



BACHELOR THESIS

## **PCB Design with SPECCTRA Autorouter**

UNIVERSITY OF APPLIED SCIENCES TECHNIKUM WIEN  
BACHELOR PROGRAMME ELECTRONIC

Thomas Kladensky  
Sternhaussiedlung 5A  
8793 Trofaiach  
Pers. Kennz. : 0410254092

**Prof. (FH) Dipl.-Ing. Dr. Martin Horauer**

Wien, 24.01.2006

---

## EIDESSTATTLICHE ERKLÄRUNG

Hiermit versichere ich, die vorliegende Arbeit selbstständig und unter ausschließlicher Verwendung der angegebenen Literatur und Hilfsmittel erstellt zu haben.

Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegt und auch nicht veröffentlicht.

Ich erteile hiermit der Fachhochschule Technikum Wien die wissenschaftlichen Nutzungsrechte an der vorliegenden Arbeit.

Wien, 24.01.2006

## Table of contents:

<b>1. INTRODUCTION</b>	<b>3</b>
<i>1.1. How does SPECCTRA work</i>	<i>3</i>
<i>1.2. Proceeding of SPECCTRA Autorouter</i>	<i>4</i>
<b>2. BASIC HANDLING OF THE AUTOROUTER</b>	<b>4</b>
<i>2.1 General Description</i>	<i>4</i>
<b>3. ESSENTIAL COMMANDS</b>	<b>7</b>
<i>3.1 Basic Configures</i>	<i>9</i>
<b>4. SPECIAL COMMANDS FOR DO FILES</b>	<b>12</b>
<i>4.1 Division power electronics</i>	<i>12</i>
<i>4.2 Division high frequency technique</i>	<i>15</i>
<i>4.3 Division bus systems</i>	<i>17</i>
<b>5. LITERATURE</b>	<b>19</b>

## Abstract

This thesis gives a short introduction how to design a professional printed circuit board (PCB) using the PCB Software SPECCTRA. It is focused to design a board with the main routing rules. The first chapter presents the basics of SPECCTRA. The next few chapters give information about crosstalk, calculating the right wire width, the most important information about bus systems and high frequency techniques. After working through this thesis you should be able to design your own PCB with SPECCTRA.

# 1. Introduction

This is a short tutorial for using the SPECCTRA Autorouter. In this tutorial you will learn the basics to use and handle SPECCTRA. You will learn about:

- How does SPECCTRA work
- Essential Commands
- Crosstalk
- Calculate and set the right wire width
- High Frequency techniques
- Specifications of Bus Systems

## 1.1. How does SPECCTRA work

SPECCTRA works on using the shape based technology. SPECCTRA differs from traditional grid-mapped systems in modelling 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.

Fig.: 1 shows the basic difference between the Shape Based system and a grid-mapped system.

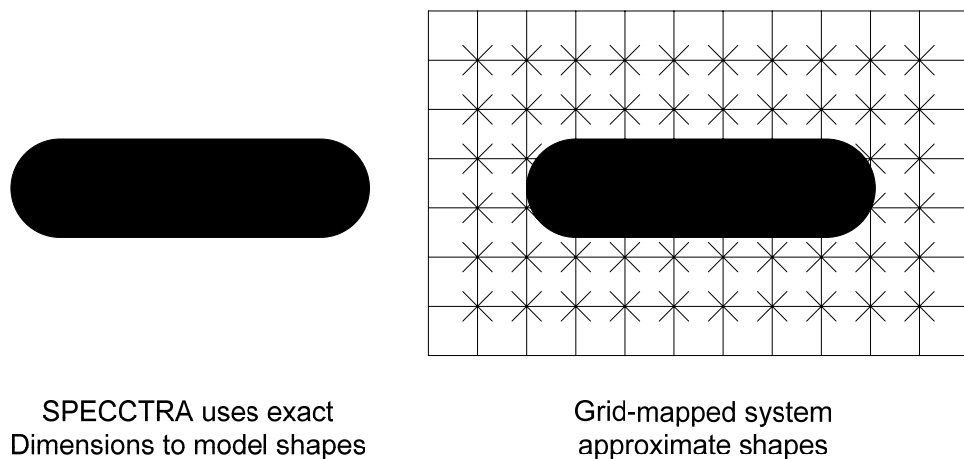


Fig.: 1 Shape Based vs. Grid-mapped Autorouters

## 1.2. Proceeding of SPECCTRA Autorouter

SPECCTRA extends the PCB CAD system by adding automatic and interactive placement and routing tools.

