Using XFig

Peter Hiscocks

Department of Electrical and Computer Engineering
Ryerson Polytechnic University

phiscock@ee.ryerson.ca

August 17, 2001

Contents

1 Producing Documents

We are blessed to live in a time when document production tools, word processors and drawing programs, have become accessible to everyone and relatively easy to use. These document processors allow us to produce documents ranging from short technical papers to extensive textbooks, documents that are that are pleasant to view, easy to read, and entirely electronic.

As recently as the mid-80's, we were still using rubber cement to paste figures into documents. If they still exist at all, the documents themselves are now falling apart: the hand-drawn diagrams are yellow, and fall off the page that they were mounted on.

In contrast, an electronic document does not age and can be revised with ease. Sections of it can be cut and pasted for use in other documents. Copies are clean: they are not marked up with corrections and revisions. They can be transmitted electronically. Best of all, in my opinion, the figures are included automatically.

1.1 Document Processing Tools

In the UNIX world, XFig is one of the most popular and used programs to produce diagrams. (Unfortunately, it is not available to run under DOS or Windows). Other tools include the LaTeX text processing program, **GNUPLOT** for generating graphs, **xdvi** and **ghostview** for viewing documents, and **xv** for manipulating images. This suite of programs makes it possible to produce high quality documents that can be printed (*treeware*) or viewed on-screen.

1.2 Bitmap and Vector Drawing Programs

There are basically two types of drawing programs in this world: bitmap and vector programs.

Paint programs, such as XPaint, work with *bitmaps*, which are groups of pixels. The drawing program lays down and manipulates individual pixels in various ways. Paint programs have their uses, but it is difficult to create and manipulate symbols, and they are generally not suitable for technical drawings. On the other hand, paint programs are good at generating different fill patterns, creating shapes (such as curves), and complicated shading.

Vector drawing programs work with *drawing elements* such as lines, circles and polygons. These elements may be grouped and manipulated in various ways, such as being moved or sized. Vector drawing programs are well suited for technical work such as electronic schematics and mechanical diagrams. XFig is a vector drawing program, and it has some paint-program features. For example, it is possible to fill areas. Unlike paint programs however, XFig has a problem with complex curves.

2 Applying XFig

The drawing program XFig can be used to create a wide variety of technical drawings. It's not difficult to learn to use XFig, but it can take a while to learn to use it in an effective and efficient manner. This report assumes that you have learned the basics of operating the program. It provides hints on various techniques that may be useful with the program. It is not intended to be a comprehensive manual for the program: that is the purpose of the HTML pages, which may be viewed by any browser, and the MAN entry.

3 Drawing Tools

The drawing tools of XFig are shown in figure 1.

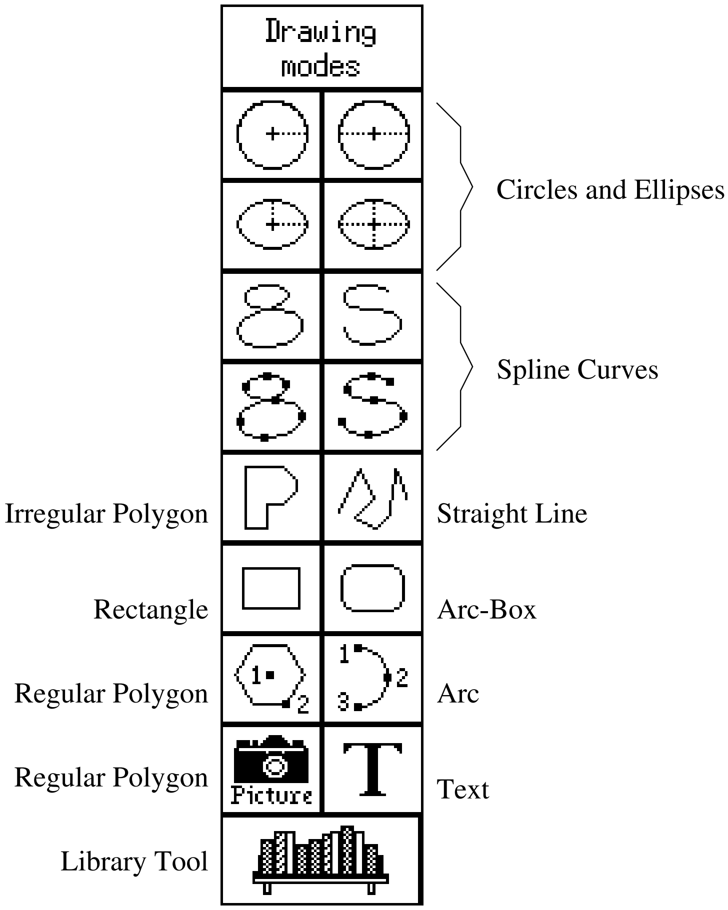


Figure 1: XFig Drawing Tools

(XFig was used to create figure 1. The procedure is described in section 10.)

Most of the drawing tool operations are straight forward, but here are some miscellaneous hints on the use of the various tools:

- The **Spline Curves** are the only available method for constructing smooth curves. The upper spline passes through the two end points and approximates the other entered points. The lower spline passes through all the drawn points.

- The **Line** is drawn by specifying the starting point and subsequent points with mouse button 1 and the end point with mouse button 2. A freehand line (and spline) may be drawn by starting with mouse button 2.

- When the **Regular Polygon** tool is selected, a parameter box appears at the bottom of the screen, allowing one to specify the number of vertices in the polygon. By creating a polygon with n vertices and then drawing construction lines from the centre of the polygon to the vertices, one can subdivide a circle into n-1 equal segments. This is useful when drawing something like a Compass Rose, for example.

- The **Arc** is specified by marking three points with mouse button 1. When a precise arc is required, it is often useful to first draw a circle with the desired radius and centre point. Then place the arc points on the circumference of the circle as required, and erase the circle.

- Many parameters become available when the **Text** tool is selected. A wide variety of text fonts are available, and the text can be printed at various sizes and justifications. To edit text, you can use the **Edit** tool (described below). Alternatively, choose the **Text** drawing tool, click the cursor at the point of the text to be modified, and type in the desired text.

- The use of the **Picture Object** tool is described in section 5 below.

- Selecting the **Library tool** pops up a panel from which one of several libraries of *Fig* objects may be loaded. After selecting one of the objects and popping down the panel, you may place the object on the canvas one or more times by pressing mouse button 1 at each location. The image of the object follows the mouse for placement ease. Pressing mouse button 2 while in this mode pops up the library panel where you may select another object or library. Another useful feature is that before placing an object on the canvas it may be rotated clockwise 90 degrees by pressing 'r', counter-clockwise by pressing 'l', flipped vertically by pressing 'v', or flipped horizontally by pressing 'h'.

