autorouter using a session file, which links the routing and placement data to a design file. You will learn more about session files later.

In this lesson, you use the autorouting Startup dialog box to start the autorouter and load a design file. The procedure varies depending on whether you are using a Windows or UNIX platform. In this section, use the procedure for your platform. You will use these same basic steps frequently throughout this tutorial. If a procedure is not labeled Windows or UNIX, it is written for both Windows and UNIX.

Start the autorouter using the Startup dialog box (Windows)

1. Click the Windows Start button and click **Programs > SPECCTRA V10.0 > SPECCTRA V10.0**.

   The Startup dialog box appears.



   You need to specify a design file to start the autorouter. You can type the name of the design file in the Design/Session File data entry box or you can use the Browse button to search for a file.

2. Click the *Browse* button to the right of the Design/Session File data entry box.

   The Open dialog box appears.

3. Change to the autorouter's tutorial directory.
4. Open *lesson1.dsn* from the *Files* list.

   The path and filename appear in the Design/Session File data entry box.

5. Click *Start SPECCTRA*.

   The autorouter starts and loads the design.

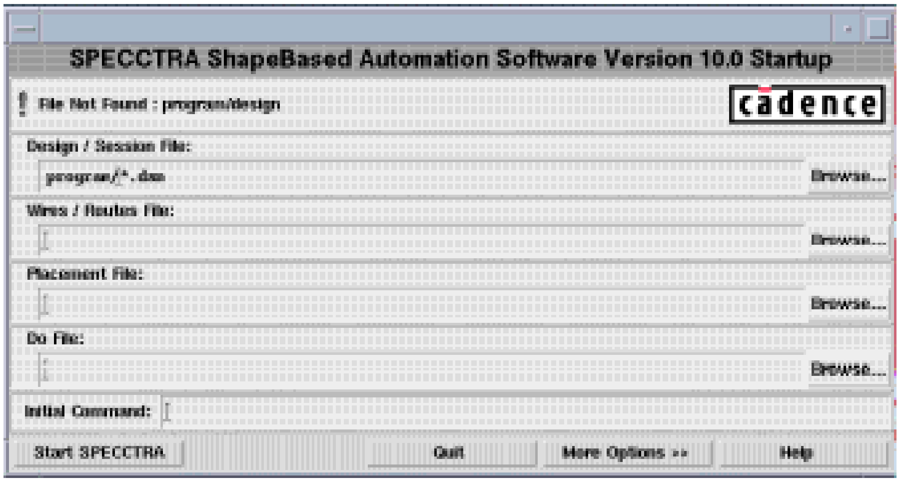
6. Do not exit the autorouter. Continue to the next section on “Controlling the Autorouter.”

   You will exit the autorouter later in this lesson.

Start the Autorouter using the Startup dialog box (UNIX)

1. Change to the autorouter’s tutorial directory.
2. Enter **specctra** at the shell prompt.

   The Startup dialog box opens.



3. Click the *Browse* button to the right of the Design/Session File data entry box.
4. The Select File dialog box opens.
5. Select *lesson1.dsn* from the files directory and click *OK*.
6. Click *Start SPECCTRA*.

   The autorouter starts and loads the design.
7. Do not exit the autorouter. Continue to the next section on “Controlling the Autorouter.”

In the following sections, you will learn how to use the GUI to control the autorouter, and how to autoroute a design, monitor progress, and examine the results.

Controlling the Autorouter

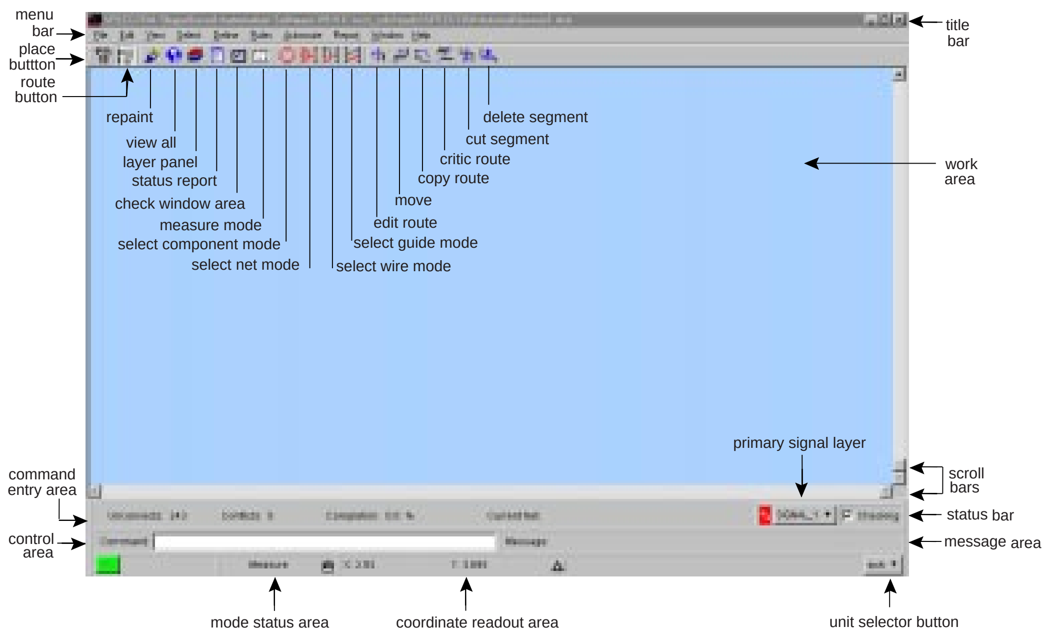
You control the autorouter by

- Entering commands from the keyboard
- Running a command file (do file)
- Using the GUI to execute commands

In a later lesson, you will enter commands from the keyboard and use a do file to control the autorouter. In this lesson you use the GUI.

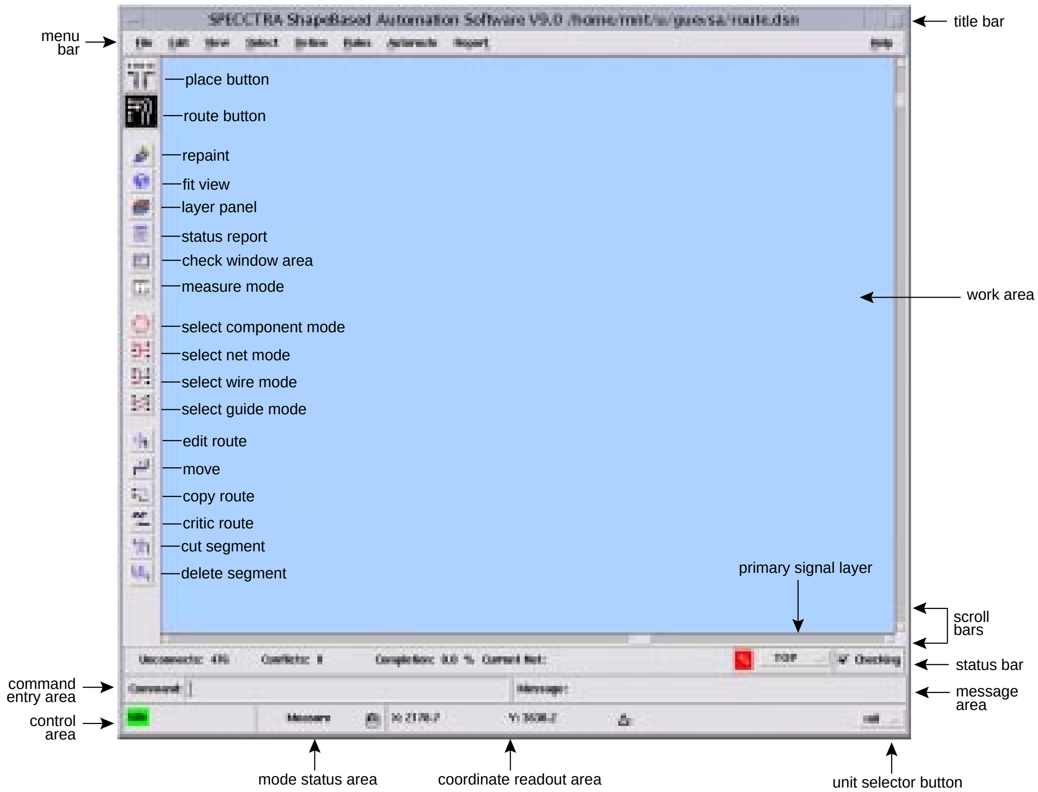
Using the graphical user interface (GUI)

The GUI consists of screen elements such as menus, tool bar buttons, dialog boxes, status bar, and message area. These and other important elements are labeled in the following figures.



The Autorouter GUI (Windows)

.

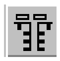


The Autorouter GUI (UNIX)

.

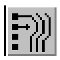
The last two figures show the GUI for the routing application mode. A slightly different GUI displays in the placement application mode. Routing is the default application mode that you see after you start the autorouter. The outing application menu bar and status bar are specific to autorouting. To switch to the placement application mode, click the *place* button.

Switch between routing and placement modes



1. Click the *place* button on the tool bar.

   The tool bar and status bar change. You will learn about placement in Lesson 2. For now, you will work in routing mode.



2. Click the *route* button on the tool bar.

Controlling the autorouter with the GUI

The easiest way to control the autorouter is to use the GUI. If you use the GUI, you do not need to know command syntax. You only need to know the task you want to complete.

In the following procedure, you use the GUI to autoroute the printed circuit board you loaded at the start of this lesson. You will see how easy it is to automatically route a design using the GUI.

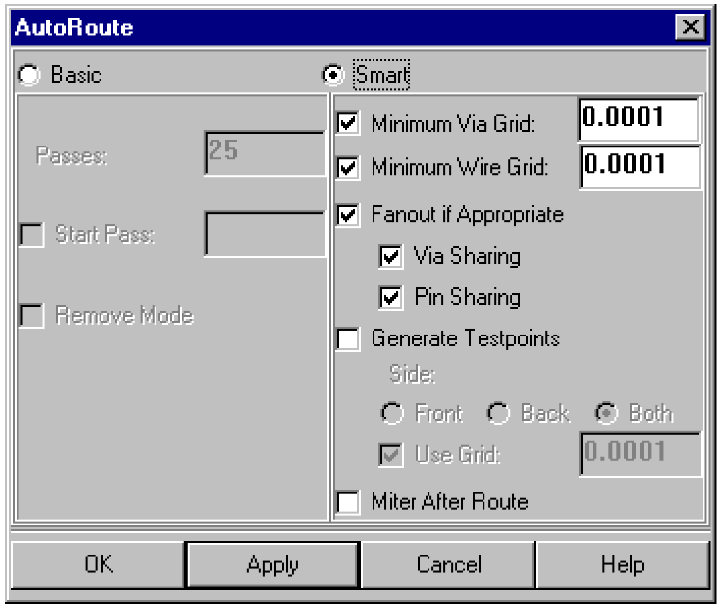
Use the GUI to autoroute a printed circuit board

1. Click **Autoroute > Route**.

   (This instruction means move the pointer over **Autoroute** on the menu bar, click the left mouse button to pull down the menu, and click *Route*.)

   The AutoRoute dialog box opens.

2. Study the default settings in the AutoRoute dialog box, and notice that the *Smart* option is set.



The Smart option uses the **smart_route** command. This option analyzes the design, performs calculations, starts the autorouter, and continues to apply routing passes until all connections are routed, or further improvement is unlikely.

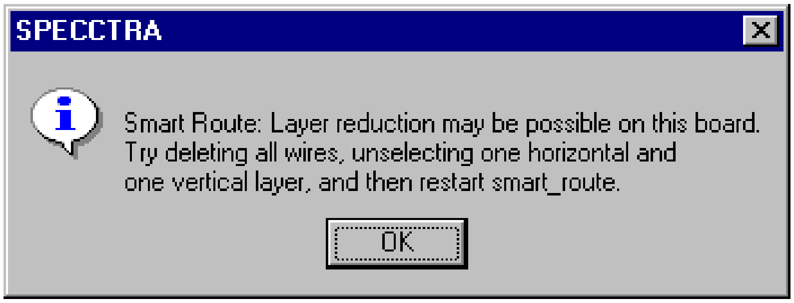
3. Click *OK*.

   The autorouter begins routing the design. The Pause button appears.

   While the design is routing, a *Pause* button appears in the control area. You can use this button to pause or stop smart_route. When the smart_route command finishes, the *Pause* button in the control area is replaced by *Idle*.

Note: You might get a message warning you that the design might converge slowly and to monitor the status file. Click *OK* to close the dialog box.

The following dialog box appears to inform you that layer reduction is possible.



Smart_route determines whether layer reduction is possible. If you were routing your own design, you might follow the suggestion and reduce routing layers. In this lesson, you will ignore the suggestion.

5. Click *OK* to close the message.
6. Wait for smart_route to complete.
7. You view the transcript of session information in the Output window. On UNIX platforms, the Output window is the shell in which you started the autorouter.
8. Use the Output window scroll bars to browse the information.

Information in the Output window includes the version number and routing results. When warnings or error messages are generated, they also appear in the Output window. A portion of a typical session transcript follows.

```
=====================================================
#        SPECCTRA ShapeBased Automation Software
# Copyright 1990-2000 Cadence Designs Systems, Inc.
# All Rights Reserved.
# ===================================================
#
# Software licensed for sale by Cadence Design Systems,
  Inc.
# Current time = Tue Aug 24 16:11:15 2000
#
```

```
# SPECCTRA ShapeBased Automation Software Version V10.0
  made 2000/08/16 at 20:03:43
# SPECCTRA ShapeBased Automation Software running in
  WindowsNT
startup C:\cct_cds\tools\SPECCTRA\tutorial\lesson1.dsn
# Design Name C:\cct_cds\tools\SPECCTRA\tutorial\
  lesson1.dsn
# SPECCTRA ShapeBased Automation Software running in
  Windows NT
# Did File Name:
  C:\cct_cds\tools\SPECCTRA\tutorial\08241611.did
# Current time = Tue Aug 24 16:11:35 2000
# PCB C:\ cct_cds\tools\SPECCTRA\tutorial
# Master Unit set up as: mm 100000
.
.
.
```

View the session transcript to monitor messages, command processing, and command results.

Monitoring progress and checking results

The autorouter generates a transcript of your work session in the Output window unless you redirect it to a file. In addition, a status file logs each routing command and the results of each pass. The status file is useful for documenting session progress and monitoring routing results. In the following procedure, you display a Routing Status Report and locate information by searching.

Locate information in the routing status report

1. Click the *Status Report* button on the tool bar.

   The Routing Status Report opens.

   You will search for the total number of connections in the design.

2. Enter **connections** in the Search entry box and click on the down arrow symbol.

   Connections is highlighted.

   The number to the right is the total number of connections in the design.

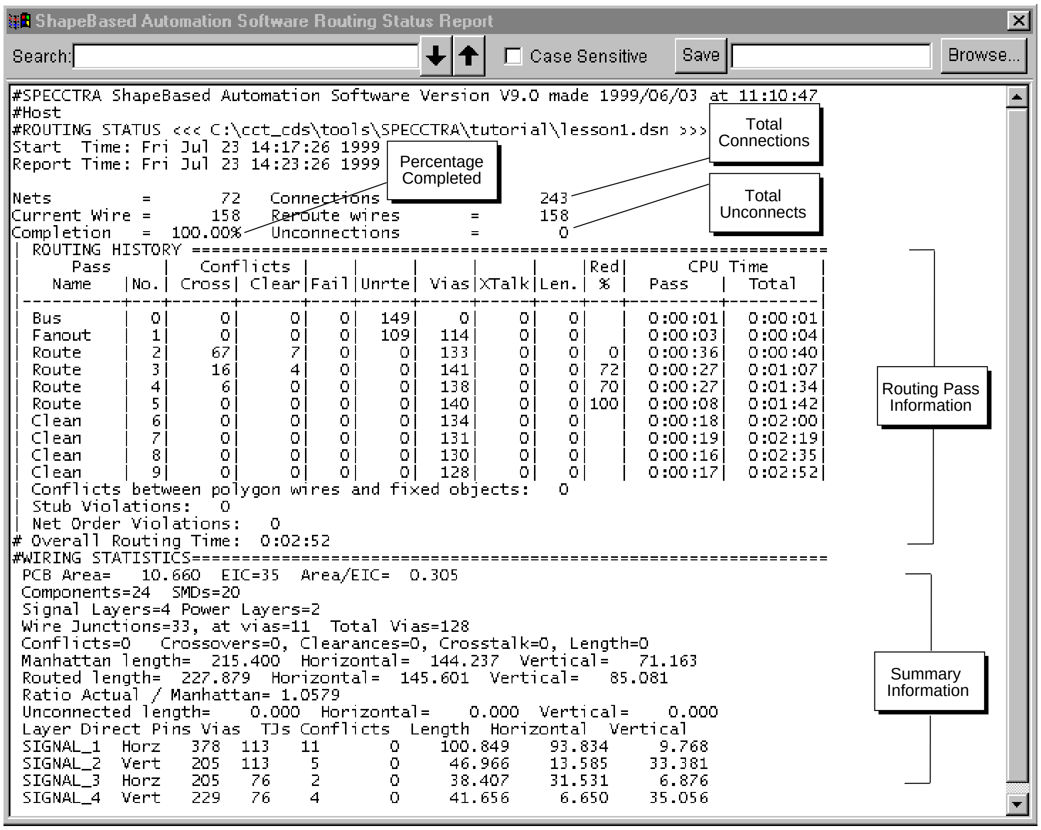
3. Use the scroll bars to observe the summary at the end of the report. The summary report includes the total number of vias used, total wire length, ratio of the manhattan length to the routed length, and the distribution of routing per signal layer.

4. Close the report.

   On the Windows platform, pull down the Control menu and click *Clos*e.

   On a UNIX platform, click *Close* at the bottom of the report window.

The following figure shows the Routing Status Report.



```
#SPECCTRA ShapeBased Automation Software Version V9.0 made 1999/06/03 at 11:10:47
#Host
#ROUTING STATUS <<< C:\cct_cds\tools\SPECCTRA\tutorial\lesson1.dsn >>>
Start  Time: Fri Jul 23 14:17:26 1999
Report Time: Fri Jul 23 14:23:26 1999

Nets         =       72   Connections             243
Current Wire =      158   Reroute wires      =    158
Completion   =  100.00%   Unconnections      =      0
| ROUTING HISTORY ===============================================================
|     Pass      |   Conflicts  |    |     |     |    |    |Red|      CPU Time     |
|    Name   |No.| Cross| Clear|Fail|Unrte| Vias|XTalk|Len.| % |   Pass  |  Total  |
|-----------+---+------+------+----+-----+-----+-----+----+---+---------+---------|
| Bus       |  0|     0|     0|   0|  149|    0|    0|   0|   | 0:00:01| 0:00:01|
| Fanout    |  1|     0|     0|   0|  109|  114|    0|   0|   | 0:00:03| 0:00:04|
| Route     |  2|    67|     7|   0|    0|  133|    0|   0|  0| 0:00:36| 0:00:40|
| Route     |  3|    16|     4|   0|    0|  141|    0|   0| 72| 0:00:27| 0:01:07|
| Route     |  4|     6|     0|   0|    0|  138|    0|   0| 70| 0:00:27| 0:01:34|
| Route     |  5|     0|     0|   0|    0|  140|    0|   0|100| 0:00:08| 0:01:42|
| Clean     |  6|     0|     0|   0|    0|  134|    0|   0|   | 0:00:18| 0:02:00|
| Clean     |  7|     0|     0|   0|    0|  131|    0|   0|   | 0:00:19| 0:02:19|
| Clean     |  8|     0|     0|   0|    0|  130|    0|   0|   | 0:00:16| 0:02:35|
| Clean     |  9|     0|     0|   0|    0|  128|    0|   0|   | 0:00:17| 0:02:52|
| Conflicts between polygon wires and fixed objects:   0
| Stub Violations:   0
| Net Order Violations:   0
# Overall Routing Time:  0:02:52
#WIRING STATISTICS===============================================================
 PCB Area=   10.660  EIC=35  Area/EIC=  0.305
 Components=24  SMDs=20
 Signal Layers=4 Power Layers=2
 Wire Junctions=33, at vias=11  Total Vias=128
 Conflicts=0   Crossovers=0, Clearances=0, Crosstalk=0, Length=0
 Manhattan length=  215.400  Horizontal=  144.237  Vertical=   71.163
 Routed length=  227.879  Horizontal=  145.601  Vertical=   85.081
 Ratio Actual / Manhattan= 1.0579
 Unconnected length=    0.000  Horizontal=    0.000  Vertical=    0.000
 Layer Direct Pins Vias  TJs Conflicts  Length  Horizontal  Vertical
 SIGNAL_1  Horz  378  113   11        0    100.849     93.834     9.768
 SIGNAL_2  Vert  205  113    5        0     46.966     13.585    33.381
 SIGNAL_3  Horz  205   76    2        0     38.407     31.531     6.876
 SIGNAL_4  Vert  229   76    4        0     41.656      6.650    35.056
```


Routing Status Report

You will learn how to read and analyze a routing status report in Lesson 3. For now, you need to know how to display the report and locate information in the report.

Using the Mouse to Zoom and Pan

You zoom in and magnify an area of the design by using the middle mouse button to drag the pointer diagonally from the lower left to the upper right corner of the area you want to magnify. As you drag the pointer, the autorouter dynamically changes your view.

You zoom out by dragging the pointer diagonally from upper right to lower left.

If you drag the pointer horizontally, the autorouter fits the whole design in the view.

You pan by clicking the mouse button at a point that you want as the center of your view.

If you have a three button mouse, you use the middle mouse button to zoom and pan your view of the design. If you have a two button mouse, you hold down the [Alt] or [◆] key and use the right mouse button to zoom and pan.

Zoom and pan using a three button mouse

1. Identify an area of the design where you want to zoom in.
2. Using the middle mouse button, drag the pointer from lower left to upper right to enclose the area and release the mouse button to zoom in.
3. Repeat the previous step to zoom in again.
4. Move the pointer left, right, above, or below the center of the current view.
5. Click the middle mouse button.

   The location of the pointer becomes the new center of view.
6. Repeat the previous step several times.

   Each time you click, the location of the pointer becomes the new center of view.
7. Drag the pointer diagonally from upper right to lower left to zoom out.

8. Click the *View All* button on the tool bar to fit the entire design in the view.

To zoom and pan using a two button mouse, you hold down the [Alt] or [◆] key and use the right mouse button.

Zoom and pan using a two button mouse

1. Identify an area of the design where you want to zoom in.
2. Pressing either [Alt] or [◆] on the keyboard and the right mouse button, drag the pointer from lower left to upper right to enclose the area and release the mouse button.

   The enclosed area is magnified to fill the view.
3. Press [Alt] or [◆] and use the right mouse button to zoom in again.
4. Move the pointer to a location that's left, right, above, or below the center of the view.
5. Press [Alt] or [◆], and click the right mouse button.

   The view pans and centers at the location under the pointer.
6. Repeat the previous step several times.

   Each time you click, the location of the pointer becomes the new center of view.
7. Press either [Alt] or [◆], on the keyboard and the right mouse button, drag the pointer diagonally from upper right to lower left to zoom out.

8. Click the *View All* button on the tool bar to fit the entire design in the view.

Note: You can also use **View > Zoom In** to zoom in and **View > Zoom Out** to zoom out.

Getting Information About Design Objects and Measuring Distance

The left mouse button performs a variety of functions. Each function is enabled by setting a mode.

The left mouse button mode displays in the mode status area beside the mouse button, as shown in the following figure.

Mode Status Area Identifies Measure as the Current Mode

Note: While using the tool, the left mouse button is always in a mode. If you attempt an operation and it does not work as you expect, check the mode status area to see whether the correct mode is set.

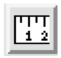
The default left mouse button mode is *Measure*. When you click a point in the work area, the tool displays the X,Y coordinates in the coordinate readout area, the output window, and in the Measure dialog box. In Measure mode, you use the left mouse button to

- Report information and properties about design objects
- Report X,Y coordinates when you click at a point in the work area
- Measure distance when you click and drag the pointer in the work area

Measurements and information about design objects are reported in the in the Output window and in the Measure dialog box. Coordinates and measured distance are reported in the Output window, the Measure dialog box, and on the status bar beside the ▲ (delta) symbol. Measured distance displays in the current measurement unit.

In the following procedure, you get information about a component pin, a via, and a wire. You also measure the center to center distance between the pins of a component.

Get information about design objects and measure distance



1. Click the *Measure Mode* button on the tool bar.

   Measure appears in the mode status area, indicating that the left mouse button is set to measure mode.

2. Click on a component pin (zoom in if necessary).

   The Measure dialog box appears.

3. Read the information about the pin in the Measure dialog box and in the Output window.

   You can scroll the Output window to view all the pin information.

4. Click on a via.
5. Read the via information in the Measure dialog box and in Output window.
6. Click on a wire.
7. Read the information about the wire in the Measure dialog box and in the Output window.
8. Click and drag the pointer from the center of one pin to the center of another (zoom in or pan if necessary).
9. Read the measured distance on the status bar beside the ▲ symbol.
10. Check the Output window and find the X,Y coordinates of the start and end points, the delta (measured) distance, and the separate X and Y distances.

The type of information you can retrieve for different objects from the Measure dialog box and the Output window is listed in the following table.

Note: The **Setup** button on the Measure dialog box opens the Interactive Routing Setup dialog box. You can then click on the Measure tab and control whether the measure information appears in the Measure dialog box or in the Output window, or both.

When you click on this object	You get this information
component	Pointer’s X,Y location Current measurement units Component outline dimensions (bounding rectangle) Layer location Reference designator Image name X,Y location in working units X,Y location in DBU Degrees of rotation
SMD pin	Pointer’s X,Y location Current measurement units Bounding rectangle for the pin and its layer location Component reference designator and image name Component location (in both measurement units and DBU) Component side location and rotation Padstack id for the pin Padstack shape description and layer location

When you click on this object	You get this information
	Component pin id Name of attached net
through-pin	Pointer’s X,Y location Current measurement units Bounding rectangle for the pin Component reference designator and image name Component X,Y location (in both measurement units and DBU) Layer location of component Component X,Y location Pin id Padstack id for the pin Padstack shape descriptions by layer including dimensions Name of attached net
via	Pointer’s X,Y location Current measurement units Padstack id for the via Padstack shape descriptions by layer including dimensions Name of attached net Wire path coordinates Width and layer locations of attached wire segments
wire segment	Pointer’s X,Y location Current measurement units

When you click on this object	You get this information
	Path coordinates
	Width and layer location of the segment
	Name of attached net

Saving Your Work

Before you end an autrouting session, you can save your work in a session file, placement file, or routes file. The following table explains the differences between these files.

