

Create and process options in \pgfdeclareshape

I'm working on a small library for Register Transfer Level (RTL) that I'll use a lot throughout my PhD thesis. I'm designing an operator symbol that can have a bit or a bus as input and/or output. If it is a bit, the connectors should be drawn as a thin line. If it is a bus, an ultra thick line. I thought of creating two options to be passed to the shape. I now how to create them, but not how to write the *if* statement to get the desired results. Following the relevant code fragments.

```
\tikzset{input type/.initial={bus}} % bus or bit string values
\tikzset{output type/.initial={bus}}

\pgfdeclareshape{operator}{
  ...
  \beforebackgroundpath{%
    % Connectors are always black
    \color{black}

    % This is the part I'm getting problems with
    % If input type is *bit*
    \pgfsetlinewidth{.5pt} % <-- ideally, it should be the thin option from tikz,
                          % but I don't know how to set it
    % If input type is *bus*
    \pgfsetlinewidth{1pt} % <-- should be ultra thick option

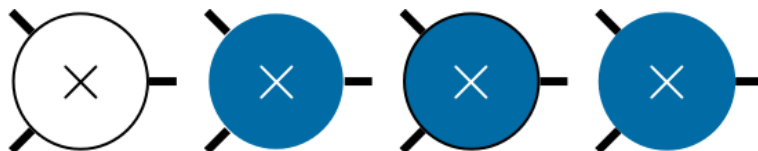
    % First input
    \pgf@process{\pgfutil@useanchor{operator}{a}}
    \pgfmoveto{\pgfpoint{\pgf@x}{\pgf@y}}
    \pgf@process{\pgfutil@useanchor{operator}{north west}}
    \pgflineto{\pgfpoint{\pgf@x}{\pgf@y}}

    % Second input
    \pgf@process{\pgfutil@useanchor{operator}{b}}
    \pgfmoveto{\pgfpoint{\pgf@x}{\pgf@y}}
    \pgf@process{\pgfutil@useanchor{operator}{south west}}
    \pgflineto{\pgfpoint{\pgf@x}{\pgf@y}}
    \pgfusepath{stroke}

    % If output type is *bit*
    \pgfsetlinewidth{.5pt}
    % If output type is *bus*
    \pgfsetlinewidth{1pt}

    % Output
    \pgf@process{\pgfutil@useanchor{operator}{r}}
    \pgfmoveto{\pgfpoint{\pgf@x}{\pgf@y}}
    \pgf@process{\pgfutil@useanchor{operator}{east}}
    \pgflineto{\pgfpoint{\pgf@x}{\pgf@y}}
    \pgfusepath{stroke}
  }
  ...
}
```

It seems to be a rather easy solution, but the manual is enormous and I didn't find any clues up till now. Thanks in advance for the help.



EDIT: Added the desired output for reference

```
{tikz-pgf} {shapes}
```

edited Nov 13 '15 at 14:31

asked Nov 13 '15 at 14:25

 **WDiniz**
62 7

1 Answer

Not a full answer, but illustrating a trick that can be used when defining shapes so that TikZ styles can be passed to the PGF layer for the `behindbackgroundpath`. This by-passes the need for convoluted `\if...\fi` statements in the shape definition as everything can be done using styles.

