Tutorial on the Quantikz Package

Alastair Kay
Royal Holloway University of London, Egham, Surrey, TW20 0EX, UK[*]
(Dated: April 29, 2019)

I've always used QCircuit for typesetting quantum circuit diagrams within LaTeX, 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

The quantikz package is available on CTAN, and will therefore be available through most (current) TeX distributions. Once installed, simply write

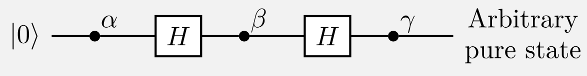
```
\usepackage{tikz}
\usetikzlibrary{quantikz}
```

in the preamble of your document Now, each time that you want to include a quantum circuit, you just enclose it in a `tikzcd` or `quantikz` environment. (Theoretically, there is an advantange of `quantikz` over `tikzcd`, but `tikzcd` is retained for backwards compatibility[1].)

The current version of TeX on the arXiv is not up to date enough to provide the quantikz package. When uploading your source to the arXiv, you need to obtain the file tikzlibraryquantikz.code.tex (you will always be able to locate it on your computer in the main tex directory if you have installed the package, but it should also accompany the source code of this file, and the most recent version will is available here). Just include the file in the main directory of your source code.

II. A SINGLE WIRE

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



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

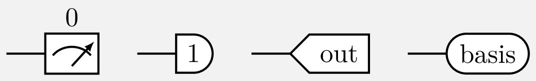
Single-qubit gates take the form `\gate{H}`, with any sequence of maths allowed in the argument[2], or, for phase gates, `\phase{phase}`. The first gate should be in the second column rather than the first (so that the preceding quantum wire has somewhere to go). You *can* put it in the first column, 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, but you can of course insert maths delimiters such as $. Notice that a multi-line `lstick` or `rstick` is trivial[3].

[*]Electronic address: alastair.kay@rhul.ac.uk
[1] `quantikz` should support the external package in tikz. I've never tried it.
[2] See Sec. X in case you need to use & characters in the maths.
[3] 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 VI C for an example of how to achieve this.

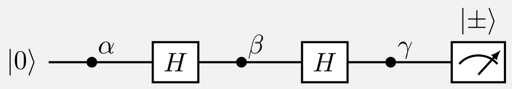
A. Measurements

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



```
\begin{quantikz}
& \meter{0} && \meterD{1} && \measuretab{\text{out}} && \
    measure{\text{basis}}
\end{quantikz}
```

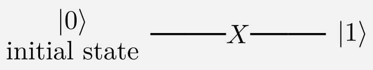
Thus, in our previous circuit:



```
\begin{quantikz}
\lstick{\ket{0}} & \phase{\alpha} & \gate{H} & \phase{\
    beta} & \gate{H} & \phase{\gamma} & \meter{$\ket{\pm
    }$}
\end{quantikz}
```

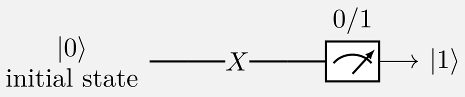
B. Wires and Arrows

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{}`.



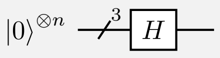
```
\begin{quantikz}
\lstick{$\ket{0}$\\initial state} & \qw & \push{X} & \qw
    & \rstick{$\ket{1}$} \qw
\end{quantikz}
```

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.



```
\begin{quantikz}
\lstick{$\ket{0}$\\initial state} & \qw & \push{X} & \qw
    & \meter{0/1} \arrow[r] & \rstick{$\ket{1}$}
\end{quantikz}
```

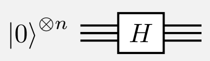
Sometimes, you may want to group a whole bunch of wires together. Use `\qwbundle` instead of `\qw`.



```
\begin{quantikz}
\lstick{$\ket{0}^{\otimes n}$} &[2mm] \gate{H}\qwbundle
    {3} & \qw
\end{quantikz}
```

The size of the strike can be altered by issuing the commands `\pgfkeyssetvalue{/quantikz/Strike Height}{new height}` and `\pgfkeyssetvalue{/quantikz/Strike Width}{new width}`. I felt this example benefited from some additional spacing (of 2mm) on one column. For more spacing options, see Sec. VI.

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

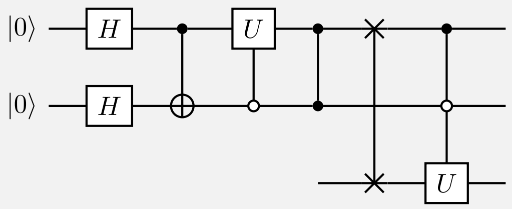


```
\begin{quantikz}
\lstick{$\ket{0}^{\otimes n}$} & \gate{H} \qwbundle[
    alternate]{}& \control\qwbundle[alternate]{}
\end{quantikz}
```

The alternate version does not represent how many wires are supposed to be in the bundle.