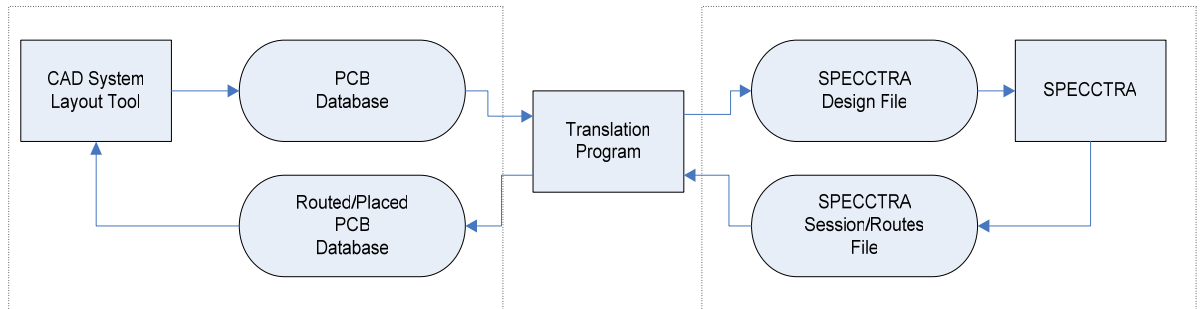


Fig.: 2 Design Flow of SPECCTRA

**File Types:** Each PCB layout system stores design information in a unique format! All files that are read and written by SPECCTRA are plain text files, which are shown in the following.

design	.dsn	Contains PCB boundary data, layer definitions, padstack definitions, component data, netlist, preroutes and design rules
session	.ses	Contains a pointer to the original design file
routes	.rte	Contains route data that can be translated to your layout system
wires	.w	Contains route data that can only be read by SPECCTRA

## 2. Basic Handling of the Autorouter

### 2.1 General Description

**Autoroute Tools:** 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. So SPECCTRA reduce the conflicts automatically! The conflicts are shown in Fig.: 3.

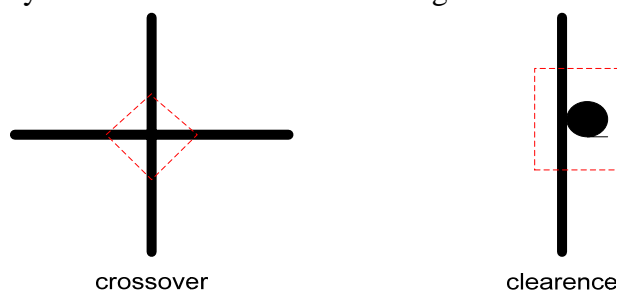


Fig.: 3 Difference between crossover and clearance conflicts

**Do file:** SPECCTRA can be controlled either using a GUI or a text-based “do file”.

A basic “do File” includes:

