

## Table of Contents

# Welcome to



[www.holophase.com](http://www.holophase.com)

- Chapter 1      [Introduction](#)
- Chapter 2      [Getting to know CIRCAD'98](#)
- Chapter 3      [The Working Commands](#)
- Chapter 4      [Setup Commands](#)
- Chapter 5      [File Commands](#)
- Chapter 6      [Block Commands](#)
- Chapter 7      [Netlist Functions](#)
- Chapter 8      [From Schematic to PCB](#)
- Chapter 9      [Printing](#)
- Chapter 10     [Gerber Photo Plots](#)
- Chapter 11     [Creating Components](#)
- Chapter 12     [Customizing Template Files](#)
- Chapter 13     [Shortcuts and Other Features](#)
- Chapter 14     [Un-Do Commands](#)
- Chapter 15     [Pixel Imaging](#)

## Chapter 1

# Introduction

CIRCAD'98 is a menu-driven printed circuit design and editing program. It is an easy program to learn, because nearly all the information needed to operate the program can be accessed through pop down and dialog boxes. The purpose of this manual, therefore, is not to teach you how to use CIRCAD'98. It will instead serve as a guide so that you have a ready reference to turn to if you get stuck.

One of most unique features in both CIRCAD and CIRCAD'98's is that they do not employ separate "modules" for schematics and PCB design. Rather, they allow you to handle all elements of PCB design -- from schematic to Gerber output -- from within one single, powerful program.

One advantage of this integrated approach is that you needn't key in the same information more than once. Component and signal data you enter at the schematic stage translates directly into information CIRCAD'98 will use to help you set up the PCB file later, cutting down on extra keystrokes and the potential for data entry errors.

CIRCAD'98 allows you to lay out circuit boards in great detail without the arbitrary restrictions found in most CAD packages, and without the tedium and bother of tape and decals. The ultimate product of CIRCAD'98 is data files that will directly drive a Gerber photoplotter in producing the artwork for a finished board (for more on circuit board manufacture, see **Chapter 10**). In addition, CIRCAD'98 can produce drill plots, parts lists, and import/export data from a number of other CAD packages.

CIRCAD'98 is for both the novice and the experienced CAD user. The first-time CAD user will find the manual invaluable in becoming familiar with CIRCAD'98, while the experienced user will probably jump right into the program, only occasionally referring to the manual.

- [On-line Manual](#)
- [File Backup](#)
- [File Import/Export Filters](#)
- [Autopanning](#)

## On-line Manual

An complete reference manual is on-line and available at any time by the Help menu or <F12> on the keyboard. The online manual is both interactive and context sensitive, providing instant access to information about whatever CIRCAD'98 feature is currently in use, without a lot of searching and guessing at key words.

## File Backup

Each time you exit from CIRCAD'98, the program will display, in sequence, each of the files you have modified, with the warning: **File has been updated but has not yet been saved. Save file before exit?** Simply selecting **No** leaves the file as it was when the session began. Selecting **Yes** saves your changes to disk.

CIRCAD'98 also contains an automatic file backup feature. If, at the end of a session, you elect to save the changes you've made to a file, CIRCAD'98 does so -- but first it moves the old version to a sub directory called 'BACKUP'. If no backup directory is available, a backup copy of the file is created in the same directory by appending the suffix '.BAK' to the file.

This enables you to recover quickly should you unintentionally save undesirable changes to one of your files. To keep your hard drive from quickly filling up with outdated files, CIRCAD'98 keeps only one previous version of each file in the BACKUP sub directory.

## File Import/Export Filters

CIRCAD'98 can automatically read (import) files that were created using other circuit board design programs, and can also write out (export) files compatible with those programs. To utilize this feature select the desired filter from the file open or save dialog boxes.



## Autopanning

Screen autopanning is a very useful CIRCAD'98 feature. A full explanation of autopanning can be found in **Chapter 4** under *Setup: Draw Modes*.

## Chapter 2

# Getting to know CIRCAD'98

CIRCAD'98 has an easy to use and intuitive user interface. Most of the commands are two letter functions such as **<PL>** for Place Line, or **<GC>** for Grab Component. This chapter will give a brief overview of some of the basic functions.

- **Using the Mouse**
- **The Cursor**
- **Opening a File**
- **The Status Line**
- **File Information**
- **The CIRCAD'98 Menu**
- **Zoom Commands**

## Using the Mouse

For all functions and operations, the left mouse button is equivalent to the Enter key. When the CIRCAD'98 cross hairs are engaged the right mouse button is equivalent to the Escape key with the exception of a click and hold which is an alternate method of panning.

This manual will often instruct you to click on something, such as a menu selection. This means either to press the left mouse button or press Enter on the keyboard. The result in either case will be the same.

The right mouse button is equivalent to the Escape key.

## The Cursor

The first thing you will see when you invoke CIRCAD'98 is the title page against a gray background, and the main menu bar across the top of the page. The title page will disappear when you begin working in the program, but the gray background is something you'll be seeing a lot of -- it is essentially the workbench on which you'll be building your circuit boards.

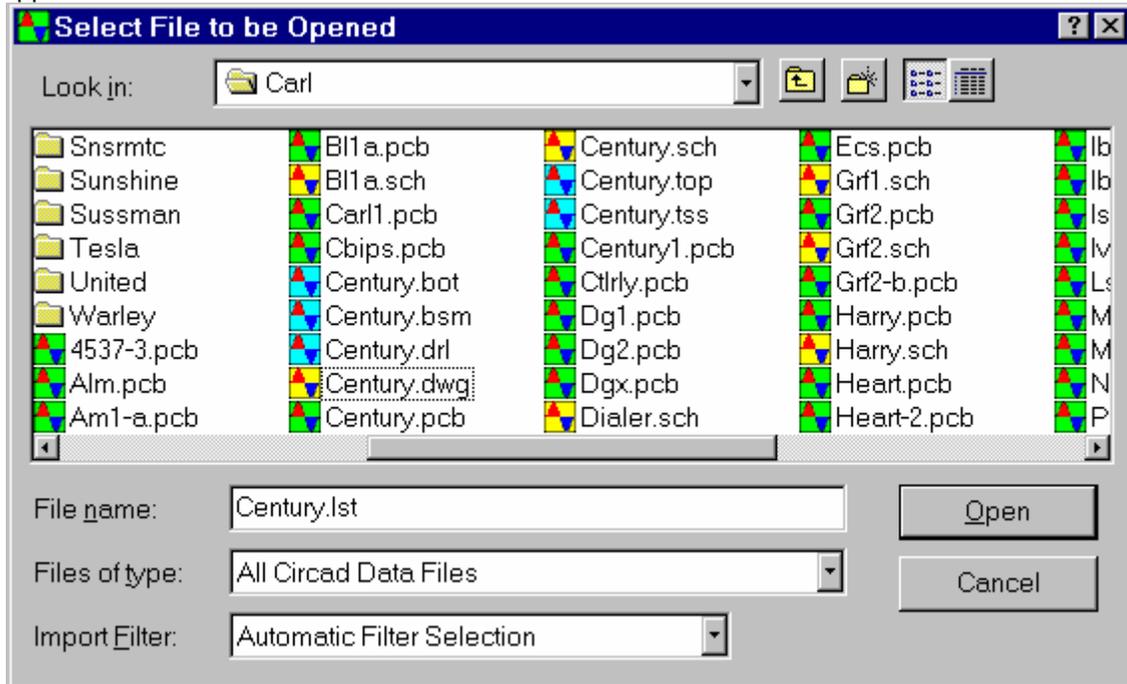
You will see the cursor also: a small arrow. If you press left mouse or Enter, the cursor becomes a pair of x and y cross-hairs.

When using CIRCAD'98 you'll use the cross-hairs when working on the "bench", but you'll use the arrow to make menu selections. A third cursor for data entry will be introduced shortly.

Note that in addition to using the mouse, you can move the cursor with the arrow keys on the keyboard. This is a useful feature for precisely placing components and tracks, but the mouse is essential for speed and for the large movements required to use the menus.

## Opening a File

To experiment with CIRCAD'98 and its features, you must first open a file. Right mouse to engage the Windows cursor and cause the main menu bar to appear, then move the arrow cursor to the first selection, **File**. Left mouse or Enter to display the pop down menu. Move the cursor to **Open** and click on it again, and a dialog box (not attached to the top of the screen) appears.



At the top of this dialog box, the "Look In" field will show the current folder you are working from. The various sub directories and files are listed below in the file list box.

Left mouse the folder named **DEMO** and double click it to open the sub directory. Scroll down to the file named **FL1-C.PCB** and click on it to open it for processing.

When the file is loaded, a demonstration circuit board will be displayed on the screen, automatically zoomed to the largest size that will fit on the screen.

## The Status Line



At the bottom of the screen is the *status line*, containing several items of information. Unlike the main menu bar, which appears and disappears according to the placement of the cursor, the status line is always present.

The cursor location coordinates are at the left of the line. Move the crosshair cursor down to the far lower-left corner of the screen, and both numbers will go to zero. As the cursor is moved to the right along the bottom of the screen, the first number (the x coordinate) will grow larger. This figure indicates distance, in inches and thousandths, from the lower-left corner, or origin. As the cursor is moved upward, the second number (the y coordinate) will behave similarly.

The workspace can extend from -32,768 inches to +32,767 inches in both the x and y directions.

After the y coordinate value you may see an asterisk. This indicates that an artificial zero point has been set. See Setup Relative Zero Point for further information.

When you are working in a file, the next item on the status line displays the current working command. Assume, for example, that the last thing you did was place a line. CIRCAD'98's status line will display the words **Place: Line** as a reminder that this command is in force until you select another. When a file is first opened, no command is displayed on the status line because a working command has not yet been selected.

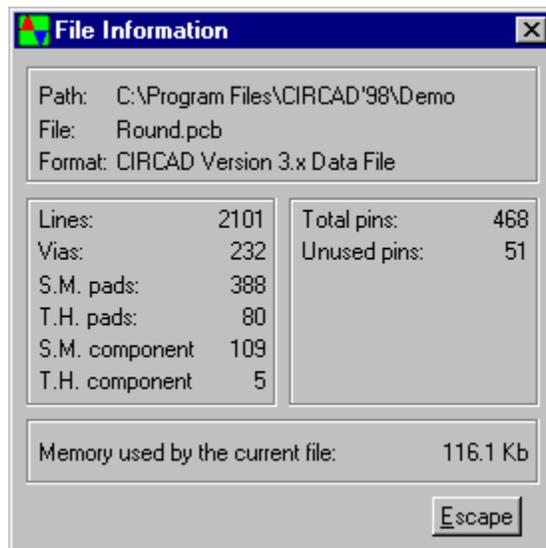
Next is **Grid**. The cursor can be set to move in any increment you select. This setting is known as *grid snap*, and the current setting is displayed here on the status line as 0.025". Setting and selecting grid snaps is discussed in **Chapter 4**.

The next item on the status line is *Layer*. CIRCAD'98 allows you to create boards with one Pad Master layer and as many as 255 user layers. The Pad Master layer is used for through-pads, while the other 255 user layers can be defined as signal, silk screen, or solder mask layers. The status line will always display the name of the current active layer. The color in the box will match the color you've assigned to that layer.

At the extreme right-hand end of the status line, you will find the current time of day displayed as a simple digital clock. The colon separating the hour and minute fields will blink one second on and one second off.

When appropriate, the status line will also briefly display prompts to let you know what to do next.

## File Information



A complete discussion of the various *File* functions is contained in Chapter 5, but it is useful at this point to look at the information available through the *File Information* function.

The File Information dialog box is intended for quick reference, and can be called up at any time.

# The CIRCAD'98 Menu

The chief purpose of CIRCAD'98 is to place, move and modify tracks and components on the circuit board. To facilitate this, each CIRCAD'98 command involves two parts:

- 1) the *action*, and
- 2) the *object* of the action.

For instance, you can either place, grab, edit, or delete something. That *something* can be a line, a pad, a component, or a piece of text, for example. Virtually all CIRCAD'98 commands, therefore, are basically composed of two words.

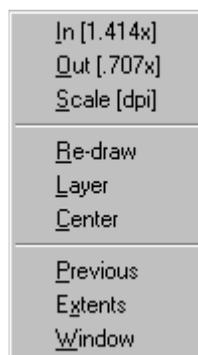
The main menu bar displays the choices for the first word of the command. Clicking on one of the choices displays a pop down menu that lists the choices for the second word of the command. If, for example, you wanted to place a line, you could click on **Place** in the main menu, and then on **Line** in the pop down menu.

## Command shortcuts

Note that a single letter in each of the selections is highlighted. This introduces another useful feature of CIRCAD'98: you can perform menu selections with the keyboard as well as with the mouse. As you grow more proficient with the program, for instance, you may find it quicker to place a line using the keystrokes <PL> rather than by clicking on **Place** and **Line** with the mouse. This holds true for the other selections on the menu as well, although it may not always be the first letter of the selection that is highlighted.

## Zoom Commands

The first order of business is to learn how to manipulate the screen and get the circuit board zoomed up to a useful size. Click on **Zoom** and you'll see a selection of available zoom commands. To initiate one of the commands, move the cursor to the desired choice and *click* on it.



### Zoom In

Magnifies the file view by a factor of 1.414, reducing the area viewed by about half, with the center of the screen as the reference point.

### Zoom Out

Does the opposite of *Zoom In* by reducing the file view by a factor of .707, doubling the area viewed.

## **Zoom Scale**

Displays the circuit board at a specified number of dots per inch (dpi).

## **Zoom Re-draw**

Clicking on **Re-draw** will immediately regenerate the screen afresh.

## **Zoom Center**

Allows you to select a reference point which will then become the screen's center. Clicking on this selection changes the cursor to a cross-hair and produces a prompt on the status line **Select central reference point**. Position the cursor wherever you'd like the new screen center to be and click the left mouse button.

## **Zoom Previous**

Restores the previous zoom settings. This is only a one position buffer, so each zoom discards the prior elements predecessor.

## **Zoom Layer**

Should you wish to change the layer you are working on (or active layer), click on the **Layer** selection. A dialog box will appear with the title: **Select layer** All 256 layers (255 plus Pad Master) will be displayed by name, with a color brick beside the layers in use to indicate the colors they've been assigned. The currently active layer is hilled.

To change the active layer, just position the cursor on the desired layer and click on it. The screen will then be re-drawn with the active layer shown "on top" of the other layers. The status line will indicate the name and color of the new active layer.

As an added convenience, the + and - keys on your keyboard's ten-key pad can be used to step up or down through the layers without invoking the *Zoom Layer* command.

## **Zoom Extents**

Clicking on **Extents** zooms the screen in to the largest magnification that will still display the whole circuit board. This is a heavily used zoom feature.

## **Zoom Window**

Another heavily used zoom feature, **Window** allows you to define a specific area and display it at maximum size on the screen. Clicking on Window produces the prompt **Select first corner point**. Position the cursor and press left mouse or Enter and the prompt will instruct you to **Select second corner point**. As the cursor is moved, the outline of the zoom window is shown. Pressing left mouse or Enter again zooms the screen in to display that window at maximum size.

Now, zoom the display to **Extents** to explore some of the other CIRCAD'98 commands. You should see the demonstration circuit board shown at the maximum size that will fit on the screen.

## Chapter 3

# The Working Commands

Place, Grab, Edit, and Delete

These four commands are the "action" commands, allowing you to install, relocate, modify and remove items as necessary. As you explore the commands, feel free to experiment with them on the demonstration file -- you will not damage it. As explained previously, at the end of each CIRCAD'98 session, you will be asked if you wish to save the changes you have made to each file you have edited. Simply selecting **No** leaves the file as it was before the session began.

- **Place**
- **Grab**
- **Edit**
- **Delete**
- **Shortcut Keys**

# Place

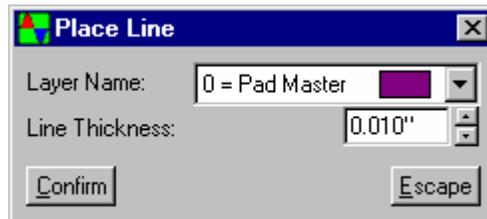
The Place functions are used to add items to a new or existing data file. The items to be placed range from primitive elements such as lines or arcs to complex constructions such as components and blocks. Once an item has been placed, it can be moved (by using the *Grab* functions), modified (by using the *Edit* functions), or erased (by using the *Delete* functions). This enables you to change your drawing as often as you wish, so that your circuit board can improve as the design evolves.

<u>L</u> ine
<u>O</u> rtho-line
<u>V</u> ertex
<u>A</u> rc/circle/ellipse
Arc graphically
<u>P</u> ad
<u>V</u> ia
<u>T</u> ext
<u>C</u> omponent
<u>F</u> illed Solid
<u>S</u> ignal Port
<u>B</u> us Port
$\pm$ - Port
Signal <u>G</u> ND
Earth <u>G</u> ND
Named <u>G</u> ND
Signal <u>N</u> ame
Signal <u>D</u> ot
Signal <u>W</u> ire
Bus <u>W</u> ire
<u>A</u> rowhead
<u>X</u> Y Dimension

## Place Line

The *Place Line* function is used for placing tracks on PCB boards and any drawing functions you may need, such as drawing board or component outlines. Note that for signal traces on schematics, you will want to use the specialized function *Place Signal Wire* instead.

To place a line, you'll need to define its thickness and the layer where you want it. Just click on **Line** and a dialog box will appear, asking for line parameters.



To change the target layer for the line, simply click on the arrow to the right of the layer name field. A drop down list will appear with all the available layers with the current layer highlighted. Select the new layer by simply clicking on your choice. If you are finished editing line parameters, simply click on **Confirm** and the **Line Parameters** menu will disappear. If however, you also want to change the line's thickness, don't click on **Confirm** just yet.

The thickness of the line to be placed is indicated in mils. To change the thickness, click on the digits, and you'll see the third type of cursor: the *data entry* cursor.

The cursor blinks as a prompt to let you know that CIRCAD'98 requires information in order to proceed. Type in the new line thickness and press **Enter**. The cursor (having changed to an arrow once again) will move automatically to the **Confirm** option. Click on it and the dialog box disappears.

Note that the status line at the bottom of the screen now indicates **Place: Line**. This will be the operative function of the mouse until you select another function, making it quick and easy to place a number of lines in succession.

The next time you click the mouse button, you'll place the first end (origin) of the new line. As you then move the cursor about, the "ghost" (or thin outline) of the line will follow it around. Pressing **right mouse** or **Escape** interrupts the process and lets you select a new origin. Pressing **left mouse** or **Enter** again nails down the endpoint of the line and the ghost becomes a fully drawn image.

At this point you can continue to place contiguous line segments by simply moving the cursor and clicking wherever you want an endpoint. You can also start over with a new (non-contiguous) segment by pressing **right mouse** or **Escape** and selecting a new origin.

Again, note that the mouse will remain in the Place: Line mode until you select another function.

## Place Ortho-Line

An ortho-line is defined as a line that can be placed only in increments of 45 degrees. This is very useful in placing lines perfectly parallel or perpendicular to one another. Placing ortho-lines is done in exactly the same manner as placing a line.

Note that this orthogonal snap mode is also available for all *Grab* functions as well, so that you can grab any element and adjust its position straight left, right, up, or down. This feature is discussed more fully in **Chapter 13** under *Ortho Snap mode*.

## Place Vertex

Place Vertex is most useful for "picking up" a line in the middle and "bending" it. Click on **Vertex**, position the cursor on a line you've just placed and click on it -- now moving the cursor moves the endpoint and pulls the line (now actually two lines) along behind it.

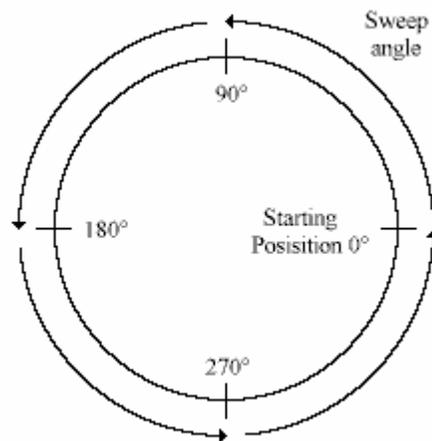
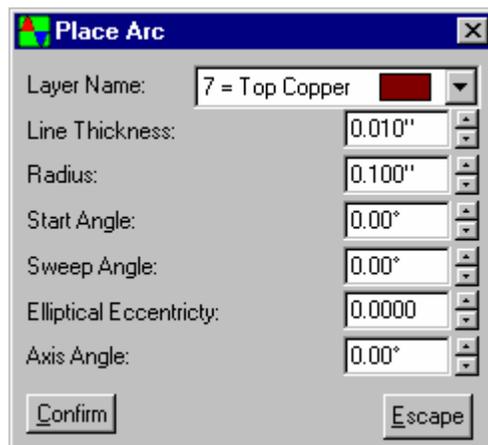
Pressing **right mouse** or **Escape** at this point interrupts the process and allows you to select another vertex. Pressing **left mouse** or **Enter** nails down the new vertex.

