

Using the `hexagon` TikZ Library

Jonathan Heathcote

1 Introduction

Drawing diagrams of hexagonal structures for presentations and papers is pretty dull, laborious work. Having recently started playing with TikZ it occurred to me that it would be nice to have some aids for working with hexagons in the library. The easiest way to install this library is to copy it into the working directory of your document¹... Before you get started you'll need to enable the library:

```
\usetikzlibrary{hexagon}
```

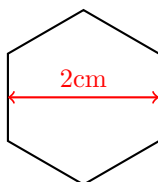
2 Drawing Hexagons

In order to draw a hexagon, simply add the `hexagon` attribute to a node:



```
\begin{tikzpicture}[thick]
  \node [draw, hexagon] {};
\end{tikzpicture}
```

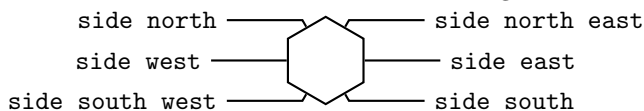
The size of the hexagon can be specified using the `minimum width` attribute:



```
\begin{tikzpicture}[thick]
  \node [draw, hexagon, minimum width=2cm] {};

  \draw [<->,red]
    (-1cm,0) -- node [above] {2cm} ++(2cm,0);
\end{tikzpicture}
```

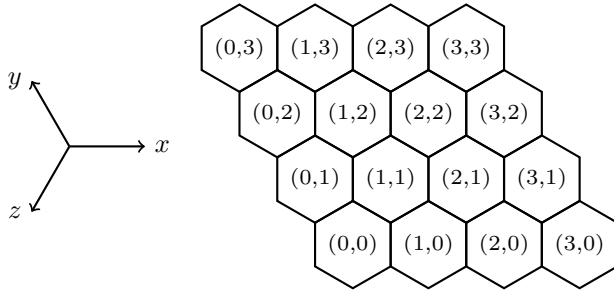
Anchors have been added to each of the edges:



¹Though it should really go in `$HOME/texmf/tex/generic/pgf/frontendlayer/tikz/libraries/tikzlibraryhexagon.code.tex`

3 Positioning Hexagons

To allow easy positioning of hexagons, an attribute `hexagonXYZ` is provided which can be added to an environment to map all (x,y) and (x,y,z) coordinates into the hexagonal space.



```
\begin{tikzpicture}[hexagonXYZ,thick]

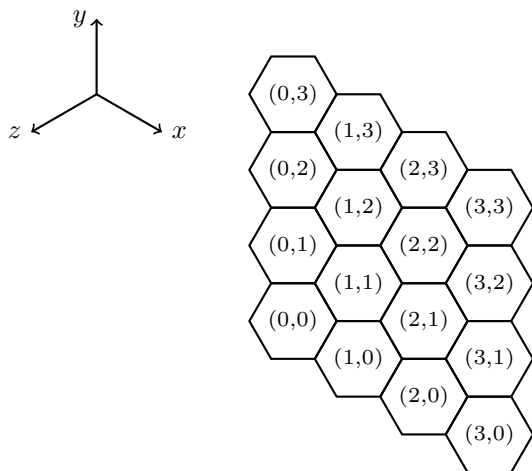
\coordinate (axis) at (-3, 1.5, 0);
\draw [->] (axis) -- ++(1,0,0) node [right] (x) {$x$};
\draw [->] (axis) -- ++(0,1,0) node [left] (y) {$y$};
\draw [->] (axis) -- ++(0,0,1) node [left] (z) {$z$};

\foreach \x in {0,...,3}{
  \foreach \y in {0,...,3}{
    \node [draw,hexagon,minimum width=1cm,font=\scriptsize,inner sep=0]
      at (\x,\y)
      {(\x,\y)};
  }
}

\end{tikzpicture}
```

4 Alternative Rotation

By default hexagons are oriented point-up (as used for the layout of chips in SpiNNaker). An alternative flat-edge-up form, `hexagonBoard`, is provided along with the associated coordinate system, `hexagonBoardXYZ`:



```

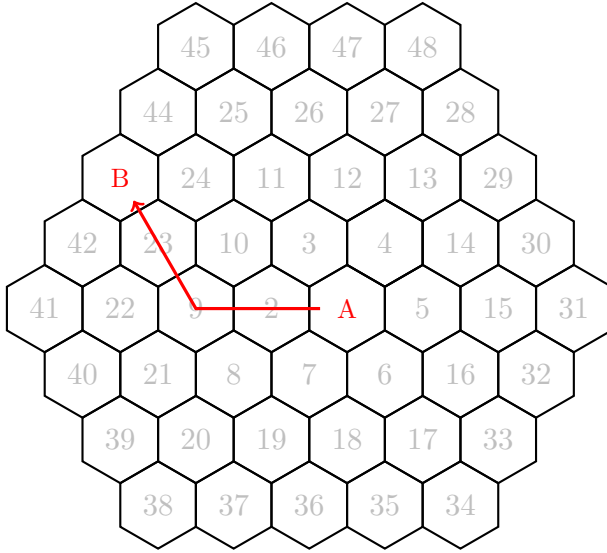
\begin{tikzpicture}[hexagonBoardXYZ,thick]
  \coordinate (axis) at (-3, 1.5, 0);
  \draw [->] (axis) -- ++(1,0,0) node [right] (x) {$x$};
  \draw [->] (axis) -- ++(0,1,0) node [left] (y) {$y$};
  \draw [->] (axis) -- ++(0,0,1) node [left] (z) {$z$};

  \foreach \x in {0,...,3}{
    \foreach \y in {0,...,3}{
      \node [draw,hexagonBoard,minimum width=1cm,font=\scriptsize,inner sep=0]
        at (\x,\y)
          {(\x,\y)};
    }
  }
\end{tikzpicture}

```

5 Drawing SpiNNaker Boards

For the moment I can't figure out how to put the calculation of where the nodes go into TikZ but here's something temporary I generated with Python...



```
\begin{tikzpicture}[hexagonXYZ,thick]
  \foreach \x/\y/\num in {%
                                0/4/45,   1/4/46,   2/4/47,   3/4/48, %
                                -1/3/44,  0/3/25,   1/3/26,   2/3/27,   3/3/28, %
                                -2/2/43,  -1/2/24,  0/2/11,   1/2/12,   2/2/13,   3/2/29, %
                                -3/1/42,  -2/1/23,  -1/1/10,  0/1/3,    1/1/4,    2/1/14,   3/1/30, %
                                -4/0/41,  -3/0/22,  -2/0/9,   -1/0/2,   0/0/1,    1/0/5,    2/0/15,   3/0/31, %
                                -4/-1/40, -3/-1/21, -2/-1/8,   -1/-1/7,  0/-1/6,   1/-1/16,  2/-1/32,   %
                                -4/-2/39, -3/-2/20, -2/-2/19,  -1/-2/18, 0/-2/17,  1/-2/33,   %
                                -4/-3/38, -3/-3/37, -2/-3/36, -1/-3/35, 0/-3/34, %
  }{
    \node [draw,hexagon,minimum size=1cm,text=gray!50!white]
      at (\x,\y) (spinnaker chip \num){\num};
  }

  % Label two chips (covering the number)
  \node at (spinnaker chip 1) [red,fill=white,circle] (A) {A};
  \node at (spinnaker chip 43) [red,fill=white,circle] (B) {B};

  % Draw the shortest path between them
  \draw [->, red, very thick]
    (A) -- (spinnaker chip 9.center) -- (B);
\end{tikzpicture}
```