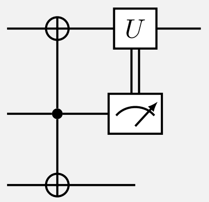
III. MULTIPLE QUBITS

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{}`[4]. These can be combined to give multi-controlled gates.



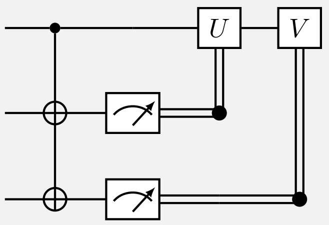
```
\begin{quantikz}
\lstick{$\ket{0}$} & \gate{H} & \ctrl{1} & \gate{U} & \
    ctrl{1} & \swap{2} & \ctrl{1} & \qw \\
\lstick{$\ket{0}$} & \gate{H} & \targ{} & \octrl{-1} & \
    control{} & \qw & \octrl{1} & \qw \\
&&&&&\targX{} & \gate{U} & \qw
\end{quantikz}
```

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



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

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

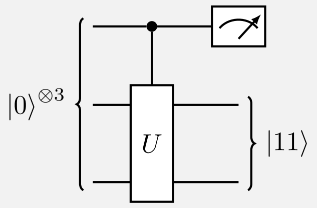


```
\begin{quantikz}
& \ctrl{1} & \qw & \gate{U} & \gate{V} \\
& \targ{} \vqw{1} & \meter{} & \cwbend{-1} \\
& \targ{} & \meter{} & \cw & \cwbend{-2} &
\end{quantikz}
```

[4] For a variant, see Sec. VII C.

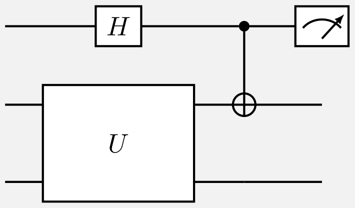
IV. OPERATING ON MANY QUBITS

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[5].



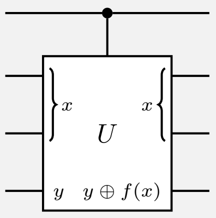
```
\begin{quantikz}
\lstick[wires=3]{$\ket{0}^{\otimes 3}$} & \ctrl{1} & \
    meter{} \\
 & \gate[wires=2]{U} & \qw\rstick[wires=2]{$\ket{11}$} \\
 &  & \qw
\end{quantikz}
```

If you want to add extra width on the `\gate`, use the second optional parameter to specify a width (note that this is a minimum width – if your supplied text is wider, the box will adjust its width accordingly.



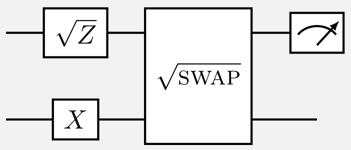
```
\begin{quantikz}
& \gate{H} & \ctrl{1} & \meter{} \\
& \gate[wires=2][2cm]{U} & \targ{} & \qw \\
&  & \qw & \qw
\end{quantikz}
```

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. The width of the gate does not automatically adjust to the contents of these extra labels, so you will have to add it with the second optional parameter of `\gate`.



```
\begin{quantikz}
&\ctrl{1} & \qw \\
&\gate[wires=3][1.7cm]{U}
        \gateinput[2]{$x$}
        \gateoutput[wires=2]{$x$}&\qw \\
& \qw & \qw \\
&\qw\gateinput{$y$}\gateoutput{$y\oplus f(x)$}&\qw
\end{quantikz}
```

You can use standard LaTeXmaths expressions for your labels. Usually, spacing can be automatically adjusted just fine. For example,

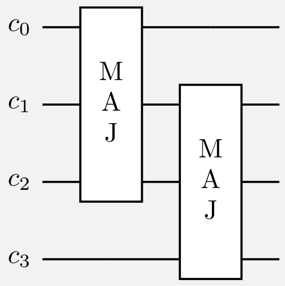


```
\begin{quantikz}
& \gate{\sqrt{Z}} & \gate[2]{\sqrt{\textsc{swap}}} & \
    meter{} \\
& \gate{X} & & \qw
\end{quantikz}
```

Sometimes, it might be that you want a multi-line label, and it should not be that each wire takes the height of those multiple lines. At this point, use the key `disable auto height`. By default, each row will be assigned the height

[5] Strictly, the `wires=` statement should not be necessary, as this is presumed to be the default key, but it is useful for readability.

