Advanced Routing
and Editing Software

User Manual

Issue 6.0 - November 2002

ICON REFERENCE CHART

Placing and Routing


Component


Package


Trace


Via


Zone


Ratsnest


Connectivity Highlight


Instant Edit

Pad Style Icons


Circular


Square


DIL


SMT


Edge Connector


Pad Stack


Polygonal SMT

2D Graphics Icons


Line


Box


Circle


Arc


Path


Text


Symbol


Marker

Editing Commands


Block Copy


Block Move


Block Delete


Block Rotate


Pick Package


Decompose


Make Package

File and Print Icons


New


Save


Print


Import Region


Export Region


Set Print


CADCAM Output


Gerber View & Panelization

Display Commands


Redraw


Flip


Show Grid


Layer Colours


Metric On/Off


Origin


Polar Coordinates


X-Cursor


Zoom in


Zoom ou


Zoom All


Zoom Area

Layout tools


Real Time Snap


Trace Angle Lock


Auto-Style Selection


Auto-Track Necking


Find


Auto number Generator


Autorouter


Autoplacer


Design Rule Checker


Connectivity Rule Checker

Rotate And Mirror Icons


Rotate Clockwise


Rotate Anti-clockwise


Flip Horizontal


Flip Vertical

TABLE OF CONTENTS

INTRODUCTION

WHAT IS ARES?

ARES (Advanced Routing and Editing Software) forms the PCB layout module of the PROTEUS system and offers netlist based PCB design complete with a suite of high performance design automation tools.

The latest version, is compatible with Windows 98/Me/2k/XP and later. It includes a brand new Auto-Placer, improved Auto-Routing, automatic Gate-Swap optimization and even more powerful support for power planes.

Layout Editor Features

Major features of ARES include:

- 32 bit high-precision database giving a linear resolution of 10nm, an angular resolution of 0.1° and a maximum board size of +/- 10m. ARES supports 16 copper layers, two silk screens, four mechanical layers plus solder resist and paste mask layers.
- Netlist based integration with ISIS schematic capture, including the ability to specify routing information on the schematic.
- Automatic Back-Annotation of component renumbering, pin-swap and gate-swap changes.
- Physical and Connectivity Rule Check reports.
- Powerful route editing features including topological route editing, auto track necking and curved trace support.
- 2D Drawing with Symbol Library.
- Comprehensive package libraries for both through hole and surface mount parts including SM782 standard SMT footprints. There are now over 1000 parts in total in the package library.
- Unlimited Pad/Trace/Via Styles.
- Full metric and SMT support. This includes all dialogue form fields as well as the co-ordinate display and grid settings.
- Output to a wide range of printers and plotters. Also output in DXF, EPS, WMF and BMP graphics formats - to file or clipboard where appropriate.
- Built in Gerber Viewer - this enables you to check your Gerber output files before spending money on bureau fees or board manufacture.

- DXF import as standard, Gerber file conversion as an option..

Auto-Placer

ARES includes an autoplacer module, an addition which in combination with the auto-router makes it possible to create PCBs almost entirely automatically. Alternatively, you can use it interactively, either by pre-placing critical components manually and auto-placing the rest, or else by using it to place small sections of the design in turn with manual adjustments being made after each section is placed.

The auto-placer is highly configurable and can be set up to handle a wide variety of board types.

Auto-Routing

The most tedious, error prone and time consuming part of the traditional electronics development process is undoubtedly routing the PCB. As a result, it is in this area that the greatest benefits of Electronic Design Automation are to be found.

Our autorouter is the result of intensive (and continuing) research into techniques for getting the highest completion rates and you will find it capable of near 100% completion of designs that could take days by hand. On the harder jobs where some routes are left undone you will find that *Topological Route Editing* is ideally suited to moving existing routes so that the rest can be completed.

The latest version incorporates a rip-up & retry algorithm to ensure maximum possible completion rate. At the same time, new algorithms have been added to generate fan-out patterns for rows of off-grid surface mount pads.

The auto-router also includes a tidy pass which reduces track length and via count as well as improving the aesthetic quality of the layout.

Power Plane Support

ARES supports Polygonal Gridless Power Planes which overcome most, if not all of the disadvantages associated with other methods of implementing copper fills. The essence of the approach is to generate polygonal boundaries around all the objects within the target area, and then to merge them together. The resultant multi-edged hole boundaries are subtracted from the original (user placed boundary) in order to establish whether or not there is complete or partial connectivity.

Our implementation of this is fast - to the point that real time update is feasible for a modest board on a fast PC, and takes care of both 'slivers' - where two holes nearly touch, but not

quite - and also of the issues arising from rendering the computed shapes with a pen of some minimum thickness.

HOW TO USE THIS MANUAL

The Graphical User Interface and the general intelligence of the software itself will enable many users to be productive almost from the outset. However, as with ISIS there is a great deal of functionality 'under the hood' and you cannot expect to master all aspects of the package immediately.

For those who need some initial tuition and guidance, we have followed the adage that the best way to learn is by doing - once you have installed the package by following the instructions in the next chapter, we suggest that you proceed to work through the extensive tutorial. This takes you right the way through the PCB design process from loading a netlist to performing the final CRC and DRC checks.

The remainder of the chapters provided background detail on all aspects of the system, and for quick reference, the final chapter deals with all the commands and any associated dialogue forms.

An index is provided as a further aid to reference.

GETTING STARTED

INSTALLATION

ARES is installed by the PROTEUS installation program and the procedures for using this are documented in the common installation instructions at the front of the binder. This installation process will also copy files for ISIS and ProSpice if you have purchased them.

After you have completed the PROTEUS installation, you may also want to configure ARES for your system and it is the procedures for this that are documented here.

START UP OPTIONS

The installation process will create a group called "Proteus 6 Professional" and ARES may be started by double-clicking the ARES icon. In addition, the setup program installs appropriate file-associations such that double-clicking on PCB layout files will also launch a copy of ARES.

SCREEN AND HARD COPY DRIVERS

Graphics Driver

ARES simply draws itself through whatever display driver you have installed for Windows. Thus the resolution, number of colours and so forth is determined by this choice, and not by ARES.

Printer & Plotter Drivers

Again, ARES will (in theory at least) generate output through any Windows printer or plotter device driver. In addition to printing through standard Windows drivers, ARES can also generate Gerber and Excellon drill output files as well as WMF, BMP, DXF and EPS files for graphics export. All this activity is controlled from the *Output* menu.

If you have problems with output, do feel free to contact us for technical support. However, the quality of some Windows device drivers leaves a lot to be desired and some problems may be beyond our control. We cannot be responsible for bugs in code that we have not written!

SYSTEM INFORMATION

The *System Info* command on the *System* menu brings up details of the Release Number of the system and to whom it is registered. Please have this information to hand when you call for technical support.

The command also shows how much free RAM you have available - this includes both free physical and virtual memory. Provided that you have a reasonable amount of free disk space, you should not encounter memory shortages with ARES, although more memory may well improve performance.

In addition, statistics about the number of components, pads, tracks etc. are displayed on this form. Also displayed is the number of 'missing' connections, which equates to the number of ratsnest lines displayed in the work area.

TUTORIAL

INTRODUCTION

The purpose of this tutorial is to familiarize you as quickly as possible with the main features of ARES to the point that you can use the package for real work. Users with modest computer literacy should find it possible to learn the package and produce their first board within a day.

The tutorial proceeds by taking you through worked examples involving all the important aspects of the package including:

- Basic techniques for placement and routing.
- Netlist based design including both manual and automatic routing .
- More advanced editing techniques such as block editing and route editing.
- Report generation - the CRC and DRC tools.
- Hard copy generation.
- Library part creation.

We do urge you to work right the way through the tutorial exercises as many things are pointed out that if missed will result in much wasted time in the long run. Also, having worked through the tutorial and thus got a basic grasp of the concepts behind the package you will find it much easier to absorb the material presented in the reference chapters.

OVERVIEW OF THE LAYOUT EDITOR

We shall assume at this point that you have Proteus installed on your PC, and launched ARES from the *Start Menu*.

The largest area of the screen is called the *Editing Window* and it acts as a partial view on the layout. You can adjust the scale at which the layout is displayed using the zoom options on the *View* menu, or with the associated function keys F6 (Zoom in) and F7 (Zoom out). In this case, the position of the mouse pointer is taken as the new centre for the *Editing Window*. You can also pan to adjacent areas by holding down the SHIFT key and 'bumping' the mouse pointer against the appropriate edge of the *Editing Window*.

The dot grid on the *Editing Window* can be toggled on and off using the *Grid* command, or by pressing 'G'. The spacing of the dots normally reflects the current snap setting, except when zoomed out. In this case, the dot spacing is set to a suitable multiple of the snap spacing.

The smaller area at the top left of the screen is called the *Overview Window*. Not entirely illogically, this is used to display an overview of the entire layout. You can move the work area to a chosen part of the layout by pointing to where you want to go on the *Overview Window*, and clicking left. The green box shows the area currently displayed in the *Editing Window*.

Just below the *Editing Window* is the *Layer Selector* which determines the current layer or layer set. The current layer applies both to the placement and selection of PCB objects.

Which layers are *displayed* can be adjusted by use of the *Layers* command on the *View* menu.

Also at the bottom the screen is the co-ordinate display which reads out the position of the cursor when appropriate. These reflect not the exact position of the pointer but the location to which it has been snapped. Two things affect this:

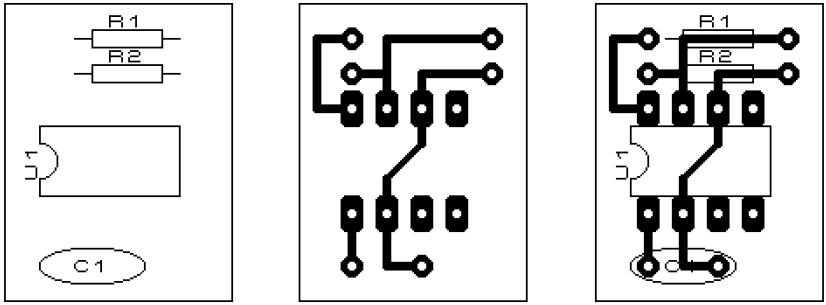
- The currently selected snap grid. The options available appear on the *View* menu and also through keys CTRL-F1 and F2 -F4 . You can re-define the snap values through the *Set Grids* command on the *System* menu.
- Real Time Snap. When this feature is enabled, the cursor will lock onto pads and/or tracks, even if they were not placed on the currently selected snap grid. RTS always snaps to pads and vias, and will also snap to tracks when the Routing mode icon is selected. RTS can be turned on and off using the *Real-Time Snap* command on the *Tools* menu or by pressing CTRL+'S'.

ARES can be set to display an X cursor at the position to which it has snapped the pointer through the *X-Cursor* command, key 'X'.

The co-ordinates can be in imperial or metric units as set by the *Metric* (key 'M') command. You can also set a false origin using the *Origin* command (key 'O') in which case the co-ordinates change colour from black to magenta.

BASIC PLACEMENT & ROUTING TECHNIQUES

Before going on to look at designing a board from a netlist we shall first cover the basics of placing and routing using the extremely simple board shown below.



Package Placement

The most direct way to build up a rough layout of a board is to drive ARES in Package mode. In this mode, you can pick component footprints or *Packages* directly from the library and place them onto the work area.

In our simple example, three packages are used

```
CAP20
DIL08
RES40
```

and you can start by picking them from the package library. To do this, first click left on the *Package* icon (see the icon sheet at the front of the manual). Next, click left on the 'P' toggle at the top left of the *Object Selector* which is now displaying the word 'PACKAGES';. The *Library Pick* form will appear and you can pick the required packages from it. When you have done this, close down the form by clicking the close button (Windows) at the top left.

The three package names should now have appeared in the *Object Selector*, with the last one you picked highlighted. Ensure that DIL08 is selected by clicking left on it, point somewhere in the middle of the *Editing Window* and hold down the left mouse button. A green outline of an 8 pin IC will appear which you can move about with the mouse. Position it roughly central and release the mouse button. Now select the RES40 package and place the two resistors, 0.1" apart (2 grid squares) and just above pin 8 of the IC. Similarly, place the capacitor outline just below pin 1.

Unless you are quite adept, you may not have the components positioned quite correctly, so we will now take a look at how to move things around. Point at a component outline (rather than its pads) and click right. This will 'tag' it, and cause it to be highlighted. Now, still keeping the pointer over it, hold the left button down and drag the mouse. This is one of the

ways to move objects. Also, you can delete a tagged object by pointing at it and clicking right. An *Undo* command is available from the *Edit* menu.

All the objects can be untagged by pointing at no object and clicking right.

Routing

Routing mode is commenced by clicking left on the *Trace* icon. The *Object Selector* will change to display a list of 'Trace Styles' - the default selection of track widths. Select T20 for a 20 thou track.

Traces are placed by clicking left at each point along the required route, and clicking right to finish.

Other points to note about routing mode:

- Clicking twice at the same point places a via and changes the current layer as defined by the *Set Layer Pairs* command on the *System* menu. The selection of via types can be displayed by clicking on the *Via* icon.
- Whilst routing, you can change the current layer by pressing the PGUP and PGDN keys. In addition CTRL-PGUP selects the top layer and CTRL-PGDN selects the bottom layer.
- Holding the CTRL key down allows you to place a curved track segment. The progress of the arc (horizontal then vertical or vice versa) is determined by how you move the mouse away from the fixed point. It is best to press and hold the CTRL key, then move the mouse, then click left, then release the CTRL key.

Annotation

When components are placed in Package mode, they have no annotation information associated with them - later on you will see how components are automatically annotated when a netlist is used.

To annotate the components, select the *Instant Edit* icon and then click left on each component in turn. Each time you do this, a form will appear with fields for the part ID and value. You might find it easier to use only the keyboard when annotating: the cursor keys will move the mouse pointer one grid step at a time and the ENTER key will do for the left mouse button and the OK button on the dialogue forms.

Alternatively, the *Auto Name Generator* command may be used to generate numerical sequences for component numbering.

Part IDs and values can be moved by tagging the parent object and then pointing specifically at the ID or value before dragging using the left mouse button. Bear in mind also that you can set different snapping grids from the *View* menu or by using keys CTRL-F1 and F2-F4.

The default size for these labels can be determined using the *Set Template* command on the *System* menu.

Board Outline

ARES has a special layer, the EDGE layer, which is intended for holding 2D graphics which represent the board outline. Objects placed on the EDGE layer will appear on artwork generated for any of the other layers.

In this case, the board outline is just a box.

To place a rectangular board edge:

1. Select the *Box* icon.
2. Select the EDGE layer from the *Layer Selector*.
3. Point at where you want the top left corner of the box, press the left mouse button and drag out the box to enclose the layout.
4. If you need to resize the box, tag it by pointing at it and clicking right, and then drag the sizing handles as required using the left mouse button.

Curved or irregularly shaped boards are fully supported; the boundary should be formed from lines and arcs, or by selecting the *Path* icon and drawing a single path object.

When you have finished experimenting with this exercise, save it if you wish and then start a new layout for the next exercise by using the *New Layout* command on the file menu.

BLOCK EDITING FACILITIES

We have seen already that an object can tagged by pointing at it and clicking right, and that once tagged it can be dragged (left button) or deleted (right button). There is, however, another way to tag objects. If you point at no object, hold down the right button and drag the mouse, a green box will appear. When you release the button, any objects inside the box will be tagged. Re-load 'PPSU.LYT' and then give this a try by dragging a tag box round the entire layout. Once you have a tag box, the *Copy*, *Move*, *Rotate* and *Delete* icons come into play - try them out, bearing in mind the following:

- The types of objects which are selected by the tag box can be changed through the use of the *Tag Filter* command. This also allows you to choose which layers are affected.
- You can undo a delete using the *Undo* command (key 'U').
- When you perform a block rotation, you will be prompted a mark an origin point about which the rotation/reflection will occur.

Try completing a few routes and then tagging just one or two components and moving them - track segments with one end inside the tag box and the other end outside are stretched.

ROUTE EDITING

Powerful route editing facilities are a major feature of ARES and we shall now take a good look at the features available in ARES.

Unlike many other PCB design packages, route editing in ARES is based around the topology of the current tracking rather than depending in any way on how the sections of track were laid down. In addition, modifications can be made to any part of a route, not just to sections between 'nodes'.

Via Placement

In almost all circumstances, vias are placed automatically for you. To see this in operation, select the *Trace* icon and place a route segment by clicking left at two points. Now click left over the second point a second time, then at a third point and finally click right to terminate the route. Clicking (left) at the same point twice causes ARES to place a via and change the current layer so that (in this case) the next segment will be placed on the top copper layer. At the same time, a via is placed at the point where the route changes layer.

Which layer is selected is determined by the *Set Layer Pairs* command on the *System* menu. It is possible to define layer pairs, triples or whatever as required. You can also change layer

manually whilst placing a route by using the PGDN and PGUP keys, but note that a via is not placed in this case.

The type of via used can be changed by selecting the *Via* icon and choosing one of the via styles from the *Object Selector.* You can also place, replace, tag, move and delete vias manually in this mode. The left and right mouse buttons operate as for manipulating components.

For multi-layer boards, you can select whether normal, blind or buried vias are placed by selecting the *Via* icon and then adjusting the *Layer Selector.*

Changing a Route's Width

If you need to change the width of a section of tracking, there are two approaches:

- Simply select the required trace style and place new tracking over old.
- Tag the tracking with the right mouse button, and then click right a second time and select the required trace style from the context menu.

Remember that to change track widths globally, you can always edit the appropriate trace style by selecting it and clicking on the *Object Selector* 'E' toggle.

Auto Track Necking

In many cases, the reason for necking down a track is to that it can pass between two pads or other obstacles without violating the design rules. The Auto Track Necking feature allows ARES to do this for you.

The function is controlled by the *Set Design Rules* command on the *System* menu. The dialogue form allows you to enter the clearances for pad-pad, pad-trace and trace-trace and also the trace style to neck to. The default neck style is T10 - a 10 thou trace.

Tagging a Route

To re-route, delete or copy a section of tracking requires that you first tag it. In a similar fashion to the object editing facilities, you do this by clicking right on it, though there are one or two subtleties...

- ARES will only 'sense' tracking on the current layer so you must set the *Layer Selector* to make the appropriate layer current. The space bar or the middle mouse button (if your mouse has one) will cause the selection of the next layer in the layer-pair sequence as defined by the *Set Layer Pairs* command. Alternatively, you can move up and down the layers with PGUP and PGDN.

- If you click on a via, or a point at which several tracks are joined, all tracks meeting at that point are tagged.

As with object editing, pointing at nothing and clicking right untags the tagged route.

We suggest that you practice tagging and untagging the routes on the layout before proceeding further - it is important, though not at all difficult, to grasp how this feature works.

Moving/Dragging a Tagged route

Once you have a section of tracking highlighted, you can drag any of its segments by pointing and dragging with the left mouse button.

- If you point at a horizontal or vertical segment, then it will move vertically or horizontally respectively, and any adjacent segments will be 'fixed up' to maintain orthogonal routing.
- If you point at a node (corner) then that single point will be dragged, and the adjacent segments will stretch diagonally.
- If you point at a diagonal segment, then a new node will be created.

Deleting a Tagged Route

There are two ways to delete tracking.

- The *Block Delete* icon will delete all highlight objects on the layout. During route editing, these are typically just the tracks that you have tagged, so this works as a quick way to delete the tagged route.
- You can also delete a single, interconnected section of tracking using the *Context Menu* - see below.

The Context Menu

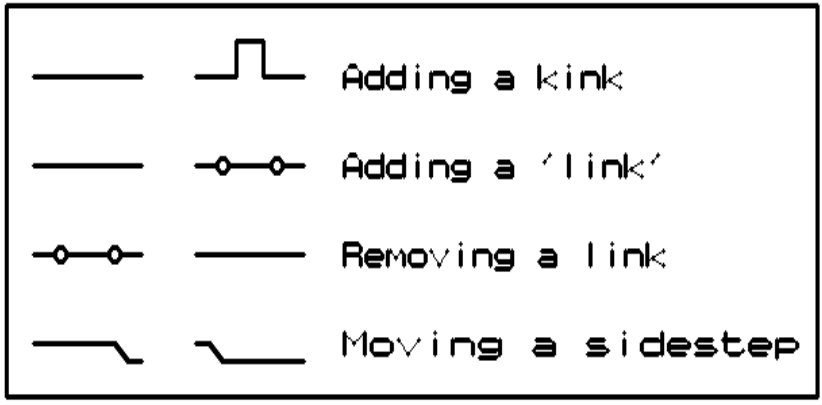
If you click right on a tagged track, a popup menu will appear with options to:

- Delete, Copy and Move the route.
- Change the layer and trace style.
- Change the via style

The *Copy* command provides the means to do memory buses and similar repeated patterns of tracking. You can make as many copies as are required by repeatedly clicking left. Click right to finish.

Re-routing

Finally, there is a very nice way of modifying the actual path taken by a route. Once you have a tagged route, you can alter its path by simply placing a new section of tracking (in the usual way) that starts and ends on the old one. ARES automatically computes which section of the tagged route is shorted out, and removes it. This is a very natural way to work - all of the route editing operations shown in the diagram below can easily be carried out with this feature.



Note that vias which are made redundant (i.e. secondary vias) will also be removed, along with the shorted out section.

You will find this feature especially useful for moving routes that turn out to be blocking other ones towards the end of the routing process.

Connectivity Highlight

Although not strictly part of the route editing facilities, Connectivity Highlight does come in most handy at the end of the routing process when you find yourself needing to check which connections have and (more importantly) have not been made. For example, you might want to know if a clock signal has been connected to all the chips requiring it.

To see the feature in operation, select the *Connectivity Highlight* icon. Click left on a component pad with attached tracking and you will see that everything connected to that pad will be highlighted in bright white. All such highlighted objects will remain highlighted, irrespective of pan and zoom operations, until you invoke the *Redraw* command (key 'R').

Also in this mode, clicking on the net selector toggle will highlight all objects assigned/connected to the currently selected net.

Clicking on the Delete icon will delete traces or vias highlighted by either of the above techniques thus providing the means to delete all or part of a net.

HARD COPY GENERATION

Last, but by no means least, we come to the crucial business of reproducing the pretty on-screen graphics on paper or film. Under Windows, most hard copy devices are supported through the normal Windows printer drivers. Additionally, we supply our own drivers for pen-plotters, Gerber photoplotters and Excellon NC drill machines.

We will deal here solely with printing to an ordinary Windows printer device - it is unlikely that you will have a photoplotter to hand! The first step is to select the correct device to print to using the *Printer Setup* command on the *Output* menu. This activates the Windows common dialogue for printer device selection and configuration. The details are thus dependent on your particular version of Windows and your printer driver - consult Windows and printer driver documentation for details.

Then, with a layout loaded, invoke the *Print* command from the *Output* menu. The dialogue forms offer a number of controls, all of which should be self-explanatory. The default settings should do for getting something and you commence output generation by clicking on OK. Output can be aborted by pressing ESC, although there may be a short delay before everything stops whilst ARES and your printer/plotter empty their buffers.

With plotters in particular, you will probably need to experiment with pens, paper, and the various settings on the *Set Devices* dialogue form in order to get optimum results. Full details may be found under HARD COPY GENERATION on page 101.