A session file...	A placement file...	A routes file...
Contains placement and route data and a pointer to the original design file. After you create a session file, you can start the autorouter and load the session file instead of the original design file. When the autorouter reads a session file, it loads the original design file and the route and placement data from the session file.	Contains placement data only. When you start the autorouter and load a placement file, you must specify the original design file.	Contains route data only. When you start the autorouter and load a routes file, you must specify the original design file.
Can be loaded only when you start the autorouter.	Can be loaded when you start the autorouter and after it's running.	Can be loaded when you start the autorouter and after it's running.

A session file...	A placement file...	A routes file...
Can be used in all layout systems to merge the autorouting placement and routing data into the layout.	Can be used in most layout systems to merge the autorouting placement data into the layout.	Can be used in most layout systems to merge the route data into the layout.
Links a design file with specific route data from a particular session. The session file is useful for managing design revisions because, it links the original design file with the route data for a particular session.		

In the following procedure, you save a routes file and a session file. You usually need to save only one file. However, both procedures are included so you can become familiar with each procedure.

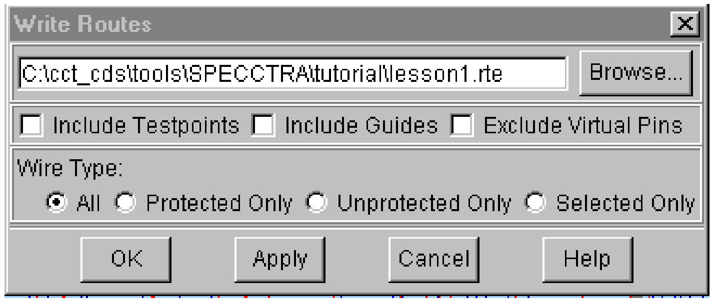
Save your work and exit the autorouter

1. Click **File > Write > Routes**.

   The Write Routes dialog box opens.

   On a Windows platform, the path to the tutorial directory and the filename *lesson1.rte* appear in the data entry box, as shown in the following figure.

On a UNIX platform, the path and */lesson1.rte* appear in the data entry box.



2. Click *OK*.

   You created a routes file. You could create a session file using **File > Write > Session**, but you will create a session file when you exit the autorouter.

3. Click **File > Quit.**

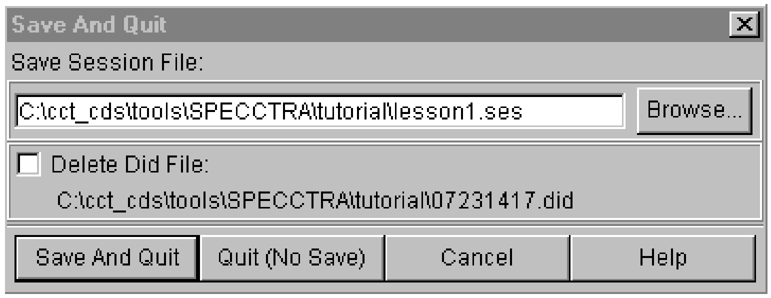
   The Save And Quit dialog box opens, as shown in the following figure.

   On a Windows platform, the path to the tutorial directory and the file *lesson1.ses* appear in the data entry box.

   On a UNIX platform, the path and *./lesson1.ses* appear in the data entry box.

4. Make sure *Delete Did File* is not checked.

   This saves the did file created during this session. A did file contains the command history from the autorouter's work session. You will look at this did file later in this lesson.



5. Click *Save And Quit*.

The autorouter saves the routing information in a session file and exits. You will use this session file to start the autorouter in the next section of this lesson.

Load a session file instead of a design file (Windows)

1. Click the Windows Start button and click **Programs > SPECCTRA V10.0 > SPECCTRA V10.0**.

   The Startup dialog box opens.

2. Enter **C:\cct_cds\tools\SPECCTRA\tutorial\lesson1.ses** in the *Design/Session File* data entry box.

3. Click *Start SPECCTRA*.

   The session file loads the design file and the wiring file from the previous session.

4. Click **Report > File.**

   The Report File dialog box opens.

5. Click *Browse*.

   The Open dialog box appears.

6. Choose the tutorial directory from the Directories list.

7. Choose *lesson1.ses* from the File Name list and click *Open*.

The path and filename are added to the Report File dialog box.

8. Click *OK*.

   A report window displays the session file.

9. Browse the session file and locate the following:

   - Name of the session file (*lesson1.ses)*
   - Name of the original design file (*lesson1.dsn)*
   - Date and time the file was created
   - Section of the session file that contains the route data (located below the network_out statement)

10. Close the session file.

11. Click **File > Quit**.

    The Quit dialog box opens, as shown in the following figure.

12. Make sure *Delete Did File* is not checked.

13. Click *Quit.*

Load a session file instead of a design file (UNIX)

1. Enter **specctra** at the shell prompt.

   The Startup dialog box opens.

2. Click the *Browse* button to the right of the Design/Session File data entry box.

   The Select File dialog box opens.

3. Change the filter to **.ses*.
4. Select *lesson1.ses* from the files directory and click *OK*.
5. Click *Start SPECCTRA*.

   The session file is loaded with the wiring from the previous session.
6. Click **Report** > **File.**

   The Report File dialog box opens.
7. Click the *Browse* button and select *lesson1.ses* from the Files directory.
8. Click *OK* from the Report File dialog box.

   A report window opens and displays the session file.
9. Browse the session file and locate the following:
   - Name of the session file (*lesson1.ses)*
   - Name of the original design file (*lesson1.dsn*)
   - Date and time the file was created
   - Section of the session file that contains the route data
10. Close the session file.
11. Click **File > Quit**.

    The Quit dialog box opens, as shown in the following figure.

12. Make sure *Delete Did File* is not checked.
13. Click *Quit.*

Reviewing Session Command History

The autorouter creates a command history file for each session. This command history file is called a *did* file.

On Windows platforms, the did file is created with an eight character, numeric filename that derives from the month, day, hour, and minute when you start an autorouting session. For example, a did file for a session you started June 15th at 11:45 is named `06151145.did`.

On UNIX platforms, the did file is created with a 10 character, numeric filename that derives from the month, day, hour, minute, and seconds when you start an autorouting session. For example, a did file for a session you started June 15th at 11:45:20 is named `0615114520.did`.

The following is an example of a did file.

```
# Cadence Design Systems, Inc.
# SPECCTRA Automatic Router
# SPECCTRA Version V10.0 made 00/10/24 at
14:08:45
# Running on host 5540a1cf
#
#             Command Line Parameters
#             -----------------------
# Design File Name : ./lesson1.ses
# No "-do" or "-docmd" switches specified
on command line.
# Colormap File Name : color.std
# FLEXlm License File Name :
/usr/local/flexlm/licenses/license.dat
# Status File Name : ./monitor.sts
# Wires File Name : lesson1.w
#
#
```

```
report c:\specctra\tutorial\lesson1.ses
quit -c
```

Each time you start the autorouter, a new did file is created. Together, the status file, did file, and session file capture the routing status, command log, and routing data for the autorouting session.

View the did file from your previous session (Windows)

1. Start the autorouter and load *lesson1.dsn.*
2. Click **Report > File**.

   The Report File dialog box opens.
3. Click *Browse* in the Report File dialog box.

   The Open dialog box opens.
4. Scroll the Open dialog box and select the did file from the previous session. The did file for the previous session is located just before the did file for the current session. The last did file in the list is the one for the current session.
5. Open the did file from the previous session.

   The path and file are added in the Report File dialog box.
6. Click *OK*.

   A report window displays the did file.
7. Browse the did file and verify the command history from the previous autorouting session.
8. Close the report.
9. Click **File > Quit**.

   The Quit dialog box opens.

10. Click *Delete Did File* to remove the did file.
11. Click *Quit.*

    The autorouter exits without saving a session file or a did file.

View the did file from your previous session (UNIX)

1. Start the autorouter and load *lesson1.dsn.*
2. Click **Report > File**.

   The Report File dialog box opens.
3. Click *Browse* in the Report File dialog box.

   The Select File dialog box opens.
4. Select the did file from the previous session. The did file for the previous session is located just before the did file for the current session. The last did file in the list is the one for the current session.
5. Click *OK*.

   The path and filename are added in the Report File dialog box.
6. Click *OK*.

   A report window displays the did file.
7. Browse the did file and verify the command history from the previous autorouting session.
8. Close the report.
9. Click **File > Quit**.

   The Quit dialog box opens.
10. Click *Delete Did File* to remove the did file.
11. Click *Quit.*

The autorouter exits without saving a session file or a did file.

In Lesson 3, you will learn how to use a did file to capture GUI commands and create a command file (do file) you can run in the autorouter.

What You Learned

In this lesson, you learned about the autorouter's user interface and how to perform basic design tasks. You learned how to

- Start the autorouter and load a design
- Use the GUI to execute commands
- Pan and zoom
- Measure distance and get information about design objects
- Save your work
- Review a session's command history

In the next lesson, you will learn how to place components interactively and automatically.

Lesson 2: Placing Components

What You Will Learn

This lesson teaches you how to use the autorouter's interactive and automatic placement tools. In this lesson you will learn

- The basic steps for placing components
- How to set placement options
- How to set placement rules
- How to preplace connectors and other critical components
- How to place large components
- How to edit the placement
- How to place small components

This lesson takes about 60 minutes to complete.

What to do Before You Begin

Before you begin this lesson, complete Lesson 1 "Learning Basic Concepts."

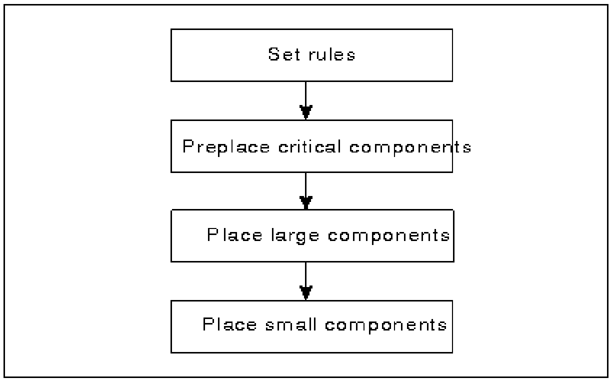
Note: You need a placement license to do the exercises in this lesson.

Understanding the Basic Steps for Placing Components

Placing components consists of four basic steps.

- Setting rules

- Preplacing critical components
- Placing large components
- Placing small components



The Basic Steps of Component Placement

In the first step, you set placement rules. During both automatic and interactive placement, the design is checked for rule violations.

In the second step, connectors and other position-critical components are preplaced and locked in position using interactive placement tools.

In the third and fourth steps, components are placed and locations are optimized to reduce manhattan lengths and guide crossings. Large components are placed first and then small components.

Setting Placement Rules

The autorouter provides a comprehensive set of placement rules. Basic placement rules control the spacing between components, the orientation of components, and the sides on which the autorouter places components. More advanced rules control floor planning based on power dissipation, power supply, and component height. In this lesson, you will set basic rules. For more information on setting advanced placement rules, see the Placement online help.

Understanding the rules hierarchy

Placement rules can be specified at different levels and, consequently form a hierarchy. Rules at higher levels in the hierarchy override rules at lower levels that are set for the same physical objects. For example, consider what happens if you set a global (PCB) spacing rule of .25 inches for all components and a spacing rule of .8 inches for a specific connector. The autorouter follows the .8 spacing rule only in the area surrounding the connector. The autorouter follows the .25 spacing rule in the areas surrounding other components. The component-level spacing rule overrides the PCB-level spacing rule.

The following table shows the levels of hierarchy in which you can set rules and the order of precedence for all placement rule levels. Global placement rules (PCB rules) have the lowest precedence, and image_image rules have the highest precedence.

Rule level	What it does
image_image	Sets rules between images. An image is the footprint definition of a component. This is the highest precedence rule.
family_family	Sets rules between families. A family is a group of images.
room_image_set	Sets rules for an image_set assigned to a room.
room	Sets rules for a room, which is an area of the design in which the autorouter places specified components.
super cluster	Sets rules for a super cluster, which is a group of components that is treated as a single component.
component	Sets rules for a component, which is an instance of an image.
image	Sets rules for an image, which defines a component footprint. An image is the footprint definition of a component.
image_set	Sets rules for images with the same type property. The type properties are large, small, capacitor, or discrete.
pcb	Sets global rules for all components in the design. This is the lowest precedence rule.

The order of precedence is fixed in the autorouter and cannot be changed.

Setting a placement grid and spacing rule

Using the Placement Setup dialog box, you can set a placement grid and a PCB spacing rule.

The autorouter does not require that you define a grid for automatic placement, but if design or manufacturing rules dictate a placement grid, you can set the grid for all components (SMD and through-pin) in the Placement Setup dialog box.

Note: You can set separate placement grids for SMD and through-pin components by using **Rules > PCB > Placement Grids**.

In addition to setting the placement grid, you can use the Placement Setup dialog box to set a PCB spacing rule. A PCB spacing rule sets the minimum spacing allowed between all components in the design. Rules are set at other levels through the **Rules** menu. In the following procedure, you will set a PCB spacing rule of .05 inches.

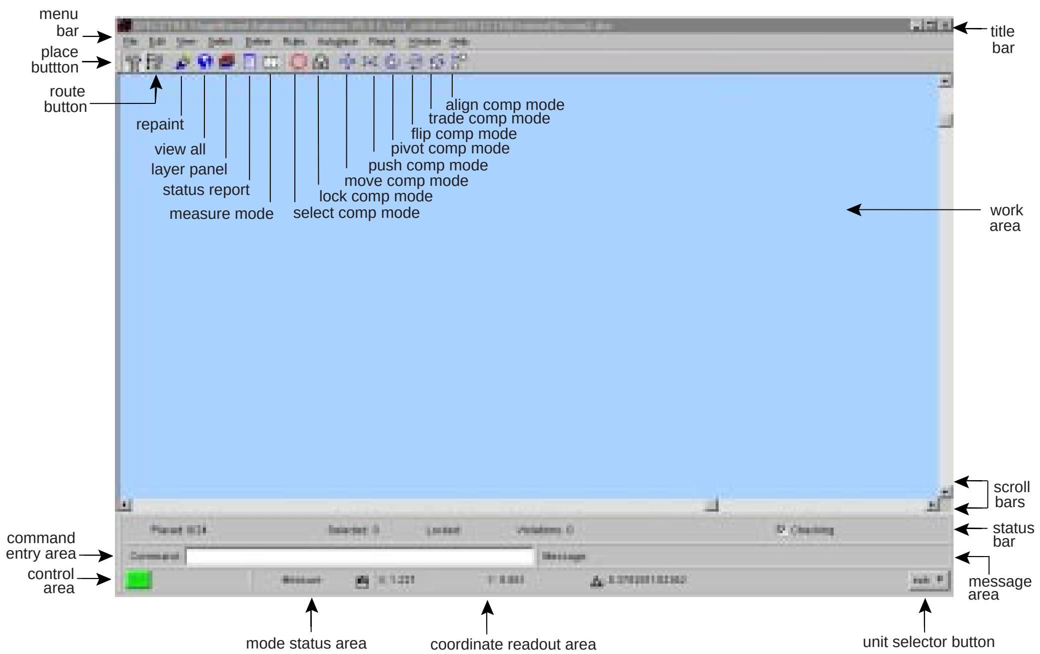
Set a placement grid and spacing rule

1. Start the autorouter and load *lesson2.dsn*.

   When you first open the autorouter, you are in Routing mode. The menu bar and status bar are specific to autorouting. You need to change to Placement mode.

2. Click the *place* button on the tool bar.

   The tool bar changes.



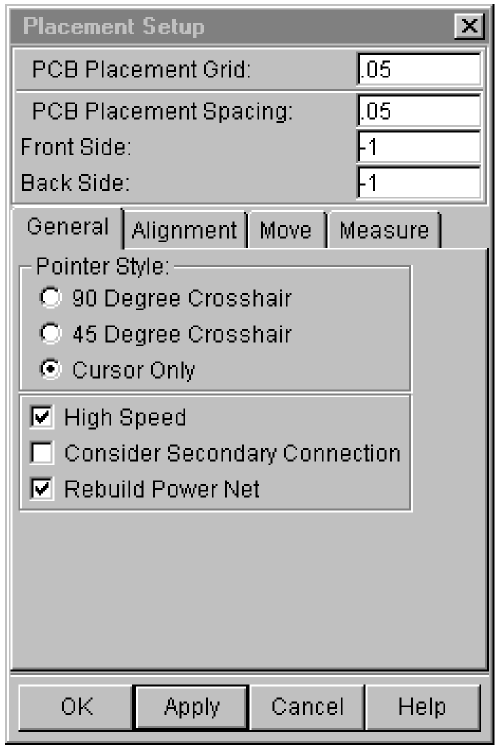
Placement GUI (Windows)

3. Click **Autoplace > Setup**.

   The Placement Setup dialog box opens.

4. Enter **.05** in the *PCB Placement Grid* data entry box.
5. Enter **.05** in the *PCB Placement Spacing* data entry box.

   The Placement Setup dialog box looks like the following.



6. Click *OK*.

You defined a grid of .05 inches and a spacing rule of .05 inches. Notice that the *Checking* option in the right side of the status area is selected. This option checks for placement rule violations while you interactively place components and prevents placing or moving a component if it violates a rule. If you want to place a component in a location that violates a rule, turn off *Checking*.

Note: You can set separate spacing rules for SMD components, through-pin components, and between SMD and through-pin components by using **Rules > PCB > Spacing**. You also can use this command to see the current PCB spacing rule.

Preplacing Connectors and Critical Components

After you set placement rules, you are ready to preplace connectors and other critical components. In this stage of placing components, you will

- Display the component reference designators
- Place components by specifying XY locations
- Place components from a defined list
- Lock the preplaced components

Displaying component reference designators

In the following procedure, you turn on the reference designator labels so that you can identify components.

Display component reference designators

1. Click **View > Labels**.

   The View Labels dialog box opens.
2. Make sure *Ref Des* is selected.
3. Make sure the *Side* is set to *Both*.
4. Click *OK*.

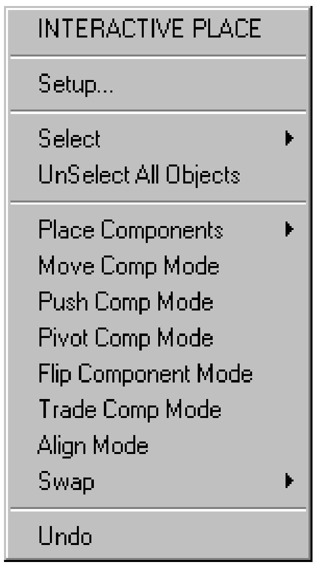
A reference designator label appears in the center of each component.

Note: The router scales the work area. If you do not see all the designator labels, you need to zoom in to adjust the magnification of the work area.

Placing components by specifying X,Y locations

Through the Interactive Place menu, the autorouter offers a variety of ways to place components interactively. You can place a single component, a list of components, and multiple components. You can place components by specifying X,Y locations, by clicking a point in the work area, or based on connectivity.

The Interactive Place menu appears when you press the right mouse button in the work area.



To select a command from the Interactive Place menu, you will follow these steps:

- Hold down the right mouse button
- Slide the pointer to highlight the command
- Release the right mouse button

In this section, you will place the origins of components J1, J2, and U9 at exact X,Y locations.

Place components by specifying X,Y locations

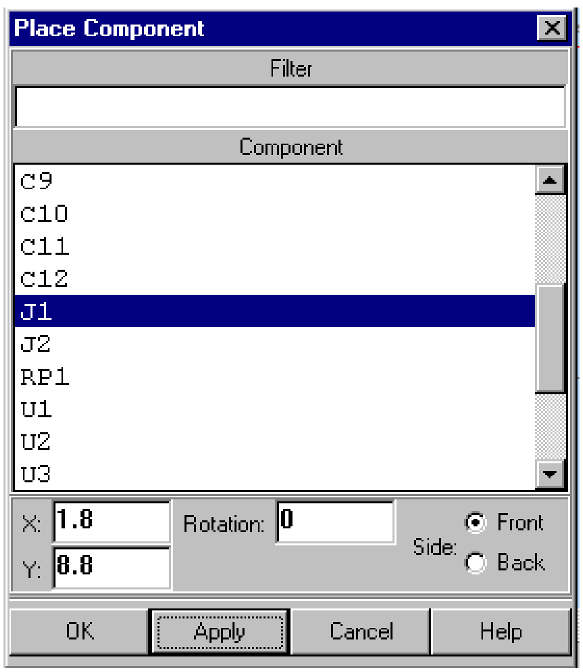
1. Press the right mouse button and select **Place Components > XY Location**.

   The Place Component dialog box opens, as shown in the following figure.

2. Scroll the *Components* list to locate J1.
3. Click *J1* in the *Components* list.

   J1 is highlighted.

4. Enter **1.8** in the *X* data entry box.
5. Enter **8.8** in the *Y* data entry box.
6. Make sure the *Rotation* is *0*.
7. Make sure the *Side* is *Front*.



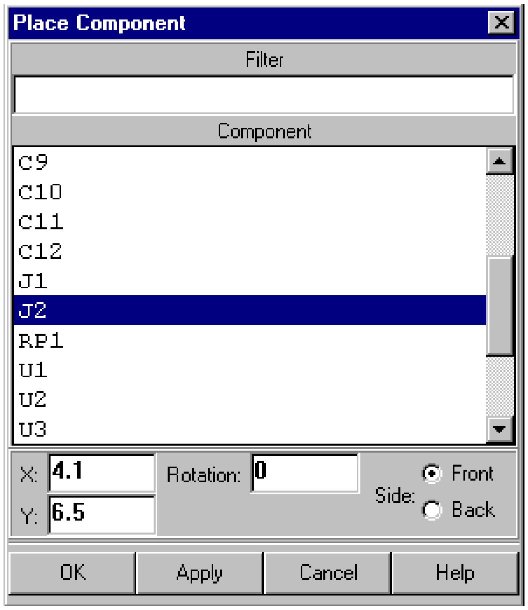
8. Click *Apply* to place the origin of J1 at 1.8, 8.8.

Next you place J2.

9. Click *J2* in the *Components* list.
10. Type **4.1** in the *X* data entry box.
11. Type **6.5** in the *Y* data entry box.

Keep the *Rotation* at 0 and Side set to *Front*.

The dialog box looks like the following.



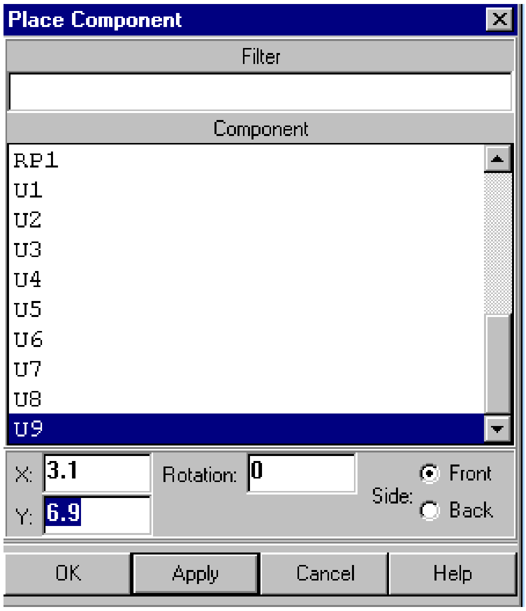
12. Click *Apply* to place the origin of J2 at 4.1, 6.5.

Next you place the PGA component labeled U9. Like the connectors, U9 has a large number of interconnections with other components in the design, which is why you interactively place it.

13. Click *U9* in the *Components* list.
14. Enter **3.1** in the *X* data entry box.
15. Enter **6.9** in the *Y* data entry box.

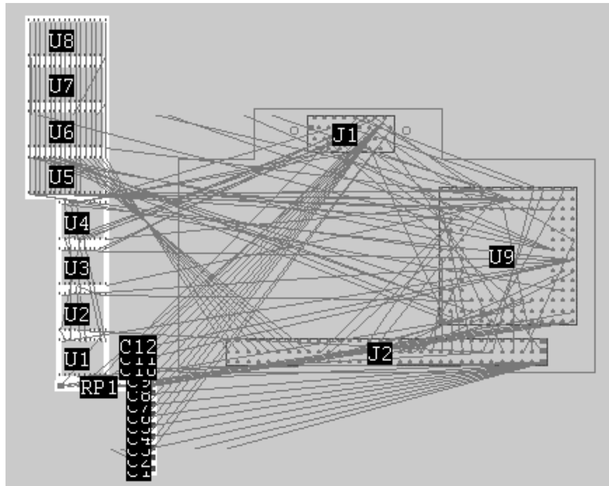
Keep the *Rotation* at 0 and Side set to *Front*.

The dialog box looks like the following.



16. Click *OK*.

    The design is shown in the following figure.


U8
U7
U6
U5
U4
U3
U2
U1
RP1
C12
J1
U9
J2

You just learned how to place components by specifying X,Y coordinates. Next you will learn how to place a list of components.

Placing components from a list

Using Place List mode, you can specify a list of components and place them in the order you specify. The first component in the list attaches to the pointer. Drag the component to the location you want, and click the left mouse button to place it. Immediately, the next component in the list attaches to the pointer so you can drag the component to the location you want and click the left mouse button to place it. You repeat this process until all components in the list are placed.

Place components from a list

1. Press the right mouse button and select **Place Components > Place List Mode**.

   The Place Component List dialog box opens.

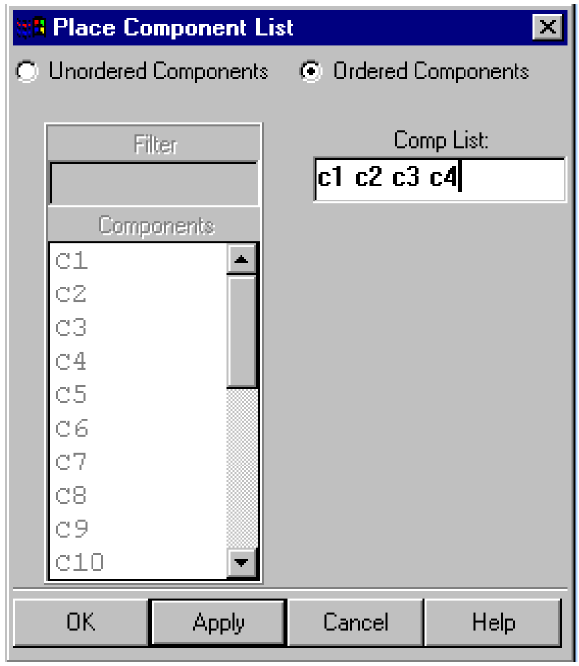
2. Select *Ordered Components*, which tells the autorouter to place the components in the order that you specify in the *Comp List*.

