

Tutorial on the Quantikz Package

Alastair Kay
Royal Holloway University of London, Egham, Surrey, TW20 0EX, UK*
(Dated: September 10, 2018)

I've always used QCircuit for typesetting quantum circuit diagrams within L^AT_EX, but found the Xy-pic based notation rather impenetrable and I struggled to adapt it for my needs (this is probably my failing rather than the package's). Thus, I wanted a tikz package that could do the same. That package is Quantikz. Those familiar with QCircuit will recognise much of the notation, although it has evolved a bit (hopefully simplified!).

I. USAGE

There is not currently a package that lets you install Quantikz automatically. Instead, you need to obtain the file tikzlibraryquantikz.code.tex (it should accompany the source code of this file, and the most recent version will always be available here.) and put it in the same folder as your main tex document. Then, in the preamble of your document, simply write

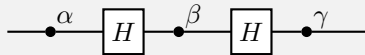
```
\usepackage{tikz}
\usetikzlibrary{quantikz}
```

Now, each time that you want to include a quantum circuit, you just enclose it in a `tikzcd` environment.

Once you start using this for multiple projects, it gets annoying to have to create a copy every time (but remember that you'll need to include a copy for arXiv/journal submission). It might be easier to create a local texmf branch that contains the relevant file in a sub-folder `tex/latex/quantikz/`. If you use MikTeX, see here for Registering a user-managed TEXMF root directory. A range of variants can be found on Stack Exchange.

II. A SINGLE WIRE

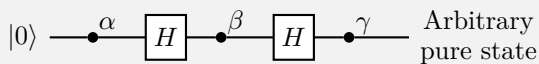
Quantum circuits are laid out with a matrix notation, with cells separated by `&` (just like all matrices, tables etc. in L^AT_EX). Here, we typeset a single quantum wire.



```
\begin{tikzcd}
& \phase{\alpha} & \gate{H} & \phase{\beta} & \gate{H} &
\phase{\gamma} & \qw
\end{tikzcd}
```

Single-qubit gates take the form `\gate{H}`, with any sequence of maths allowed in the argument, or, for phase gates, `\phase{phase}`. The first gate should be in the second column rather than the first (so that the preceding quantum wire has somewhere to go). You *can* put it in the first column, it just won't have an incoming wire.

Text can be added at the start of a quantum wire using `\lstick{}`, and at the end using `\rstick{}`. These are not maths environments.



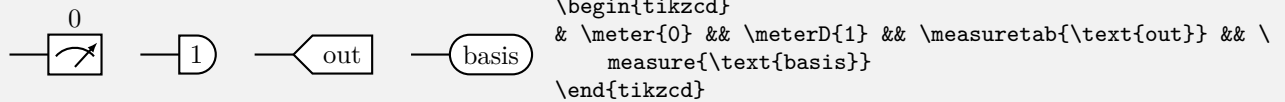
```
\begin{tikzcd}
\lstick{\ket{0}} & \phase{\alpha} & \gate{H} & \phase{\beta} & \gate{H} & \phase{\gamma} & \rstick{Arbitrary\\pure state}\qw
\end{tikzcd}
```

*Electronic address: alastair.kay@rhul.ac.uk

Notice that a multi-line `lstick` or `rstick` is trivial¹.

A. Measurements

There are several measurement devices to choose from, and the measurement basis can be specified as the parameter:

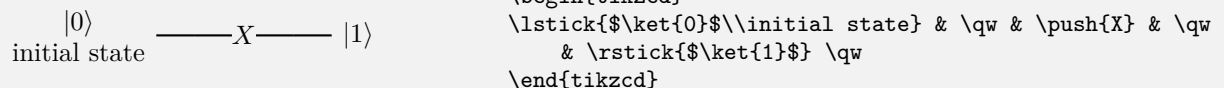


Thus, in our previous circuit:

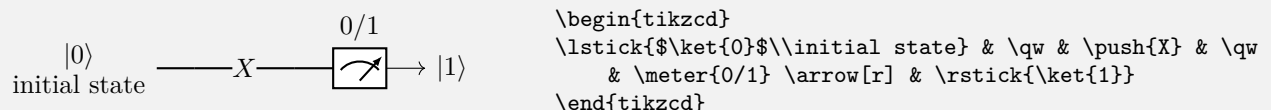


B. Wires and Arrows

There are a few bells and whistles: `\qw` connects a quantum wire from the current cell back to the previous one. Most gates have this built in (`\rstick` is the notable exception). Equally, sometimes you want a ‘do nothing’ operation, and `\qw` is perfect for that. Also, you can just insert text into a cell without boxing it (as you would for a gate). This can be useful for errors, and is achieved with `\push{}`.