You may also wish to compensate for any inaccuracy of scaling in your output device using the layout CALTEST.LYT (in the "Samples\Schematic and PCB Design" directory of your Proteus installation) and the *XCompensation*/*YCompensation* fields which are located on the *Print* dialogue form.

CADCAM output and Gerber Viewing are described in CADCAM OUTPUT on page 107.

THE PACKAGE LIBRARY

Packages are made by placing pads and silk screen graphics in the work area, tagging them, and then invoking the *Make Package* command on the *Library* menu. As a quick example, try the following:

1. Select the *Circular Pad* icon and then style C-80-30 from the Object Selector.
2. Place two circular pads 0.5" apart
3. Then select the *Box* icon and draw a box round the two pads.
4. Tag the whole ensemble by dragging a box round it with the right mouse button.
5. Invoke the *Make Package* command on the *Library* menu and key in `THING` for the name

If you now select the *Package* icon, you will see that THING has appeared in the Object Selector, and can be placed like any of the packages you have used so far. The reference or anchor point for the package is always the first pad placed, unless one was explicitly defined using an ORIGIN marker when the package was created.

There is rather more to it than this - you can, for example, have pads which are on one side only, and silk screen legends on both sides of the board. For further details, see LIBRARY FACILITIES on page 57.

To edit an existing package, pick it from the library, tag it, and then invoke the *Decompose* command. This will break the package into its constant elements (pads and 2D graphics). It is not recommended to do this to a component that is part of a layout.

THE SYMBOL LIBRARY

Symbols are simply groups of 2D graphics objects which are stored in a library for general use. Typical applications include things like drilling targets, graphics for non-electrical components like brackets and heat-sinks, and possibly your company logo.

You make a symbol by tagging the 2D graphics objects which form it (including, if you like, other symbols), and then invoking the *Make Symbol* command. Note that layer information is ignored when making a symbol.

A symbol may also be edited using the *Decompose* command as described for packages.

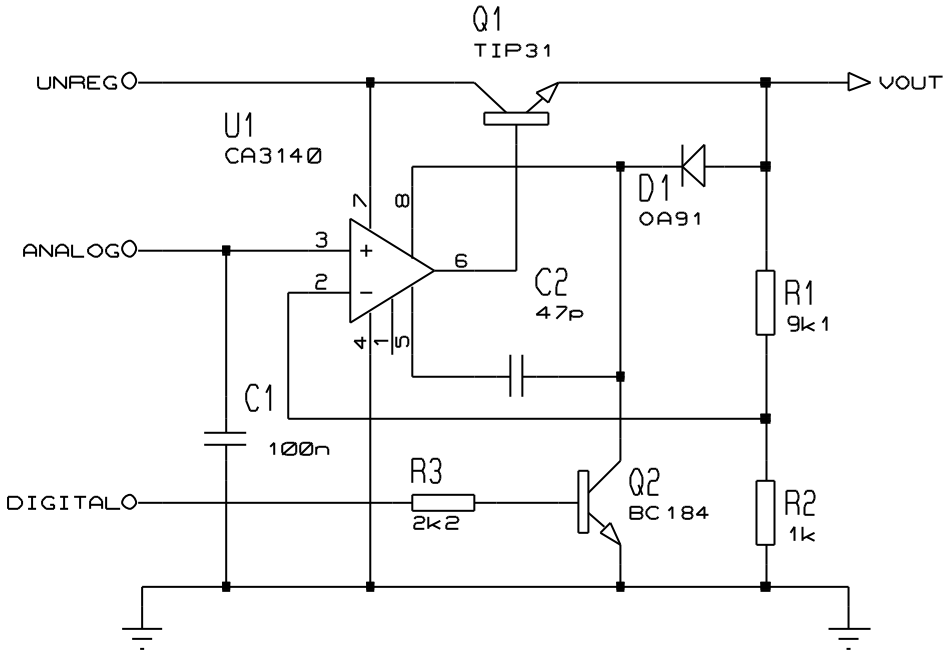
ROUTING A PCB FROM A NETLIST

In the previous exercise, you used ARES as no more than a computerized drawing board. However, ARES is really intended to be used in conjunction with ISIS. Since it is much easier to check that a schematic is correct, the overall result is an improvement in the quality of your design process. Also, since ARES can use the netlist to show you what is connected to what, there is no longer the need for constantly referring to data books in order to check pinouts and so forth.

Preparing a Schematic for PCB Design

For the purposes of this exercise, you need to load up the sample design PPSU.DSN. You will find it in the "Samples\Tutorials" directory within your Proteus installation.

This design is shown below:



The ISIS library parts already contain packaging information for the PCB, by virtue of their **PACKAGE** properties. Thus, you can go straight into PCB design by selecting the *Netlist to ARES* command from the *Tools* menu in ISIS.

📖 For a fuller discussion of preparing schematics for PCB design, see the chapter entitled *ISIS & ARES* in the ISIS manual.

Placing the Components

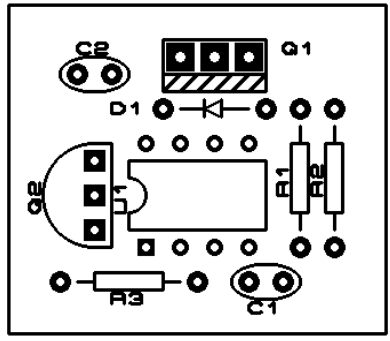
The sample file PPSU.LYT is actually ready placed so the first thing you need to do is remove the components from it. To do this, drag a tag-box round the components (i.e. just inside the board outline) using the right mouse button and then click the *Delete* icon.

Once this is done, selecting the *Component* icon will show all the components in the *Object Selector.* Object placement is the same as in the previous exercise though this time you will need to use the *Rotation* icons to select the orientation for some of the parts. This can be done either before placement, or else if you tag an object and then operate the icon, the object will rotate in sympathy.

You can also rotate a component whilst placing it using the '+' and '-' keys on the numeric keypad.

We suggest that you start with the op-amp and then place the small components around it. In a real situation, it is normal practice to sketch out a floor plan on paper before you start though in this case you can use the diagram below as a guide.

As you place components, they disappear from the selector - this gives you a clear indication of how many components remain to be placed. If placed components are deleted they are restored to the selector. Note though that this does ***not*** apply if components were placed in Package mode and then annotated - only components specified in a netlist are treated this way.



Editing a Placed Component

There are several things you can do to a placed component:

- Move it wholesale by tagging it and dragging with the mouse over the body or pads but not the labels. Whilst dragging, the '+' and '-' keys on the numeric pad will rotate it.
- Move its labels by tagging the whole object and then dragging.
- Move any of its pads by tagging the component, selecting the *Instant Edit* icon, and then dragging the pad with the left mouse button.
- Rotate or reflect it by tagging it and then clicking the *Rotate* icons and/or the *Mirror* icon.

  You can set it to non-orthogonal angles by typing into the angle edit-box on the toolbar.
- Edit its labels by tagging it and then clicking left without moving the mouse.
- Change any of its pads by placing new ones over them in Pad mode.
- Change its package style by selecting package mode, choosing the new package and then placing it such that its pin 1 touches the old package's pin 1. It is not recommended to replace a package with one with a different number of pins as netlist information may be lost. This situation can be retrieved by re-loading the netlist.

The Ratsnest

As you place the components, you will see that green 'ratsnest lines' or 'connections' appear while the object is being placed It should be fairly intuitive that the longer the ratsnest lines, the less optimum the components position and this is indeed the case - optimizing component placement is equivalent to minimizing the total connection length. Unfortunately, solving this problem is more an art than a science!

- Remember that the '+' and '-' keys will rotate each component whilst you are placing or dragging it. Combined with the ratsnest display, this gives you a rapid means to optimize the orientation of each part as you place it.
- Also worthy of note is the fact that this time there is no need to annotate the parts - when you place U1 it comes out as a DIL08 and there is nothing else to do.

ARES re-calculates the ratsnest not only as components are placed, but also while they are being dragged. This can mean that ratsnest lines may ‘jump about’ as the component’s pads move between various possible connection points. It should be obvious, that with something

like a decoupling capacitor, the nearest power pins to it will depend on the region of the board in which it is located.

In addition, ARES shows *Force Vectors* for each placed component. These appear as yellow arrows which point to the optimum location for the component. The shorter the force vectors, the better the overall placement. The force vector for a component being dragged is also updated in real time, although its effect on the force vectors of the other components is not shown until it is placed. Once a component is tracked to, its force vector will automatically disappear.

One final detail is to place the board edge. To do this, select the *Graphics* icon, the *Box* icon and the EDGE layer on the *Layer Selector.* Then drag out a box for the board edge around all the components.

You are now ready to begin routing.

Manual Routing

At this point, we suggest that you load the file PPSU.LYT which represents where you should have got to so far. You will find this file in the "Samples\Tutorials" directory within your Proteus installation..

You should now see that all *Ratsnest* interconnections are displayed. Select the *Trace* icon. and click left on pin 4 of the op-amp. At this point several things will happen:

- A prompt message will appear in the status bar indicating that you are routing part of the ground net.
- All three connections connecting to pin 4 will highlight, indicating which pins it is legal to route to.
- The trace selector will automatically display trace style T25. This is associated with the net-strategies feature, but for now simply take it that this has been predefined as the default thickness for power nets.

  Note that this only occurs if the *Auto-Trace Selection* option on the *Tools* menu is on.
- A green 'rubber' line will appear from pin 4 to the current mouse pointer position - this shows you where a track segment would be placed if you clicked left.
- Point at the left hand pin of C1 and click left a second time. ARES will sense that you have completed the route and will replace the connection line with a segment of 25 thou tracking. Click left on U1 pin 4 again, move up 1 grid square, click left, move over to 1 square above the lower pin of R2 and then down onto the pin.

Now make the connection for U1 pin 4 to Q2's emitter. In this case, note that you need not take the route pin to pin - the Advanced Netlist Management features enable ARES to tell which ratsnest line to remove, even if you take the route from Q2's emitter to the route corner above U1 pin 4.

The next connection to do is the one from U1 pin 2 to R1. Assuming that we want to route this on the top side of the board running above the lower row of U1's pins, press the space bar or else the middle button of your mouse if it has one. This selects the other layer in the current layer pair - in this case 'Top Copper'. Route the connection as before, noting that the trace type selected is now DEFAULT as defined in the SIGNAL strategy. Also note that the trace comes out in red, indicating (with the default colours) that it is on layer C1. Now complete this net by making the connection to R2

It is in fact possible to route this board with no vias at all. However, for the purposes of tutorial we shall route the connection from U1 pin 3 to C1 using two vias. Start by selecting 'Bottom Copper', click left on U1 pin 3 and then click left twice at the point two squares below. Clicking left twice at a point places a via and also swaps the primary and secondary layers. The via will take on a layer range determined by the current strategy's via mode (normal, buried or blind). Having got one via, move across to the point two squares above the target pin, click left twice and complete the route on the underside of the board.

Auto Routing

Using the auto-router is extremely simple - after all the whole point is for the computer to do the work. To see it in action, start by re-loading PPSU.LYT and then invoke the *Auto-Router* command from the *Tools* menu. The default settings will do for this example board so click on OK, sit back and watch. The status display at the bottom of the screen shows what is happening and how things are going. The yellow route is the one being considered for routing. On modern PCs, this board will route to completion in the twinkling of an eye!

Mixed Manual and Auto-Routing

Although the above exercise routed the board entirely automatically you can, if you wish, exert a lot more control over the proceedings. Reload PPSU.LYT and then select the *Main Mode* icon followed by the *Ratsnest* icon. The selector will then display a list of the nets in the design. Select the GND net and click on the selector 'T' toggle. This tags all the connections in the net.

In this mode you can also:

- Tag connections by clicking right over them.
- Tag a connection by clicking right over it.

- Untag all connections by clicking right over nothing.

Given that the auto-router can be set to route all connections or just either the tagged or untagged connections, this gives you total control if you want to mix manual and auto-routing.

Router Strategies

ARES handles the problem of routing different nets with different trace/via widths and so forth in a very sophisticated and convenient manner. Each net in the design is assigned (either implicitly or explicitly) a named strategy which defines how it is to be routed. In practice this means that a net called 12VRAIL can be assigned the POWER strategy on the schematic but that the details of POWER routing can be left undefined until the PCB begins to take shape. At the same time, the need to allocate all sorts of separate properties to each net in the design is avoided.

For a taste of what is possible, invoke the *Set Strategies* command from the *System* menu, select the POWER strategy and then click on the selector toggle. The *Edit Strategy* form allows you to determine how nets assigned to this strategy should be routed.

You will see that there are fields for the trace and via styles, the algorithm to use (i.e. POWER, BUS or SIGNAL), controls for whether diagonals are allowed and whether corners should be optimized (i.e. cut at 45 degrees). Up to 4 passes are allowed per strategy and each pass can use different layers. If the H and V layers for a pass are the same then single sided routing will be attempted.

You can also set individual design rules for each strategy. This can be useful where some tracks must carry high voltages and need to be more widely spaced than those carrying only low voltage signals.

Further details on all these features are to be found under NETLIST MANAGEMENT on page 63 and AUTO-ROUTING on page 83.

Power Planes

As an example of the sophisticated power plane capabilities, we will create a simple ground plane for the PPSU board. Start as follows:

1. Load the file PPSU.LYT.
2. Route it with the autorouter.
3. Select the *Connectivity Highlight* icon.
4. Select the GND=POWER net in the *Object Selector*.
5. Click the 'T' for tag toggle on the *Object Selector*.

6. Click the *Delete* icon.

This sequence of events routes the board, and then removes the ground tracking placed by the router.

Then

7. Select the *Power Plane Generator* command from the *Tools* menu.
8. Select the GND net (this may appear as “GND=POWER”).
9. Click OK.

ARES will then generate the ground plane.

REPORT GENERATION

Aside from Gerber Photoplot and Excellon NC Drill tool information which is really part of the printing/plotting sub-system, ARES can generate two reports detailing how well the current layout meets its specification:

- Connectivity Rule Check - Checks for electrical errors (extra/missing connections)
- Design Rule Check - Checks for physical errors (overly small copper clearances)

Connectivity Rule Checker

This tool establishes which pins are connected to each other (by tracking and vias) and compares this with which pins have been assigned to the same net in the netlist. A report indicating the 'net-groups' within each net is produced and presented in a pop-up window. If you click on the items in the list, the net or nets affected by the error will be highlighted.

If you design from a netlist in the first place, and especially if you use the auto-router, you are unlikely to make mistakes other than missing connections out. However, a zero errors CRC report does give you extra confidence that your design is correct.

Design Rule Check

In PCB layout, (physical) design rules are Pad-Pad, Pad-Trace and Trace-Trace clearance. The auto-router will, of course, obey the correct design rules in choosing where to place routes but the *Design Rule Checker* will also verify that manually placed routes are in order. Where violations occur, these are marked on screen with a red circle and a line showing which objects are too near and a window appears listing all the violations that were found. You can click on the items in list list to highlight each violation in turn; if you double click, ARES will zoom right in to show the violation in detail.

DRC errors will also be flagged if two objects are physically touching but not 'properly' connected - an example of this would be two pads which overlap but do not have a track running between them. This type of error will also show up in the CRC report as a missing connection.

Further details of these reports can be found in REPORT GENERATION on page 99.

GENERAL CONCEPTS

SCREEN LAYOUT

The Menu Bar

The *Menu Bar* runs across the top row of the screen and its main purpose (the selection of commands from the menus) is the same as with any other Windows application. In addition, the title bar area above the menu names is used to display certain prompt messages which indicate when the program has entered a particular editing or processing mode.

The Toolbars

As with other modern Windows applications, ARES provides access to a number of its commands and modes through the use of toolbars. The toolbars can be dragged to any of the four edges of the ARES application window.

Command Toolbars

The tools located along the top of the screen (by default) provide alternative access to the menu commands, as follows:

File/Print commands

Display Commands

Editing Commands

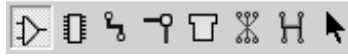
Layout Tools

If you are working on a relatively small monitor, you can hide any or all of the command toolbars using the *Toolbars* command on the *View* menu.

Mode Selector Toolbar

The toolbar located down the left hand edge of the screen select the editor mode, i.e. what happens when you click the mouse on the *Editing Window*.

Placing & Routing



Pad Placement 

2D Graphics 

Note that the mode toolbar cannot be hidden, as its functions are not duplicated on the menus.

Orientation Toolbar

The orientation toolbar displays and controls the rotation and reflection for objects placed onto the layout.

Rotation 

Reflection 

The edit box allows you type a rotation angle in directly; this is the only way to enter non-orthogonal angles.

When an existing object is tagged, the rotation and reflections icons highlight in red to show that they will modify the orientation of an object on the layout. When the icons are not highlighted, they serve to determine the orientation for new objects.

The Editing Window

The *Editing Window* displays the part of the board that you are currently working on.

The contents of the *Editing Window* may be redrawn using the *Redraw* command which also redraws the *Overview Window*.

Panning

You can reposition the work area over different parts of the layout in several ways:

- By clicking left at a point on the *Overview Window* - this re-centres the work area about the marked point.
- By moving the mouse over the *Editing Window*, holding down the SHIFT key, and 'bumping' the pointer against one of its edges. This pans the display in the appropriate direction.
- By pointing in the *Editing Window* and pressing the F5 key. This re-centres the display about the cursor position.

- By using the *Pan* icon on the toolbar.

Zoom In / Zoom Out

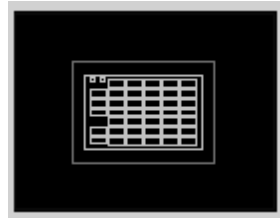
You can magnify or reduce the display of the board using the *Zoom In* and and *Zoom Out* commands which are also invoked by the F6 and F7 shortcut keys. Pressing F8 will display a view of the entire board. You can also use the corresponding icons on the toolbar.

If the keyboard is used to invoke the command, then the *Editing Window* will be redrawn to display a region centred around where the mouse cursor was pointing before. This also provides a way to effect a pan by pressing the zoom key for the current level and simultaneously pointing with the mouse at where you want centre of the new display to be.

Variable Zoom

An arbitrary degree of magnification can be achieved using the Shift-Zoom feature. A given area of the board can be selected to fill the *Editing Window* by holding down the SHIFT key, pressing the left mouse button and dragging out a box around the desired area. The area can be marked on either the *Editing Window* or the *Overview Window*.

The Overview Window



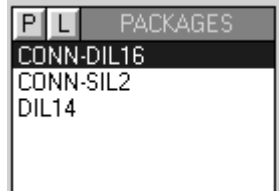
This window shows a simplified representation of the whole drawing. The blue box marks the outline of the work area as set by the *Set Work Area* command whilst the green box indicates the region on view in the *Editing Window*.

Clicking left at a point on the grid re-centres the *Editing Window* around this point.

The *Overview Window* is also used to display previews of objects that are selected for placement. This features helps you to orient (rotate and mirror) an object correctly before placing it on the board.

The width and height of the *Overview Window* can be adjusted by dragging its borders. If you drag the vertical border right over to the other side of the application Window, ARES will re-organize the display so that the *Overview Window* and *Object Selector* are located at the right hand side.

The Object Selector



The Object Selector is used for picking components, packages, pad & trace styles and so on from those that are currently available. It always carries a label indicating what it is listing and

this serves as a prompt additional to the state of the Icon Panel as to which mode is current.

The width and position of the *Object Selector* can be adjusted in conjunction with the width and height of the *Overview Window*, as described above.

The Layer Selector

The *Layer Selector* is used to display and select the current layer or layer set. A layer is a single layer, whereas a layer set is a combination of single layers, such as 'ALL'.

The layer selector can be operated with either the mouse or the keyboard. The keyboard controls are as follows:

SPACE	Selects the next layer in the current layer pair sequence.
PGDN	Selects the next layer down the list.
PGUP	Selects the previous layer up the list
CTRL-PGDN	Selects the last layer in the list.
CTRL-PGUP	Selects the first layer in the list.

Which layers are displayed in the list is determined by the current editor mode, such that PCB objects cannot be placed on inappropriate layers.

Which layers are displayed is controlled with the *Layers* command on the *View* menu whilst the layer pair sequences can be redefined using the *Set Layer Pairs* command on the *System* menu.

This latter command defines the layer that the SPACE key will select for each layer. Thus you can make pairs, triplets quadruplets or whatever.

Co-ordinate Display

The current co-ordinates of the mouse pointer are displayed down at the bottom right of the screen by default. The read-out can be in imperial or metric units and a false origin may be set. Further details are given under CO-ORDINATE SYSTEMS on page 30.

The *X-Cursor* command will display a small or large cross, in addition to the mouse arrow, at the exact location of the current co-ordinates. This is particularly helpful when used in conjunction with the Real Time Snap feature, since it gives you an immediate indication of what ARES thinks you are pointing at.

CO-ORDINATE SYSTEMS

The fundamental unit of linear measurement in ARES is 10nm (ten nanometers). Given 32 bit representation of the co-ordinates this allows a board size up to +/- 10m (ten meters) with

some headroom for calculations. The 10nm unit divides exactly into both 1um (micron) and 0.1 thou (one ten thousandth of an inch) giving exact representations of both imperial and metric dimensions to these resolutions.

Rotational angles are stored to 0.1degree.

The co-ordinate display can be switched to reading metric units by invoking the *Metric* command or by pressing the 'M' key. Restoring imperial mode is achieved by invoking the *Metric* command again.

Dimension Entry Fields

A number of the dialogue forms contain fields that deal with dimensions such as pad style sizes, track widths, design rule clearances and so forth. All these fields are handled in a such a way that both imperial and imperial units can be used at will. The operation of these fields is in accordance with the following rules:

- Values are displayed in units chosen by a heuristic algorithm which detects whether a dimension is a whole number of imperial or metric units. Thus 25.4mm (however originally entered) will always display as 1in, when the field first appears.
- Displayed values always carry a unit. If you delete the existing value, the original displayed unit will be used.
- If you prefer, you can enter a new value with an explicit unit. Valid units are

th	thou	(10e-3 inch)
in	inch	
um	micron	(10e-6 meter)
mm	millimetre	(10e-3 meter)
cm	centimetre	(10e-2 meter)
m	meter	

- Whatever units are chosen, values must be less than +/- 10m, and are held at a resolution of 10nm.

Output Origin

For CADCAM output in particular, but also when working on a board which has to fit into a pre-designed mechanical casing, it is useful to be able to define a reference point on the board which relates to the mechanical design's co-ordinate system. The output origin defines this point within ARES, and is drawn in blue on the *Editing Window* as a target symbol.

- You can re-position the output origin using the *Output Origin* command.

- The output origin does not affect the co-ordinates used in Region files.

False Origin

Although the *Origin* command appears on the *View* menu, it should only be used via its keyboard short cut (key 'O'). Its function is to zero the co-ordinate display at the current mouse position, a feature that is most useful when laying out a complex pattern of pads given a set of dimensions on a drawing of the component.

When a false origin is set, the co-ordinate display changes colour from black to magenta as a reminder.

Cancelling a false origin is done by invoking the *Origin* command a second time.