4 Editing Tools

The editing tools of XFig are shown in figure 2.

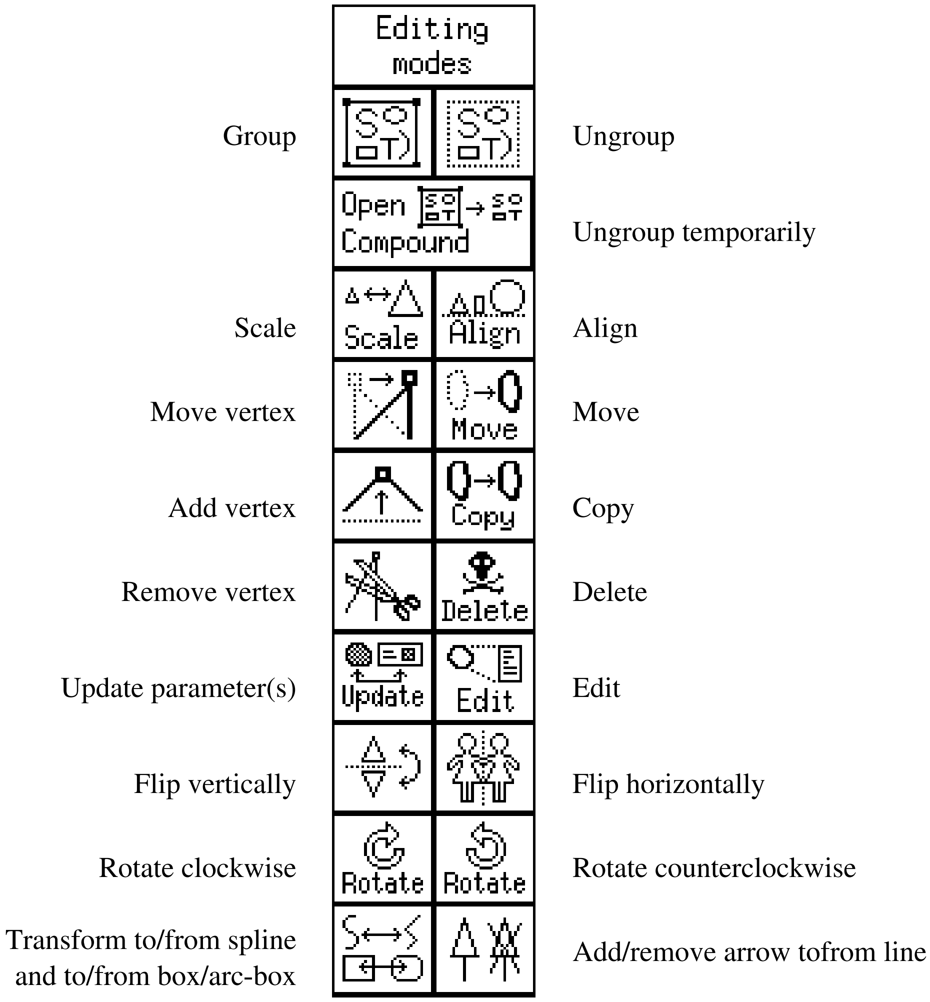


Figure 2: XFig Editing Tools

These tools are described in the following sections:

4.1 Grouping

One of the most useful functions in any drawing program is the ability to specify a *group* of symbols into what is called a **compound** object. This effectively allows you to create any object and then move it as a unit in the drawing.

For example, an electronic symbol consists of an arrangement of circles and lines. Once grouped into a compound, the symbol becomes a transistor and may be positioned, rotated, or become part of a larger symbol.

On digital schematics, grouping may apply to a bunch of gates that perform some function, such as buffers on a digital bus. They may be grouped and moved as an 8-bit buffer unit.

In XFig, symbols are grouped as follows:

1. Select the **compound** tool by clicking on it with mouse button 1. The tool darkens to indicate it is selected. At the same time, handles (squares) will appear on each symbol of the drawing.
2. Select the items one at a time to be grouped, by clicking on each item handle with mouse button 1. The handle of each item will darken when it is selected. If you change your mind, click on the same handle to unselect that item.
3. When ready to make the compound, click mouse button 3. Handles of the selected items disappear and new handles appear at the corners of the new compound object.

Often, it is faster to group a whole bunch of things at once. This is done as follows:

1. Select the **compound** tool by clicking on it with mouse button 1.
2. Select a group of items by pressing mouse button 2 and dragging a rectangle around the groups. When the rectangle is positioned correctly, click mouse button 2 again, and everything within the group is selected at once.
3. Click mouse button 3. Handles will then appear at the corners of the new compound object.

Ungrouping (breaking) is much easier:

1. Select the **break compound** tool (mouse button 1).
2. Select the compound to be 'broken' (mouse button 1). The handles of the individual objects appear.
3. Click on the handle of the compound object to be broken. It becomes individual objects, and their individual handles are shown.

If you only want to modify an object in a compound object but don't want to hassle with ungrouping and re-grouping it, the **open compound** tool is useful.

1. Select the **open compound** tool (mouse button 1).
2. Select the compound to be 'broken' (mouse button 1). All the other objects on the canvas dissappear except the compound and the handles of the individual objects in that compound appear. Also, a popup panel appears with two buttons: 'Close This Compound' and 'Close All Compounds'.

3. After editing the objects (including adding new objects and deleting objects) click on ‘Close This Compound’ to automatically re-group the objects into the compound and make the other parts of the drawing re-appear. You may repeatedly open compound objects within other compound objects. If you are inside a compound that is several levels deep inside other compounds, the ‘Close All Compounds’ button will close them all up in one step.

4.2 Scaling

In general, I don’t use this tool very often. When you are working to a rectangular grid (snap-to-grid drawing), you expect various symbols to line up in a particular way. For example, the base of a transistor should always be on a grid point. However, scaling symbols destroys that simple relationship.

That said, the scaling control is quite versatile: an object may be scaled horizontally, vertically or diagonally. The last method changes the size of the object without changing its proportions.

4.3 Align

This tool is useful when you want a precise alignment, vertical or horizontal, and snap-to-grid is not an option.

1. Group the objects to be aligned as described in section 4.1 above.
2. Select the **Align** tool.
3. From the **Alignment Parameters** at the bottom of the drawing area, select the type of alignment desired (one of six possiblities).
4. Click on the group and the objects in it will be aligned as required.