You may change the target layer at any time by using the + and - keys of the ten-key pad on the keyboard. A via will be placed automatically when changing from one copper layer to another.

## Place Arc/circle/ellipse

This function is not normally used for placing tracks, but rather for the placement of component outlines on silk screen layers, and occasionally for creating new components to add to the library.

Clicking on **Arc/circle** invokes a dialog box similar to that of *Line*, but requiring slightly more information.



To change the target layer for the arc, simply click on the layer name field. A dialog box will appear listing all the available layers with the current layer hilited. Select the new layer by simply clicking on your choice.

**Line thickness** specifies the width of the "pen" that is used to "draw" the arc.

**Radius** specifies the maximum distance from the center of the arc to the center of the "pen".

**Start angle** specifies the point where the "pen" is engaged during the "draw" operation.

**Sweep angle** specifies the distance through which the pen is to be active during the "draw" operation. A negative value will draw the arc in a clockwise direction.

**Elliptical eccentricity** specifies the ratio of the minor radius to the major radius. A value of one yields a circular arc.

**Rotation angle** specifies the position of the zero point of the arc with respect to the background coordinate system.

**Axis Angle** specifies the position of the zero point of the arc with respect to the background coordinate system.

Clicking on **Confirm** then places a ghost of the arc under the cursor, so you can position it precisely. Pressing **left mouse** or **Enter** completes the function.

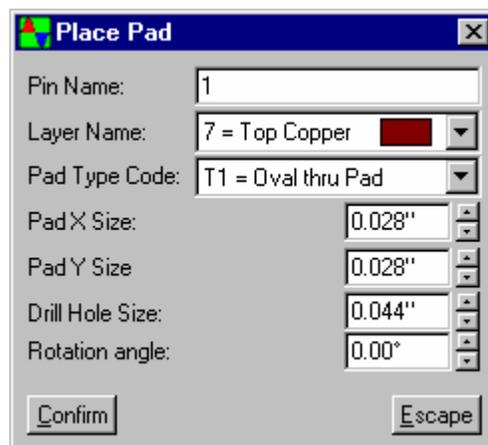
At this point, pressing **right mouse** or **Escape** will exit from the place arc function. Once the arc is placed, pressing **left mouse** or **Enter** will initiate the **Place Arc/circle** function all over again. This way, several identical arcs can be placed in rapid succession.

## Place Arc graphically

This function is used to place an arc on to the drawing using a graphical user interface rather than the numeric interface described in **Place Arc/circle/ellipse**. When this function is invoked CIRCAD'98 will prompt to Select a center point. This will be the center point of the circle that the arc will be created from. The next prompt will ask to Select radius/start point. In this step 2 parameters are defined at once. One, the radius of the circle that the arc is to be created from, and two, the starting point of one of the end points of the arc. The final prompt will be to Select end point. The arc that is constructed will be from the start point of the arc to the intersection of the arc and an imaginary line from the center of the source circle and the crosshair cursor.

## Place Pad

CIRCAD'98 can place either surface or through pads. Clicking on **Pad** produces another dialog box asking for **Place Pad Parameters**.



**Pin name:** The first item on the menu is pin name or number. This is only necessary if you are creating a new component -- if you are placing a free pad or mounting hole, no pin number is needed (see **Chapter 11** for more information on creating new components).

**Layer name:** Click on the layer name displayed, and a drop down list will appear with the available layers. The currently selected layer is highlighted; to select a different layer, simply click on it.

If the pad is to be a through pad, you will need to place it on the Pad Master layer (connects to all layers). If the pad is to be a surface pad, you must select one of the outermost copper layers on which to place it.

**Pad type code:** This selection allows you to specify the type of pad you want placed. Click on the code indicated and a dialog box will appear listing the types of pads available:

- T0 Schematic "pin" connection.
- T1 Oval or round through-pad.
- T2 Rectangular through-pad (often used on components to indicate pin number 1).
- T3 Round through-via.
- S0 Padstack special.
- S1 Oval surface pad.
- S2 Rectangular surface pad.
- S3 Schematic "dot" connection.

The current pad type is hilited. You can change pad type simply by clicking on another type.

**X size, Y size:** These selections are used to indicate the x and y dimensions. For Round pads they can be used to make the pad oval.

Clicking on **Confirm** makes the dialog box disappear, and the prompt asks you to **Position cursor for pad placement** (note that no ghost pad appears under the cursor).

Position the cursor where you want to place the pad and press **left mouse** or **Enter**, and the pad will be placed.

In practice, components from the library already include their appropriately spaced pads, so free pad placement is not common. Placing vias, however, is very common.

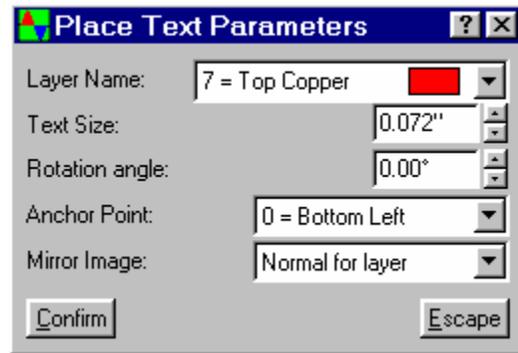
### **Place Via**

While the size of pads necessarily varies according to the size of the pins they are designed to accept, most vias in a given file will likely be the same size. Hence, CIRCAD'98 does not display a parameter selection menu when **Place Via** is selected -- instead, CIRCAD'98 immediately prompts you to **Position the cursor for via placement**.

This makes via placement very fast and easy. Changing the parameters of vias can be easily done through the *Setup Via Params* command (see **Chapter 4**).

Note that CIRCAD'98 features automatic via placement whenever you are in the *Place Line* or *Place Vertex* mode. All you need do is change layers while in the process of laying track by using the + or - keys the 10-key pad, and a via will automatically be placed as CIRCAD'98 switches layers. Further details on this feature can be found in **Chapter 13** under *Automatic Via Placement*.

## Place Text



Clicking on **Text** creates a parameter menu that asks for layer, size, and rotation of the text string to be placed.

**Layer name** specifies the target layer for the text.

**Text size** specifies the height of the font to be used.

**Rotation** specifies the angle counter-clockwise from normal.

**Anchor Point** specifies the point that the text is to be "grabbed" from. Here is a list of the 16 anchor points

0=Bottom left	4=Center Left	8=Top Left	12=Base Left
1=Bottom Center	5=Center	9=Top Center	13=Base Center
2=Bottom Right	6=Center Right Left	10=Top Right	14=Base Right
3=Bottom Offset	7=Center Offset	11=Top Offset	15=Base Offset

Select and confirm the desired parameters, and the dialog box will disappear. Click on the desired starting point for the text string, and a dialog box titled **Place Text Data** will appear. Now type in the characters desired. If the characters will be on more than one line, you must complete the first string by pressing **<Enter>** and reposition the cursor to begin the next string.

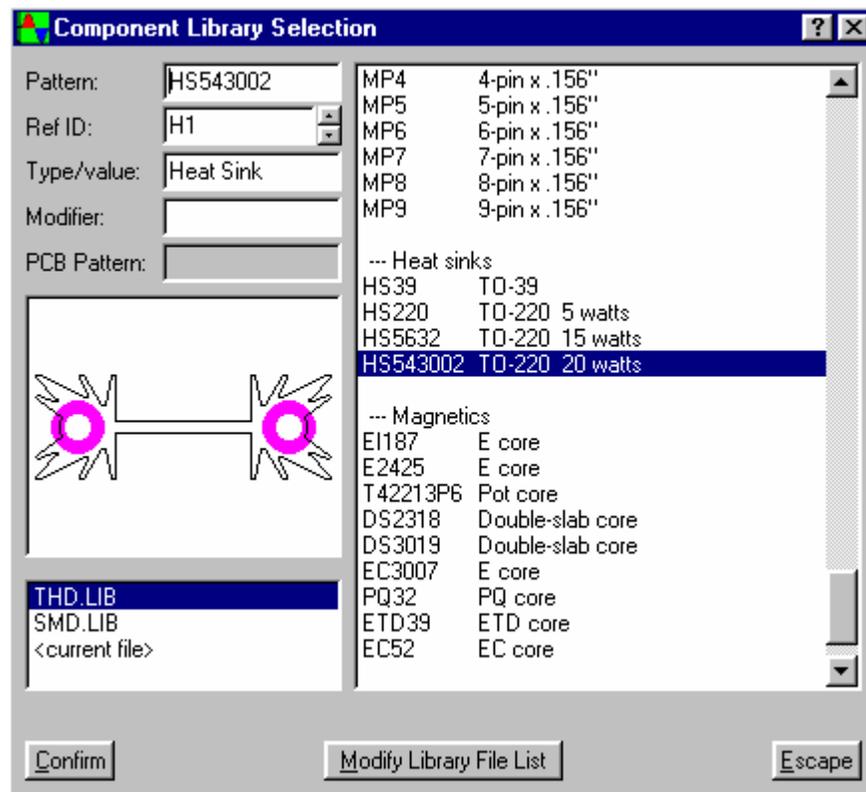
Note that if the text is being placed on the bottom side, it is shown in mirror image. This is because CIRCAD'98 never "turns a circuit board over" -- all layers are shown as if they were seen from the top side, "through" the other layers.

The text can be changed to show normally or mirrored on any layer using the *Setup Layers* function -- see **Chapter 4**.

## Place component

This function will be used frequently, both in schematic files and PCB files, and so warrants extensive discussion. The *Place component* command allows you to specify the type of part you want to place, or to select a part from one of CIRCAD'98's component libraries. You also have the option of creating your own supplementary libraries and components (see **Chapter 11** for details). These user-defined libraries can also be accessed through this command.

In order to understand how CIRCAD'98 handles component data, a few words on components in general are in order.



Schematic components and PCB components differ somewhat in their critical characteristics: A schematic component is defined primarily by its logical (or functional) qualities, while the critical parameters of a PCB component must also include its physical characteristics, or its "package". Thus, CIRCAD'98's schematic libraries are entirely different from its PCB libraries. CIRCAD'98 will automatically access the correct set of libraries, depending on whether you are working in a schematic file or a PCB file.

CIRCAD'98 allows you to avoid the manual picking process by specifying the PCB package at the schematic level. The netlist file, which you will eventually read into your new PCB file, will include the proper parts list. CIRCAD'98 will even select the correct parts and line them up at the bottom of the new PCB file, ready for placement. This feature can save a lot of repetitive keystrokes and reduce data entry errors.

With all that said, the procedure for placing components should be easier to grasp. If you are working in your schematic file and wish to place a part, click on **Place component**. You will then see a dialog box with five parameters for the part to be selected, as well as lists of the currently available libraries and the parts in the selected library. An explanation of these parameters follows.

**Pattern** is best understood as the name assigned to each component in the libraries. In a schematic library, pattern refers to a component's logical characteristics; in the schematic libraries, the patterns match the names in the applicable trade reference book for that library. In a PCB library, pattern refers to the component's physical package; in the PCB libraries, the patterns assigned follow a familiar descriptive scheme for component footprints.

**Reference ID** is the unique number assigned to each component in any single schematic or PCB, and is necessary for high level functions such as ratnesting, and design rules checking. CIRCAD'98 will automatically assign a unique sequential reference ID to each component you place. In some cases, you may wish to assign your own reference ID, and this is where you would key it in.

A note on Reference ID's assigned to multi-section chips on schematics (such as AND gates and flip-flops): CIRCAD'98 will keep track of which chips already on the schematic have open sections available, and will assign subsequent components to those open sections using suffixes 'b', 'c', 'd', and so on. If no open sections are available, the part will be assigned a new Ref ID with the suffix 'a'.

**Type/value** specifies the specific component type (i.e., 74LS121) or value (i.e., 4.7 $\mu$ F).

**Modifier** is not needed in all cases, but may be used to specify a tolerance or voltage rating.

**PCB Pattern** applies only when placing components in schematic files. This parameter is required in order to use the auto component selection feature. This is where you may specify the "footprint" the schematic component will take when the netlist is read into the raw PCB file. This field can use wildcard characters to search for desired pattern names. Simply enter some portion of the name followed by an asterisk (SM\* for example) and then press enter. CIRCAD'98 will search all of the current library files and display the results in the listbox on the right. The wildcard parameters are as follows:

? Represents any single character in this position.

\* Represents any string of 0 or more characters until the next search argument is reached.

If you do not want to enter the PCB pattern, or are not yet sure what it will be (or the pattern you desire does not yet exist in a PCB component library and must be created), you can leave it blank for now and return later to add it using the "Edit component" command. Or you could leave the parameter out altogether; CIRCAD'98 would simply not include this part in its automatic component placement routine, and you would then need to pick the part manually at the PCB design stage.

When placing components, you need not fill in every parameter; CIRCAD'98 needs only the pattern, and will automatically assign a sequential reference ID. You can always return later and use the *Edit component* command to modify any parameter. Simply hit enter on the empty parameter fields, and CIRCAD'98 will leave them blank.

When you have clicked **Confirm**, the box will be cleared from the screen, and you will see the "ghost" of the selected component attached to the cursor. You are now ready to place the component.

At this point, you can rotate the component in increments of 90 degrees before placing it. This is accomplished by pressing <R> on the main keyboard. Note: It is a rare case indeed when you would want to rotate a component at odd angles; however, this function is available simply by pressing <A> on the main keyboard. A dialog box will appear for the entry of the desired rotation angle.

When the desired rotation and position is achieved, press <Enter> or left mouse to nail the

component down. Once the component is placed, the *Grab component* function is always available for adjustments.

Just for confirmation, you may now want to select the *Edit component* command from the main menu and click on the component you just placed; you will again see the component parameter menu, listing the parameters of the component you just specified.

**Library Browse** mode. In many cases, you will not know the pattern for the component you wish to place, and would prefer instead to browse the various component libraries.

This is easily accomplished: When you first select *Place component* command select one of the available libraries. You will see displayed a listing of its components by pattern name, along with a brief description. The components are grouped by function and listed in alphanumeric order: All capacitors are together, all "AND" gates are together, and so on. As you scroll through the list a preview of the hilited component will be displayed on the left.

You will notice that any applicable parameters that are specified for that component are also indicated, and that a sequential reference number has been assigned. At this point, you can edit these parameters or fill in any blank parameters as required.

When you have clicked **Confirm**, the box will be cleared from the screen, and you will see the "ghost" of the selected component attached to the cursor. You are now ready to place the component by clicking the left mouse button.

If you are in a schematic file at the time, only schematic libraries will be listed for selection. Similarly, only the PCB libraries will be displayed if you are working in a PCB file.

For information on creating custom libraries, see **Chapter 11**.

**Browsing user-defined libraries.** **Chapter 11** details the differences between standard libraries and user-defined libraries, and explains how to create such libraries. Properly set up, these user-defined image libraries can be accessed using the library selector guide as described above, just like CIRCAD'98's standard component libraries.

You can add your own component patterns to the library file to suit your needs, or you can even create a number of separate library files. See **Chapter 11** for more information.

### **Place Filled Solid**

The *Place Filled Solid* function is used to construct a region which will be completely filled without voids. The region can be of any arbitrary shape, and can be placed on any target layer. It may be used to create large areas of solid copper or large cleared areas in the solder masks.

When you select this function, you will be prompted to enter a sequence of vertex points. As you enter these points, a set of "ghost" lines will show your progress. After selecting a minimum of three data points, the function will be completed by pressing <right mouse> or <Escape>.

### **Place Signal Port**

#### **Place Bus Port**

#### **Place + Pwr Port**

#### **Place -- Pwr Port**

These items are available for placement only in schematic files, and are used to create a name tag for a signal wire. A simple dialog box will prompt you for the signal name, and then the port is ready to be positioned and placed with a click of the left mouse button.

### **Place Signal GND**

### **Place Earth GND**

### **Place Named GND**

These items are available for placement only in schematic files, and are used to create a name tag for a signal wire. For a named ground port, a simple dialog box will prompt you for the signal name, and then the port is ready to be positioned and placed with a click of the left mouse button.

### **Place Signal Name**

### **Place Signal Dot**

### **Place Signal Wire**

### **Place Bus Wire**

These items are also available for placement only in schematic files, and are used to place text, vias, wires, and bus wires. Where applicable, a dialog box will prompt you for a signal name before placing the item.

### **Place Arrowhead**

The *Place Arrowhead* function is used to construct an arrowhead onto the currently assigned layer. This is useful when placing dimensions onto your board or drawing files.

### **Place XY Dimension**

The *Place XY* dimension function places a special purpose line that is designed for dimensioning areas of drawing. This function automatic activates the ortho mode while placing the line, however this can be released by pressing the <T> key. Once the dimension line has been placed it can be grabbed and stretched from either end using the Grab Vertex function. The dimension line can also be grabbed in its entirety by using the Grab Component function.

# Grab

Grab is the function you'll use to "pick up" a previously placed element and adjust its position.

A note on Ortho Snap mode. In some cases, you will want to adjust the position of an item straight up or down without changing the x coordinate; similarly, you may occasionally want to adjust an item's position straight left or right without changing the y coordinate. CIRCAD'98's Ortho Snap mode makes this operation simple -- this feature is explained fully in **Chapter 13** under Ortho Snap mode.

## Grab Line

This function is used to move a line segment intact, without changing its length or rotation. When you click on the line to be grabbed, it will become a ghost image, along with any other lines it may be attached to. Moving the cursor moves the line, "stretching" its attached lines along with it. Pressing <left mouse> or <Enter> places the line in its new position, along with any attached lines.

## Grab Vertex

This function is used to move the end-point of a line or the junction of two or more lines, stretching and rotating the lines in the process. When you click on the vertex to be grabbed, it becomes a ghost image, along with the lines it is attached to. Moving the cursor moves the vertex, "stretching" its attached lines along with it. Pressing <left mouse> or <Enter> places the vertex in its new position, along with any attached lines.

## Grab Arc edge

## Grab Arc end point

## Grab Arc origin

These functions are used to move or edit an arc, a circle, or an ellipse. When you click on the arc to be grabbed, it becomes a ghost image; moving the cursor moves the arc. Pressing <left mouse> or <Enter> places the arc in its new position.

## Grab Pad/via

This function is used to move a single via or a free-pad (one not contained within a component). Note that only free pads can be grabbed; those which are part of components cannot be grabbed separately. When you click on the pad/via to be grabbed, it becomes a ghost image; moving the cursor moves the pad. Position the cursor in the desired location and press <left mouse> or <Enter> to place the pad/via.

## Grab Text

This function is used to move or rotate a single text string. Note that only one text string can be grabbed at a time. When you click on the text string to be grabbed, it becomes a ghost image; moving the cursor moves the text string. Pressing <left mouse> or <Enter> places the text string in its new position.

## Grab component

This function is used to move or rotate a single component. When you click on the component to be grabbed, it becomes a ghost image, along with the lines it is attached to. Moving the cursor moves the component, "rubber-banding" its attached lines behind it. Pressing <left mouse> or <Enter> places the component in its new position, along with any attached lines.

## Grab Comp Only

This is similar to *Grab component*, except that attached lines do not "stretch" as the component is moved.

When you click on the component to be grabbed, it becomes a ghost image; moving the cursor moves the component. Pressing <left mouse> or <Enter> places the component in its new position.

### **Grab Filled Solid**

This function is used to move a solid region intact, without changing its shape or size. When you click on the solid to be grabbed, it will become a ghost image. Moving the cursor moves the entire solid area. Pressing <left mouse> or <Enter> places the solid area in its new position.

### **Grab Item**

This is a very powerful but difficult to use command that will allow you to grab any of the primitive elements from which CIRCAD'98 is composed. For example, components are made up of lines, pads and arcs, but these cannot be grabbed separately from the component using the *Grab Line*, *Grab Pad* or *Grab Arc* commands. The *Grab Item* command, however, will allow you to "reach inside" a component to grab one of these constituent parts. This command is not frequently needed, but is a valuable feature nonetheless. *Grab Item* is so powerful that you must be explicit about what you are trying to grab in order to avoid grabbing the wrong item.

You can grab only items that are in the same layer you are working on; if the item to be grabbed is on another layer, you must change layers before you can grab it (to do this, see the *Zoom Layer* command in **Chapter 2**). You also must grab the item at the right point: arcs and pads can be grabbed by their edge; line endpoints must be grabbed from their exact endpoint, and line segments can be grabbed from any point along their center line. When you click on the item to be grabbed, it becomes a ghost image; moving with the cursor . Pressing <left mouse> or <Enter> places the item in its new position.

### **Rotate 90°**

This command enables you to rotate any item by 90 degrees. First use one of the *Grab* commands to select the desired item, then type <GR>; the item will rotate one quarter-turn counterclockwise and the menu will disappear, leaving the "ghost" of the item attached to the cursor and ready to complete the *Grab* command. For rotations of greater than 90 degrees, you may type <GR> as many times as necessary.