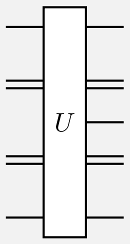
that a gate with label U would be. This can be overridden by the third optional parameter of the gate command, if desired.



```
\begin{quantikz}
\lstick{$c_0$} & \gate[3,disable auto height]{\begin{
    array}{c} \text{M} \\ \text{A} \\ \text{J} \end{
    array}} & \qw & \qw \\
\lstick{$c_1$} & & \gate[3,disable auto height]{\begin{
    array}{c} \text{M} \\ \text{A} \\ \text{J} \end{
    array}} & \qw \\
\lstick{$c_2$} & & & \qw \\
\lstick{$c_3$} & \qw & & \qw
\end{quantikz}
```

A. Different Connections

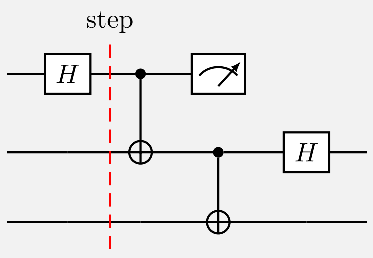
It is assumed that most of the time you want to connect your gates with quantum wires. However, it is possible to override that, and either use classical wires or no wire. These appear as option keys for the gate command, both of which are a comma-separated list of incoming wires that should be altered.



```
\begin{quantikz}
& \gate[5,cwires={2,4},nwires={3}]{U} & \qw \\
& & \cw \\
& & \qw \\
& & \cw \\
& & \qw
\end{quantikz}
```

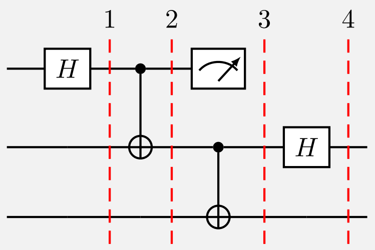
V. SLICING

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.



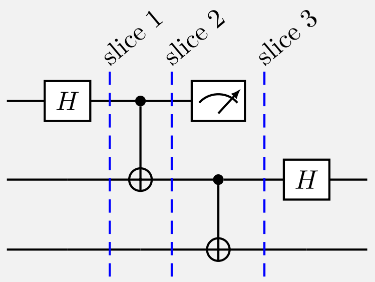
```
\begin{quantikz}
& \gate{H}\slice{step} & \ctrl{1} & \meter{} \\
& \qw      & \targ{} & \ctrl{1} & \gate{H} & \qw \\
& \qw & \qw & \targ{} & \qw & \qw
\end{quantikz}
```

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).



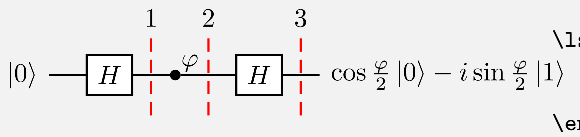
```
\begin{quantikz}[slice all]
& \gate{H} & \ctrl{1} & \meter{} && \\
& \qw      & \targ{} & \ctrl{1} & \gate{H} & \qw \\
& \qw & \qw & \targ{} & \qw & \qw
\end{quantikz}
```

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{quantikz}[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{quantikz}
```

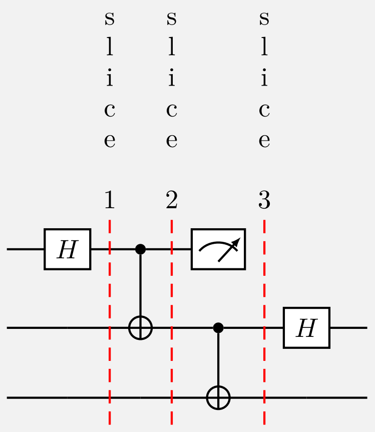
This also gives the possibility to alter spacing if it would otherwise be a bit too tight.


1
2
3
$|0\rangle$
H
φ
H
$\cos\frac{\varphi}{2}|0\rangle - i\sin\frac{\varphi}{2}|1\rangle$