Nifty.

4.4 Move, Add and Remove Vertex

These tools are **very** useful. They can be used on many of the drawing objects to modify the shape of the object. For example, a line vertex may be moved, a line may be broken into additional sections (kinked), or a line may be reduced to fewer segments (straightened) with these tools.

One caution: when adding a vertex to a spline, the add point must be on an imaginary straight line between two existing vertices, **not** on the spline trace itself. This can take a while to figure out if you don’t know what’s going on . . . .

4.5 Delete

Select **Delete** and then click on an object to erase it. If you make a mistake, **Undo** will bring the object back, but keep in mind that there is only one level of undo. You can only **Undo** the latest action.

To erase a group of objects, select **Delete** and then use the middle mouse button to drag a selection rectangle around the objects. Click the middle button again and the objects are erased.

4.6 Update Parameter

This tool is not a must-use when you are first learning XFig, but it can be a real time-saver later. The **Update** tool allows you to change an object's properties by clicking on the object. As a consequence, it is especially useful when a large number of objects must be changed. For example, you might decide that every text string on a drawing should be larger. One possible technique is to **Edit** every text string. However, **Update** is much faster.

Here's how it works: When you select the **Update** function, the **Update Parameters** are shown at the bottom of the screen. Unfortunately, there's not enough room to show them all, but the important ones are there. (You may have to scroll it using the scrollbar below it to see all the parameters). Now, when you click on an object, its properties will be changed to the properties shown in the **Update Parameters** list.

There's another way to do this, which is quite useful:

1. Select the **Update** tool

2. Choose some object A that has the parameters that you would like to copy to another object. Click on that object with mouse button 2 instead of button 1.

3. The parameters of *this* object are now transferred to the **Update Parameters** display.

4. Now click on the object B to be updated, using mouse button 1, and object B will acquire the settings of the object A.

Furthermore, and this can be a life-saver, if you update a compound object such as an object with several text strings included in it, *every sub object in that container* (in this case, every text string in that object) will acquire the new properties.

If you only want to affect certain parameters of an object, there are little red boxes in the upper-right corners of the **Update Parameters** buttons. You can select and deselect those settings that you want to use for updating objects or the panel itself. In addition, you may turn all the boxes on by pressing the solid red button in the box labelled **Update Control** which appears on the left side of the **Update Parameters** panel. There is also a black button which turns **off** all the update boxes and a half red/half black button which toggles the current state of the update boxes.

4.7 Edit

The **Edit** tool allows you to view or change the properties of an object using a dialog popup. The most common edits are

- altering the content of a string of text

- changing text justification left, right or center

- changing the point size of a text string

- enabling the **special** flag on a text segment so LaTeX will process it as a special symbol. Don't forget to enclose the text string in $ signs, as required by LaTeX , if the string represents a math symbol.

- changing the width of a line segment

- changing the **depth** of an object so that it is displayed above or below some other object. Objects at level 0 are on top, higher numbered objects are at lower (possibly hidden) levels.

4.8 Horizontal and Vertical Flips

Not much to say: they work as advertized. In addition, for these and the rotation buttons, pressing mouse button 2 instead of 1 will first make a copy of the object then flip or rotate it.

4.9 Rotations

Again, pretty straightforward, but notice that the rotation angle can be changed in the Parameter Display list.

4.10 Spline to/from Straight Line Segment / Box to/from Arc-Box

This will convert a spline to an ordinary line or a box to an arc-box and vice versa.

4.11 Add / Remove Arrowhead

Not only does it work, but the algorithm is smart enough to know which way the arrow should point. Works with all types of lines, too, not just straight ones. To remove arrowheads, click mouse button 2.

5 Importing and Tracing: the Picture Object Tool

It is possible to import a 'picture object', such as a bitmap (GIF, JPEG, etc.) or a drawing or photograph that has been scanned, into XFig. This may then be scaled, rotated, and incorporated into the final drawing. Formats such as EPS, GIF, JPEG, TIFF, and others are supported.

I found this very useful in the production of a textbook. The author had produced a large number of hand-drawn charts and graphs that were to be incorporated into the final text. Using a flatbed scanner on a DOS system at the university, I scanned each of the images into an eps file. This was aggravated by frequent crashes of the DOS system, and the files had to be scanned at low resolution to fit onto a floppy disk.

The quality of the scan was unuseable in the final text: line widths were variable and there was a lot of image noise. I loaded each eps image into XFig, traced the image, and then deleted the eps image. The resultant tracing was of a form that could be incorporated directly into the text, with the bonus that it was not only of far higher quality than the scan, but also a much smaller file size.

Tracing the image was facilitated by the XFig **undo** feature. In the process of tracing, I would periodically delete the eps image to compare the tracing with the original. Then, when I knew what had to be fixed, I would **undo** the delete to restore the eps image for further tracing.

In another case, I wished to produce an annotated photograph: a photograph containing descriptive text with pointers to different areas of the photo. This is easy: import the photo, mark it up with lines, arrows and text, and then export or print it.

The procedure for importing a picture image takes several steps:

1. Ensure that the picture image (foo.eps, for example) is in a known location in the file system.

2. Select the **picture object** tool by clicking on it. Nothing happens just yet.

3. Using Button 1 of the mouse, drag a rectangle (any size, it doesn't matter) onto the drawing canvas.

4. Click Button 1 again to lock the size of the rectangle. A dialog box appears.

5. Type the file name (foo.eps) into the file selector field and click on **Apply**. The image loads into the rectangle.
6. Click on the **use orig size** button to establish the eps image at its original size, or enter a percent of full-size value and click the **Scale by %** button and then click the **Apply** button again. The image adjusts itself to the correct size.
7. Click on **Done** to exit the dialog box.

   The picture image can now be rotated, moved and sized as desired.

6 Creating Schematic Drawings

Extremely complex electronic diagrams may be done with ease using XFig. I've moved away from pencil and paper: once I have a rough idea of the circuit, I draw it out using XFig and then proceed to analyse and test it further. Pencil drawings now seem too slow and difficult to change.

I have created a library of schematic symbols as one xfig drawing, called *symbols.fig*, which is shown in figure 3. In version 3.2.1 of XFig with the addition of the **Library Tool**, these symbols have been incorporated into the **Electrical** and **Logic** libraries to make it easy to create schematics.

You simply click on the **Library Tool** button to pop up the library panel and pull down the menu of current libraries and choose the one you want. This loads the library of objects into xfig where they appear in a scrollable list. Single clicking on an object will make its image appear in the *Object Preview* window. When you have found the object you want press the *Select object* button to select it and pop down the library panel. Double clicking on the object name will select it and pop down the panel in one step.