Grid Dots

A grid of dots is displayed in the *Editing Window* - this can be toggled on and off using the Display-Grid command. The spacing of the dots normally equals the current snap setting (see below), but if you are zoomed out a good way, it will be multiplied by a factor of 2,4,8 etc. The threshold at which this occurs can be adjusted on the *Set Grids* command on the *System* menu.

Snapping Grids

Although the motion of the mouse arrow itself is smooth, you will see that the co-ordinates normally jump in fixed multiples of, for example, 50 thou. The size of these multiples is set by the *Snap* commands and the purpose of this is to facilitate the placement of objects on a fixed grid.

Under normal circumstances, boards tend to be routed on a 50 or 25 thou grid.

Three snap grids for each of imperial and metric modes are available at any one time, though you can redefine their values using the *Set Grids* command on the *System* menu. The current snap options are selectable from the *View* menu or by using keys F2-F4.

The snapping points are computed starting from the current origin as set by the *Set Origin* command. This provides the means to measure out pad placement for a new component, since you can set the origin at the first pin, set one of the snap values to the pin spacing, and then advance the cursor along the pad positions using the cursor keys.

Real Time Snap

As well as snapping to the grid dots, ARES will also snap the cursor to pads and/or tracks which lie off the currently selected snap grid. This process takes place in real time - as the pointer is moved - hence the name Real Time Snap.

The rules for what is snapped to are as follows:

- Pads are always snapped to. Only pads whose layer range overlaps the current layer range are scanned.
- Tracks are snapped to when the either the *Trace* icon or *Via* icon is selected. Only tracks on the current layer are scanned.
- Ratsnest lines are snapped to when the *Ratsnest* icon is selected.
- The range within which the pointer will 'see' an object is either half the current snap spacing or, if grid snap is disabled, the actual bounds of the object concerned.

RTS is extremely useful when routing between components whose pins are on different grids, since it alleviates the need to constantly switch between routing grids. In this context, the *Trace Angle Lock* feature is also very important.

With a large board or a slow computer, RTS can result in a noticeable lag between the motion of the pointer and the cursor. In this case, you may find it helpful to disable RTS except when you really need it using the *Real-Time-Snap* command on *Tools* menu.

OBJECT PLACEMENT

There are five basic types of object in ARES:

Components
Packages
Pads
Graphics
Zones

All are placed in much the same way - select the appropriate object type from the toolbar and then click left on the *Editing Window* to place the object. If you hold the left button down, you can move the object around before actually placing it. Also, when appropriate, the *Rotation* and *Mirror* icons can be used to pre-set the orientation of placed objects.

Placing Components

ARES makes a distinction between *Components* and *Packages* in that a component is a part specified in the netlist whereas a package is not linked to the netlist, has no part ID, and plays no part in netlist driven design verification / modification. Assuming you are using a netlist, 99% of the parts you place would usually be components. You would only use packages for parts which, for whatever reason, were not entered on the schematic. Heatsinks are the most

common example of this. If a netlist is in use, parts which are not specified in it must not be given names.

To place a component

1. Select the *Component* icon from the *Mode Selector* toolbar. In this mode, the *Object Selector* lists all components not yet placed.
2. Choose the component you want to place from the *Object Selector*.
3. Set the *Rotation* and *Mirror* icons to determine the required orientation.

   You can set the rotation to non-orthogonal angles by typing into the text edit-box.
4. Set the *Layer Selector* to determine the required layer. Note that the *Layer Selector* and the *Mirror* icon are interlocked, since a reflected component must be on the underside of the board.
5. Point at the desired position for the component on the board and press the left mouse button.
6. If you hold the button down, you can drag the component around before it is placed; at the same time, ARES will show ratsnest lines and/or a force vector to indicate how the component connects to any others already placed.

   Further, the '+' and '-' keys on the numeric keypad will rotate the component anticlockwise and clockwise respectively in steps of 90 degrees.

Components on the underside of the board

When a component is placed on the solder side of the board, it is effectively 'turned over'. This means that it suffers both a reflection in 2D space and an inversion of the layers of its pads. More specifically:

- The silk screen legend moves from the Top Silk Screen to the Bottom Silk Screen.
- Pads on the Top Layer only, move to the Bottom Layer and vice versa. This caters for underside placement of surface mount components.
- Pads on multiple layers - most commonly normal through hole pads - and padstacks are left alone. On the assumption that a padstack has been defined in a particular way to reflect the usage (signal, ground plane, etc.) of a particular layer, it would be unhelpful to invert pad stacks for solder side placements.

Packages

As discussed above, the term *Package* is used to refer to a placed library part which has pins but is *not* linked to the netlist. In ARES, they are useful in the following circumstances:

- If no netlist is available, or you are doing a simple 'quick and dirty' job, you can lay out a board by placing packages and wiring them up using traces and vias. In doing this, you are using ARES as a computerised light box, rather than as a true EDA tool.
- Sometimes you need to place objects which have pads and graphics but are not actually components in the electrical sense of the word. Heatsinks with mounting holes are a common example of this. These can be placed as unlabeled packages, and as such, will play no part in the netlist driven functions of the design work.
- In order to edit a library-package, it is necessary to place it as a package, tag it and then decompose it using the *Decompose* command. The individual elements can then be edited as required prior to re-making the library-package using the *Make Package* command.

To place a package

1. Select the *Package* icon from the *Mode Selector* toolbar. In this mode, the *Object Selector* lists all the packages picked from the libraries.
2. Assuming that the package you want has not yet been picked from the libraries, click the 'P' for pick toggle on the *Object Selector.*
3. Pick the package or packages you want from the *Library Pick Form.*
4. When you have finished picking packages, close the form in the usual way.
5. Highlight the package you want to place in the *Object Selector.*
6. Proceed from step [3] in *To place a component* above.

📖 Some further information on creating library-packages is available under THE PACKAGE LIBRARY on page 58.

Pads

The main reason for placing lone pads is when defining a new library-package but they are also useful for test points, drilling alignment targets and such like. ARES offers five shapes of pad, namely: Circle, Square, DIL, SMT and Edge Connector.

In addition, ARES supports pad stacks, in which a different pad shape can appear on each layer of the board.

The various pad shapes may be selected from the *Mode Selector* toolbar.

Given the selection of a pad shape, an actual pad style of that shape may be selected from the *Object Selector.* You'll see that all the pad styles have names - this is a unique feature of ARES and has two advantages over other systems:

- There is less chance of confusion or error when choosing a style (provided that you use sensible names!).
- There is no limit to the number of styles you can have.

Pads are placed in much the same way as components and packages. The *Rotation* icon affects the placement of DIL, SMT and Edge Connector pads whilst the *Layer Selector* again determines the layer-set of the pads.

You can edit a pad style by selecting it in the *Object Selector* and then clicking on the toggle. Any changes to the pad dimensions take effect as soon as the display is redrawn - you can force a redraw with the *Redraw* command. This feature is particularly powerful should you find the need to tweak pad styles globally in order, for instance, to change the pad sizes used in the standard package library.

New pad styles may be created using the *New Pad Style* command. See Style Management & DEFAULT.STY on page 55 for more information on styles.

2D Graphics

The 2D drawing facilities are intended, in the main, for adding text and graphics to the silk screen layers though they can also be placed on other layers. Objects placed on the *Edge* layer appear on artwork produced from any of the other layers making it a suitable location to draw the board outline.

There are 8 types of 2D graphic object, namely:

Line
Box
Circle
Arc
Path
Text
Symbol
Marker

The corresponding icons are contained in the *Mode Selector* toolbar.

- Lines are drawn by clicking left at each end.
- Boxes and circles are dragged out using the left button.
- Arcs are placed dragged out into a quadrant using the left button. They can be further adjusted by tagging and dragging the four handles. ARES arcs are actually Bezier curves.

- Paths are closed figures with boundaries defined as multiple line and arc segments. Click left at each required vertex and hold the CTRL key down to place arc sections. Further adjustments may be made by tagging the path and moving the 'handles' about.

  We strongly recommend that paths are not used to create copper fills as they are not-recognized by the connectivity database. Use zones instead.

- Text is placed by clicking at the bottom left corner of the text region. A form then appears which allows input of the text itself and also allows control of the text dimensions. Text orientation can be set using the Rotation and Mirror Icons.

- Selecting the *Symbol* icon displays symbols in the *Object Selector.* Symbol selection is identical to package selection as described earlier. More information on the symbol library facility is contained in THE SYMBOL LIBRARY on page 62.

- Markers are used in the creation of library parts to define the origin, and also the position for component labels.

Zones

Zones represent the means by which ARES handles power planes. In this section we will just describe briefly how to place a simple power plane; further documentation on power planes is given under POWER PLANES on page 91.

To place a simple power plane

1. Select the *Zone* icon from the *Mode Selector* toolbar.
2. Select the required boundary trace style for the zone from the *Object Selector*.
3. To define the zone boundary, either:
   - Drag out a box with the left mouse button held down.
   - Click at several points to form a polygonal boundary.

   If a zone boundary is placed over existing tracking, it will pick up its net from that tracking. This makes it quite easy to place lots of small zones in order to 'fill in' between tracks.
4. Select the required net and hatch/fill style from the *Edit Zone* dialogue form, then click OK.
5. ARES will generate the ground plane.

OBJECT EDITING

Due to is its object oriented design, all objects in ARES are treated in much the same way and we can discuss generally their dragging, copying, movement, rotation, deletion and editing. You should note that traces and vias are not regarded as objects for this purpose; their editing is dealt with further under Manual Routing on page 21. However, there are operations where traces attached to objects are affected by operations performed on the object - such cases are discussed here.

Tagging a Single Object

Any object may be tagged by pointing at it and clicking right. This action highlights the object and selects it for further editing operations.

Objects are regarded as occupying a layer-set, and will be found if the current layer-set as indicated by the *Layer Selector* overlaps that occupied by the object. This is necessary since two objects may occupy the same area on different layers, as in the case of component and solder side pads being placed for an edge connector. However, it does mean that you must remember to check/set the *Layer Selector* appropriately before tagging.

- When you tag an object, the layout editor will change to the mode that is used to place that type of object. This is done so that the *Layer Selector* can then offer the correct set of layers for that object.
- Component outlines occupy the silk screen layer, but component pads occupy the copper layers.

Tagging a Group of Objects

A group of tagged objects may be assembled by either tagging each one in turn as described above, or by dragging a box around them using the right mouse button. Only objects wholly enclosed in the box will be tagged and, unlike tagging a single object, all layers are scanned.

The box is called a tag-box, and will remain on the screen until all the objects are untagged as described in the next section. The tag-box is of particular significance as regards the block editing commands and their effect on any routing within it.

Only one tag-box can exist at any one time - drawing a new one will delete the old one, and untag the objects within it.

Untagging All Objects

To untag all the objects, you need simply to point where there is no object and click right. This will also remove the tag-box if one has been defined.

Performing an explicit *Redraw* command (key 'R') will also untag all objects - this can be handy if the board is very densely packed and there is nowhere free of objects on the screen.

The Tag Filter

The tag filter determines the type(s) of objects that are tagged when a tag box is drawn. This makes it possible to tag only components, only tracks and so forth. In addition, you can select the layers upon which objects are tagged.

In conjunction with the *Block Delete* command, this makes it possible to delete particular objects on particular layers.

If you change the tag-filter whilst a tag-box is present, the effect of the tag-box will be re-applied using the new filter settings.

The tag-filter dialogue form can be displayed using the *Tag-Filter* command on the *Edit* menu, or by pressing CTRL-X.

Deleting an Object

Any tagged object can be deleted by pointing at it and clicking right. Any traces connected to its pads will remain in place, allowing a different pad or package to be placed over them.

Dragging an Object

You can drag (i.e. re-position) any tagged object by pointing at it and then dragging the mouse with the left button depressed.

Any traces attached to the object will be rubber banded (stretched) unless both ends connect to it, in which case the route will be moved wholesale.

Editing an Object

Some objects, especially packages, components and graphics text have properties that can be edited with a dialogue form. There are two ways to do this:

- A tagged object can be edited by pointing at it and then clicking left as if to drag it but without moving it.
- If you have several objects to edit it may be more convenient to select the *Instant Edit* icon. Clicking left on any editable object will then bring up its dialogue form.

Highlighting a Component by Name

When the *Instant Edit* icon is selected, the *Object Selector* lists all the components in the design. Clicking on the selector toggle tags the selected component causing it to be highlighted, and displayed in the centre of the screen.

Alternatively, you can locate a component using the *Goto Component* command on the *View* menu.

ROUTE PLACEMENT & EDITING

ARES provides a variety of methods for the manual placement and editing of tracking and vias. In particular, existing routes may be modified by any of the following methods:

- Over placement - place a new section of route over an old one and the new will replace the old. This provides an easy way to change the thickness of sections of tracking.
- Segment dragging - the path of a tagged route can be easily changed by dragging its segments. Neighbouring segments are automatically stretched so as to maintain an orthogonal path. This is a very intuitive method of working for users accustomed to other Windows drawing packages.
- Re-routing - if a new route is placed such that it replaces part of the path of a tagged route, then the bypass section of the existing route is automatically removed. This method is the most flexible, especially if the new section needs to change layers or have a significantly different number of segments. It is particularly effective when trying to clear space on the board towards the end of the routing process.

In all cases, ARES will validate newly placed or modified route sections against the connectivity data specified in the netlist. Sections of tracking which violate the connectivity rules are marked as 'dirty' and will flash in yellow to indicate that they are in illegal positions.

Trace Placement - No Netlist Loaded

Placing a new trace on the primary layer is done by first selecting the *Trace* icon and the required trace style from the *Object Selector* and then clicking left at each node along the required path before clicking right to finish.

Clicking left twice at the same location will cause a via to be placed and the next layer in the layer pair sequence to be selected. The type of via used can be changed by selecting the *Via* icon and choosing one of the via styles from the *Object Selector.* You can also place, replace, tag, drag and delete vias manually in this mode. The mouse buttons work in the same was as for ordinary objects.

You can change layers whilst routing using the *Layer Selector* keyboard controls: PGUP, PGDN, CTRL-PGUP, CTRL-PGDN.

Should you start to enter a trace, and then decide you wish to abort it entirely, simply press the ESC key.

Trace Placement - Netlist Loaded

If a netlist is loaded and you commence routing from a pin assigned to a given net, ARES will:

- Indicate the net name of the given net.
- Indicate to which other pins the first one connects to.
- Select the appropriate trace and via styles for the given net according to the strategy to which it is assigned. This function can be disabled by toggling the *Auto Trace Selection* option on the *Tools* menu. This can be useful if the odd part of a net needs to be a different thickness. See ROUTING STRATEGIES on page 71 for more information on strategies.

You can then place route segments and vias as described in the previous section. Assuming that you take the route to one of the indicated pins, ARES will automatically terminate routing mode when you click left on it.

Should you wish to connect to a point other than a selected pin, this is possible by clicking right to terminate routing as described in Trace Placement - No Netlist Loaded on page 42. ARES will still detect any ratsnest lines that have been connected and will remove them from the display though this takes a little longer than if you route directly from pin to pin. In fact, ARES will remove connected ratsnest lines even if neither end of the placed route starts on a pin - *Advanced Netlist Management* can check a whole net's connection status very rapidly.

ARES will also check whether the placed route connects to anything belonging to another net. This could be a pin, via, or part of another trace. This prevents major blunders where a

route is placed across several others which you failed to see. If such a violation is detected ARES displays an error message and the tracking will be marked as dirty. Dirty tracking flashes between bright yellow and its normal colours, except when it is tagged.

When *Auto Track Necking* is enabled, ARES also checks for physical design rule violations. In the first instance, it will neck the track to avoid them, but if there is still a violation then it displays a warning message and beeps. The route is, however, placed. This process can take some time, and if you find that trace placement is becoming unduly tardy, you can try disabling ATN from the *Tools* menu.

Curved Track Segments

ARES fully supports curved tracks and supports them as true arcs for output purposes. The segments also play a full role in connectivity and design rule checking.

To place a curved segment, first commence routing as usual (e.g. click left on a pad) but then, before moving the mouse, hold the CTRL key down. Move the mouse roughly to trace the arc, click left to fix the arc endpoint and then release the CTRL key. You can place a curved segment at any stage of laying a route - it does not have to be the first or last segment.

Two important points about curved tracks:

- Because ARES renders a curved track as an arc, the mechanism that 'backs off' a trace end from any pad to which it connects does not operate for curved track segments. It follows that if curved segment connects directly to a pad, the centre hole of the pad may be obscured.
- It is possible, using the route editing facilities described in the following sections, to break a curved segment into sub-sections. ARES will allow this but the sub-sections will then be treated as runs of short linear segments. As such, they may not render satisfactorily on pen plotters.

Auto Track Necking

In many cases, the reason for necking down a track is so that it can pass between two pads or other obstacles without violating the design rules. The *Auto Track Necking* feature allows ARES to do this for you.

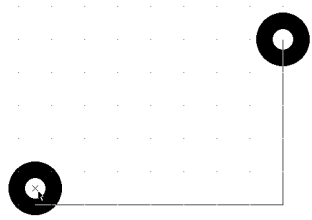
The function is controlled by the *Set Design Rules* command on the *System* menu. The dialogue form allows you to enter the clearances for pad-pad, pad-trace and trace-trace and also the trace style to neck to. The default neck style is T10 - a 10 thou track.

To disable ATN totally, use the toggle command on the *Tools* menu. You may find this speeds up route placement considerably, since the analysis involved in performing ATN can be quite complex, even if the end result is that the route is OK.

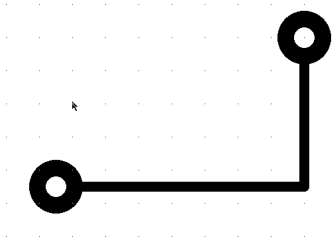
Trace Angle Lock

Many people like to keep all tracking running at either 90 or 45 degrees as this gives a tidy, professional look to a layout. To make this easier, ARES has a *Trace Angle Lock* command which restricts route segments to 90 or 45 degree angles only.

TAL also has a more subtle effect when combined with *Real Time Snap*. Consider the following routing problem... You are trying to route from a pad on the grid to one that is off it. You have moved down, clicked left and moved the pointer over to the destination pad. *Real Time Snap* has detected the pad, and the cursor has locked onto it. However, because your first click left was on a grid dot, the horizontal part of the track doesn't line up with the pad centre. The diagram below shows the position at this stage:



When *Trace Angle Lock* is enabled, clicking over a pad or track in this circumstance will activate *Trace Angle Fixup*. The result is that the previous node - in this case the route corner, is moved so that trace angle lock rules can be maintained. The result is a tidy connection as shown below overleaf.



Trace Angle Lock & *Fixup* are active by default. You can disable or re-enable them using the *Trace Angle Lock* command on the *Tools* menu.

Tagging a Route

To re-route, delete or copy a section of tracking requires that you first tag the route containing it.

To tag a route

1. Select the *Trace* icon from the *Mode Selector* toolbar.
2. Set the *Layer Selector* to the layer on which the trace lies.
3. Point at a part of the route on the selected layer, and click right.

If you click on a via, or pin, or a point at which three or tracks meet, all the tracking originating from that point will be selected.

Changing the Width of a Route

ARES provides two methods of changing track width. The first relies simply on the general ability of ARES to cope with new objects being placed over old.

To change the width of tracking by overplacement

1. Ensure that there is no tagged route, and that *Auto Trace Selection* is disabled on the *Tools* menu.
2. Select the new trace style in the *Object Selector*.
3. Place the new tracking directly over the old. ARES will work out that there is overplacement, and will replace the old tracking with the new in the database.

The second follows the principle of allowing various editing operations to be performed on the currently tagged object.

To change the width of a tagged section of tracking

1. Tag the required section of tracking as described under Tagging a Route on page 45. The *Object Selector* will display the current trace width. If the tagged route has several widths, then the width at the point under the cursor is displayed.
2. Click right on the tracking to display the context menu.
3. Select a new trace style from the menu.

Changing the Layer of a Route

To change the layer of a tagged section of tracking

1. Tag the required section of tracking as described under Tagging a Route on page 45.
2. Click right on the tracking to display the context menu.
3. Select a new layer for the tracking from the menu.

Modifying a Route

To rubber band a section of tracking

1. Tag the route as described under Tagging a Route on page 45.
2. Drag the segments and/or corners of the route using the left mouse button.

If you drag horizontal or vertical segments, they will move vertically or horizontally respectively with adjacent segments between stretched to accommodate this. On the other hand, if you drag on the route corners, then the adjacent segments will simply stretch diagonally. Finally, if you drag in the middle of diagonal segments, a new corner will be created.

To re-route a section of tracking

1. Tag the route as described under Tagging a Route on page 45.
2. Place a new section of tracking starting and finishing on the tagged route. ARES will then remove the section of tracking that the new route shorts out.

Note that vias which are made redundant (i.e. secondary vias) will also be removed, along with the shorted out section.

Copying a Route

To duplicate a route

1. Tag the route as described under Tagging a Route on page 45.
2. Click right on the tracking to display the context menu.
3. Select the *Copy* command.
4. Click left at each position for the duplicate routes; click right to finish.

This feature provides an excellent way do memory buses and other repeated patterns of tracking.

Deleting a Route

To delete a route

1. Tag the route as described under Tagging a Route on page 45.
2. Click right to display the context menu.
3. Select the *Delete* command.

Any interconnections specified in the netlist that were formed by the deleted tracking will automatically appear as ratsnest lines.

Note that you can also delete *all* tagged tracking by using the *Block Delete* icon. This can be quicker than the above procedure if no other tracking is tagged.

Tidying the Routes

Extensive route editing can lead to the creation of unwanted nodes between tracks segments that join at 180 degrees. These nodes waste memory and also degrade the quality of plotted output. One of the processes carried out by the *Tidy* command is to scan for and remove such nodes.

The *Tidy* command may be found on the *Edit* menu.

BLOCK EDITING COMMANDS

Four block operations are provided: *Copy*, *Move*, *Rotate* and *Delete* and each operates on the currently tagged objects and the traces & vias within the tag-box. If there is no tag-box, then a default one is assumed comprising the total area occupied by the currently tagged objects. If there are no tagged objects, nothing at all will happen.

Block Copy

After creating a tag-box as described in Tagging a Group of Objects on page 39, clicking on the *Copy* icon will cause a second box to appear over the tag-box which you can then re-position using the mouse. Clicking left will cause a copy to be made of the objects within the tag-box whilst clicking right will abort the operation.

Traces are only copied if both ends of them are inside the tag-box.

Beware that copying components will generally leave the netlist management out of kilter as there will be multiple placements of the same component references. If, having routed a module that is to be repeated several times, your intention is to copy it out to save re-routing,

you should use the *Auto Name Generator* to renumber the components in the copied section(s) and then re-load the netlist.

Block Move

After creating a tag-box as described in Tagging a Group of Objects on page 39, clicking on the *Move* icon will cause a second box to appear over the tag-box which you can then re-position using the mouse. Clicking left will cause the tag-box and its contents inside it to be moved whilst clicking right will abort the operation.

Trace segments with one end only inside the tag-box will be rubber-banded.

Block Rotate

After creating a tag-box as described in Tagging a Group of Objects on page 39, clicking on the *Rotate* icon will allow you to rotate and/or reflect the block by any angle.