In this (edited) version the symbol is placed using a node `label`.

```
\documentclass[tikz,border=5]{standalone}
\makeatletter
\pgfdeclareshape{operator}{
  \nodeparts{
    \savedmacro\operatorparameters{%
      \pgfmathsetlength\pgf@x{\pgfkeysvalueof{/pgf/minimum width}}%
      \pgfmathsetlength\pgf@y{\pgfkeysvalueof{/pgf/minimum height}}%
      \pgfmathsetlengthmacro\radius{max(\pgf@x,\pgf@y)/2}%
      \addtosavedmacro\radius%
    }%
    \pgfmathsetlength\pgf@x{\pgfkeysvalueof{/pgf/outer xsep}}%
    \pgfmathsetlength\pgf@y{\pgfkeysvalueof{/pgf/outer ysep}}%
    \pgfmathsetlengthmacro\outersep{max(\pgf@x,\pgf@y)}%
    \addtosavedmacro\outersep%
  }
  \anchor{center}{\pgfpointorigin}%
  \anchor{north}{\operatorparameters%
    \pgfpointpolar{90}{\radius+\outersep}}
  \anchor{south}{\operatorparameters%
    \pgfpointpolar{270}{\radius+\outersep}}
  \anchor{east}{\operatorparameters%
    \pgfpointpolar{0}{\radius*sqrt(2)}}
  \anchor{west}{\operatorparameters%
    \pgfpointpolar{180}{\radius+\outersep}}%
  \anchor{north west}{\operatorparameters%
    \pgfpointpolar{135}{\radius*sqrt(2)}}
  \anchor{south west}{\operatorparameters%
    \pgfpointpolar{225}{\radius*sqrt(2)}}
  \anchor{north east}{\operatorparameters%
    \pgfpointpolar{45}{\radius+\outersep}}%
  \anchor{south east}{\operatorparameters%
    \pgfpointpolar{315}{\radius+\outersep}}%
  \behindbackgroundpath{%
    \operatorparameters%
    \pgfpathmoveto{\pgfpointpolar{135}{\radius}}%
    \pgfpathlineto{\pgfpointpolar{135}{\radius*sqrt(2)}}%
    \pgfpathmoveto{\pgfpointpolar{225}{\radius}}%
    \pgfpathlineto{\pgfpointpolar{225}{\radius*sqrt(2)}}%
    \pgfpathmoveto{\pgfpointpolar{0}{\radius}}%
    \pgfpathlineto{\pgfpointpolar{0}{\radius*sqrt(2)}}%
    \pgflib@sh@operator@connectors%
  }
  \backgroundpath{%
    \operatorparameters%
    \pgfpathcircle{\pgfpointorigin}{\radius}%
  }
}

\pgfkeys{/pgf/.cd,
  operator connectors/.store in=\pgflib@sh@operator@connectors,
  operator connectors=,
}

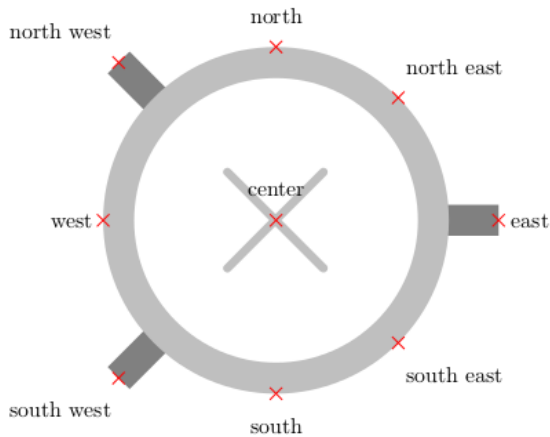
\tikzset{%
  operator connectors/.style={
    /pgf/operator connectors={\begingroup\tikzset{#1}\tikz@finish}
  },
  operator symbol/.style 2 args={
    label={\every operator symbol/.try,#1}center:{#2}}
},
```

This site uses cookies to deliver our services and to show you relevant ads and job listings. By using our site, you acknowledge that you have read and understand our [Privacy Policy](#), [Terms of Service](#), and our [Cookie Policy](#). Your use of Stack Overflow's Products and Services, including the Stack Overflow Network, is subject to these policies and terms.

```

\begin{tikzpicture}[minimum size=1cm]
\node [shape=operator,
  operator symbol={font=\Huge, scale=4, gray!50}{\times},
  operator connectors={draw=gray, line width=0.5cm},
  minimum size=2in, line width=0.5cm,
  draw=gray!50] (s) {};
\foreach \name/\anchor in {center/above, north/above, south/below,
  east/right, west/left, north east/above right, south east/below right,
  north west/above left, south west/below left}
  \draw [draw=red, shift=(s.\name)] node [\anchor] {\name}
    (-.1,-.1) -- (.1,.1) (-.1,.1) -- (.1,-.1);
\end{tikzpicture}
\end{document}

```



edited Nov 23 '15 at 10:34

answered Nov 13 '15 at 17:33



Mark Wibrow

60.2k 4 104 170

Very nice, thank you! I'll implement this later into my code. Nice way to process options. – **WDiniz** Nov 16 '15 at 9:16

Another thing, I noticed that you've drawn the operator symbol, but thing is it can vary. In the example, I used a multiplier, but it can be any of the four arithmetic operations or comparisons (<, >, = != etc.). As the shape is basically the same, it makes no sense to have one shape for each one, but having just the symbol vary. How to implement that in your version? Also, the input and output connector can be different, it is possible to have a input bus (thick style) and a output bit (thin style), it happens with comparisons. Thank you again for your help up till now. – **WDiniz** Nov 16 '15 at 9:28

1 @WDiniz, if the symbol is always going to be a LaTeX text symbol, then you could completely remove the `\beforebackgroundpath` and just add (for example) `label={text=white, font=\huge}center:${\times}` to one of the operator style definitions. – **Mark Wibrow** Nov 17 '15 at 15:23

Yes, it works as you said. Thank you, a really good solution. I learned a lot from your example. – **WDiniz** Nov 18 '15 at 15:23

1 @WDiniz maybe you could look at the `execute at begin node` key which can be used to insert stuff at the beginning of a node. Or the `node contents` key (PGF3.0 onwards) which is more sophisticated but can be a bit more fiddly to use. – **Mark Wibrow** Nov 26 '15 at 8:27