Now, as you move the mouse over the canvas you will see the object you are dragging. If you need to rotate it or flip it before placing it, press 'r' to rotate it clockwise (right), 'l' for counter-clockwise (left), 'h' to flip it horizontally, or 'v' to flip it vertically.

When you have it in the correct orientation and position on the canvas click mouse button 1 to place it. You can place the same object over and over again.

If you want to select another object from the library, click mouse button 2 and the library panel will pop up again.

As with all xfig operations, clicking mouse button 3 cancels the library operation.

Besides using the **Library Tool**, there are various ways to use parts of existing figures to create new figures.

The Merge Load

To merge another drawing into your drawing without destroying the drawing, under the **File** menu, choose the figure to merge and press the **Merge** button. This will merge the new figure into your existing figure as a compound object which then may be moved, scaled, etc.

Alternatively, you can *cut and paste* from another instance of XFig, as described next.

Cutting and Pasting

Unix is a multi-tasking operating system, so it is possible to invoke more than one instance of XFig. Start up XFig twice: one instance will be used to draw the picture, which I will call **schematic.fig**. The other instance should be started, and the file *symbols.fig* loaded into it. I will call this **symbols.fig**.

It is now possible to *cut and paste* symbols from **symbols.fig** into **schematic.fig**.

1. View the **symbols.fig** drawing by opening its window.
2. Select the **copy** function. A little box 'handle' appears with each symbol.
3. Right Click (button 3) on the handle. A message 'object copied to scrapfile.fig' appears on the status line.
4. View the **schematic.fig** drawing by opening its window. Click on the **PASTE** function at the top of the window, and the selected symbol will appear in your drawing.

If you only need a few symbols, you can cut and paste one at a time. If you need a bunch, make a group in **symbols.fig** and paste that group into **schematic.fig**.

An example of a schematic diagram created with this process, is shown in figure 4.

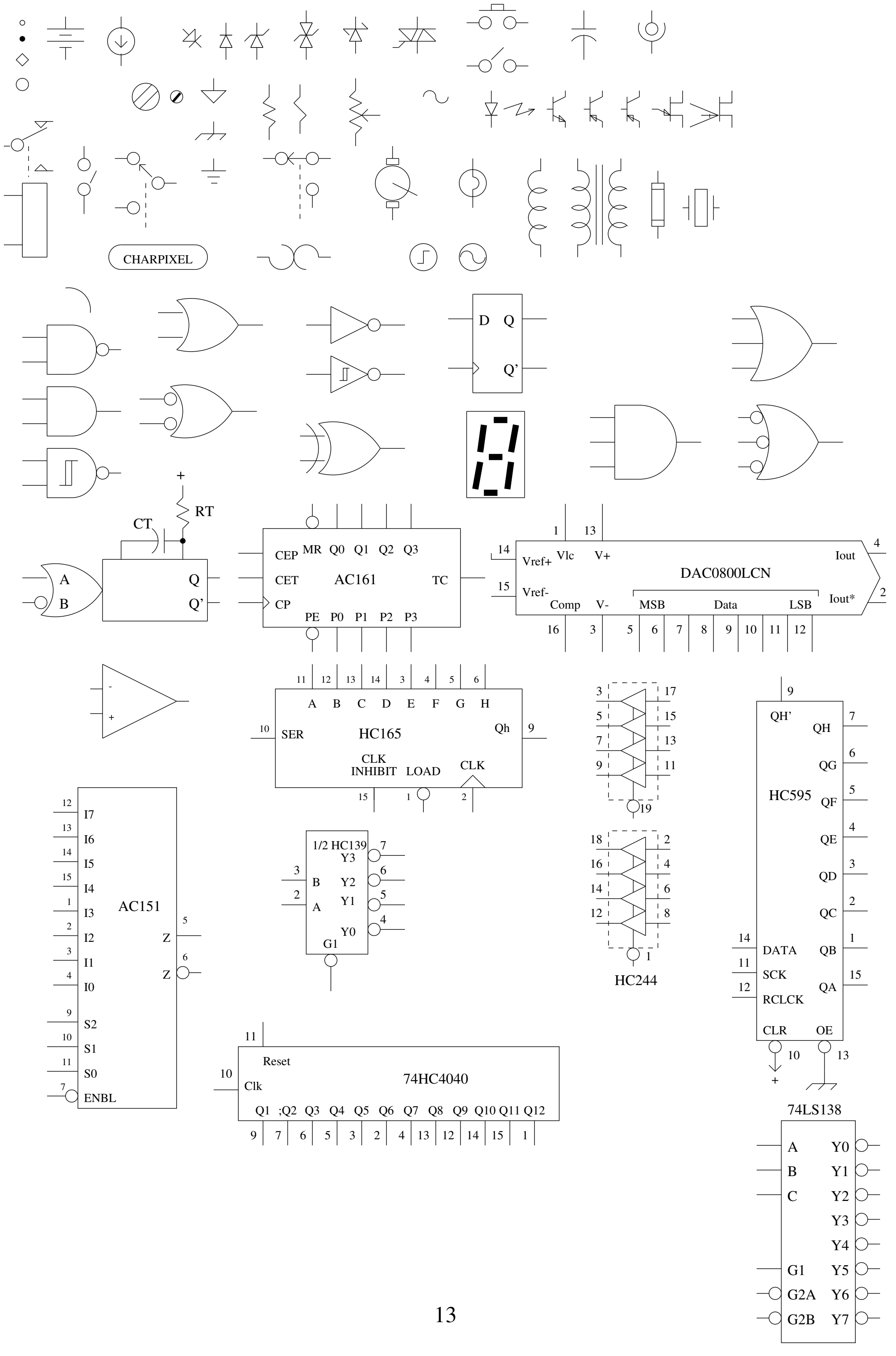
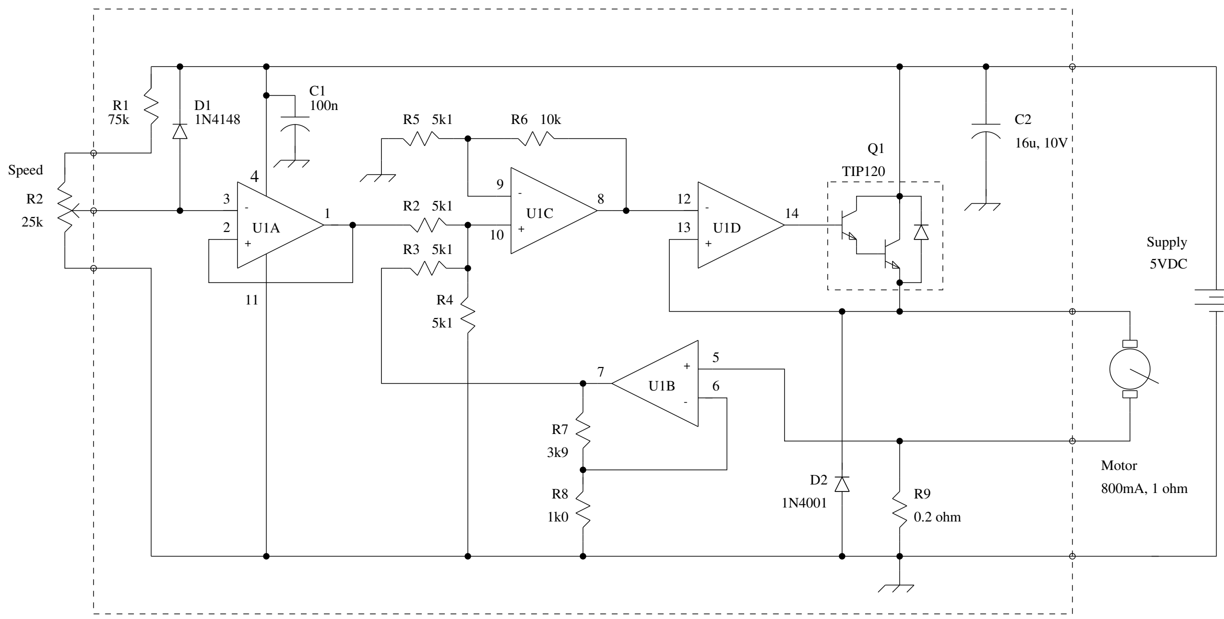