- Trace segments with one end only inside the tag-box will be rubber-banded although this generally leaves something of a mess - it's best to rotate only completely enclosed sections of a layout.
- Rotation of whole groups of components to non-orthogonal angles (even 45°) creates large numbers of off grid pads. Routing such boards can be very tricky, even with features such as real time snap and trace angle fixup.

  The autorouter, being grid based, will also perform poorly under such circumstances.

Block Delete

After creating a tag-box as described in Tagging a Group of Objects on page 39, clicking on the *Delete* icon will delete all tagged objects.

Remember that you can tag objects and tracking in a number of ways:

- Using the right mouse button to select objects individually.
- Using the tag-box and tag-filter to select specified objects within an area.
- Using connectivity highlight to select tracking on particular nets, or running from particular pins - see page 66 for more information.

FILING COMMANDS

ARES makes use of the following file types:

Layout Files	(.LYT)
Backup Files	(.LBK)
Region Files	(.RGN)
Library Files	(.LIB)
Netlist Files	(.SDF)

Layout files contain all the information about one board, and have the file extension 'LYT'. They contain within them copies of all packages and styles used on the board, so the complete design can be given to someone else solely by giving them a copy of the layout file. Backup copies of layout files made when saving over an existing file are given the extension "LBK".

A region of a board can be exported to a region file and subsequently read into another layout. Region files have the extension 'RGN' and are read and written by the *Import* and *Export* commands on the *File* menu. They are analogous to section files in ISIS. ARES region files are in an ASCII format and for the advanced user, this opens the possibility of manually editing the layout database, or else writing utility software to perform specialist operations on it. Import of data from other PCB design packages is also a possibility. Contact our technical support department or look on our Web Site if you want a copy of the format specification.

Package and symbol libraries have the extension 'LIB'. 6 libraries are supplied:

PACKAGE.LIB	Standard through hole footprints
SMTDISC.LIB	Discrete SMT footprints
SMTCHIP.LIB	IC SMT footprints
USERPKG.LIB	For your own footprints
SYSTEM.LIB	Standard symbols
USERSYM.LIB	For your own symbols.

Full details of library facilities are given under LIBRARY FACILITIES on page 57.

Native format netlist files have the extension 'SDF' standing for Schematic Description Format. Netlisting features are covered in detail in NETLIST MANAGEMENT on page 63.

Starting a New Layout

To start over on an empty work area, use the *New Layout* command on the *File* menu. The layout filestem is set to 'UNTITLED' until such time as you invoke the *Save As* command.

Loading a Layout

A layout may be loaded in various ways:

- From the command line as in:

  ```
  ARES my_board
  ```

- By using the *Load Layout* option once ARES is running.
- By double clicking the file in *Windows Explorer*.

Saving a Layout

You can save a board when quitting ARES via the *Exit* option, or at any other time, using the *Save Layout* command.

- In both cases it is saved to the same file from which it was loaded; the old file being re-named. On the first occasion that the file is saved, it is renamed to "Last Loaded <filename>.LBK" whilst on subsequent occasions it is renamed to "Backup of <filename>.LBK". This means that a copy of the file as it was at the start of an editing session is always preserved.
- If no filename was given at load time or the *New* command was issued, the name 'UNTITLED.LYT' is used.

The *Save As* command allows you to save the layout to a file with a different name.

Import / Export

The *Export Region* command creates a region file out of all currently tagged objects. This file can then be read into another layout using the *Import Region* command. After you have chosen the region file, operation is identical to the block-copy function.

Some difficulty may arise if an attempt is made to import a region saved from a layout in which the pad/trace/via styles in use are different from the current one.

A common difficulty arises from exporting a region file from a design where you have made local changes to pad/trace styles. Region files do not save this information and so when you import the region file all pad/trace styles will revert back to their default values. The solution of course is to create new styles in the first instance reflecting the modifications that you want to make. You can then safely create your board and export/import it via region files into other designs.

Region files may also be loaded using the *Load* command, and this is done to facilitate the loading of data produced by the external converters DXFCVT and GERBIT.

Note that region files are not good way to panelize multiple boards for manufacture because netlist information is lost and imported ground planes will lose connectivity as a result. The *Gerber Viewer* provides a much better way to perform panelization.

Auto-Save

ARES has an auto-save facility whereby your work will be backed up automatically at regular intervals. The interval defaults to 15 minutes and can be changed using the *Set Environment* command on the S*ystem* menu.

Should ARES terminate unexpectedly, the next time it starts it will look for the auto-save file from the previous session and prompt to reload it.

The auto-save file will be stored in your Windows temporary directory, normally addressed by the environment variable TEMP. PROTEUS auto-save files have the extension 'ASV'.

This feature is not an excuse for not saving and backing up regularly - there is no guarantee that the auto-save file will be recovered in all cases.

PAD & TRACE STYLES

ARES features a sophisticated and flexible system for defining shapes and sizes for pads and tracks. Each type of pad and track is given a style name, and then this name is then used to refer to the style whenever it is used. The dimensions and other characteristics for each style are held in a layout-global table, and this makes it possible to change the dimensions of all the pads or tracks in a particular style extremely easily. At the same time, the use of names means that there is no serious limit on the number of pad and trace types that you can have per layout.

Pad Styles

ARES supports 7 types of pad, namely:

Circular PTH
Square PTH
DIL PTH
Circular SMT
Rectangular SMT
Polygonal SMT
Edge Connector

and the list of styles defined for each type displays in the *Object Selector* whenever the appropriate pad icon is highlighted.

When one of these lists is displayed, you can edit an existing pad style by highlighting it in the selector and then clicking on the selector toggle (the 'E' symbol). All fields on the resulting dialogue form have context sensitive help associated with them.

Polygonal Pads

ARES supports a user defined, polygonal pad style in order to cater for parts requiring unusual pad shapes.

To define a new polygonal pad style

1. Using the 2D graphics *Line*, *Arc* or *Path* icons, draw a closed path to define the shape of the new pad style. It does not matter which layer you draw on for this purpose.
2. Place an *Origin* marker within the closed path to specify the origin of the pad style. The origin is used as the connection point for tracking, and also serves as the point of alignment if the pad style is used within a pad stack.

   Note that the origin must lie within the closed boundary i.e. within the copper area of the pad.
3. Tag the 2D graphics and origin marker by drawing a tag-box with the right mouse button.
4. Invoke the *New Pad Style* command from the *Edit* menu.
5. Enter a name for the new pad style.
6. Set the *Polygonal* pad style type under *SMT*.
7. Click OK.
8. Specify a *Guard Gap* for the pad style, if required.
9. If you wish to create the style only in the current layout, and not in DEFAULT.STY, then select *Local Edit* instead of *Update Defaults.*
10. Click OK to create the new pad style.

Note that there is no method to change the outline of a polygonal pad once it has been made - you must re-define it from scratch.

Pad Stacks

The ordinary through hole pad styles can be placed on a single layer or on all the copper layers, but this does not provide a very convenient or satisfactory method for defining component pins which have different pad shapes on different layers. In particular, the ARES

connectivity system does not recognize connectivity between single layer pads on different layers.

Instead, ARES provides the facility to define *Pad-Stacks.* There are several points about a pad stack:

- A pad stack can have a circular hole, a rectangular slot or be surface only. The latter type of pad stack is used where you wish to defined explicit styles for the solder resist and/or solder paste mask apertures.
- For each layer, you can assign a different pad style, or no pad.
- For obvious reasons the hole or slot diameter of a pad stack is the same for all layers.
- You must use a pad stack to create a pad shape with a slotted hole. You cannot specifye a slotted hole in an ordinary pad style.

To define a new pad stack

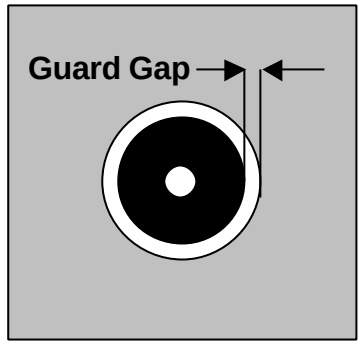
1. Select the *Create New Pad Stack* command from the *Edit* menu.
2. Enter a name for the pad stack, and a default pad style. All layers will start out with this style of pad.
3. Click OK to proceed to the *Edit Pad Stack* dialogue.
4. On this form, you can adjust the pad-layer assignments, and also assign the layer-global drill mark and drill hole/slot size.

To edit an existing pad stack

1. Select the *Pad Mode* and *Pad Stack* icons.
2. Select the pad stack you want to edit from the *Object Selector.*
3. Click the 'E' for edit toggle on the *Object Selector* to bring up the *Edit Pad Stack* dialogue form.

Pad Styles and the Solder Resist Layer

The solder resist artworks are generated automatically from the pad shapes present on the top and bottom of the board. Each pads' radius is enlarged by the amount specified in the *Guard Gap* field of its style, as shown overleaf:



Note that in the case of a pad stack, you can define the pad style used for the top and bottom resist layers explicitly, using the selectors in the *Edit Pad Stack* dialogue form. Note that the guard gap of this style is still applied in determining the final size of the aperture.

You can also specify a pad stack which has no solder resist - i.e. the pad will be covered by the resist coating.

Pad Styles and the Solder Paste Mask Layer

The solder paste mask artworks are also generated automatically from the SMT pads present on the board. The paste mask aperture is exactly the same as that used for the pad itself. Once again, in the case of pad stacks, different pad style may be specified explicitly for this purpose.

Trace Styles

Currently there is just one trace style, though in the future we may implement additional types for micro-strip etc. The available trace styles are displayed when the *Trace* icon is selected. You can then edit any the styles by highlighting its name and clicking on the selector toggle.

Trace styles have a single attribute which defines their width.

New trace styles can be created using the *New Trace Styles* command on the *Edit* menu. There is no way to delete trace styles once they have been created.

Via Styles

Via styles are much the same as pad styles except that only circular or square vias are permitted. The list of via styles is displayed when the *Via* icon is selected.

New via styles can be created using the *New Via Styles* command on the *Edit* menu. There is no way to delete via styles once they have been created.

Style Management & DEFAULT.STY

When you start a new layout, an initial set of styles appears and these come from a system file called DEFAULT.STY which is kept in the "Library" directory of your Proteus installation. This file contains a selection of common pad styles that we feel you may find useful.

When you create or edit pad style you will see that the dialogue form contains a *Changes* box. If you select *Local Edit* then only the copy of the style maintained within the layout will be modified. If you select *Update Defaults* then DEFAULT.STY will also be updated.

Package library parts contain there own copies of styles used by their pads. When the package is brought into a layout for the first time, the style definitions will be taken from the library part if they do not already exist within the layout.

LIBRARY FACILITIES

GENERAL POINTS ABOUT LIBRARIES

As supplied there are 2 symbol libraries (which are shared with ISIS):

SYSTEM.LIB
USERSYM.LIB

and 6 package libraries:

PACKAGE.LIB
CONNECTORS.LIB
SMTCHIP.LIB
SMTDISC.LIB
SMTBGA.LIB
USERPKG.LIB

Detailed information - including pictures of all the footprints - can be found in the file LIBRARY.PDF. You will need to install the Adobe Acrobat reader if you have not already done so.

You can, of course, create further libraries of your own using the Library Manager

Library Discipline

USERSYM.LIB and USERPKG.LIB are set to read/write; the rest are set to read-only. The idea is that you should only add things to USERSYM.LIB (new symbols) and USERPKG.LIB (new packages). This means that we can supply you with updates to the parts we have defined without risk of overwriting similarly named objects in your own libraries.

If you must change things in the read-only libraries, you can set them to read/write using the *Properties* command in *File Manager* under Windows or with the DOS ATTRIB program. The command

```
ATTRIB filename -R
```

sets a file to read/write whilst

```
ATTRIB filename +R
```

sets a file to read-only. Wildcards may also be used.

The *Library Manager* also includes an option to toggle the read/write status of libraries.

The Pick Command

The *Pick* command serves as an alternative to the library browser, primarily for when you know the name of the package or symbol you are looking for. You can search for an exact match, or else use the various pattern matching options to search for a part when you have a rough idea of its name.

If you pick a package or symbol that is already loaded into the layout, ARES will update it from the libraries on disk.

Note that where are there are two or more packages or symbols with the same name spread across several libraries, the *Pick* command will load newest one - i.e. the one most recently created. This is generally a helpful behaviour in that if you have re-defined one of our parts and put it in your USERPKG.LIB, your version will be newer than ours.

The Tidy Command

The *Tidy* command on the *Edit* menu performs several functions, one of which is to remove all packages and symbols from the selectors that are not actually in use on the layout.

This saves memory and also simplifies picking the ones you are actually using.

THE PACKAGE LIBRARY

A package is a collection of pads and silk screen graphics used to site a component on the board. A library of many common package styles is provided and this will save you a great deal of time compared with manual methods of PCB layout. However there will be occasions when you will have to create your own packages, and the steps to follow are set out overleaf.

Making a Package

To make a new library package:

1. Select the appropriate pad shape icons and place the pads of the package as required. Use the *Layer Selector* to place the pads on the appropriate layers. See below for more information.
2. By default, ARES will give the pins numbers from '1' upwards, numbering them in the order they were placed. If this is not what is required, you must either renumber them with the *Auto Name Generator*, or else edit them manually. See below for more information.
2. Select the appropriate 2D graphics icons and place silk screen graphics as required. A package can have legends on both the component side and solder side screens; these will be swapped if the package is placed with the mirror icon active.
3. Tag all the objects by dragging a box round them.
4. Invoke the *Make Package* command on the *Edit* menu, type in a name and select the library to store it in. The package will then be stored in the chosen library and added to the contents of the *Object Selector*, ready for immediate placement.
5. The reference or 'anchor point' for a package can be determined by placing the ORIGIN marker. If there is no ORIGIN marker, the reference will be the centre of first pin placed. Markers are accessed by selecting the *Marker* icon.
6. It is also possible to define non-default positions for the reference and/or value labels. To do this, place the REFERENCE or VALUE markers at the required position(s).

Choosing the Correct Layers for the Pins

Ordinary pad styles can be placed on single copper layers, or on all the copper layers as defined by the current setting of the *Layer Selector.*

- For ordinary through hole parts, you should place the pads on all the layers.
- For SMT parts, you should place the pads on the top layer only. If such a part is placed on the Solder Side, the pads will be automatically transferred to the bottom copper layer.
- For doubled sided edge connectors, you should place separate fingers on both the top and bottom copper layers.
- If you need different shaped pads on different layers for a through hole pin, you should use pad stacks as defined under *Pad Stacks* on page 52.

The Replicate Command

ARES provides a command to replicate tagged objects at a specified spacings in X and/or Y. This can be extremely useful when creating new packages as you can use it to lay out rows of pads with the spacings entered directly from the keyboard.

To place a row of pads by replication

1. Place a single pad of the require style at the first position of the row.
2. Ensure no other objects are tagged, and then tag the pad with the right mouse button.
3. Invoke the *Replicate* command from the *Edit* menu.
4. Enter the required X-Step (for a row) or Y-Step (for a column). Remember to enter a dimension suffix for these values - e.g. 20th or 1.5mm.
5. Enter the number of copies - this should be one less that the total number of pads required as you have placed one already.
6. Click OK - ARES will replicate the original pad as specified.

☞ You can create grids of pads by replicating rows.

Pin Numbering

When a netlist is loaded, ARES attempts to match pin numbers appearing in the netlist with the pin numbers assigned to the pins of the component packages. Where a match cannot be found, this is an error and either the ISIS library part or the ARES library part must be changed.

When you are defining an ARES library part, there are three ways to control the pin numbering:

- By default, the pins will be numbering from '1' upwards in the order that they were placed.
- If you invoke the *Auto Name Generator* command on the *Tools* menu, you can then click on the pads in any order you like to assign incrementing pin numbers.
- If you select the *Instant Edit* icon, you can click on each pin and key in the pin numbers manually.

Note that trying to use a mix of default and manual pin numbering is likely to lead you to confusion and mistakes.

Other Points

ARES normally chooses locations for the part ID and value; these labels can always be moved after a part has been placed by tagging it, pointing at the label you wish to move, and dragging with the left mouse button depressed. The locations chosen will generally be good enough as is, unless many components are closely packed together.

However, if you wish to force the label to a particular default position, you can use the REFERENCE and VALUE markers, as described above.

The size of the labels is set by the *Set Template* command on the *System* menu. This will apply globally except for labels whose size has been manually edited.

Editing a Package

To edit an existing library package,

1. Select the *Main Mode* and *Package* icons.
2. Pick the package to be edited from the library (using the toggle on the *Object Selector*) and place it on the work area.
3. Tag the package and then invoke the *Decompose* command from the *Edit* menu. This will break the package into its constituent pads and 2D graphics and also place an ORIGIN marker at the location of the part's placement reference.
4. Edit the pads and graphics as required.
5. When you have finished you can use the *Make Package* command to re-store the package either with the same name as before, or as a new part.

Of course, if the original package came from PACKAGE, which is normally read only, you will have to store it back to USERPKG. We recommend against using *Library Manager* to re-copy the part to PACKAGE as this can cause problems when we issue a new PACKAGE.LIB - how are you going to sort out which parts have been edited by you in order to preserve them?

Note that the pins will carry the pin numbers from the original package so decomposing a DIL14, sticking two pins on the end, and then re-making is not the way to make a DIL16 unless you are prepared the to re-number all the pins manually.

THE SYMBOL LIBRARY

A symbol is a group of graphic objects which are treated as a single object. For instance, using 3 lines and 2 arcs you can form an AND gate symbol.

A symbol may be created by tagging the objects you wish to form it and then invoking the *Make Symbol* command on the *Edit* menu. Layers are ignored for this purpose. A form will appear allowing you to name the symbol which will then be stored in the Symbol Library and made available for immediate placement from the *Object Selector.* If there is already a symbol with the same name, it will be replaced.

ARES quite happily allows a symbol to contain other symbols and/or other graphic objects. This allows you to make, for instance, a NAND gate out of the previously defined AND gate plus a circle.

As with packages, the placement reference for a symbol may be determined by placing an ORIGIN symbol.

ISIS and ARES symbols are essentially exchangeable, and all symbol libraries have the type 'PROTEUS SYMBOL LIBRARY'. However, ARES does not support all of the drawing appearance features of ISIS, so correct rendering of ISIS symbols with complex fill patterns is not guaranteed, especially for CADCAM output where we can say with some certainty that *it will not work*!

NETLIST MANAGEMENT

NETLIST FEATURES

A netlist produced using ISIS or some other schematic capture package contains as a minimum, a list of the components used in the design and a specification of how their pins are to be connected. The ability of ARES to use this kind of information distinguishes it from budget packages which provide what amounts to a computerized version of rub-down transfers and drafting film.

Our own netlist format is called SDF which stands for Schematic Description Format. As well as component name and connectivity information, an SDF file can also detail information concerning the package to be used for each component and the routing strategy to be used for each net. As a result, it is possible to completely specify a PCB with an SDF file (and therefore from within ISIS), save for the physical positions of the components and the detailed routing of the interconnections.

The Load Netlist Command

The *Netlist Loader* is the means by which you import the data held in an SDF file into the current PCB layout.

A netlist can be loaded with the work area completely empty, with a set of placed and pre-annotated components on the work area or, perhaps in order to effect a modification to an existing design, with a completely placed and routed layout loaded.

Using the Netlist Loader on an Empty Layout

With no components placed, all parts specified in the netlist are simply noted and displayed in the *Object Selector* ready for placement as described in Placing Components on page 33.

The Netlist Loader & Existing Components

Where components or annotated packages are already on the board, any that have been given the same names as those in the netlist are linked into the netlist database. They will then play a full part in subsequent netlisting activities such as ratsnest compilation. Components or annotated packages which are not specified in the netlist are tagged. This is a half way house between ignoring them and deleting them - one click on the *Delete* icon will remove them if they really are superfluous.

ARES checks to see if the library package specified for a component in the netlist matches that of the placed component on the layout. If the specified package differs, one of two things will happen:

- If the new package has the same number of pins as the existing one, then a replacement will occur such that the new package is placed with its pin 1 lying over the old package's pin 1. The component is tagged to highlight the fact that something has happened to it.
- If the new package has a different number of pins, the component is removed from the layout and added to the *Object Selector* for manual replacement.

Unannotated packages are ignored by the netlist loader. This makes them a suitable way to place footprints for parts which, for whatever reason, you do not want subject to full rigors of netlist based design. You should also note that because their pins are not specified in the netlist, they can be connected to anything without CRC violations occurring.

The Netlist Loader & Existing Tracking

In the case where there is tracking as well as components on the board, ARES checks the connections made by the tracking to see whether they agree with the netlist. Where tracking is found that joins two nets (presumably as a result of a change to the schematic) the traces and vias involved are assigned to the VOID net.

VOID tracking and vias appear in flashing yellow, are ignored by the connectivity scanner and are not printed/plotted. They can be removed in one of two ways:

- By invoking the *Tidy* command on the *Edit* menu. This does other things too and can take some time on a large board.
- By selecting the *Connectivity Highlight* icon, then selecting the VOID net in the *Object Selector* and finally clicking its toggle to highlight the VOID tracking. Clicking on the *Delete* icon will then remove them.

VOID tracking shows you which bits of tracking need to be removed after a design change. If, having seen this, you decide against the design change, the procedure is to quit ARES without saving and then restore the schematic to its original state.

New connections specified in the netlist will automatically appear as extra ratsnest lines.

Problems with Pin Numbers

As the netlist is loaded, ARES pulls in the specified package for each component and then attempts to match the pin numbers used for that component in the netlist against the pin numbering in the library-package. If the netlist references a pin number that does not appear

in the library package then it displays an error message box and you must click OK to acknowledge. Problems in this area usually arise from one of the following causes:

- Specifying the wrong package - e.g. a bipolar transistor outline for a regulator.
- Failing to manually number pads which require non numeric 'numbers'. A typical example might be a DIN connector which has pin numbers like A1, A2 etc. Remember that by default, ARES always numbers the pads in the order you placed them.
- Leaving spaces on the ends of ISIS device pin names/numbers or in ARES pin numbers. This can be checked by editing the relevant object and examining closely the state of the baseline of the Data Entry Field on which the text sits. If there are any gaps in it, these designate the presence of spaces.

If you use the *Packaging Tool* in ISIS when creating new library parts, you should avoid any of the above problems.

Packaging Considerations

At some point during the design process, it is necessary to specify the library-package to be used for each component. With the ISIS/ARES system, this can be done:

- At the ISIS end by storing the package name in the **PACKAGE** property. This can be done either manually by editing each part in turn, or automatically using the Property Assignment Tool and/or ASCII Data Import facilities

  This approach is by far the preferred one as it avoids having to re-enter data each time the netlist is loaded.
- At the ARES end as the netlist is loaded.

In the latter case, ARES will prompt you for the package name to use for each component in the netlist.

A limited facility is provided to automate this process for commonly used parts - the file DEVICE.XLT can contain lines specifying device/package pairs such as

```
7400, DIL14
```

To use this feature, you must check the *Enable Device->Package Lookup* checkbox on the *Environment Settings* dialogue form. This command resides on the *System* menu.

You can then add entries to the DEVICE.XLT file either with a text editor or by selecting the *Store* button on the package input dialogue form.