- Setup and file management
- Rule setting
- Prerouting
- Routing
- Postrouting

***Do File Example:***

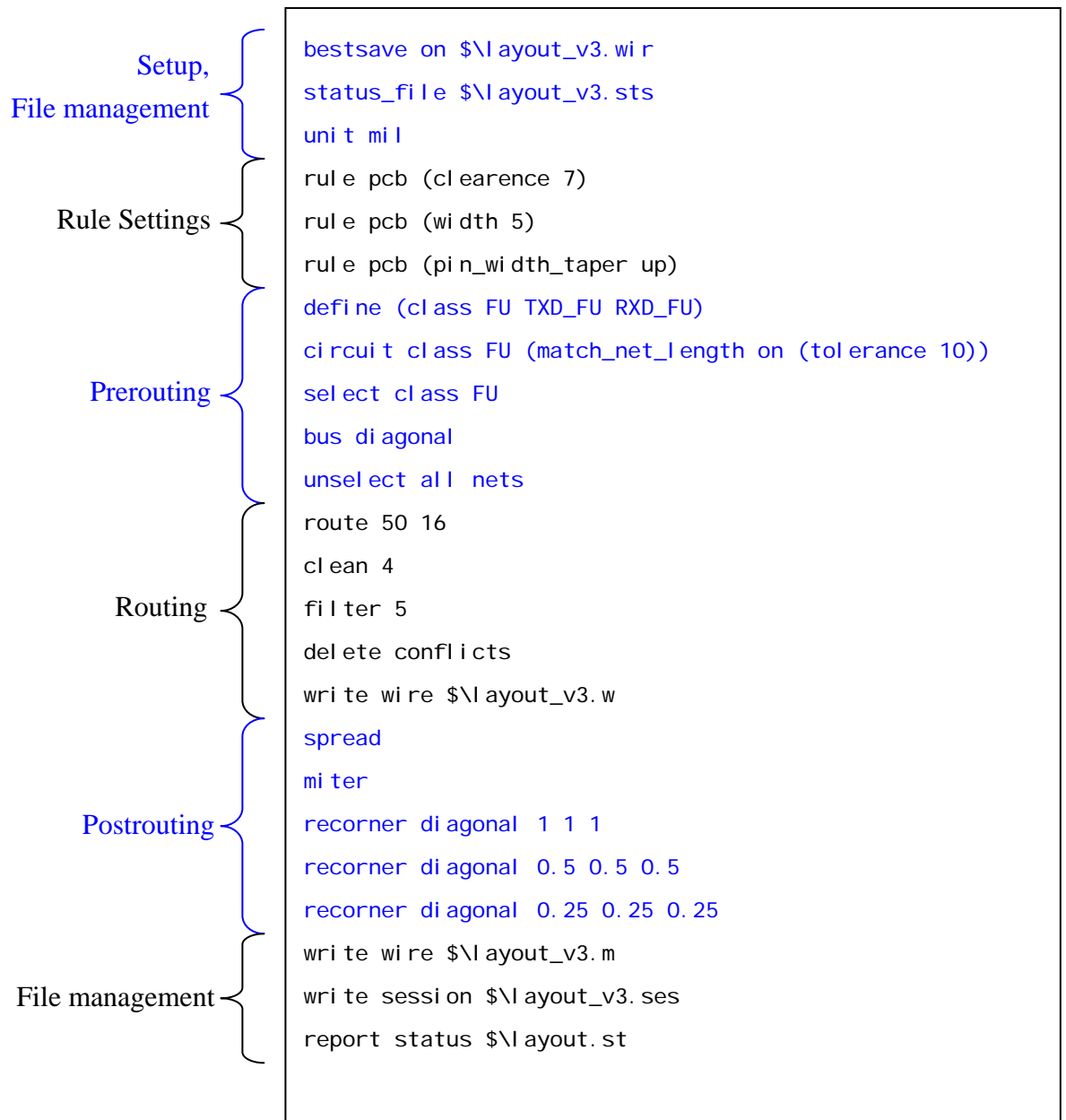


Fig.: 4 Basic Do File

**Interactive routing:** The interactive tools use the same Shape Based technology and gridless environment as the AutoRoute tools.

You can use the interactive tools for:

- Route critical connections before Autoroute
- Reroute connections after Autoroute

You set the interactive routing and editing environment by choosing “Setup” from the Interactive Routing menu. The Interactive Routing menu is displayed, by pressing the right mouse button in the work area. You can use the Interactive Routing Setup dialog box to define grids, set global (PCB) clearance and wire width rules, and to set the interactive routing environment in three different categories:

- General
- Bus
- Style

**Status report:**

**Overview report:**

```
#SPECCTRA ShapeBased Automation Software V9.0.3 made 2000/03/24 at
16: 48: 18
#Host
#ROUTING STATUS <<< C:\4_LAYER\Logi kpl ati ne. dsn >>>
Start Time: Wed Nov 30 15: 34: 27 2005
Report Time: Wed Nov 30 15: 42: 17 2005

Nets = 133 Connections = 437
Current Wire = 9 Reroute wires = 9
Completion = 98.17% Unconnections = 3
```

Fig.: 5 Overview Report

Nets	Defines the number of nets on your PCB board
Completion	Indicates how far the PCB is ready
Connections	Shows the number of connections
Unconnections	Indicates the unrouted nets