Figure 3: Schematic Symbols



NOTES:

Q1 may be any NPN type TIP series Darlington transistor.

Q1 must be on a heat sink, area 3 inches squared

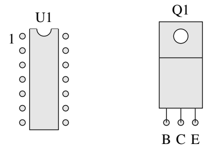
Op amps are from Quad Op Amp, LM324N or LM324AN (DIP package).

Recalculate resistor values to suit other components.

For D1, any small signal silicon diode may be substituted

For D2, any 1N4000 series diode or 1 amp diode may be substituted.

R9 is 5, 1 ohm resistors in parallel



Motor Speed Controller

Ryerson Polytechnic University
Peter Hiscocks, October 30, 1997
Print at 70%
Revision: 0.1
File: motcon2.fig

Figure 4: Motor Control Circuit

As time passes, you will build up a library of useful functions that you can add to the installed libraries or your own libraries. Or you can cut and paste figures from previous projects into new projects. For example, I have created an elaborate 68HC11 board schematic that has been pasted into various project schematics.

Why not use a Schematic CAD Program?

Programs for the design of printed circuit boards often include a schematic editor. The printed circuit layouts are fine, but I'm not a big fan of the schematic drawings they produce. The drawings are often difficult to read and do not show a logical flow from left to right, top to bottom. The points where wires attach to a symbol are often fixed, regardless of the position of the symbol, which makes for a confusing drawing. On the other hand, the CAD schematic interfaces directly with the layout function: if the schematic is correct, the layout will be correct.

It would be nice if it were possible to convert a schematic, drawn under XFig, into a form that could be imported into a printed-circuit board design program, such as ORCAD.

7 Incorporating Figures in Text

XFig includes an **Export** facility, in which the file description may be exported to different file formats. The most useful ones are **Latex Picture**, **Pictex**, **Postscript**, **Encapsulated Postscript** and bitmap formats such as **GIF**, **JPEG**, **TIFF** and several others.

The different file formats allow figures, drawn in XFig to be read by other computer programs and incorporated in various text documents.

The advantages, disadvantages and uses of these formats are as follows:

7.1 Latex Picture Format

The LaTeX text processing language has a limited number of drawing commands: circle, box, and line, and these may be used to draw simple pictures. For the **Latex Picture** format, XFig copies out a description of the picture in commands that can be directly processed by LaTeX. The file may be called into the main text document with the LaTeXcommands:

```
\begin{figure}[htb]
\begin{center}
\input{figures/cooling-function.latex}
\end{center}
\caption{Cooling Functions}
\label{fig:cooling-function}
\end{figure}
```

where the figure to be incorporated is in the sub-directory *figures* and is named *cooling-function.latex*. The figure will be centred, have a caption **Cooling Functions**, and may be referred to in the text by its symbolic name, **fig:cooling-function**. (That way, the program numbers the figures automatically, and the author doesn't have to be concerned with that chore.)

Latex figures are processed quickly, which is not the case for some other formats. One big advantage of the **Latex Picture** format is that any LaTeX symbol or mathematical expression may be incorporated in the picture. In electrical engineering, one quite often needs such symbols as Ω and μ on drawings. This is accomplished by writing `$\Omega$` and `$\mu$` on the diagram where they are required. If the **Text Flag** for that text is set to

Special, then it will be incorporated verbatim in the description of the drawing and LaTeX will put the correct symbol on the drawing.

The Text Flag may be accessed by selecting the XFig **Edit** function and then selecting some text. The edit dialog box then pops up and includes a switch that can make the text **Special**.

When the **Text** drawing mode is selected, the **Special** switch is accessible under the **Text Flags** control at the bottom of the drawing canvas.

7.2 Pictex Format

There are certain drawing elements that LaTeX cannot cope with directly. For example, lines at certain slopes, circles of certain diameter, and continuous spline curves are not allowed. To cope with this, the **Pictex** format does all drawing by placing a series of points. As a result, the the **Pictex** format can describe extremely complex images, but there is a price to pay. The **Pictex** format is both larger and slower to process (*much* slower) than the Latex format. A document with many Pictex inclusions can bring a 486-66 to its knees, so it is best to use Pictex only when needed.

The Pictex format has the advantage that, like the Latex format, it can incorporate LaTeX symbols directly.

The incantation to include a Pictex image is exactly the same as shown above for a Latex image, with the difference of the file name suffix, which is *.pictex*, of course.

The start of the LaTeX document must include the statement

```
\usepackage{piclatex}   % Pictex drawing package
```

so that it includes the necessary macros to process pictex drawings.

(If you use this package, you will get these error messages at the beginning of processing the document:

```
l.1444 \setplotsymbol({\fiverm .})%       ** initialize plotsymbol
l.1730 \setshadesymbol ({\fiverm .})%     ** initialize plotsymbol
```

Simply press return to get past this: the messages may be ignored.)