Connectivity Highlight

Connectivity Highlight mode is selected by clicking left on the *Connectivity Highlight* icon. You can then:

- Highlight/tag a group of pads, tracks and vias connected to a pad by clicking left on the pad.
- Highlight all pads, tracks and vias connected to a net by selecting the required net from the *Object Selector* and then clicking left on the 'T' toggle.
- Delete all tagged pads, tracks and vias by clicking left on the *Delete* icon. Combined with the above this provides the means to rip up all tracking associated with a given net.
- Untag all pads/tracks/vias by clicking right at a point where there is no object or else by invoking the *Redraw* command.

RATSNEST FEATURES

The term *Ratsnest* is used to describe the pattern you get on the screen when the pin to pin connections specified in the netlist are shown as single straight lines rather than copper tracking. Once a netlist has been loaded and the components placed (or vice versa for that matter) the ratsnest gives a good visual impression of the complexity of the routing task that is before you or the autorouter. Furthermore, it gives an indication of the quality of component placement since the presence of lots of long ratsnest lines suggests that some components would be better placed closer together.

Automatic Ratsnest Recalculation

ARES keeps the ratsnest up to date and optimized at all times. Add a component or delete a track and new ratsnest lines will appear. Make connections and ratsnest lines will disappear. It's as simple as that.

The term 'optimization' refers to that fact that ARES always displays the 'Minimum Spanning Tree' for each net. This means that the displayed ratsnest lines always represent the shortest interconnection pattern between the pads.

The ratsnest can be turned off wholesale by using the checkbox on the *Layers* dialogue on the *Display* menu. Ratsnest lines related to a particular strategy can also be turned on and off using the *Set Strategies* command on the *System* menu.

Force Vectors

Force vectors are provided as a further aid to component placement. They appear as yellow arrows which point from the centre of each component to the point where it's ratsnest length (as defined by its current ratsnest lines) would be shortest. Another way of thinking about them is to consider each ratsnest line as a rubber band. The force vector then points to the place where the component would move to if released (strictly speaking this would also depend on the elastic linearity of the rubber bands, but we won't get into that - it's a good analogy!)

One oddity that can arise is that if you move the component to the position marked by the tip of its force vector, the ratsnest itself may change because a better minimum spanning tree may then be found. This can mean that the force vector then points to somewhere slightly different. We do not regard this as solvable; most PCB packages do not re-optimize the ratsnest after each edit, so such effects would not be so obvious - but they are still there.

The force vectors can be turned off by using the checkbox on the *Layers* dialogue on the *Display* menu.

Ratsnest Mode

Ratsnest mode is selected by clicking the *Ratsnest* icon. You can then

- Specify connections manually - useful if you are working without a schematic but still wish to use the auto-router.
- Perform pin and gate-swaps (where legal) by dragging tagged ratsnest lines from one pad to another.
- Tag a connection by pointing at it and clicking right.
- Tag all the connections for one net by selecting the net in the *Object Selector* and clicking left on the 'T' toggle.

In conjunction with the autorouter's ability to route all, tagged or untagged connections, ratsnest mode gives you the means to autoroute the board selectively.

Manual Ratsnest Entry

If you are working without a schematic, it is possible to enter ratsnest connections manually. Connections are specified by clicking left the pads to be interconnected. ARES will automatically create net-names for connections specified in this way.

Note that where component pads are on single layers, you must use the layer selector to select the correct layer for each pad. If a ratsnest line must change layer, then you can press the SPACE bar to toggle between top and bottom whilst placing the ratsnest line.

PIN_SWAP/GATE SWAP

When used in conjunction with ISIS, ARES supports pin-swap/gate-swap changes to the connectivity whilst the layout is being routed. This means that you can choose to interchange the wiring to like pins, and/or interchange the use of like elements of multi-element parts. A full discussion of how to prepare your library parts to exploit this feature is given in the ISIS manual, but from the ARES side there are two ways of using this feature:

Manual Pin-Swap/Gate-Swap

To perform a manual pin or gate swap

1. Select the *Ratsnest* icon.
2. Click right on the source pin. For a gate-swap, this can be any member of the gate. The ratsnest lines connected to the pin will highlight. In addition, legal destination pins will also highlight.
3. Hold down the left mouse button and drag the ratsnest lines to the required destination pin.
4. Release the left button. ARES will make the change, updating the ratsnest and force vectors as appropriate. In the case of a gate-swap, ARES will move other ratsnest lines automatically.

It is possible to combine a pin-swap and a gate-swap in one operation - for example, swapping input A gate 1 (pin 1) with input B gate 2 (pin 5) on a 7400 will do just this.

WARNING

Pin-swaps and Gate-swaps constitute changes to the connectivity of your design. ARES uses the pin-swap and gate-swap data specified in the ISIS libraries to decide what is, and is not, a valid swap. If there are errors in this data, then ARES may well suggest illegal swaps. We will not, under any circumstances, be held liable for any costs incurred or losses arising as result of such mishaps, whether the error be in your library parts or ours or in the software itself. We strongly recommend that you check that the swaps you make are really legal, and that your prototype your PCB prior to the manufacture of large quantities.

Automatic Gate-Swap Optimization

In the case of a board with a large number of possible gate-swaps (the SHIFTPCB sample is a spectacular example) in can be very hard to find the best arrangement. For these cases we have provided an automatic gate-swap optimizer.

To perform automatic gate swap optimization

1. Invoke the *Gate-swap Optimizer* command from the *Tools* menu.
2. ARES will make repeated passes trying the current set of possible swaps. The process repeats until no reduction in ratsnest length is achieved.

WARNING

The Gate-Swap Optimizer relies entirely on the gate-swap data specified in the ISIS component libraries to decide what is, and is not, a valid swap. If there are errors in this data, then the swap-optimizer is likely to make erroneous changes to the connectivity of your design. We will not, under any circumstances, be held liable for any costs incurred or losses arising as result of such mishaps, whether the error be in your library parts or ours or in the software itself. We strongly recommend that this command be used only if you are going to prototype your PCB prior to manufacture.

Synchronization with the Schematic

Whether manual or automatic swaps are to be performed, it is a requirement that the changes can be reflected or 'back-annotated' into the schematic. For this to occur successfully, the schematic must not have been changed as well.

PROTEUS manages this by using the netlist file as a kind of token. If ARES cannot find an up to date netlist, it will not allow changes, and if ISIS makes changes it deletes the netlist.

When ARES does make changes it writes out a back annotation file (extension 'BAF') the next time the PCB is saved. ISIS picks this up the next time it comes to the foreground. If ARES has changes that are not saved, ISIS will not allow changes.

This scheme will prevent the making of simultaneous changes to both schematic and PCB in normal use. It is, of course, possible to circumvent the token mechanism by renaming files, editing and copying back, editing on other machines etc. If you deliberately contrive to modify the schematic and PCB at the same time, then you must live with the consequences! The only cure for such situations is to check carefully that the schematic and PCB are the same and then re-load the netlist into ARES.

Further discussion of pin-swap/gate-swap is appears in the chapter entitled *ISIS and ARES* in the ISIS manual.

ROUTING STRATEGIES

In ARES a strategy defines everything about how the nets assigned to it are to be routed. This information includes:

- The trace and via styles to use.
- The via type to use - normal, buried or blind.
- Assorted controls for the auto-router.
- The design rules for nets routed with this strategy.

The beauty of encapsulating all these properties in a single entity and then assigning nets to it as opposed to assigning the properties separately to each net is that it greatly reduces the amount of information that need be entered on the schematic.

Strategies and the Netlist

ISIS can associate a named property - in this case a strategy name -with a net by placing a net label such as `STRAT=POWER`. This will then be processed by the ISIS netlist compiler and will appear as a net property just after the net name.

For example:

```
VDD,2,STRAT=POWER
U1,14
U2,14
```

If you are using some other schematics package, much depends on its ability to process net properties in some manner similar to the above. If so, then SDFGEN will include them in its output in the same manner as above. If not, then you have three choices:

- Use net names of the form such as `VDD=POWER`. ARES will parse this into a net called VDD assigned to a strategy called POWER.
- Edit the SDF netlist with a text editor and add in the strategy information by hand. By judicious use of the macro facilities of your text editor, this can be a very convenient and flexible approach.
- Ignore the strategy capabilities altogether as far as the schematic side of things is concerned. Given the automatic strategy assignment described in the next section, this may be adequate and is certainly the simplest option.

Special Strategy Names

The most common reason for needing different strategies is to distinguish power connections from signal connections in order that the power routes can be made out of thicker tracking. Also, especially when using the autorouter it is often an idea if memory bus connections are given special treatment as they really need to be routed in an orderly, regular fashion.

In order to help with achieving the above distinctions, the strategy names **POWER**, **BUS** and **SIGNAL** are given special meaning and certain forms of net name will automatically be assigned to these strategies.

- The net names **GND** and **VCC** default to the **POWER** strategy unless they are explicitly assigned to a different one in the netlist.
- Net names of the form D[0] (it is the square brackets that are important) default to the **BUS** strategy. This is now somewhat obsolete (at least as far as ISIS is concerned) since you can place a bus label such as STRAT=BUS directly on a bus segment.
  Nevertheless, this syntax may be of use with other schematics packages which do not have net properties.
- All other net names default to the **SIGNAL** strategy.

This scheme will enable many boards to be handled without need to explicitly specify strategies on the schematic or in the netlist.

Editing a Strategy

Strategies are created automatically by the *Netlist Loader* as their names are encountered. However, you will normally then need to edit them in order to set the various fields to whatever you require.

Strategies may be edited using the *Set Strategies* command on the *System* menu.

The meaning of the various fields is explained below:

Name

Selects the strategy being edited. Selecting a different strategies saves the changes made to the current one.

Priority

This is used by the auto-router - connections assigned to the highest priority (smallest number) strategy will be routed first. Strategies with the same priority will be routed simultaneously. This will probably result in the router spending much more time 'mapping',

but may lead to higher completion rates than routing them in turn, especially on single sided boards.

Trace Style

The name of the trace style to be used for connections assigned to this strategy.

Via Style

The name of the via style to be used for connections assigned to this strategy.

Power/Bus/Signal Switch

This is used by the auto-router as a guide to the type of connections to be made.

Corner Opt

This button determines whether the auto-router is to optimize corners by cutting them at 45 degrees.

Diagonal

This button determines whether the auto-router is allowed to use diagonal routes.

Pass H / Pass V

Each strategy can involve up to 8 layers on which connections assigned to it will be routed by the auto-router.

The (H) layer is for predominantly horizontal routes and the (V) layer for predominantly vertical routes. Single sided routing can be achieved by specifying the same layer for both H and V routes.

Neck Style

If not blank, this field enables *Auto Track Necking* for the autorouter, and defines the style which it will neck to.

Design Rules

These fields specify the minimum clearances allowable for nets that are associated with this strategy. In the case that tracking on two nets with different strategies is in proximity, the smaller clearances are used .

Hide Ratsnest

Check this box if you wish to hide the ratsnest lines for nets that are associated with this strategy.

BACK ANNOTATION

Back annotation is the process whereby changes to the component annotation on the PCB can be fed back into the schematic. This would normally be done if it was felt that a particular order of annotation on the PCB would aid the production department in making the board. In fact, the most natural order for this purpose is if the placement path for placing the components in successive order is a minimum.

Manual Re-Annotation

It is perfectly possible to re-annotate the board manually by editing each component reference label as required. PROTEUS uses internal unique identifiers to track 'was-is' data so it does not matter what changes you make, in what order, or when they are eventually read back into ISIS.

The only restriction is that it is not advisable to renumber connectors made from *Physical Terminals* in ISIS. This is because ISIS cannot re-annotate them - it would involve it finding not only the terminals but also the references to the part name in the *FIELD blocks. The work around is to manually re-annotate in both ISIS *and* ARES. A warning to this effect is displayed if you attempt such a change.

Automatic Re-Annotation

Should your sole aim be to renumber the components to give a geometrically sensible placement order, then the *Component Re-Annotator* command on the *Tools* menu will do this for you.

The process starts at the top left of the board and proceeds for all components picking the next nearest one not yet processed.

Back-Annotation to ISIS

Both changes to the component numbering and also changes to the connectivity caused by pin-swap/gate-swap operations will be automatically picked up by ISIS. This works whether ARES is running or not.

You have a choice of allowing these updates to occur entirely automatically (in which case, your PCB will be saved when you switch to ISIS) or to do it manually, in which case ISIS will prevent you from editing the schematic until the PCB is saved. This choice appears on the *Set Environment* command on the *System* menu in ISIS.

Further discussion of this mechanism is provided in the ISIS manual in the chapter entitled *ISIS AND ARES*.

REVERSE NETLISTING

ARES can create a netlist from a placed and routed PCB. This has a number of uses:

- As a way to check the wiring where no schematic is available.
- In cases where you wish to re-route a board, without risking changing its connectivity, and do not have the schematic.
- As an aid in reverse engineering designs brought in from Gerbit or elsewhere. (Some day we may allow you to load the netlists back into ISIS!).
- As a debugging aid where there is some doubt about whether the PCB corresponds with the schematic.

To create a netlist from a PCB

1. Load the PCB in the usual way.
2. Invoke the *Save Netlist* command from the *File* menu.
3. Choose a filename for the netlist.
4. Click OK

In cases where the layout file already contains a netlist, the net-names used in this netlist will be used. Otherwise numeric net names will be generated automatically.

SDFGEN - CONVERTING 3RD PARTY NETLISTS

Although we strongly recommend that you use ISIS to prepare your netlists for loading into ARES, we have included a utility which can convert some of the more popular netlist formats to SDF. The SDFGEN program is self-documenting and typing

```
SDFGEN
```

will display basic usage information.

The main problem with taking this route is that not all schematics packages support net properties (which are required to get best effect from the ARES strategy management features) and, probably more seriously, 3rd party device libraries will not use the same naming conventions as ISIS. This means that you will be on your own as far as compiling a device to package translation file is concerned.

AUTO-PLACEMENT

INTRODUCTION

As well as an auto-router, ARES incorporates an algorithm for automatic component placement. As with auto-routing, this may not perform as well as a human operator in all circumstances but on the other hand it can save an enormous amount of time an effort. In addition, the resulting placement can usually be 'hand tweaked' - perhaps more so than can be said of auto-routed tracking. At any rate, we would recommend that most users at least try the auto-placer as a way of quickly achieving a starting point.

The overall system is now sufficiently developed that small-medium complexity boards such as our CPU sample can be converted from schematic to PCB with no human interaction whatsoever!

USING THE AUTO-PLACER

In operation, the placer requires only that the board outline be defined (by placing lines and arcs on the *Edge* layer) and that a netlist be loaded to specify the components to be placed. A more detailed sequence of events is set out below:

To auto-place a PCB layout

1. Create a schematic in ISIS assigning group properties to those components which you wish to be located physically close to each other in the final layout.
2. Invoke the *Netlist to ARES* command on *Tools* menu in ISIS.
3. Draw a board outline of the correct dimensions using the 2D graphics tools on the board *Edge* layer. Ensure that the boundary forms a closed polygon. The best way to do this is to use a *Path* object.
4. Place any components which are immovable on the board. Hand placed components are viewed as fixed, these will be treated as obstacles by the auto-placer.
5. Select *Auto-Placer* command from the *Tools* menu in ARES and adjust the various options as you see fit. See below for more information about this.
6. Select those components from the list which you wish to be placed automatically.
7. Modify the weightings and design rules to suit the board.
8. Click OK.

9. Move and rotate the components as required and start the auto-placer again.

10. Once you have finished with the auto-placer manually place any remaining components and start routing.

THE AUTO-PLACER DIALOGUE

The auto-placer is highly configurable and as a result the dialogue form may at first appear somewhat daunting. However, it will operate quite happily with the default settings for most purposes. The function of the various fields is discussed below.

The Component Selector

The left hand pane show a list of all the components still to be placed, when the dialog form is first displayed the components are listed in alphabetical order with all the components selected for placement.

The function of the *All* and *None* buttons should be fairly obvious; the *Schedule* requires a little more explanation. By the default, the components are presented in alphanumeric order but by pressing the *Schedule* button they are then presented in the order that they will be placed by the auto-placer. In this case, components not-selected for placement always appear at the bottom of the list.

Design Rules

The *Placement Grid* defines the step size used by the auto-placer when trying to find a position for each component. This should always be set to a multiple of the grid size you are going to use for auto-routing. Values other than 100^{th}, 50^{th} or 25^{th} would be highly unusual.

The *Edge Boundary* defines the minimum distance to be allowed between any component and the edge of the board as drawn on the edge layer.

Trial Placement and Cost Weightings

Auto-placement is achieved by trying the next component at various positions on the board in an attempt to minimise a number of key factors. The relative importance of these factors is set by you, the user, by moving the sliders to the right (more important) or left (less important) for each of the seven categories.

Grouping

It is often necessary to force two (or more) components to be placed together (such as de-coupling capacitors) and this can be achieved in one of two ways. Firstly you can manually edit a placement to move the components to the desired location, however this rather negates

the use of auto-placement. Secondly you can give the auto-placer a helping hand by telling it which components are to be grouped together.

A particular problem arises with decoupling capacitors since the auto-placer will, other things being equal, tend to place all of them in one place since this will give the most routable solution! To prevent this you can use the **GROUP** property in ISIS. If the decoupling capacitor C1 is intended to decouple the chip U1, then you should add the property

```
GROUP=U1
```

to the capacitor. Then, when the placer comes to place C1, it will give high costs to positions which are far from U1 and a placement close to the device will result.

If a number of components are to be grouped together they may all be given a common group name, rather than a component name, i.e. to group U1, U2, U3, C1 & C5 together you could add the property "GROUP=TIMER" to each component.

Hint: To add properties to a large number of components use the Property Assignment Tool in ISIS.

Ratsnest Length

Arguably the most important weighting factor in terms of routability, this assigns importance to the minimisation of the length of inter-connectivity of components before routing. Note however that in our experience this should be balanced with some knowledge of the number of ratsnest crossings. A board with short total ratsnest length and a large number of crossings can often be unroutable.

Ratsnest Crossings

This weighting defines the importance of minimising the number of ratsnest lines which cross. As a component is moved around the board the ratsnest length and number of crossing changes, both of these factors will affect the overall routability of the final placement. It is important to get a good balance between this and ratsnest length.

Congestion

Small components such as resistors and capacitors may be encouraged to avoid highly congested areas of the board by increasing the importance of the congestion weighting. Note that when a small component has been associated with another component via the GROUP property congestion is ignored for that component.

DIL packages

Components which are allocated DIL packages are treated as a special case. It is seen as desirable to align DIL packages such that they all have the same orientation. To this end a number of controls on the dialogue are tailored to the placement of DIL packages. *Preferred DIL rotation* is by default horizontal but the shape of your board may dictate that vertically placed DIL packages may make better use of the available space. Two further controls in the cost weightings group on the dialogue form define how heavily penalised a component will be if the package is placed with either 90 or 180 degree rotation from the default. If you are not concerned about the orientation of your DIL packages then set these costs to zero.

Alignment

A successful PCB layout is judged not only on routability but also on aesthetics. One key factor in whether a PCB looks right is whether components are aligned. This weighting factor stresses the importance of aligning component edges. Increased alignment can sometimes also aid routing since if like components are aligned then it follows that their pins are also aligned.

Options

Push & Shove

The auto-placer recognises two types of component, those initially placed by the user which are deemed to be fixed and immovable and those placed by the auto-placer which may or may not have been moved by the user. The latter will be pushed and shoved around the board where possible in an attempt to best place the remaining components. This shuffling can become restrictive and counter-productive on boards where the pre-placed components are in the centre of the board. In such cases *Push & Shove* should be disabled.

Swap Pass

Since the components are placed sequentially, it may well be that further improvements can be made to the placement by interchanging like package styles once all the parts are down. This process can be enabled by selecting the *Swap Pass* checkbox.

OCCUPANCY DEFINITIONS

Each component that is placed on the board is treated as an obstacle for the remaining components to be placed. By default the area which is occupied by a component is assumed to be the smallest box which contains all silk screen graphics and component pins. Any two such boxes may not touch or overlap.

It is quite feasible that you will wish to modify the area taken by an object such that, for example, a DIL package has additional space on the sides occupied by the pins, or that a connector has increased occupancy on one side only. To achieve this ARES provides an *Occupancy* layer onto which 2D graphics may be drawn in much the same way as silk screen graphics are added.

Occupancy graphics are not normally seen but may be examined and edited by first decomposing the package, amending or adding graphics as necessary and then invoking the *Make Package* command from the *Tools* menu.

To add an occupancy definition to a package

1. Place a copy of the package on the layout.
2. Tag it and then invoke the *Decompose* command on the *Edit* menu.
3. Select the *Graphics Mode* icon.
4. Select the *Occupancy* layer in the *Layer Selector*.
5. Place either a box or a circle to define the occupancy.
6. Tag all the elements of the package.
7. Use the *Make Package* command to store your handiwork back the library.

The current implementation does ***not*** support the use of path objects in the context. Multiple boxes and circles are allowed at present, but this may be dropped when we code support for path objects as testing for multiple objects carries a speed penalty.

LIMITATIONS

The auto-placer is another very useful tool in the armoury provided by PROTEUS but there are a number of limitations of which you should be aware:

- It can place components on the top of the board only. Boards requiring double sided placement are beyond its scope at present.
- Whilst the auto-placer can operate on boards on which some components have been pre-placed, our experience is that the more of these there are, the less well it performs. If possible, it is better to let it place everything and then move things about afterwards.

  Pre-placing components in the middle of the board will cripple its performance because this interferes with its ability to shuffle the components about as placement proceeds. If you must have pre-placed components in the middle you may find it does a little better if the *Push & Shove* option is disabled but in general it will do badly.

- Boards which are decidedly non-rectangular also interfere with the *Push & Shove* mechanism and again the placer will perform badly in such cases. Boards with small cut-outs etc. should not present a problem, however.

Naturally we hope to code around these limitations in a future version. Our current assessment is that double sided placement is not too difficult an extension; the other issues are more intractable.

AUTO-ROUTING

INTRODUCTION

The ARES auto-router is the one of the most powerful tools in the PROTEUS system and is capable of saving you an enormous amount of time and effort. However, you'll be glad to know that despite its considerable internal sophistication, actually using it is very simple.

Features

- Grid based operation. The ARES auto-router divides the board into a grid of cells of user selectable size and decides where it can and cannot place tracks and vias according to whether the cells are 'free' or occupied by existing pads & tracking.

  This approach limits the router to placing tracks at coordinates on the specified grid, but special routines are included to deal with both single off grid pads and also rows of off grid pads. The latter involves generating a 'fan out' from the pads to the nearest grid squares and is especially effective for routing to surface mount chips.

  Special code is also included to allow routing at 45 degrees, something which a number of the recent 'shape based' routers are incapable of doing.

- Multi-strategy routing. There are a number of techniques by which a router can search for a viable route for a connection and ARES tries different techniques in turn to find the best connections. In broad terms, we use first a line search algorithm to establish good horizontal & vertical discipline and then use a costed maze search to complete the routing. A special route scheduling technique ensures that all connections are routed at minimum cost to ensure best manufacturability.

- True multi-layer routing. The router can route on up to 8 layers simultaneously enabling it to tackle the most densely packed boards and to make best use of multi-layer routing space when available.