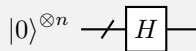


Since we have built quantikz on top of tikzcd, any of the standard arrow commands will work. For example, after a measurement, you might want to use an arrow to report a particular measurement outcome using `\arrow[r]`. The `r` conveys that the arrow should head one cell to the right. You can use combinations of up (u), down (d), left (l) and right as you wish. For more styling options, see the tikzcd manual.



Sometimes, you may want to group a whole bunch of wires together. Use `\qwbundle` instead of `\qw`.

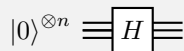
¹ The only issue is that the height of the label will not be taken into account in the vertical spacing of quantum wires. You may have to adjust this manually. See Section VIA for an example of how to achieve this.



```
\begin{tikzcd}
\lstick{$\ket{0}^{\otimes n}$} & \gate{H} \qwbundle&
\qw
\end{tikzcd}
```

The size of the strike can be altered by doing `\pgfkeyssetvalue{/quantikz/Strike Height}{new height}` and `\pgfkeyssetvalue{/quantikz/Strike Width}{new width}`. To adjust the spacing, you'll need to alter the column spacing. This can be done either globally (see later for usage of `row sep` which is similar to the `column sep` you'd need), or on a column by column basis, as done in the previous example with the `[2mm]` statement, adding 2mm of horizontal space between two columns.

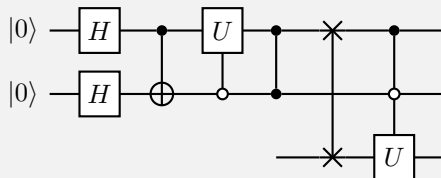
You can access an alternative form for the multi-qubit wire using the `alternate` key



```
\begin{tikzcd}
\lstick{$\ket{0}^{\otimes n}$} & \gate{H} \qwbundle[
alternate]& \control\qwbundle[alternate]
\end{tikzcd}
```

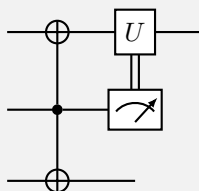
III. MULTIPLE QUBITS

New rows are started with `\\`. If the wires don't interact, then you proceed exactly as before. Naturally, there are several gates that induce interaction between wires: `\ctrl{3}` makes a control-qubit (for controlled-not, controlled-phase, controlled- U , etc), where the target is 3 wires below. `\octrl` is near-identical, except the control circle is open, as usually used to convey being controlled off $|0\rangle$ instead of $|1\rangle$. These will need to be matched with target commands. These could be normal gates or `\targ{}` (controlled-not), `\control{}` (controlled-phase), `\ocontrol{}` (controlled-phase, controlled off $|0\rangle$). There is also a swap gate, `\swap{2}`, with target `\targX{}`. These can be combined to give multi-controlled gates.



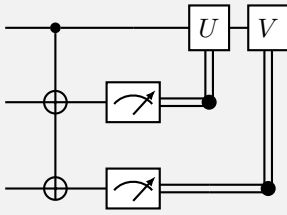
```
\begin{tikzcd}
\lstick{$\ket{0}$} & \gate{H} & \ctrl{1} & \gate{U} & \ctrl{1} & \swap{2} & \ctrl{1} & \qw \\
\lstick{$\ket{0}$} & \gate{H} & \targ{} & \octrl{-1} & \control{} & \qw & \octrl{1} & \qw \\
& & & & \targX{} & \gate{U} & \qw
\end{tikzcd}
```

If you want a gate that is one control and several targets, you need the command `\vqw{1}` (vertical quantum wire) to create vertical connections, or possibly `\vcw{1}` to create the classical version.



```
\begin{tikzcd}
& \targ{} & \gate{U} & \qw \\
& \ctrl{1} & \vqw{-1} & \meter{} \vcw{-1} \\
& \targ{} & \qw
\end{tikzcd}
```

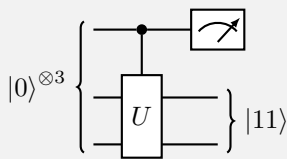
If you want to use classical controls with both horizontal and vertical components, you can do the following.



```
\begin{tikzcd}
& \ctrl{1} & \qw & \gate{U} & \gate{V} & \\
& \targ{} & \vwq{1} & \meter{} & \cwbend{-1} & \\
& \targ{} & \meter{} & \cw & \cwbend{-2} & \\
\end{tikzcd}
```

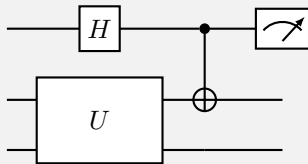
IV. OPERATING ON MANY QUBITS

We have already met the `\lstick`, `\rstick`, and `\gate` commands. By default, these all act on a single quantum wire. However, they also take the optional parameter `wires=n` to specify that they should extend over n wires. The syntax is otherwise unchanged.