### **Rotate N°**

This command enables you to rotate any item by any angular amount. First use one of the *Grab* commands to select the desired item, then type <GN>; a dialog box will appear for you to specify the desired rotational value. When you press <Enter>, the dialog box will disappear and the item will be rotated as directed, leaving the "ghost" of the item attached to the cursor and ready to complete the *Grab* command.

### **Mirror X**

### **Mirror Y**

These commands are similar in operation to *Rotate 90°*, in that you first use one of the *Grab* commands to select the item to be mirrored. The item will be "flipped over" along the horizontal or vertical axis as requested.

### **Mirror T/B**

Mirror T/B is an abbreviation for 'Mirror Top-to-bottom'. This command is similar in operation to *Mirror X*, with the additional feature that all SMT pads and silk screen data are flipped to the opposite surface of the circuit board. It is used in PCB files only, and is specifically designed for instances when you need to place a surface mount component on the bottom side of the board.

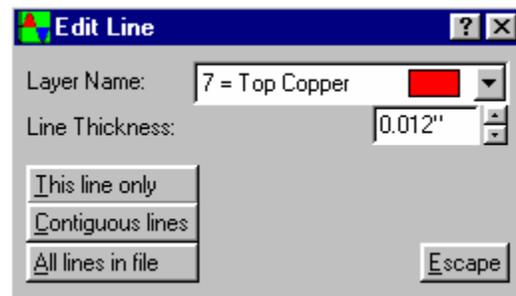
## **Scale XY**

These commands enable you to magnify or de-magnify any item. They are similar in operation to rotate and mirror commands, in that you first use one of the *Grab* commands to select the desired item, then type <GS> or <GZ>. A dialog box will appear for you to specify the desired magnification factor.

## Edit

The Edit functions are used to modify the parameters of items already on the board. This gives you the utmost flexibility in fine-tuning your circuit boards as you build them. Clicking on **Edit** produces another pop down menu of items you can edit.

### Edit Line



Click on the line you've selected and a dialog box of line parameters will appear (similar to the one displayed in the *Place Line* mode), indicating the current thickness of the line and the layer it is on. Change the layer as desired by clicking on the color brick and selecting a new layer color from the available. Change the line thickness as desired by clicking on the current value indicated and keying in the new value.

You also have a number of confirmation options. You can confirm the changes for **This line only**, or for **Contiguous lines**, or for **All lines in file**. The last two options will confirm the changes ONLY for all lines in the file that have the same thickness and layer parameters. This feature is a real time saver. If the line being edited is contained within a block or is part of a hilite, the block or hilite will act as an additional layer of restriction.

### Edit Arc/circle

Click on the arc/circle you've selected and a dialog box of Arc/circle parameters will appear (similar to the one displayed in the *Place Arc/circle* mode), indicating the current parameters of the selected arc.

You can now modify any of the parameters listed (for a complete explanation of these parameters, see the earlier section in this chapter on *Place Arc/circle*). When the parameters have been modified as necessary, click on **Confirm** to make the menus disappear; the arc will be redrawn immediately using the new parameters.

### Edit Pad/via

Click on the pad or via you've selected and a dialog box of pad parameters will appear, indicating the current parameters of the selected pad or via. The list of parameters closely resembles the one displayed in the *Place pad* mode.

**Signal name:** This is the net number or name assigned in the netlist. If the file does not yet contain a netlist, this space will be blank.

**Pin name:** This is the pin number or name assigned to this pin within the component. This parameter can be edited from this menu, but the change will only apply to the component(s) already within the current data file; the original library component patterns will not be affected. If you want to modify the original component as well, use the *Library View* command to edit the library.

**Layer name:** The name and color of the layer the pad is presently assigned to is displayed. To change the layer, simply click on the layer name to pop up a **Select layer** menu. The current layer is hilited; just click on the new layer desired and the menu will disappear. Your new layer will now be displayed on the pad parameters menu.

**Pad type code:** This indicates the type of pad, and allows you to change the pad type if needed. Click on the code indicated and a dialog box will pop up, listing the types of pads available.

- T0 Schematic "pin" connection.
- T1 Oval or round through-pad.
- T2 Rectangular through-pad (often used on components to indicate pin number 1).
- T3 Round through-via.
- S0 Padstack special.
- S1 Oval surface pad.
- S2 Rectangular surface pad.
- S3 Schematic "dot" connection.

**X size, Y size:** They are used to indicate the x and y dimensions. For Round pads they can be used to make the pad oval

You also have the option of confirming the changes for **This pad only** (which modifies a single pad), **Entire component** (which changes only those pads in the component which are the same size and type as the one you are editing), or **All pads in file** (which changes only those pads in the file which are the same size and type as the one you are editing). If the pad being edited is contained within a block or is part of a hilite, the block or hilite will act as an additional layer of restriction.

### **Edit Text data**

Click on the text string of your choice, key in the new text and pressing left mouse or <Enter>. If the new text needs re-centering, you can do so easily using the *Grab Text* command.

### **Edit Text parms**

This allows you to change the layer placement, size or rotation of any text string you click on. Note that if you place text on a bottom layer, it will be displayed as a mirror image, as if you were looking at it through the circuit board.

### **Edit component**

Click on the component you've selected and a dialog box of component parameters will appear, indicating the current parameters of the selected component.

You can now click on any of these parameters and enter new ones. Note that if you actually need to change the component pattern, you must delete the existing component and use the *Place component* command to place a new one.

### **Locate text**

This function will allow the user to search the data file for a specific text string.

### **Locate component**

With this command, you can search for any given component. CIRCAD'98 will scan the file, find the component, and place the cursor on it. Clicking on **Locate Cmp** invokes the Locate component dialog box. Enter search parameters into any of the five fields and CIRCAD'98 will do the rest.

### **Krump component**

With this command, you can add items to a component, to remove items from a component, or to dissolve a component into its constituent parts. The item or component acted upon must be first be attached to the cursor by using one of the *grab* functions. The operation of this function is smoothest when invoked through the keyboard by typing <EK>.

### **Krump Filled Solid**

With this command, you can dissolve a solid filled area into a set of perimeter lines, or construct a solid filled area from a set of perimeter lines.

To dissolve a filled solid, you must first "grab" the desired solid and then enter the key-strokes <EF>. The filled solid will be reduced to a set of perimeter lines of the original region. The lines may then be modified by using any of the standard editing functions.

To construct a filled solid from a set of perimeter lines, you must first "grab" one of the perimeter lines and then enter the key-strokes <EF>. The file will be scanned for a set of lines to define the perimeter of a closed region. The entire group of lines will then be compiled into a solidly filled area.

### **Convert Line to Arc**

This function will convert the line that cursor is over into an arc. When the cursor intersects the endpoints of the line, the arc will be a half circle with a diameter of the distance between the endpoints. As the cursor is rolled away from the arc, the diameter of the circle that the arc is constructed from will increase.

### **Resequence Components**

Allows you to renumber the components with the following parameters:

**Component centroid point** resequences the components by the centroid point of the component. The centroid point is the center of the component as determined by its extents.

**Component insertion point** resequences the components by the insertion point of the component. The insertion point of a component is where the component is grabbed from.

**Horizontal rows** resequence the components in rows that run horizontally across the drawing.

**Vertical columns** resequence the components in columns that run up and down the drawing sheet.

**Lower left** resequence the components with the above parameter starting from the lower left corner of the drawing.

**Lower right** resequence the components with the above parameter starting from the lower right corner of the drawing

**Upper left** resequence the components with the above parameter starting from the upper left corner of the drawing

**Upper right** resequence the components with the above parameter starting from the upper right corner of the drawing

**Start value** this is the value that the program will use for the first component

**Step value** the amount that the program increment each item i.e. U1, U2, U3 or U10, U20, U30.

**Forward annotate to PCB/Back annotate to schematic** indicates whether the schematic or PCB file is to be updated with the new data.

**Multi-sheet schematic** indicated whether all of the sheets of a multi sheet schematic are to be updated. This option will be available only if resequencing is started from the first sheet of a multi-sheet schematic (.sh1).

**Sheet start-step** enable step value between sheets i.e. sheet 1 be 100's, sheet 2 be 200's and

so on.

**Start-step value** amount to add to each successive sheet.

### **Snap Item**

With this command, you can jump the cursor to the endpoint of a nearby line segment or the centerpoint of a nearby pad. The closest nearby item is chosen as the 'snap-to' point. In densely populated regions, it is useful to change to the desired target layer before invoking this function.

# Delete

The Delete commands allow you to delete anything you've already placed on the board.

**Important: CIRCAD'98 will remain in the Delete mode until you select another command**

Clicking on **Delete** produces another familiar pop down menu with the following selections:

## Delete Line

This command deletes single line segments only. Note that lines which are part of a component cannot be deleted.

## Delete Multi-line

Using a procedure similar to *Delete Line*, this command deletes any contiguous group of lines by clicking on just one of them, deleting any vias involved as well, and stopping only at components and pads.

## Delete Vertex

An vertex is defined as a junction, or "corner" in a track. This command deletes a selected vertex and redraws the affected line as though the vertex, or corner, was never there.

## Delete Arc/circle

Note that arcs which are part of a component cannot be deleted.

## Delete Pad/via

Note that pads which are part of a component cannot be deleted.

## Delete Text

Note that the entire text string will be deleted -- use the *Edit Text data* command to delete portions of text strings.

## Delete component

The next component you click on will disappear.

## Delete Filled Solid

The next filled solid you click on will disappear.

## Delete Iso-islands

This selection is used to delete any isolated islands of copper that may have been created by the flood fill or cross-hatch functions.

## Delete Grabbed Items

This command will delete any item you have already grabbed with one of the *Grab* commands.

## Delete Hilited Items

This command will delete all hilited lines, arcs, and pads that are not part of a component (see the *Net: Hilite* command in Chapter 7).

## Shortcut Keys

You already know that you can type the underlined letter (usually the first letter) of any menu command as a shortcut. In addition, there are a few other command shortcuts:

The + and - keys on the number keypad allow you to quickly select a new active layer by stepping up (+) or down (-) through the layers. This is a shortcut of the *Zoom Layer* command (see Chapter 13).

The <R> key will <rotate> any grabbed item 90° in the counterclockwise direction. This is a shortcut of the *Rotate 90°* command.

The <A> key will <rotate> any grabbed item by an arbitrary number of degrees. This is a shortcut of the *Rotate n°* command.

The <X> and <Y> keys will <mirror> a grabbed item in the x and y directions, respectively. This is a shortcut of the *Mirror X* and *Mirror Y* commands.

The <M> key will <mirror> a grabbed component from the top to the bottom or from the bottom to the top of the board. This is a shortcut of the *Mirror T/B* command.

The <K> key will <scale> any grabbed item. This is a shortcut of the *Scale* command.

Also see Chapter 13

## Chapter 4

# Setup Commands

With the *Place*, *Grab*, *Edit* and *Delete* commands, you learned to work with an existing file. In order to change the parameters of an existing file you will use the Setup commands.



The Setup commands tell CIRCAD'98 how to draw a given file on the screen. You'll use Setup to name board layers, assign them colors, turn layers on and off, and so on.

Note that the setup values you select for a given file are resident in that file. This means that when you save a file, it retains the assigned setup values when called up again. It also means that if you have several files open at once and change the draw mode (for example) on one, the draw mode on the other open files is not affected.

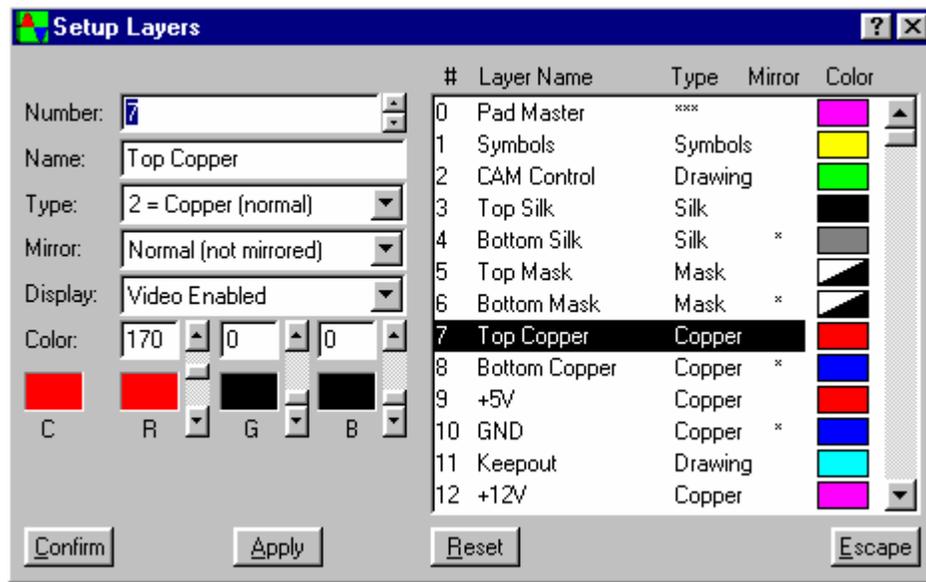
- **Setup Layers**
- **Setup Draw Modes**
- **Setup Grid Dots**
- **Setup Pixel Image**
- **Setup Library File**
- **Setup via Parameters**
- **Setup Clearances**
- **Setup Snap Table**
- **Setup Grid Snap**
- **Setup Zero Point**
- **Setup Mirror View**
- **Setup Ortho Mode**
- **Setup Keystroke Macros**

- **Setup Options**

# Setup Layers

The Setup Layers function is used to define the layer names, types, and colors for a given circuit board. This information is contained within the circuit board file, so each board can have its own separate setup parameters.

Clicking on **Layers** invokes the **Setup Layers** dialog box.



All 255 possible layers are displayed along with the current values assigned to the various parameters.

**Layer number:** This field indicates the numeric identifier assigned to each layer. This "layer number" can be modified by changing the current value.

**Layer name:** This field shows the name of each layer. When selecting a new name for a layer, it is good practice to use one that is self explanatory of the function of the layer. If a layer is not to be used (such as inner layers on a single sided board) the name can be deleted entirely.

**Layer type:** This field specifies the type of each layer. If you click on it a drop down will show the 8 types of layers available.

**Silk** is used for the top and bottom silk screen paintings.

**Mask** is used for the top and bottom solder resist maskings.

**Copper** is used for all signal layers on circuit boards.

**Plane** is used for power planes and ground plane.

**Symbol** is used for pictures not to be silk screened.

**Paste** is used for surface mount solder paste stencils.

**Schematic** is used for schematic signals on schematic files.

**Drawing** is used for mechanical or non-circuit drawings.

**Text mirror:** This field simply tells CIRCAD'98 whether you want the text to be automatically mirror imaged. This is mandatory on all under-side layers on a circuit board.

**Display:** This field tells CIRCAD'98 if it is to "draw" the currently selected layer. If the "Video Disabled" option has been selected the color brick for that layer will be a diagonally half filled brick indicating the color that will be used when "Video Enabled" option is selected

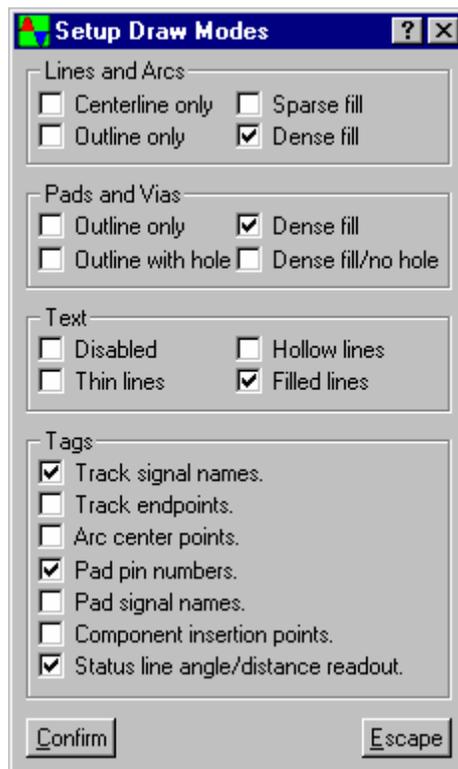
**Color:** This field defines the screen colors CIRCAD'98 will use to draw each layer, both in terms of color number and an actual color "brick". Use the up/down arrows or the scroll bar next to each field to increase or decrease the amount of each color that is to be added in order to achieve the desired final color.

You may select any of these colors by simply clicking on your choice, or you can turn a layer off by selecting color option zero

## Setup Draw Modes

Draw Modes refers to the mode in which CIRCAD'98 draws the file on the display screen.

Click on **Setup** then **Draw Modes** and a dialog box will appear listing the various draw mode options for lines, text, pads.



In the illustration of file FL1-C.PCB, the line draw mode is 3, denoting dense fill. Modes for pads and text are also shown (2 and 3, respectively). To change any draw mode, just put the cursor on the current mode and click on it.

## Setup Grid Dots

Grid dots are the white dots on the gray workspace that makes it look like a pegboard. They are useful as reference points, and can be placed at any interval you find helpful. Click on **Setup** then **Grid Dots** and you will see the **Set Grid Dot Spacing** dialog box, showing the current spacing (in mils). Entering a value of zero will remove the grid completely.

You can press <Escape> to keep the current spacing, or enter a new spacing value, or enter zero to have no grid dots at all. When you press <Enter>, the screen will immediately be redraws with the selected spacing.

### **Grid Dot color**

Color of the grid dots. To change the color click on the color brick and select a color from the pop up table.

### **Background Color**

Color of the "desktop" that the drawing is on. To change the color click on the color brick and select a color from the pop up table.

### **Hilight color**

Color to be used when hilighting an item. The hilight function can accessed by rolling the cursor over an item and pressing <ALT> H. To change the color click on the color brick and select a color from the pop up table.

### **Attention color**

Color to be used when CIRCAD'98 needs to inform the user of a possible error. An example of the attention color is when CIRCAD'98 is performing a Net Scan and encounters a clearance violation. To change the color click on the color brick and select a color from the pop up table.

## Setup Pixel Image

The [Setup Pixel Image](#) functions are used to control the aspects of an attached pixel image file. If there is no pixel image attached one can be loaded from here.

### **Load new image**

Loads a new image into the current file. CIRCAD'98 can have 2 images loaded at an given time.

### **Save Current Image**

Saves the currently active image to a disk file.

### **Release Current Image**

Releases the currently displayed image from memory.

### **Select Alternate Image**

Displays the second attached image if there is one.

### **Move the Image**

Moves the currently active image to a new location on the drawing sheet.

### **Scale/align the Image**

Allows the user the compress, expand and rotate in small amounts as necessary.

## Gamma

Change the contrast of the attached image. The Gamma function has four modes as follows:

**Normal:** CIRCAD'98 uses its SmartGamma™ logic to enhance the image

**Linear:** CIRCAD'98 will display each pixel at its original intensity.

**Invert:** Inverts the intensity of each pixel. Intensity 1 becomes 255, 2 becomes 254 and so forth. For color images this will invert the color of each pixel.

**Custom:** Allows a defined set of contrast intensities to be entered. Contrast intensities that are side of the defined range will be ignored.

## Change Resolution

Sets the dpi of the image to the value "Resolution (dpi)" field.

## Mirror X

Flips the image about the X axis.

## Mirror Y

Flips the image about the Y axis.

## Rotate 90°

Rotates the image counterclockwise by 90°

## Rotate 180°

Rotates the image counterclockwise by 180°

## Rotate 270°

Rotates the image counterclockwise by 270°

## Setup Library File

This selection is used to establish a working set of source library files to be used by the *Place component* functions. When invoked, a dialog box will appear listing the currently active library files. You may then add, change, or delete items from the list as needed. Use of this function is described in detail in **Chapter 11** under *Modifying the library guide* in Component creation (in detail).

## Setup Via Params

This function is used to establish the pad size and hole size to be used by the *Place Via* functions. When invoked, a dialog box will appear listing the default via parameters. For circuit boards, the layer number must be set to zero (Pad Master). For schematic sheets, the layer must be equal to the schematic signals layer.

To change the layer name, click on the field, and a drop down will display the available choices. Clicking on **Via size** or **Hole size** gives you the blinking data-entry cursor that allows you to key in new values. The up down arrows can also be used to edit the values.

Once these values are confirmed, the menu disappears; the next time you use the *Place Via* command, the new parameters will be in effect. Previously placed vias are not affected by new via parameters.

## Setup Clearances

This function is used to establish clearance zones around lines and pads. When invoked, a dialog box will appear listing the current clearance parameters. These clearances are used by the various automatic processing functions such as flood fill, cross-hatch, and DRC scan. When selected, this command will invoke a dialog box listing the assorted clearance parameters.

