

# An Approach to Drawing Circuit Diagrams for Text Books

M.P.Maclenan & G.M.Burns  
South Bank Polytechnic, Borough Road, London  
email: burnsm@vax.southbank-poly.ac.uk

## Abstract

The development of a library of pictograms is described. The pictograms are defined using macros embodied in P<sub>I</sub>C<sub>T</sub>E<sub>X</sub> and used to create an application to enable high definition circuit diagrams to be easily included in T<sub>E</sub>X documents.

## Introduction

The problems of including pictures within a T<sub>E</sub>X document have been considered by several people e.g. Norris *et al* and recently by Rogers and by Childs *et al*. Norris imports a PostScript file, Roger's paper suggests creating a mini-graphics package while Childs argues for a specialised hardware. *Textures* which we use on Apple hardware provides a `\special{picture#}` to be included which allows for such combinations of pictures and text to be created as



TWICKENHAM  
RUGBY  
FOOTBALL  
CLUB

This facility is excellent if mere representation is all that is required, but there is little flexibility once the picture is imported to T<sub>E</sub>X and making changes becomes tedious. We consider here a use of P<sub>I</sub>C<sub>T</sub>E<sub>X</sub> to create an application, referred to as C<sub>I</sub>R<sub>T</sub>E<sub>X</sub> for brevity, which might be used to produce electronic circuit diagrams. The use of P<sub>I</sub>C<sub>T</sub>E<sub>X</sub> was considered to have the following advantages:

1. it maintains device independence
2. it allows a mixture of good graphic pictograms with the elegance of T<sub>E</sub>X typesetting, ideal for circuit diagrams
3. compositors need only be conversant with T<sub>E</sub>X and the simple process of creating diagrams embodied in C<sub>I</sub>R<sub>T</sub>E<sub>X</sub> to produce diagrams.

## Concepts Used in C<sub>I</sub>R<sub>T</sub>E<sub>X</sub>

The simple principle adopted is that small, recurring objects are defined in a co-ordinate system using

1pt as the base dimension of the space. From these simple objects, small pictograms are defined and from these pictograms, complete circuit diagrams can be drawn. Thus

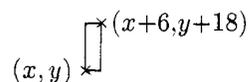
`\` and `|` are used to make

Section 9.2 of The P<sub>I</sub>C<sub>T</sub>E<sub>X</sub> Manual discusses the use of `\setdimensionmode` to `\put` objects in an absolute space with respect to a given origin. However, by working consistently with 1pt as the base dimension, co-ordinate space was eventually preferred. There are two possible approaches to defining the objects and pictograms, in particular:

1. to define the objects in terms of the absolute coefficients  $(x, y)$  of the final diagram
2. to build up a box using co-ordinates relative to a given origin within the box and then to `\put {box} at x y` where  $(x, y)$  is now absolute within the final diagram.

In the final version of C<sub>I</sub>R<sub>T</sub>E<sub>X</sub>, both approaches are used. It was initially decided to relate *all* objects and pictograms to an origin at their bottom left hand corner, but eventually it was realised that it was convenient if the origin of some of them was logical rather than predefined in this way.

For example, consider a simple vertical rectangle used, according to BS3939, to represent a resistor.



Using approach A, `\Rv` can be defined by

```
\def\Rv#1#2{% #1=$x$, #2=$y$
\cxone=#1 \advance\cxone by 6
\cyone=#2 \advance\cyone by 18
\putrectangle corners at
{#1}{cyone} and {cxone}{#2}}
```

It is then used simply as `\Rv{x}{y}` where  $x, y$  are absolute coordinates in the space of the final diagram.

Approach B follows the suggestion made in Section 8.2 of the `PiCTEX` manual on how to include pictures in pictures. Thus `\Rv` can be defined by

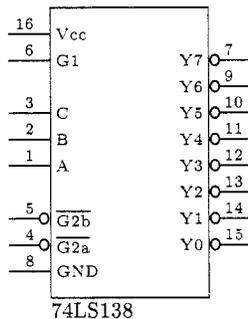
```
\def\Rv{\%
\putrectangle corners at 0 18 and 6 0
}
```

`\Rv` is subsequently placed in the final picture using `\put {\Rv} [bl] at x y`. In this case, approach B is preferred because it is simpler and use of `\put` allows for an  $x_{shift}$  and/or a  $y_{shift}$  so that the resistor can be located at the known end of a vertical line and offset 3pt to place the origin. Note that we found that the inclusion of `\beginpicture ... \endpicture` as suggested in Section 8.2 was found to be unnecessary in the definition.