### Routing Pass Information:

ROUTING HISTORY												
Pass Name	No.	Conflicts		Fail	Unrte	Vi as	XTal k	Len.	Red %	CPU Time		
		Cross	Clear							Pass	Total	
Route	1	400	164	48	3	439	0	0	0	0: 00: 02	0: 00: 02	
Route	2	408	94	4	3	346	0	0	10	0: 00: 02	0: 00: 04	
Route	3	214	73	0	3	358	0	0	42	0: 00: 02	0: 00: 06	
Route	4	129	46	0	3	369	0	0	39	0: 00: 01	0: 00: 07	
Route	5	85	33	3	3	391	0	0	32	0: 00: 01	0: 00: 08	
Route	6	43	14	0	3	409	0	0	51	0: 00: 02	0: 00: 10	
Route	7	30	6	0	3	417	0	0	36	0: 00: 01	0: 00: 11	
Route	8	22	8	0	3	418	0	0	16	0: 00: 01	0: 00: 12	
Route	9	19	8	0	3	424	0	0	10	0: 00: 01	0: 00: 13	

Fig.: 6 Routing Pass Information

### Summary Information:

#WIRING STATISTICS												
PCB Area=39005150.650 EIC=46 Area/EIC=847938.058												
Components=131 SMDs=100												
Signal Layers=2 Power Layers=2												
Wire Junctions=79, at vias=49 Total Vias=431												
Conflicts=4 Crossovers=2, Clearances=2, Crosstalk=0, Length=0												
Manhattan length=462656.560 Horizontal=230518.290 Vertical=232138.270												
Routed length=547007.044 Horizontal=278752.930 Vertical=268254.700												
Ratio Actual / Manhattan= 1.1823												
Unconnected length=11780.000 Horizontal= 3930.000 Vertical= 7850.000												
Layer Direct Pins Vias TJs Conflicts Length Horizontal Vertical												
Top Horz 442 431 26 2 289947.104 207062.580 82885.110												
Bottom Vert 389 431 4 2 257059.940 71690.350 185369.590												

Fig.: 7 Summary Information

## 3. Essential Commands

### Circuit Command:

Length	Control max. and min. length of a net
Delay	Control max. and min. delay of a net
Noise and Crosstalk	Control automatic shielding Set the switch window and sample window to define net coupling relationship
General	Control routing priority
Autorouting	Limit routing length

### Example:

```
circuit class AdrBus (match_net_length on (tolerance 10))
```

**Class Descriptor:** A class is a convenient way to manage more than one net at the same time. You can assign a certain number of nets to a class. So you can edit the properties of those nets.

**Example:** `define (class AdrBus A1 A2 A3 A4 A5 A6 A7 A8 A9 A9 A10  
A11 A12 A13 A14 A15 A16 A17 )`

**Clearance Descriptor:** The clearance descriptor sets a rule that controls clearances between routing objects in your design. Use the clearance rule to set the minimum distance between wires and objects. A value of 0 means the edges of objects can meet each other. A value of -1 means the rule is not specified.

**Example:** `rule pcb (clearance 7)`

**Layer Rule Descriptor:** With the Layer Rule descriptor you can set PCB rules for the entire routing process.

**Example:** `Layer_rule Layer1 (rule pcb (clearance 7))`

**Length Descriptor:** The length descriptor sets a circuit rule that controls maximum and minimum routed wire lengths.

**Example:** `circuit class ID1 (max_length 5000)`

**Route Descriptor:** The route command starts the Autorouter. You can use route without a pass number to run a single Autorouting pass, or you can specify a number of Autorouting passes.

You use the route command to:

- Start the initial Autorouting of a PCB
- Specify the number of routing passes

**Example:** `route 25`

Standard routing passes are 25 or 50. The numbers of routing passes are dependent on the degree of difficulty.

**Shield Descriptor:** The shield descriptor sets a circuit rule which controls whether shielding is applied to wires.

You can use three keywords:

- Parallel
- Tandem
- Coax

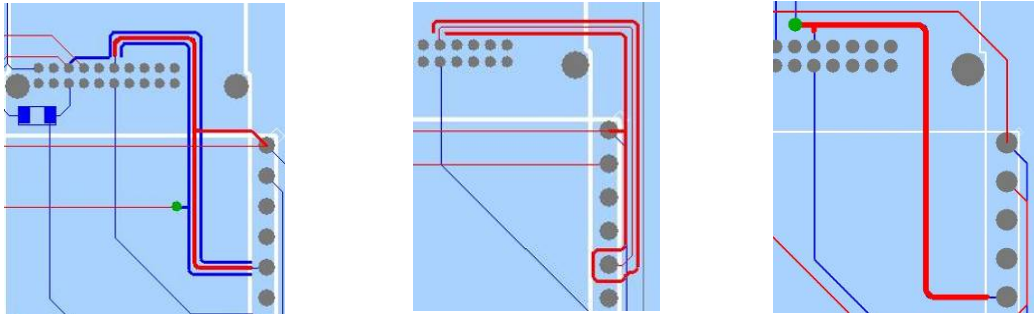


Fig.: 8 Kinds of shielding (from left: coax-, parallel-, tandem shielding)

Shields are routed during automatic and interactive routing of nets.