7.3 Postscript Format

Many printers accept postscript formatted files directly. The **Ghostview** previewer may be used to observe a postscript file, and can print selected pages from the postscript file to a printer. Thus, if you were creating a document to be taken to another printer and printed, you would probably export to the postscript format.

When exporting as postscript, always preview the image using Ghostscript, to ensure the image is the right size and orientation.

Postscript images are not directly importable into documents: that is the function of the Encapsulated Postscrip-t format.

7.4 Encapsulated Postscript

There is very little difference between a postscript file and an eps file: only the difference of the *bounding box* statement. The preamble of the postscript file says

```
%%BoundingBox: 65 242 547 550
```

The preamble of the eps file says

```
%%BoundingBox: 0 0 482 308
```

The postscript file specifies an absolute position for the image: the eps file is intended to be offset by some amount, determined by the program incorporating the diagram. This is useful to know, since one has often to convert from one form to another, and this can be done by editing the Bounding Box statement. The other difference is that the PostScript file contains a *showpage* command which tells the printer to actually print the page after rendering it.

Encapsulated Postscript files may be incorporated in a LaTeX drawing with an incantation like the following:

```
\begin{figure}[htb]
\begin{center}
\epsfig{file=figures/symbols.eps}
\end{center}
\caption{Schematic Symbols}
\label{fig:symbols}
\end{figure}
```

where the diagram is in the *figures* subdirectory and is named *symbols.eps*.

The start of the LaTeX document must also include the statement:

```
\usepackage{epsfig}
```

to tell LaTeX to incorporate the necessary macros to process eps documents.

For reasons noted below, the processing of eps images is instant because LaTeX does not process them. It simply makes a note of the location of the drawing and its size.

The big disadvantage of encapsulated postscript format is that it cannot easily incorporate LaTeX symbols: the XFig **Special** flag does not work. There **are** ways to get around this, but they are complicated and, in my opinion, it is easier to simply use the *Pictex* format when the diagram is complex and LaTeX math or greek symbols must appear on the diagram.

The Case of the Missing Pictures

There is one important thing to understand about incorporating eps documents in a LaTeX document. When you process LaTeX , it produces a *.dvi* (Device Independent) file. Then this file may be previewed using the viewer program *XDVI*, which shows very nice images of the generated text and its included pictures and drawings.

Now, the eps files that constitute the images in the document are not actually processed into the text of the *.dvi* file. The *.dvi* file includes pointers to the eps files (and bounding box information), and *XDVI* simply goes to those files when it needs to display a picture. This is fine: if you have an eps image being shown by *XDVI* and you edit the image in some way, then that edit will be reflected immediately in the *XDVI* display: you don't have to re-LaTeX the file.

However, if you were to carry the *.dvi* file to another location and then display or print the file, the images would not be present unless you explicity brought them along.

The solution to this is to ensure that any document containing eps images is converted to postscript format before transporting it. The program *dvips* is commonly used for this, and it inhales both the *.dvi* text file and the *.eps* image files, and generates a composite postscript (*.ps*) file. This postscript file is now the final material, text and pictures, and it can be previewed using *ghostview*.

7.5 GIF Format

I have not actually used this facility (version 2.1 doesn't include it) but it should make it possible to incorporate XFig pictures in Windows-based programs such as WordPerfect or Microsoft Word.

Microsoft Word claims to be able to import eps files, but it requires a TIFF-based preview bitmap in the eps file to view it on the screen. It will, however print correctly even without a preview.

8 Printed Circuit Board Design: Filled Areas

XFig can serve as a printed circuit board layout tool for simple PC board designs, when most of the circuit is discrete components. It's not really up to a complex design or one where a large number of IC's are involved, but for analog circuits or preliminary layout of a design, it's quite useful.

The first hurdle is the scale factor. Integrated circuits and electronic components are laid out on a hundred-mil grid: 0.1 inches or some multiple thereof. The XFig grid, on the other hand, is sixteen divisions to an inch. (This refers to version 3.2 patch level 1 of XFig, and later versions may differ.)

One solution is to define two XFig divisions as 0.1 inches. They are actually 0.125 inches, so the drawing must be scaled by a factor of 0.8 to produce a full-sized print. Fortunately, there is a **Magnification** function on the **Print** dialog box, which can be used to print at a magnification of 80%.

To draw a copper area on a PC board, use one of the closed polygon functions. When one of these is selected, a **fill style** selector appears on the bottom line of the screen. The default fill pattern is **none**. Pop this control open and then select a desired fill pattern. (To save ink, choose one of the less dense patterns.)

Now you're ready to create the design.

Use the outlines from the *Electrical Library* file. Move the packages around until you have what looks like a feasible layout.

Each connection on the PC board, between two or more pins, is an *island* of copper. To create an island, start by drawing an arbitrary shaded polygon. Don't worry about the shape or size at this point, just get some polygon drawn. You can use the smooth-edged polygon or the one composed of straight line segments.

There are three **Edit Mode** tools to modify the shape of this polygon: move point, add point, delete point. When you select one of these tools, the vertices of the polygon are shown as handles. Using the three tools, you can modify and drag the polygon into any shape you want. Adding and moving points allows you to spread the polygon to the required locations. Removing points allows you to undo mistakes or clean up busy areas of the drawing.

When the board is finally laid out, print the drawing at 80% (full size) and then use a copier to print onto an adhesive film material. Peel off the paper backing and stick the now-printed film to a bare copper board. (It's important that the board copper surface be clean and free of contaminants. Polish the board with fine steel wool until it shines. Then wipe the board with nailpolish remover to remove any chemical contaminants. It's also critical that the adhesive film be well attached to the copper, and not contain any air bubbles.)

Using the printed pattern as a guide, use a razor knife to cut away the film where the board should be etched. When this is complete, immerse the board in ferric-chloride etchant to create the final board.

An example drawing of a board made with this technique is shown in figure 5.