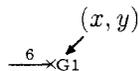
An example of logical placement is

```
\label#1#2#3#4{\% #1=$x$ #2=$y$
% #3=pin number #4=label $\cdots$}
```

used to draw this IC.



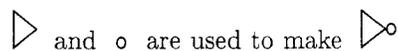
Here, each label is placed as



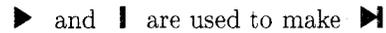
The parameter `#4` can contain any text, including symbols, so that the depth of this sub-box is unknown. Since its depth will define the depth of `\label`, logical placement is essential to maintain alignment.

Once these concepts were understood, a library of objects and pictograms was developed.

### Tricks



Initially, the inversion bubble that converts a buffer to an inverter was constructed using `\circulararc`, which took a long time to compile and used considerable amounts of memory. It was noticed that a lower case `o` in courier font is circular, and at 10pt, of a perfect size to be used as `\bubble` which can be `\put` immediately!



Infill of shapes proved to be difficult. The diode triangle is constructed by successively `\puting` on a centerline vertical rules of diminishing height and offset 0.4pt to each other. The down arrow of the *npn* transistor is drawn similarly by `\plotting` a series of lines radiating from the tip of the arrow.

Labels of some signals are drawn with a bar over the symbol to indicate that the signals are active low. Given that the symbols are of unknown length, this was solved by

```
\def\actlow#1{\hbox{#1}
\setbox1=\hbox{#1}\kern-\wd1
\raise5.5pt\hbox{\%
\vrule height 0.4pt width \wd1
}}
```

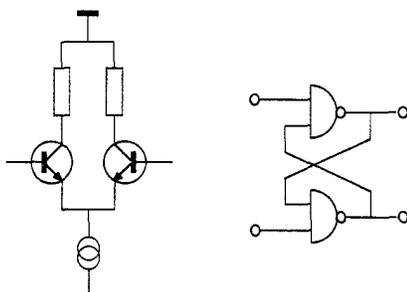
### Problems

It must be stressed that problems discussed here relate to our use of `PiCTEX` in trying to create an application for which it was not intended and not to `PiCTEX` itself.

1. `\linethickness` has no effect on `PiCTEX` macros producing curves so that, for example, the circular outline used to draw the transistor could not be changed.
2. `\loop ... \repeat` does not seem to work within any of our defined macros which involve use of `PiCTEX`.
3. During development, variable names were chosen which by chance coincided with names used in `PiCTEX`. To avoid repetition of the problems this caused, all `CiRTEX` variables now start with a c e.g. `\cxone`
4. Our initial use of `\setbox1` gave very odd results until the valuable advice to always use `\newbox` was remembered (which produced `\box22!`).
5. Our attempt at object-orientated development implies deeper and deeper grouping which may contribute to greater use of memory than is necessary.

## Conclusion

The extent of development now allows us to quickly draw diagrams like these,



which can be totally integrated into a  $\text{\TeX}$  document. Indeed, it is now possible to produce diagrams such as that in the Appendix, but this takes 37 kbytes of memory and over 6 minutes to compile on a Macintosh II! The reason for this is simple, that  $\text{\P\TeX}$  must calculate many of the basic shapes of the objects. A tantalising reference is made on page 389 of the *The  $\text{\TeX}$ book* to a font  $\backslash\text{qc}$  which contains four quarter circles as characters but no hint is given as to its availability. If a font which contains *all* the basic object shapes which we use could be developed and combined with the computational power implied in the use of  $\backslash\text{put}$  within  $\text{\P\TeX}$ , then as Rogers suggests, we would be able to produce a package capable of easy production of circuit diagrams of suitable quality to be included in text books.

## Bibliography

- Norris & Oakley. 1988. "Electronic Publishing,"  
 *$\text{\TeX}$ 88 Exeter Proceedings*.  
Rogers, David. 1988. *TUGboat* 10(1), April.