```
\begin{quantikz}[slice all,slice style={shorten <=-0.1cm,
    shorten >=-0.1cm},slice label style={yshift=0.1cm}]
\lstick{$\ket{0}$} & \gate{H} & \phase{\varphi} & \gate{H
    } & \rstick{$\cos\frac{\varphi}{2}\ket{0}-i\sin\frac
    {\varphi}{2}\ket{1}$}\qw
\end{quantikz}
```

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

The `vertical slice labels` key reorients the labels as below.

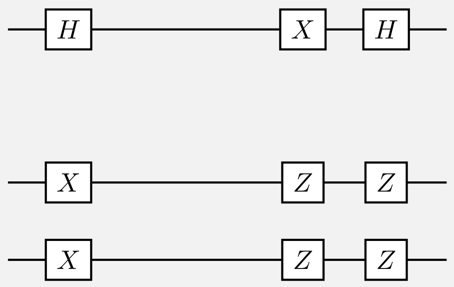


```
\begin{quantikz}[slice all,remove end slices=1,slice
    titles=slice \col,vertical slice labels]
& \gate{H} & \ctrl{1} & \meter{} && \\
& \qw      & \targ{} & \ctrl{1} & \gate{H} & \qw \\
& \qw & \qw & \targ{} & \qw & \qw
\end{quantikz}
```

VI. SPACING

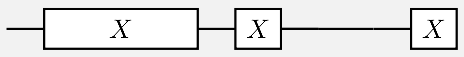
A. Local Adjustment

There are several different ways in which we can manipulate the spacing of a diagram. Adding space to an individual row or column can be done in the standard way of tables in LaTeX. Here we add 2cm of space to the column between the H and X gates, and 1cm of space between the top two rows.



```
\begin{quantikz}
& \gate{H} &[2cm] \gate{X} & \gate{H} & \qw \\[1cm]
& \gate{X} & \gate{Z} & \gate{Z} & \qw \\
& \gate{X} & \gate{Z} & \gate{Z} & \qw
\end{quantikz}
```

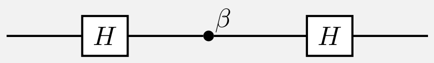
If you don't know how much space you need, but it should be determined by the size of some text, you can use `\hphantom{}` (widens the gate, in a similar way to `\gate[1cm]{}`) or `\hphantomgate{}` (increases the length of a wire) for horizontal spacing, and `\ghost{}` for vertical spacing.



```
\begin{quantikz}
 & \gate{X} \hphantom{very wide} & \gate{X} & \
     hphantomgate{wide}\qw & \gate{X}
\end{quantikz}
```

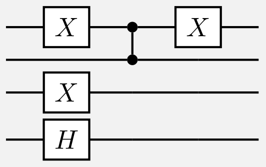
B. Global Adjustment

Standard tikz commands facilitate a global adjustment of row and column spacing. For example, a ridiculous horizontal spacing:



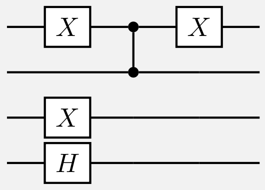
```
\begin{quantikz}[column sep=1cm]
& \gate{H} & \phase{\beta} & \gate{H} & \qw
\end{quantikz}
```

This specifically adjusts the *gap* between the rows and columns, not the distance between the centres of the rows and columns. 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. For example, look at the gap between the top two wires and the bottom two wires:



```
\begin{quantikz}[row sep=0.1cm]
& \gate{X} & \ctrl{1} & \gate{X} & \qw \\
& \qw & \control{} & \qw & \qw \\
& \gate{X} & \qw & \qw & \qw \\
& \gate{H} & \qw & \qw & \qw
\end{quantikz}
```

If you want to make sure that every quantum wire is equally spaced, do the following to `row sep`:



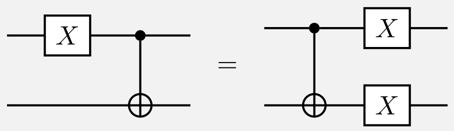
```
\begin{quantikz}[row sep={0.6cm,between origins}]
& \gate{X} & \ctrl{1} & \gate{X} & \qw \\
& \qw & \control{} & \qw & \qw \\
& \gate{X} & \qw & \qw & \qw \\
& \gate{H} & \qw & \qw & \qw
\end{quantikz}
```

This is particularly useful to achieve alignment of several circuits, as in VI C.

C. Alignment

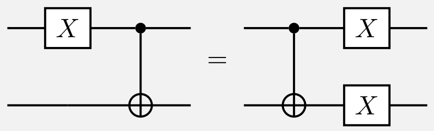
How do we centre a circuit diagram? Simply surround it with `\begin{center}` and `\end{center}` commands, or within any standard equation environment.

Vertical alignment between different circuits can be more fiddly, depending on how much of a perfectionist you are. Sometimes, they work immediately, but the wires don't always align perfectly with each other. Generally the problem is that the highest gate in each row is different (here, the LHS is missing an X gate on the second row)



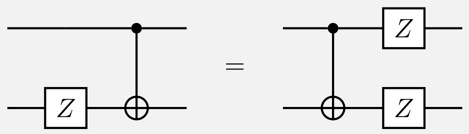
```
\begin{quantikz}
& \gate{X} & \ctrl{1} & \qw \\
& \qw & \targ{} & \qw
\end{quantikz}
=\begin{quantikz}
& \ctrl{1} & \gate{X} & \qw \\
& \targ{} & \gate{X} & \qw
\end{quantikz}
```

Ensuring an even spacing between rows, as described in Sec. VI, can help (but is not always appropriate). Often the easiest is to fudge it using the `\ghost` command which will add a 0-width gate of the height corresponding to its argument. So, having identified the problem with the above circuit, we can replace it with:



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

If you cannot identify the offending gate, and particularly if the operation is not a standard `\gate` command, you might be better off combining the two circuits in a single 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.

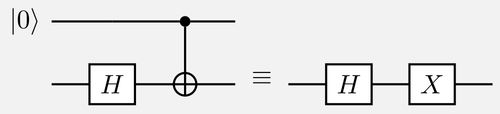