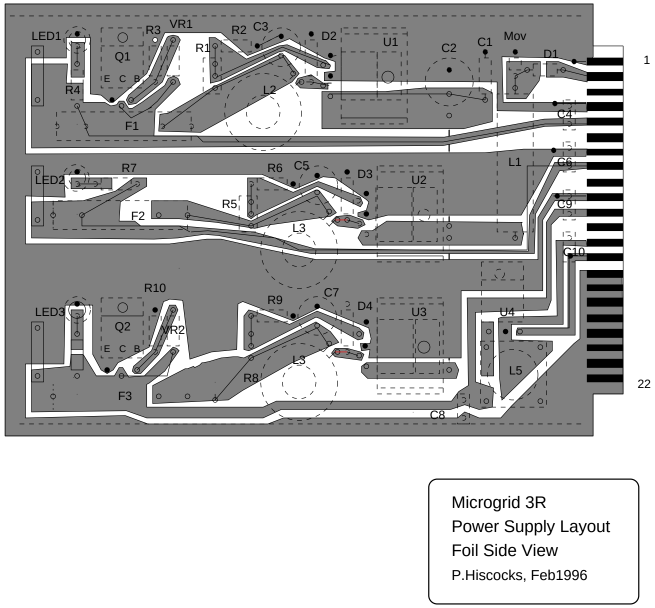


Figure 5: Power Supply Layout

9 Perf Board Layout

Prototype electronic circuits are often constructed on unclad perforated board. Wire wrap sockets and discrete components are glued to the board, and then wires run between the IC and discrete component pins.

Alternatively, some types of prototype board such as Veroboard have copper strips on the wiring side that can be used to make connections. You install wire jumpers running North-South on the component side of the board and connect these to copper traces on the wiring side to run East-West. The copper traces are broken at various points, using a twist drill to eat away the copper. The results are compact, reliable and attractive.

XFig is useful in laying out a prototype circuit of this type. The parts are positioned on the drawing and lines drawn to indicate the wiring connections that will take place. One colour of line can be used to indicate wires on the component side, and a second colour to indicate copper connections on the wiring side. Parts and connections can be moved over and over until the design is optimized for neatness and size. Copper run breaks can be indicated on the drawing.

When the design is finalized, it can be printed at full size and glued to the perf board. The printed pattern is a very useful guide to the position of components, connections and copper break points. This greatly simplifies the construction and debugging task: building the board becomes an exercise in following the drawing pattern.

An example design of this type is shown in figure 6. Notice that the board consists of two identical circuits. This is where a drawing program like XFig is really useful: generating the second layout is simply a matter of copying the original.

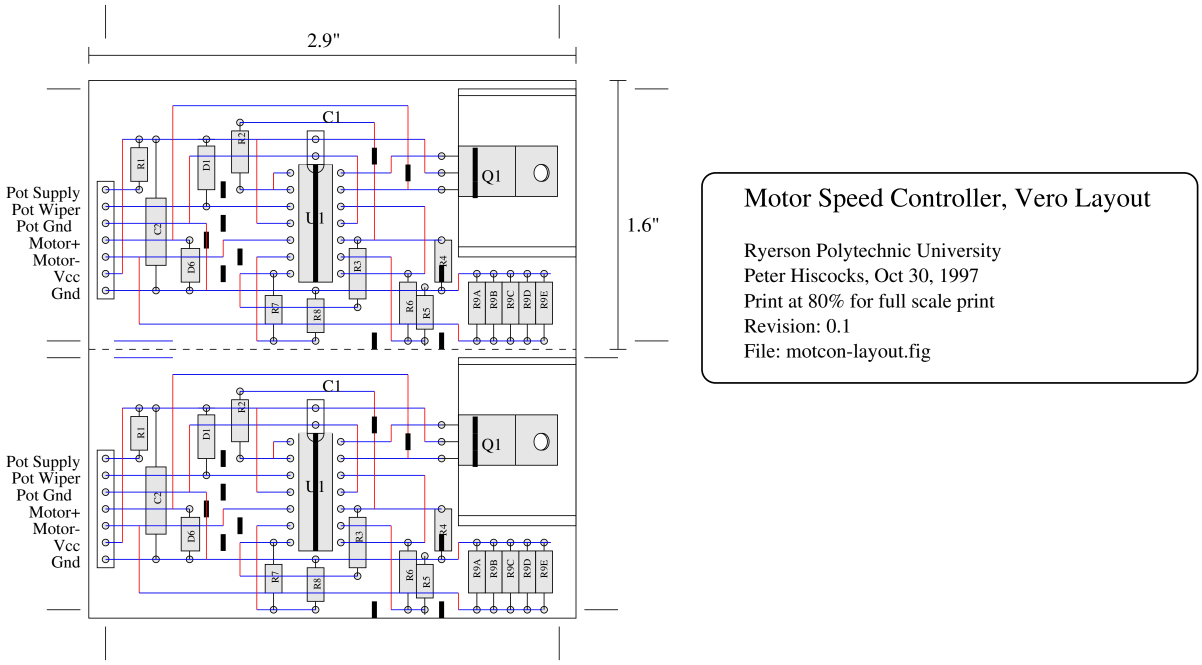


Figure 6: Veroboard Layout

10 Screen Capture to Xfig

The program **xv** is another wonderfully useful program available in the Unix world. It is a general purpose image capture and manipulation program.

For this document, we needed images of the toolbars of the XFig program. Fortunately, among its many useful features, **xv** has a *grab* function which allows one to capture an area of the video screen.

First, **xv** was used to grab the toolbar area of an XFig window. Then, using other **xv** tools, the image was then de-coloured (converted to grey scale) and thresholded to a black and white image. The **xv** '**crop** tool was used twice on the black-white image of the XFig toolbars to make to the draw and edit toolbars, which were saved as postscript (*ps*) files. (There is no option to save an image as an eps file, so postscript is the next best thing.)

Each postscript file was then loaded into XFig, marked up with textual annotations, and saved as an *eps* file. Ghostview was used to preview these eps files to ensure that everything was there.The eps file was then called into the LaTeX document to make the desired figure.

Astonishingly, it all worked.

You can also capture windows and areas of the screen using the **Picture Object** feature, but XFig unmaps all of its windows from the screen during this operation to 'get out of your way' so you would have to run two XFigs at the same time and use one to capture parts of the other.

11 Making Overheads using Xfig

Xfig is a great tool for creating the overheads for a slide show. Often, the situation is this: you have created a document on a particular topic and now you need to do a presentation. Each of the overheads is to be a summary of one of the important ideas in the original paper.

Luckily, you don't need to retype each of the overheads. The text can be cut and pasted from the original document. The facility to do this is really useful, but really well hidden.

You need to ensure that the 'text-paste' operation is set up properly. I can't speak for other machines, but on mine (the 'Manhattan' distribution of RedHat) the function of this key is defined in the file

```
/usr/X11R6/lib/X11/app-defaults/Fig
```

Edit this file, looking for the following stanza:

```
! make the F20 key paste text in the canvas
Fig*canvas.translations: #override \n\
                          <Key>F1: PasteCanv()\n
```

In this particular case, the text-paste operation is defined to be the 'F1' key. Originally, it was 'F18' or some other strange value and I had to change it. You can change it to 'F1' or some other key of your choice.

