

Creating multipole circuit components for circuitikz.

John Kormylo

Creating a node (multipole) circuit component consists of creating a shape, as described by the Tikz manual (p 1026) under `\pgfdeclareshape`. The main difference is that you need to add anchors for all the pins so that you can connect to them later. The good news is that the size of your component does not expand to fit the label. The bad news is that everything is done using graphics primitives, not the usual `\draw` type commands.

The following was used to create an 8 pin IC. This should probably be stored in a separate file and read using `\input` somewhere in the header.

```
\pgfdeclareshape{ic8pin}{
\anchor{center}{\pgfpointorigin} % within the node, (0,0) is the center

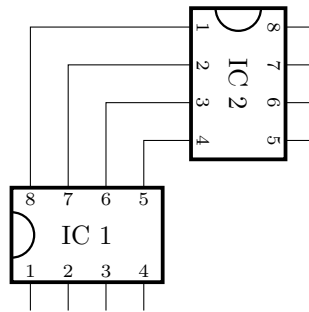
\anchor{text} % this is used to center the text in the node
{\pgfpoint{-.5\wd\pgfnodeparttextbox}{-.5\ht\pgfnodeparttextbox}}

\savedanchor\icpina{\pgfpoint{-.75cm}{-.625cm}} % pin 1
\anchor{p1}{\icpina}
\savedanchor\icpinb{\pgfpoint{-.25cm}{-.625cm}} % pin 2
\anchor{p2}{\icpinb}
\savedanchor\icpinc{\pgfpoint{.25cm}{-.625cm}} % pin 3
\anchor{p3}{\icpinc}
\savedanchor\icpind{\pgfpoint{.75cm}{-.625cm}} % pin 4
\anchor{p4}{\icpind}
\savedanchor\icpine{\pgfpoint{.75cm}{.625cm}} % pin 5
\anchor{p5}{\icpine}
\savedanchor\icpinf{\pgfpoint{.25cm}{.625cm}} % pin 6
\anchor{p6}{\icpinf}
\savedanchor\icping{\pgfpoint{-.25cm}{.625cm}} % pin 7
\anchor{p7}{\icping}
\savedanchor\icpinh{\pgfpoint{-.75cm}{.625cm}} % pin 8
\anchor{p8}{\icpinh}

\foregroundpath{ % border and pin numbers are drawn here
\pgfsetlinewidth{0.05cm}
\pgfpathrectanglecorners{\pgfpoint{1cm}{.625cm}}{\pgfpoint{-1cm}{-.625cm}}
\pgfusepath{draw} %draw rectangle
\pgfsetlinewidth{0.03cm}
\pgfpathmoveto{\pgfpoint{-1cm}{-.3cm}}
\pgfpatharc{-90}{90}{.3cm}
\pgfusepath{draw} %draw semicircle
\pgftext[bottom,at={\pgfpoint{-.75cm}{-.55cm}}]{\scriptsize 1}
\pgftext[bottom,at={\pgfpoint{-.25cm}{-.55cm}}]{\scriptsize 2}
\pgftext[bottom,at={\pgfpoint{.25cm}{-.55cm}}]{\scriptsize 3}
\pgftext[bottom,at={\pgfpoint{.75cm}{-.55cm}}]{\scriptsize 4}
\pgftext[top,at={\pgfpoint{.75cm}{.55cm}}]{\scriptsize 5}
\pgftext[top,at={\pgfpoint{.25cm}{.55cm}}]{\scriptsize 6}
\pgftext[top,at={\pgfpoint{-.25cm}{.55cm}}]{\scriptsize 7}
\pgftext[top,at={\pgfpoint{-.75cm}{.55cm}}]{\scriptsize 8}
}}
```