```
\begin{quantikz}
& \qw & \ctrl{2} & \qw&&& \ctrl{2} & \gate{Z} & \qw
    \\[-0.3cm]
&&&&=&\\[-0.3cm]
& \gate{Z} & \targ{} & \qw&&& \targ{} & \gate{Z} & \qw
\end{quantikz}
```

1. Perfecting Vertical Alignment

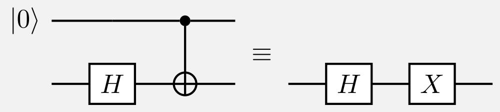
If you want total control over vertical alignment between several different circuits, the trick is to use the `baseline` key. All the baselines of different items are placed on the same level. This is particularly helpful if you give certain wires in the circuit a name, and that will let you place the baseline directly on a level with that wire.

To name a wire as 'name', place an `\alias{name}` command in any cell which is either empty, or contains a `\qw`, `\cw` or `\push` *only*, and place it before that command[6]. That lets you align circuit identities such as



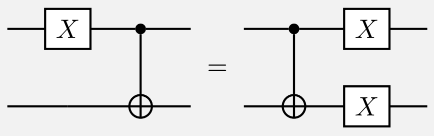
```
$$
\begin{quantikz}[baseline=(W.base)]
\lstick{\ket{0}} & \qw & \ctrl{1} & \qw \\
& \gate{H} & \targ{} & \alias{W} \qw
\end{quantikz}
\equiv\begin{quantikz}[baseline=(W.base)]
& \gate{H} & \gate{X} & \alias{W} \qw
\end{quantikz}
$$
```

This has aligned the bottom wire of the first circuit with the wire in the second circuit, making the correspondence more obvious. However, the ≡ sign is now in the wrong place, with its baseline placed at the same point as the baseline of the circuits. We can fix this with a vertical shift of the two diagrams. We demonstrate two different ways of achieving the same effect (the only important feature is that both diagrams have the same shift).



```
$$
\begin{quantikz}[baseline={($(W.base)+0.3*(0,1cm)$)}]
\lstick{\ket{0}} & \qw & \ctrl{1} & \qw \\
& \gate{H} & \targ{} & \alias{W} \qw
\end{quantikz}
\equiv\begin{quantikz}[baseline={([yshift=0.3cm]new.base)
    }]
& \gate{H} & \gate{X} & \alias{new} \qw
\end{quantikz}
$$
```

If you have an even number if wires, you might choose to calculate the centre of the circuit from `current bounding box.center`, or you could perform a calculation to average the position of two wires. This calculation can then be adjusted to take into account the desire to vertically align on the centre of an equals sign rather than the baseline (which is below the bottom of the equals)



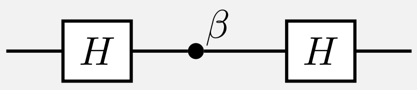
```
$$
\begin{quantikz}[baseline={($(W.base)!.5!(W2.base) -
    height("$\vcenter{}$")*(0,1pt)$)}]
& \gate{X} & \ctrl{1} & \alias{W} \qw \\
& \ghost{X}\qw & \targ{} & \alias{W2} \qw
\end{quantikz}
=\begin{quantikz}[baseline={($(W.base)!.5!(W2.base) -
    height("$\vcenter{}$")*(0,1pt)$)}]
& \ctrl{1} & \gate{X} & \qw \\
& \targ{} & \gate{X} & \qw
\end{quantikz}
$$
```

[6] Alternatively, for any node that accepts tikz styling parameters directly as an option, passing the option `alias=name` should do.

For more details on the possible syntax here, see the `calc` library in tikz.

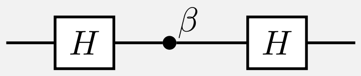
D. Scaling

When we altered the spacing, that left all the gate elements, text etc. the same size, and just altered the spacing between them. If you want to override the standard size of a circuit (gate elements, text and spacing), you can make it a node inside a `tikzpicture`:



```
\begin{tikzpicture}
\node[scale=1.5] {
\begin{quantikz}
& \gate{H} & \phase{\beta} & \gate{H} & \qw
\end{quantikz}
};
\end{tikzpicture}
```

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

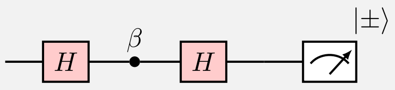


```
\begin{adjustbox}{width=0.8\textwidth}
\begin{quantikz}
& \gate{H} & \phase{\beta} & \gate{H} & \qw
\end{quantikz}
\end{adjustbox}
```

VII. TYPESETTING

A. Global Styling

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

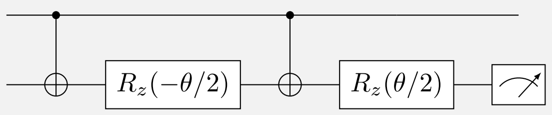


```
\tikzset{
  operator/.append style={fill=red!20},
  my label/.append style={above right,xshift=0.3cm},
  phase label/.append style={label position=above}
}
\begin{quantikz}
& \gate{H} & \phase{\beta} & \gate{H} & \qw & \meter{$\
    ket{\pm}$}
\end{quantikz}
```

The global styles are:

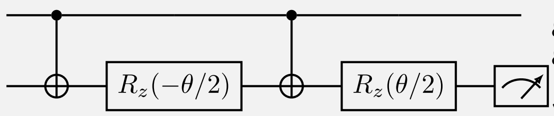
Style Name	Affected Command(s)
operator	`\gate`
meter	`\meter`
slice	`\slice`
wave	`\wave`
leftinternal	`\gateinput`
rightinternal	`\gateoutput`
dm	left braces (`\gateoutput`,`\lstick`)
dd	right braces (`\gateinput`,`\rstick`)
phase	`\phase`, `\control`
ophase	`\ophase`, `\ocontrol`
circlewc	`\targ`
crossx2	`\swap`,`\targX`
my label	measurement bases in `\meter`
phase label	phases in `\phase`
gg label	main gate label in `\gate`
group label	label in `\gategroup`

There are two keys that change styles globally: 'thin lines' to make the lines thin, more in keeping with what QCircuit produced



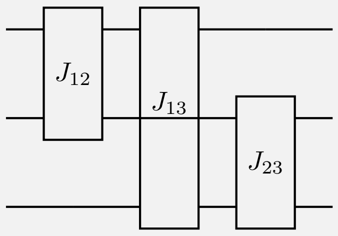
```
\begin{quantikz}[thin lines]
& \ctrl{1} & \qw & \ctrl{1} & \qw & \qw \\
& \targ{} & \gate{R_z(-\theta/2)} & \targ{}
  & \gate{R_z(\theta/2)} & \meter{}
\end{quantikz}
```

and 'transparent', should you want the background of all the gates to be transparent:



```
\begin{quantikz}[transparent]
& \ctrl{1} & \qw & \ctrl{1} & \qw & \qw \\
& \targ{} & \gate{R_z(-\theta/2)} & \targ{}
  & \gate{R_z(\theta/2)} & \meter{}
\end{quantikz}
```

One place where this might be useful is as a "pass-through" on a gate, such as

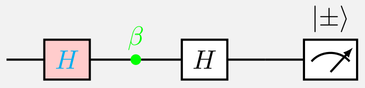


```
\begin{quantikz}[transparent]
& \gate[2]{J_{12}} & \gate[3,label style={yshift=0.2cm}]{
    J_{13}} & \qw & \qw \\
& & \linethrough &\gate[2]{J_{23}} & \qw \\
& \qw &&& \qw
\end{quantikz}
```

While the `\linethrough` would normally be hidden behind the gate, making the gate transparent leaves the line visible. Then one just has to move the J_{13} label (see next section).

B. Per-Gate Styling

Individual gates can be modified using optional arguments of the calling function.