Now you're ready for a paste operation:

- Select the text widget in Xfig.
- Select any of the Xfig text attributes, such as Font and Size that you want the pasted text to have.
- Position the text cursor where you want the text to appear in the Xfig window.
- Set up the text to be pasted so that it's visible in an XTerm.

- In the XTerm window, use the left mouse button to highlight the text to be copied.
- Move the focus back to the Xfig window and press button F1.
- The text should paste correctly into the Xfig window.

An example is shown in figure 7. I selected the 'Bookman-LightItalic' 12 point font in Xfig. Then I copied the text from a previous paragraph of this paragraph into Xfig. I outlined the text with a box and saved it as an encapsulated-postscript file at 70% magnification. Then I added the instructions to this LaTeX document to import the eps file as figure 7.

Xfig is a great tool for creating the overheads for a slide show. Often, the situation is that you have created a document on a particular topic. Now you need to do a presentation. Each of the overheads is a summary of one of the important ideas in the original paper.

Figure 7: Pasted Text

Initially, the box was the wrong size. I had to adjust it in Xfig and re-export the drawing until it was correct. If you're using the xdvi viewer, each time you resave the eps file, the new version appears automatically in xdvi. (You do have to close and open the xdvi window.) You don't have to reprocess the LaTeX document.

Now, how would you assemble a bunch of these slides as a slide show?

One method would be to create a LaTeX document that imports each overhead as a separate page in the document. As you made up the slide show, you could process it as a LaTeX document and preview it using the *xdvi* previewer. If you need to incorporate mathematics into the overheads, paste in the LaTeX text, mark it as 'special' text in Xfig, and export it to a file format such as Pictex that understands math symbols. When the entire slide show is assembled, use *dvips* to convert it to a postscript document. This is the completed slide show. Use *ghostview* to present the overheads one after another or in whatever order is appropriate.

12 Configuring Xfig

The installation and configuration process has three stages:

- Install the program
- Fit the screen size
- Get colours to work.

12.1 Installation

On a Redhat Linux system, or some other system that uses the RPM (Redhat Package Management) system, the installation steps are as follows:

- Mount the CDROM that came with the distribution.
- Find the Redhat directory

- In that directory, find the `xfig` package, which will be named something like `xfig3.2.2.rpm`.
- Install it with the RPM command:

```
rpm -i xfig2.2.2.rpm
```

(Grizzled old Unix veterans now heard muttering about how easy people have it these days, not having to use MAKE files etc etc.) The `xfig` package is also available on the web. You could use Google to find it.

A Slakware system is similar, except that packages end in a `.tgz` extension and the command is `installpkg`.

If you are generating outputs in latex or other formats for incorporation in documents, you will also need to install the `transfig` package. This handles the translation of the document description from `fig` into various other formats.

12.2 Screen Size

It is not uncommon when starting xfig for the first time to find that the window exceeds the screen size and the colours are missing. Fortunately, `xfig` and `X Window` are extremely configurable, so these glitches can be fixed.

The screen size may be reduced to fit by means of the command line options that set the width of the canvas, the height of the canvas, and the number of buttons per row on the control panel. (Refer to the `man` entry for xfig for a list of all the options.)

For example, to make xfig fit on my Toshiba laptop, for which the screen resolution is 800x600 pixels, I used the command line

```
xfig -pwidth 8 -pheight 5.5 -but_per_row 3
```

This squishes the canvas down to 8 inches by 5.5 inches and stacks the buttons three to a row.

To avoid having to type in this command line every time you start xfig, you can create an alias in one of the shell configuration files. In my case, a linux system, the shell is Bash and I added the following line to the `.bashrc` file:

```
alias xfig='xfig -pwidth 8 -pheight 5.5 -but_per_row 3 &'
```

The ampersand in the command line puts the xfig process in the background so that you get the command prompt back after starting xfig. (Remember, to list files that begin with a *dot*, you must use the `-a` option to `ls`, as in `ls -a`.)

To get this alias to take effect, you need to logout and login again or use the `source` command to get Bash to read the configuration file, as in `source .bashrc`.

Typing 'xfig' now invokes this entire command line automatically.

Incidentally, you can find out which shell your system is using with the command

```
echo $SHELL
```

If you are using some shell other than Bash, you'll have to read the manual entry on that shell to determine the names of the configuration files, their configuration files, and syntax of the entries.

12.3 Colour

When xfig is installed, its X Window configuration files will also have been automatically installed in an `application defaults` directory. On my Redhat system, this is

```
/usr/K11R6/lib/X11/app-defaults
```

and on the Slakware system

```
/usr/K11R6/lib/app-defaults
```

There, you will find configuration files for many X Window programs, among which should be `Fig` and `Fig-color`. (Non Americans should take special notice the spelling of *color*, which Brits and Canadians spell *colour*).

The objective is to get `xfig` to read the `Fig-color` file when it starts up.

When X Window starts up, it executes a program `xinit` (see the `man` entry for `xinit`). This program looks for a file named `.xinitrc` in the user's home directory. The `.xinitrc` script is an important file – it specifies the X window manager and the various X windows programs that appear on the desktop at startup. The `.xinitrc` file in turn identifies a configuration file named `.Xdefaults`. To get xfig to read it's application default file `Fig-color`, add to the `.Xdefaults` text the magic line:

```
Fig*customization: -color
```

In some cases, the X configurations are contained in a file named `.Xresources`. In that case, edit the `.xinitrc` file to read that instead.

Restart X Windows. Depending on your system, simply pressing `<ctrl><alt><backspace>` will take you back to the logon screen. Log on as normal and start `xfig`. It should be in glorious colour.

On some machines, pressing `<ctrl><alt><backspace>` takes you back to the command line dumb terminal screen. Run `startx` to restart X Window.

If there is no file `.xinitrc` in your home directory, use the `locate` command to search for `.xinitrc` or `xinitrc`. There will probably be an example somewhere in the system that can be copied into your home directory and edited to suit.

If there is no file `.Xdefaults` in your home directory, simply create it, containing the magic line given above.

12.4 References

To learn the details of BASH and its configuration files, there is:

Learning the BASH Shell, 2nd Edition
Cameron Newham and Bill Rosenblatt
O'Reilly & Associates Inc, 1998

For more information on using and configuring the X Window System, you might want to read

X Window System User's Guide, Volume 3
Valerie Quercia and Tim O'Reilly
O'Reilly & Associates Inc, 1993