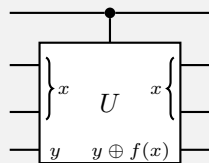
```
\begin{tikzcd}
& \lstick[wires=3]{\ket{0}^{\otimes 3}} & \ctrl{1} & \meter{} & \\
& & \gate[wires=2]{U} & \qw & \rstick[wires=2]{\ket{11}} & \\
& & & & & \\
\end{tikzcd}
```

If you want to add extra width on the `\gate`, you can either alter the text by adding spaces either side, or you could use an `\hphantom` command in one of the underlying cells.



```
\begin{tikzcd}
& \gate{H} & \ctrl{1} & \meter{} & \\
& \gate[wires=2]{U} & \targ{} & \qw & \\
& & \hphantom{wide label} & \qw & \qw & \\
\end{tikzcd}
```

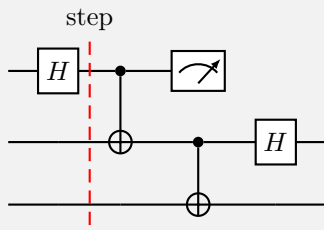
Often, it is nice to be able to label the inputs and outputs on the box, which we do with the `\gateinput` and `\gateoutput` commands. These commands take 1 compulsory argument – the label text. By default, this applies to a single wire, but you can again use the optional `wires` parameter to extend it. The starting point is determined by where the command is placed.



```
\begin{tikzcd}
& \ctrl{1} & \qw & \\
& \gate[wires=3]{U} & & \\
& \gateinput[2]{x} & & \\
& \gateoutput[wires=2]{x} & \qw & \\
& \hphantom{U very long} & \qw & \qw & \\
& \qw & \gateinput{y} & \gateoutput{y \oplus f(x)} & \qw & \\
\end{tikzcd}
```

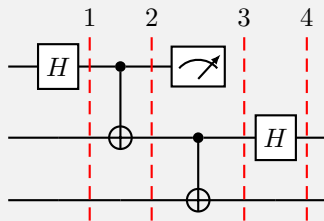
V. SLICING

It is often helpful to ‘slice’ up a circuit for the sake of explaining it step by step. To do this, we provide the `\slice{title}` command, which inserts a dashed vertical line after the column in which the command is added.



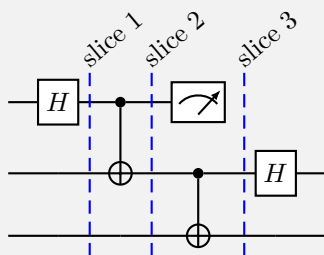
```
\begin{tikzcd}
& \gate{H}\slice{step} & \ctrl{1} & \meter{} \\
& \qw & \targ{} & \ctrl{1} & \gate{H} & \qw \\
& \qw & \qw & \targ{} & \qw & \qw
\end{tikzcd}
```

You can also slice every step by using option `slice all`, and the labels will be automatically numbered. This is likely to behave strangely unless you explicitly ensure that all rows have the same number of entries (i.e. short rows should have extra `&` characters added).



```
\begin{tikzcd}[slice all]
& \gate{H} & \ctrl{1} & \meter{} & \\
& \qw & \targ{} & \ctrl{1} & \gate{H} & \qw \\
& \qw & \qw & \targ{} & \qw & \qw
\end{tikzcd}
```

If you need to adjust where the last slice is, use the optional parameter `remove end slices`, which counts the number of columns fewer to add slices to. You can also change the title of each of the slices, by setting `slice titles`. Include the macro `\col` in your specification if you want to use the step number. Note, however, that the columns won't space themselves out to accommodate a very wide label. You can style the slicing lines with the `slice style` key, and the labels with `slice label style`. These can be used to rotate the labels and create a bit more space!



```
\begin{tikzcd}[slice all,remove end slices=1,slice titles
=slice \col,slice style=blue,slice label style={
inner sep=1pt,anchor=south west,rotate=40}]
& \gate{H} & \ctrl{1} & \meter{} & \\
& \qw & \targ{} & \ctrl{1} & \gate{H} & \qw \\
& \qw & \qw & \targ{} & \qw & \qw
\end{tikzcd}
```

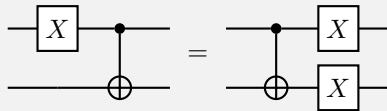
If you get compile errors when trying to slice, check the last line of your matrix, and make sure it doesn't end in `\\`.