Section	Parameter	Value
Flood Fill	Flood Line Size	0.011"
	Flood to Line Clearance	0.009"
	Flood to Pad Clearance	0.013"
Net Scan	Netnest Line Size	0.011"
	Line to Line Clearance	0.008"
	Line to Pad Clearance	0.012"
	Pad to Pad Clearance	0.024"
Gerber Output	Mask Pad Swell	0.010"
	Paste Pad Shrink	0.005"
	Plane Pad Swell	0.005"
	Plane Hole Swell	0.030"
	Solid Fill Line Size	0.002"

**Flood line:** The *Block: Flood Fill* function uses adjacent line segments to fill in the plane sections of the circuit board. This value defines the thickness of the line segments that will be used, which in turn will affect how small an opening the flood fill will be able to penetrate. The cross-hatch and outline fill functions also use this same parameter.

**Flood to line:** This value defines the clearance allowed between the flood fill plane and any tracks on the board.

**Flood to pad:** This value defines the clearance allowed between the flood fill and any pads on the circuit board.

**Line to line:** Used by the net scan functions, this value defines the minimum allowable clearance between adjacent signal lines.

**Line to pad:** Used by the net scan functions, this value defines the minimum allowable clearance between signal lines and pads.

**Pad to pad:** Used in the net scan functions, this value defines the minimum allowable clearance between adjacent pads.

**Mask pad swell:** The Gerber output routine automatically swells the size of the pads on the solder mask file to guarantee non-overlap with the tinned pad area. This value defines the magnitude of the swell.

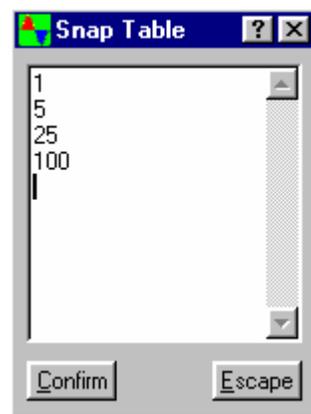
**Paste pad shrink:** The Gerber output routine automatically shrinks the size of the surface mount pads in the solder paste stencil file to guarantee confinement within the tinned pad area. This value defines the magnitude of the shrink.

**Plane pad swell:** On through pads which attach to and make connection with a power or ground plane, the Gerber output routine creates a special heat-relieved connection to prevent heat wicking away from the joint during the soldering operation. This connection is essentially an insulating (non-copper) ring around the pad, with four small radial webs to make the electrical connection. This value defines how much the pad should be enlarged before it is processed.

**Plane hole swell:** On through pads which pass through a ground or power plane without making a connection, the Gerber output routine places no pad but instead creates a clearance zone around the drill hole from the pad. This value defines the magnitude of the swell.

## Setup Snap Table

This function allows you to set grid snap settings. Click on a blank line and enter the new setting. Additional Setting can be entered by pressing <Enter> and entering another setting. This process can be repeated as many time as desired. If a setting is entered in millimeters i.e. 1.0mm, when that grid snap is active all of the readouts will be in millimeters.



## Setup Grid Snap

Grid snap is defined as the smallest spatial increment in which the cursor will move. For placing parts, a coarse grid is expedient, but for intricate work (such as grabbing and repositioning items) a fine grid is more precise.

Each time you select this function, the current grid snap is set to the next sequential element in the rotary grid snap table.

Clicking on any of the grid snap settings will set that as the current grid snap. This is useful to jump directly to a particular setting if you have a large list and don't want to use the <ALT G> to

scroll through the rotary grid snap table.

## Setup Zero Point

This function is used to set or clear the zero reference point that is used for the screen coordinate system. Details of this feature can be found in **Chapter 13**.

An audible report is produced when relative addressing is engaged. If relative addressing is already in effect, this function will restore absolute addressing.

## Setup Mirror View

This function may be used to flip the entire drawing sheet (board and all) over onto its back so as to be viewed from the opposite side. The shortcut for this function is <ALT V>.

## Setup Ortho Mode

This function is used to toggle the current "ortho-snap" mode. Ortho-snap mode is observed by the place line and place vertex functions, and by all of the grab functions. Details of this feature can be found in **Chapter 13**.

An audible report is produced when ortho-snap mode is engaged. An asterisk will be displayed after the grid-snap on the status line while ortho-snap mode is enabled.

## Setup Keystroke Macros

A keystroke macro may be used to assign an individual function or an entire set of functions to any of the user definable keys. The user definable keys are: F1 ~ F12, ALT-F1 ~ ALT-F12, CTRL-F1 ~ CTRL-F12, SHIFT-F1 ~ SHIFT-F12, and ALT-1 ~ ALT-0. If a call to another keystroke macro is encountered during execution of a keystroke macro, the current macro will be terminated, and execution will continue from the beginning of the new macro. A keystroke macro may be recorded from the keyboard by pressing ALT-\* (this is the \* on the ten key pad, not the \* above the 8 key). All keystrokes will be recorded until ALT-\* is pressed again, unless CIRCAD'98 loses control (such as the invocation of a pop-up dialog box). The recording process will continue once CIRCAD'98 regains control, however no intervening commands will be recorded. Once the keystroke macro is complete pressing the \* key will run the recorded macro.

A coefficient may be entered in front of a command to act as a repeat value, i.e. 4Up would issue 4 up arrows which would move the cursor vertically 4 grid snaps.

To gain direct access to any of the standard functions, enclose the command in <>. For example, <PL> would call the place line function, but it would not invoke its dialog box or set it as the active function.

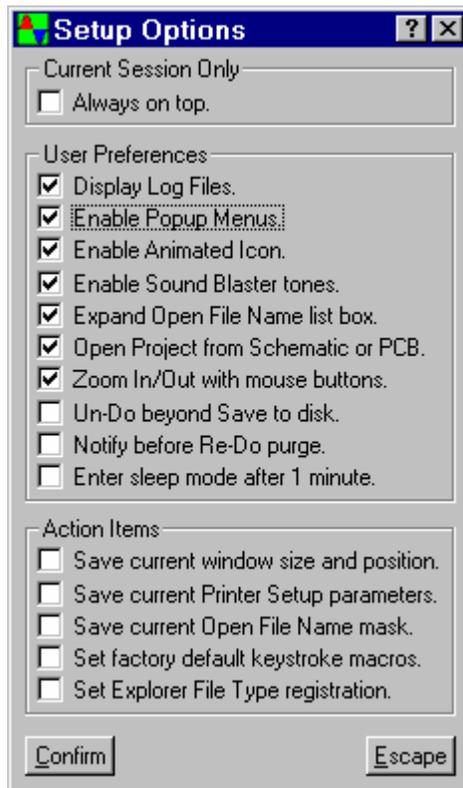
The following table lists the permissible entries for keystroke macros:

Normal	Shift	Ctl	Alt
Esc			#Esc
1	!	^1	#1
2	@	^2	#2
3	#	^3	#3
4	\$	^4	#4
5	%	^5	#5
6	^	^6	#6
7	&	^7	#7
8	*	^8	#8
9	(	^9	#9
0	)	^0	#0
-	_	^-	#-
=	+	^=	#=
BS		^BS	#BS
Tab	sTab	^Tab	#Tab
q	Q	^Q	#Q
w	W	^W	#W
e	E	^E	#E
r	R	^R	#R
t	T	^T	#T
y	Y	^Y	#Y
u	U	^U	#U
i	I	^I	#I
o	O	^O	#O
p	P	^P	#P
[	{	^[	#[
]	}	^]	#]
CR		LF	#CR
a	A	^A	#A
s	S	^S	#S
d	D	^D	#D
f	F	^F	#F
g	G	^G	#G
h	H	^H	#H
j	J	^J	#J
k	K	^K	#K
l	L	^L	#L
;	:		#;
'	"		#'
`	~		#`
\		^\	#\
z	Z	^Z	#Z
x	X	^X	#X
c	C	^C	#C
v	V	^V	#V
b	B	^B	#B
n	N	^N	#N
m	M	^M	#M
,	<		#,
.	>		#.
/	?		#/
PrSc	sPrSc	^PrSc	#PrSc

Space

F1	sF1	^F1	#F1
F2	sF2	^F2	#F2
F3	sF3	^F3	#F3
F4	sF4	^F4	#F4
F5	sF5	^F5	#F5
F6	sF6	^F6	#F6
F7	sF7	^F7	#F7
F8	sF8	^F8	#F8
F9	sF9	^F9	#F9
F10	sF10	^F10	#F10
Home	Num7	^Home	#Home
Up	Num8	^Up	#Up
PgUp	Num9	^PgUp	#PgUp
Num-	Num-	^Num-	#Num-
Left	Num4	^Left	#Left
sNum5	Num5	^Num5	#Num5
Right	Num6	^Right	#Right
Num+	Num+	^Num+	#Num+
End	Num1	^End	#End
Down	Num2	^Down	#Down
PgDn	Num3	^PgDn	#PgDn
Ins	Num0	^Ins	#Ins
Del	Num.	^Del	#Del
F11	sF11	^F11	#F11
F12	sF12	^F12	#F12

## Setup Options



The Setup Options dialog box is used to control various generic CIRCAD'98 parameters.

**Always on top** causes CIRCAD'98 to position itself so that no other applications will cover it.

**Display Log Files** invokes the View Window each time a .log file is created. An example of this is the error log generated by the Netlist In function.

**Enable Popup Menus** that occur when the mouse is clicked while no function is in effect.

**Enable Animated Icon** controls the lock or unlock status of the CIRCAD'98's scrolling icon. Some video configurations have poor processing of animated icons.

**Enable Sound Blaster Tones** sends the CIRCAD'98 sounds to the installed sound board rather than the PC speaker.

**Expand Open File Name list box** vertically stretches the files portion of the File Open dialog box and forces the details  button.

### **Open Project from Schematic or PCB**

Opens all files with the same name when opening either a .pcb or .shc file.

**Zoom In/Out with mouse buttons** allows CIRCAD'98 to zoom in on a left mouse click and out on a right click when there is no function in effect.

**Un-Do beyond save to disk** will not clear the Un-do buffer during a periodic save.

**Notify before Re-Do purge** displays a message before starting a function that will clear the Re-Do buffer.

**Enter sleep mode after 1 minute**

Releases the CIRCAD screen after 1 minute. This will prevent crashes on some computers when entering sleep mode

**Save current window size and position** stores CIRCAD'98's window parameters to be recalled upon the next entry to CIRCAD'98.

**Save current Printer Setup parameters** saves the current printer setup. These settings are unique to CIRCAD'98 and will not affect the printing of any other Windows application. They will be loaded each time CIRCAD'98 is invoked.

**Save current Open File Name mask** saves the currently selected File Open mask as the default.

**Set factory default keystroke macros**

This option will restore the default values for the keyboard macros. Here is a list of the default values:

F1 Place Line	Alt F1 Grab Line
F2 Place Arc	Alt F2 Grab Arc edge
F3 Place Pad	Alt F3 Grab Pad
F4 Place Text	Alt F4 Grab Text
F5 Place Ortho-line	Alt F5 Grab Item
F6 Place Component	Alt F6 Grab Component Only
F7 Place Via	Alt F7 Grab Component
F8 Place Vertex	Alt F8 Grab Vertex
F9 Zoom Re-draw	Alt 2 Fraction $\frac{1}{2}$
F10 Zoom Re-draw layer	Alt 4 Fraction $\frac{1}{4}$
Ctrl F1 Edit Line	Shift F1 Delete Line
Ctrl F2 Edit Arc	Shift F2 Delete Arc
Ctrl F3 Edit Pad	Shift F3 Delete Pad
Ctrl F4 Edit Text data	Shift F4 Delete Text
Ctrl F5 Edit Text parameters	Shift F5 Delete Multi-line
Ctrl F6 Edit Component	Shift F6 Delete Component
Ctrl F7 Locate Component	Shift F7 Delete Grabs
Ctrl F8 Krump Component	Shift F8 Delete Vertex
Ctrl F9 Snap to Item	

**Set Explorer File Type registration** allows the user to setup the file extensions that are to be associated with CIRCAD'98. Once a file extension has been associated with CIRCAD'98 un-associating them will not restore the original associations, it will simply remove them.

## Chapter 5

# File Commands

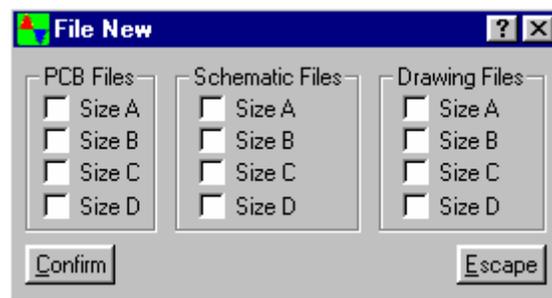
The File commands are used to create new files, manage existing ones, and generate various types of output files. Note that CIRCAD'98 can have any number of files open at once. They are held in memory in the order you called them up, and you can sequence through them using the *Next File* and *Prev File* commands.

- **New File**
- **Open File**
- **Re-Open File**
- **Total Recall**
- **Load File**
- **Close File**
- **Erase File**
- **Next File**
- **Previous File**
- **Select File**
- **Save File**
- **Save As ...**
- **File Information**
- **Gerber Out**
- **Plotter Out**
- **BOM Output**
- **Manual/Help**
- **Exit/Quit**

## New File

Click on **New File** and a submenu pops up asking you to select a "template" file. A template file is a time saver that enables you to copy certain commonly used elements into the new file, so you needn't do it manually each time. There are twelve supplied template files. These files contain the "blank forms" for the various sizes of drawing sheets. You will possibly want to customize these template files for added convenience. This procedure is detailed in **Chapter 12**.

When you've selected the desired template file, press Confirm to create the new file and copy in the template so you're ready to start designing. Next the "Save As..." dialog box will be displayed allowing you to select path and file name.



## Open File

This command is explained in detail in Opening a file in **Chapter 2**.

## Re-Open

The **Re-Open** menu contains individual files that have been opened with CIRCAD'98's File Open command. Files sent to CIRCAD'98 from Explorer will not be added to the list. Multiple files opened from CIRCAD'98's File Open command are not added to the list.

## Total Recall

The **Total Recall** dialog box holds the last 250 files that have been opened. This is similar to the Re-Open with two minor exceptions. The Re-Open menu will only hold 10 entries, and all files opened by CIRCAD'98 will be in the list.

## Load File

This command is similar to the ~~Open File~~, with the exception that a new position the rotary list of open files is not created. Instead the selected file is merged with the currently active file. Only gerber files and .bmp files can be loaded with this function.

## Close File

This command closes the file you are in at the time and then displays the next file in memory.

If the file has been modified during the current session, CIRCAD'98 will pop up a menu asking if you want to save the changes. If you elect not to save the changes, the file will remain as it was at the start of the session. If you do save the changes, an automatic backup file of the old version will be created as a safety measure.

## Erase File

This command closes the file you are in at the time removes it from disk, thus freeing its disk space for use by other files. A warning message will be issued with a mandatory confirmation request before the file erasure will occur.

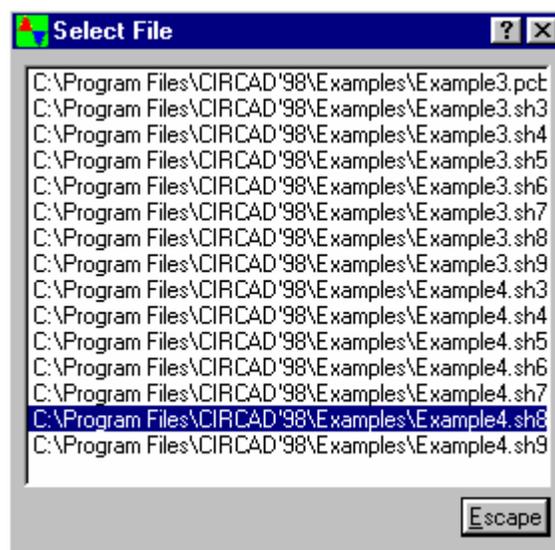
## Next File

When more than one file is open at once, this command is used to switch the display and editing system to the next sequential element in the set of active data files. The files are accessed in a rotary fashion in the order in which they were opened. The shortcut for this function is <ALT N>.

## Previous File

Similar to *Next File*, but moves backward through the series of open files.

## Select File



The Select File function allows you to jump directly to any open file without having to use **File Next** or <ALT N>.

## Save File

This command saves the current file to disk and creates a backup file for the old version. Instead of being closed, however, the file remains displayed on screen. Saving frequently as you work protects you from data loss in the event of power failure or Windows® system crashes.

The first time a file is saved, the file name extension on disk of the opened file is appended with .BAK for backup purposes.

## Save As ...

This command allows you to rename a file and/or store it in a different sub directory. CIRCAD'98 invoke the standard windows "Save As..." dialog box and ask you to enter the output file information.

Note that no files are deleted in this operation: If you save a file under a new name or sub directory, the old version remains on disk under the old name.

However, if you save a file under a name that already exists in the current sub directory, the old file will be overwritten. To prevent accidental data loss, CIRCAD'98 will display a warning message and ask you to confirm your choice of filename.

## File Information

Discussed in detail in **Chapter 2**, this command displays statistical information on the current data file.

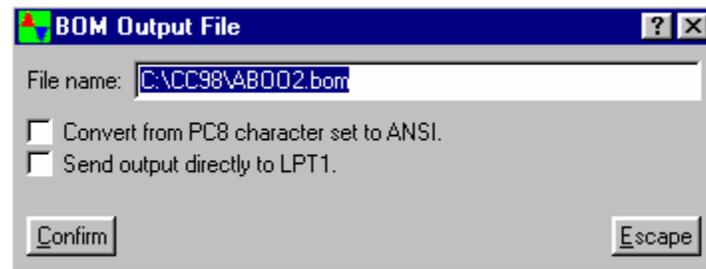
## Gerber Out

This command creates files that will drive a Gerber photoplotter. See **Chapter 10** for details.

## Plotter Out

This command creates files to drive a graphics printer. See **Chapter 9** for details.

## BOM



This command writes out a bill of materials and parts listing from the current data file to disk in the form of a printable ASCII data file. The above shown dialog box will appear for the entry of the desired destination file name and printing options.

## Manual/Help

This selection may be used to access the on-line reference manual in an interactive mode.

## Exit/Quit

Click on this selection to leave CIRCAD'98. Each file you have modified will be displayed in turn, offering you the opportunity to save the modifications before exiting the program.

## Chapter 6

# Block Commands

Block commands enable you to select, or block, a specified portion of the file and manipulate it in a variety of ways. The block commands act on any complete items within the block, such as line segments, endpoints, components, pads, etc. Any items that are only partially contained in the block will be ignored.

To explore the commands, click on **Block**, and the pop down menu will display a list of options.

- **Block Rectangle**
- **Block Polyline**
- **Block Release**
- **Block Move**
- **Block Copy**
- **Block Delete**
- **Block Delete Outside**
- **Block Rotation**
- **Block Scaling**
- **Block Layers**
- **Build component**
- **Nest component**
- **Block Flood Fill**
- **Track Isolation**
- **Isolation Output**
- **Mirror X**
- **Mirror Y**
- **Short-cut keystroke commands**

## Block Rectangle

This command defines the boundaries of a rectangular containment block. When you click on **Rectangle**, the status line will prompt you to **Select first corner point**; this is very similar to the procedure used in the *Zoom Window* command. Position the cursor where you want the first corner of the rectangular block to be and click to lock it down. The prompt tells you to **Select second corner point**, and when you've clicked on it, the block colors reverse out to show the defined block.

CIRCAD'98 then prompts you to **Select a source reference point**. This source reference point is the spot on the block where the cursor will be attached -- when you manipulate the block, this point is where CIRCAD'98 will "pick it up." For instance, if you later elect to rotate the block, CIRCAD'98 will rotate it around the source reference point you select.

You may want to zoom in on the block in order to place the reference point precisely -- in the very center of pin number one, for instance. You may do this by using the zoom command, as usual. You will notice, however, that the other commands on the main menu are crossed out, indicating that they are disabled while CIRCAD'98 waits for you to select the source reference point. When you have zoomed in, you can then place the reference point.

Note that the source reference point need not actually be located inside the block.

The status line will display the message **Block has been defined**. At this point, you can then use the remaining block commands to manipulate the block.

## Block Polyline

This command defines the boundaries of an irregularly shaped containment block. When you click on **Polyline**, the status line will prompt you to **Select first vertex point**; this is very similar to the procedure used in the *Place Filled Solid* command. Position the cursor where you want the first vertex point of the block to be and click to lock it down. The prompt then tells you to **Select second vertex point**. After you have selected at least three vertex points, pressing <right mouse> or <Escape> will complete the process and the block colors will reverse out to show the defined block.

CIRCAD'98 then prompts you to **Select a source reference point**. The remainder of the block definition is identical to that of a rectangular block explained above.

## Block Release

Clicking on release simply clears the defined block, returning the system to the "unblocked" state.