3. Type the following in the *Comp List* data entry box:

   c1 c2 c3 c4

Note: Make sure you leave a space between each reference designator.

The Place Component List dialog box looks like the following.



4. Click *OK*.

   C1 attaches to the pointer. You will need to rotate it 90 degrees counterclockwise before you place it.

5. Press the right mouse button and select **Pivot Mode > 90**.

   A pivot arm attaches to the center of the component. This pivot arm looks like a string that pivots the component in the direction you move the pointer.

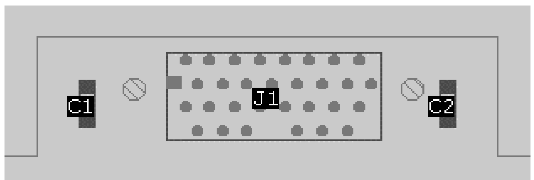
6. Move the pointer away from the center of the component in a counterclockwise direction.

   The component rotates 90 degrees counter clockwise. increments as you move the pointer.

DR:90 appears near the right side of the status bar when the component is 90 degrees from its original position. DR stands for Delta Rotation.

7. Click the left mouse button when C1 is 90 degrees counterclockwise from its original position.

   You are ready to place C1. The following figure shows where to place C1 and C2.



8. Drag C1 to the left of J1 and click to place it.

   You can use the middle mouse button for the following operations to pan or zoom to a new location.

   C2 attaches to the pointer after you place C1.

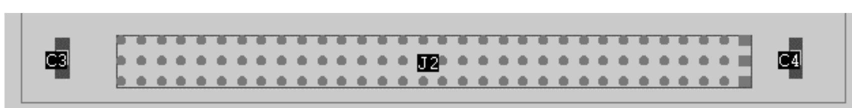
9. Repeat steps 5 through 7 to rotate C2 90 degrees counterclockwise.

10. Drag C2 to the right of J1 and click to place it.

    C3 attaches to the pointer.

11. Repeat steps 5 through 7 to rotate C3 90 degrees counterclockwise.

    You are ready to place C3. The following figure shows where to place C3 and C4.



12. Drag C3 to the left of J2 and click to place it.

C4 attaches to the pointer.

13. Repeat steps 5 through 7 to rotate C4 90 degrees counterclockwise.
14. Drag C4 to the right of J2 and click to place it.

Note: If you do not like where you placed a component and you want to move it, press the right mouse button and select *Move Component*. Click the component, drag it to the new location, and click to place it.

Next you will lock the preplaced components so the autorouter cannot move them during automatic and interactive placement.

Locking components

After you place critical components, you lock them in position so the autorouter does not move them during automatic or interactive placement. If you need to move a component at a later time, you can unlock it, move it, and lock it again.

Lock critical components

1. Click the *Lock* button on the tool bar.

   Lock Position appears in the mode status area.
2. Click a point to the left of C1.
3. Drag the pointer diagonally around C1, J1, and C2, and release the left mouse button when the bounding box encloses the components.

   The locked components display a magenta border. Magenta is the default *locked* color.
4. Repeat steps 2 and 3 to lock components C3, J2, and C4.

Note: You can unlock the components while you are in Lock Position mode. To unlock the components, click on each locked component or draw a bounding box around the components. You also can use **Edit > [Un]Lock Components** and **Edit > [Un]Lock Components Mode** to lock and unlock components.

Placing Large Components

The autorouter assigns the large and small properties to components based on the number of pins. A large component contains four or more pins. A small component contains three or less pins. The autorouter usually places large components with the highest connectivity first.

In this section you will

- Define areas where you do not want components placed
- Automatically place large components
- Interchange placed components to reduce manhattan lengths, minimize crossovers, and reduce congestion

Defining areas where the autorouter cannot place components

Before you automatically place the large components, you need to provide routing space around the connectors. To do this, you will specify an area around each connector where the autorouter cannot place components. These areas are called placement keepout areas.

Define placement keepout areas

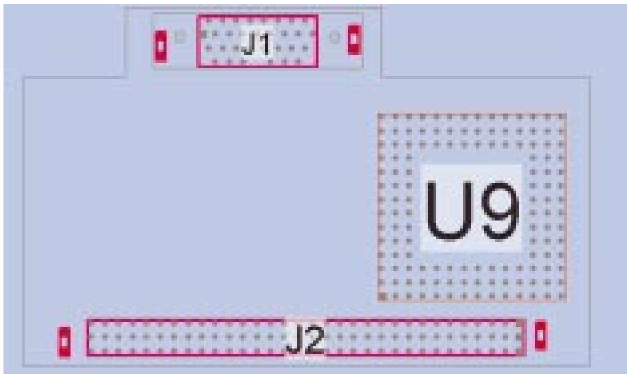
1. Click **Define > Keepout > Draw Mode**.

   Draw Keepout appears in the mode status area. You will draw a bounding box as close as possible to C1, J1, and C2.

2. Click a point to the lower left of C1.

3. Drag the pointer diagonally around C1, J1, and C2, and release the left mouse button when the bounding box encloses the components. If you do not like how you drew the box, click **Define > Keepout > Draw Mode** to remove the box and start with step 2 to draw a new bounding box.

   In the following figure, the ratsnest is not displayed. On your screen, the area around C1 and C2 might be difficult to see because the ratsnest is displayed.

   The ratsnest lines are called unroutes. You will turn off the display of unroutes, so that you can see the keepout area more easily.

4. Click **View > Guides > Off**.

   The keepout area is now easier to see.

5. Press the right mouse button and select *Define Polygon as Keepout*.

   The Add Polygon as Keepout dialog box appears, as shown in the following figure.

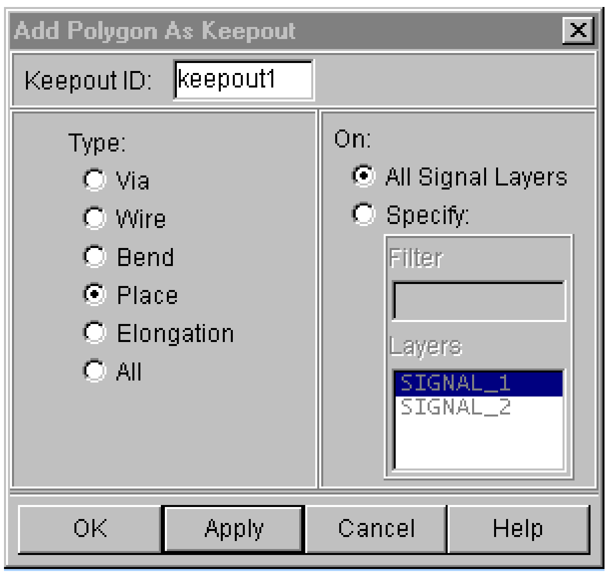
6. Make sure the Keepout ID is *keepout1*.

   The *Keepout ID* is the name the autorouter assigns to the keepout area.

7. Select *Place* as the *Type*.

   This sets the area as a placement keepout, in which the autorouter cannot place components.

   The Add Polygon As Keepout dialog box looks like the following.



8. Click *OK*.

   The keepout area appears as a bounding box filled with a crosshatched pattern.

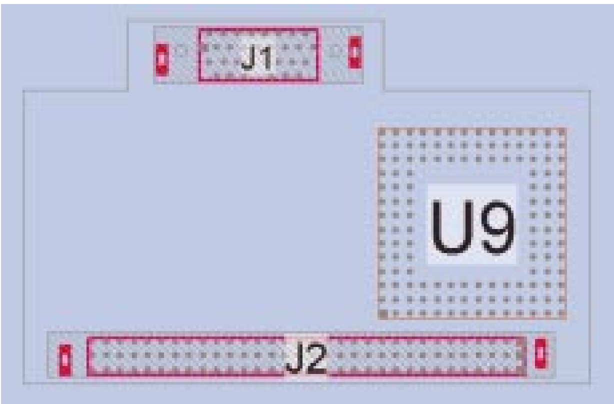
Note: Information about keepouts will display in the measure box and output window if the object selection button is turned on in the layers panel next to the keepout button.

Next you create a keepout area around C3, J2, and C4.

9. Make sure Draw Keepout mode is still set by checking the mode status area.
10. Click a point to the lower left of C3.
11. Drag the pointer diagonally around C3, J2, and C4, and release the left mouse button when the bounding box encloses the components. Remember to draw the boundary box as close as possible to the components.
12. Press the right mouse button and select *Define Polygon as Keepout*.

    The Add Polygon as Keepout dialog box opens.
13. Make sure the Keepout ID is *keepout2* and *Place* is selected.
14. Click *OK*.

    The keepout areas appear as crosshatched rectangles, as shown in the following figure.



Next you will automatically place the large components. First, display the unroutes so you can see the connectivity while the autorouter places the components.

15. Click **View > Guides > All**.

    The guides display.

Note: To create a keepout area by entering coordinates, use **Define > Keepout > By Coordinates**.

Automatically placing large components

You initially place ICs and other large components by using the InitPlace Large Components dialog box. You use the dialog box to control the preferred component spacing, placement side, and component orientations.

These preferences are not rules. Preferences set in the InitPlace Large Components dialog box are followed if they do not violate placement rules.

You will place the large SMD components on the front side with a vertical orientation and the PTH component (RP1) on the front side with a horizontal orientation.

Automatically place large components

1. Click **Autoplace > InitPlace Large Components**.

   The InitPlace Large Components dialog box opens, as shown in the following figure.

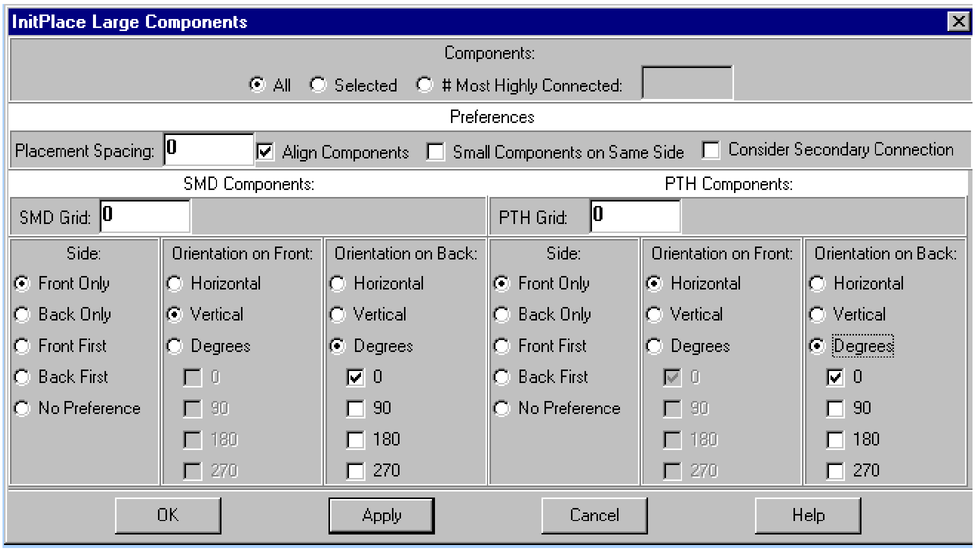
2. Make sure *All* is selected on the components panel.

   This option places all large, unplaced components.

3. Select *Front Only* for the *Side* on the *SMD Components* panel.

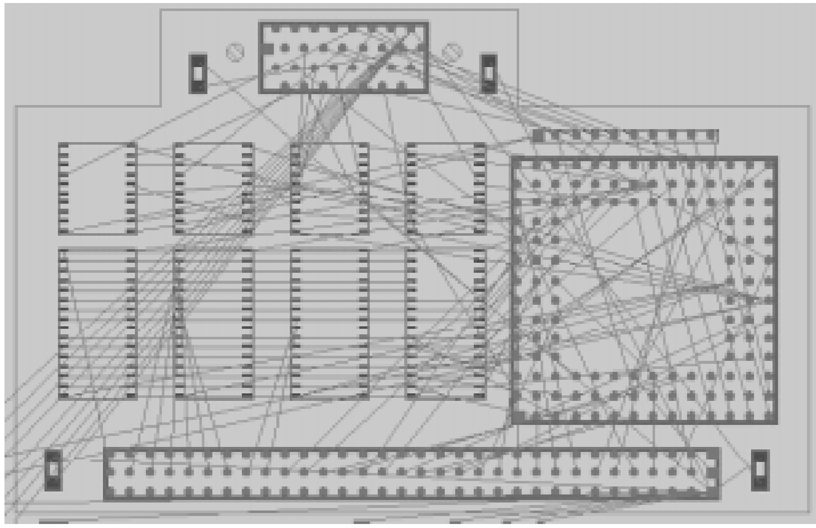
   This option is on the left side of the dialog box.

4. Select *Vertical* for *Orientation on Front* on the *SMD Components* panel.
5. Select *Front Only* for the *Side* on the *PTH Components* panel.
6. Click *Horizontal* for *Orientation on Front* on the *PTH Components* panel.



7. Click *OK*.

   The autorouter places the SMD and PTH components, as shown in the following figure.



Interchanging components

After the initial placement, you interchange component positions to reduce manhattan lengths, minimize crossovers, and reduce congestion.

The interchange operation is sometimes referred to as a pairwise interchange because the operation applies to a single pair of components at a time. The goal is to place interconnected components in close proximity to reduce wire lengths.

Multiple interchange passes usually produce the best results. Use eight or more interchange passes. If an interchange pass does not improve weighted manhattan lengths compared to the previous pass, the interchange operation stops and the remaining passes are skipped. You can use the placement status report to compare manhattan lengths before and after interchange passes.

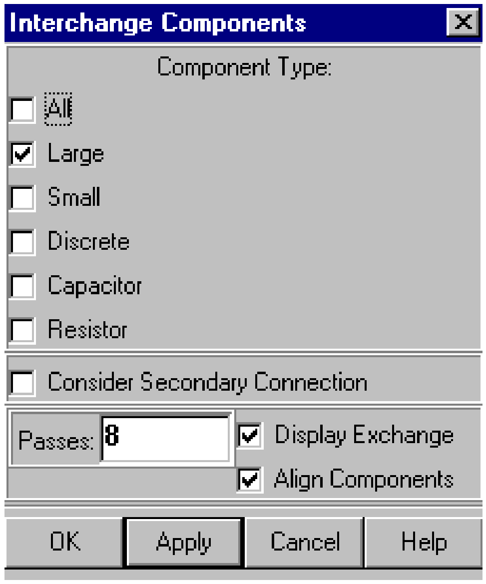
Interchange large components

1. Click **Autoplace > Interchange Components**.

   The Interchange Components dialog box appears.

2. Make sure *Large* is selected.
3. Make sure **8** is in the value for *Passes*.

   The Interchange Components dialog box looks like the following.



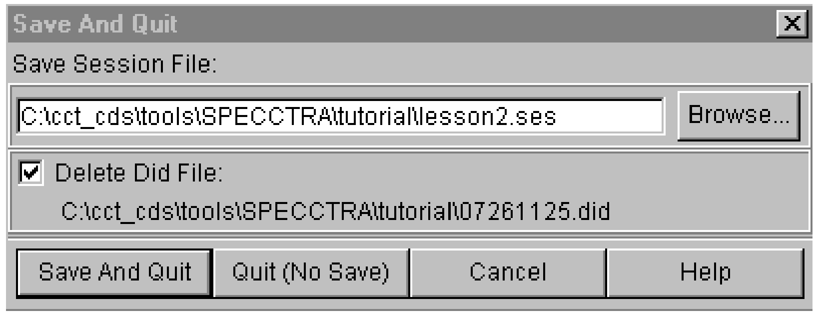
4. Click *OK*.

   After each component interchange, the work area is repainted because *Display Exchange* in the dialog box was selected. This button controls whether graphics are updated with each component interchange. To improve performance with larger designs, you unselect *Display Exchange*.

5. Click **File > Quit.**

   The Save and Quit dialog box opens, as shown in the following figure.

6. Click *Delete Did File* to remove the did file.

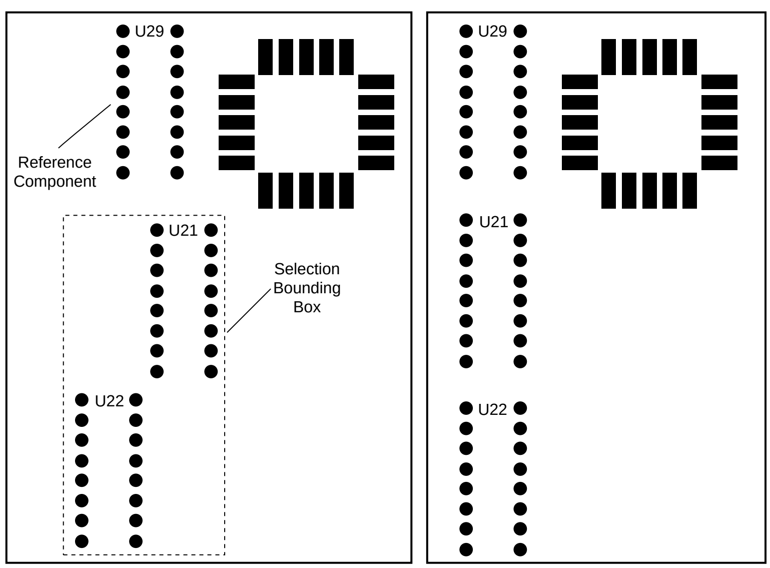


7. Click *Quit (No Save)* to exit without saving a session file and did file.

Next you will learn to interactively align components.

Aligning components

You can align components by using Align Comp mode. In this mode, you select the components you want to move, then you select a reference component to which these components align.



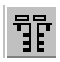
Before Alignment After Alignment

If *Checking* is on and the alignment causes a placement violation, the autorouter does not align components. If you want to ignore rule violations, turn off *Checking* in the status bar. You'll leave *Checking* on in the following procedure.

Align components

1. Start the autorouter and load *lesson2.dsn*.

   You need to change to Placement mode.



2. Click the *Placement* button on the tool bar.

   You will load a placement file.

3. Click **File > Read > Placement**.

   The Read Placement dialog box opens.

4. Click the *Browse* button.

   The *Open* dialog box appears for Windows platforms.

   The *Select File* dialog box appears for UNIX platforms.

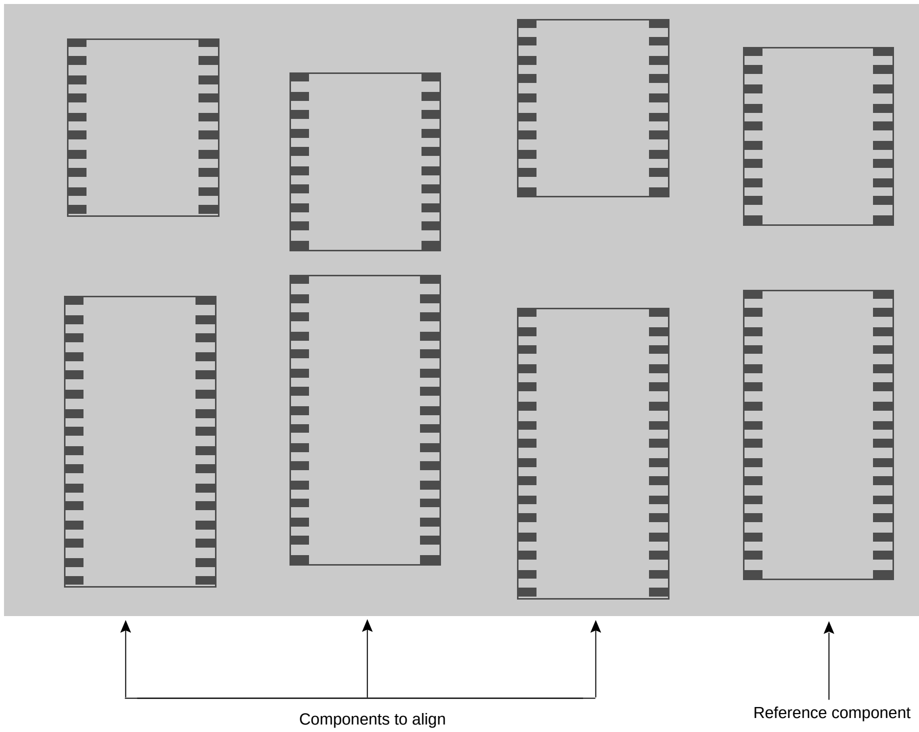
5. Change to the autorouter's tutorial directory for Windows platforms and open *lesson2.plc*.

   Make sure *lesson2.plc* is selected and click *OK* for UNIX platforms.

   The filename is added in the Read Placement dialog box.

6. Click *OK*.

   The placement file loads. The Small Outline Integrated Circuits (SOICs) are not aligned, as shown in the following figure.



7. Click **View > Guides > Off** to turn off the guides.

   You will align the SOICs.

8. Press the right mouse button and select *Align Mode*.

   Align Comp appears in the mode status area.

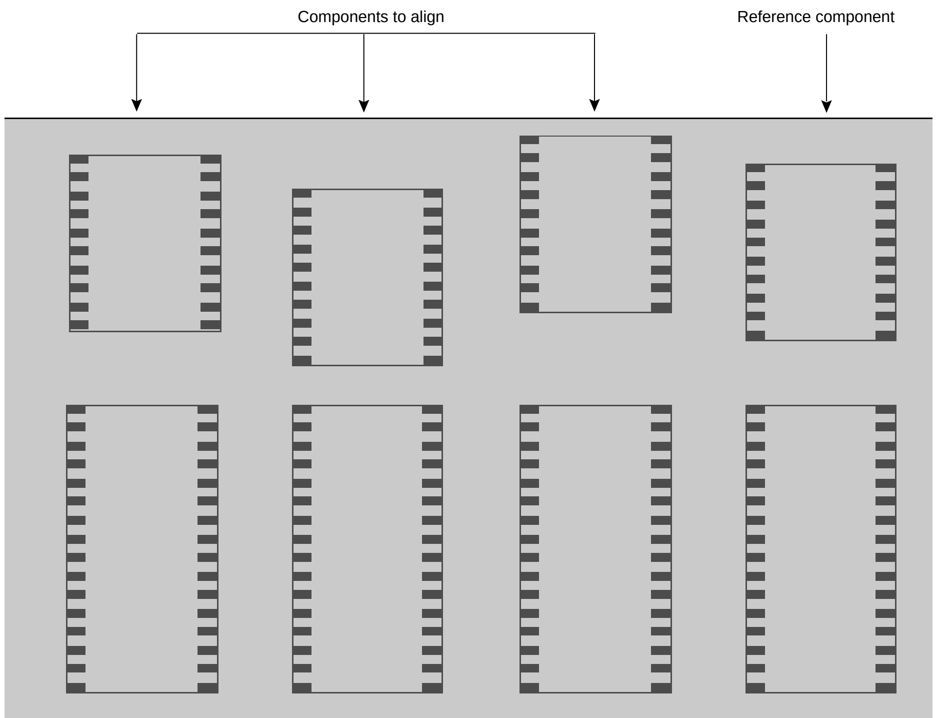
   You will use the right SOIC as the reference component to align the three SOICs.

9. Drag the pointer to enclose the three 32-pin SOICs that are located on the left as shown in the previous figure.

   The three SOICs are selected. You will align these SOICs to the 32-pin SOIC that is located on the right.

10. Click in the 32-pin SOIC that is located on the right.

    The components align, as shown in the following figure.



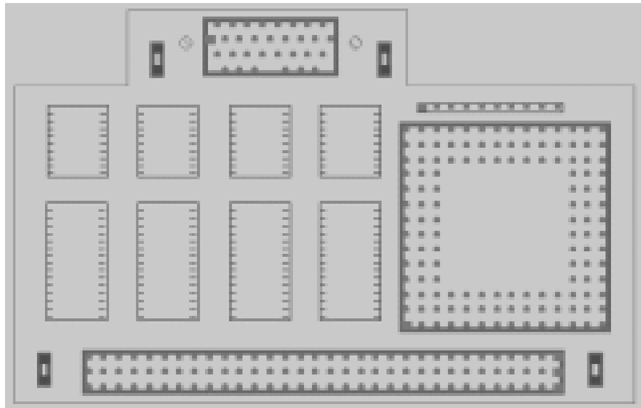
Next you will align the three 20-pin SOICs that are located above the 32-pin SOICs as shown in the previous figure. Align Comp mode is still active.

11. Drag the pointer to enclose the three 20-pin SOICs that are located on the left.

    The three SOICs are selected. You will align these SOICs to the reference 20-pin SOIC component that is located on the right.

12. Click in the 20-pin SOIC that is located on the right.

    The components align, as shown in the following figure.



Note: The autoouter uses a reference point on each component to make the alignment. In the previous procedure, the alignment was relative to the upper left pin of the reference component, which is the default. You can change this reference point by using the Placement Setup dialog box.

Placing the Capacitors

You can place small signal components by using **Autoplace > InitPlace Small Components.** To place decoupling capacitors, you will use another method.

The autorouter can *learn* a component pattern, which is the side, location, and orientation of a small component relative to a large component. The autorouter applies this component pattern by placing other instances of the small component image in the same pattern to other instances of the large component image.

In this section, you will place the capacitors by having the autorouter learn and apply component patterns. You will follow these steps:

- Display power pin labels
- Move, flip, and pivot a capacitor
- Learn the component pattern
- Apply the component pattern to other instances of the large component image

Displaying power pin labels

You will need to display the power pin labels to orient the components. To display the power pins, turn on the power pins layer.

Display power pins labels

1. Click the *Layer* button on the tool bar.

   The Layers panel appears. The Layers panel controls layer visibility, layer routing direction, and layer selection.
2. Click *Power pins* on the Layers panel. Power pins is located near the bottom of the Layer panel.

   The power pins display.
3. Click *Close* on the Layers.

Now that you can see the power pins, you will move C5 (on the PCB bottom side) so that the power pin of C5 is under the power pin of U1 (on the PCB top side).

Moving, flipping, and pivoting a component

An easy way to place a component is to use Move Comp mode. When you click on a component in this mode, the component's reference designator displays. As you move the pointer, a ghost image of the component follows. The next click places the component at the location under the pointer.

You will need to display the reference designators, which appear at component centers.