The anchors “center” and “text” are required for any node. You will note that there is a TeX macro version of the anchor and an anchor name as well. Name duplication between shapes doesn’t seem to be a

problem. The big ugly section starting with `\foregroundpath` is where the node is drawn. The aspect ratio is based on the physical dimensions of the Dual Inline Pin (dip) standard.



```

\begin{center}
\begin{circuitikz}
\draw (1,1) node[ic8pin] (IC1) {IC 1};
\draw (3,3) node[ic8pin,rotate=-90] (IC2) {IC 2};
\draw (IC1.p1) -- (0.25,0);
\draw (IC1.p2) -- (.75,0);
\draw (IC1.p3) -- (1.25,0);
\draw (IC1.p4) -- (1.75,0);
\draw (IC1.p5) |- (IC2.p4);
\draw (IC1.p6) |- (IC2.p3);
\draw (IC1.p7) |- (IC2.p2);
\draw (IC1.p8) |- (IC2.p1);
\draw (IC2.p5) -- (4,2.25);
\draw (IC2.p6) -- (4,2.75);
\draw (IC2.p7) -- (4,3.25);
\draw (IC2.p8) -- (4,3.75);
\end{circuitikz}
\end{center}

```

You will need to read *Part IX The Basic Layer* from the Tikz manual and practice drawing shapes with the primitives.

Changing the Size

In the Circuitikz manual we see that we can change the size of components using

```
\ctikzset {bipoles/resistor/height =.6}
```

for example. It would be nice be able to do something like that with the 8 pin IC. Perhaps something like

```
\ctikzset {multipoles/pin spacing = 4mm}.
```

If you are going to change the pin spacing on one IC, you probably want to do it on all of them. Either that or do it on a node by node basis.

```
node[dip8,pin spacing=4mm]
```

The former is done often enough in the .sty files.

```
\ctikzset{multipoles/.is family}
\ctikzset{multipoles/pin spacing/.initial = 5mm} % default value
```

It turns out that node options work like to-path options, but they show up in the /tikz/ family with the styles and shapes. I hope no one wants to name a shape “pin spacing.”

```
\pgfkeys{/tikz/pin spacing/.initial = 0mm}
\pgfkeys{/tikz/pin spacing/.default = 0mm}
```

Defaulting to 0 means that if the node specification is given, it will be used. Otherwise, not.