## Block Move

Clicking on move attaches a ghost of the block to the cursor in a similar fashion to the familiar *Place* command. You can then move the cursor to the new location, press <left mouse> or <Enter>, and place the block.

## Block Copy

Copy attaches a ghost of the block to the cursor, just like *Block Move*. However, you can place as many copies of the block as you like, just by pressing <left mouse> or <Enter> repeatedly. The original block is left in place; when the last copy has been placed, use the *Block Release* command to clear the defined block.

## Block Delete

The delete command delete any items (components, text strings, line segments, etc.) that are completely contained by the block, without disturbing other items not completely contained. If in doubt about what remains after a deletion, redraw the screen using the *Zoom Redraw* command.

## Block Delete Outside

The delete command delete any items (components, text strings, line segments, etc.) that are completely outside by the block, without disturbing other items that are completely contained. If in doubt about what remains after a deletion, redraw the screen using the *Zoom Redraw* command.

## Block Rotation

The rotation command allows you to rotate the block by any amount you specify (in degrees). Clicking on it produces a dialog box asking you to enter the block rotation (counter-clockwise).

## Block Scaling

With the scaling command, you can enlarge or reduce the block. A dialog box will prompt you to enter the magnification factor.

## Block Layers

Block commands act on all complete items within the block on all layers, unless instructed otherwise. If you want to exclude certain layers from a block command, click on **Layers** and a **Select Layers** dialog box will display the currently selected layers for the block. Simply click the layer to be included and leave out the ones to be excluded.

Note that if you define a new block, the table gets reset to include all layers.

## Build component

The Build Comp command is used to create new components, and is explained fully in **Chapter 11**.

## Nest component

The Nest Comp command is used to contain and label an entire circuit board for later panalization.

## Block Flood Fill

This selection is used to fill a region of a PCB with a set of line segments. This is useful for certain types of ground plane areas.

The line width and clearance zones around objects may be modified by using the **Setup Clearances** function. The current working layer is used as the target fill layer. Alternate source layers may be set by using the **Block Layers** function.

If you want the flood fill area to cover the entire board, an easy way to do this is to block an area larger than the board and then invoke Block Flood. To delete the area outside the board that gets filled, first highlight the unwanted section by using **Net Scan**, then delete it by using **Delete Hilight**.

When using Flood Fill, it is common for isolated pockets of copper to be created within the board that serve no electrical purpose. These are known as iso-islands, and can be easily deleted using the "Delete Iso-islands" feature, described in **Chapter 3**.

## Track Isolation

The Track Isolation function is used to construct a set of outlines around the tracks and pads on a PCB file, thus isolating them from one another, and from the remaining sea of copper on the blank board. Optionally, you may also construct a set of flood lines to clear out these areas of unwanted copper.

When you select this function, a pop-up dialog box will appear which will display the current

isolation parameters at which time you will be able to enter new values.

Once isolated and checked for correctness, the PCB data may then be converted to mill, drill, and route files by using the **Isolation Output** function.

A containment block must be defined in the current data file before this function can be used.

### **Isolation Output**

The Iso Output function is used to create a set of files to be used to drill, mill, and route a circuit board on a prototype board machine.

The file must first be processed by the **Track Isolation** function before this function can be used.

When you select this function, a pop-up dialog box will appear which will display the current output parameters at which time you will be able to enter new values.

A containment block must be defined in the current data file before this function can be used.

### **Mirror X**

The Mirror X command allows you to "flip over" in a horizontal direction all data within a defined block

### **Mirror Y**

This function is exactly the same as *Mirror X*, except that the block is flipped over in the vertical direction.

## **Short-cut keys (Chapter 6)**

### Short-cut keystroke commands

A few convenient keystroke short-cut commands are available when you are in the *Block Move* or *Block Copy* modes. Simply pressing <X> or <Y> will mirror the defined block along the x or y axis. Simply pressing <R> will rotate the block 90° counter-clockwise.

## Chapter 7

# Netlist Functions

A netlist is an ASCII file consisting of a complete list of every signal net or interconnection among all pins of a given circuit board or schematic. You needn't use a netlist to manually route a circuit board, but importing or creating a netlist gives you access to some very useful CIRCAD'98 features, such as:

- Ratnesting, to facilitate logical component placement and assist in manual routing.
  - Autorouting, to quickly route even the most complex boards on the fewest possible layers (DOS version only).
  - Design rule checks, to compare the routed board to the netlist and test for a variety of errors.
- 
- **Netlist Input**
  - **Netlist Out**
  - **Rat-nest**
  - **Rat-line**
  - **Un-ratnest**
  - **Hilite**
  - **Name Net**
  - **Image**
  - **Clear Hilite**
  - **Net Scan (Design Rules Check)**
  - **Compatible Netlist Formats**

# Netlist Input

The best way to explain how CIRCAD'98's netlist processing works is to demonstrate it using the supplied files LPS-C.pcb and LPS-C.net. LPS-C.pcb is a circuit board with components placed, but without tracks; LPS-C.net is the matching netlist. The reference ID number and pin numbers of every component on the board match up to a corresponding component in the netlist (if they did not, CIRCAD'98 would warn you of an error when the netlist is read in).

Start by opening the file LPS-C.pcb. Then click on **Nets** and **Netlist In**. CIRCAD'98 assumes you want the netlist file having the same filename as your open .raw file, with the suffix .net. CIRCAD'98 will read the netlist in to the .RAW file, highlighting the pins as it goes. If no discrepancies between the file and the netlist exist, CIRCAD'98 will turn off the highlighting as soon as the netlist has been completely read in.

If discrepancies exist, CIRCAD'98 will beep and display one of the following error messages.

- **Input line invalid** -- Indicates that CIRCAD'98 does not recognize the format of the .net file, and cannot process it.
- **Signal name invalid** -- The signal name in the netlist contains characters that cannot be processed.
- **Component name invalid** -- The component name in the netlist contains characters that cannot be processed.
- **Component not found** -- The component indicated in the netlist cannot be found on the board.
- **Pin not found** -- A component pin in the netlist cannot be found on the board. This condition may be caused by specifying an incorrect pattern for a given component.
- **Crosslinked pin** -- A component pin is included in more than one signal net. This condition may be caused by having more than one source Component with the same reference ID.
- **Pattern not found** -- A component pattern in the netlist cannot be found in any of the specified library files.
- **Pattern mismatch** -- The component pattern in the netlist does not match that which is already specified on the board.
- **Type/value mismatch** -- The component type/value parameter in the netlist does not match that which is already specified on the board.
- **Modifier mismatch** -- The component modifier parameter in the netlist does not match that which is already specified on the board.

If any errors are encountered while processing the netlists, an error log file will be created containing the entire set of errors. This file will have the same path and name as the target data file with a suffix of .log. Listing or printing this file will prove useful when "ferreting out" the errors from the source schematic file. For details, see **Chapter 13**.

## Netlist Out

This function is used to create a netlist file for transferring the essential information from a schematic to a printed circuit board. Un-named signals are automatically scanned and assigned a sequential signal number. Just click on **Nets** and **Netlist Out**.

## Rat-nest

The Rat-nest command will use the netlist information that has already been read in to the file to draw straight lines between all interconnected pins -- just click on **Nets** and **Rat-nest**.

This can help you place components in the locations that make the most sense; that is, where signals cross each other as little as possible and are as short as possible. Just grab and move each component in a trial-and-error fashion until you are satisfied that the layout makes sense.

Rat-nest is also indispensable for routing boards manually, giving you a ready guide as to which pins need tracks between them. After you have placed a number of tracks, just rat-nest again to get a fresh look at the routing that remains to be done.

## Rat-line

The Rat-line command is similar in operation the *Rat-nest* command, with the restriction that it performs its action on a single signal net only. To use this function, you must first position the cursor over one of the pads in the desired net.

## Un-ratnest

The Un-ratnest command will clear and remove all of the rat-lines from the current data file. The net names will not be affected.

## Hilite Net

It is often convenient to identify all connections made by a given net. For example, you might want to identify one net at a time to facilitate manual routing, or you may want to trace out a complex net for any number of other reasons. CIRCAD'98's *Hilite Net* feature "lights up" any net you select, whether or not the track has already been laid. Note -- no DRC functions are performed by this operation.

Just click on **Nets** and **Hilite Net**, place the cursor on any signal pad along the desired signal net, and click on it. All pads, lines, and arcs having the same signal net name as the selected pad will be hilited.

The net will remain hilited until you select another net to hilite; it will then release the hilite on the first net and light up the second one.

When you are finished with the hiliting feature, just click on a blank part of the board; that is, one that has not tracks or pads, or type <NC> for clear hilite. CIRCAD'98 will then release the hiliting on your most recently hilited net.

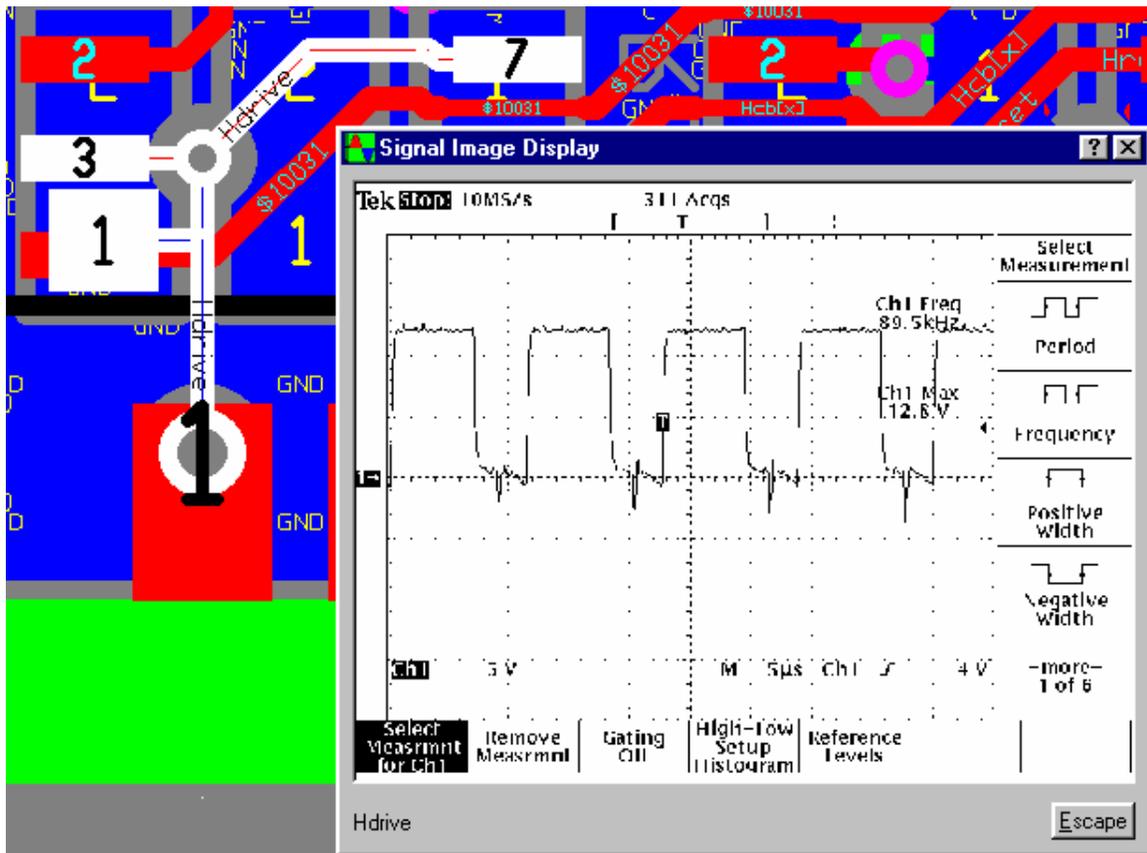
## Name Net

You may want to assign a particular name to a net, such as "+14V" or "Analog ground", replacing the name or number currently assigned in the netlist. To do this, first highlight the net by using one of the techniques described above; then click on **Nets** and **Name Net**. A dialog box will ask you to enter the signal name (the name currently assigned to that signal will be displayed). Just type in the new name and press <Enter>.

The original .NET file is not changed by this; only the netlist within the current data file reflects the name change. Of course, if you select *Netlist Out*, the new netlist will be written out with the new name.

## Image

The image function is used to display a .BMP file with the same name as the currently hilited signal track. The .BMP file that is to be associated must be be in the same directory as the current file.



## Clear Hilite

The Clear Hilite command will clear any existing hilite condition and return the hilited items to their normal display state.

When you are finished with the any of the hiliting functions, use this command to "un-hilite" the nets. Simply click on Netlist and Clear Net, and the board will return to normal.

## Net Scan (Design rules check)

At any time during or after you have completed manual routing, CIRCAD'98 can perform a design rule check, or net scan. In addition to comparing the routed board to the netlist, it will perform a scan of the board with respect to a number of design rules. For instance, it will warn you of tracks and vias placed closer together than pre-defined minimum limits, or of tracks that are connected to the wrong set of pins.

Click on **Nets**, and you will see the menu selections **Net Scan**, **Net Scan +**, and **Net Scan -**. The former scans a single net, and the latter two runs through the netlist in a sequential order.

The following error messages may be displayed, and are self-explanatory:

- Two or more signals are cross-linked (use the **Hilite Net** function to help find the error).
- Signal routing is not complete (the missing track will be rats-nested).
- Signal is linked to unknown pins.
- Spacing/Clearance violation (the offending items will be highlighted).

CIRCAD'98 will pause after each error.

## Compatible Netlist Formats

In addition to its own netlist format, CIRCAD'98 currently has the ability to import netlists that are in either the *CALAY* format or the *Tango/Autotrax* format, which can be generated by any number of schematic or netlist programs. If desired, you also can build a netlist file manually using a text editing program.

## Chapter 8

# From Schematic to PCB

The typical circuit starts out as an idea which gets jotted down on a dinner napkin at the cafeteria. Once back from lunch, the basic idea is soon converted into an approximate schematic on a real sheet of paper. After the circuit has won preliminary approval, it is necessary to transfer the information to the computer at which time the schematic must be refined and filled out to its final configuration. The computerized schematic can then be used to generate much of the data that is needed to build the finished printed circuit board.

CIRCAD'98 is designed to make this process as straightforward as possible by providing an unbroken pathway from schematic to printed circuit board to check-plot to final photo plotting and assembly drawings, all in a single integrated program.

The following tutorial will walk you through the process, from schematic to finished board design. CAM output functions (pen/laser plots, Gerber files, etc.) are covered in detail in their own chapters.

- **Schematics**
- **Netlists and Bill of Material**
- **Circuit Boards**

# Schematics

Because CIRCAD'98 is able to hold more than one file open at a time, you can start a new file at any time. Just invoke the main menu bar (press the <space bar> on the main keyboard) and click on **File**. As a shortcut, you could simply type <F> to make the main menu bar and the File menu appear simultaneously.

Select **New File** by clicking on it, and a dialog box will ask you to name your new file. Assigning schematic files a suffix of .SCH is a useful convention for easy identification; if your schematic will have more than one sheet, the suffixes .SH1, SH2, etc. should be used (note that you can use the **Save As** function to rename any file at a later date).

When you have typed in the file name and pressed <Enter>, another dialog box will ask you to select a template. Since CIRCAD'98 can work with many different file types and sizes, a large number of setup parameters are needed to establish the proper working environment for a given data set. The template system is provided as a mechanism for both saving time and reducing errors. The basic set of templates can be easily customized later when your familiarity with the program increases.

For now, simply select the template for the desired size of the schematic you plan to create, and you will see the empty drawing sheet appear at optimum magnification. Use the **Place Text**, **Edit Text** and **Delete Text** functions as necessary to enter the necessary data into the title block in the lower right-hand corner of the drawing sheet. Now you are ready to begin entering your schematic.

## Placing parts from the source libraries

You'll want to start by placing the required components. Select **Place** and then **component** from the main menu, and a dialog box will appear, asking you to enter the component parameters: **Pattern**, **Ref ID**, **Type/value**, and **Modifier**.

If you know the pattern name of the part you wish to place, you can type this in directly and press <Enter> to move on to the **Ref ID** field.

However, you may prefer to browse one of CIRCAD'98's source libraries to select the part you need. To do this, simply select the library that you wish to browse and all of the components in that library will be displayed.

When you have found the part you want, you may select it by pressing <Enter> or <left mouse>. CIRCAD'98 will insert the appropriate component parameters in all the data fields provided for in the library component. You may use the **Modifier** field to include additional information about the component as required.

Note that as you place each component, CIRCAD'98 automatically assigns a **Ref ID** to each one. CIRCAD'98 will continue to assign sequential **Ref IDs** as you place each component.

In the case of multi-section chips (such as AND gates and flip-flops), CIRCAD'98 will keep track of which chips already on the schematic have open sections available, and will assign subsequent components to those open sections using suffixes *b*, *c* and so on. If no open sections are available, CIRCAD'98 will assign the part a new number with the suffix *a*.

After you have entered the **Modifier** (if any), the dialog box will disappear and you will see the 'ghost' of the part attached to the cursor. When you have properly positioned the component, press <Enter> or <left mouse> to place it. Place the remainder of the parts in a similar fashion.

A special note on TTL and CMOS parts: CIRCAD'98 assigns power and ground automatically to these parts through special "hidden" pins at the top and bottom of each symbol. These power and ground pins carry the signal names VCC and GND for TTL parts, and VDD and VSS for CMOS parts. All VCC pins on a given schematic will automatically be placed into the signal net list. The same holds true for GND, VDD and VSS pins.

### **Placing parts from the image library**

No set of generic libraries could possibly include the huge variety of connectors and other special components that designers sometimes require. For this reason, a second type of library is available for 'storage' of special components that you yourself may create on an as-needed basis.

The image library is easy to access: When you invoke the **Place component** dialog box appears, select the desired library from the list of available libraries. The image library will be processed by the dialog box in a similar manner to the standard text libraries.

To create components and add them to the library, follow the procedure outlined in **Chapter 11**.

### **Placing signals**

CIRCAD'98 offers several different types of element for connecting the various components on the schematic. All of these are available under the **Place** command on the main menu bar.

**Signal Wire.** The simplest of all elements, signal wires are used to connect two or more component pins together. Signal wires which cross are not seen as electrically connected unless their junction is marked by a signal dot.

**Signal Dot.** Signal dots are used to join two signal wires together.

**Signal Name.** Used to attach a name to a signal wire.

**Bus Wire.** This is used to place a thick bus line, to which many signals can be attached through bus ports.

**Signal Port.** A signal port allows you to run a given signal to many different pins throughout the schematic, without actually drawing a trace to connect them all. When you select the command **Place Signal Port**, CIRCAD'98 asks you to give the signal a name. When you place the signal port, a tag appears with that signal name on it. You can place any number of signal ports with the same name; all signal ports to which you assign that same name are seen by CIRCAD'98 as electrically connected, i.e., as a single net.

**Bus Port.** These are the entry or exit points on a bus to which individual signals are attached. The identity of each signal is specified by a signal name parameter.

**+ Pwr Port, -- Pwr Port, GND Port.** Similar in operation to signal ports, these ports allow you to connect power and ground as required, without cluttering the schematic with traces.

The procedure for moving and deleting any of the signal ports is similar to the procedure used for components.

## **Netlists and Bills of Material**

If you want to end up with a good printed circuit board, you must start with a good schematic. You may print your schematic as often as necessary to make sure it is exactly what you want -- now is the time to iron out any snags, before you proceed to the final step.

The next step in the process is to use your finished schematic data file to generate a bill of materials (BOM) and a netlist. With a BOM, you can have a ready reference for placing components on the bare board. In fact, if you filled in the component parameter **PCB Pattern** for all schematic components, those component patterns will be read into the blank PCB file along with the netlist, and CIRCAD'98 will place the needed components at the bottom of the file, ready for final placement. With a netlist, you can "ratnest" all the nets on the PCB to optimize component placement and facilitate manual routing.

To generate a BOM, select **File** and **BOM Output** from the main menu. Since the file is a straight ASCII file, it can be called up, edited and printed by any standard line editing program.

Two sheets will be printed. One will list all components in the order of their **Ref ID**, indicating their position on the schematic, pattern, value, and modifier. The other sheet will list all components by part type/value, indicating the quantity for each part and a spread sheet of the **Ref IDs**.

To generate a netlist, select **Nets** and **Netlist Out** from the main menu. Again, the .NET file is a straight ASCII file which can be called up, edited and printed using any standard line editing program.

## Circuit Boards

Next, create a new PCB file. As with the schematic, select **File** and **New File** from the main menu bar. Name the new file with a .PCB suffix, select the proper template, and you're ready to create the board.

Use the **Place Line**, **Place Arc**, etc. commands to draw the outline of the board (on the pad master layer) and any other details such as dimensions or miscellaneous boilerplate text. Now you are ready to read in the netlist you generated at the schematic stage.