Move, flip, and pivot a component

1. Click **View > Labels**.

   The View Labels dialog box opens.

2. Make sure *Ref Des* is selected.
3. Click *OK*.
4. Press the right mouse button and select *Move Comp Mode*.

   Move Comp appears in the mode status area.

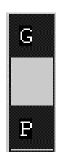
5. Click on C5, which is the bottom component in the column of unplaced components. You can use the middle mouse button to zoom in if necessary.

   The component attaches to the pointer. You need to flip the component to the back side by using the Move Comp menu.

6. Press the right mouse button and select *Flip* from the Move Comp menu.

   This menu contains commands that manipulate the component before you place it.

   Next you need to pivot the component so that the power pin is at the bottom of the component, as shown in the following figure.

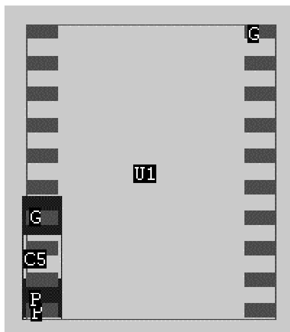
7. Press the right mouse button and select **Pivot > 90**.

   This command pivots C5 in 90 degree increments.

C5 attaches to the pointer.

8. Move C5 to the lower left of U1 (so that the power pin of C5 is under the power pin on U1). You can use the middle mouse button to zoom in if necessary.
9. Click to place the power pin of C5 under the power pin of U1.

   The following figure shows the relationship of C5 to U1.



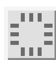
Note: After you attach a component to the pointer and move the pointer, you can return the component to the previous location by pressing the right mouse button and selecting *Cancel*.

Next the autorouter will learn this component pattern and apply it to the components with the same image ID.

Learning and applying the component pattern

Now that you placed C1, the autorouter can learn the component pattern of the small component (C5) in relationship to the large component (U1). The autorouter learns the orientation, location, and side of the small component with respect to the large component. The autorouter then places other instances of the capacitor image in the same pattern to other instances of the SOIC image.

Learn and apply the component pattern



1. Click the *Select Comp* button on the tool bar.

   Sel Comp appears in the mode status area.

2. Drag the pointer to enclose U1 and C5.

   The two components are selected and *Selected:2* appears in the status area.

3. Click **Autoplace > Small Comp Pattern > Learn**.

   The autorouter stores orientation, location, and side information about the relationship between C5 and U1.

4. Unselect *C5* and *U1* by dragging the pointer over them.
5. Click **Select > Images > Sel Image Mode**.

   Sel Image appears in the mode status area. Using this mode, you can select all instances of an image by clicking on one instance.

6. Click on U1.

   U1, U2, U3, and U4 are selected.

7. Click **Autoplace > Small Comp Pattern > Apply to Selected**.

The learned pattern is applied, and the power pins of C6, C7, and C8 are placed under the power pins on U2, U3, and U4.

8. Click **Select > UnSelect All Placement Objects**.

   Next you place the power pins of C9, C10, C11, and C12 under the power pins of U5, U6, U7, and U8 by repeating the steps from the previous section.

9. Press the right mouse button and select *Move Comp Mode*.

   You will move the power pin of C9 under the power pin of U8.

10. Click on C9.

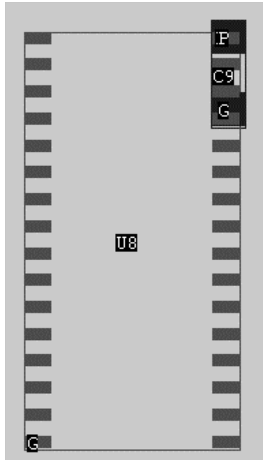
    You need to flip C9 to the back side.

11. Press the right mouse button and select *Flip*.

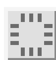
    Next you pivot C9 so that the power pin is at the top.

12. Press the right mouse button and select **Pivot > 90**.
13. Move C9 to the upper right of U8 (so that the power pin of C9 is under the power pin on U8).
14. Click to place the power pin of C9 under the power pin of U8.

    The following figure shows the relationship of C9 to U8.



Next you will learn and apply this component pattern to the unplaced capacitors.



15. Click the *Select Comp* button on the tool bar.

    Sel Comp appears in the mode status area.

16. Select *U8* and *C9* by dragging the pointer over them and releasing the left mouse button.

    The two components are selected, and *Selected:2* appears in the status area.

17. Click **Autoplace > Small Comp Pattern > Learn**.

    The autorouter stores orientation, location, and side information about the relationship between C9 and U8.

18. Unselect *C9* and *U8* by dragging the pointer over them.
19. Click **Select > Images > Sel Image Mode**.
20. Click on *U8*.

    U5, U6, U7, and U8 are selected.

21. Click **Autoplace > Small Comp Pattern > Apply to Selected**.

    The learned pattern is applied, and the power pins of C10, C11, and C12 are placed under the power pins of U5, U6, and U7.

22. Click **Select > UnSelect All Placement Objects**.

Congratulations! All components are placed.

Quitting the Autorouter and Saving Placement Results

When you are satisfied with your placement results, save your work. You can save your work in a placement file or a session file. If you save a placement file, you can reload the file at the start of a session or anytime during the session. The placement file is useful if you want to perform multiple placement trials and compare the files to choose the best result. You create a placement file by using **File > Write > Placement**.

If you create a session file, you load the session file only when you start the autorouter. A session file contains a reference to the original design filename as well as detailed placement, floorplan, swap, netlist, and route data. You will exit the autorouter and create a session file in the next procedure.

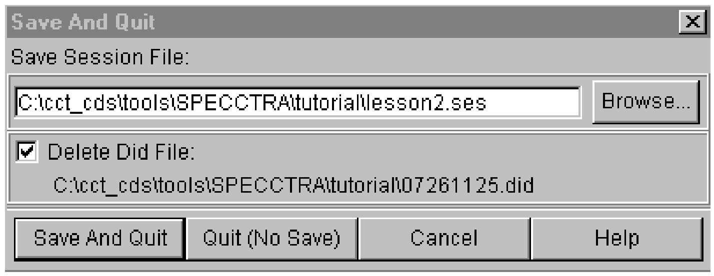
Quit the Autorouter and save a session file

1. Click **File > Quit.**

   The *Save And Quit* dialog box appears with *lesson2.ses* in the *Save Session File* data entry box, as shown in the following figure.

2. Click *Delete Did File* to remove the did file.

   In Lesson 1, you learned that a did file contains the command history from a the autorouter session. You do not need this file for the sesssion, so you will delete it.



3. Click *Save and Quit*.

The autorouter exits and saves the placement information in the session file.

What You Learned

In this lesson, you learned how to place components interactively and automatically. You learned

- The basic steps used for placing components
- How to set placement options
- How to set placement rules
- How to preplace connectors and critical components
- How to place large components
- How to edit the placement
- How to place small components

In the next lesson, you'll learn how to autoroute a PCB design.

Lesson 3: Routing a PCB Design

What You Will Learn

This lesson teaches you how to get the best results from the autorouter. In this lesson, you will learn how the autorouter works and how to

- Set routing rules
- Use a do file
- Monitor autorouting progress

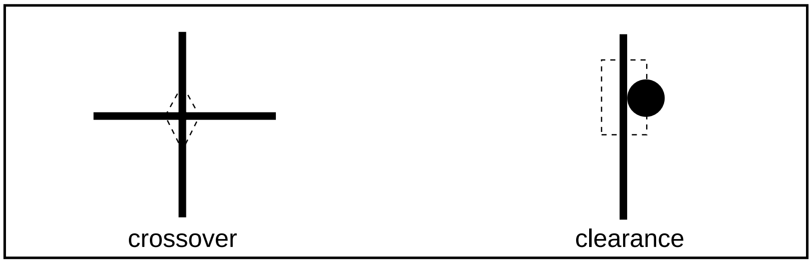
This lesson takes about 90 minutes to complete.

What to do Before You Begin

Before you begin this lesson, complete Lesson 1. You do not need to complete Lesson 2 but you should read the lesson to understand the automatic and interactive placement tools.

Understanding How the Autorouter Works

The autorouter works differently from other autorouters. It allows crossing and clearance conflicts during certain routing phases. These conflicts are shown in the following figure.



Crossing and Clearance Conflicts

During the first routing pass, the autorouter allows conflicts to route every connection. After the first routing pass, the cost for creating these conflicts increases with each pass.

During the first five routing passes, all connections are ripped up and rerouted. After the first five routing passes, the strategy changes, and only wires involved in conflicts are ripped up and rerouted. Wires that are not involved in conflicts are ignored during this phase.

This adaptive autorouting uses different routing functions and requires many routing passes. An important part of autorouting strategy involves choosing which routing commands to use and when to use them. You will learn more about these commands in this and the next lesson.

Using the four basic autorouting commands

In Lesson 1, you used the **smart_route** command to route a PCB design. This command uses four basic routing commands, which you can use individually. These commands are described in the following table.

Command	What it does
bus	Routes pins that share the same X or Y coordinate. This command is useful for rerouting memory arrays, backplanes, and other pins that share a common X or Y corrdinate.
fanout	Escapes SMD pads and through-pins to a via.
route	Routes with conflicts, and after the first five passes, reroutes only connections involved in conflicts. Escapes SMD pads to vias as needed.
clean	Rips up and reroutes all connections. Adding new conflicts is prohibited.

Note: Smart_route also uses **if**, **then**, and **else** commands with system variables to optimize routing.

Use the basic autorouting commands

1. Start the autorouter and load *lesson3.dsn*.

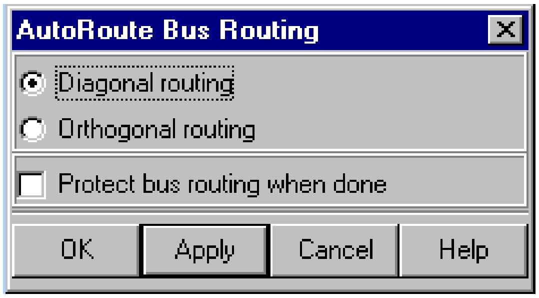
   You will route pins that share the same X or Y coordinate.

2. Click **Autoroute > Pre Route > Bus Routing**.

   The AutoRoute Bus Routing dialog box opens.

3. Make sure the *Diagonal routing* option is selected, as shown in the following figure.

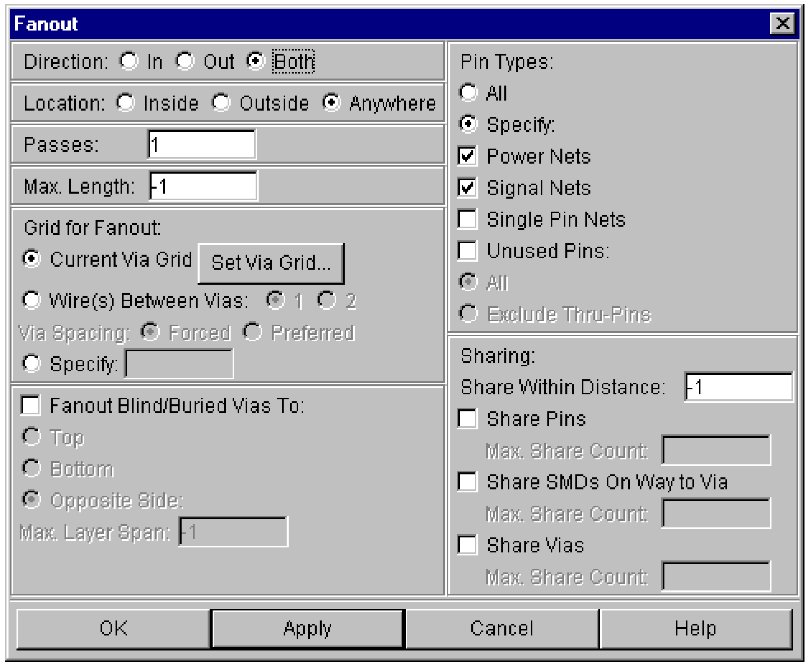
   This option uses diagonal wire segments, rather than orthogonal wire segments, to connect the pins.



4. Click *OK*.
5. Click **Autoroute > Pre Route > Fanout.**

   The Fanout dialog box opens.
6. Make sure *Specify, Power Nets, and Signal Nets* are selected under *Pin Types*.

   These choices escape all power pins and signal pins that interconnect with one or more pins.
7. Make sure all other options under *Pin Types* are unselected, as shown in the following figure.



8. Click *OK* and wait for fanout to complete.

   Next you will automatically route the design.

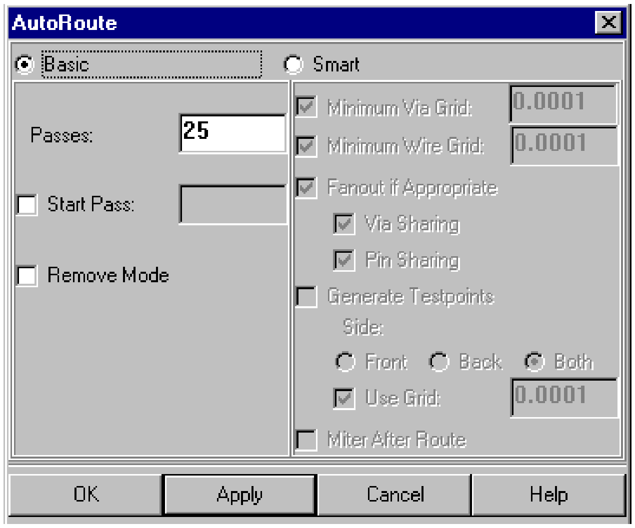
9. Click **Autoroute > Route.**

   The AutoRoute dialog box opens.

10. Click the *Basic* option in the AutoRoute dialog box, as shown in the following figure.

    This option runs basic route passes instead of using smart_route.

11. Note the default number of passes (25) used for basic routing.



12. Click *OK*.

    The **route** command applies 25 routing passes. The routing takes several minutes to complete.

13. Notice the yellow rectangles and diamond shapes that mark the temporary conflicts during autorouting. The rectangles indicate clearance violations. The diamond shapes indicate crossovers.

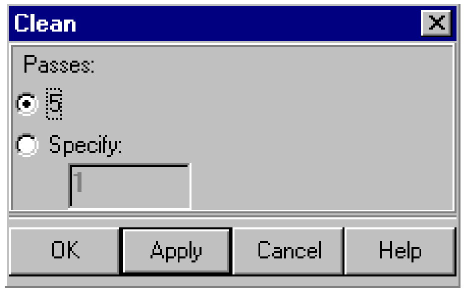
    While the tool is routing the design, the *Pause* button appears in the control area.

14. Wait for *Idle* to replace the *Pause* button in the control area.

    Use the **clean** command to rip up and reroute all connections. This command improves manufacturability by removing unnecessary vias and bends, reducing routed length, and changing SMD entries and exits.

15. Click **Autoroute > Clean**.

    The Clean dialog box opens, as shown in the following figure.

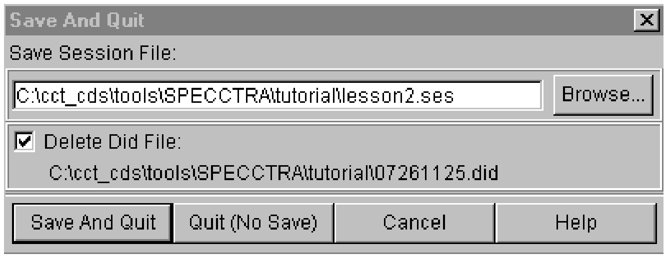


16. Click *OK*.
17. Wait for *Idle* to replace the *Pause* button in the control area.

18. Click the *Status Report* button on the tool bar.
19. View the report and determine the actual number of route passes used to route the design.

    If routing completes before all route passes are used, the remaining route passes are skipped and the clean passes are run.
20. Close the status report.
21. Click **File > Quit.**

    The Save and Quit dialog box opens, as shown in the following figure.
22. Click *Delete Did File* to remove the did file.



23. Click *Quit (No Save)* to exit the autorouter without saving a session file and a did file.

Note: Another way to exit the autorouter without saving a session file is to enter **quit** in the command entry area. The did file is not deleted when you use this method.

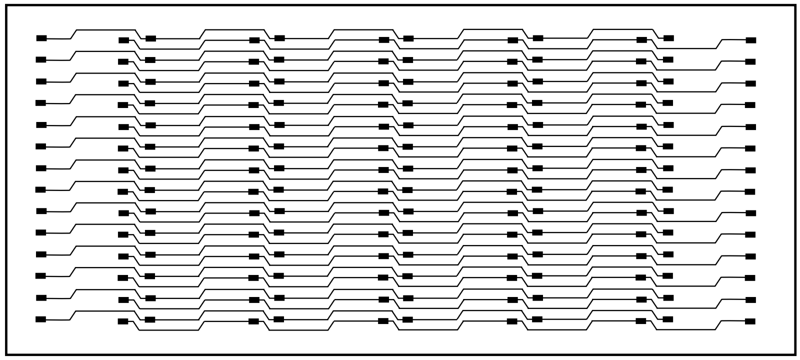
Each basic autorouting command serves a special purpose and is used at a specific phase of the autorouting session. The three phases of autorouting are

- Prerouting
- General purpose routing
- Post-routing

Understanding prerouting commands

Prerouting commands, as the name suggests, are executed before route and clean commands. The **bus** and **fanout** commands are prerouting commands.

The **bus** command routes pins that share the same X or Y coordinate and are attached to the same net. Because of its simplicity, connections are routed quickly, and use a minimum of space. The **bus** command is used at the beginning of an autorouting session, before you use other basic autorouting commands. The following figure shows the results of the **bus** command with the diagonal option turned on.



Bus Diagonal is Fast and Produces Space Saving Wiring Patterns

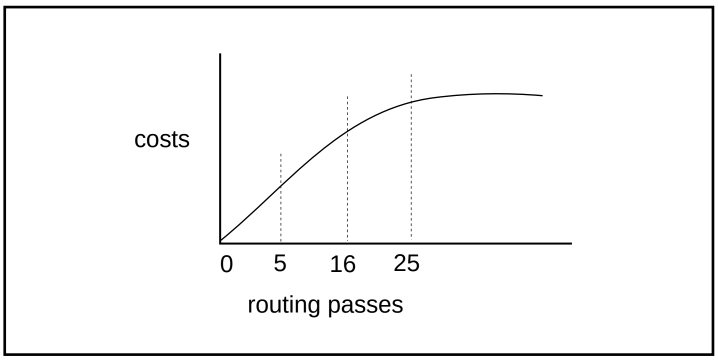
The **fanout** command routes short escape wires and vias from SMD pads and through-pins marked as exposed. This command adds a via that the autorouter can connect to instead of connecting directly to the pad or pin. For power pins that connect to decoupling capacitors, you can control the fanout order by using the **power_fanout** rule. You can choose whether you want the escape wire to connect to the escape via first or the decoupling capacitor first.

The **fanout** command should follow the **bus** command but precede the route command. You can use fanout with up to five passes. Use multiple fanout passes with via and pin sharing enabled to optimize SMD escapes.

Understanding the general purpose routing commands

The general purpose autorouting commands are **route** and **clean**. These commands can route and reroute all connections in the design, except those routed by the **bus** command.

The **route** command uses costs that limit or control resources. These costs control the number of vias per connection, the number of conflicts per connection, and the distance that wires can route on a layer in the wrong direction. The autorouter dynamically changes these costs over a series of route passes based on the routing history and the current wiring conditions. The following figure shows approximately when routing costs change during a series of 25 route passes.



Costs Change During a Series of Route Passes

Because costs change during a series of route passes, you must specify a sufficient number of passes. After pass five, only connections involved in conflicts are rerouted. If the wiring completes before all passes are used, the extra passes are skipped.

The **clean** command works differently from the **route** command. This command rips up and reroutes all connections (except those routed by the **bus** and **fanout** commands) instead of only the connections involved in conflicts.

The **clean** command is usually applied over two to four passes and should always follow a series of route passes. Because **clean** rips up and reroutes all connections, tries to reroute them using a different path, and does not add new conflicts, it usually completes more connections, reduces routing length and conflicts.

A typical sequence of basic autorouting commands is

```
bus diagonal
fanout 5
route 25
clean 2
route 50 16
clean 4
```

During clean passes, the built-in costs for vias and wrong-way routing are highest, thus improving the quality of routing after each clean pass.

Note: In the typical sequence of basic autorouting commands, four clean passes are included at the end. Always use at least four clean passes after all routing completes.

Understanding the autorouter's smart_route command

The **smart_route** command sets wire and via grids, performs bus and fanout operations, and runs a series of route and clean passes until routing completes.

A single **smart_route** command can replace the four basic autorouting commands, because **smart_route** combines **bus**, **fanout, route**, and **clean** commands. The four basic autorouting commands are used instead of smart_route if rules must change between bus, fanout, route, or clean passes or if other commands are needed between these different routing functions.

While **smart_route** is running, the autorouter monitors and analyzes routing progress. If it detects design problems, warning or error messages appear. If the autorouter reaches a point where further improvement is unlikely, the autorouter switches to a different method and completes as many connections as possible with zero conflicts.

In the following procedure, you create an artificial design problem by unselecting all signal layers. Smart_route detects the problem and generates an error message. If there are not enough signal layers in a design, the same error message is displayed.

Use smart_route to determine design problems

1. Start the autorouter and load *lesson3.dsn*.

2. Click the *Layer* button on the tool bar.

   The Layer panel opens.

3. Click the SIGNAL_1 layer selection button located near the top of the layers panel.

   On a Windows platform, the selection button looks like .

On a UNIX platform, the selection button looks like  .

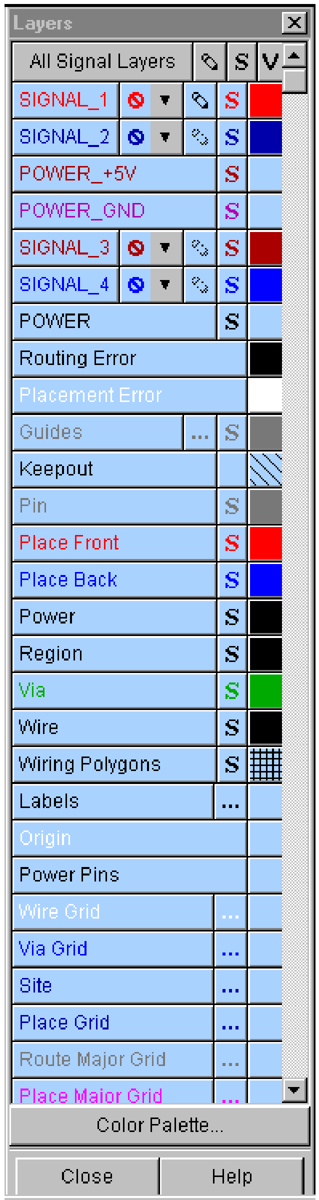
4. Click on the  button.

   This symbol means the layer is unselected.

5. Repeat the previous step to unselect SIGNAL_2, SIGNAL_3, and SIGNAL_4 layers.

   By unselecting all signal layers, you create a design problem.

The layer panel is shown in the following figure.



6. Click **Autoroute > Route**.

   The AutoRoute dialog box opens.

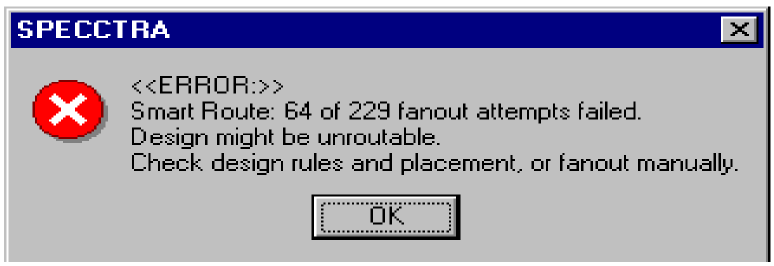
7. Make sure *Smart* is selected.

8. Click *OK*.

   After several routing attempts, an error message displays.

9. Read the error message and click *OK*.

   The following figure shows an example of the error message.



A Message Popup Displays an Error Message that Indicates a Design Problem

Next you will select all four signal layers to make them available for routing. You will enter **smart_route** from the keyboard instead of using the GUI. Smart_route analyzes early routing results and determines that you might have more routing layers than necessary to route the design.

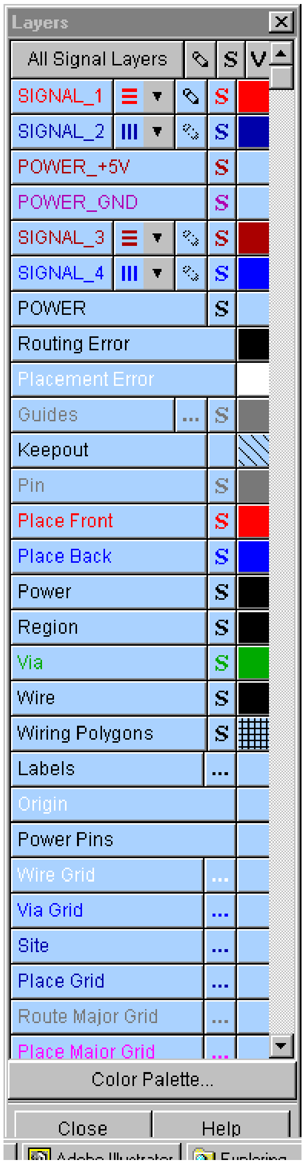
Use smart_route to optimize your design

1. Click the SIGNAL_1 layer selection button and set the routing direction to horizontal.

2. Click the SIGNAL_2 layer selection button and set the routing direction to vertical.

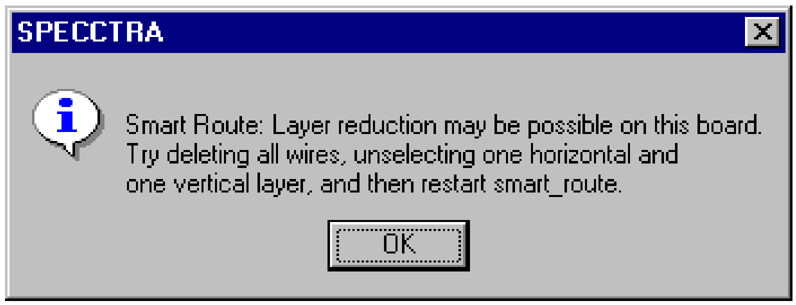
3. Repeat steps 1 and 2 to set SIGNAL_3 to horizontal and SIGNAL_4 to vertical.

The Layer panel is shown in the following figure.



4. Close the Layer panel.
5. Enter **smart_route** in the command entry box below the routing status line and press the return key.