**Example:** `circuit net CLK (shield on (type coax) (use_net GND))`

**Width Descriptor:** The width descriptor sets a rule that controls the wire width.

**Example:** `rule pcb (width 5)`

### 3.1 Basic Configures

**Smart Route:** The easiest way to route your PCB is to use the command “smart route”. SPECCTRA will use the default settings and route it without any rules, but this is at the most times not really helpful. If you click Autoroute → route, a little checkbox will open! Click to the check box smart route. Here you can only change the minimum grids, the fanouts, generate testpoints and miter after routing. There are already two ways to start smart route. The first is to tip

the command in the command bar and press enter. SPECCTRA will use the default settings of:

- Minimum via grid (0.1)
- Minimum wire grid (0.1)
- Fanout if Appropriate  
(Via sharing *on*, pin sharing *on*)
- Generate testpoints *off*
- Miter after Route *off*

Click on Autoroute → Route and a little dialog box will open! In this Dialog box you can change the default settings! For example Miter after Routing should be on, because all wires have to be in a minimum 45° angle!

**Miter:** The command “miter” checks the PCB for 90° angled wires and rounds them out! Click ok and SPECCTRA Autorouter will perform your whole PCB!

**Fanout:** The command “fanout” escapes SMD pads and through-pins to a via. Click autoroute → pre route → fanout and you can change your settings or leave it to the defaults!

**Clean:** The command “clean” rips up and reroutes your connections. Adding new conflicts is prohibited.

**Bus routing:** To accomplish the command bus routing can also be done in two ways. First is already to tip the command in the command bar. Here are two different commands possible:

- `bus diagonal`  
-- SPECCTRA uses diagonal wire segments
- `bus orthogonal`  
-- SPECCTRA uses orthogonal wire segments

The other way is to click on the button Autoroute → Pre Route → Bus Routing and select the desired command.

**Crosstalk:** We talk about Crosstalk when the spacing between the traces is too narrow and the electromagnetic fields of the signals traveling along the adjacent traces interfere. Crosstalk is shown in Fig. 9.

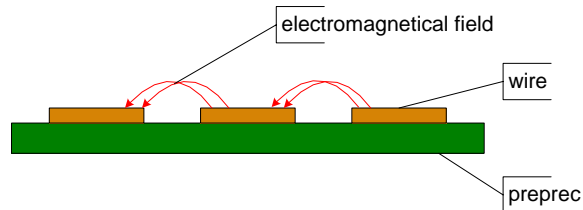


Fig.: 9 Crosstalk – worst case

Crosstalk can be corrected by increasing the spacing between the spacing between tracks as you can see in Fig. 10.

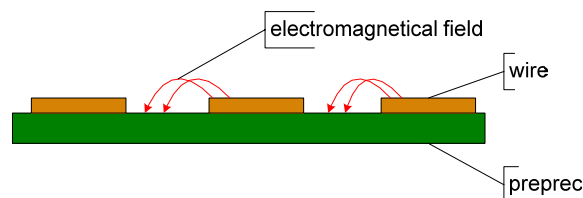


Fig.: 10 Crosstalk – best case

To set these spaces with SPECCTRA is very simple! First you have to decide where on your board should be increased the spacing between traces!

To calculate the minimum Distance you have to:

- Know the signals on sensitive traces
- Calculate the round trip time of the trace

For short sections, where wires come to a minimum distance together, the phenomenon “Crosstalk” can be disregarded, because “Crosstalk” does not have time to develop.