- Rip-up & retry operation. On the harder boards, the router may not manage to route all the connections at the first attempt. In this case, the board is analysed for blockages and having removed them, it then attempts to reroute the remaining connections using the freed space. Clearly this is an iterative process but our code contains special logic to prevent cyclical behaviour. The result is that the router can make headway from as low as 80% initial completion to 100% completion given sufficient run time.

The operation of the auto-router is closely tied up with the netlisting and strategy management features which are covered in the previous chapter.

THE AUTO-ROUTER COMMAND

The auto-router dialogue form has 5 main controls:

- *All Routes/Tagged Routes/Untagged Routes* - This selects which connections will be attempted. *All Routes* would be the normal choice but in conjunction with the tag/untag connection capabilities of ratsnest mode the others to enable you to auto-route selected parts of the layout.
- *Grid* - this sets the grid spacing that the auto-router will lay down tracks on. Smaller grid sizes allow higher density routing provided that the design rules and trace style selected allow this - it is no use selecting a 25 thou grid with 50 thou tracking and a 20 thou trace-trace gap. Note also that memory requirements and routing time increase in inverse proportion to the square of the grid size. Whilst 50 thou routing generally works in seconds or minutes, 10 thou routing needs several megabytes of memory and can take hours. Fortunately, there are very few situations where a grid smaller than 25 thou is needed, especially as SMT components are handled by the fan out mechanism.
- *Edit Strategies* - this button takes you to the *Edit Strategies* dialogue form where you can specify the track widths, via styles and design rules for different groups of nets in the design. See page 72 for more information.
- *Rip-Up & Retry* - if checked, the router will engage its rip-up & retry passes once basic routing has run as far as it can. Otherwise, routing stops once no more routes can be placed.
- *Infinite Retry* - normally, the rip up and retry algorithm will stop if it detects a stalemate condition - that is where the same tracks are repeatedly ripped up and put down. Select infinite retry causes the router to go on forever, with the added twist that the board is tidied when a stalemate is detected. The tidy process often ‘frees’ the stalemate and a few more tracks may be routed.
- *Tidy Pass* - selecting this option runs the tidy pass, a process which reroutes each track in such a way as to reduce track length and via count whilst also improving the aesthetic quality of the layout. The tidy process can be very long for large boards, as the process runs for all tracks and then repeats until no more tidying is possible. A good one to run overnight!
- *Protect Manual Tracks* - when the rip-up & retry pass engages, it needs to know whether it is allowed to remove existing manual tracking or otherwise. If you have pre-routed existing tracking and do not want it moved, then check this field.

  This option also prevents manually laid tracks being tidied.

- *Enable Map Cacheing* - this basically makes the router run faster at the expense of using more memory. If you have 8Mb or less and get out of memory messages or disk 'thrashing' whilst the router is running, try deselecting this option.

Once underway, the auto-router displays a status line indicating how routing is proceeding. As well as the strategy of the connection being processed, the status line also displays:

- What the router is doing - this can be one of MAP, NOVIA, MAZE, NECK, RIPUP, RETRY or TIDY. MAP means that the router is re-generating its copper utilization map, NOVIA and MAZE are routing algorithms.

  NECK is a further MAZE routing pass but using the NECK style (if any) for the current strategy. In this mode, the router 'pretends' that it is routing with tracks in the neck style, and then places them with the normal style. The *Auto Track Necking* feature then ensures that the tracks are necked down where they pass between thin gaps.

  RIPUP means that the router is analysing for routes blocking the track displayed in red, and RETRY appears when the router is trying to replace tracks it has just ripped up.

- TR - the number of connections To Route.
- RC - the current number of Routes Completed.
- PK - the PeaK number of routes completed. Once rip-up & retry has been engaged, the RC value can actually go down temporarily if several tracks are removed to get one new one in. The PK value shows the best position achieved; this position is saved for later retrieval if the routing process is interrupted.
- RF - the number of Routes Failed by the current pass.
- PC - the Percentage of connections Completed.

The connection currently being considered is drawn in yellow.

The auto-router can be stopped by pressing ESC. Where rip-up and retry mode is in progress, you will get an option to retrieve the 'best position' which corresponds to the state of the board when the Peak Routed value was last achieved.

HINTS AND TIPS ABOUT AUTO-ROUTING

Most auto-routing exercises proceed along the following lines:

- Load the netlist and place some or all of the components.
- Set up routing strategies to define track and via styles for routing.
- Run the auto-router with rip-up & retry disabled.
- Unless complete routing occurs, examine why/where the routing has become congested and move components around accordingly. To remove tracking placed by the router, you can use the *Delete* icon in *Route Placement* mode.
- When you are satisfied with the component placement, enable *Rip-Up & Retry* mode and set the router going again. You should observe a gradual rise in the *Peak Routed* (PK) status box. Progress tends to be exponential - that is, after basic routing, the first 10 further tracks take as long as the next 5. On large, difficult boards overnight runs may prove worthwhile.
- Sometimes, the router may get stuck on the last few tracks, in which case you can break out and try to complete the board manually from the best position found by the router.

Apart from the above, the following sections form answers to commonly asked questions about using the auto-router.

Single Sided Boards

To route single sided boards, you need to edit the routing strategies so that all passes specify the single layer on which you want to route. This will force ARES to route both horizontal and vertical traces on the underside of the board.

The completion rate figure for single sided boards will obviously be much lower than for double sided. However, results of around 80% are still possible with good component placement.

Routes which are not completed will be left displayed as ratsnest lines; some manual route editing will be required to convert them to be sensible wire links.

Avoiding Using Component Pads as Through Holes

For small scale prototype, hobby or educational production it can be useful to produce boards in which the pins of some components (e.g. ICs, electrolytic capacitors) are *not* used as through holes.

The way to achieve this is to redefine the component library parts to have pads on the *Bottom Copper* layer only. The router will then only route to these pads from the underside.

Off Grid Pads

The ARES routing algorithm uses a map or 'grid' of the board which is generated at the chosen resolution. It then uses this to decide where it can and cannot place tracks and vias. However, because the grid squares are at fixed co-ordinates, it does mean that the router can only place tracking 'on the grid' - i.e. at co-ordinates of multiples of the grid setting.

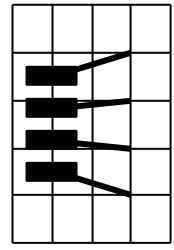
This is not really a problem provided that the component pads are also on the grid - in fact it leads naturally to a tidy result and is much faster than gridless techniques. Where component pads are not on the grid (as may happen if the pin spacing is metric, for example) the router routes to the nearest grid square and then a special routine adds an extra track segment to complete the connection to the pad centre.

What is important, however, is that you try to avoid placing components off grid without good reason. Placing normal imperial spaced components off grid can seriously degrade the router performance. For example, if a part with 0.1" pad spacing is placed at (125,120) and routing is attempted at 50 thou grid, then pins 1 and 2 will totally block the grid square at (150,120) with the result that the router will not be able to route tracks between the pads.

Surface Mount Components

Surface mount components, especially ICs, present something of a problem to a traditional grid based router because the pin spaces are rarely 25 or 50 thou. Thus tracks running on the grid cannot connect directly to the pads because the centres of the grid squares do not coincide with the pad centres.

To overcome this, whilst retaining the advantages of grid based routing (speed and technology based on extensive research) we have implemented special logic to deal with rows of off grid pads. What this logic does is to create an exclusion area around the row of pads, and then to generate within that a *Fanout* from the pad centres to the nearest suitable row of grid cells.



The fanout logic also allows fan-ins so that the area inside the chip can be utilized for routing too.

There are a number of things to bear in mind about fanouts:

- The trace style use for the fanout segments – that is the tracking from the SMT pad to the on grid routing point is created in the FANOUT trace style. The width of this trace

style must be set such that a track can leave the SMT pad at 45 degrees without violating the design rules. Since this width depends on the SMT pad spacing, the design rules and the autorouter grid it is not possible for ARES to set it automatically – instead, you should reduce it from the default value of 8th if you get DRC violations.

- The fanout logic operates outside of the autorouter code, and is therefore unable to 'see' objects placed without the fanout regions. Consequently, you must not place objects (e.g. decoupling capacitors) in the fanout areas, as ARES may well just route over them. This will, of course be picked up by a subsequent DRC check, but would result in the need to re-work the board.
- For similar reasons to the above, it is not advisable to pre-route to SMT pads that will be the subject of fanout processing. It is acceptable to pre-route a whole SMT chip, but not to pre-route to only some of its pads. Again, the consequence will likely be that ARES will place fanout segments over the manually laid tracking.

Routing to Internal Power Planes

ARES includes a router pass that deals with the connection of SMT pads to inner power planes. This is a special situation because it requires the placement of routes that goes from pad to via, rather than from pad to pad. Such routes do not correspond to particular ratsnest lines, either.

This routing pass (called PPGVIA) is activated automatically, whenever there are zones which have their *Route to this Zone* checkbox enabled. This is so by default for all zones created by the *Power Plane Generator.* Consequently, the zones should be placed prior to commencing autorouting.

The PPGVIA pass assumes that routing is allowed on the both the power plane layer(s), and on the layer occupied by the SMT pad; there is no need to allow this specifically in the strategy. However, by doing so, you will also allow the router to complete power connections using the normal routing strategies. In particular, the router will then share vias for adjacent SMT pads.

For boards with partial or split power planes, the router will complete the connectivity by conventional tracking, provided that the power strategy allows routing on the top and bottom layers, as discussed above.

TIDY PASS

ARES also incorporates a mode of operation in the auto-router which we call the *Tidy Pass*. This is provided primarily as a means of tidying layouts produced by the auto-router but there is nothing to stop you using it on manually auto-routed boards also.

To automatically tidy a PCB

1. Invoke the *Auto-Router* command from the *Tools* menu.
2. Select the *Tidy Pass* checkbox.
3. If tidying a manually routed board, deselect the *Protect Manual Tracks* checkbox.
4. Click OK.

The tidy pass can be run as part of the main auto-routing process or at a later time. It makes no difference whatsoever.

The tidy process operates by ripping up each track in turn and then re-laying it using costing heuristics that discourage vias but which also allow diagonal movement more cheaply. In addition costs associated with anticipated board usage are removed such that kinks and pointless detours in tracks are also removed. Of course, once a particular track is moved, it may reveal a new and better route for another track that has already been tidied. The tidy process thus iterates until no more tidying can occur. This does mean that the tidy process can become equivalent to routing the board 5 or more times. If it is a large board, this may take hours rather than minutes. In such cases, the best thing to do is leave it till it is convenient to run the process overnight.

POWER PLANES

INTRODUCTION & BACKGROUND

The way in which a PCB design program handles power planes has become much more important in recent years, mainly as a result of the new legislation on electromagnetic compatibility. There are at least three ways in which PCB software can implement power planes:

Grid Based Power Planes

Many mid-to-high end PCB design programs perform some kind of flood fill operation to produce power planes. Essentially, the target area is divided into squares and tests are made to see which squares can be filled with copper, and which are occupied by other objects.

This approach, although relatively straightforward to implement, suffers from a number of problems:

- Because a rectangular grid is used, any diagonal boundaries invariably come out in a staircase pattern, which is non-ideal for RF boards in particular. In addition, the power plane cannot 'squeeze through' narrow or non-orthogonal gaps and so optimum connectivity is not always achieved.
- If any of the pads to be connected to the ground plane are off grid, and if the ground plane is hatched, it is difficult to ensure that they are properly connected.
- In order to maintain the connectivity database, it is usual to represent the ground plane hatching as ordinary tracks. Not only does this use large amounts of memory and slow down the software, but it also makes it difficult to edit or remove the ground plane after it has been generated.

The last two problems could probably be overcome by sufficient coding but the first point is very intractable.

Negative Image Power Planes

Another approach is simply to draw enlarged copies of all the pads and tracks which are not to connect to the ground plane, and say that the result is a negative image of the power plane. Although used in some quite expensive software, this approach is badly flawed:

- The software cannot check if full connectivity is actually achieved - it only needs one track running right across the ground plane area to break it into two unconnected

regions. If you are not allowed tracking on ground plane layers, then this usually forces the use of a multi-layer board.

- Even if connectivity is achieved, it is possible for 'slivers' to exist where the enlarged obstacle images nearly touch - perhaps leaving just 1 thou of copper between them.

Polygonal Gridless Power Planes

Although by far the hardest approach to implement, this scheme has none of the disadvantages of the other two and, so far as we know, no significant ones of its own either.

In essence, the idea builds on the negative image approach:

- The process starts by generating polygonal boundaries surrounding all the pads and tracks within the target area. We call these polygons *Holes*.
- The holes are 'added' together in the sense that where two or more overlap they are replaced by a single hole which encloses them both. This process continues until a much smaller number of many-edged holes remain.
- These holes are subtracted from the original boundary of the target area. This boundary is also a polygon, so you can have any shape of area you like filled with copper.

  It is at this stage that the software can find out if the connectivity is complete or otherwise. If a hole cuts right across the boundary, then the boundary is split into two separate regions of copper.
- Where holes are created by pads which are on the same net as the ground plane, the software checks to see if and how a thermal relief can be placed, and amends the connectivity database accordingly.

With this much implemented, there still remain two problems:

- Where two holes nearly touch, 'slivers' of copper can exist which give theoretical but unmanufacturable or unsatisfactory connectivity.
- If the power plane is to be drawn with a Gerber photoplotter or a pen-plotter, only a pen of some minimum thickness can be used to draw it. If such a pen is used to draw the boundary of a polygon, then it will actually end up enlarged by half a pen-width, and this could violate the design rules. In addition it is impossible to draw very pointed corners with a 'fat' pen.

Both these problems can be solved by starting out from the premise that all the boundaries will be drawn with a pen of specified thickness. Then, all the holes are computed as enlarged by half this pen width over their nominal size. Thus the second point (above) is taken care of

at the hole generation stage. But in addition, *no polygon boundary can be less than a pen width in thickness* and so no slivers are created either.

Ground Planes Without a Netlist

One issue that cropped up with annoying regularity in technical support for ARES II was a requirement to produce a ground plane for a PCB which had been produced without a netlist. The problem with this is that of knowing which pads are actually connected to ground. In ARES II - which performed a flood fill from the pads on the specified net to generate a power plane - this problem appears more or less insurmountable.

Note that there is a selector checkbox on the *Edit Pin* dialogue form so that you can select individual pads to have thermal reliefs, or else connect solidly to the power plane.

USING POLYGONAL POWER PLANES

There are two alternative ways of using the polygonal power plane functionality:

- As supported in ARES II, the *Power Plane Generator* command.
- Using the *Zone* icon you can place rectangular or polygonal regions of copper which can form arbitrarily shaped ground planes occupying any chosen region of the board.

In both cases, the zones can be assigned to be hatched or solid, and computed with boundary tracks of specified thickness.

The Power Plane Generator Command

This command provides the simplest way to create a power plane and causes the generation of a single zone object occupying the entire area of a given layer of the board.

To use the Power Plane Generator command:

1. Choose a net for the power plane. If no net is specified, the power plane will connect to any pads which are marked for *Thermal* or *Solid* connection.
2. Choose a layer for the power plane.
3. Choose a boundary style for the power plane.

ARES will then generate the power plane, and update the ratsnest display to show the effect on connectivity.

Zone Placement Mode

The most flexible and powerful way to use the polygonal power plane functionality is to use the zone placement mode:

To manually place a power plane:

1. Select the *Zone* icon.
2. Select the required layer using the *Layer Selector*.
3. Select the boundary trace style for the zone from the *Object Selector.*
4. *Either:*

   - Drag out a rectangular power plane by clicking and dragging from one corner to another with the left mouse button depressed.

   Or:

   - Mark a polygonal power plane by clicking left at each vertex. In this mode, you can also hold down the CTRL key to place curved sections of boundary.
5. ARES will then pop up the *Edit Zone* dialogue enabling you to choose a net and fill style for the zone.
6. When you click OK, ARES will generate the power plane and update the ratsnest display to show its effect on connectivity.

Editing a Power Plane

Power planes are held as zone objects which behave similarly to other objects in ARES.

To edit a zone:

1. Select the *Zone* icon.
2. Select the zone's layer using the *Layer Selector*.
3. Click right then left somewhere on the zone *boundary* to tag then edit the zone.

The *Edit Zone* dialogue form contains the following fields:

Net

The net to which the zone connects. Pads on this net will connect to the zone with short track segments in the style selected by the *Relief* field. If no net is selected, then the zone will connect to pads whose *Relief* checkboxes are set.

Layer

The layer on which the zone is placed. This can be any copper layer.

Boundary

The trace style in which the inner and outer boundaries of the zone are drawn. This also determines the thinnest section of copper by which the power plane can make a connection. Setting this larger will prevent the copper flowing through small gaps (e.g. between pins) but making it smaller means that connectivity may be made only by thin sections of copper.

If the zone is hatched, the boundary style is also used for the internal hatching.

Relief

The trace style with which thermal relief connections to component pins are made. Connection to vias are by direct contact.

Do not use a relief track style that is larger than the boundary thickness or the reliefs may 'stick out' of the boundary.

Type

This can be set to one of *Solid, Outline, Hatched* or *Empty.*

Solid and *Hatched* zones should be self-explanatory - the step for the latter being definable by the *Step* field.

Outline zones draw only their outer boundaries. This option is provided mainly so that you can turn off the redrawing of large zones which might otherwise obscure the view of something else you are working on. However, you may find other applications…

Empty zones are intended to be used for creating holes in other zones - see *Overlapping Zones*, below.

Clearance

The distance by which the power plane is separated from other copper objects.

Relieve Pins

If this option is selected, then pins which connect to the zone, and which are not individually marked for *Solid* connection will be connected using thermal reliefs. If the option is unchecked, then pads which connect to the zone, and which are not explicitly marked for *Thermal* connection will be connected by solid copper.

It follows that you can achieve any desired mix of solid and thermal relief connections.

Exclude Tracking

If checked, the zone will treat tracks on its own net as obstacles. Otherwise the zone flows over such tracking, effectively ignoring it. Tracks on other nets, or loose pieces of track on no net are always treated as obstacles.

Suppress Islands

If checked, the zone regeneration logic will not draw regions of copper that do not connect to any pads, even if they touch the zones boundary box.

Route to this Zone

If checked, the auto-router will invoke its PPGVIA pass to attempt to complete connections on the Zone's net by routing to vias placed within the zone's boundary.

Deleting a Power Plane

This is very simple since the zone is a self contained object 'owning' all the copper regions within its boundary:

To delete a zone:

1. Select the *Zone* icon.
2. Select the zone's layer using the *Layer Selector*.
3. Click right twice *somewhere on the zone boundary* to delete the zone.

Automatic Regeneration of Power Planes

If you place, move or delete tracking or vias onto a board containing one or more power planes, ARES detects whether the power plane needs to be regenerated and then either:

- Proceeds to recompute the internal boundaries of the zone immediately. On a fast PC (e.g. Pentium) and a board of low to medium complexity this real time update mode of working can be quite viable with regen times of the order of 1 or 2 seconds.

  The regeneration of complex zones is run in the background such that you can carry on editing; the zone will redraw as soon as the regen process completes. For long and complex regenerations, ARES draws the zone in hatched form whilst its regeneration is pending.

  The *Background Regen Threshold* field on the *Set Zones* dialogue form determines the number of holes required in a zone for it to be deemed complex and thus processed using background regens.

or:

- Redraws the zone in hatched form to show that it is invalid. In this case, all such invalid zones can be regenerated by invoking the *Regen* command, key 'R'. This mode saves much frustration if the zone regen times are rather long, as will be the case on slower PCs or with complex boards.

You can toggle auto regeneration on and off using the *Auto Zone Regen* command on the *Tools* menu, key 'Z'.

Quick Redraw Mode

When a zone is drawn 'properly', all the external and internal polygon boundaries are drawn in tracking of the specified style. However, this can be quite time consuming and so a *Quick Redraw* option is provided in which

- The polygon boundaries are not drawn at all.
- Thermal relief segments are drawn as single pixel lines.
- If the zone is hatched, then the hatching lines are drawn as single pixel lines.

The quick redraw mode, and also the zone auto-regen feature are editable from the *Set Zones* command on the *System* menu. Both settings can be saved in the configuration file by invoking the *Save Preferences* command.

Quick redraw is disabled by default.

Auto-Routing & Power Planes

In general, it is best to place power planes prior to auto-routing. There are two principal reasons:

- The power plane will make connections to any through hole pads within itself, and this will remove the need for the autorouter to do so.
- In the case of SMT components, the automatic via placement pass (PPGVIA) is activated by the presence of the zone(s), and so it is essential that they be placed prior to routing, and that their *Route to this Zone* checkboxes are enabled. For more information see page 88.

In cases where the power plane is on a layer shared with other tracking, the power plane may become split by tracking placed by the router. In this case, some ratsnest lines may be left at the end of the routing pass, and the best thing to do is to re-run the auto-router in order to complete these by conventional tracking.

REPORT GENERATION

CONNECTIVITY RULE CHECK

When you have completed routing a board it is useful to check whether it really corresponds with the netlist from which it was created. It is important to understand that ARES relies on exact correspondence between track ends and pad centres to establish connectivity, and that tracks which run onto pads, but do not terminate properly at their centres will show both as missing connections (because no connection is detected) and physical design rule errors (because a tracks is in contact with a pad, but not connected to it).

Basic CRC Functions

The CRC check shows:

- Each pair of pins joined by a ratsnest line is listed..
- Any extra (non-specified) connections to each net are highlighted.

If you click on the entries in the listing, ARES will tag the associated tracking in a similar fashion to the *Connectivity Highlight* function.

Advanced CRC Functions

The CRC check also:

- Completely re-initializes the netlist management part of the layout database. Any floating tracking - that is tracking not connected to anything - is detached from netlist management whilst any detached tracking that has become connected to a net is brought under netlist management.
- Traces and vias involved in any extra connections are assigned to the VOID net. Further information on VOID tracking is given in The Netlist Loader & Existing Tracking on page 64.

DESIGN RULE CHECK

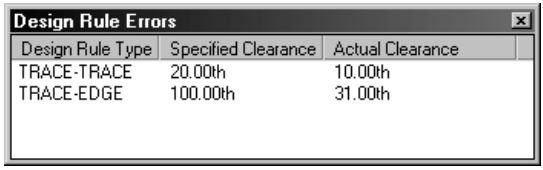
In PCB design, the physical design rules are Pad-Pad, Pad-Track and Track-Track clearance. ARES can check any layout, irrespective of whether it was created from a netlist or not, to see whether it meets a given set of design rules.

To perform a global design rule check:

1. Invoke the *Design Rule Checker* command on the *Tools* menu.
2. Enter the required design rules into the dialogue form and click on OK.

Where errors are found they are marked on screen with a red circle and a white line joining the two objects that are in conflict. These 'error flags' are actually a kind of object and you can 'edit' them to display more information about the DRC violation. The DRC flags can be cleared by invoking the *Redraw* command.

ARES also displays a popup windows listing each violation.



If you click on an entry, ARES will highlight the associated DRC flag, and if you double click an entry, ARES will both highlight it and zoom to its location on the board.

One thing to beware of is objects which are touching but not (as far as ARES is concerned) connected. This can happen if two large pads overlap or else a track does not terminate at the centre of a pad. Such situations will be marked as DRC violations since ARES sees them as two unconnected objects which are too close to each other. To eliminate such errors, make sure that such objects are connected by tracking running from pad centre to pad centre.