   After several routing passes, a message popup tells you that you might be able to reduce the number of layers. An example of the message popup is shown in the following figure.



Smart_route Message Popup Indicates that Layer Reduction is Possible

6. Click *OK* to close the message popup.
7. Wait for *Idle* to replace the *Pause* button in the control area.
8. Enter **quit** in the command entry.

This command exits the autorouter without prompting you for a confirmation.

Understanding the post-routing commands

Post-routing commands refine the wiring and add test points after routing is complete. The autorouter's post-routing commands are **spread**, **miter**, **testpoint**, and **recorner**. The following table provides a brief description of the post-routing commands.

Post Routing Commands

Command	What it does
spread	Adds extra clearance between wires
testpoint	Adds test points
miter	Changes 90 degree corners to 45 degree diagonals using a range of setback values
recorner	Changes 90 degree corners to 45 degree diagonals

All post-routing commands, except **recorner,** are DFM options. An explanation of the DFM post-routing options is beyond the scope of this tutorial. See the online help for more information about using the **spread**, **miter**, and **testpoint** commands.

When you use the **recorner** command, you must include a corner type and a setback value. For example, the following command changes wrong way routes (slant) to 45 degree diagonals, if the slants are one inch or more in length.

```
recorner slant 1
```

The easiest way to change all 90 degree corners to 45 degree diagonals is to use **recorner diagonal** with three setback values. An example of how you use this command is

```
unit inch
recorner diagonal 1 1 1
```

In this example, the three setback values change 90 degree corners that meet the one inch criteria to 45 degrees.

The **recorner** command is iterative. If at least one corner in the design changes to a diagonal, all remaining corners are attempted again. This iterative process continues until no further changes occur.

You achieve optimum results when you use a series of **recorner** commands with progressively smaller setback values. In the following procedure, you enter two **recorner** commands, and use setback values for the second that are half the value of the first.

Change 90 degree corners to 45 degree diagonal corners

1. Start the autorouter and load *lesson3.dsn*.
2. Click **File > Read > Routes**.

   The Read Routes dialog box opens.
3. Click the *Browse* button.

   The Open dialog box opens for Windows platforms.

   The Select File dialog box opens for UNIX platforms.
4. Change to the autorouter's tutorial directory for Windows platforms and open *recorner.rte*.

   Select *recorner.rte* and click *OK* for UNIX platforms.

   The filename is added to the Read Routes dialog box.
5. Click *OK*.

   The routes file loads.

Note: If you have the DFM option, you can use a single miter command to replace steps 6 through 8.

---

6. Enter **recorner diagonal 1 1 1** in the command entry box.

   All 90 degree corners that meet the one inch setback criteria change to 45 degrees.
7. Wait for *Idle* to replace the *Pause* button in the control area.
8. Enter **recorner diagonal 0.5 0.5 0.5**.

9. Wait for *Idle* to replace the *Pause* button in the control area.
10. Examine the wiring changes made by the two previous recorner commands.
11. Enter **quit** in the command entry box to exit the autorouter.

With more complex designs, the following command series is suggested.

unit inch
recorner diagonal 1 1 1
recorner diagonal 0.5 0.5 0.5
recorner diagonal 0.25 0.25 0.25
recorner diagonal 0.125 0.125 0.125

You can use other units and other values, but for best results always use a series of recorner commands. Start with a large setback value, and halve the previous value for the next recorner command in the series.

Setting Routing Rules

The autorouter supports a large number of routing rules. Routing rules can be set at many different levels. The routing rule levels are listed from highest to lowest precedence in the following table.

Rule level	What it does
region class to class	Sets rules that apply between net classes when the nets are routed in a region.
region net	Sets rules that apply to nets in a region.
region class	Sets rules that apply to a class of nets in a region.
region	Sets rules for an area of the PCB.
padstack	Sets clearance rules for via padstacks.
class to class layer	Sets rules that apply between net classes when the nets are routed on particular layers..
class to class	Sets rules that apply between wires, pins, and vias of one or more net classes.
fromto layer	Sets rules that apply to connections if they are routed on a particular layer.
fromto	Sets rules for pin-to-pin connections.
group layer	Sets rules that apply to a group of fromtos if the fromtos are routed on a particular layer.
group	Sets rules for a group of fromtos.
net layer	Sets rules that apply to nets if they are routed on a particular layer.
net	Sets rules that apply to nets.
group set layer	Sets rules that apply to a set of groups when they are routed on a particular layer.
group set	Sets rules that apply to a set of groups.
class layer	Sets rules that apply to a class of nets if the nets are routed on a particular layer.

Rule level	What it does
class	Sets rules that apply to a class of nets.
layer	Sets rules that apply to wires routed on a particular layer.
pcb	Sets global rules for all nets in the design. This is the lowest precedence rule.

Understanding the autorouter's rule hierarchy

The rule levels described in the previous table form a hierarchy. Higher level rules always override lower level rules that are set for the same physical objects. For example, a wire-to-wire clearance rule for a net overrides a wire-to-wire clearance rule for a class that includes that net.

Rules at the PCB level have the lowest precedence in the hierarchy. Rules at the region class to class level have the highest precedence in the hierarchy. The order of precedence is fixed in the autorouter and cannot be changed.

Setting width and clearance rules

You can set width and clearance rules at different levels of the hierarchy and for different objects. The object types supported by the autorouter are described in the following table.

Object type	**What it means**
area	Keepout regions and PCB boundary
pad	SMD pad
pin	Through-pin
via	Layer feed-through
wire	Copper path

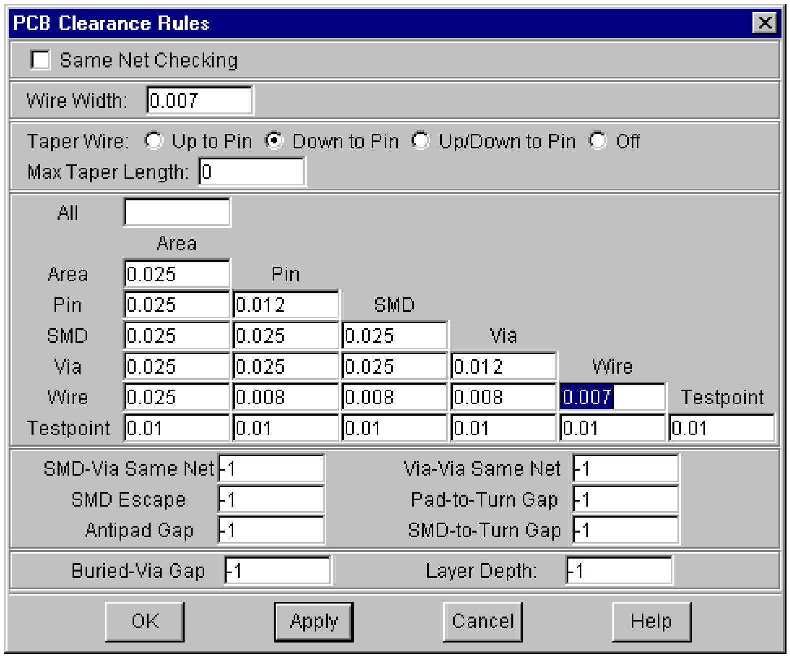
Two object types separated by the underscore character indicate an object-to-object rule. For example, wire_smd is used to set clearance rules between wires and SMD pads.

Set width and clearance rules at the PCB level

1. Start the autorouter and load *lesson3.dsn*.
2. Click **Rules > PCB > Clearance**.

   The PCB Clearance Rules dialog box opens, as shown in the following figure.
3. Enter **0.007** in the *Wire Width* data entry box.
4. Enter **0.007** for *wire_wire clearance*.

   This data entry box is located in the row labeled Wire under the column labeled Wire.



5. Click *OK*.

   The width and clearance rules that you changed in the PCB Clearance Rules dialog box are applied.

6. View the session transcript in the output window and find the following commands:

   rule pcb (clearance 0.007 (type wire_wire))
   rule pcb (width 0.007)

7. Enter **quit** in the Command entry box to exit the autorouter.

In Lesson 4, you will learn to set additional routing rules.

Autorouting With a Do File

There are three methods for issuing commands and controlling the autorouter. You can

- Use the GUI
- Enter commands from the keyboard
- Use a do file

Until now, you either used the GUI to run commands or you entered commands from the keyboard. The preferred method for controlling the autorouter is with a do file. A do file is a text file that contains one or more autorouting commands. Each command occupies a separate line in the do file. Commands are organized in the order you want them to run from the start of the file to the end.

The do file also serves another purpose. It serves as a record of the rules and commands you use during an autorouting session. If you need to apply a design revision to a completed design, you can edit the original do file and reuse it.

You start with a basic do file that includes commands that address the following:

- Setup and file management
- Rule setting
- Prerouting

- Routing
- Post-routing

Using a basic do file

A basic do file (basic.do) is included with the tutorial files. A listing of the commands in *basic.do* follows. The PCB width and clearance rules are not intended as typical values but are included as examples.

```
bestsave on $\best.wir
status_file $\route.sts
unit mil
rule pcb (clearance 7 (type wire_wire))
rule pcb (width 7)
smart_route
write routes $\final.rte
report status $\final.sts
```

Note: The $\ notation represents the path for the file in the autorouter. When you want to write a report or data file to the same directory where the design file is located, use the $\ notation. On UNIX platforms, substitute the ***/*** character for the **** character.

Often, you do not need to include PCB clearance and width rules in the do file, because they are translated from your layout system and embedded in the design file. If you want to override rules that were set in the layout system or you want to apply other rules, add your rules to the do file. Make sure these rules precede all routing commands.

Use a basic do file to autoroute a PCB design

1. Start the autorouter and enter the path for *lesson3.dsn* in the Design/Session File data entry box.
2. Enter *$\basic.do* in the Do File data entry box located on the dialog box. On a UNIX platform, substitute / for \.

3. Click *Start SPECCTRA*.

   The autorouter starts, loads the design file, and runs the do file. A Pause button appears in the control area.

   After several routing passes, a message popup tells you that you might be able to reduce the number of layers.

4. Click *OK* to close the message popup.
5. Wait for *Idle* to replace the *Pause* button in the control area.
6. View the session transcript in the Output window, and verify that all commands were run.

Note: If a do file contains a syntax error, the do file stops working and the error is reported in the Output window.

7. Click the *Status Report* button on the tool bar.
8. Browse the status report and check the routing results.
9. Close the report.
10. Click **Rules > PCB > Clearance**.

    The PCB Clearance Rules dialog box opens.

11. Notice the width and wire-to-wire clearance rules are updated and reflect the 7 mil rules that were set by basic.do.
12. Click *Cancel* to close the dialog box.
13. Enter **quit** in the Command entry box to exit the autorouter.

Creating a do file

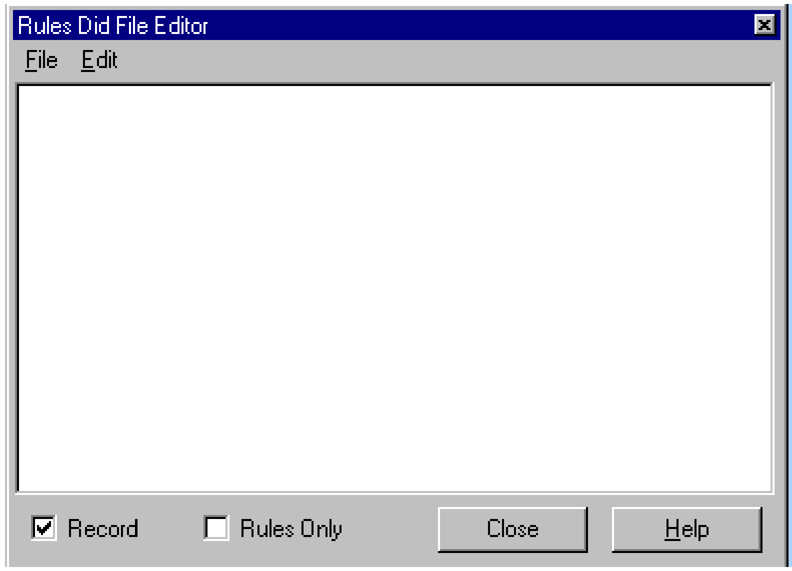
In the previous steps, you used the basic do file that is supplied as part of this tutorial. Next you learn an easy method for creating your own do files.

After you become more familiar with the autorouter, you can construct a do file from your memory of commands and syntax. While you are learning, you can use the autorouter's built-in Rules Did File editor to capture and edit syntactically correct commands. The editor records rule setting commands and other commands that you enter through the GUI. You can record these commands to create a syntactically correct do file. You can control which types of commands are recorded and edit the recorded commands.

Create a do file with the Rules Did File Editor

1. Start the autorouter and load *lesson3.dsn*.
2. Click **Edit > Rules Did File**.

   The Rules Did File Editor opens, as shown in the following figure.
3. Remove the check from the Rules Only box.



Note: By default, the autorouter records only the rule setting commands when Rules Only is checked. Remove the check from Rules Only to record other commands. When you close the dialog box, the settings remain in effect. For this procedure, you will leave the dialog box open.

Next you use the **File > Bestsave** command to save routed wires at the end of a routing pass if routing improved since the previous save.

4. Click **File > Bestsave**.

   The Bestsave dialog box opens, as shown in the following figure.

5. Enter **$\best.wir** in the data entry box.

   On a UNIX platform, substitute / for \.

6. Click the *Enable* option on the Bestsave dialog box.

7. Click *OK*.

   Notice that the command is recorded in the Rules Did File Editor.

   Next you will change the measurement unit to mils.

8. Click the *Unit* button (near the bottom right corner of the window) and choose *mil*.

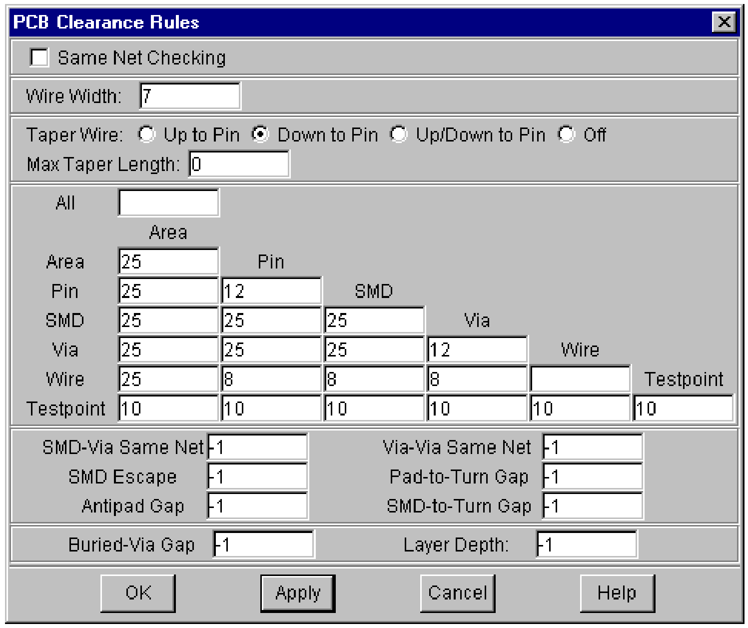
9. Click **Rules > PCB > Clearance**.

   The PCB Clearance Rules dialog box opens, as shown in the following figure.

10. Enter **7** in the *Wire Width* data entry box.

11. Enter **7** in the *wire_wire clearance* data entry box.

    The *wire_wire clearance* data entry box is in the row labeled Wire, under the column labeled Wire.



12. Click *OK*.
13. Click **Autoroute > Route**.

    The AutoRoute dialog box opens.
14. Make sure the *smart* option is selected.
15. Click *OK*.

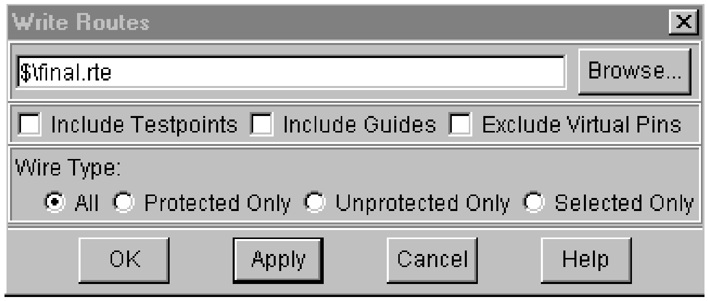
    Next locate the *Pause* button, which appears on the left side of the status area. You use this button if you want to pause and then stop the autorouter. You will use this button now because you are creating a do file, rather than autorouting the design to completion.
16. Click the *Pause* button and then click *Stop*.

    The autorouter pauses then halts.
17. Click **File > Write > Routes**.

    The Write Routes dialog box opens, as shown in the following figure.

18. Enter **$\final.rte**

    On a UNIX platform, substitute / for \.



19. Click *OK*.

    You save the routes file even though it is incomplete to record the command in the Rules Did File editor.

20. Review the recorded commands in the Rules Did File editor.

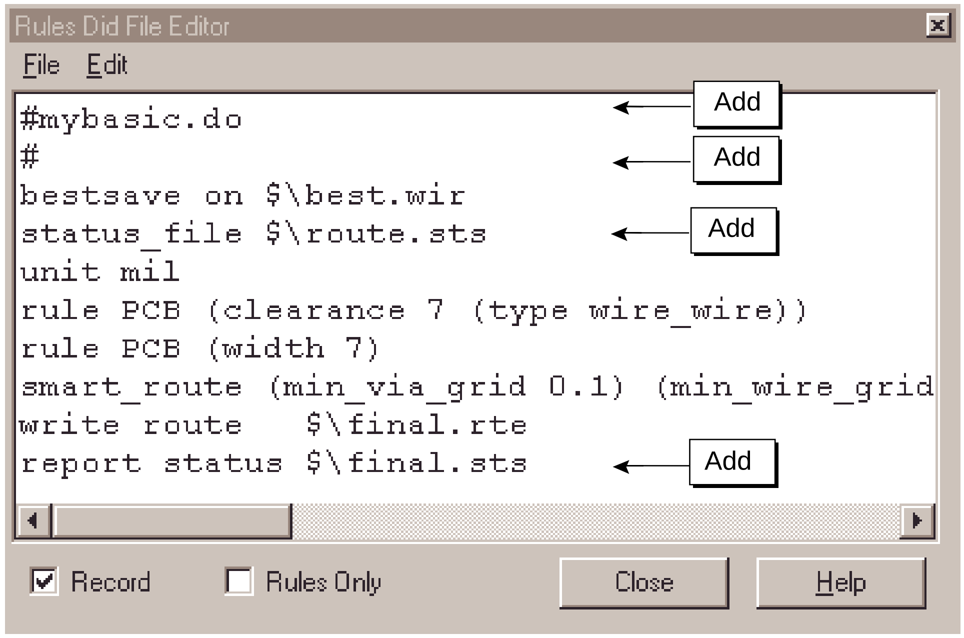
Edit the do file with the Rules Did File Editor

In the previous procedure, you used the Rules Did File editor to record commands that you entered through the GUI. You can record all commands this way. In the following procedure, you will add commands and comments to the editor by them.

Note: A pound sign (#) precedes a comment. The autorouter ignores comments in do files.

---

1. Add the following commands and comments to the Rules Did File editor. On UNIX platforms, substitute the **/** character for the **** character.



2. Save the file in the tutorial directory as *mybasic.do*.
3. Close the Rules Did File editor.
4. Enter **quit** in the Command entry box to exit the autorouter.

Monitoring Autorouting Progress

In Lesson 1 you learned about the status report and about the kinds of information it contains. Now you will learn how to use the status report to monitor autorouting progress and determine when and how to adjust your strategy.

The **smart_route** command uses the same information that is recorded in the status report to analyze autorouting progress. Progress is gauged by monitoring *failures*, *unroutes*, and *conflict reduction* from pass to pass. These terms are explained in the following table.

Term	What it means
conflict	Conflicts consist of two types. Crossing conflicts occur when two wires cross. Clearance conflicts occur when a wire or via violates a clearance rule.
failure	A failure occurs when the autorouter attempts to route or reroute a connection, fails, and restores the connection to its prior state.
reduction %	Reduction % indicates the percentage of conflict reduction compared to the previous pass.
unroute	An unroute is a connection that is not routed.

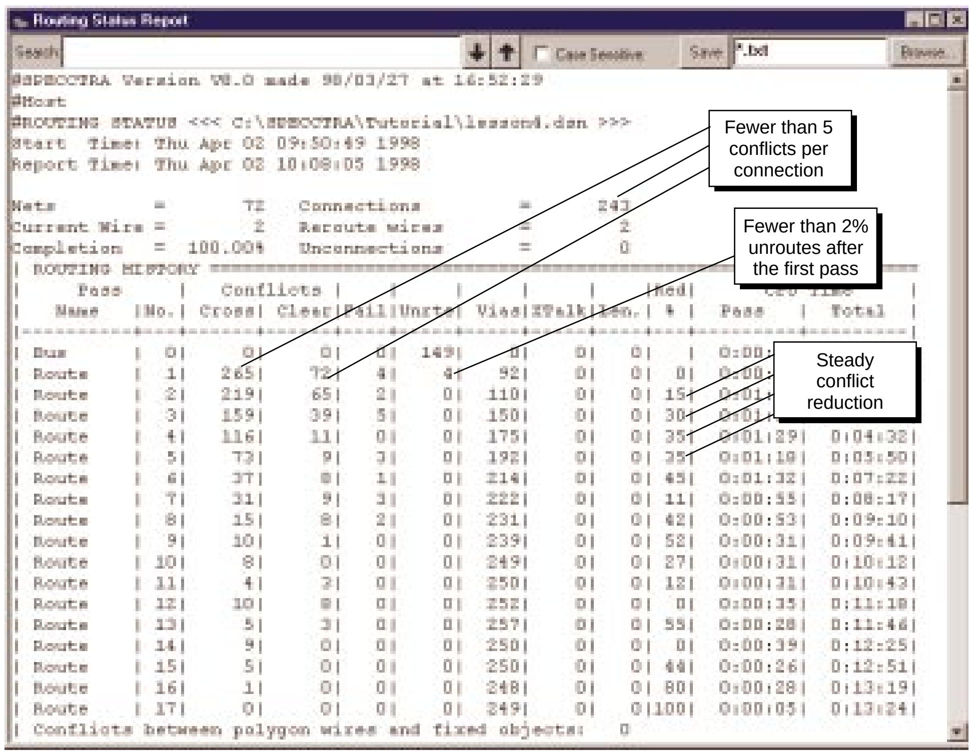
Using the status report

The Routing History table is updated in the status report for each routing pass. This table contains the information you need to analyze routing progress.

When you use **smart_route**, it analyzes the status information for you and alerts you when potential problems are detected or when a design reduction might be possible. If you use basic autorouting commands instead of **smart_route**, *you* must monitor and analyze the status report to determine whether adjustments are needed. To gauge whether the autorouter is making good progress and determine what must be done if progress is poor, you need to understand the indicators of both good and poor routing progress.

Analyzing conflicts

The number of conflicts after the first route pass is a good indication of whether a design can be routed completely. When the sum of crossing and clearance conflicts is less than five times the total number of connections, routing is likely to complete. The following figure shows a status file with a good ratio between total conflicts and total connections after the first route pass.




```
#SPECCTRA Version V8.0 made 98/03/27 at 16:52:29
#Host
#ROUTING STATUS <<< C:\SPECCTRA\Tutorial\lesson4.dsn >>>
Start  Time: Thu Apr 02 09:50:49 1998
Report Time: Thu Apr 02 10:08:05 1998

Nets         =       72   Connections        =      243
Current Wire =        2   Reroute wires      =        2
Completion   =  100.00%   Unconnections      =        0
| ROUTING HISTORY ==========================================================
|    Pass      |   Conflicts  |    |     |     |     |    |Red|   CPU Time    |
|   Name   |No.| Cross| Clear|Fail|Unrte| Vias|XTalk|Len.| % |  Pass  | Total |
|----------+---+------+------+----+-----+-----+-----+----+---+--------+-------|
| Bus      |  0|     0|     0|   0|  149|    0|    0|   0|   | 0:00:
| Route    |  1|   265|    72|   4|    4|   92|    0|   0|  0| 0:00:
| Route    |  2|   219|    65|   2|    0|  110|    0|   0| 15| 0:01:
| Route    |  3|   159|    39|   5|    0|  150|    0|   0| 30| 0:01:
| Route    |  4|   116|    11|   0|    0|  175|    0|   0| 35| 0:01:29| 0:04:32|
| Route    |  5|    73|     9|   3|    0|  192|    0|   0| 35| 0:01:18| 0:05:50|
| Route    |  6|    37|     8|   1|    0|  214|    0|   0| 45| 0:01:32| 0:07:22|
| Route    |  7|    31|     9|   3|    0|  222|    0|   0| 11| 0:00:55| 0:08:17|
| Route    |  8|    15|     8|   2|    0|  231|    0|   0| 42| 0:00:53| 0:09:10|
| Route    |  9|    10|     1|   0|    0|  239|    0|   0| 52| 0:00:31| 0:09:41|
| Route    | 10|     8|     0|   0|    0|  249|    0|   0| 27| 0:00:31| 0:10:12|
| Route    | 11|     4|     3|   0|    0|  250|    0|   0| 12| 0:00:31| 0:10:43|
| Route    | 12|    10|     8|   0|    0|  252|    0|   0|  0| 0:00:35| 0:11:18|
| Route    | 13|     5|     3|   0|    0|  257|    0|   0| 55| 0:00:28| 0:11:46|
| Route    | 14|     9|     0|   0|    0|  250|    0|   0|  0| 0:00:39| 0:12:25|
| Route    | 15|     5|     0|   0|    0|  250|    0|   0| 44| 0:00:26| 0:12:51|
| Route    | 16|     1|     0|   0|    0|  248|    0|   0| 80| 0:00:28| 0:13:19|
| Route    | 17|     0|     0|   0|    0|  249|    0|   0|100| 0:00:05| 0:13:24|
| Conflicts between polygon wires and fixed objects:   0
```

Indicators Show the Design is Routable

If the total number of conflicts after the first pass is more than five times the total number of connections, this usually indicates the design will not route because of one or more of the following reasons.