Formulas for estimating crosstalk are listed in [SIS1].

```
Used Commands: rule net CLK1 (max_noise 700)
                rule pcb (parallel_noise (gap 8) (weight 5))
                rule pcb (parallel_segment (limit 25) (gap 10))
                rule pcb (tandem_noise (gap 6) (weight 4))
                rule pcb (tandem_segment (gap 5) (limit 3000))
```

**Daisy Chain or Starburst wiring:** Nets are usually ordered for starburst routing. Starburst is the most efficient one because SPECCTRA orders and routes starburst nets with minimum wire lengths.

**Different between Daisy Chain and Starburst:**

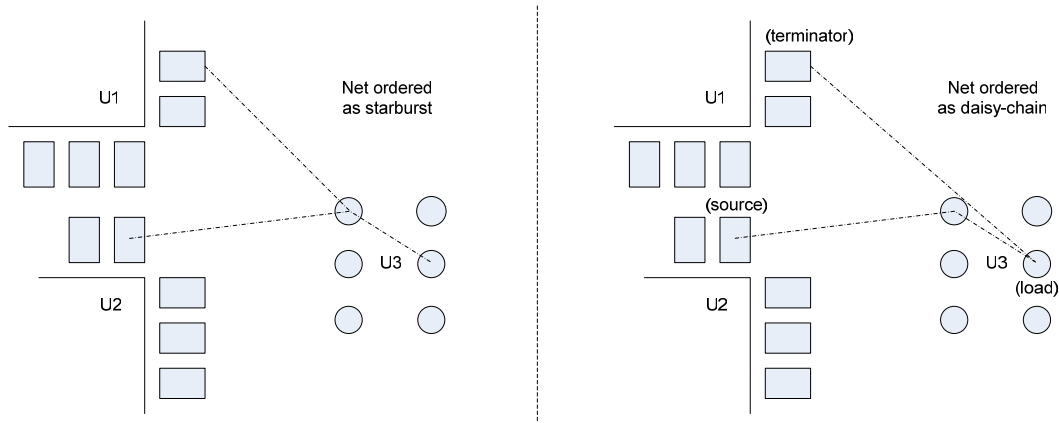


Fig.: 11 Starburst vs. Daisy Chain

To select other nets you need the command “order” and “assign”.

**For Example:** `assign_pin source U1 (pins 2)`  
`assign_pin terminator U2 ( pins 23)`  
`order daisy net sig1`

The net sig1 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 will chain to loads and load pins chain to terminators. cf. Fig. 11.

## 4. Special Commands for Do Files

### 4.1 Division power electronics

**Component properties:** You can assign physical properties to components and images. Physical properties consist of:

- Type
- Height
- Power dissipation

To add, change, or remove component properties in SPECCTRA you have to click Define → Properties → Component and SPECCTRA will open the Define/Forget Component Properties dialog box.

Used Commands: `component_property R1 R2 (type resistor)`  
`component_property IC14 (power_dissipation 500)`  
`forget component_property (component U4) (property height)`

**Using Thermal Constraints:** Thermal constraints limit power dissipation in critical areas of the PCB. You set thermal constraints by:

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

Used Commands: `define (room r1 (rect bottom 1.550 4.890 7.630 9.750)`  
`(height 0.5 -1))`  
`room_rule memory (include U1 U2 U16 (type hard))`  
`(exclude remain (type hard))`

### Controlling wire width:

**Controlling “fromtos” connected to virtual pins:** You control the width of a wire between a physical pin and a virtual pin by using the define-net command. This command creates the virtual pin and assigns a “fromto” rule. For example, to control the wire width of “fromto” U1-1 connected to virtual pin VP1 on the net CLK1, add the following commands to your do file.

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

You can see the result in Fig. 12.

**Reducing or enlarging wire width:** With the “pin width taper” rule you can reduce or enlarge the wire width during the routing process. Before SPECCTRA will connect the wire to the pad, he enlarge the wire width as the same diameter of the pad. For example, to enlarge all wire segment widths to the same as the connecting pin, add the following command to your do file.

```
rule pcb (pin_width_taper up)
```

You can see the result in Fig. 12.

**Setting global width rules:** To set a global width rule of 10 mils between all object types, add the following commands to your do file.