```
\begin{quantikz}
& \gate[style={fill=red!20},label style=cyan]{H}
  & \phase[green,label position=above]{\beta}
  & \gate{H} & \qw & \meter{$\ket{\pm}$}
\end{quantikz}
```

The specific syntax varies a little depending on the type of gate

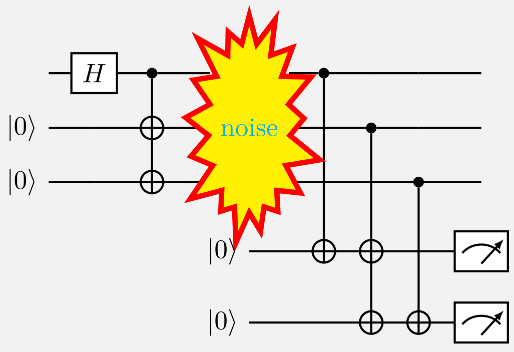
gates that don't accept formatting parameters	\qw,\vqw,\qwbundle,\cw,\vcw,\cwbend,\push
gates that accept tikz node formatting keys directly in optional argument	\phase, \control, \ocontrol, \targ, \targX, \meter, \meterD, \measure, \measuretab, \wave
gates that accept node formatting keys as `label style` key in optional argument	\gate, \slice, \lstick, \rstick, \gategroup, \gateinput, \gateoutput

Those that accept the `label style` key also accept a second key for formatting other elements of the display

\gate,\gategroup	`style`	styles the box
\slice	`style`	styles the line
\lstick,\rstick,\gateinput,\gateoutput	`braces`	styles the brace

If you want to input several styling parameters with one of these keys, just group them together in a set of curly braces, {}. Typical styling parameters include `draw=` specifying line colour, `fill=`, specifying fill colour, `inner xsep=` and `inner ysep=` specifying horizontal and vertical margins respectively, `xshift=` and `yshift=` for adjusting horizontal and vertical positioning. Beyond that, the tikz manual is your friend!

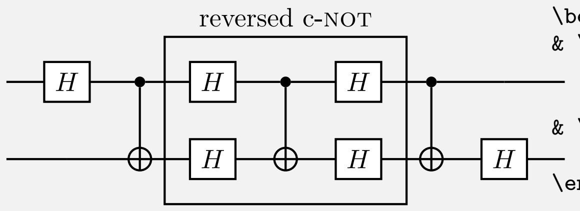
This styling is really quite flexible, as we can override the default shapes with anything that we want:



```
\begin{quantikz}[row sep=0.3cm,column sep=0.3cm]%
& \gate{H} & \ctrl{2}& \qw &
  \gate[3,style={starburst,fill=yellow,draw=red,line
       width=2pt,inner xsep=-4pt,inner ysep=-5pt},
     label style=cyan]{\text{noise}} & \ctrl{3} & \qw & \
        qw& \qw\\
\lstick{$\ket{0}$} & \qw & \targ{} & \qw & \qw & \qw & \
     ctrl{3} & \qw & \qw \\
\lstick{$\ket{0}$} & \qw & \targ{}& \qw & \qw & \qw & \qw
      & \ctrl{2} & \qw\\
&&&&\lstick{$\ket{0}$} & \targ{} & \targ{} & \qw & \meter
     {} \\
&&&&\lstick{$\ket{0}$} & \qw & \targ{} & \targ{} & \meter
     {}
\end{quantikz}
```

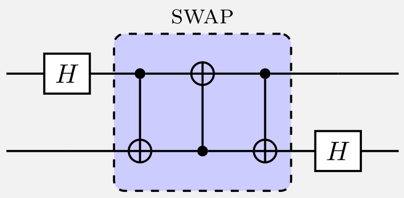
C. Boxing/Highlighting Parts of a Circuit

It is often useful to highlight parts of a circuit. We do this with the `\gategroup` command. 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.



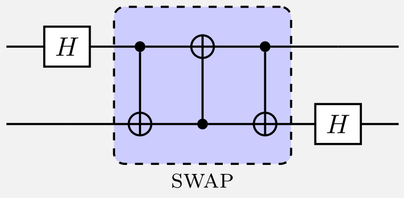
```
\begin{quantikz}
& \gate{H} & \ctrl{1} & \gate{H}\gategroup[wires=2,steps
    =3,style={inner sep=6pt}]{reversed c-{\sc not}} & \
    ctrl{1} & \gate{H} & \ctrl{1} & \qw & \qw \\
& \qw & \targ{} & \gate{H} & \targ{} & \gate{H} & \targ{}
     & \gate{H} & \qw
\end{quantikz}
```

By default, this box is drawn on top of the circuit itself. If you want it to be behind (for example, should you want it to have a background colour), then use the `background` option.