Select **Nets** and **Netlist In** from the main menu. If you took the time when placing your schematic components to enter the **PCB Pattern** parameter, then the netlist will include these patterns, and will use them to select each component and place it at the bottom of your new PCB file, ready for final positioning. Components on your schematic for which you did not enter a PCB Pattern will need to be selected from the libraries manually.

The error log is a great help in doing this. As CIRCAD'98 reads the netlist into the PCB, it generates a printable report of pins, components, etc. that could not be found. This report is written out to disk automatically under the same filename as the PCB with the suffix .LOG (covered in detail in Chapter 7 under *Netlist Input* and in **Chapter 13** under *Error Log*). Use the error log to pick the final necessary components from the libraries.

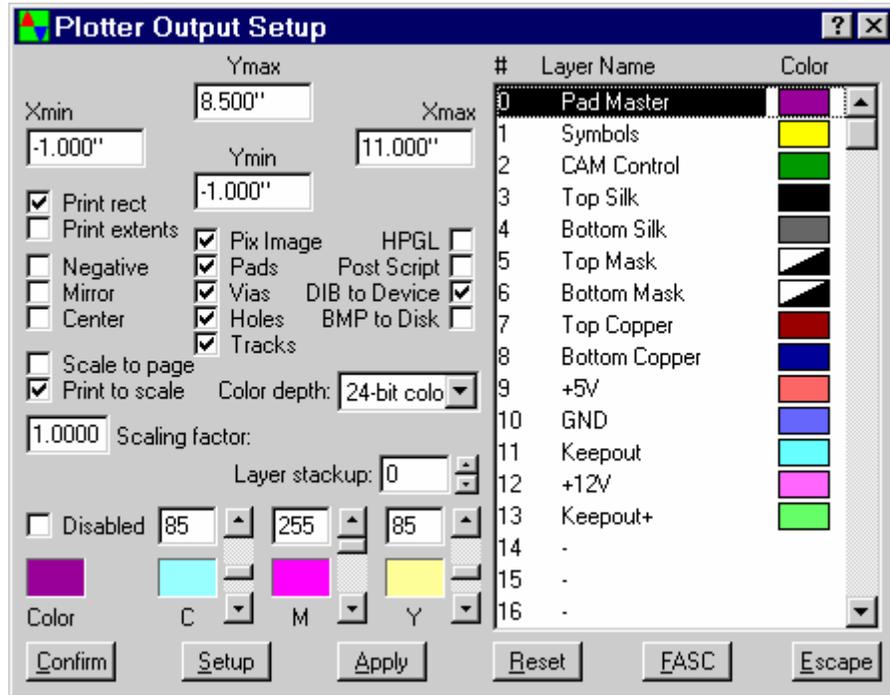
Now you are ready to move the components to their proper places on the circuit board. The first step in this process is to produce a ratnest of necessary interconnections. The ratnest consists of straight lines to temporarily connect all pins included in the net, and can give you a feel for placing the components so as to minimize congestion and signal crossing. Time spent now experimenting with different component locations position can translate into a much easier routing job at the next step.

Plotting the circuit board and generating Gerber output are covered in detail in **Chapter 9** and **Chapter 10**.

## Chapter 9

# Printing

The Plotter Output Setup dialog box is used to format CIRCAD'98's output to Windows®. The interface between Windows® and the printing device (i.e. print quality, number of copies, paper orientation.) can be accessed by the pressing the Setup button.



### Xmin, Ymin, Xmax, and Ymax

These values indicated the rectangle to be printed. If a block is defined, these values will be the corners of the block.

### Print Rect/Print Extents

These mutually exclusive fields indicate whether the entire file should be printed, or only the file rectangle defined by the Rectangle Values indicated above.

### Center

This check box will center the printed image on the page.

### Mirror

This check box will mirror the image when printed. This is particularly useful when printing one layer (usually a bottom layer) on transparency.

### Scale to page/Print to scale

These will compress/expand the file to best fit the size of the paper defined under the setup button. If the BMP to Disk option is selected, the scaling factor determines the DPI in increments of 300 of the BMP file created. A scaling factor of 1 will result in a 300 DPI BMP, 2 will yield a 600 DPI file and so on.

### BMP to Disk

Selecting this option will send the output to a specified disk file as if a Save As... has been

performed. The saved file will be in BMP format rather than as a CIRCAD'98 data file.

### **DIB to Device**

Selecting this option will send the data to the printer as a DIB(Device Independent Bitmap). Unfortunately not all printers support this data type, therefore it may be necessary to select an alternate output format (such as Post Script).

### **HPGL**

This selection will save the data to a file in the standardized HPGL (Hewlett Packard Graphics Language) format.

### **Post Script**

This selection will send the data to the printer in the standardized Post Script format. If the printer does not support Post Script files the result will be several pages of ASCII text.

### **Pixel Image**

This selection will print the attached pixel image along with the CIRCAD data file.

### **Pads/Vias/Holes**

Click on the desired check boxes to change the print status of an individual item. An unchecked item will not be printed.

### **Tracks**

This check box indicates that the copper tracks are to be printed in the plot.

### **Reset**

Restores the fields for the selected layer to the original settings. Once the settings for any layer have been changed and another has been clicked on the only way to restore all the values to the original settings is to Escape from the dialog box.

### **Setup**

The setup button will call the dialog box provided by your printer manufacturer for the interface between the printer and Windows®.

### **FASC**

FASC is an acronym for Fetch Actual Screen Colors. This will set the plot colors to the value assigned to the layer.

## Chapter 10

# Gerber Photo Plots

The **Gerber Out** function is used to create a set of data files on disk in **RS-274-D** format for subsequent plotting by a Gerber photo plotter.

When you select this function, a dialog box will appear for the specification of the desired output data files.

An aperture list (wheel) is automatically created by the program before the Gerber output files are written.

The relative zero point in effect at the time of output will be used for the Gerber output data files.

- **[Bridging the Communication Gap](#)**
- **[PCB Manufacturing: A Brief Overview](#)**
- **[Generating Output, Step by Step](#)**

## Bridging the Communication Gap

The convenience and speed of personal computer-based circuit board design allows you to easily revise and refine your boards until they are as near to perfect as you care to make them. Now you must communicate your computer files to the manufacturer who will produce the boards.

In an ideal world, the board would be produced directly from data files on your diskette; however, manufacturing equipment is extremely expensive, and it does not yet make economic sense for the industry to reconfigure that equipment to be computer-compatible. As a result, many circuit board manufacturers or "board houses" have at best a limited knowledge of computers. It is up to you, then, to produce output that matches the input requirements of the existing manufacturing technology.

Usually, there are three outside agencies involved in producing a circuit board:

1. The printing service bureau, which will generate film, or artwork, to be used by the board house in etching the board.
2. The board house, which will drill the raw board stock, plate it, use the film (from the printing service bureau) to etch it, and apply the solder mask and silk screen layers.
3. The stuffing house, which will install the parts according to your instructions.

Each agency requires proper instructions to perform its tasks; fortunately, CIRCAD'98 makes it easy for you to produce the appropriate combinations of data files, drawings, and lists for each stage of the manufacturing process.

The printing service bureau requires that you furnish data files to drive the Gerber photoplotter used to generate the artwork. A separate data file is used for each layer. In most cases, you must also provide a matching *aperture list* which will be used to correlate aperture number and aperture size as required by your board.

In most cases, you will also furnish the printing service bureau with a drill file. They will use this file to generate a paper tape that will in turn be used by the board house to drive the Excellon drill machine. A few board houses have the equipment to generate their own drill tape, and so you could give the drill file directly to them.

The board house will require the finished Gerber artwork, a drill tape (unless they can generate their own from your drill file), and a good, old-fashioned drill drawing. This will list the dimensions of the board, and the number, size and location of all holes.

The stuffing house will want a bill of materials, a stuffing guide listing the location of each part, and a placement drawing (a plot of the silk screen layer). If your board uses surface mount components, the stuffing house will also need Gerber artwork (produced by the printing service bureau from your files) to make a solder paste stencil.

CIRCAD'98 produces all of these; moreover, the drill drawings, silk screen plots, and so on will be of a quality that will earn you the goodwill of all who must rely on them. Smoother working relationships and a high-quality finished product will result.

# PCB Manufacturing: A Brief Overview

A brief description of the PCB manufacturing process may be a useful prelude to generating CIRCAD'98 output. For the sake of simplicity, the following outline applies to two-layer board production.

The process starts with the raw material: a large panel of fiberglass circuit board stock coated with a thin layer of copper on each side, from which a number of boards will be cut.

1. The board is drilled for all holes, pads, and vias by the Excellon drill machine.
2. The board is completely re-plated with another layer of copper, coating the bores of all holes with copper and thus establishing electrical connections between the two sides of the board.
3. Both sides of the board are coated with a thin layer of photosensitive plastic emulsion; then one sheet of the Gerber artwork is applied to each side of the board, one for the top side copper and one for the bottom side copper, positioned precisely to coincide with the drilled holes. On the artwork, tracks and pads are transparent, and all other areas are black. All bubbles are removed and then ultraviolet light is used to expose the emulsion and make a contact print of the tracks and pads on the board.

NOTE: The image on the Gerber artwork (also a photographic emulsion) is of necessity on one side of the film only. For the top side contact print, this Gerber emulsion is directly against the board emulsion, resulting in a sharp, clean contact print. On the bottom side, however, the Gerber emulsion will be on the wrong side of the film and so will be separated from the board emulsion by the thickness of the film; the result is a fuzzy contact print. This is why all artwork intended for the bottom side of the board should be mirror image. This way the bottom side can also be an emulsion-to-emulsion contact print for maximum sharpness. CIRCAD'98 makes this easy by allowing you to convert any number of Gerber output files to mirror-image.

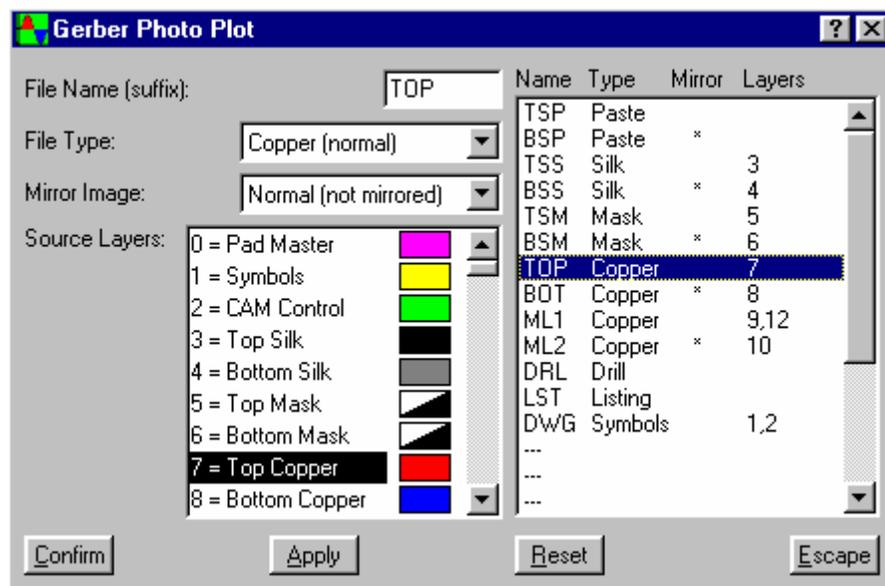
4. The artwork is removed and the board is dipped in a bath that removes all photographically exposed plastic, leaving the copper tracks and pads bare.
5. The board is plated with lead/tin solder, which adheres only to the bare copper tracks and pads.
6. The board is dipped in a bath that removes all remaining plastic emulsion.
7. The board is dipped in a chemical solution that etches away all exposed copper, leaving only the copper tracks and pads that are protected by the coating of lead/tin solder. However, the edges of the copper tracks and pads are slightly undercut by the etching action. The end result of this operation is a bare fiberglass board with solder-coated copper tracks.
8. The next step is known as solder reflow. The board is dipped in an inert oil that is hot enough to melt the solder coating; this allows the solder to flow over the exposed copper edges of the tracks and seal out air, preventing future corrosion. Any tiny scratches in the solder will also be sealed in the process.
9. The board is meticulously cleaned, then baked to remove all moisture.
10. The solder mask (created from Gerber artwork) is applied, covering the entire board except for pads and any areas you may have left bare, such as edge card connectors.
11. The silk screen (also made from Gerber artwork) is applied. CIRCAD'98's component library and superior drawing features result in a very clear, clean silk screen that accurately depicts

the components on the board. This helps to cut down on stuffing errors. And it's easy to put silk screen on the bottom of the board with CIRCAD'98, simplifying troubleshooting.

12. The finished board is now cut out of the large panel by the Excellon drill machine, using a router bit. At present, drill machine operators create the drill tape for this final, critical step directly from the dimensions shown on your drill drawing, rather than from a file. This gives the operator a measure of security that the finished board will not be destroyed by a file containing incorrect commands. Thus, a clear and accurate drill drawing is a necessity.

## Generating Output, Step by Step

When your board is finished and you are ready to generate output, you'll first call the board up on screen. For demonstration purposes, however, just call up the CIRCAD'98 file LPS-C.pcb Now click on **File** and **Gerber Plot** and you'll see the **Gerber Plot** menu, listing the various CAM output files. Most of these are Gerber files; three of them are data files for drill drawings, aperture lists, and Excellon drill files.



Note the **Confirm** button at the bottom left of the menu: This instructs CIRCAD'98 to write the files out to disk, so don't click on this until you are ready to write the files.

The right most column shows the filename suffixes for each CAM output file as well as a flag to indicate whether the layer is to be mirrored, and the source layers that are to be associated with the output data file. Here is a brief description of the fields and their meaning:

### File Name [Suffix]

When these CAM output files are written out disk, they will retain the name of the source file, but the suffix (.PCB) will automatically be replaced by the appropriate suffix for easy identification (this makes consistent file naming conventions fairly important). In the case of the demo board, the filenames would be LPS-C.TSS, LPS-C.TSM, and so on. The files denoted by the suffixes are as follows:

TSS: Top silk screen (in Gerber format)

TSM: Top solder mask (in Gerber format)  
TOP: Top side copper (in Gerber format)  
BOT: Bottom side copper (in Gerber format)  
ML1: Middle layer one (in Gerber format)  
ML2: Middle layer two (in Gerber format)  
BSM: Bottom solder mask (in Gerber format)  
BSS: Bottom silk screen (in Gerber format)  
DRL: Drill data (in Excellon format)  
SYM: Drill symbols (in CIRCAD'98 format)

You can change these suffixes to suit your own file naming conventions by clicking on them and entering the new suffix with the keyboard; the next time you write them out to disk, the new suffixes will be used. You can also add layers for multilayered boards by clicking on the empty block just below the last filename shown and entering the new information.

**File Type** tells CIRCAD'98 what type of file each CAM output file should be; CIRCAD'98 will then automatically include the proper information in that file. This will make more sense as each file type is explained.

There are seven different types of CAM output files. Here is a description of each file type:



### **Copper**

This file type is a Gerber file. CIRCAD'98 automatically includes all pads and segments on the designated source layers -- and draws those pads as solid dots, as is required for a Gerber photoplot.

### **Plane**

This file type is also a Gerber file, and is used for power and ground plane layers on multi-layered boards. Even though plane layers are covered in copper, they still have through-pads passing through them. Each pad that passes through the layer without connection will automatically be drawn by CIRCAD'98 with an annular insulating (non-copper) ring around it. Each pad that does connect to the plane will automatically be drawn as a special thermally relieved pad (an annular insulating ring broken by four small webs) that prevents excessive heat wicking later when the component is soldered in.

### **Mask**

This file type is also a Gerber file, and is used to create the solder resist coating for each side of the board. CIRCAD'98 automatically draws this file type to exclude solder mask from all through-pads and from all surface-pads on copper layers with matching mirror-image codes. The source

pads will be swelled a small amount so the solder mask will completely clear the pads.

If you have edge card connectors or other spaces that need to be clear of solder mask, press <Escape> twice to leave the Cad Output menus, change to the appropriate solder mask layer(s), and place a filled solid to cover the area you wish to leave clear. Note that the entire layer is a "negative" image -- the solid filled regions denote areas where solder mask will not be applied.

### **Paste**

This file type is also a Gerber file, and is used to make a stencil for solder paste application. Used to attach surface mount components, this cardboard stencil has openings over every surface mount pad. CIRCAD'98 automatically includes surface mount pads from copper layers with matching mirror-image codes. The pads will be shrunk a small amount so the stencil holes will be slightly smaller than the pads so solder paste stays within the edges of the pads.

### **Drill**

This file is not a Gerber file, but an Excellon drill file, which must be converted at the board house or at the printing service bureau into a paper tape that runs the drill machine. CIRCAD'98 automatically includes all through-pads in this file.

### **Symbols**

This is a CIRCAD'98 file that will be used to generate the drill drawing needed by the board house. CIRCAD'98 will automatically scan the board to determine the hole sizes required, sort them from smallest to largest, assign tool numbers and symbols for each hole size, and count the holes. The board dimensions must be entered manually, however. To do this, write the file out to disk, open the file in CIRCAD'98, edit it using *Place Text* and *Place Line* commands, and save it to disk.

To plot the file, call it up in CIRCAD'98 and send it to your printer or pen plotter as you would any other CIRCAD'98 file (see **Chapter 9**, Pen Plots).

### **Silk**

This file type is a Gerber file used to generate the silk screen. CIRCAD'98 automatically includes only line segments, arcs and text on designated layers; pads are excluded.

### **Mirror Image**

As discussed in item 3 under "PCB Manufacturing", it is desirable to produce mirror-image Gerber artwork for the bottom layers of the board so that an emulsion-to-emulsion contact print can be achieved. You can do this by simply selecting **YES** in the mirror image column (clicking on the entries causes them to toggle between **YES** and **NO**). CIRCAD'98 will convert the files automatically to produce mirror image Gerber artwork.

### **Source Layers**

This column instructs CIRCAD'98 as to which board layers to include on any given CAM output file. To change the layer number selected, place the cursor on the layer number to be changed and click hilite it. Now, simply select the desired layers in the source layers box.

The selection of source layers for a given Gerber file is fairly straightforward. The source layer for the bottom copper (.BOT) Gerber file should be the bottom copper layer of the board, the source layer for the top solder mask (.TSM) Gerber file should be the top mask layer of the board, and so on. The silk screen and symbol layers bear some explaining, however.

Note that more than one source layer can be selected to produce a single Gerber layer. This feature is useful on both silk and copper layer types. On silk screening, it is possible to place the component shapes on a master silk layer, and the top/bottom text data on two other silk layers. the final output for each of the top and bottom silk would then be the combination of the master

silk and the appropriate text silk layer. For copper layers, a ground flood layer can be combined with a normal tracks layer to produce a composite signal/ground layer. Keeping the flooded ground layer separate in the source file has many advantages when it comes time to modify the board to the next revision level.

The source layer for the Symbol (drill drawing) file is the *Symbols* layer. This is the layer that contains only the outlines and text of the various drawing sheets included as file templates in CIRCAD'98. CIRCAD'98 will automatically include the pad master layer in this file.

### **Writing CAM files out to disk**

When you have completed filling in the CAM Output menu, clicking on **Confirm** will write all the files out to your hard disk. You are then ready to plot the drill drawing for the board house and transport the Gerber files to the printing service bureau, whether by modem transmission, email, or floppy disk.

But first, you may want to call up the Gerber files and check them over visually; CIRCAD'98 will automatically read the Gerber files through its Gerber import filter and display them on screen.

You can now view each one of your Gerber files and see it exactly as the finished product will appear. You may well find details that need cleaning up; if so, it is better to close the Gerber file, reopen the circuit board file (.PCB), make the necessary changes in that file, and then write the Gerber files out to disk once more (overwriting the previous Gerber files in the process). This ensures consistency throughout the files.

### **Generating printed reports**

The final pieces of the output puzzle are the written reports. Three of these [the Gerber aperture list, the Excellon tool list, and the file list] are created at the same time as the Gerber data files. This leaves only the bill of materials and the parts list. To produce these files, you must click on **File** and **BOM Output**.

The bill of materials is nicely sorted by component type and value, indicating the required number of each value, and listing the reference ID number and location (using an alphanumeric grid system) of each component. The parts list contains the same information but is sorted by reference ID number. The Gerber draft codes and Excellon tool codes are listed in order from small to large, and the .lst file lists the file suffixes in use with the corresponding file type and whether the file is mirror-image or not.

With these written reports and CAM output files in hand, you are now ready to provide clear, easy-to-follow directions to the printing service bureau, board house, and stuffing house.

## Chapter 11

# Creating components

CIRCAD'98's standard component libraries do not contain the entire universe of available components. To do so would not only require an inordinate amount of storage space, it would be largely a waste of that space because most of the more obscure components would never be used. For this reason, CIRCAD'98 includes a selection of the most commonly used component patterns, and offers you the ability to create more as your needs require.