Cause	Meaning	Solution
Insufficient routing layers	Layers are unselected or too few signal layers are defined in the layout system.	Use the Layers panel to determine whether signal layers are unselected.
Layer routing directions are incorrect	An insufficient number of vertical or horizontal routing layers are defined. The autorouter eventually resolves this, but it takes longer to route the design.	Use the Layers panel to determine whether signal layer directions are set correctly.
Power and ground nets are being routed	Power and ground nets are not defined as copper planes, and the autorouter is routing power and ground connections.	Use the Layers panel to determine whether power layers are defined correctly. If power layers are defined correctly, you cannot select or unselect the layers and cannot set the routing direction in the Layers panel. If power layers are not defined correctly, fix the problem in the layout system and re-extract the design.
Clearance rules are incorrect	An error was made in setting clearance rule	Make sure that the rule values are set

Cause	Meaning	Solution
	values.	correctly for the current unit setting.
Prerouted wires are protected	Protected preroutes prevent the autorouter from ripping up and rerouting connections to eliminate conflicts.	Try autorouting with the preroutes unprotected.

Use the status report to determine a design problem

1. Start the autorouter and load *conflict.dsn*.

   If a warning dialog box appears, close it.

2. Enter **route 1** in the command entry box.

   The *Pause* button appears in the status bar.

3. Wait for *Idle* to replace the *Pause* button in the control area.

4. Click the *Status Report* button on the tool bar.

5. Observe the number of crossing and clearance conflicts and the failures in the Routing History table.

   The number of failures indicates a design problem, which in this case are the result of too few routing layers.

6. Enter **quit** in the command entry box to exit the autorouter.

Analyzing failures

A failure occurs when the autorouter cannot route or reroute a connection. Routing failures are the same as unroutes during the first route pass and can indicate a difficult design, a design error, or unrealistic design rules.

There are usually no failures during the first route pass, because the cost for allowing conflicts is lower during the first pass than at any other time. If a small percentage of the total connections (less than 2%) fail during the first pass, this could indicate a difficult design. If more than 2% of the total connections fail during the first pass (and the total number of conflicts is high), the cause is usually a design problem.

Failures that occur during and after the second route pass indicate the number of connections that could not be routed with a different path and were restored to their previous state. Failed connections after the first route pass are not necessarily a problem, unless they accompany excessive conflicts or unroutes.

Analyzing conflict reduction

After the first route pass and through pass five, the total number of conflicts should reduce by at least 30% per pass. If conflict reduction is less than 30% per pass during this phase, there can be several causes.

Cause	Meaning	Solution
Difficult design	The design is very dense, and a larger than usual number of routing passes is required to route to 100%.	Increase the total number of routing passes. Using several hundred routing passes with difficult designs is not unusual.
Insufficient routing layers	An insufficient number of signal layers is defined or layers are unselected.	Use the Layers panel to determine whether signal layers are unselected.

Cause	Meaning	Solution
Autorouter is via starved	The via grid is too coarse, which prevents the autorouter from eliminating conflicts by adding vias.	Set the via grid to zero or to a minimum resolution. For example, the following commands set both wire and via grids to 1 mil. unit mil grid smart (via 1)(wire 1)

After pass five, conflict reduction proceeds at a slower pace. Although the percentage of reduction is usually much less for each pass of this phase, conflicts should always follow a downward trend over a series of ten passes.

Analyzing unroutes

Unroutes should not exist after the fifth routing pass. If unroutes remain after the fifth route pass, the unroutes will probably remain at the conclusion of the autorouting session. The typical causes for unroutes after route pass five are explained in the following table.

Cause	Meaning	Solution
Keepouts block pins	Pins covered by keepouts cannot be routed.	Check the unrouted connections to determine whether they are blocked or covered by keepouts.

Cause	Meaning	Solution
Component pins overlap	If components are placed incorrectly and pins overlap, the pins cannot be routed.	Make sure the pins do not overlap.
Pins are outside the PCB boundary	Pins that are outside or too near the PCB boundary cannot be routed.	Check the unrouted connections to determine whether they are either outside or too near the PCB boundary.

In this section, you learned how to detect autorouting problems and what you can do about them. Use this information, to solve routing problems when you are routing your designs.

What You Learned

In this lesson, you learned how to automatically route a PCB design. You learned how the autoroute works and how to

- Set routing rules
- Use a do file
- Monitor autorouting progress

In the next lesson, you will learn how to set rules and control the autorouter.

Lesson 4: Setting Rules and Controlling the Autorouter

What You Will Learn

This lesson teaches you how to set wiring rules and use commands that control the way a design is routed. This lesson differs from the previous lessons because you enter most commands from the keyboard instead of using the GUI.

In this lesson you will learn how to

- Set rules at different levels in the rules hierarchy
- Set wire and via clearance rules for different objects
- Select layers for routing
- Set layer routing direction
- Select vias for routing
- Select connections for routing
- Use routing keepins
- Control whether nets are routed with starburst or daisy-chain wiring
- Define a class of nets and a group of fromtos
- Assign rules to classes and groups
- Assign routing priorities

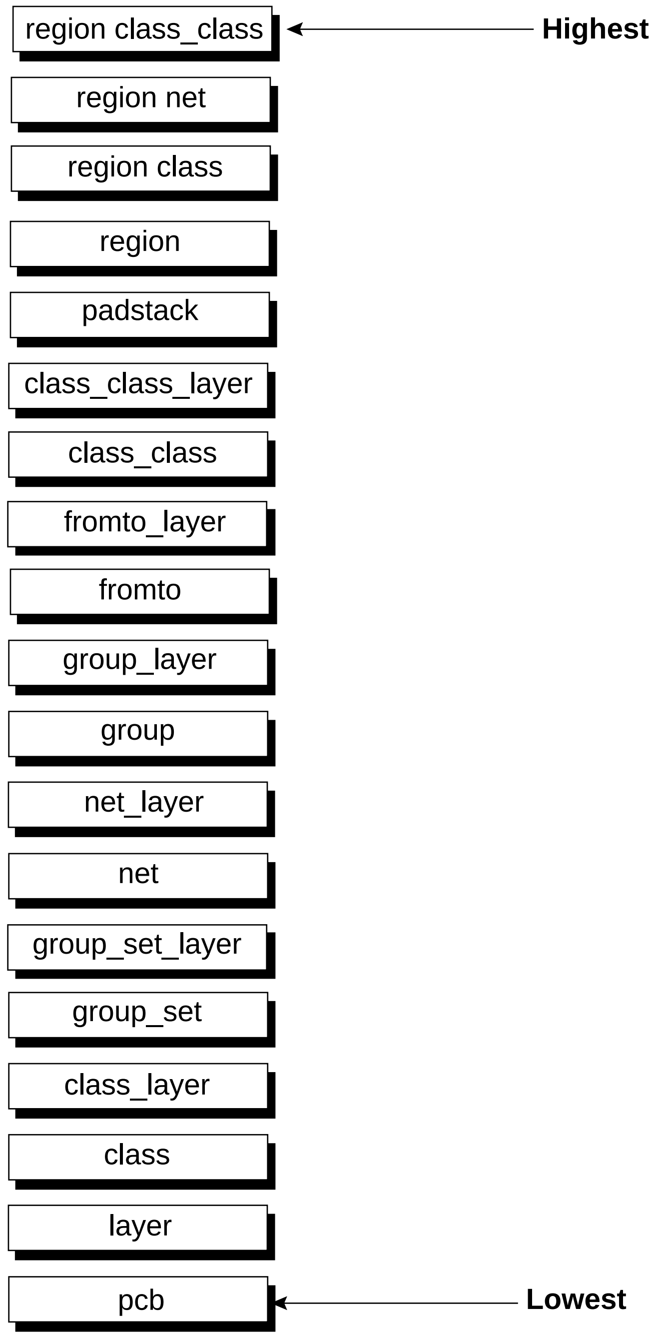
This lesson takes about 60 minutes to complete.

What to do Before You Begin

Before you begin this lesson, complete Lesson 3 “Routing a PCB Design.”

Understanding the Autorouter’s Hierarchical Rules

In Lesson 3, you learned that you can set rules at 19 different levels to control how a design is routed. The following figure shows the 19 rule levels.



The Rule Hierarchy Offers 19 Levels of Control

Rules set at higher levels override those set at lower levels, when they apply to the same object. For example, if you set the following rules:

unit mil
rule pcb (clearance 8 (type wire_wire))
rule net (CLK (clearance 10 (type wire_wire))

all nets obey the pcb 8 mil wire-to-wire clearance rule, except net CLK, which obeys the 10 mil clearance rule. Because both rules control wire-to-wire clearance, the higher precedence net rule overrides the pcb rule for the net CLK.

If you set the following rules:

unit mil
rule pcb (clearance 8 (type wire_smd))
rule net CLK (clearance 10 (type wire_pin))

the CLK net obeys both the 8 mil wire-to-smd PCB clearance rule and the 10 mil wire-to-pin net clearance rule. Both rules are followed, because they are set for different objects and they do not conflict.

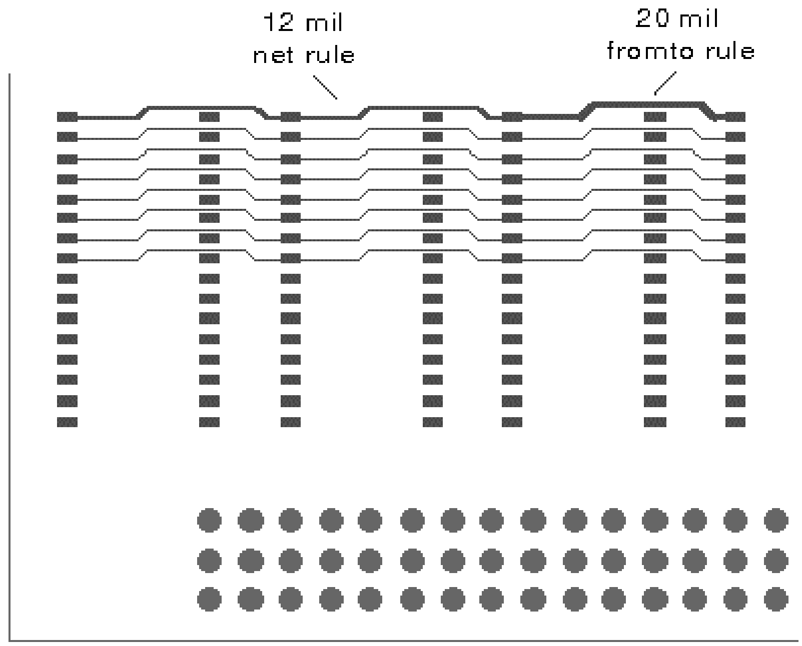
In the following procedure, you set wire width rules at the pcb, net, and fromto levels.

Set wire width rules at the pcb, net, and fromto levels

1. Start the autorourter and load *lesson4.dsn*.
2. Type **rule pcb (width .008)** in the command entry box.
3. Type **rule net sig1 (width .012)**.
4. Type **define (net sig1 (fromto U7-1 U8-1 (rule (width .020))))**.
5. Type **select net sig1 sig2 sig3 sig4**.
6. Type **bus diagonal**.
7. Type **unselect all nets**.

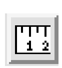
8. Examine the wiring that was routed using **bus diagonal**.

   Notice that the top net (sig1) obeys the 12 mil rule except for one fromto in the net that follows the 20 mil rule. The other nets obey the 8 mil pcb rule. The following figure shows an example of the routing.



Fromto Rule Overrides Net Rule Which Overrides PCB Rule

Use the measure icon to get information about a design



1. Click the *Measure* button on the tool bar.

   Measure appears in the mode status area.

2. Click on each of the different width wire segments.
3. Read the width information in the Measure dialog box.
4. Type **quit** in the command entry box to exit the autorouter.

Using Commands to Set Rules and Control the Autorouter

The autorouter provides numerous commands you can use to set rules and control how a design is routed. With a single command or simple combination of commands you can

- Set wire and via clearance rules for different objects
- Define wire and via grids
- Select and unselect layers for routing
- Control layer routing direction
- Control which vias are used for routing
- Control which connections are autorouted
- Control whether nets are routed with a starburst or daisy-chain topology
- Define classes of nets and assign rules
- Define groups of fromtos and assign rules
- Assign routing priorities to nets and classes

Setting wire and via clearance rules for different objects

You can set separate clearance rules for wires and vias and other objects. The following table describes how you can set the rules for wire to object and via to object.

Object to Object keyword	What it means
area_via	Clearance rule between PCB boundary or keepouts and vias
area_wire	Clearance rule between PCB boundary or keepouts and wires
pin_via	Clearance rule between through-pins and vias
pin_wire	Clearance rule between through-pins and wires
smd_via	Clearance rule between SMD pads and vias
smd_wire	Clearance rule between SMD pads and wires
via_via	Clearance rule between vias
via_wire	Clearance rule between vias and wires
wire_wire	Clearance rule between wires on different nets

In the following procedure, you set several pcb clearance rules, select certain nets, and route them. Usually, you would not select a few nets for routing. The nets are selected in these steps to reduce routing time. The purpose of this procedure is to show how you set wire clearance rules for different objects.

Set wire-to-smd, wire-to-pin, and via-to-via clearance rules

1. Start the autorouter and load *lesson4.dsn*.
2. Type **rule pcb (width .008)** in the command entry box.

3. Type **rule pcb (clearance .008 (type smd_wire))**.
4. Type **rule pcb (clearance .016 (type pin_wire))**.
5. Type **rule pcb (clearance .014 (type via_via))**.
6. Type **select net sig17 sig18 sig19 sig20 sig21 sig22 sig23 sig24 sig25 sig26 sig27**.

   If one or more nets are selected, only those connections are routed.
7. Type **bus diagonal**.

   Only the selected connections that meet the bus criteria are routed.
8. Type **route 25**.
9. Wait for *Idle* to replace the *Pause* button in the control area.
10. Click **Select > Unselect All Routing Objects**.
11. Zoom in or pan to view the wiring added by the two previous commands.
12. Click the *Measure* button on the tool bar.
13. Measure the wire_smd clearance, which should be at least .008 inches.
14. Measure the wire_pin clearance.

    Notice the 0.016 wire-to-pin clearance rule centers wires between through-pins.
15. Measure the via_via clearance.

    Vias obey the 0.014 (minimum) via-to-via clearance rule.
16. Type **quit** in the command entry box to exit the autorouter.

Selecting layers for routing

When a signal layer is *selected*, it is available for routing. If a signal layer is *unselected*, it is not used for routing unless it is an external layer on which SMD components are mounted or certain nets are assigned to route on that layer. You can assign a net to route on a particular layer (if you have the ADV option) by using the **circuit** command with a use_layer rule.

The commands that enable or disable a signal layer for routing are **select layer** and **unselect layer**. When you use these commands, include the name of the layer you want to control, for example:

```
select layer TOP
unselect layer INT1
```

Setting layer routing directions

If layer routing directions are not set in the layout system, the autorouter sets each signal layer's routing direction to horizontal or vertical. The top signal layer is set to horizontal, and the next signal layer below the top layer uses a vertical direction. The autorouter alternates between horizontal and vertical directions for each additional signal layer.

You can override the default layer routing directions by using the Layers panel or by using the **direction** command. If the interconnection between components is biased more in a horizontal or vertical direction, change the routing direction of some signal layers to accommodate the bias. Include the layer name and the *horizontal* or *vertical* keyword with the direction command to control layer routing direction. For example:

```
direction SIGNAL_2 vertical
```

Select layers for routing and set layer routing direction

1. Start the autorouter and load *lesson4.dsn*.
2. Click the *Layers* button on the tool bar.

   The Layers panel opens. The default signal layers directions display as follows:

   SIGNAL_1 
   SIGNAL_2 
   SIGNAL_3 
   SIGNAL_4 

3. Type **unselect layer SIGNAL_2** in the command entry box.

   The button in the Layers panel for SIGNAL_2 changes to .

4. Type **direction SIGNAL_1 vertical** in the command entry box.
5. Notice the button in the Layers panel for SIGNAL_1 changes to .

   Use the **direction** command to change the routing direction for the other signal layers.

6. Type **direction SIGNAL_2 horizontal** in the command entry box.

   Notice in the Layers panel that the direction command changes the layer's routing direction *and* selects the layer.

7. Type **quit** in the command entry box and exit the autorouter.

Selecting vias for routing

The vias you can use in the autorouter are the vias included by your layout system. Usually, all vias in a design are selected and available for routing, but you can unselect vias that you do not want to use.

The **select** and **unselect** commands control which vias are available for routing. Vias that are defined in the design but are unselected, are not available for routing.

When two or more vias are available for a transition between two layers, the autorouter chooses the one that uses the fewest layers and has the smallest shapes. You can use blind and buried vias if you have the HYB option.

Note: You can assign specific vias to nets, classes, and groups of fromtos if you have the tool's ADV option. See the *Controlling Via Use* in the Automatic Routing Procedures online help.

Select a single via for routing

1. Start the autorouter and load *lesson4.dsn*.
2. Click **Report > Specify**.

   The Report specify opens.
3. Click on Vias.

   The Via Report opens.
4. Browse the report and observe that all vias are selected for routing.
5. Close the report when you finish browsing.
6. Type **unselect all vias** in the command entry box.
7. Type **select via VIA35**.
8. Repeat steps 2 and 3.

9. The report indicates that all vias are unselected for routing except VIA35.
10. Close the report when you finish browsing.
11. Type **quit** in the command entry box to exit the autorouter.

Selecting Connections and Autorouting

If one or more connections are selected, only those connections are routed. Usually you do not route fewer than all connections. You can assign connections a priority if you want to control when certain nets are routed. See “Assigning routing priorities to nets and classes” later in this lesson.

Occasionally you might want to route fewer than all connections in order to check rule settings or to experiment. You can choose the connections you want to route by selecting them. The methods for selecting connections are by

- Component
- Net
- Class
- Group
- Fromto

The following table describes the commands that implement the selection methods.

Command	What it does
select component <*id*>	Selects the connections that are attached to the component identified by <*id*>, which is the component's reference designator
select net <*id*>	Selects all connections on the net identified by <*id*>
select class <*id*>	Selects all the nets that are members of the class identified by <*id*>
select group <*id*>	Selects all the fromtos (connections) that are members of the group identified by <*id*>
select fromto	Selects all unrouted connections

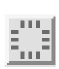
The autorouter highlights selected connections in the "select color," which is usually yellow.

After you select connections, you can apply rules and controls to the connections, then route the connections. Usually, you unselect connections after they are routed.

Use the **unselect** command to return components, nets, classes, groups, and fromtos to normal status. The **unselect all routing** command is used when you want to make certain that all previously selected routing objects are unselected, and that all unrouted connections can be routed.

Autoroute selected connections

1. Start the autorouter and load *lesson4.dsn*.



2. Click the *Select Component* button on the tool bar.

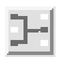
Sel Comp appears in the mode status area.

3. Click on a pin or inside the outline of any component.

   All pins on the component and all attached connections are selected.

4. Click on any pin of the same component.

   The component is unselected.

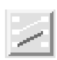


5. Click the *Select Net* button on the tool bar.
6. Click on any pin that has a guide attached.

   All connections on the net are selected.

7. Drag the pointer across several pins.

   The nets attached to the pins are selected.



8. Click the *Select Guide* button on the tool bar.
9. Drag the pointer across several guides.

   The connections attached to the guides are selected.

10. Type **unselect all routing** in the command entry box.

    All connections are unselected.

11. Type **select net sig1 sig2**.
12. Type **bus diagonal**.
13. Type **select component U1**.
14. Type **route 25**.

    Although this command specifies 25 route passes, only two are needed to route the selected connections. The remaining passes are skipped.

15. Type **unselect all routing**.

Using Routing Keepin Regions

Another method you can use to limit autorouting to fewer than all connections is to create one or more fences. A fence is a keepin boundary for the connections it encloses.

Connections that span a fence's boundaries or that lie completely outside the boundaries, are ignored during routing. You can define multiple, separate fences to restrict the routing in several areas. If you create fences that overlap, they merge to form a single, polygonal fence region.

You create fences using one of the following methods.

- Draw a fence.
- Use a dialog box to enter the fence coordinates.
- Use the **fence** command.

The easiest method for creating a fence is to draw it, but often you need to create the fence in your do file. In a do file, you use the **fence** command. Before you write the do file, use the mouse to draw the fence, determine the coordinates from the did file, and then use the coordinates in your do file.

Draw a fence and only route the enclosed connections

1. Type **delete all wires** in the command entry box.

   A message popup prompts you to confirm the command. (The autorouter should be running with *lesson4.dsn* loaded.)

2. Click *Yes* to delete all wires.

   You need to display component reference designators.

3. Click **View > Labels**.

   The View Labels dialog box opens.

4. Make sure *Ref Des* is selected.

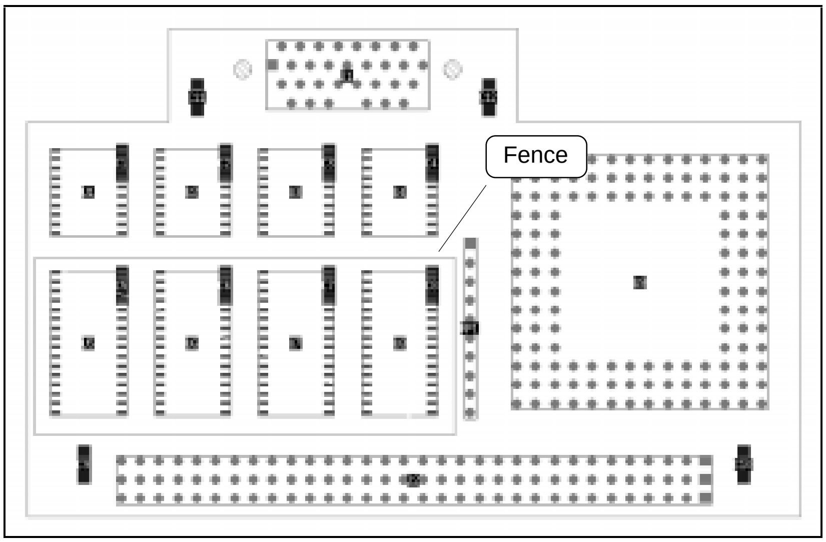
5. Click *OK*.

   The component reference designators display.

6. Click **Define > Fence > Draw Mode**.

   Draw Fence appears in the mode status area.

7. Drag the pointer and draw a fence around components U5, U6, U7, and U8, as shown in the following figure. If you do not like the fence you drew, click **Define > Fence > Draw Mode** to remove the fence and start with step 6 to draw a new fence.



8. Press the right mouse button and choose **Define Polygon As Fence** from the menu.

9. View the session transcript in the Output window, and locate the **mode edit fence**, **area init_pt**, and **area add_pt** commands.

   The X,Y coordinates beside the **area_init_pt** and **area_add_pt** commands represent diagonally opposite corners of the fence.

10. Read the coordinate information beside the **area init_pt** and **area add_pt** commands.

    You could use this coordinate information with the fence command in a do file. For example,

        fence 0.6 6.85 2.75 7.75.

11. Type **bus diagonal** in the command entry box.
12. Type **route 25**.

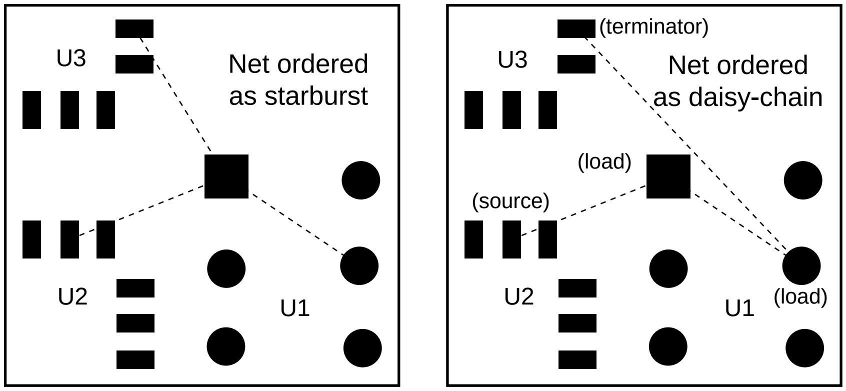
    The autorouter routes only the connections that are completely inside the fence region.

13. Wait for *Idle* to replace the *Pause* button in the control area and type **quit** in the command entry box to exit the autorouter.

Note: You can also use soft fences if you have the FST option. Soft fences are useful for maintaining separation between analog and digital circuits. See the **set** command in the autorouter's online help for additional information about soft fences.

Controlling Whether Nets Are Routed With Starburst or Daisy-Chain Wiring

Nets are usually ordered for starburst routing because it produces the most efficient wiring. Ordering determines how the pins on a net are wired. Starburst is the most efficient because the autorouter orders and routes starburst nets with minimum wire lengths. The following figure shows the basic difference between starburst and daisy-chain wiring.



Comparison Between Starburst and Daisy-Chain Wiring

Nets are ordered for daisy-chain routing in the layout system and translated to the design file, or they can be ordered in the autorouter. All nets default to starburst ordering unless they are explicitly ordered as daisy-chain.

Use the **order** command to define net ordering. For example, **order daisy net** <*net id*> orders a single net for daisy-chain routing. Replace <*net id*> in the command with the name of the net you want to order.

When you order a net for daisy-chain routing, you often want to control which pins are source, load, and terminator. If you do not assign these properties, the autorouter treats all pins on a daisy-chain net as loads and chains them together in an optimum configuration.

You use the **assign_pin** command to assign source, load, or terminator properties to pins. If a net has only one pin on a given component, you can assign a property to that pin by referencing the component and then ordering the net. For example:

```
assign_pin source U23
assign_pin terminator U29
order daisy net SO1
```

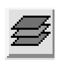
If a net has several pins on a single component, you assign pin properties by adding the pin IDs to the command. For example:

assign_pin source U33 (pins 2)
assign_pin terminator U33 (pins 11)
order daisy net PRX

Note: You assign source, load, or terminator properties to pins before you use the **order** command. If you assign pin properties but do not use the **order** command, the properties are ignored by the autorouter.