```
\begin{quantikz}
& \gate{H} & \ctrl{1}\gategroup[2,steps=3,style={dashed,
    rounded corners,fill=blue!20, inner xsep=2pt},
    background]{{\sc swap}} & \targ{} & \ctrl{1} & \qw &
     \qw \\
& \qw & \targ{} & \ctrl{-1} & \targ{} & \gate{H} & \qw
\end{quantikz}
```

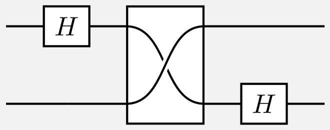
The `label style` key can be used to tune the label's properties, including positioning. For example,



```
\begin{quantikz}
& \gate{H} & \ctrl{1}\gategroup[2,steps=3,style={dashed,
    rounded corners,fill=blue!20, inner xsep=2pt},
    background,label style={label position=below,anchor=
    north,yshift=-0.2cm}]{{\sc swap}} & \targ{} & \ctrl
    {1} & \qw & \qw \\
& \qw & \targ{} & \ctrl{-1} & \targ{} & \gate{H} & \qw
\end{quantikz}
```

Note that it is often good to use `anchor=mid` for label anchors because if you have multiple labels, this will help get them horizontally aligned. It just means you have to use some `yshift` to move the label off the border around the gategroup.

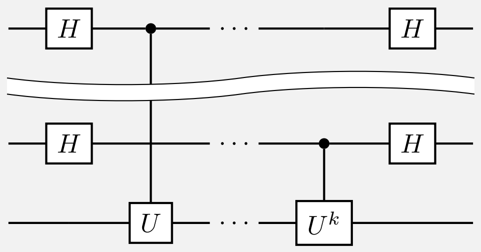
At this point, we mention an extra option for the gate command. The `swap` keyword turns the gate into a variant of a SWAP gate which explicitly shows the wires (the `wires` setting is fixed to 2, overriding any value you might supply).



```
\begin{quantikz}
& \gate{H} & \gate[swap]{} & \qw & \qw \\
& \qw & & \gate{H} & \qw
\end{quantikz}
```

VIII. OTHERWISE UNDOCUMENTED FEATURES

A wave for separating rows. Do we really need this? Perhaps not, but it's fun!



```
\begin{quantikz}
& \gate{H} & \ctrl{3} & \ \ldots\ \qw & \qw & \gate{H} &
    \qw \\
\wave&&&&&&\\
& \gate{H} & \qw & \ \ldots\ \qw & \ctrl{1} & \gate{H} &
    \qw \\
& \qw & \gate{U} & \ \ldots\ \qw & \gate{U^k} & \qw
\end{quantikz}
```

IX. CONVERTING FROM QCIRCUIT

I've updated all of my existing teaching materials from QCircuit to Quantikz with very little trouble. There are a few standard replacements:

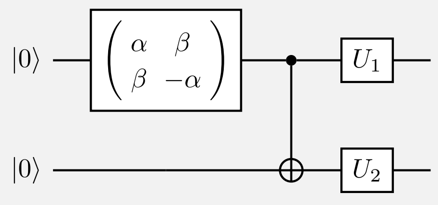
QCircuit notation	Quantikz notation
`\QCircuit @C=n @R=m {#}`	`\begin{quantikz}[row sep=m,col sep=n]#\end{quantikz}`
`\multigate{n}`	`\gate[n+1]`
`\targ`	`\targ{}`
`\control`	`\control{}`
`\meter`	`\meter{}`
`\measureD`	`\meterD{}`

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]`.

It should not be necessary to use `\ghost` commands in the way they were used in QCircuit. The `\cghost` and `\nghost` commands have been replaced by option keys for the gate command, `cwires` and `nwires` respectively.

X. 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 circuit.
- 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


$|0\rangle$
$\begin{pmatrix}\alpha & \beta \\ \beta & -\alpha\end{pmatrix}$
U_1
$|0\rangle$
U_2

```
\begin{quantikz}[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{quantikz}
```

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

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

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 `\cwbend` 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.
- If you're using transparency, and the width of gates seems to be greater than you expected, it may be worthwhile removing the .aux file and recompiling. If your tex editor isn't good at resetting the .aux file, the system may be remembering older widths.
- The code provides definitions for `\bra`, `\ket`, `\proj`, `\braket`. If you don't like them, just define your own versions *before* you load the quantikz library.

For any bug reports (please make sure you've checked the above list first!) or feature requests, please contact alastair.kay@rhul.ac.uk.

XI. CITATION

If you found this package useful, please consider citing the arXiv version of this document, arXiv:1809.03842.