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

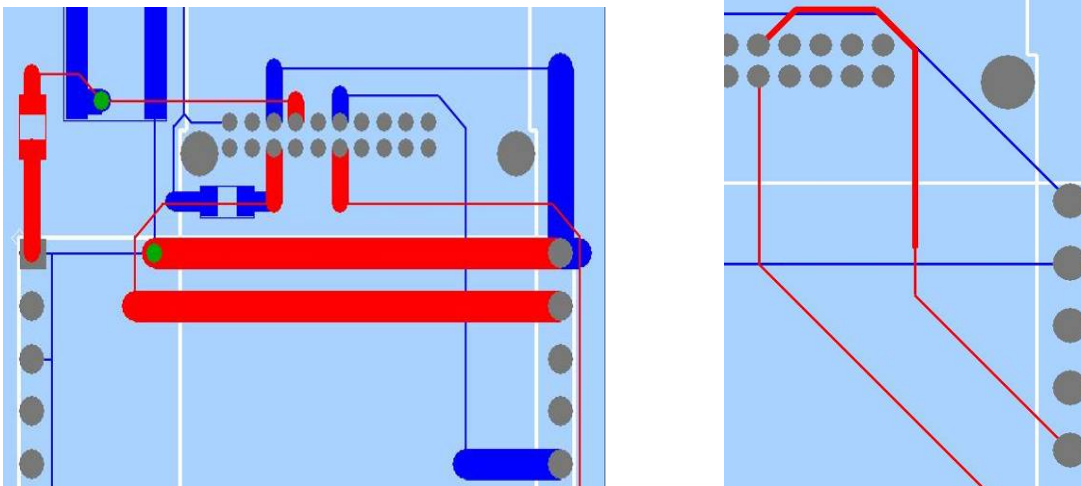


Fig.: 12 Left: pin width taper rule, Right: virtual pin rule

### Setting Clearance rules:

**Setting via to via clearance:** If the clearance between vias is too small, SPECCTRA can't create copper planes over the whole size of those two vias, cf. Fig.13. You can set a global pcb rule for the clearance between vias. This rule is called via-to-via clearance. Add the following command to your do file.

```
rule pcb (clearance 50 (type via_via))
```

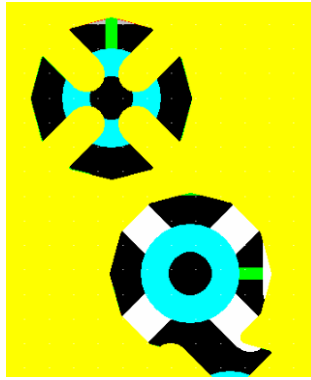


Fig.: 13 Via Clearance

**Setting a SMD-to-via clearance rule:** You can set a global pcb level that is called SMD-to-via clearance rule to control the minimum distance between SMD pads and vias. For example, to set a clearance of 25mil between SMD pads and vias on the same net, add the following command to your do file.

```
rule pcb (clearance 25 (type smd_via_same_net))
rule pcb (clearance 25 (type smd_via))
```

**Setting global clearance rules:** It is very useful to set global clearance rules, because manufacturer can only print the board with a minimum required distance between wires, vias and all other printed images on your board. To set a global clearance rule of 10 mils between all object types, add the following command to your do file.

```
rule pcb (clearance 10)
```

## 4.2 Division high frequency technique

### **Placing a guard ring for high sensitive signal wires and high datarate signals:**

You can shield critical nets with a power or ground net on the same layer (parallel shielding) or on adjacent layers (tandem shielding). Coax shielding combines parallel and tandem shielding. You can control the shield wire width and the gap between shield and signal wires in parallel shielding, and the overhang width of tandem shield wires.

***For Example*** `circuit net sig1 (shield on (type coax) (use_net GND))`  
`rule net sig1 (tandem_shield_overhang 5)`  
`rule net sig1 (shield_width 12)`

```
rule net sig1 (shield_gap 10)
```

In this example net sig1 will be shielded by a coaxial shield. The first line describes the net sig1 with a coax shield. The used net for the shield is net GND.