VI. TYPESETTING

A. Alignment

How do we centre a circuit diagram? Simply surround it with `\begin{center}` and `\end{center}` commands, or within any standard equation environment.

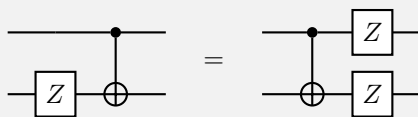
Vertical alignment between different circuits can be more fiddly, depending on how much of a perfectionist you are. Sometimes, they work immediately:



```
\begin{tikzcd}
& \gate{X} & \ctrl{1} & \qw \\
& \qw & \targ{} & \qw \\
\end{tikzcd}
= \begin{tikzcd}
& \ctrl{1} & \gate{X} & \qw \\
& \targ{} & \gate{X} & \qw \\
\end{tikzcd}
```

but the wires don't always align perfectly with each other. Ensuring an even spacing between rows can help (but is not always appropriate). See Sec. VIC.

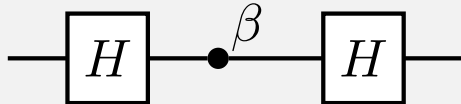
You might be better off writing it as a single quantum circuit with no wires joining the two parts. The spacing works better if your circuit has an odd number of wires, otherwise you have to add an extra row in the middle, and then change the spacing of that row with its neighbours.



```
\begin{tikzcd}
& \qw & \ctrl{2} & \qw & \ctrl{2} & \gate{Z} & \qw \\
& \gate{Z} & \targ{} & \qw & \gate{Z} & \targ{} & \gate{Z} & \qw \\
\end{tikzcd}
```

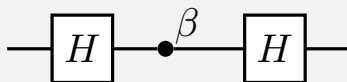
B. Scaling

If you want to override the standard size of a circuit, you can make it a node inside a `tikzpicture`:



```
\begin{tikzpicture}
\node[scale=2] {
\begin{tikzcd}
& \gate{H} & \phase{\beta} & \gate{H} & \qw \\
\end{tikzcd}
};
\end{tikzpicture}
```

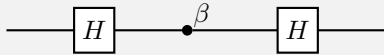
It's also possible to rescale to a fixed width, so long as you declare the `adjustbox` package in the document preamble.



```
\begin{adjustbox}{width=0.8\textwidth}
\begin{tikzcd}
& \gate{H} & \phase{\beta} & \gate{H} & \qw \\
\end{tikzcd}
\end{adjustbox}
```

C. Spacing

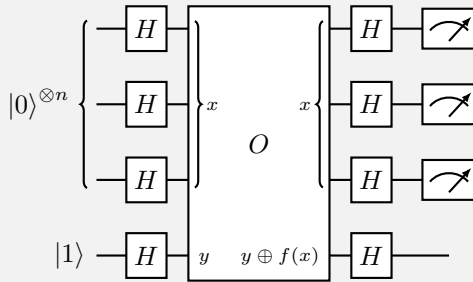
If you wish, it is possible to adjust spacing in the circuits using standard tikz commands. For example, a ridiculous horizontal spacing:



```
\begin{tikzcd}[row sep=0.5cm,column sep=1cm]
& \gate{H} & \phase{\beta} & \gate{H} & \qw
\end{tikzcd}
```

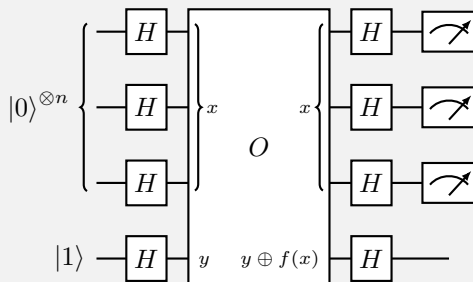
For horizontal spacing of an individual column, use `\hphantom{}` if it's a place where there's already a gate, or there shouldn't be a quantum wire. If there's only a quantum wire, use `\hphantomgate{}`. For spacing of an individual row, see the example on typesetting equation identities in Sec. VIA.