- [Library file types](#)
- [Component creation \(overview\)](#)
- [Component creation \(in detail\)](#)
- [Modifying existing components](#)

## Library file types

CIRCAD'98's standard component libraries are not graphics data files, they are ASCII files in a specialized format. This means that you cannot open a library such as TTL.LIB and browse through a collection of schematic drawings of each component contained therein -- all you would see is the raw data defining each component. Instead, you browse CIRCAD'98's libraries with the familiar selector guide in available through the *Place component* command. And when you do place a component, you can see that CIRCAD'98 "knows" how to draw each one of its thousands of standard components. This is a far more efficient way of storing the enormous amounts of information contained in CIRCAD'98's standard libraries.

When you create your own customized component libraries, however, they will be image files, simply because the first logical step in creating a user-defined component is to draw it. This means that you will be able to browse a library of visual images. The procedure for doing this is detailed in **Chapter 3** under *Place component*.

Note, though, that the custom libraries you create and the components they contain will also be accessible through CIRCAD'98's component selector guide when you are in the Library Browse mode of Place component -- just as CIRCAD'98's standard libraries are. CIRCAD'98 automatically captures the required information from your customized components in order to update the component selector guide.

## Component Creation (overview)

Each CIRCAD'98 user has his or her own needs when it comes to creating circuit boards, so the ability to customize component libraries is critical. Taking the time to do it right will ensure that you get the maximum benefit from CIRCAD'98's schematic and PCB editing power, including higher-level functions such as netlist, and design rule checking.

Assuming you are starting from scratch, creating libraries and components is done in six basic steps:

1. Establishing a user-defined schematic library file (this file may contain as many individual components as you wish) and saving it to the CIRCAD'98\LIBRARY sub directory. If you prefer to use the supplied sample library SCH.LIB and add your components to it, open this file using the *File Open* command.
2. Creating a new component, or pattern, for the schematic library file and re-saving the library file.
3. Establishing a user-defined PCB library file and saving it to the CIRCAD'98.LIB sub directory (again, this file can contain as many components as is convenient). If you prefer to use the supplied sample library PCB.LIB and add your components to it, open this file using the *File Open* command.
4. Creating the PCB component pattern (sometimes called a package or a "footprint") that will correspond to the new schematic component just created in step 2, and re-saving the library file.
5. Adding the newly established library files to the appropriate Library Selector menus so they are accessible through the Browse Mode of the Place component command (not necessary if you elected to add your new components to CIRCAD'98's existing sample library files).

6. Reopening your user-defined libraries using the regular *File Open* command from time to time in order to add new components as required.

By following these steps, your new components and libraries will be as fully functional as CIRCAD'98's standard libraries.

## Component Creation (in detail)

The following steps may be used as a guide when creating your own components.

### Establish a new schematic library

The schematic library you are about to create is an ordinary CIRCAD'98 file like any other, so begin by using the *New File* command. Select a schematic file template (very important to customize it as you wish -- perhaps adding a title or inserting boilerplate information or dividers for organization).

Now might be a good time to use the *File Save As ...* command to name your new library file and save it to the CIRCAD'98.LIB sub directory.

If a user-defined schematic component library already exists and you wish to add an additional component to it, or if you intend to use CIRCAD'98's sample libraries, use the *File Open* command, go to the CIRCAD'98.LIB sub directory, open the desired library file, and proceed to the following step.

### Create a schematic component

Now you will use CIRCAD'98's working commands to draw the image of the new component as you will want it to appear on a schematic when you call it out from the library. If you are creating a five-pin Molex connector, for example, you would place five pads (on the Pad Master layer) to represent the pins, with a set of narrow lines (on the Symbols layer) to represent the outline of the connector. Make the drawing as elaborate or as simple as you want; this is the drawing you will be looking at each time you call this component out of the library in the future.

It is nearly essential to place a few strings of identifying text, such as the *Reference ID* and the *Type/Value* for the component. These parameters are displayed automatically by CIRCAD'98 whenever a text string is encountered that consists of a single ampersand followed by a single numeric digit (i.e. &1, &2, or &3). When first placed, these text strings will be displayed verbatim, but later will reflect the appropriate parameter when the component creation is complete.

Properly identifying pin names is critical to the proper operation of netlist and related functions. When you select *Place Pad*, CIRCAD'98 displays a dialog box listing a number of pad parameters (see **Chapter 3**). The first of these, Pin name, is where you will designate the pin number or name for the new component. As you place each pin in your new component, you will want to carefully refer to your component (or the actual component itself) to make sure that your pin designations match those of the actual parts you will be using later.

Note too that you can assign numbers to pads or names such as 'Cathode'. When you assign a number to a pad, CIRCAD'98 automatically increments the pad number as you place subsequent pins. For example, if the pad you just placed was specified to be pin 7, the next time you use *Place Pad*, the default parameter for pin name will be 8. This can be very useful when placing multiple pads in sequence. You will need to be careful, however, to key in the value manually when beginning a new sequence or when using a non-numeric pin name. Do this by clicking on the existing pin name and entering the new name at the blinking data entry cursor.

This procedure makes sure that CIRCAD'98 knows which pin is which. However, you may want

to add labels to your component pins for the convenience of those who will be reading the schematic. This is done by placing text normally with the *Place Text* command (See Chapter 3).

When you are satisfied with your drawing, you still don't have a component yet -- just a collection of lines, pads, arcs and text strings. To make it into a component, use the Block commands as follows:

1. Use *Block Define* to define a block around the entire component, including text, as described in **Chapter 6** (note that the source reference point you select will be the point by which CIRCAD'98 will grab, rotate, or otherwise manipulate the new component).
2. When the block has been defined, call upon the *Block Build component* function.
3. CIRCAD'98 will display a component parameter list just like the one you've seen in the Place component and Edit component modes, with blanks for Pattern, Ref ID, Type/Value, Modifier. A full description of each of these parameters can be found in Chapter 3 under *Place component*. An additional parameter, 'PCB Pattern', will be explained shortly.

Decide what pattern this new component will be known as from now on, type that in on the **Pattern** line, and press <Enter>. In the case of our Molex connector example, that pattern might be 'MP-5'.

On the **Ref ID** line, you should enter a standard prefix code for this type of component as a convenience (such as 'R?' for resistors or 'Q?' for transistors). On the **Type/value** line, you may want to type in a question mark and the abbreviation for the proper units (?V, ? $\mu$ F, etc. -- see appendix V for more on special characters) as a future prompt. The **Modifier** field is available for any additional parameters that may be desired, such as an in-house corporate part number.

The final line specifies the **PCB Pattern**. This parameter tells CIRCAD'98 which physical PCB package you will want this component to be assigned when it comes time to import your schematic data into a new PCB file. CIRCAD'98 will use this information to select the required PCB components and place them at the bottom of the new PCB file, ready for you to place them.

A PCB pattern that is appropriate for your new schematic component may not yet exist in CIRCAD'98's standard libraries, and will need to be created; this process is covered in the next two sections. However, if you know what the PCB Pattern to be created will be called, enter it now. Note that you can use the same name for the PCB pattern that you used for the schematic pattern -- CIRCAD'98 will not get them mixed up. In the case of our example, giving this new connector the PCB Pattern 'MP-5' would be a perfectly acceptable choice.

4. After you enter the PCB Pattern, the menu and the defined block will disappear from the screen; you have just created a schematic component. Use the *Grab component* command to verify that you can move the entire drawing around as a unit. You can now position the component as desired in your library file and add any labels or descriptions or rules to organize the library.

Create other schematic components as desired. When you are done, close the library file, being sure to save your changes.

### **Establish a new PCB library**

As with the schematic component libraries, the PCB component library you are about to create is an ordinary CIRCAD'98 file like any other, and so you will begin by using the *New File* command. Select a PCB file template (very important!) for the new PCB component library file and customize it as you wish -- perhaps adding a title or inserting boilerplate information or dividers for organization.

Now might be a good time to use the *File Save As ...* command to name your new library file and save it to the CIRCAD'98.LIB sub directory.

If a user-defined PCB component library already exists and you wish to add an additional component to it, or if you intend to use CIRCAD'98's sample libraries, use the *File Open* command, go to the CIRCAD'98.LIB sub directory, open the desired library file, and proceed to the following step.

### **Create a PCB component**

Now you will use CIRCAD'98's working commands to draw the desired PCB component, as you did previously for a new schematic component.

In the case of a PCB component, the most important thing is its physical characteristics: the number, size, and placement of pins, and the silk screen outline and text. On the physical circuit board that will be created from your data file, proper component installation and clearances will depend directly on how accurately you place the pads and silk screen in the pattern when you create the new component, so it is important to take the time to do it right.

Some of the necessary measurements can come from the component's data sheet, if available. If not, calipers can be used to take the measurements from actual components in hand. In the case of the silk screen outline, it is recommended that the outside dimensions of the actual component be physically measured to ensure that the finished board will have the necessary parts clearances -- data sheets often do not give a true representation of the part's outline.

As with schematic component creation, correct pin designations are critical to the proper operation of your new PCB component. The names and numbers you assign to pins in the Place Pad operation must coincide with the schematic and the real-world part you will be using. The procedure for doing this is the same as described in the section above on creating schematic components, but is repeated here for convenience.

When you select *Place Pad*, CIRCAD'98 displays a dialog box listing a number of pad parameters (see **Chapter 3**). The first of these, **Pin name**, is where you will designate the pin number or name for the new component. As you place each pin in your new component, you will want to carefully refer to your component (or the actual component itself) to make sure that your pin designations match those of the actual parts you will be using later.

Note too that you can assign numbers to pads or names such as 'VCC'. When you assign a number to a pad, CIRCAD'98 automatically increments the pad number as you place subsequent pins. For example, if the pad you just placed was specified to be pin 7, the next time you use Place Pad, the default parameter for pin name will be 8. This can be very useful when placing multiple pads in sequence. You will need to be careful, however, to key in the value manually when beginning a new sequence or when using a non-numeric pin name. Do this by clicking on the existing pin name and entering the new name at the blinking data entry cursor.

It is nearly essential to place a few strings of identifying text, such as the *Reference ID* or the *Type/Value* along with the *Modifier* for the component. As in the schematic, these parameters are displayed automatically by CIRCAD'98 whenever a text string is encountered that consists of a single ampersand followed by a single numeric digit (i.e. &1, &2, or &3). When first placed, these text strings will be displayed verbatim, but later will reflect the appropriate parameter when the component creation is complete.

When you are satisfied with your drawing, you still don't have a component yet -- just a collection of lines, pads, arcs and text strings. To make it into a component, use the Block commands as follows:

1. Use *Block Define* to define a block around the entire component, including text, as described

in Chapter 6 (note that the source reference point you select will be the point by which CIRCAD'98 will grab, rotate, or otherwise manipulate the new component).

2. When the block has been defined, call upon the *Block Build Cmp* function.

3. CIRCAD'98 will display a component parameter list just like the one you've seen in the Place component and Edit component modes, with blanks for Pattern, Ref ID, Type/Value, Modifier.

Decide what pattern this new component will be known as from now on, type that in on the **Pattern** line, and press <Enter>. In the case of our Molex connector example, that pattern might be 'MP-5'. Note that this pattern must coincide with the PCB Pattern parameter specified for any schematic components which will be using this physical package.

On the **Ref ID** line, you should enter a standard prefix code for this type of component as a convenience (such as 'R?' for resistors or 'Q?' for transistors). On the **Type/value** line, you may want to type in a question mark and the abbreviation for the proper units (?V, ?, ? $\mu$ F, etc. -- see appendix V for more on special characters) as a future prompt. The **Modifier** field is available for any additional parameters that may be desired, such as an in-house corporate part number.

4. After you enter the Modifier, the menu and the defined block will disappear from the screen; you have just created a PCB component. Use the *Grab component* command to verify that you can move the entire drawing around as a unit. You can now position the component as desired in your library file and add any labels or descriptions or rules to organize the library.

Create other schematic components as desired. When you are done, close the library file, being sure to save your changes.

### **Modify the library selector list**

In order to be able to access your new libraries efficiently from the Browse component mode of the Place component command, you must modify the library selector list (this is not necessary if you added your new components to CIRCAD'98's sample image libraries or did not need to create new library files).

Note that the library selector list data is resident in each CIRCAD'98 schematic or PCB file. This means that changes you make to the list in a given file are not global for all CIRCAD'98 files. You must have a CIRCAD'98 schematic or PCB file open in order to modify the selector list -- and the modified selector list will only appear when you are working in that particular file. This feature allows you to specify certain libraries for any given file and to leave out libraries which will not be needed, speeding up the library search process.

If you want to make the revised library selector list global for all new files you create, it is a simple matter to go to each of the template files, modify their selector lists, and save them to disk. Henceforth, whenever you create a new file and call in one of the templates, the revised library selector list will be used. Locating and customizing template files is covered in detail in **Chapter 12**.

To modify the library selector list, first open the file for which you wish to modify the list. Note that you can modify the PCB library selector list only from a PCB file; the same is true for schematic selector guides and Schematic files.

Then, drop down the *Setup* menu from the main menu bar and select *Library File*. Displayed on screen will be the same library selector menu you see when using the *Place component* command, but in a form in which it can be edited. You can now click on any of the 16 available lines and type in the names of the libraries you wish to be included in the selector, in any order you wish. Note that when in the search mode, CIRCAD'98 will search the libraries in this order as well. When you are done, click on **Confirm** to return to the CIRCAD'98 editing screen. The

selector list modification is now complete.

Having completed these steps, your new component libraries are fully functional. This means that the components therein are accessible through the browse and the search modes of the *Place component* command, just like the standard CIRCAD'98 component libraries. All critical characteristics of the components are known to CIRCAD'98, facilitating higher-level functions such as netlist, ratnest, and so on. And, when you export your schematic netlist to a new PCB file for PCB design, the component selection feature will place all components (for which a PCB pattern exists and was specified) at the bottom of the file, ready for placement.

## Modifying existing components

Once a component has been created, CIRCAD'98 commands cannot ordinarily be used to manipulate any one of its parts (pads, lines, text, etc.) apart from the others. However, you may occasionally want to modify an existing component -- without having to delete it and create it again from scratch.

The *Krump component* function allows you to do this. It is essentially to "dissolve" the component back to its constituent parts. First, use the *Grab component* command to grab the component to be modified. Then use the *Edit: Krump Cmp* command to dissolve the component.

A dialog box will appear, warning you that **The selected component is about to be dissolved. Do you wish to proceed?** The effect of this will be to turn the component back into a collection of individual items. It will still look the same, but you will be able to manipulate its parts using normal CIRCAD'98 commands.

Click on the **Yes** button, and the dialog box will disappear, leaving the component in its original position. You can now modify the component using CIRCAD'98 commands.

When you are ready to make the modified data back into a component, simply follow the steps outlined in the previous section, *Creating New components*.

## Chapter 12

# Customizing Template Files

Template files are significant time savers, as they enable you to create a new file without having to establish innumerable setup values from scratch, such as drawing sheet outlines, boilerplate text, layers, colors, draw modes, library selector guides, and so on.

When you create a new file and CIRCAD'98 prompts you for a template, you see a dialog box offering a choice of 12 standard templates: four sizes of drawing sheets, 4 sizes of schematic sheets, and 4 sizes of PCB sheets. You also have an additional selection, **Specialty File**. Selecting this option will produce a text entry box that will let you key in the name of any template file you may create yourself.

You can customize any or all of the 12 standard templates in any way you wish and save them to disk so they can be called up quickly from the template selection menu box. Or you can leave these templates pristine and use them to create any number of new template files, which will be available by using the **Specialty File** selection and typing in the name of the template. There is no limit to the number of user-defined specialty templates you can create and utilize.

- [Customizing the standard templates](#)
- [Creating your own template](#)

## Customizing the standard templates

The standard template files can be opened, edited, and saved just like any other CIRCAD'98 file. The template files can be found in the CIRCAD'98.TPL sub directory, and are identified as follows:

CIRCAD.TPA = A size PCB  
CIRCAD.TPB = B size PCB  
CIRCAD.TPC = C size PCB  
CIRCAD.TPD = D size PCB  
CIRCAD.TSA = A size schematic  
CIRCAD.TSB = B size schematic  
CIRCAD.TSC = C size schematic  
CIRCAD.TSD = D size schematic  
CIRCAD.TDA = A size drawing  
CIRCAD.TDB = B size drawing  
CIRCAD.TDC = C size drawing  
CIRCAD.TDD = D size drawing

One cautionary note: before making significant changes to these templates, you may want to save the originals under another name in the same sub directory so you can retrieve them later if necessary.

At first, your modifications to these templates will probably be limited to customizing the boilerplate text with your company name and other details. Later, as your familiarity with the program increases, you'll develop preferences as to layer setup, draw modes, autopanning, and so on, and will probably want to customize your templates accordingly.

## Creating your own template

A specialty template file is a CIRCAD'98 file like any other. Any PCB, schematic or drawing file can be used as a specialty template file. The only requirement is that it be located in the CIRCAD'98\TEMPLATE sub directory with the rest of the template files -- otherwise, CIRCAD'98 will be unable to find it.

Make as many specialty template files as you need; there is no limit on the number CIRCAD'98 can support. They can be as simple or as elaborate as your needs dictate, and their uses are limited only by your imagination.

## Chapter 13

# Shortcuts and Other Features

- Ortho Snap Mode
- Automatic Via Placement
- Automatic component selection
- Relative Zero Point
- Error Log
- Quick layer Changes
- Absolute cursor positioning
- Keystroke macros
- Shortcut keys

## Ortho Snap mode

In some cases, you will want to adjust the position of an item straight up or down without changing the x coordinate; similarly, you may occasionally want to adjust an item's position straight left or right without changing the y coordinate. This operation might be slightly tedious if not for CIRCAD'98's *Ortho Snap* mode.

In Ortho Snap mode, any item that is picked up using one of the *Grab* commands can only be moved in the x direction or the y direction, but not both. The Ortho Snap mode is turned on and off using the *Setup Ortho Mode* command. The current status of the Ortho Snap setting is indicated by an asterisk on the status line just to the right of Grid Snap setting whenever Ortho Snap mode is enabled.

## Automatic Via Placement

CIRCAD'98's automatic via placement feature allows you to lay tracks while hopping from layer to layer as necessary, inserting vias as you go. This feature can be a great time-saver when manually routing a dense board.

The auto via placement mode is always in effect, and does not require initialization. There are two ways of using the feature:

1. When in Place Line mode. If you are in Place Line mode, a via will be placed automatically when you change layers. You can do this quickly using the <+> and <-> keys on your 10-key numeric keypad (described later in this section under 'Quick layer changes'). You can also change layers in the conventional manner using the Zoom Layer command.

2. When in Place Vertex Mode. If you need to place a via in a track that has already been placed, use the *Place Vertex* command. Put the cursor on the line where you want the via to be placed and click on that spot to tack down a vertex. Switch to the new layer, move the cursor to the location of the next vertex and click on that spot to tack down a line the next line segment. A via will automatically appear at the point of the previous vertex where the layer change was made. You can continue tacking down vertices on the new layer without adding any new vias. To complete the process, change back to the original layer and click on the final vertex point to place the final via at that location. Pressing <Escape> or <right mouse> will get you out of the place vertex operation.

The vias placed automatically will conform to the parameters previously established in the *Setup Via Parameters* command. For a detailed discussion of setting these parameters, see **Chapter 4**.

## Automatic component selection

When your schematic is finalized and it's time to create a raw PCB board, each schematic component on your bill of materials must be "picked out of inventory" in an appropriate PCB package and placed in the new PCB file. This is ordinarily a tedious operation offering many opportunities for data entry errors to creep in.

CIRCAD'98, however, makes the most of the data already on disk in your schematic file to reduce or eliminate the time spent on component selection. This is done by giving you the opportunity to select the PCB package (or pattern) at the time you place the component in your schematic. Then, when you write a netlist of the schematic out to disk, a parts list will automatically be

included. When the netlist is later read into your new PCB file, the parts list is read in also. For every schematic component for which a PCB pattern was specified, CIRCAD'98 then automatically selects the appropriate PCB pattern and lines all the components up across the bottom of the new PCB file, ready for you to place them on the board. Any schematic components for which a PCB pattern was not specified will not be placed; these will become apparent when you examine your netlist error log.

This feature cuts out an entire tedious step, and lets you proceed straight to component placement, ratsnesting, and routing.

## Relative Zero Point

CIRCAD'98 employs an absolute coordinate system which tracks the position of the cursor in inches from the origin and displays the coordinates on the status line. Often, however, you will need to take a measurement from one item to another on the circuit board, or place an item a certain distance away from another item.

This could, of course, be accomplished by noting the cursor coordinates of the one item, moving the cursor to the other item, and calculating the difference -- but there is an easier way: set a local zero point and use relative addressing.

To use this feature, simply place the cursor where you want the measurements to begin and type <SP> to invoke *Setup Zero Point* function. This establishes a temporary coordinate system with the current cursor position as the origin (0,0). Then it is a simple matter to move the cursor to the new measurement position and note the cursor coordinates.