In the next 3 lines you define some rules for net sig1:

- Tandem\_shield\_overhang 5 → to specify the extra amount added to each side of the tandem shield wire.
- Shield\_width 12 → The width of the Shield is 12 mil
- Shield\_gap 10 → The gap between the Shield and net sig1 is 10 mil

Result:

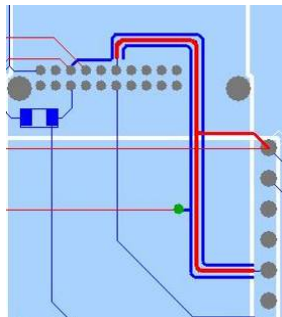


Fig.: 14 Coaxial Shielding

**Note:** Sensitive wires for example are:

- Clock circuits
- Transmit and receive circuits

**Ordering Power planes:** In high frequency techniques there have to be planes, because if you do not have planes the Loop area will be big. A big Loop area produces problems. One of these problems is Delay. For creating power planes you have to:

- Control the Impedance,
- Keep the Loop areas small,
- and control Crosstalk

Used Commands:

```
rule class CLASS1 (tjunction on) (junction_type
supply_only)

(pcb split_plane (structure
(layer s1 (type signal) (direction horizontal))
(layer p1 (type power) (use_net +5V GND))
(layer s2 (type signal) (direction horizontal))
(plane +5V (polygon p1 0.010 0.560 0.160
0.560 1.480 1.00 1.480 1.00 0.700 1.280
0.700 0.560 0.160))
(plane GND (polygon p1 0.010 1.740 1.480
1.740 0.160 1.300 0.160 1.300 0.720 1.740
1.480))
```

The last command describes two planes (split – plane, cf. Fig. 15)

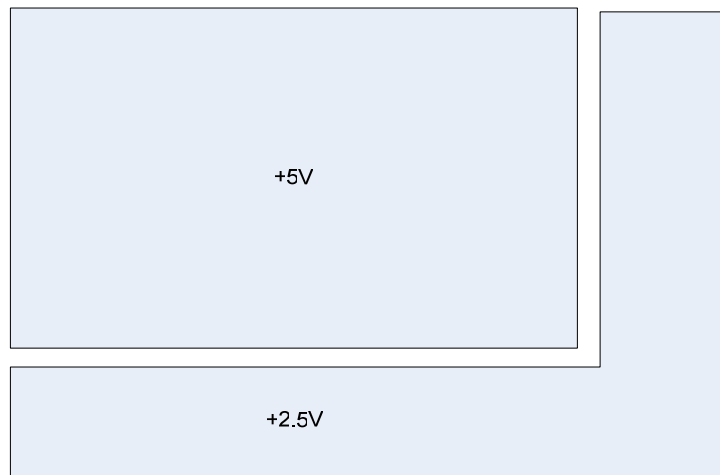


Fig.: 15 Split planes

### 4.3 Division bus systems

**Preventing rerouting particular wires:** In the routing phase, a bus should be routed first and after it should be protected. Use the following command.

```
protect all wires (attr bus)
protect all wires (attr fanout)
protect all testpoints
```

These Commands are very useful, if you won't change the position of the bus, testpoint, or fanout!

**Manage the length on bus systems:** The most important thing if you ever route a bus is to manage the length on all wires of your bus. The length of all wires of your bus should have the same length. Use the following command.

```
define (class BUS1 ADR0 ADR1 ADR2 ADR3 ADR4 ADR5 ADR6
      ADR7 ADR8 ADR9 ADR10 ADR11 ADR12 ADR13 ADR14
      ADR15)
circuit class BUS1 (match_net_length on (tolerance
      10))
```

**Manage the maximum and minimum length:** Sometimes the length of a wire must have a restricted length. For example, to assign actual maximum and minimum length rules to net SIG3 and assign maximum and minimum ratio rules to SIG4, add the following commands to your do file.

```
circuit net SIG3 (length 3100 3000 (type actual))
circuit net SIG4 (length 1.2 1.1 (type ratio))
```

When, the maximum bus length should be 3 cm, use the following command!

```
circuit class BUS1 (max_length 1181)
```

## 5. Literature

- [SP1] Cadence Design Systems, SPECCTRA Tutorial, Product Version 9.0, August 1999
- [SP2] Cadence Design Systems, SPECCTRA Design Language Reference, Product Version 9.0, August 1999
- [PCB1] Christopher T. Robertson, Printed Circuit Board Designer's Reference, Prentice Hall PTR, October 2003
- [SIS1] Douglas Brooks, Signal Integrity Issues and Printed Circuit Board Design, Prentice Hall PTR, June 2003
- [PCB2] David L. Jones, PCB Design Tutorial, June 2004
- [SIC1] Mick Grant, Signal Integrity Considerations for High Speed Digital Hardware Design, Calyptech, November 2002
- [PCB3] Henry W. Ott, PCB Design Guidelines, Henry Ott Consultants, 2000