Depending on what gates you have on each wire, the spacing may not be the same between each wire. Sometimes this is desirable, particularly if a gate in one particular row is much larger than anything in the other rows. At other times, it just makes your diagram look a little odd. If you want to make sure that every quantum wire is equally spaced, do the following to `row sep`:



```
\begin{tikzcd}[row sep={1cm,between origins}]
\lstick[wires=3]{\ket{0}^{\otimes n}} & \gate{H} & \gate[wires=4]{0}\gateinput[wires=3]{x} & \gateoutput[wires=3]{x} & \gate{H} & \meter{} \\
& & & & & \\
& & & & & \\
& & & & & \\
\lstick{\ket{1}} & \gate{H} & \phantom{really wide}\gateinput{y}\gateoutput{y\oplus f(x)} & \gate{H} & \qw
\end{tikzcd}
```

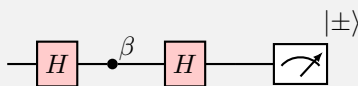
Strictly, the `wires=` statement should not be necessary, as this is presumed to be the default key. Thus, we can get the same results with



```
\begin{tikzcd}[row sep={1cm,between origins}]
\lstick[3]{\ket{0}^{\otimes n}} & \gate{H} & \gate[4]{0}\gateinput[3]{x} & \gateoutput[3]{x} & \gate{H} & \meter{} \\
& & & & & \\
& & & & & \\
& & & & & \\
\lstick{\ket{1}} & \gate{H} & \phantom{really wide}\gateinput{y}\gateoutput{y\oplus f(x)} & \gate{H} & \qw
\end{tikzcd}
```

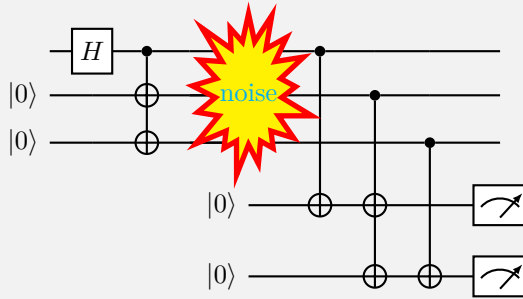
D. Global Styling

Global properties that affect all circuit elements of a given type can be affected through `tikzset`.



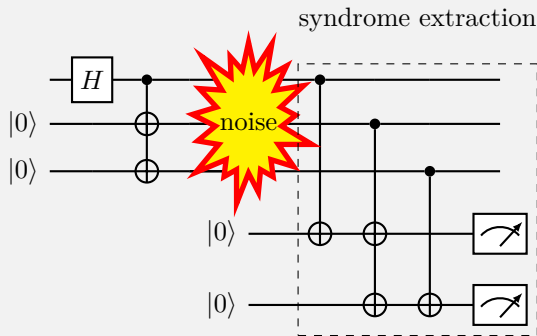
```
\tikzset{
  operator/.append style={fill=red!20},
  my label/.append style={above right,xshift=0.3cm}
}
\begin{tikzcd}
& \gate{H} & \phase{\beta} & \gate{H} & \qw & \meter{\ket{\pm}}
\end{tikzcd}
```

The global styles are:



```
\begin{tikzcd}[row sep=0.3cm,column sep=0.4cm]%
& \gate{H} & \ctrl{2}& \qw & \gate[3,style={starburst,
  fill=yellow,draw=red,line width=2pt,inner xsep=-2pt,
  inner ysep=-5pt},label style=cyan]{\text{noise}} & \ctrl{3} & \qw & \qw & \qw \\
|0\rangle & \oplus & & & & & & & \\
|0\rangle & \oplus & & & & & & & \\
|0\rangle & & \oplus & \oplus & & & & \meter \\
|0\rangle & & \oplus & \oplus & & & & \meter \\
& & & & & & & \end{tikzcd}
```