All that remains is to scale the anchors and the drawing. It turns out that putting the resize code into a saved macro enables one to change the scale factor **before** defining the anchors. I also adjust the line thickness and pin numbers, but \tiny is as small as it gets.

```
\newlength{\IClen} % scale factor
\newcommand{\pinsize}{\ifdim\IClen<7mm \tiny \else \scriptsize \fi}

\pgfdeclareshape{dip8}{
\anchor{center}{\pgfpointorigin} % within the node, (0,0) is the center

\anchor{text} % this is used to center the text in the node
{\pgfpoint{-.5\wd\pgfnodeparttextbox}{-.5\ht\pgfnodeparttextbox}}

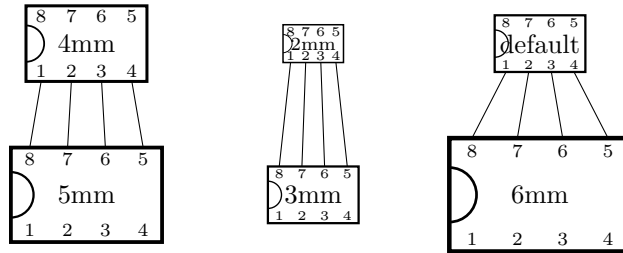
\savedmacro{\resize}{ % called automatically
\setlength{\IClen}{\pgfkeysvalueof{/tikz/pin spacing}} % from node[]
\ifdim\IClen=0mm \setlength{\IClen}
{\pgfkeysvalueof{/tikz/circuitikz/multipoles/pin spacing}} \fi % from \ctikzset{}
\setlength{\IClen}{2\IClen} % multiply pin spacing by 2 (default = 1cm)
}
\savedanchor\icpina{\pgfpoint{-.75\IClen}{-.625\IClen}} % pin 1
\anchor{p1}{\icpina}
\savedanchor\icpinb{\pgfpoint{-.25\IClen}{-.625\IClen}} % pin 2
\anchor{p2}{\icpinb}
\savedanchor\icpinc{\pgfpoint{.25\IClen}{-.625\IClen}} % pin 3
\anchor{p3}{\icpinc}
\savedanchor\icpind{\pgfpoint{.75\IClen}{-.625\IClen}} % pin 4
\anchor{p4}{\icpind}
\savedanchor\icpine{\pgfpoint{.75\IClen}{.625\IClen}} % pin 5
\anchor{p5}{\icpine}
\savedanchor\icpinf{\pgfpoint{.25\IClen}{.625\IClen}} % pin 6
\anchor{p6}{\icpinf}
\savedanchor\icping{\pgfpoint{-.25\IClen}{.625\IClen}} % pin 7
\anchor{p7}{\icping}
\savedanchor\icpinh{\pgfpoint{-.75\IClen}{.625\IClen}} % pin 8
\anchor{p8}{\icpinh}

\foregroundpath{ % border and pin numbers are drawn here
\pgfsetlinewidth{.05\IClen} % line thickness
\pgfpathrectanglecorners{\pgfpoint{\IClen}{.625\IClen}}{\pgfpoint{-\IClen}{-.625\IClen}}
\pgfusepath{draw} %draw rectangle
\pgfsetlinewidth{.03\IClen} % line thickness
\pgfpathmoveto{\pgfpoint{-\IClen}{-.3\IClen}}
\pgfpatharc{-90}{90}{.3\IClen}
\pgfusepath{draw} %draw semicircle
```

```

\pgftext[bottom,at={\pgfpoint{-0.75\IClen}{-0.55\IClen}}]{\pinsize 1}
\pgftext[bottom,at={\pgfpoint{-0.25\IClen}{-0.55\IClen}}]{\pinsize 2}
\pgftext[bottom,at={\pgfpoint{0.25\IClen}{-0.55\IClen}}]{\pinsize 3}
\pgftext[bottom,at={\pgfpoint{0.75\IClen}{-0.55\IClen}}]{\pinsize 4}
\pgftext[top,at={\pgfpoint{0.75\IClen}{0.55\IClen}}]{\pinsize 5}
\pgftext[top,at={\pgfpoint{0.25\IClen}{0.55\IClen}}]{\pinsize 6}
\pgftext[top,at={\pgfpoint{-0.25\IClen}{0.55\IClen}}]{\pinsize 7}
\pgftext[top,at={\pgfpoint{-0.75\IClen}{0.55\IClen}}]{\pinsize 8}
}}

```



The above was generated using.

```

\begin{center}
\begin{circuitikz}
\draw (0,0) node[dip8](a){5mm};
\ctikzset{multipoles/pin spacing = 4mm}
\draw (0,2) node[dip8](b){4mm};
\draw (a.p5) -- (b.p4) (a.p6) -- (b.p3) (a.p7) -- (b.p2) (a.p8) -- (b.p1);
\ctikzset{multipoles/pin spacing = 3mm}
\draw (3,0) node[dip8](c){3mm};
\draw (3,2) node[dip8,pin spacing=2mm](d){\scriptsize 2mm};
\draw (c.p5) -- (d.p4) (c.p6) -- (d.p3) (c.p7) -- (d.p2) (c.p8) -- (d.p1);
\draw (6,0) node[dip8,pin spacing=6mm](e){6mm};
\draw (6,2) node[dip8](f){default};
\draw (e.p5) -- (f.p4) (e.p6) -- (f.p3) (e.p7) -- (f.p2) (e.p8) -- (f.p1);
\end{circuitikz}
\end{center}

```

Note that “default” reverts to the last size specified by `\ctikzset`.