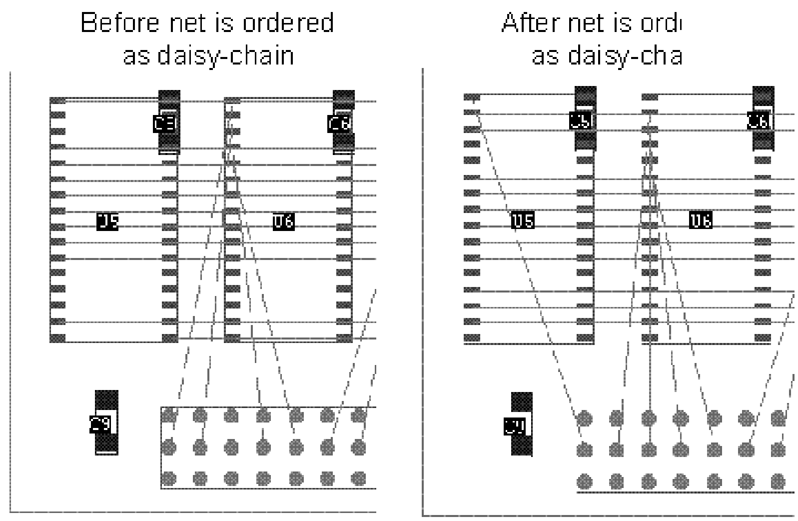
Assign pin properties and order a net for daisy-chain routing

1. Start the autorouter and load *lesson4.dsn*.



2. Click the *Layer* button on the tool bar.
3. Click on *Labels* in the Layer panel.
4. Type **select net sig1** in the command entry box.
5. Type **assign_pin source J2 (pins 32B)**.
6. Type **assign_pin source U5 (pins 1)**.
7. Type **order daisy net sig1**.

   The net is ordered for daisy-chain routing. When two or more pins on a net are assigned the same property, they are chained together. Source pins then chain to loads, and load pins chain to terminators (if any are defined).
8. Notice that the guide has moved from U6-1 to U5-1 per (step 6).



9. Type **route 25**.

   Net sig1 is routed in daisy-chain fashion.

10. Type **unselect all routing**.

Routing Nets

In this section, you will learn how to route a set of nets or fromtos with the same rules. You will

- Define a class of nets
- Define a group of fromtos

- Assign rules to classes and groups
- Assign routing priorities to nets and classes

Defining a class of nets

If you have a number of nets that you want to reference as a set, you can assign them a class name. You use the **define** command to create a class id and assign the net ids. The command is

define (class <class id> <net id> ... <net id>)

Use any combination of alpha and numeric characters for <*class id*> and substitute any number of net names you want in place of <*net id*>.

After defining a class, you can apply rules and reference all nets in the class by using the class name. For example, the command to select a class is

select class <class id>

Substitute a class name for <*class id*>.

Define a net class, select the class, and route it

1. Type **delete all wires** in the command entry box.

   A message popup prompts you to confirm.
2. Click *Yes* to delete all wires.
3. Type **define (class C1 sig17 sig18 sig19 sig20 sig21 sig22 sig23 sig24)**.

   You created a class named C1 that consists of the eight nets.
4. Type **select class C1**.
5. Type **bus diagonal**.

   Only the selected nets are routed.

6. Type **unselect all nets**.
7. Type **quit** in the command entry box to exit the autorouter.

Assigning rules to classes and groups

A common reason for defining a class of nets or a group of fromtos is so you can easily assign rules. (A fromto is a single pin-to-pin connection on a net.) If several nets or fromtos must follow the same rules or be routed in the same way, define a class or group and apply the rules to all members with a single reference.

Class and group rules are followed according to their rankings in the rule hierarchy. Net rules override class rules, group rules override net rules, and fromto rules override group rules. Refer to the Rule Hierarchy figure in the beginning of this chapter to review the rule hierarchy.

Define a class and a group, assign rules, and route

1. Start the autorouter and load *lesson4.dsn*.
2. Type **define (class C1 CLK sig34 sig35 sig47)** in the command entry box.

   You defined a class named C1 that consists of the nets CLK, sig34, sig35, and sig47.
3. Type **define (group G1 (fromto J1-8 U3-13)(fromto J1-12 U4-8)(fromto J1-10 U3-3)(fromto J1-14 U3-15))**.

Note: If you define a group and omit a fromto in error, you must use **forget group** <*group id*> before you can redefine the group.

---

4. Type **rule class C1 (width .008).**
5. Type **rule group G1 (width .020).**
6. Type **select group G1.**

7. Type **route 25.**
8. Type **select class C1.**
9. Type **route 25.**
10. Zoom in and examine how the wiring for group G1 between connector J1, U3, and U4 differs from the other wiring.

    The fromtos in group G1 follow a 20 mil width rule. The other connections follow the 8 mil width rule that is set for class C1.
11. Type **quit** in the command entry box to exit the autorouter.

Assigning routing priorities to nets and classes

You can schedule when nets and classes are routed by assigning them priorities. Assigning priorities is preferred to selecting nets to route them. You assign priorities using the **circuit** command.

Routing priorities are assigned with numeric values in the range from 1 to 255. A value of 255 assigns the highest priority, and a value of 1 assigns the lowest priority. Nets that are assigned higher priorities are routed before those assigned with lower priorities. All nets are assigned a default priority value of 10.

If you must assign several priority values to different nets or classes of nets, assign values that differ by at least ten. Priority values that differ by less than ten might be ignored due to other factors that also control routing priority. For example, the autorouter usually routes larger width wires before smaller width wires.

Assign priorities to nets and route them

1. Start the autorouter and load *lesson4.dsn*.
2. Type **define (class C1 sig1 sig3 sig5 sig7)** in the command entry box.
3. Type **circuit class C1 (priority 100).**
4. Type **circuit net sig8 (priority 255).**
5. Type **select net sig1 sig2 sig3 sig4 sig5 sig6 sig7 sig8.**
6. Type **route 25.**

   Net sig8 routes first because it is assigned the highest routing priority. Next, the nets in class C1 are routed. Then, the nets sig2, sig4, and sig6 are routed.
7. Wait for *Idle* to replace the *Pause* button in the control area and type **quit** in the command entry box to exit the autorouter.

What You Learned

In this lesson, you learned how to set rules and control the autorouter. You learned how to

- Set rules at different levels in the autorouter's hierarchy
- Set wire and via clearance rules for different objects
- Select layers for routing
- Set layer routing direction
- Select vias for routing
- Select connections for routing
- Use routing keepins
- Control whether nets are routed with starburst or daisy-chain wiring
- Define a class of nets and a group of fromtos
- Assign rules to classes and groups
- Assign routing priorities

Study the commands that set rules and perform specialized routing functions. In particular, study the **circuit**, **define**, **rule**, **fanout**, **bus**, **smart_route**, **route**, and **clean** commands. These commands are fundamental to using the routing tools.

In the next lesson, you will learn how to interactively route a PCB design.

Lesson 5: Interactive Routing and Editing

What You Will Learn

This lesson teaches you how to use EditRoute, which is the tool set for interactively routing and editing PCB wiring. In this lesson, you will learn how to

- Set up your interactive routing environment
- Use the routing tools
- Control the routing layer
- Add vias as you route
- Move wiring
- Copy wiring
- Eliminate extra angles from a wire
- Replace a via with another type
- Change the width of a wire segment

This lesson takes about 60 minutes to complete.

What to do Before You Begin

Before you begin this lesson, read the introductory chapter, which explains conventions, terminology, and symbols that are used throughout this tutorial.

Interactively Routing

EditRoute is the interactive routing and editing tool set for the autorouter. These interactive tools use the same ShapeBased technology and gridless environment as the AutoRoute tools. You can use the interactive tools to preroute critical connections before you autoroute and to change or reroute connections after autorouting. See the online help for more information about setting your interactive routing environment.

Setting your interactive routing environment

You set the interactive routing and editing environment by choosing **Setup** from the Interactive Routing menu. The Interactive Routing menu displays when you press the right mouse button in the work area.

Note: If another Windows application uses the right mouse button, you might need to hold down the `[Ctrl]` key when you press the right mouse button to display the Interactive Routing menu.

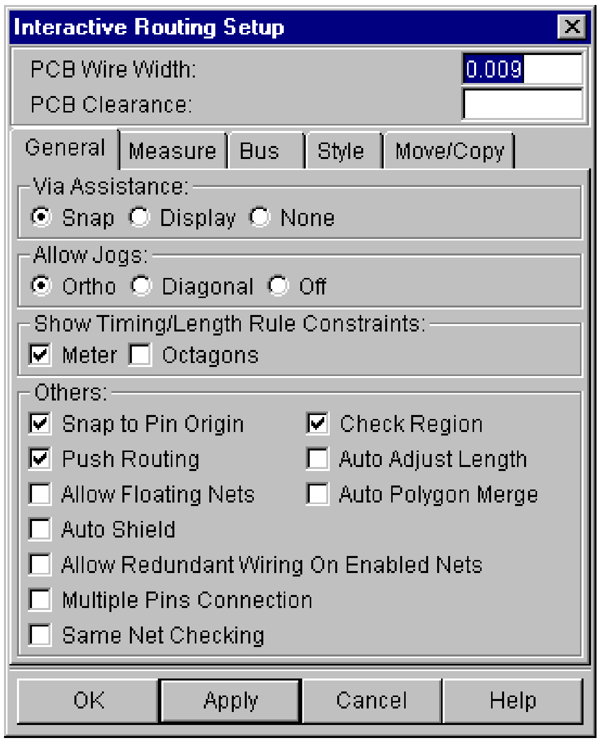
The Interactive Routing Setup dialog box contains five tabs, which are General, Measure, Bus, Style, and Move/Copy. You can use these tabs to set Measure mode options, set move and copy options, and to set interactive routing controls in three categories: General, Bus, and Style. You can also use the Interactive Routing Setup dialog box to set global (PCB) clearance and wire width rules.

The Interactive Routing Setup dialog box opens with the General tab displayed. To change to an other category, click on the corresponding tab. General displays controls you can use to:

- Set via assistance, wiring jog controls, and timing/length rule indicators.

- Enable or disable wire snap to pin orgin, push routing, region rule checking, and redundant wiring on enabled nets.
- Allow or disallow floating nets, automatic polygon merging, automatic shielding, and automatic length adjustments.

The following figure shows the Interactive Routing Setup dialog box with the default category General.

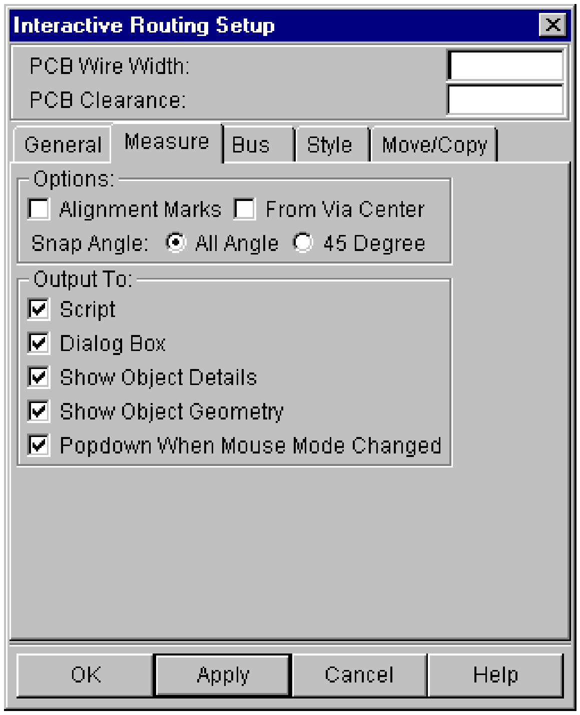


Measure displays controls you can use to:

- Control whether horizontal and vertical tick marks display.

- Control whether the tool measures blind and buried via gap clearances from via centers or from via edges.
- Control how the pointer snaps to the location you are measuring.
- Control whether measurement information is displayed in the Output window or in the Measure dialog box, or both when the Script or Dialog Box is checked.
- Control whether additional information is displayed in the Measure dialog box for objects you measure.
- Control whether the information displayed in the Measure dialog box for an object includes its shape and its location in the design.
- Control whether the Measure dialog box automatically opens when you click or drag the pointer in the work area.

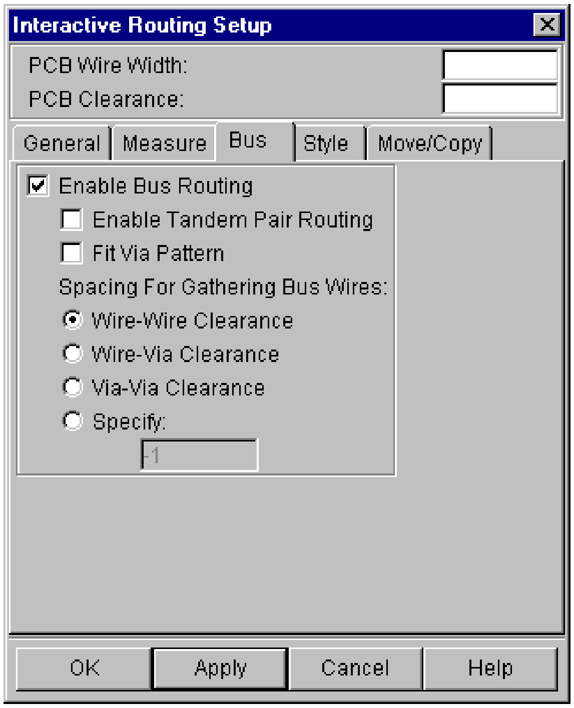
The following figure shows the Interactive Routing Setup dialog box for Measure.



Bus displays controls you can use to:

- Enable or disable bus routing
- Control tandem layer pair routing and via pattern fitting
- Set the bus wire clearance

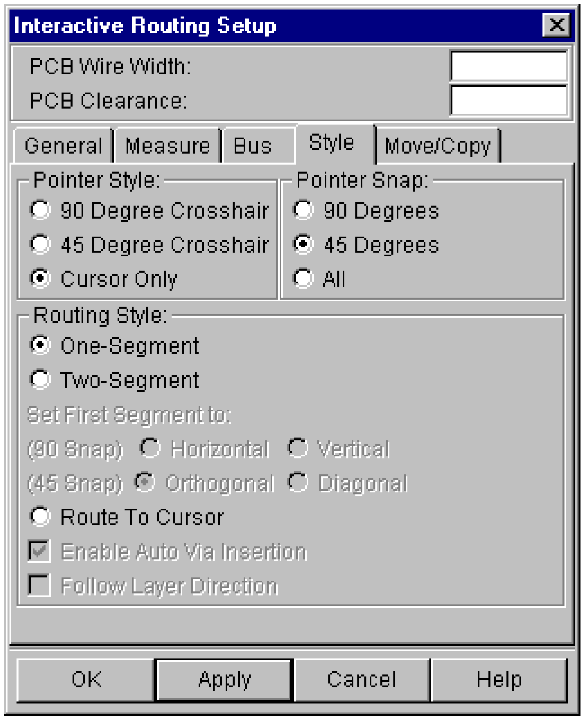
The following figure shows the Interactive Routing Setup dialog box for Bus.



Style displays controls you can use to:

- Set the pointer style to 90 degree crosshairs, 45 and 90 degree crosshairs, or arrow only pointer style.
- Control whether the pointer snaps to any angle, or snaps to a 45 or 90 degree when routing.
- Control the one or two segment routing style.

The following figure shows the Interactive Routing Setup dialog box for Style.

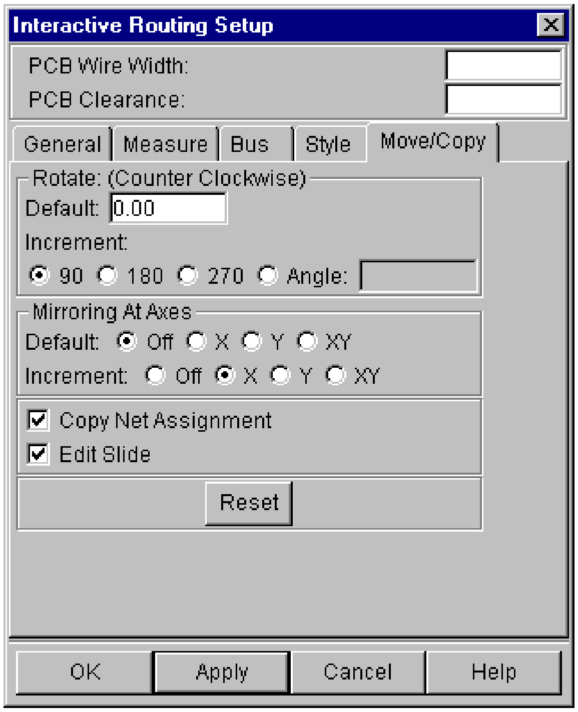


Move/Copy displays controls you can use to:

- Set the default rotation angle and mirroring axis for automatic object rotation and mirroring during move or copy operations.
- Change the incremental rotation angle for incremenetal object rotation.
- Change the incremental mirroring axis for incremental object mirroring.

- Control whether copies of wiring polygons have the same net assignment as the original or are unassigned (for Copy Polygon mode only).
- Control whether the tool slides the objects, attempting to push obstacles out of the way, or moves them over obstacles and drops them in the new location (for Move mode only).

The following figure shows the Interactive Routing Setup dialog box for Move/Copy.



In this section, you will load a wires file, which contains routed wires. You will then use the Interactive Routing Setup dialog box to change the pointer style and change the pointer snap angle, which will affect the interactive routing environment. Next you will route a connection to see how Edit Route pushes wires aside as you are routing.

Load a wires file

1. Start the autorouter and load *lesson5a.dsn*.
2. Click **File > Read > Wires**.

   The Read Wires dialog box opens.
3. Click the *Browse* button.

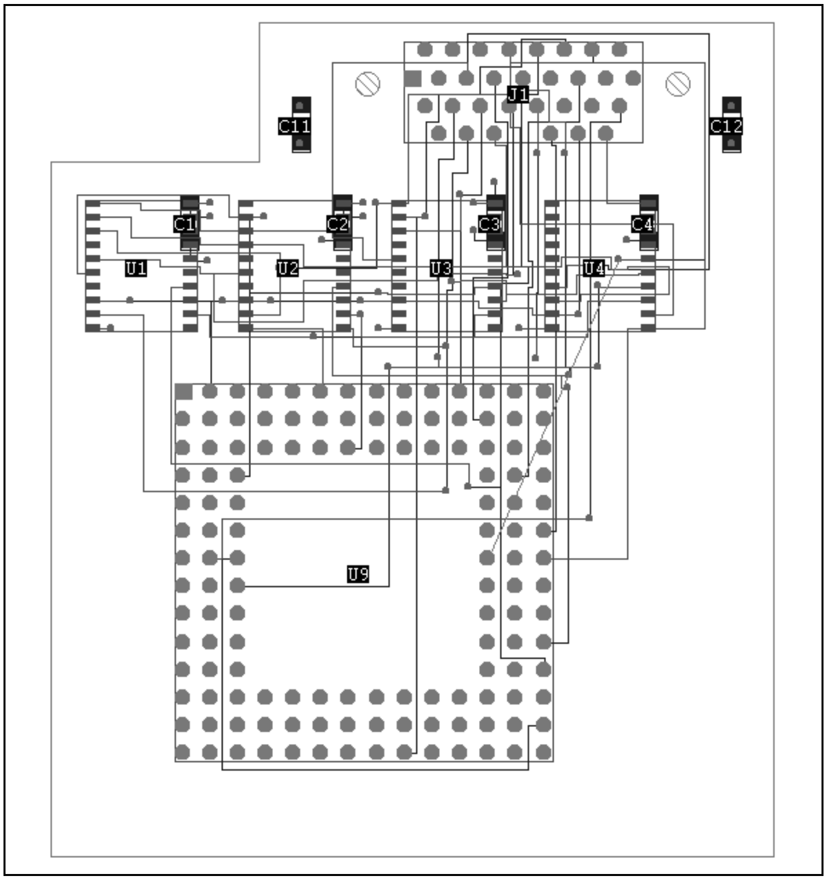
   The Open dialog box opens for Windows platforms.

   The Select File dialog box opens for UNIX platforms.
4. Change to the autorouter's tutorial directory for Windows platforms and open *lesson 5a.w*.

   Select *lesson 5a.w* and click *OK* for UNIX platforms.

   The filename is added to the dialog box.
5. Click *OK* on the Read Wires dialog box.

   The wires file loads. All connections except one are prerouted as shown in the following figure.



lesson5a.dsn With All Connections Except One Prerouted

Set up your interactive routing and editing environment

1. Move the pointer to the work area and press the right mouse button.

   The Interactive Routing Menu opens.

2. Choose *Setup* from the Interactive Routing Menu.

   The Interactive Routing Setup dialog box opens.

3. Click on *Style.*

You will set the pointer style to a 90 degree crosshair and set the pointer snap angle to 90 degrees.

4. Select *90 Degree Crosshair* for the *Pointer Style*.
5. Select *90 Degrees* for the *Pointer Snap*.
6. Click *OK*.

   Next you will display reference designators and pin IDs.

7. Click **View > Labels**.

   The View Labels dialog box opens.

8. Select *Ref Des* and *Pin IDs*.
9. Click *OK*.

   The Reference designators display with the components.

Route a connection using Edit Route

1. Zoom in to view the unrouted connection of U4-16 to U5.

   You will route the unrouted connection of U4-16 to U5 using Edit Route mode.

2. Press the right mouse button and choose *Edit Route Mode*.

   A pointer appears in the work area and Edit Route appears in the mode status area.

3. Click pin 16 on U4.
4. The pointer changes to 90 degree crosshairs.
5. Press the right mouse button and choose *Cancel*.

   The routing operation stops. Edit Route still appears in the mode status area.

   The crosshairs do not display unless you are routing a connection.

6. Click on U4-16 and route the connection out of the right side of the pin and down half the distance to the target pin on U5. Zoom in if necessary.

   The route adds unnecessary angles to the surrounding wires. You will learn how to remove unnecessary angles from wires by using Critic Route later in this lesson. You will ignore these angles for now.

Note: If you make a mistake, you can press the right mouse button and choose *Undo* to undo the last action or *Cancel* to cancel the last command.

---

7. Press the right mouse button and choose *Setup***.**
8. Click *45 Degree Crosshair* for the *Pointer Style*.
9. Click *45 Degrees* for the *Pointer Snap*.
10. Click *OK*.

    The pointer appears as both 45 and 90 degree crosshairs.

11. Use a 45 degree corner to finish routing the connection to U5.

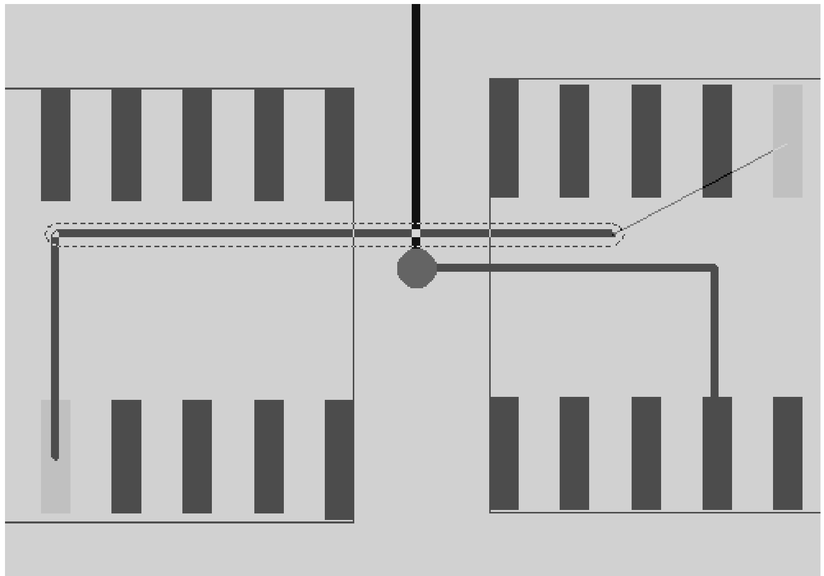
    When you click the target pin on U5, the connection completes, and the wire disconnects from the pointer.

12. Type **quit** in the command entry area to exit the autorouter.

    You do not need to save the session file or the did file.

Routing and replacing wires

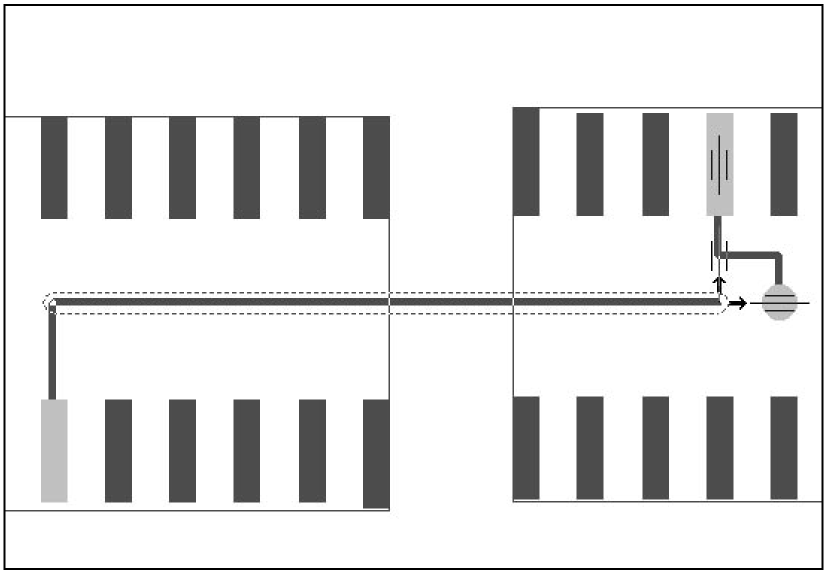
You use the Edit Route mode to route connections and to replace existing wires and vias. While you are routing or replacing a wire, an envelope surrounds the wire. This clearance envelope corresponds to the clearance rule for the wire you are routing. The envelope helps you see the clearance from the wire to other objects. The following figure shows the clearance envelope that surrounds a wire during an interactive routing session.



A Clearance Envelope Surrounds the Wire You Are Routing

The clearance envelope changes if, for example, you add a via and change to a layer with a different clearance rule.

As you route a connection, alignment marks and arrows display to help you align the end of the wire with a target wire, pin, or via. When the end of the wire aligns with a target, one or more arrows appear at the wire end, and alignment marks appear at the target object. The following figure shows how the alignment marks and arrows appear during a routing session.



Alignment Marks and Arrows Help You Align the Wire with a Target Pin or Via

You will use Edit Route mode to route several connections. As you route each connection, use the clearance envelope to help you route the wires as close together as possible without creating clearance violations, and use the alignment marks and arrows to make straight connections to the target pins.

Interactively route several connections

1. Start the autorouter and load *lesson5b.dsn*.

   Display the reference designators and pin IDs.

2. Click **View > Labels**.

   The View Labels dialog box displays.