HARD COPY GENERATION

PRINTER OUTPUT

Output through standard Windows device drivers is performed using the *Print* command on the *Output* menu whilst the device to print to may be selected using the *Set Printer* command. This command also allows you access to the device driver specific setup dialogue form. The *Print* dialogue itself has a number of options which are explained in the following sections.

Output Mode

Four modes are available:

ARTWORK This mode produces a normal PCB artwork with pads, tracking and graphics drawn to specified dimensions. (The thickness of silk screen graphics lines is set up on the *Set Template* command).

RESIST This mode produces a negative image of the solder resist by drawing pads, without drill mark holes, and enlarged in size by the guard band value associated with each pad or via style. If you don't want holes in the resist over the vias, then set the guard bands of via styles to -1. Tracking and silk screen graphics are not drawn in resist mode. Note also, that you would only normally produce RESIST plots for the *Top Copper* and *Bottom Copper* layers.

MASK This mode produces a negative image of the SMT solder paste resist by drawing just the SMT pads at actual size. Again, you would only normally produce MASK plots for the *Top Copper* and *Bottom Copper* layers.

DRILL This mode produces a special plot with each size of drill hole indicated by a different symbol. 15 such symbols are provided by default with names $DRILL00 up to $DRILL14. If you manage to create a board with more than 15 hole sizes, ARES will draw the extra holes as circles of actual size. If you create additional $DRILL symbols, these will be used automatically, although we think the exercise will stretch your imagination!

Rotation

The Virtual Graphics Device sub-system can produce output with the x dimension either vertical or horizontal on your peripheral. This lets you orient your work most naturally within the Layout Editor and still be able to make best use of the size of your peripheral when it comes to producing artwork.

The terms x-horizontal and x-vertical will depend also on whether you have selected portrait or landscape output mode for the print device itself. On a portrait page we define horizontal to mean parallel to the short edge.

Reflection

ARES can produce normal (i.e. as seen on screen) or mirrored output. Choose whichever best suits your PCB production process.

Incidently, Windows (apart from NT) cannot mirror TrueType fonts, so this is why ARES does not support their use - there is no way to output them on reflected output, or to Gerber files for that matter.

Scaling

You can scale the output from 1:1 up to 4:1. Obviously, there is a trade off between the final board quality and the maximum size of board you can handle for a given sized peripheral. You will have to experiment for best effect here.

Layers

Except for a drill plot, it is normal procedure to produce a different artwork for each layer of the board - in ARES this means the copper and silk screen layers. Many people dispense with the solder side silk screen - it is most useful if a board has components on both sides, or must be tested or maintained solely from underneath.

PLOTTER OUTPUT

Windows support for Pen plotters is, unfortunately, very poor. Although drivers are supplied for HPGL and other plotters, the implementation of these drivers is very sketchy. Better drivers may be available for particular plotters, but we have not relied on this in designing the plotter support within ARES for Windows. Instead, we rely on the Windows plotter driver only to draw straight lines and then ARES itself does the rest. To do this, it uses the plotter driver module from the DOS version. This module allows control of the following options which are adjusted using the *Set Plotter* command on the *System* menu.

Plotter Pen Colours

The plotter drivers are able to generate a multi-coloured plot when ALL layers are selected on the *Print* dialogue form.

The *Set Plotter* command on the *System* menu is used to determine the colours for each layer. Where through hole pads are encountered, the pen number is determined by ORing the colours for the outer layers of the pad in the same way as is done for screen colours.

Note that this only applies to an ALL layers plot - single layer plots always use the black pen.

Unfortunately, the Windows plotter drivers do not (as far as we have been able to find out) support the direct selection of plotter pens by number. Therefore it is up to you to find out how your plotter driver maps colours onto pens and to select appropriate colours on the *Set Plotter Pens* dialogue.

Plotter Tips

On the *Set Plotter* command form are fields for *Pen Width* and *Circle Step*.

Pen width

This is a value in thou which should correspond (at least initially) to the width of pen you are using.

Circle step

This value is the size of step (in thou) that ARES will use when plotting the outer radius of a 1 inch diameter pad. A smaller number results in faster but more jagged circular pads whilst a value of 0 will instruct ARES to use the plotters internal circle drawing command. Whilst this is generally fast and smooth, it does mean that the pen is picked up and put down for each pass round the pad.

Getting high quality plots will require experiment with pens, film, ink, and the settings described above. You may also find it necessary to tweak the sizes of some of the pad, via and trace styles - the problem is that there are too many variables outside our control to guarantee that the dimensions of plotted objects will accurately tie up with their nominal size. As a starting point, we recommend a 0.25mm or 0.30mm tungsten carbide tipped pen, professional drafting film (obtainable from a good stationers) and Marsplot 747 ink.

Finally, the *Line Width* command on the *Set Template* command form is ignored when plotting - a single pass of the pen is used for all silk screen graphics for reasons of speed. If you need a different line thickness, use a thicker pen for your silk screen plots!

POSTSCRIPT OUTPUT

Postscript, invented by Adobe Inc, is most commonly used in professional DTP and typesetting. However, it is an extremely flexible language well suited to accurately specifying a PCB artwork. More importantly, the £10K photo-typesetters which are used by professional publishers are capable of producing >1200dpi images on film. Access to these machines is facilitated by the existence of bureaux and our experience is that they charge less than Gerber Photoplotting bureaux for an equivalent plotted area. Try calling a few local printers from the Yellow Pages to track down a local bureau.

Those of you with a Postscript laser printer will be able to proof your boards on it before generating a file for photo-typesetting.

With ARES for Windows, you just select and install the appropriate Postscript printer driver, and use the *Print* command in the normal way. If you use a local phototypesetting bureau, they should be able to supply you with the correct driver for their phototypesetter.

CLIPBOARD AND GRAPHICS FILE GENERATION

As well as printing directly to Windows print devices, ARES for Windows can generate output for use by other graphics applications. You have the choice of generating this output as either a Bitmap or a Windows Metafile, and you can transfer the output to the other applications either through the clipboard, or by saving it to a disk file.

Bitmap Generation

The *Export Bitmap* command on the *Output* menu will create a bitmap of the board and place it either on the clipboard or in a disk file. You have the following additional options:

- Resolution - choose from 100 to 600 DPI. Memory usage increases in proportion to the square of the resolution.
- Number of colours - mono resolution is simple black & white, display resolution uses the same bitmap format as your display adaptor.
- Other options are as for PRINTER OUTPUT on page 101.

Metafile Generation

The Windows Metafile format has the advantage of being truly scalable where a bitmap is not. However, not all Windows applications (e.g. *Paintbrush*) can read a metafile.

The *Export Metafile* command on the *Output* menu will create a metafile of the board and place it either on the clipboard or in a disk file. You have the following additional options:

- Colour - choose mono or colour output.
- Placeable header - if you are generating a disk file, there are two formats - one with a placeable header and one without. Some applications need the header (e.g. Word) and others will not load a metafile that has one!. All you can do is experiment.
- Black background - if checked, this option causes ARES to draw a black rectangle before drawing anything else in the metafile.
- Enhanced - generated an Enhanced metafile. Enhanced metafiles are supported by Window 95 and Windows NT applications only.
- Other options are as for PRINTER OUTPUT on page 101.

DXF File Generation

The DXF format can be used to transfer output to DOS based mechanical CAD applications (it is better to use a clipboard metafile to transfer to Windows based CAD programs). The file is generated by a Labcenter output formatter, rather than by Windows.

Our current experience is of considerable incompatibility and disagreement between applications on what constitutes a valid DXF file. To put this another way, given 6 applications supporting DXF, only about 30% of file exchange pairings seem to work! For Windows work, the Clipboard provides a much more reliable transfer medium.

DXF file generation shares the same output options as PRINTER OUTPUT on page 101.

EPS File Generation

An EPS file is a form of Postscript file that can be embedded in another document. Although popular in the world of DTP, for Windows based DTP work you are much better off transferring graphics using a clipboard metafile.

EPS file generation shares the same output options as PRINTER OUTPUTon page 101.

Overlay Bitmap Generation

The *Export Overlay* command on the *Output* menu will create an overlay diagram and place it as a bitmap either on the clipboard or in a disk file.

The overlay image is created by first rendering the copper layers in a ‘tint’ and then superposing one or more silk screen layers on top of this. The effect is to show the positions of the components with the tracking faintly visible underneath.

You have the following additional options:

- Resolution - choose from 100 to 600 DPI. Memory usage increases in proportion to the square of the resolution.
- Number of colours - mono resolution is simple black & white, display resolution uses the same bitmap format as your display adaptor.
- Heavy - this checkbox causes the ‘tint’ for the copper layers to be darker - 75% instead of 50%.
- Other options are as for PRINTER OUTPUT on page 101.

CADCAM OUTPUT

THE CADCAM OUTPUT COMMAND

All CADCAM output - Gerber files, NC Drill and Tool Information is generated from a single command. The dialogue form allows you to select:

- A common filestem for all files produced. This defaults to the layout filestem but a neat trick is to change it to something like `A:MYBOARD` such that all files produced will go directly onto a floppy disk in drive A.
- Whether the files are to contain normal or mirror image co-ordinates.
- Whether the screen X co-ordinate is directed to the CADCAM x co-ordinate (X-Horizontal) or to the CADCAM y co-ordinate (X-Vertical).
- Whether the newer RS274X or the older RS274D Gerber format is used. RS274X has the major advantage that aperture information is embedded within the files and does not have to be manually entered by the photoplotting bureau.
- Which layer is used for mechanical routing data - see page 109 for more information.
- Which layers are output.

Any files generated from a previous run but now corresponding to a disabled layer are deleted - the new tool information table may be wrong for them.

Finally, an 'INF' file is produced which lists all files generated together with the tool settings for the photoplotter and the drilling machine.

GERBER OUTPUT

The Gerber format, named after Gerber Scientific Instruments Inc., is almost universal in the PCB industry as regards specifying PCB artwork although we suspect that it may gradually be superseded by Postscript.

A photoplotter is essentially the same as a pen plotter except that it writes with a light beam on photographic film rather than with an ink pen on paper. The aperture through which the beam shines can be varied, allowing it to expose a pad in a single flash and a track with just one movement of the plotting head.

In order for a photoplot to be produced (usually by a bureau, as photoplotters are not cheap) it is first necessary to set the machine up with a set of apertures corresponding to the various pad shapes and track widths used on the board. Each different aperture is referenced by a so

called D-Code in the Gerber file and a table listing D-Codes against aperture shape and size must thus be compiled. This table is compiled automatically and then written out as part of the tool information file.

ARES is clever enough to tell if two pad styles result in the same aperture, and it will only use one D-code in this situation. Also, styles which are present in the selectors but not used on the board do not waste table space.

Other points to bear in mind:

- An aperture cannot render the image of a pad with a centre hole - the piece of metal for the hole would fall out! Photoplots thus come out with completely solid pads and boards produced from them are very difficult to drill manually. This, among other things is why Postscript is a much better format.
- Rectangular pads that are rotated to non-orthogonal angles have to be rendered by hatching them with a fine line. Once again, Postscript overcomes this problem.
- Polygonal ground planes also have to hatched (even if you have selected them as solid) and the resulting output files can be massive. However, the boundaries will be drawn exactly as computed - using an aperture corresponding to the boundary thickness for the zone - so there is no risk of design rule violations.
- The origin for the Gerber co-ordinate system is defined by the *Set Output Origin* command on the *Output* menu.

NC DRILL OUTPUT

Many PCB manufacturers now have Numerically Controlled drilling machines which provided with the positions and drill tools to use for each pad, can automatically drill a circuit board. Nearly all these machines use the Excellon format, invented by Excellon Industries.

The format is very simple, consisting of the (x,y) locations of the holes and T-Codes specifying which tools to use. As with Gerber Format, a table must be supplied to the manufacturer listing the hole diameter corresponding with each T-Code. ARES handles this in the same way as it handles each Gerber D-Code - it assigns a new T-Code for each new hole size it encounters, and this information is also put in the tool information file.

ARES uses the same co-ordinate systems for Gerber and NC Drill output - based on the position of the *Output Origin* - so your manufacturer should have no problem in aligning the drilling machine with the artwork.

The drill sizes used are taken from the Drill Hole attributes of the pad styles. Please make sure that the default sizes we have allocated are suitable for your application before having large numbers of boards manufactured.

MECHANICAL ROUTING AND SLOTS

There are two sets of circumstances in which it may be necessary to form make a non-circular hole within a board:

- Some components have solder lugs rather than round pins, and these are best mounted on a pad which has a slotted rather than a drilled hole.

  This can be achieved by defining a slotted pad-stack. See page 52 for more information.

- Sometimes there is a need to make a large cut-out in a board, usually to accommodate some aspect of the mechanical design of the product.

  This can be achieved by drawing 2D graphics on the appropriate MECH layer - see below.

Unfortunately, there is no widely accepted mechanism within the industry for specifying the location of slots and cut-outs within the CADCAM data. Therefore, the best we can do is to output the coordinates of each routing stroke in Gerber format, and then leave it to individual board manufacturers to convert the data for their own particular routing machines.

To do this, we make use of one of the MECH layers within ARES. The *CADCAM Output* dialogue form allows you to specify which MECH layer is to be used, and any slotted pads which then generate appropriate routing strokes on that layer. 2D graphics placed on that layer will then also be interpreted as defining routing strokes.

It is important to ensure that your board manufacturer understands that the specified mech layer contains routing co-ordinates in Gerber format. You will need to tell them this explicitly.

We will continue to monitor the industry to see if a more standardized method for specifying mechanical routing operations will emerge.

PICK AND PLACE FILE

ARES incorporates the ability to produce a file for use as a starting point in setting up automatic insertion machines. Essentially this file lists the component layers, positions and rotations in standard quote/comma delimited format.

An example file is shown below:

```
LABCENTER PROTEUS PICK AND PLACE FILE
=====================================

Component positions for K:\Prodev\Ares\ppsu.LYT

Fields: Part ID, Value, Package, Layer, Rotation, X, Y
Units:  Rotation - degrees, X, Y  - thou
Notes:  The X, Y value is the centre of package as drawn in ARES.
        The origin for these values is the Output Origin.
          The values are a guide only and must be checked manually
when
        setting up automatic insertion equipment.

"U1","","DIL08",TOP,0,6000,5000
"Q1","","TO220",TOP,180,6050,5375
"D1","","DIODE30",TOP,180,6050,5250
"R1","","RES40",TOP,270,6300,5050
"R2","","RES40",TOP,270,6400,5050
"Q2","","TO92",TOP,90,5650,5050
"R3","","RES40",TOP,180,5800,4750
"C1","","CAP10",TOP,180,6200,4750
"C2","","CAP10",TOP,0,5650,5350
```

There are a number of points that must be appreciated about this file:

- The origin for the co-ordinates is the *Output Origin* - this is the same origin as is used for the Gerber and Excellon outputs.
- The (x,y) coordinates are quantified in units of 1 thou and represent the centre of the component's package. This position may or may not correspond to the origin of the component for auto-insertion, but it will be in approximately the correct place. Our understanding is that this is helpful in that it provides a starting point for manual alignment of the placement head.
- The rotations are in anti-clockwise values in degrees relative to the orientation of the package when it was defined. Since there is no standard for default orientations of packages these values may be of limited use unless they can be combined with a translation table that is specific to ARES packages including ones you have defined yourself. This is a matter between yourself and whoever is providing the auto-insertion facility.

As far as we are concerned, this is an evolving and experimental area of the software; a number of users have requested an output file of this nature but there is as yet no industry standard for us to base it on. Any feedback on this issue will be greatly appreciated.

GERBER VIEWING

Since photoplotting charges can be quite substantial, it is useful to be able to view Gerber files to ensure that all is well prior to sending them to the bureau.

To facilitate this, ARES provides a *Gerber View* command which will load and display selected files produced by the *CADCAM* command.

To view Gerber output files

1. Invoke the *Gerber View* command from the *Output* menu. If you have modified but not saved the current design you will be prompted to do so, as the current design data is lost when the Gerber files are read.
2. Choose an 'INF' file from the file selector. The *Gerber View* can only read Gerber files produced by ARES for which an INF file exists.
3. ARES will then parse the file and display a dialogue form for selecting which layers to view. By default, all available layers are loaded.

Although it is possible to perform editing operations on the Gerber View data, we do not recommend or support this mode of operation, aside for panelization which is discussed in the next section.

Do not confuse this feature with GERBIT (the Gerber Import Tool), which is available as an optional extra. See GERBER IMPORT TOOL *on page 119 for more information on GERBIT.*

PANELIZATION

Panelization refers to the processes of combining several board artworks onto one panel in order to reduce manufacturing costs. It is common practice to panelize both multiple copies of the same board and/or several different boards.

ARES provides facilities to achieve this through the Gerber Viewer.

To create a panelized artwork

1. Use the *CADCAM Output* command to produce Gerber and Excellon files for each board to be included on the panel.
2. Invoke the *Gerber View* command and choose the INF file of the first board to go on the panel.
3. Accept the default settings for the layers (these should correspond with the layers you chose to generate at step **1**) and check the *Panelization Mode* checkbox.

4. Click OK. ARES will import the CADCAM data for the board and display it as a tagged set of objects.
5. Use the *Set Work Area* command on the *System* menu to specify the dimensions for the panel.
6. Use the *Block Move* command to re-locate the board image to where you want it on the panel.
7. If the panel is to contain multiple images of a single board only, you can now use the *Block Copy* command to lay them out.
8. If the panel is to contain images of additional boards, you can import them into the panel by returning to step **2** above.
9. When the panel is complete, use the *CADCAM Output* command to generate final artwork files for the panel. It is these files that you need to send to your board manufacturer.

There are a number of points to note about the panelization process:

- You can add additional text and graphics to the panel as required to designate board IDs and other manufacturing information.
- When importing multiple board images, ARES will assign new pad and track style codes (D-Codes) for each image. However, when the CADCAM files are re-generated, ARES will re-combine styles that are actually the same, so that the total usage of aperture and tool codes will not be excessive.
- The resulting set of files may be quite large - especially if the board contains ground planes or non-orthogonal components. However, that is an inevitable consequence of producing a file that contains multiple board images. Fortunately, the files are ASCII, and will compress quite well with WINZIP or similar, should you need to transmit them by email.
- The process is not suitable for panelizing boards containing buried or blind vias, because the layer ranges of the drill holes are not preserved. If you need to do this, we recommended you make use of a 3rd party Gerber viewer/editor package.

DXF IMPORT TOOL

INTRODUCTION

The DXFCVT converter is fully integrated into ARES and is driven through the *Import DXF* tool on the *File* menu. It facilitates the import of mechanical data in DXF format, and we envisage it being used in situations where a board needs to be fitted into an existing mechanical design. It could also be used to import other graphics such as company logos.

The conversion process is as follows:

- Generate the DXF file from your mechanical CAD application. This should, if possible, be a 2D drawing. ARES cannot handle 3D data and the converter thus ignores all z co-ordinates in the DXF file so 'flattening' 3D drawings to two dimensions.
- Determine the mechanical layers in the DXF file that are to be combined and placed on particular ARES layers. The DXF Import dialogue form allows you to specify one or more layer assignments with each assignment specifying which of the mechanical layers in the DXF file are to be combined and placed on the specified ARES layer.

Note that it is particularly important that the correct units are used for DXF import. As per the original DXF specification all co-ordinates are interpreted as being in inches. If after importing your file you find that the graphics appear wildly out of scale then you probably have a DXF file whose units are in millimetres and you will need to specify a scaling factor of 0.03937 . This causes the importer to divide all co-ordinates by 25.4 (1" = 25.4mm).

You should appreciate that Autocad (the application around which the DXF format was based) supports much more powerful mechanical drafting than that offered by ARES. Thus some entities in a DXF file are ignored by DXFCVT, and others are approximated. However, for the purpose of displaying simple cabinet internals and the like inside ARES, DXFCVT is quite adequate to the task.

☞ See the *Limitations* section for specific information on the limitations of the DXFCVT converter.

SETTING UP

Generating The DXF File

The DXF file to be imported should be generated by the mechanical CAD application you use. Consult the documentation or on-line help associated with the CAD application to determine how best to do this.

As far our import converter is concerned, the only requirements are that:

- The file exported by the application be in plain ASCII and ***not*** in the alternative binary (DXB) format.
- Co-ordinates in the DXF file are in floating-point inches with the value 1.0000 being assumed as one inch - this is the norm for DXF files.

Layer Assignments

Having generated the DXF file you must now decide on what *layer assignments* you require. Each layer assignment tells the converter which layers in the DXF file are to be combined and placed on a single ARES layer. The controlling dialogue form is laid out in a fashion that should make this process fairly transparent and Context Sensitive Help is provided for all fields on the form should you encounter any problems.

Note that, for a given conversion, an ARES layer may only be assigned once. An attempt to assign a layer twice will result in an error.

PERFORMING A CONVERSION

Basic Conversions

To perform a DXF file conversion you need to load the converter with the name of the DXF file to be converted. This process is implicit with ARES V in that immediately on selecting the *Import DXF* tool you will be asked to open the appropriate DXF file. A successful loading of the file will result in the appearance of the *DXF Import* dialogue form. All available options on the dialogue form are covered via Context Sensitive Help on the appropriate field.

Conversion Errors

With regard to DXF file parsing errors, the converter has been written to be as fault-tolerant as is consistent with reading a DXF file. Parsing errors are only generated when:

1. The DXF file doesn't contain an ENTITIES section. A DXF file without entities is deemed 'empty', which makes conversion impossible.
2. The DXF file contains a section (e.g. HEADER, ENTITIES) or a primitive (e.g. LINE, CIRCLE, etc.) that doesn't contain a single record. All sections should at least contain an end-of-section record and all primitives should contain at least one record. This implies the DXF file is corrupted.
3. The end of a BLOCK entity in a non-BLOCKS section or the end of the BLOCKS section before the end of a BLOCK entity. This implies the DXF file is corrupted.
4. The failure to extract (parse) data of a type consistent with the group record code. For example a record with a group code of 10 implies an integer value and a record with a group code of 41 implies a floating point value. An error is generated if (regardless of formatting) a value cannot be parsed.

Whenever an error occurs the converter reports the error in the form:

```
ERROR   (000413): Entity Group Expected - EOF found?
```

and the conversion is aborted.

For file parsing errors, the number of the last line read from the DXF file is shown in brackets after the ERROR keyword (note that the converter reads whole groups - two lines - from the DXF file at a time, and so any error may be on the line indicated or the preceding line). For non-parsing errors, the line number is displayed as dashes.

Conversion Warnings

Apart from errors (described above) all other reports of unusual or possibly erroneous conversion behaviour are treated as warnings. Warnings are not fatal (they do not stop the operation and completion of the converter) but are useful if the resulting region file is not what was expected. In order to be complete, the converter issues warnings whenever it encounters a situation that may or may not lead to unusual region files and this includes any record it reads and discards as being either of unknown purpose or unconvertible. In order not to slow the conversion down (with screens of scrolling text), warnings are only displayed if requested (through the *Generate Warnings* checkbox on the dialogue form).

Warnings are displayed in a similar format to errors (described above) except that the DXF file line number line and message text is preceded by the word `WARNING`.