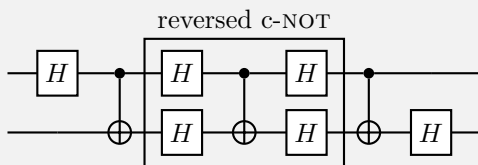
We can also supplement with standard tikz commands by using the ‘execute at end picture’ key. The cells of the matrix contain nodes called `\tikzcdmatrixname-i-j`, where *i* is the row number and *j* is the column number. Thus,



```
\begin{tikzcd}[row sep=0.3cm,column sep=0.4cm,execute at
  end picture={
    \node [fit=(\tikzcdmatrixname-1-6)(\tikzcdmatrixname-4-6)(\tikzcdmatrixname-5-9),
      draw,dashed,inner sep=4pt,label={\text{syndrome extraction}}] {};
  }]
& \gate{H} & \ctrl{2}& \qw & \gate[3,style={starburst,
  fill=yellow,draw=red,line width=2pt,inner xsep=-2pt,
  inner ysep=-5pt}]{\text{noise}} & \ctrl{3} & \qw & \qw & \qw \\
|0\rangle & \oplus & & & & & & & \\
|0\rangle & \oplus & & & & & & & \\
|0\rangle & & \oplus & \oplus & & & & \meter \\
|0\rangle & & \oplus & \oplus & & & & \meter \\
& & & & & & & \end{tikzcd}
```

F. Boxing/Highlighting Parts of a Circuit

In fact, this boxing command is sufficiently useful that we implement it as `\gategroup`. The optional parameters `wires` (again, the default) and `steps` specify the number of rows and columns that the group spans respectively. The mandatory argument is the label for the box (although this can be empty). The top-left corner of the box coincides with the cell in which the command is placed.



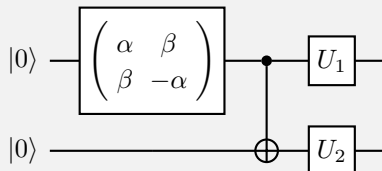
```
\begin{tikzcd}
& \gate{H} & \ctrl{1} & \gate{H}\gategroup[wires=2,steps
  =3,style={inner sep=6pt}]{reversed c-\sc not} & \ctrl{1} & \gate{H} & \ctrl{1} & \qw & \qw \\
& \oplus & & & \oplus & & & \gate{H} & \gate{H} \\
& & & & & & & & \end{tikzcd}
```


QCircuit notation	Quantikz notation
<code>\QCircuit @C=n @R=m {#}</code>	<code>\begin{tikzcd}[row sep=m,col sep=n]#\end{tikzcd}</code>
<code>\multigate{n}</code>	<code>\gate[n+1]</code>
<code>\targ</code>	<code>\targ{}</code>
<code>\control</code>	<code>\control{}</code>
<code>\meter</code>	<code>\meter{}</code>
<code>\measureD</code>	<code>\meterD{}</code>

Updating the `\gategroup` command requires a little more care since the first two arguments have to be removed, and the command placed in the correct cell, at which point `\gategroup{i}{j}{k}{l}{m}` becomes `\gategroup[wires=k+1-i,steps=l+1-j]`. My primary use of `gategroup` was to achieve the effect now achieved with `\lstick[k+1-i]`.

IX. TROUBLESHOOTING

- Have you checked that all commands that need them are followed by an empty argument, `{}`? Things like `\meter`, `\control` (basically, those that can accept an optional styling parameter) look like they don't take any parameters, but they have to be followed by the pair of braces or you'll get very odd effects.
- If you get a whole bunch of unexpected text in one of your cells instead of a gate, make sure that the gate command is the first command in the cell, and that other commands (such as `\qwbundle`) appear after.
- If you're getting errors about cells not being found (and especially if you're doing any slicing), check that your last row doesn't end with `\\`, and make sure that your last row contains as many cells (even if they're empty) as there are columns in the matrix.
- If you are trying to typeset your circuit inside some sort of tabular or align environment and are getting an error "single ampersand used with wrong catcode", trying adding the `[ampersand replacement=\&]` option to `tikzcd`, and instead of separating every cell by `&`, use `\&`. This can also be useful if you want to insert a normal matrix as a gate



```
\begin{tikzcd}[ampersand replacement=\&]
\lstick{\ket{0}} \& \gate{\left(\begin{array}{cc} \alpha & \beta \\ \beta & -\alpha \end{array}\right)} \& \ctrl{1} \& \gate{U_1} \& \qw \\
\lstick{\ket{0}} \& \qw \& \targ{} \& \gate{U_2} \& \qw
\end{tikzcd}
```

Beamer is particularly annoying with doing this. It probably helps to issue a global command

```
\tikzcdset{
  every matrix/.style={ampersand replacement=\&}
}
```

somewhere in your document preamble. The `tikzcd` manual suggests another possible strategy, but I haven't had much success with it.

- If you have a `\qwbundle` in the bottom right cell of the matrix, sometimes you get an error (I have no understanding of why). Add an extra `&` after, and all seems to be well.