3. Select *Ref Des and Pin IDs*.
4. Move the mouse to the work area and press the right mouse button.

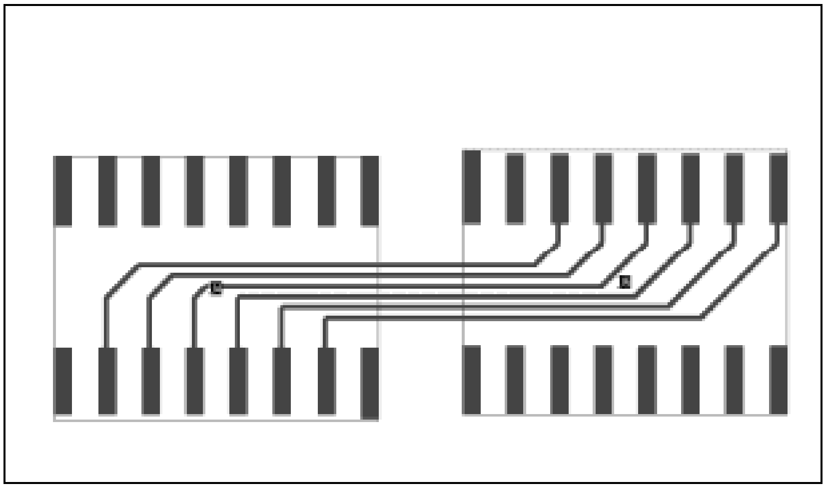
   The Interactive Route dialog box opens.

5. Choose **Edit Route Mode** from the Interactive Routing menu.

   Edit Route appears in the mode status area.

6. Zoom in to view U4 and U5.

7. Route the six connections between U4 and U5, as shown in the following figure.



Route the Six Connections Between U4 and U5

8. Type **quit** in the command entry area to exit the autorouter.

   You immediately exit the autorouter without saving a session file. The did file remains in the tutorial directory. You will not need the did file, so you can delete it.

Next you will learn how to finish an unrouted or partially routed connection.

Finishing the routing

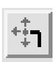
While you are routing in Edit Route mode, you can finish an unrouted or partially routed connection by choosing **Finish Route** from the right button menu. Finish Route adds wire segments and vias, if required, to complete a connection.

Finish Route searches for a path within a limited area around the connection you are routing. If a path cannot be found in the search area, the connection cannot be finished automatically and the operation is abandoned.

In the following procedure, you finish several connections using the Finish Route option in Edit Route mode.

Finish routing connections

1. Start the autorouter and load *lesson5b.dsn*.
2. Display the reference designators and pin IDs.



3. Click the *Edit Route* button on the tool bar.

   Edit Route appears in the mode status area.
4. Zoom in to U4, U5, and the lower half of U6.
5. Click on one of the pins in the upper row of U4.
6. Digitize a portion of the connection above U4.
7. Press the right mouse button and choose **Finish Route**.

   The connection is completed for you.
8. Click on another unrouted pin in the upper row of U4.
9. Press the right mouse button and choose **Finish Route**.

   The connection routes automatically.

   You will route the remaining unrouted pins on the upper row of U4. You could do this by selecting each pin and repeating the previous step, but you will learn a faster way to route these connections.

10. Click on another unrouted pin in the upper row of U4.
11. Press `[Ctrl-A]`.

    Using these two keys, you can repeat the last menu command. In this step, you repeated step 9.
12. Repeat steps 10 and 11 to route the remaining unrouted pins on the upper row of U4.
13. Exit the autorouter by using **File > Quit** or by typing **quit** in the command entry area.

    You do not need to save a session file or the did file.

Reversing and restoring interactive operations

If you make a mistake, the autorouter lets you undo a previous interactive routing function. If you did not intend to undo something, you can restore it using the **redo** command.

Function	What it Does	How to Execute it
undo	Reverses the last interactive function	Press the `F3` function key, choose **Undo** from the right button menu, or choose **Edit > Undo** from the menu bar
redo	Restores the last function you reversed with undo	Hold down the `[Shift]` key and press `F3` or choose **Edit > Redo** from the menu bar

Note: You can undo one or a series of interactive functions, but if you interrupt an interactive routing, editing, or placement session with a select, report, or autorouting or autoplacement function, undo and redo memory is cleared and you cannot undo or redo prior functions.

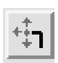
The following procedure uses undo and redo to reverse and then restore interactive routing functions.

Use undo and redo to reverse and restore interactive routing

1. Start the autorouter and load *lesson5c.dsn*.
2. Press the right button menu and choose **Setup** from the Interactive Routing menu.

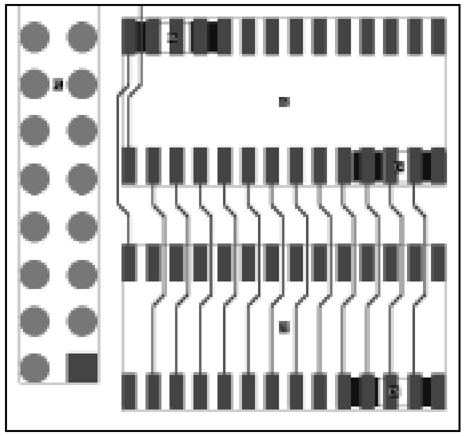
   The Interactive Routing Setup dialog box opens.
3. Click on *General* next to *Setup*.

   A drop down menu opens.
4. Select *Style*.
5. Click *90 Degree Crosshair* for the *Pointer Style*.
6. Click *OK.*



7. Click the *Edit Route* button on the tool bar.

   Edit Route appears in the mode status area.
8. Display the reference designators.
9. Route several connections between U2 and U3 as shown in the following figure.



Route Several Connections Between U2 and U3

10. Press F3 to undo the last function.
11. Press `[Shift]F3` to redo the function you reversed with undo.
12. Complete all the connections between U2 and U3
13. Use F3 to undo all previous routing steps.
14. Use `[Shift]F3` to restore all the routing you removed with undo.
15. Exit the autorouter.

    You do not need to save a session file or the did file.

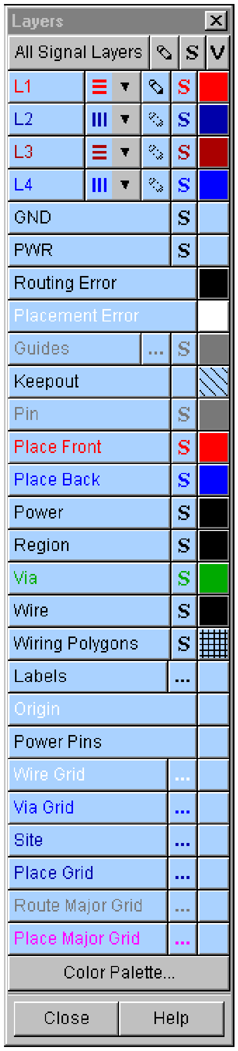
Adding vias and changing layers

While you are interactively routing, the layer you are routing on is the *primary layer*. The layers that you can change to when you add a via are the *secondary layers*. The primary layer is always named on the status bar in the area beside the pencil button.

If you add a via and change to a different routing layer, the layer you change to becomes the primary layer, and the previous routing layer becomes a secondary layer. You control the initial primary and secondary routing layers by using the Layers panel. In the Layer panel, you can disable layers if you do not want to use them during interactive routing.

A pencil button in the Layer panel identifies whether a signal layer is the primary layer or a secondary layer. The bold pencil identifies the primary layer. A dim pencil identifies a secondary layer. If neither the dim nor bold pencil appears for a signal layer, the layer is disabled for interactive routing.

The following figure shows the Layer panel with L1 as the primary layer and L2, L3, and L4 as secondary layers.



Layers panel Shows L1 as Primary Layer and L2, L3, L4 as Secondary Layers

You set a layer as primary, secondary, or disabled by clicking on the pencil. If you click on a dimmed pencil, the layer becomes primary. If you click a second time, the pencil box is emptied and the layer is disabled for interactive routing.

After you choose which layer is primary, which layers are secondary, and whether layers are disabled, you use the mouse or function keys to add vias and change the routing layer. The layer you change to when you add a via depends on how you set up layers as primary, secondary, or disabled.

When you digitize twice at the end of a wire segment, the autorouter adds a via, determines an optimum secondary layer, and changes to that layer. Instead of allowing the autorouter to choose the layer, you can choose the layer change by using the F5 and F6 function keys.

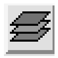
The F5 key adds a via and changes routing to the next available layer *below* the current routing layer. Each time you press F5, the next active signal layer becomes the current routing layer. If you press F5 and no layer is available below the current routing layer, the system beeps and the message “*No enabled layer below <layer id>*” displays in the message area.

The F6 key adds a via and changes routing to the next available layer *above* the current routing layer. Each time you press F6, you move up to the next available signal layer. If no layer is available above the current routing layer, the system beeps and a warning message appears in the message area.

In the following procedure, you set up the primary and secondary routing layers, disable two signal layers for interactive routing, and use the F5 and F6 function keys to add a via.

Set up the routing layers, and use function keys to change layers and add a via

1. Start the autorouter and load *lesson5d.dsn*.

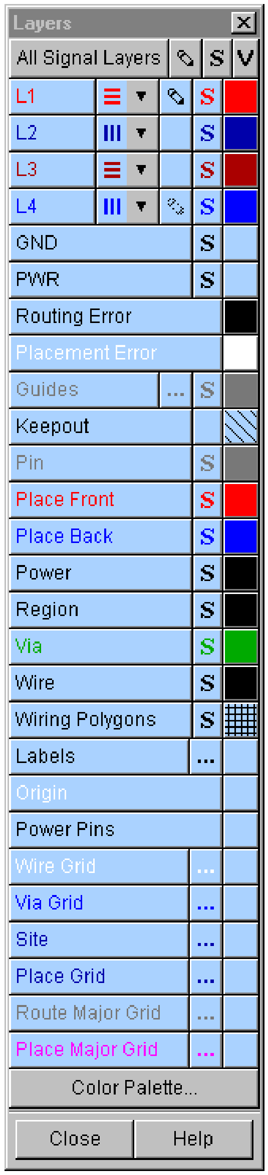


2. Click the *Layer* button on the tool bar.

   The Layer panel opens, as shown in the following figure. You will disable layers L2 and L3.

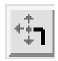
3. Click twice in each pencil box to disable layers L2 and L3.

The pencil boxes for layers L2 and L3 are empty, which means these layers are disabled.



4. Click *Close*.

5. Display the reference designators.



6. Click the *Edit Route* button on the tool bar.

   Edit Route appears in the mode status area, and L1 appears on the status bar as the primary layer.

7. Click on pin 1, which is the square pin of P1.

   You will need to zoom in to see pin 1.

8. Move the crosshairs a small distance to the right and away from the pin.

9. Press F5.

   The wire changes to the next available layer below L1, which is L4. (Remember that you disabled L2 and L3.) L4 appears in the status area as the primary layer.

10. Press F5 again.

    The message area indicates that no additional layers are available below the current layer.

11. Press F6.

    The wire changes to the available layer above L4, which is L1.

12. Press F5 to set the primary layer to L4.

13. Digitize a wire segment that ends just above the target pin on U3.

14. Press F6.

    A via is added. The primary layer changes to L1.

15. Finish routing the connection to U3.

16. Exit the autorouter.

    You do not need to save a session file or the did file.

Editing Wires and Vias

You edit wires and vias by setting one of the following editing modes for the left mouse button.

- Move
- Copy Route
- Critic Route
- Change Via
- Change Wire Width

You set an editing mode by choosing the mode from by pressing the right mouse button or by clicking a button on the tool bar.

When you edit wires and vias, the autorouter observes clearance rules if *Checking* is enabled on the status bar. If Checking is enabled, and you attempt an edit that violates a clearance rule, the function is ignored.

Moving wires and vias

You use Move Route mode to move wire segments and vias. As you move a wire segment, the segment stretches at its corners and at the points where it attaches to pins and vias.

If you move a wire segment against a component pin, and Allow Jogs is enabled under the General tab in the Interactive Routing Setup dialog box, the segment jogs around the pin if possible. When you move a wire against another, or against a via (and Push Routing is enabled under the General tab in the Interactive Routing Setup dialog box), the stationary wire or via is pushed to avoid a clearance violation.

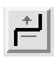
If you move a via, the attached wires stretch to follow. You cannot move a via to a location that creates a clearance violation if *Checking* is selected on the status bar.

If you move a via against a wire and Push Routing is enabled under the General tab in the Interactive Routing Setup dialog box), the wire deforms around the via to avoid a clearance violation. If you move a via against a second via, the second via is pushed to avoid a violation.

Next you will use Move mode to move wires and vias. You will see how Move mode works with Push Routing enabled and disabled.

Move wires and vias with Push Routing enabled and disabled

1. Start the autorouter and load *lesson5e.dsn*.
2. Display the reference designators.
3. Zoom in on DS1, U4, U5, and on the via above U4.



4. Click the *Move* button on the tool bar.

   Move appears in the mode status area.
5. Click on the horizontal segment of wire that connects U4 and U5.

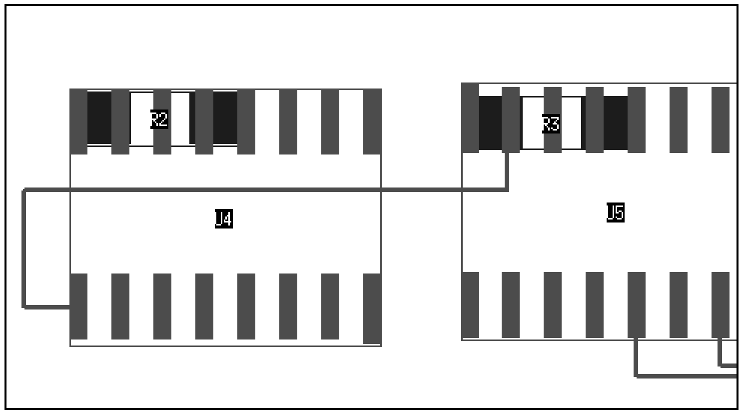
   The pointer changes to an arrow (Windows) or a four-sided arrow (UNIX) to indicate you can move the wire.
6. Move the pointer up until the wire is stopped by the upper row of pins.

   The wire cannot move beyond the upper row of pins because the vertical segment creates a short circuit with the top left pin of U4.
7. Move the pointer down until the wire is stopped by the lower row of pins.
8. Press the right mouse button and choose *Cancel*.
9. Click on the vertical segment of wire in U4.

   Next you will move the vertical segment of the wire in U4 so that the horizontal of the wire can jog around the pins.
10. Move the pointer to the left and outside the outline of U4.

11. Click to drop the segment between DS1 and U4.

    The following figure shows the wire path after you move the vertical segment to the left.



New Wire Path From U4 to U5

12. Click on the horizontal segment of the wire that connects U4 and U5.
13. Move the pointer up to the via above the upper row of pins in U4.

    The segment now jogs over the upper row of U4 pins.

14. Move the pointer back down until the wire is stopped by the lower row pins in U4.

    Notice how the segment jogs around each pin in the row.

15. Press the right mouse button and choose *Cancel*.
16. Press the right mouse button and choose *Setup* from the Interactive Routing menu.

    The Interactive Routing Setup dialog box appears.

17. Unselect *Push Routing* under the General category to disable *Push Routing*.
18. Click *Apply*.

19. Drag the Interactive Routing Setup dialog box to the side of the screen and out of the way.
20. Click on the horizontal segment of the wire that connects U4 and U5.
21. Move the pointer up.

    The wire is stopped by the upper row of pins because *Push Routing* is disabled.
22. Press the right mouse button and notice that *Allow Jog Diagonal* is disabled.
22. Choose *Cancel*.
23. Select the *Push Routing* option in the Interactive Routing Setup dialog box under the General cateogy and click *OK* to enable *Push Routing*.
24. Click on the horizontal segment of the wire that connects U4 and U5 and move the pointer up to the via above U4.

    The wire segment attached to the pointer pushes the via.
25. Press the right mouse button and choose *Cancel*.
26. Exit the autorouter.

    You do not need to save a session file or the did file.

Copying wires

You copy wires by using Copy Route mode, which you set by choosing **Copy Route Mode** from the right mouse button menu or by clicking the Copy Route button on the tool bar.

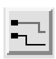
You set an editing mode by choosing the mode from by pressing the right mouse button or by clicking a button on the tool bar.

With Copy Route mode set, click on the wire or on a pin attached to the wire you want to copy. Then click on the target pin you want to route with the copied wire. Target connections must be unrouted, and you must be able to wire target connections with the same wire path as the connection you copy. If the paths of the copied connection and the target connection are different, the copy function fails and the message "*Path cannot be copied*" appears in the Message area.

You will copy a wire path and click on an adjacent pin to replicate the path. You can also drag the pointer across several pins, to replicate the wire path to multiple connections in a single operation.

Copy a wire path to several unrouted connections

1. Start the autorouter and load *lesson5f.dsn*.
2. Display the reference designators.
3. Zoom in on U4 and U5.

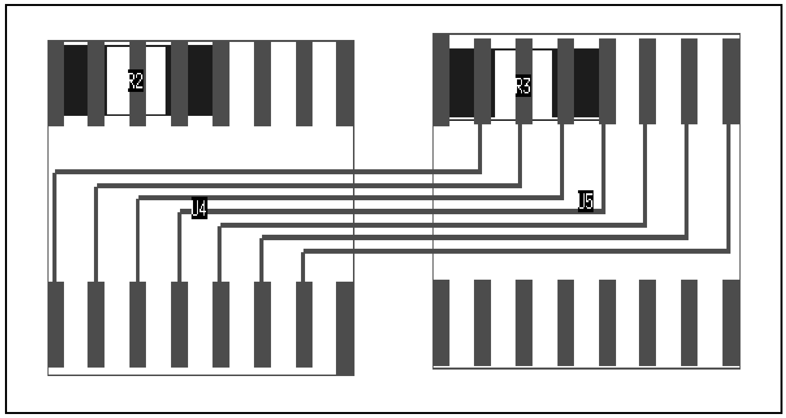


4. Click the *Copy Route* button on the tool bar.
5. Click on the wire that connects U4 and U5.

   The wire is copied to memory and is highlighted.
6. Click on the bottom pin in U4 that is next to the pin that is highlighted.

   The wire you copied in the previous step is replicated.
7. Drag the pointer across the five adjacent pins to the right.

   The original wire is copied to the five connections. The following figure shows the copy results.



Results After Wire on the Left is Copied to Five Adjacent Pins

8. Exit the autorouter.

   You do not need to save a session file or the did file.

Removing unnecessary angles from wires

You remove unnecessary angles from wires by using Critic Route mode, which you set by choosing **Critic Route Mode** from the right button menu or by clicking the Critic Route button on the tool bar. Critic Route mode removes unnecessary angles by moving existing wire segments.

Critic Route mode cannot remove unnecessary angles if one or more new wire segments are needed to perform a function or if the function creates a violation. You remove unnecessary angles from a single wire by clicking on the wire, or you can remove unnecessary angles from multiple wires by dragging the pointer over them.

In the following procedure, you remove the extra angles from a single wire by clicking on the wire. You remove the extra angles from multiple wires by dragging the pointer across them.

Remove unnecessary angles from wires

1. Start the autorouter and load *lesson5g.dsn*.
2. Click **File > Read > Wires**.

   The Read Wires dialog box opens.
3. Click the *Browse* button.
4. Choose *lesson5g.w* from the tutorial directory.
5. Click *OK*.

   The file *lesson5g.w* appears in the Read Wires data entry box.
6. Click *OK*.
7. Display the reference designators.
8. Zoom in on U4 and U5.

9. Click the *Critic Route* button on the tool bar.
10. Click on the wire attached to the lower left pin of U4.

The extra angles are removed from the wire.

11. Drag the pointer across all wires that connect between U4 and U5.

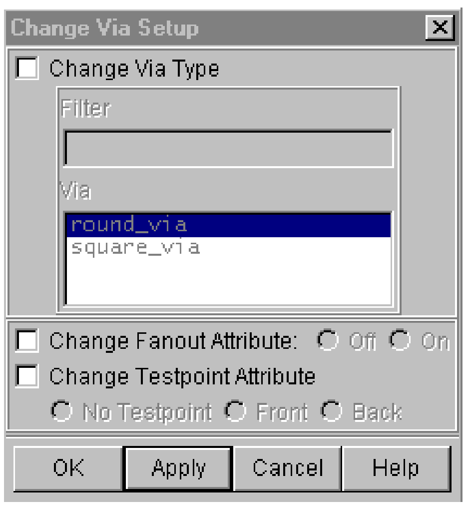
    The extra angles are removed from all the wires in a single operation.

12. Exit the autorouter.

    You do not need to save a session file or the did file.

Replacing vias with another type

You replace an existing via with a different type by using Change Via Mode, which you set from the right button menu. When you set Change Via mode, you must choose which via you want to use to replace existing vias. The Change Via Setup dialog box, which is shown in the following figure, determines the replacement via.



Change Via Setup Dialog Box Determines the Replacement Via

After initially choosing a replacement via, you can choose a different replacement via by pressing the right mouse button and choosing **Setup Change Via**.

You will set Change Via mode and choose a replacement via, replace several existing vias, and then choose a different replacement via.

Set Change Via mode, choose a replacement via, and replace existing vias

1. Start the autorouter and load *lesson5h.dsn*.
2. Click **File > Read > Wires**.
3. Click the *Browse* button.
4. Choose *lesson5h.w* from the tutorial directory.
5. Click *OK*.

   The file *lesson5h.w* appears in the Read Wires data entry box.
6. Click *OK.*
7. Display the reference designators.
8. Zoom in on the wiring in the area between U4 and U6.
9. Press the right mouse button and choose **Change > Change Via Mode**.

   The Change Via Setup dialog box opens.
10. Choose *square_via* from the Via list.
11. Click *OK*.

    Change Via appears in the mode status area.
12. Click on a via that is attached to one of the connections between U4 and U6.

    A square via replaces the round via.

13. Press the right mouse button and choose **Setup Change Via**.
14. Choose *round_via* from the Vias list.
15. Click *OK*.
16. Click the square via.

    The square via changes to a round via.
17. Exit the autorouter.

    You do not need to save a session file or the did file.

Changing the width of an existing wire segment

When you route a connection in the autorouter, whether you are autorouting or interactively routing, wire width is determined by a rule setting. You can change the width of an existing wire segment without setting a rule by using the Change Wire mode, which you choose from the right button menu. The Change Wire Setup dialog box is shown in the following figure.

Change Wire Setup Dialog Box Controls Width for Change Wire Mode

After initially setting the Edit Wire Width value, you can set a different value by selecting Change Wire Mode and choosing **Setup Wire Parameters** from the right button menu.

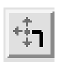
The Change Wire mode changes the width of a wire segment if the action does not create a violation. If changing the width of a wire segment results in a violation, the action is ignored.

Because Change Wire mode does not set or change a wire width rule, changed widths are not retained if you reroute the segments. If you want to retain the new widths during an autorouting or rerouting operation, use the **protect** command to prevent the autorouter from altering the wires.

You will set the Change Wire mode and set the edit wire width. You will then protect the edited wires to prevent them from changing during subsequent autorouting or rerouting operations.

Set the Change Wire mode, change the width of several wire segments, and protect the wires

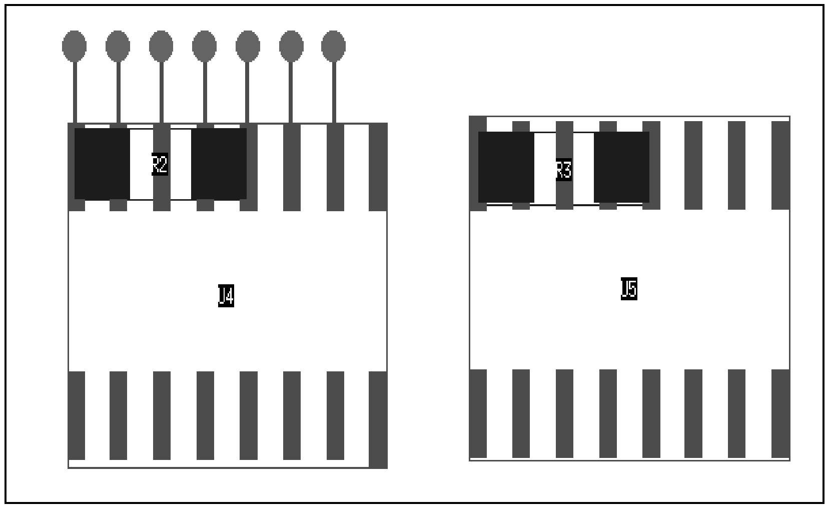
1. Start the autorouter and load *lesson5i.dsn*.
2. Display the reference designators.
3. Zoom in on U4 and U5.



4. Click the *Edit Route* button on the tool bar.
5. Route a short escape wire from the top left pin of U4.
6. Click twice at the end of the escape wire to add a via.
7. Press the right mouse button and choose *Done*.

   A short escape wire and via is routed from the top left pin of U4.
8. Repeat steps 6 and 7 to route the remaining six pins in the top row of U4.

   The wiring is shown in the following figure.



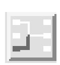
You will change the wire width of these segments to 10 mils.

9. Press the right mouse button and choose **Change > Change Wire Width Mode**.

   The Change Wire Setup dialog box opens.

10. Type **10** in the *Wire Width* data entry box.
11. Click *OK*.
12. Click on each escape wire attached to U4.

    The wire width changes to 10 mils.



13. Click the *Select Wire* button on the tool bar.
14. Drag the pointer across all seven escape wires.

    The wires change from red to yellow, which is the select color.

15. Type **protect selected** in the command entry area.

    A message popup prompts "*Protect All Selected Wires*?"

16. Click *Yes*.

17. Press the right mouse button and choose **UnSelect All Objects**.

    The escape wires have a thin white line through their centers and the pins and vias have a small white circle to indicate that they are protected.

19. Exit the autorouter.

    You do not need to save a session file or the did file.

What You Learned

In this lesson, you learned how to

- Set up your interactive routing environment
- Use the routing tools
- Control the routing layer
- Add vias as you route
- Move wiring
- Copy wiring
- Eliminate extra angles from a wire
- Replace a via with another type
- Change the width of a wire segment

Congratulations!

You have completed the autorouting tutorial.

If you worked through each lesson in this tutorial, you learned how

- The autorouter works
- To use the autorouter's GUI
- To place components interactively and automatically
- Set up the autorouter and analyze early results
- Set rules and control the autorouter
- To route wires automatically
- How to use the interactive routing tools.

With the information and procedures you learned, you are ready to begin your own projects.

When you are using the autorouter, remember there is a wealth of information available to you online. Browse the autorouter's Help menu and explore the topics. Online help is available on a broad range of topics from general information to step-by-step procedures for accomplishing your design tasks.

Index

S

T

U

V

W

Z