LIMITATIONS

The principle limitations of the DXFCVT converter are summarised as follows:

- DXF files must be in plain ASCII. The converter does not support the DXB binary file format.
- Three-dimensional DXF drawings are flattened to two-dimensions ('plan' view) by the converter as ARES only supports two dimensions.
- The only entities (alone or as part of a BLOCK entity) supported are the BLOCK, LINE, POLYLINE, CIRCLE, ARC, TEXT, SOLID, TRACE, and INSERT entities. All other entities are ignored and do not affect file conversion. The SOLID and TRACE entities are converted as polygons. In particular, the DIMENSION entity is ***not*** supported.
- Only those attributes of an entity that can be honoured by ARES are converted. Thus text attributes such as font face, weight (boldness) and italic slope angle are discarded and do not affect the file conversion.

Other minor limitations include:

- Line types (dots, dashes, thicknesses, etc.) are lost. All lines are single-pixel.
- Colours are lost - all converted entities are displayed in a colour appropriate to the ARES layer they are on.
- Text styles (from the DXF TABLES section) are ignored. The converter relies (for those attributes it converts, such as rotation, mirroring, etc.) on these attributes being specified as part of the TEXT entity - this is the norm.
- The conversion of DXF TEXT entities drawn in a proportional text style will convert to a fixed-pitch font in ARES where the width of the characters may result in the overall length of the string in ARES being different to the overall length of the string in the DXF drawing. This may then lead to alignment problems unless the user has made judicious use of the DXF text alignment flags (e.g. text that is specified as fitting between two points).

  As the ARES text primitive is limited to fixed-pitch characters which only allows for the specification of a character height and character width for the text string, the only way around this problem would be to adjust the ARES character width such that the number of characters in the string multiplied by the adjusted character width is the same as the overall length of the string in the DXF file. Unfortunately, except for certain justifications, the DXF file does not contain the overall string length and so this isn't possible.

The actual behaviour of the converter is to calculate the character height as being equal to the height of the DXF TEXT entity and the character width as being equal to the TEXT entity height scaled by the TEXT entity's x-scaling factor (the latter defaults to 1.0).

- The following text attributes and properties are **not** converted:
  - ? Vertical fonts (i.e. where characters are stacked on top of each other rather than alongside each other).
  - ? Obliquing angle (for italics).
  - ? Styles - all text is in ARES's fixed-pitch vector font.
  - ? Control characters. These occur in the DXF file as a caret followed by an ASCII character - all such characters in the string are preserved.

  The following text attributes and properties **are** converted:
  - ? Rotation.
  - ? X and Y mirroring.
  - ? Justification, except attribute '5' ("Text is fit [sic] between two points (width varies)") as it is ambiguous and hasn't been tested.

- Only the following entities are converted as noted.:
  - ? LINE and 3DLINE entities.
  - ? CIRCLE entities.
  - ? ARC entities (by default, by linearisation to multiple chords).
  - ? SOLID, TRACE and 3DFACE entities are converted to unfilled three/four sided polygons. The 'hidden' attributes of 3DFACEs is not honoured.
  - ? TEXT entities is converted as described above.
  - ? BLOCK entities.
  - ? INSERT entities are converted but any associated ATTRIB entities are discarded - this leads to text declared in the INSERT via a ATTDEF entity being lost.
  - ? POLYLINE and associated VERTEX entities. All polylines are assumed to be continuous (i.e. **not** a mesh) and all VERTEX entities are assumed to be single point (i.e. not Bezier or spline).

  In addition to the above some other omissions are:
  - ? DIMENSION entities - these are usually also exported as LINE, SOLID and TEXT entities as well, so there should be no loss of visible information.
  - ? ATTRIB and ATTDEF entities, as outlined above.
  - ? SHAPE entities - these will (if a problem) be added in a future release.

- All DXF file co-ordinates are assumed to be mathematical, in inches, and with an implied origin at (0,0). It is the user's responsibility to specify alternate scaling parameters (via the appropriate options on the dialogue form) to comply with this assumption.

Positive angles are assumed to measured in degrees counter-clockwise from an implied zero axis corresponding to the positive x-axis. The necessary DXF file header variables are checked and a warning issued if this is not the case.

- Layer names are honoured throughout out the conversion, with the layer name "0" (the single character zero) being taken to have the layer name of it's parent. For example, for a BLOCK object with entities on layers 0, BLUE and GREEN and an INSERT of the BLOCK on layer RED, the BLOCK entities on layer 0 are deemed to be on layer RED.

  How the converter determines which entities with a BLOCK are to be output is determined by which layers are enabled and the settings chosen under the *Symbols* group on the dialogue form.

GERBER IMPORT TOOL

INTRODUCTION

GERBIT is available as an optional extra - it is not part of the standard Proteus package. It facilitates the conversion of PCBs designed with other CAD programs to ARES format. The conversion process involves the following steps:

- Generating the required Gerber files with the source CAD system. At the same time, you need to generate an aperture list so that you know which 'D-Codes' specify which pad and track types.
- Creating new pad and trace styles in ARES to correspond with those used in the source layout(s).
- Processing the sets of Gerber files with GERBIT to create a single ARES region file for each layout.
- Where several boards are to be converted, it is often worthwhile generating a dummy board in the source CAD system containing all the library parts used in the real ones. This can then be converted and the library parts captured from it into the ARES libraries.
- Reading each region file into ARES, and reconstituting its components by over-placing library packages over their pads. It is for this reason that the previous step can often be useful.

It is vital to appreciate that the Gerber format specifies the board at the level of individual pads and tracks, and that all information pertaining to which pads belong to which components is lost. The extent to which this matters depends on how much editing you want to do on the imported boards, but ARES provides specific features & tools to help with the reconstitution process. Nevertheless, reconstituting a board requires a considerable degree of manual effort.

REQUIREMENTS FOR CONVERTIBILITY

The scheme we have implemented for Gerber import makes certain demands on and assumptions about the source CAD system. These are:

- The ability to generate standard Gerber photoplotting files. GERBIT can accept any M.N format (e.g. 2.3 or 2.4) in imperial or metric units. GERBIT assumes 2.3 imperial by default, so it is simplest if you can set the source system to generate this. The end of block character must be '*' and the character set ASCII.

- The ability to create a human readable aperture list corresponding to a batch of Gerber files.
- At minimum, the system must be able to generate separate Gerber files for each of the copper layers. Additionally, it will prove of great assistance if it can generate files for the component and solder side silk screen layers. If it can generate a file with either pads and vias only, or better still, component pads only, this will further facilitate the conversion and capture of library parts.
- The system must be set so as *not* to create pads out of multiple flashes or, worse still, draw strokes. The conversion process assumes that a flash corresponds to a single pad, and a draw stroke corresponds to a track segment.
- If the system can produce an Excellon format NC drill file, this can be used to assist with the combination of vertically aligned pads on each layer into single through hole pads in ARES. If no drill file is available, GERBIT will perform this task by matching co-ordinates, but this may lead to occasional problems if there are pads which are coincident, but not through hole (e.g. SMT pads).

WARNING:

Note that the Gerber Import Tool will only work with files in the RS247D Gerber Format. Files in the new RS274X format cannot be imported using this utility.

TUTORIAL

Before we discuss some of the more advanced issues involved in Gerber import, we will first take you through an example conversion session.

This section presents a walk through using the ARES door bell sample. Note that this is rather contrived since we have generated the GERBER files from ARES in the first place.

Preparing for Conversion.

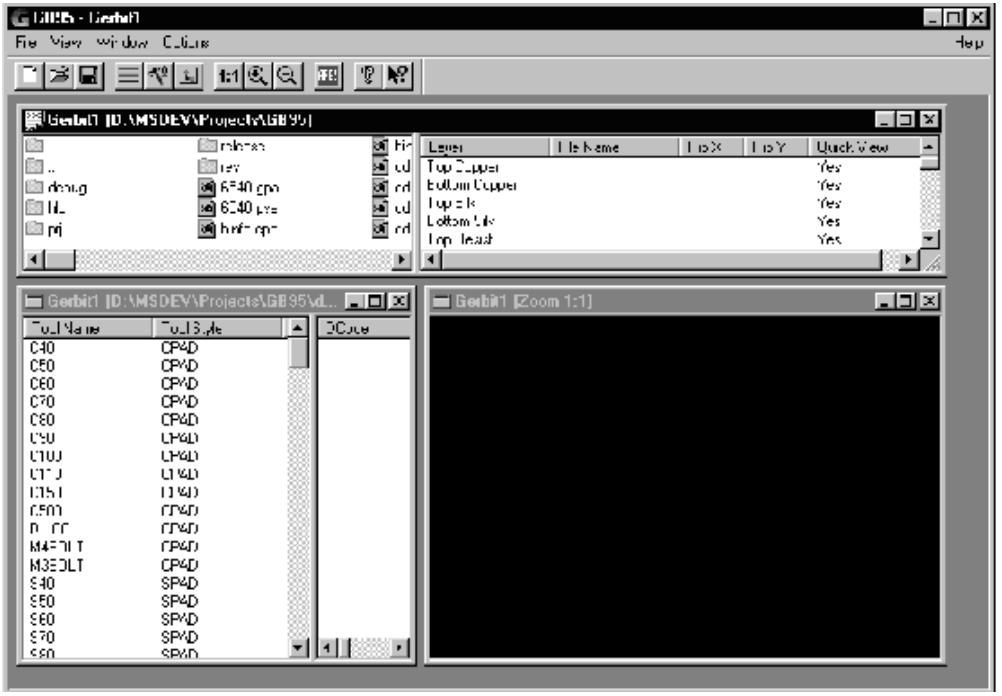
The exercise starts with the GERBER files having already been generated from the CAD system, these are

DBELL.TOP	Component side copper
DBELL.BOT	Solder side copper
DBELL.TS	Component side silk screen
DBELL.DRL	NC drill file
DBELL.INF	Aperture information file

Start Gerbit now and select *New* from the *File Menu* (File | New). Since this is the first time that you have used Gerbit you will be asked to locate your DEFAULT.STY file, this is in the

same directory as your ARES program and contains definitions for all the pad and track styles that ARES knows about.

Your are now presented with three views, the uppermost view is the file/layer view, the lower left is the style/tool view and the lower right is the preview window.



In the *File View* window double click on the file entitled DBELL.INF. This will launch a text viewer which allows examination of any ASCII file.

The contents of the information file is shown below:

```
LABCENTER ARES III TOOL INFORMATION FILE
========================================
Tool set up for C:\PROTEUS\SAMPLES\DBELL.LYT

File List
---------
Top Copper      c:\Ares\dbell.TOP
Bottom Copper   c:\Ares\dbell.BOT
Top Silk        c:\Ares\dbell.TS
Drill           c:\Ares\dbell.DRL

Photoplotter Setup
------------------

Format: ASCII, 2.4, imperial, absolute, eob=*, LZO
Notes:  D=Diam, S=Side, W=Width, H=Height
D10  T25
D11  T15
D12  T10
D13  C50
D14  C80
D15  S50
D16  C60
D17  S60
D18  C100
D19  CIRCLE D=8th                    DRAW
D70  CIRCLE D=50th                   DRAW

NC Drill Setup
--------------
Format: ASCII, 2.4, imperial, absolute, eob=<CR><LF>,
        no zero suppression.
Notes:  Tool sizes are diameters.
T01  20th
T02  30th
```

It is important to study this file before converting as it provides us with key information regarding the conversion process. Firstly we can note that there are four GERBER files to convert and secondly that they are in 2.4 imperial format.

As is happens imperial 2.4 is the default so we do not need to change the resolution for this project. We now have to place the GERBER files in the Layer view and this is achieved using drag and drop. Left click on the file called DBELL.TOP and keep the left mouse button pressed drag the file to the right until the top copper layer in the layer view window is highlighted in blue. Now release the mouse button.

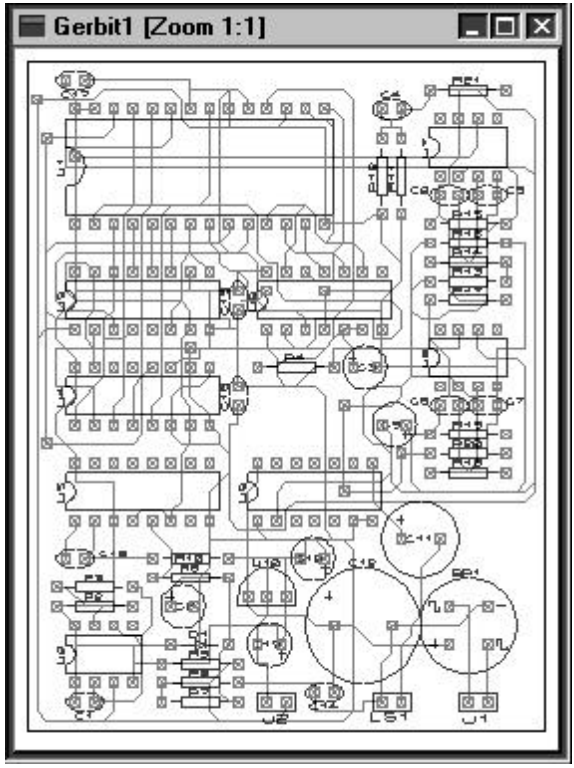
A number of things will happen at this point.

- The name of the file will appear in the layer view.
- A list of D-Codes used in the GERBER file will appear in the Tool View
- A preview of the board graphic will appear in the preview window.

Now drag DBELL.BOT onto the bottom copper layer and DBELL.TS onto the top silk layer.

DO NOT drag the drill file across yet (we will demonstrate its use later).

The preview window as shown below does not accurately represent the board since the tracks are all the same thickness and the pads are all assumed to be a checked box.



To complete the process we must use the style and tool views to allocate the correct apertures as defined in the DBELL.INF file. This is once more a drag and drop process. Select the tool view window and drag the apertures from default to their appropriate tool codes such that you have:

D10 T25
D11 T15
D12 T10
D13 C50
D14 C80
D15 S50
D16 C60
D17 S60
D18 C100
D70 T50

you will notice that DCODE 19 is defined as

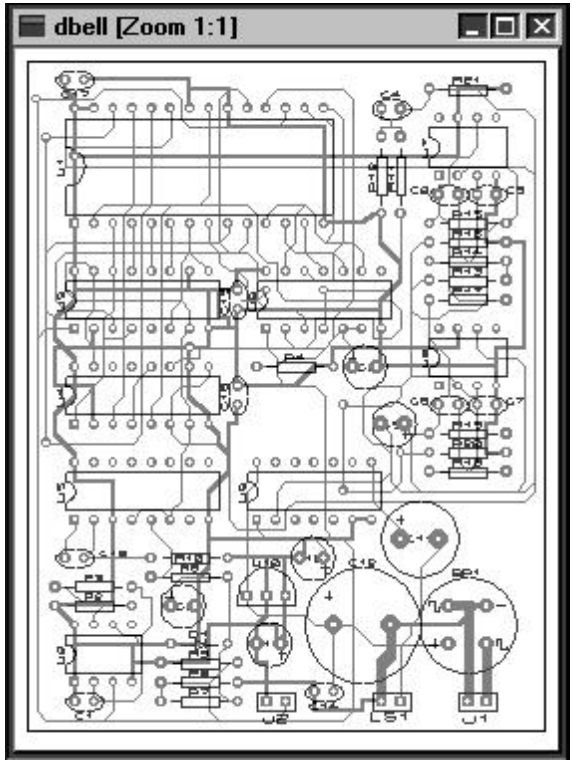
D19 CIRCLE D=8th DRAW

this means a trace (DRAW) of thickness 8th and that no such style exists in ARES. This will be a common problem and you will need to be aware of the range of pads and trace styles which are used in you GERBER files. You can then define the required styles in ARES.

Alternatively you can create a pad or trace style specifically for this board. Simply select Create New Trace Style (or Pad Style) from the options menu. You are then asked to name the style and supply the required dimensions. When defining trace styles you only need to define the track thickness. When defining pads the dialog box changes to reflect the dimensions required for the pad style selected.

Once the new trace style has been defined it appears in the tool view along with tools defined in the default.sty file.

As you drag each of these styles onto the tool view the preview will update to reflect the changes. With off the tools in place you should see the following.



All that now remains is to generate a region file to import into ARES and this is achieved by selecting *Options* and then the *Make Region File* command .

THE FILE/LAYER VIEW

The File/Layer View is split into two sections with a file navigator window on the left and a PCB layer selector on the right. Files are dragged from the file navigation window onto the layer view to define the structure of the board. As files are dragged across the layer to which the file is to be applied is highlighted.

Right clicking in the file view window displays a pop up menu with three options, display disk drives, Change directory and view file. By default the file browser window does not display the disk drives available on your machine (or network), selecting this option displays the disk drives and allows you to navigate to them by double clicking the desired disk drive icon. In a similar manner you may change directory by double clicking the directory folder icons or you may alternatively select change directory from the pop up menu. You are then asked to enter a directory name. The third and final menu option is view file and is used to allow you to

examine text files (usually information files) in a read only text window. Simply right click on the file you wish to view and select view file, alternatively double click on the file to view.

The layer view also has a right hand mouse button menu, which has the following options.

Flip X	flip the currently highlighted layers left to right.
Flip Y	flip the currently highlighted layers top to bottom.
View	Include/Exclude highlighted layers from graphic preview.
Clear	Reset highlighted layer information to default settings and remove file name.
Colour	Define the graphic display colour of the currently selected layer.

THE TOOL VIEW

The tool view is again split into two windows with a list of all pre-defined tools in the left hand window and a list of the tools needed by the Gerber files in the right hand window. As files are dragged onto the layer view the right hand pane is updated to show the tool codes used by the Gerber file which has been added to the project. There is no way for the Gerbit program to know what pad and trace styles these codes represent, but it will indicate the tool type as either a Flash (Pad) or Draw (Trace). You must then drag the tool types from the left hand pane into the tool views right hand pane with reference being made to the Gerber information document.

New pad and trace styles may be added to the list of defined types using *Create New Pad Style* or *Create New Trace Style* . Once defined these styles may be used as any other predefined style. Note however that they are only available to the current design. If a style is to be repeatedly used add the style to the list of Ares defined pads and traces.

THE QUICK VIEW WINDOW

The quick view window is used to provide a graphic representation of the Gerber board at its current state. Before tools are defined the board is displayed with thin lines representing all tracks and a hollow box representing all pads. As the tools are added to the design the preview is updated to reflect the changes made. You may zoom in or out of the drawing using the three zoom options under the view menu. Normal provides and approximate one to one mapping of the PCB if the board appears far too small or far too large at one to one check the resolution settings for the Gerbit project. If the tracks appear as a tangled mess check that the positioning is set to either relative or absolute as required.

When zooming out a maximum zoom ratio of 1:10 is applied. When zooming in the maximum board size is 32 in or a maximum ratio of 10:1 whichever is smaller. Hence a board nominally

2inx2in may be zoomed to a ration of 10:1 whilst a board 4inx4in may only be zoomed to 7:1. A board which is nominally larger than 32 in will not be previewed.

The primary usage of the preview window is to ensure correct orientation of the layers since some cad systems flip tracking on the bottom copper. If you wish to see a more accurate view of the board load the region file generated by Gerbit into Ares.

ADVANCED CONSIDERATIONS

The following topics may be relevant to you once you have mastered the basics of Gerber import, or if you encounter difficulties with converting particular types of board.

Memory Usage

In general, ARES will need considerably more memory to handle a board during Gerber import, than it would if the same board were created in the usual way. This is because all the silk screen graphics are held in a much lower level way than they would be if they part of library packages.

Another thing to bear in mind is that captured library parts will often be considerably larger than they would be if they created from 2D graphics. This is particularly true for silk screen outlines involving curves, as these will be rendered as lots of short straight lines in Gerber format. Some CAD systems use an awful lot of lines too.

Finally, the *Convert Vias* command should be used on small numbers of vias at a time since when a via is converted to a lone pad, the lone pad form uses about three times as much memory.

Pin Numbering

ARES uses the pin numbers assigned to library parts when it is loading a netlist in order to tie up package pins with device pins in ISIS. Now it is perhaps unlikely that this will matter much for imported boards, since you are unlikely to have an ISIS schematic. However, if you intend using Gerber Import to capture library parts for future use with netlists then pin numbers do matter.

If you simply convert a group of vias to pads, tweak the silk screen and invoke the *Make Package* command, the pins of the package will be numbered in the order that they appeared in the Gerber file. Unfortunately, this is in no way guaranteed to be the correct even for simple packages like DIL outlines. The only way to find out is to use *Make Package* and then check the assigned pin numbers by zooming in – the pin numbers will be drawn on the pads once there is sufficient space for them to be legible.

On the (more than likely) assumption that the pin numbering is wrong, you will need to adopt a slightly more involved approach to capturing packages. Having converted a set of vias to pads, the additional steps required are as follows:

4. Select the *Instant Edit* icon.
5. Invoke the *Auto Name Generator,* set the prefix to the empty string and the count to '1'.
6. Click on the pads in the order you want them numbered.

ARES will indicate the number it has assigned to each pin as you mark it.

Converting a Batch of Layouts

If you have several boards to convert, rather than just one, it may be worth capturing all the library parts first. This is because it can be rather messy trying to capture them in situ from existing boards.

The trick is to create a 'dummy' board in the source CAD system with all the library parts you wish to capture nicely spaced out on it. You can the convert this board and capture the parts off it without having to mess about deleting stuff off the silk screen and so forth.

Also, don't forget the *Write Style Set* and *Read Style Set* features can save you having to re-enter the aperture list on each run, provided that it is actually the same for each set of files.

Alignment

Due to the number of stages through which design passes during Gerber import, the alignment of its pads onto the absolute grid sometimes goes astray. For example, the relative spacing of the pads might be 50 thou, but they might all be at multiples of 2 and 52 thou.

Although *Real Time Snap* will facilitate editing such a board to some extent, it will be better if you re-align it. This can be done by tagging the entire board, and then invoking the Align command on the Edit menu. This offsets all the tagged objects such that the majority of the pads lie on the current snap grid.

Systems Which Output a Single Pad File

We have come across one CAD system that outputs pads in a single 'pad master' file, and then outputs separate tracks only files for the top and bottom of the board. From an ARES viewpoint, the entities from the pad file need assigning to all layers, whilst the track files need assigning to the top and bottom copper layers, respectively. Unfortunately, GERBIT will only let you assign one Gerber file to a layer per session.

If you are faced with this, or a similar situation, you need run two sessions using the following strategy:

- On the first session, assign the pad file to *both* the top and bottom layers and the silk screen file to the *Top Silk* layer. Also, on the *Aperture Selection Screen*, use the *Write Style Set* command to preserve the style set for the second run.
- On the second session, assign the top and bottom tracking files to the *Top Copper* and *Bottom Copper* respectively

With ARES import each region file in turn, making a note of the co-ordinate readout the first time and then duplicating it the second time. Provided that the source design has a border on all the layers, this will guarantee registration between the two imports.

An alternative approach would be to assign the tracking files to a pair of inner layers in the first session. You can then either settle for this format for your imports, or if you are confident at using a text editor, you could edit the region file and change the layer assignments within it. Track blocks are designated with the line:

```
*LAYER <name>
```

where <*name*> is the short layer name.

INDEX

N

O

P

R

S

T

U

V

W

Z