To clear the local zero point and return to the normal absolute coordinates, simply invoke *Setup Zero Point* function a second time. The current status of the local zero point mode is indicated by an asterisk on the status line just to the right of the xy position whenever relative addressing is in effect.

Once the system is back in absolute addressing mode, a new local zero point can be selected whenever it is desired.

This function comes in handy when creating components.

## Error Log

Each time you import a netlist into a PCB file, CIRCAD'98 notes any errors and writes them out to a file having the same path and filename as the PCB file, except for the suffix .LOG. This is a text file which can be viewed or printed using standard text editors.

This feature is particularly useful when producing a board revision. If a schematic has been modified and then the PCB is to be revised accordingly, it is a simple matter to generate a netlist from the revised schematic and then read it into the old version of the PCB file. The error log will note any and all differences between the new netlist and the old file, providing you a "roadmap" to making the necessary PCB revisions.

For a listing and explanation of the types of errors CIRCAD'98 can detect, see **Chapter 7** under Netlist Input.

## Quick layer changes

Changing layers is ordinarily done using the *Zoom Layer* command, but since this is a very common operation, a more convenient method is also available.

You can hop up or down one layer using the <+> or <-> keys on your 10-key pad. Pressing either key repeatedly can quickly toggle you through all the layers; the status line keeps you abreast of your current position.

## Absolute cursor positioning

The cursor can be jumped to any x,y position, regardless of the current grid snap setting, simply by typing the desired coordinates using number keys on the top row of the main keyboard. Upon entry of the first digit, the status line will display a prompting message and will echo the remaining input keystrokes. Relative positioning is achieved by preceding the x and y values with a <+> or <-> sign. To jump to a negative position absolute, it is necessary to enter <0-x> or <0-y> for the desired coordinate.

## Keystroke macros

A keystroke macro function is available by using the multiply <\*> key on the ten-key pad. The dual sequence <Alt-\*> is used at both the beginning and the ending of the macro. All intervening keystrokes (but no mouse movements) are copied into a special keystroke buffer. This keystroke buffer will be played back each subsequent time the multiply <\*> key is pressed.

## Shortcut keys

A number of the most commonly used functions are available at the touch of a single key, thus bypassing the system of menu selection completely. A summary of the available functions are as follows:

- A Angular rotation of grabbed items.
- Alt-B Define a block to be moved.
- C Center the screen on the current cursor.
- Alt-G Select the next sequential grid snap setting.
- H Highlight a named signal net.
- Alt-H Scan a signal net.
- I Zoom in (1.414x).
- J Global highlight a named signal net.
- Alt-J Re-name a highlighted signal net.
- K Re-scale grabbed items.

L Select a new layer.

M Mirror grabbed items top to bottom.

Alt-N Select the next sequential open data file.

O Zoom out (0.707x).

Alt-O Greek Omega ( $\Omega$ ).

Alt-P Plot/print the current file.

R Orthogonal rotation of grabbed items.

Alt-R Set/clear relative zero point.

T Toggle ortho-snap mode.

U Upside-down rotation of grabbed items.

Alt-U Greek mu ( $\mu$ ).

Alt-W Save the current file to disk.

X Mirror grabbed items in the X direction.

Alt-X Exit from CIRCAD'98 and return to DOS.

Y Mirror grabbed items in the Y direction.

Alt-2 Fraction one-half ( $1/2$ ).

Alt-4 Fraction one-fourth ( $1/4$ ).

F9 Clear the screen and re-draw all layers.

Ctl-F9 Snap the cursor to the nearest item.

F10 Clear the screen and re-draw the current layer only.

Alt-F10 Invoke the on-line help system.

Alt-Hyphen Select the previous sequential open data file.

F1 Place Line  
F2 Place Arc  
F3 Place Pad  
F4 Place Text  
F5 Place Ortho-line  
F6 Place Component  
F7 Place Via  
F8 Place Vertex  
F9 Zoom Re-draw  
F10 Zoom Re-draw Layer

Alt F1 Grab Line

Alt F2 Grab Arc edge  
Alt F3 Grab Pad  
Alt F4 Grab Text  
Alt F5 Grab Item  
Alt F6 Grab Component Only  
Alt F7 Grab Component  
Alt F8 Grab Vertex

Ctrl F1 Edit Line  
Ctrl F2 Edit Arc  
Ctrl F3 Edit Pad  
Ctrl F4 Edit Text data  
Ctrl F5 Edit Text parameters  
Ctrl F6 Edit Component  
Ctrl F7 Locate Component  
Ctrl F8 Krump Component  
Ctrl F9 Snap to Item

Shift F1 Delete Line  
Shift F2 Delete Arc  
Shift F3 Delete Pad  
Shift F4 Delete Text  
Shift F5 Delete Multi-line  
Shift F6 Delete Component  
Shift F7 Delete Grabs  
Shift F8 Delete Vertex

## Appendix A

# Specifications

The following list describes the basic features of the system.

**Resolution:** The absolute resolution of a defined point is  $1/65,536$  of an inch.

**Workspace Size:** The workspace size is user selectable in one mil increments to any value within the range of from -32,768 inches to +32,767 inches in both the X and Y axis.

**Layers:** There is one master layer and two hundred fifty five user layers. The master layer is used for through-pads and board outlines. The user layers can be independently named and tagged as containing signal traces, solder resist masking, or silk screening data.

**Zoom Scale:** The zoom scale can vary in a nearly continuous manner from 65,535 pixels per inch (zoomed in) to 1 pixel per inch (zoomed out).

**Grid Snap:** The grid snap can range up to 1 inch in increments of  $1/65,536$  of an inch. The active value is selected sequentially from a list that is user definable. The list can contain up to 16 elements. In addition to this list, the grid snap can be set manually to any value, including values that are not contained in the standard list.

**Line Width:** Line widths can range from  $1/65,536$  of an inch to 1 inch.

**Pad Size:** Pad size can range from  $1/65,536$  of an inch to 1 inch. The x and y dimensions can be set independently of one another.

**Pad Shape:** Pads can be round, square, oval, or rectangular.

**Drill Holes Size:** The drill hole size can range from  $1/65,536$  of an inch to 1 inch.

**Text Size:** Text string data can range from  $1/65,536$  of an inch to 1 inch.

**Text Orientation:** Text can be rotated or tilted at any angle and can be mirrored in any orientation. The rotational angle is specified internally in  $1/65,536$  of a complete circle.

**Character Set:** The character set used has been derived from the extended IBM-PC set. Some of the more useful special characters are defined in Appendix B.

**Circle Size:** The radius of a circle can range from  $1/65,536$  of an inch to 8,192 inches. The line that comprises the circumference of the circle can range from  $1/65,536$  of an inch to 1 inch in thickness.

**Arc Resolution:** An arc is implemented as a general case of a normal circle. The beginning angle and the subtended angle are specified internally in  $1/65,536$  of a complete circle.

## Appendix B

### File Elements

- **Line Segments**
- **Arcs and Circles**
- **Pads**
- **Text Strings**
- **Components**

## Line Segments

The line segment is the simplest and most basic of the primitive elements used by CIRCAD'98. It is the fundamental building block upon which all other primitive elements are constructed. When data files are converted for export to alien devices such as plotters, everything is reduced to a series of simple line segments. Due to the primary significance of line segments, a complete understanding of line segments is essential to the understanding of CIRCAD'98 itself.

A physical line segment can be described as the locus of all points on a plane within a specified distance from an ideal mathematical line segment. The *plane* of the line segment designates its layer number or color code. The *specified distance* is equal to half the width of the physical line segment. Just as the physical line segment extends wider than the mathematical one, it also extends longer [past the mathematical endpoints] and has a semi-circular shape on the ends.

Internally, a line segment occupies a single element in a file list. The two endpoints [(x1,y1) and (x2,y2)] are each represented by a signed, ordered pair of 32-bit binary numbers. This allows both the range and domain to vary from -32,768.000 to +32,767.99998 with a specified precision of 1/65536 of an inch.

The track width of the line segment is represented by an unsigned 16-bit binary number specifying the diameter of the "pen" that is used to draw the line segment. This restricts the line width to a maximum of .99998 inches.

## Arcs and Circles

In order of complexity, an arc or circle segment follows directly from a line segment. Mathematically speaking, a circle is described as the locus of all points on a plane at a specified distance  $r$  from a central point  $(x_0, y_0)$ . An arc is described as any sub-section of a complete circle, starting at a given 'arc angle' and subtending a given 'sweep angle'.

Internally, an arc occupies a single list element. The central point [(x0,y0)] is represented by a signed, ordered pair of 32-bit binary numbers. The radius is represented by an unsigned 30-bit binary number. The starting angle and sweep angle are each represented by an 16-bit binary number specifying the angle in 1/65536 of a complete circle.

Since the final output from CIRCAD'98 will be directed to a vector plotter or a device that emulates a vector plotter, the arcs are actually constructed from a number of short line segments constituting an inscribed 'n-gon' of the mathematical circle.

The line segments used to construct the arc behave as any other line segment. Also, they are subject to the same constraints as a normal line segment.

## Pads

A pad is a complex primitive that takes on a pseudo three-dimensional structure in that it permeates multiple layers on the circuit board. Simply described, a pad defines a location on the board that will receive a pin from an external device such as a component or a connector.

The simplest form of a pad is a surface-pad residing on one of the two outermost layers of the circuit board. It is used for old fashioned edge card connectors and for newfangled surface mount components. It consists of a small area of tin or gold plated copper on either the top or bottom surface of the board with a corresponding hole in the appropriate solder-resist masking

layer. Surface-pads can be specified as being either oval or rectangular in shape.

A more complex form of a pad is a through-pad, which penetrates all layers of the board. It is used for standard components and connectors of the normal type. To make any sense as a through-pad, a hole must be drilled in the board and electroplated to make contact with all layers. Through-pads can be specified as being either round or square in shape.

In addition to standard connections with line segments on normal layers, through-pads can be attached with a special thermally relieved connection to one of the designated power or ground plane layers of the board.

## Text Strings

A text string is a short sequence of alphabetic, numeric, or special characters that are processed as a single entity and displayed as a single line. The characters are constructed of short line segments from a stroke-oriented character generation table that is loaded at the time of program startup.

Text strings are intended for use primarily as simple labels on components and as such, no provisions have been made for a variety of character fonts, styles, or typefaces. The basic block font is the only one allowed. Characters can, however, be magnified in a linear manner to provide physically larger characters with the same shape and style. The magnification factor is specified as a positive integer less than or equal to 127. Since the basic character is six mils in height, the resulting magnified characters can be as large as three-quarters of an inch in height.

## Components

A component is simply a pre-defined grouping of any number of the above mentioned primitive elements. The individual items within a component lose their distinct identity and can be processed only as part of the whole. It is good practice to keep components as general as possible in the main library and customize each one by rotation, translation, and parameterization at the time each one is placed on the board.

When a component is created, it is given a 'pattern' name to identify it such that at a later time, it can be extracted from the component library and customized to a specific application.

At the time that a component is placed on a board, it is given a unique 'reference identifier' [i.e., Q147], along with a 'type or value' designation that will be used in the bill of materials.

## Appendix C

### Data File Format

- Internal Data Structure
- External Data Structure

# Internal Data Structure

Internal to the program, all data is stored in a recursive doubly linked list structure. There are several such interspersed lists to contain the various data types. Each list element is exactly 16 bytes in length and is aligned to a 16-byte boundary. The list element number is defined to be its absolute memory address divided by 16. Defined in this manner, the list element number is identical to the segment portion of an 8086 segment: offset data address. The hardware instructions to load segment registers can thus be utilized to efficiently step through a data structure as large as the allowable system memory. Using this clever technique of addressing memory, the otherwise clumsy Intel addressing scheme is used to a significant advantage.

The doubly linked list structure provides an environment in which individual elements or groups of elements can be efficiently added, deleted, or moved to a new location at any time. The list structure is recursively defined in that an entire list of any size can be linked as a sublist to any other list. The pointers are arranged in such a way that a list can be traversed from any point in either direction at the current level or in depth. The in-depth search can be either linear or recursive, depending on the type of function desired.

At the head of each list is a list origin element. It contains the upward link to the next shallower list. All elements contain bidirectional pointers to the next sequential elements. A list is said to be empty when the origin element points to itself. Two element types contain a downward link to the next deeper sub-list. They are used to implement the full-nested list structure.

To make the data structure complete, two element types contain linkage pointers to linear sublists. In a schematic representation, these linear sublists would constitute a tail hanging off of the main list. These linear sublists are useful for text strings and other types of purely sequential data.

## **Circular list elements.**

### **Origin:**

0-1 Downward pointer to linear sublist containing the name of this list. A zero value indicates a null linear sublist.

2-3 Backward pointer to last element in this list.

4-5 Forward pointer to first element in this list.

6.7 Flag bit to be set as a list resource allocation indicator.

6.6-4 List origin identification code [000].

6.3-0 List type code [Prime=000, component=110, General=111]. This code is redundant with list element bits

6.6-4 at the next higher level.

7 List update/cleanup flag bits.

8-9 Reference X coordinate in mils [signed].

10-11 Reference Y coordinate in mils [signed].

12-13 Pointer to linear sublist containing various data in an unspecified format. Normally set to binary zeros when not in use.

14-15 Upward list pointer to next higher level list. On the prime origin, this entry points to itself.

### **Line Segment:**

0-1 Downward pointer to linear sublist containing a text string representing the signal name.

2-3 Backward pointer to preceding list element.

4-5 Forward pointer to next sequential list element.

6.7 Flag bit.

6.6-4 Line segment identification code [001].

6.3-0 Layer number.

7 Line thickness in mils. Zero indicates an area fill.

8-9 Beginning X coordinate in mils [signed].

10-11 Beginning Y coordinate in mils [signed].

12-13 Ending X coordinate in mils [signed].

14-15 Ending Y coordinate in mils [signed].

### **Arc/Circle:**

0-1 Downward pointer to linear sublist containing a text string representing the signal name for circular arcs.

0 Rotation angle in /128 radians [unsigned] for elliptical arcs.

1 Elliptical eccentricity expressed as a fraction  $r/R$  [unsigned] for circular arcs. A zero value indicates unity.

2-3 Backward pointer to preceding list element.

4-5 Forward pointer to next sequential list element.

6.7 Flag bit.

6.6-4 Arc/circle identification code [010].

6.3-0 Layer number.

7 Line thickness in mils.

8-9 Central X coordinate in mils [signed].

10-11 Central Y coordinate in mils [signed].

12-13 Radius in mils [signed]. A negative value indicates an elliptical arc.

14 Subtended arc angle in /128 radians [unsigned].

15 Beginning arc angle in /128 radians [unsigned].

**Pad/Via:**

0-1 Downward pointer to linear sublist containing a text string representing the signal name.

2-3 Backward pointer to preceding list element.

4-5 Forward pointer to next sequential list element.

6.7 Flag bit.

6.6-4 Pad/via identification code [011].

6.3-0 Layer number.

7.7-4 Power/ground plane attachment layer number.

7.3-0 Pad type code.

8-9 Central X coordinate in mils [signed].

10-11 Central Y coordinate in mils [signed].

12 Pad size or X size in mils [unsigned].

13 Hole size or Y size in mils [unsigned].

14-15 Downward pointer to linear sublist containing a text string representing the pin name.

**Text String:**

0-1 Downward pointer to linear sublist containing the text string.

2-3 Backward pointer to preceding list element.

4-5 Forward pointer to next sequential list element.

6.7 Flag bit.

6.6-4 Text string identification code [100].

6.3-0 Layer number.

7 Text size magnifier.

8-9 Reference X coordinate in mils [signed].

10-11 Reference Y coordinate in mils [signed].

12-13 Longitudinal string length in mils [signed].

14 Rotation angle in /128 radians [unsigned].

15 Reserved. Always set to binary zeros.

**Block Reference:**

- 0-1 Linking pointer to the origin element of a circular sublist containing the block data.
- 2-3 Backward pointer to preceding list element.
- 4-5 Forward pointer to next sequential list element.
- 6.7 Flag bit.
- 6.6-4 Block reference identification code [101].
- 6.3-0 Reserved. Always set to binary zeros.
- 7 Reserved. Always set to binary zeros.
- 8-9 Reference X coordinate in mils [signed].
- 10-11 Reference Y coordinate in mils [signed].
- 12-13 Downward pointer to linear sublist containing parameters for this block.
- 14 Rotation angle in /128 radians [unsigned].
- 15 Reserved. Always set to binary zeros.

**component:**

- 0-1 Downward pointer to the origin element of a circular sublist containing the component data.
- 2-3 Backward pointer to preceding list element.
- 4-5 Forward pointer to next sequential list element.
- 6.7 Flag bit.
- 6.6-4 component identification code [110].
- 6.3-0 Flag bits used by background cleanup task.
- 7 Sublist update/cleanup flags.
- 8-9 Beginning X coordinate in mils [signed].
- 10-11 Beginning Y coordinate in mils [signed].
- 12-13 Ending X coordinate in mils [signed].
- 14-15 Ending Y coordinate in mils [signed].

**List Pointer:**

- 0-1 Downward pointer to origin element of a circular sublist.
- 2-3 Backward pointer to preceding list element.
- 4-5 Forward pointer to next sequential list element.

- 6.7 Flag bit.
- 6.6-4 List identification code [111].
- 6.3-0 Flag bits used by background cleanup task.
- 7 Sublist update/cleanup flags.
- 8-9 Beginning X coordinate in mils [signed].
- 10-11 Beginning Y coordinate in mils [signed].
- 12-13 Ending X coordinate in mils [signed].
- 14-15 Ending Y coordinate in mils [signed].

### **Linear list elements**

#### **Text String Element:**

- 0-1 Forward pointer to next sequential list element. A zero value indicates end of list.
- 2-15 ASCII text data. A zero byte indicates end of data in the current list element. The text string will be continued in the next sequential list element.

#### **General Purpose Element:**

- 0-1 Forward pointer to next sequential list element. A zero value indicates end of list.
- 2-15 General data storage of unspecified format.

## **External Data Structure**

The external data structure closely follows the internal format. All pointer fields are removed, and the data is represented as a linear string. Each data field is dumped out in ASCII format with a single trailing blank as a delimiter. Each primitive is on its own data line with a trailing CRLF sequence.

#### **Origin:**

1. List origin identification code [0].
2. List type code [component=6, General=7]. This code is redundant with item 1 at the next higher level.
3. Reference X coordinate in mils [signed].
4. Reference Y coordinate in mils [signed].

#### **Line Segment:**

1. Line segment identification code [1].
2. Layer number [0-15].
3. Line thickness in mils. Zero indicates an area fill.

4. Beginning X coordinate in mils [signed].
5. Beginning Y coordinate in mils [signed].
6. Ending X coordinate in mils [signed].
7. Ending Y coordinate in mils [signed].

**Arc/Circle:**

1. Arc/circle identification code [2].
2. Layer number [0-15].
3. Line thickness in mils.
4. Central X coordinate in mils [signed].
5. Central Y coordinate in mils [signed].
6. Radius in mils [unsigned].
7. Beginning arc angle in /128 radians [unsigned].
8. Subtended arc angle in /128 radians [signed]. A negative value indicates a clockwise arc.
9. Elliptical eccentricity expressed as a fraction  $r/R$  in  $n/256$  units [unsigned]. A zero value indicates unity.
10. Axis rotation in /128 radians [unsigned].

**Pad/Via:**

1. Pad/via identification code [3].
2. Layer number [0-15].
3. Pad type code.
4. Central X coordinate in mils [signed].
5. Central Y coordinate in mils [signed].
6. Pad X size in mils [unsigned].
7. Pad Y size in mils [unsigned].
8. Hole size in mils [unsigned].
9. Pin number/name [1-12 characters].
10. Signal name [0-12 characters].

**Text String:**

1. Text string identification code [4].

2. Layer number [0-15].
3. Text size in mils. A negative value indicates X mirrored text.
4. Reference X coordinate in mils [signed].
5. Reference Y coordinate in mils [signed].
6. Longitudinal string length in mils [signed].
7. Rotation angle in /32768 radians [unsigned].
8. Text string [0-200 characters].

**component:**

1. component identification code [6].
2. Layer number [0].
3. Rotation angle in /128 radians [unsigned].
4. Beginning X coordinate in mils [signed].
5. Beginning Y coordinate in mils [signed].
6. Ending X coordinate in mils [signed].
7. Ending Y coordinate in mils [signed].
8. component name/parameters string [0-200 characters].

**List Pointer:**

1. List identification code [7].
2. Layer number [0].
3. Process code [0].
4. Beginning X coordinate in mils [signed].
5. Beginning Y coordinate in mils [signed].
6. Ending X coordinate in mils [signed].
7. Ending Y coordinate in mils [signed].
8. List name string [0-200 characters].