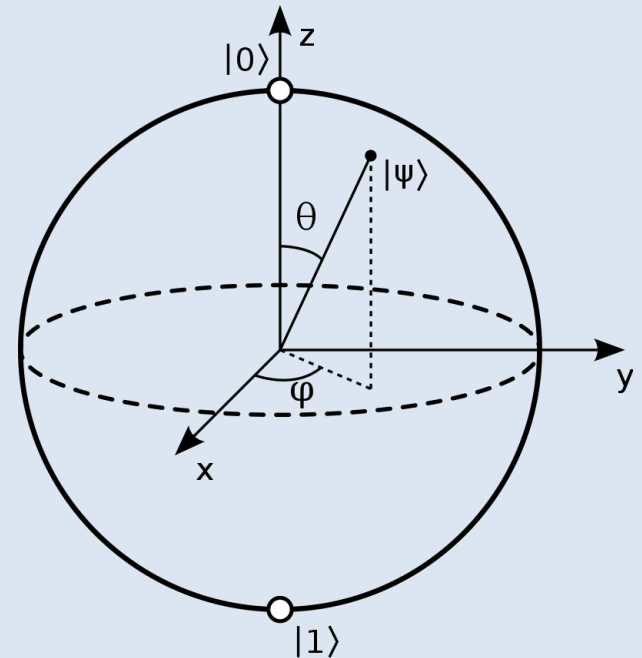


Basics

- Single Qubit: $|\psi\rangle = \cos \frac{\theta}{2} |0\rangle + e^{i\phi} \sin \frac{\theta}{2} |1\rangle$
- Bit and Phase information
 - Measurement:
 - formally: projection onto a basis
 - Change
 - Bit-Flip $X(a|0\rangle + b|1\rangle) = b|0\rangle + a|1\rangle$
 - Phase-Flip $Z(a|0\rangle + b|1\rangle) = a|0\rangle - b|1\rangle$



Bases

- **Computational Basis** = $(|0\rangle \equiv \begin{pmatrix} 1 \\ 0 \end{pmatrix}, |1\rangle \equiv \begin{pmatrix} 0 \\ 1 \end{pmatrix})$

$$Z|0\rangle = |0\rangle, \quad Z|1\rangle = -|1\rangle$$

$$X|0\rangle = |1\rangle, \quad X|1\rangle = |0\rangle$$

- **Hadamard Basis** = $(|+\rangle \equiv \frac{|0\rangle+|1\rangle}{\sqrt{2}}, |-\rangle \equiv \frac{|0\rangle-|1\rangle}{\sqrt{2}})$

$$X|+\rangle = |+\rangle, \quad X|-\rangle = -|-\rangle$$

$$Z|+\rangle = |-\rangle, \quad Z|-\rangle = |+\rangle$$

- **Conversion**

$$H|0\rangle = |+\rangle, \quad H|1\rangle = |-\rangle, \quad H^2 = \mathbb{1}$$

Basics

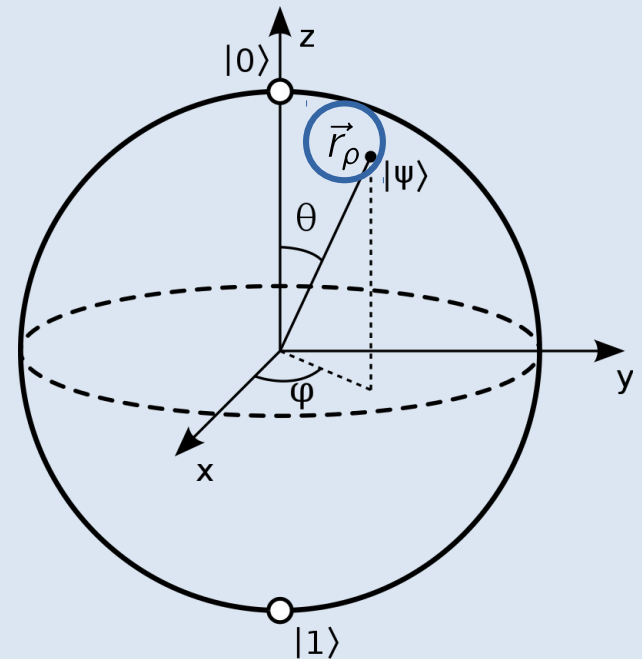
- Bloch Sphere:

$$\begin{aligned}\rho &= |\psi\rangle \langle\psi| \\ &= \frac{1}{2}(\mathbb{1} + X \cos \phi \sin \theta + Y \sin \phi \sin \theta + Z \cos \theta) \\ &= \frac{1}{2}(\mathbb{1} + \vec{r}_\rho \cdot \vec{\sigma})\end{aligned}$$

- Rotations:

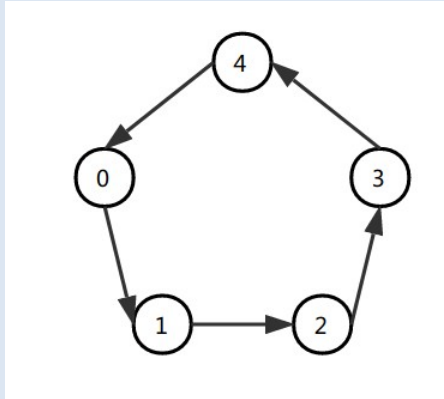
$$R_{\hat{n}}(\alpha) = \exp\left(-i\frac{\alpha}{2}\hat{n} \cdot \vec{\sigma}\right)$$

$$R_{\hat{x}}(\pi) = X, \quad R_{\hat{z}}(\pi) = Z \quad (R_{\hat{y}}(\pi) = Y)$$

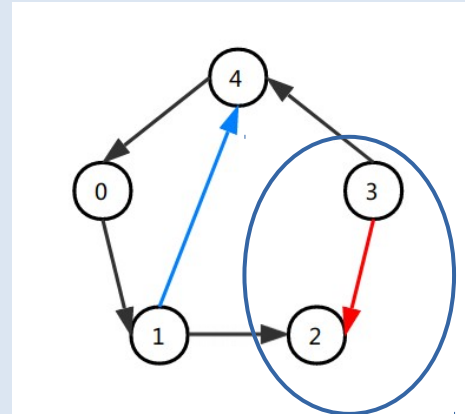


Running 2-Qubit Gates & Connectivity

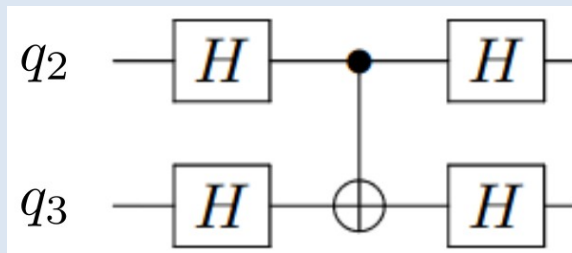
Suppose a device has this layout:



But we want to do this:

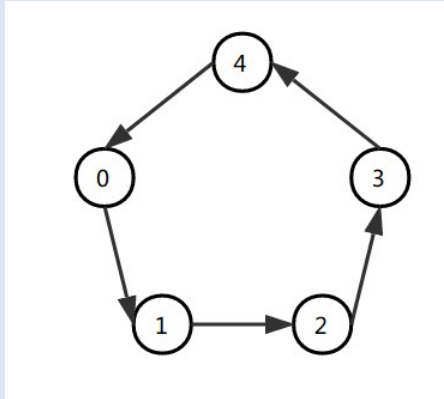


Wrong **direction**! Circumvent it with

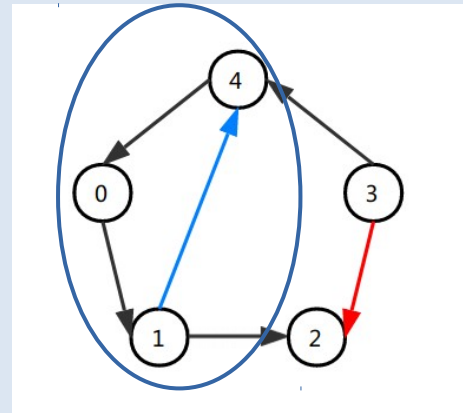


Running 2-Qubit Gates & Connectivity

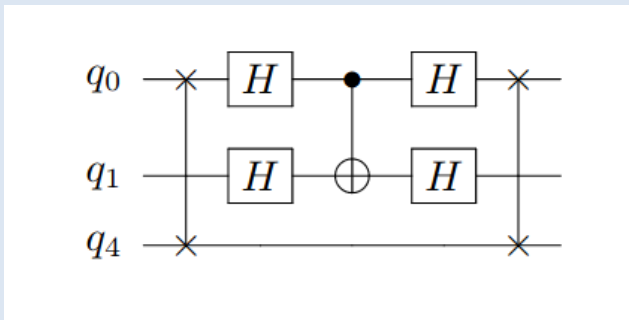
Suppose a device has this layout:



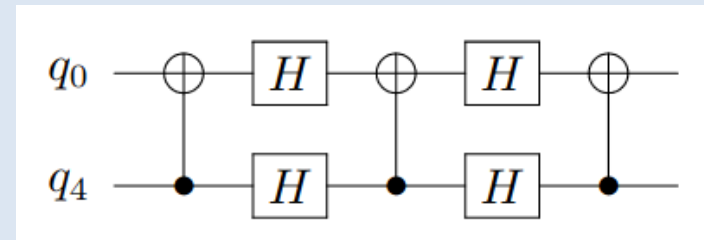
But we want to do this:



No direct connection available! Circumvent it with e.g.



SWAP, CNOT, SWAP



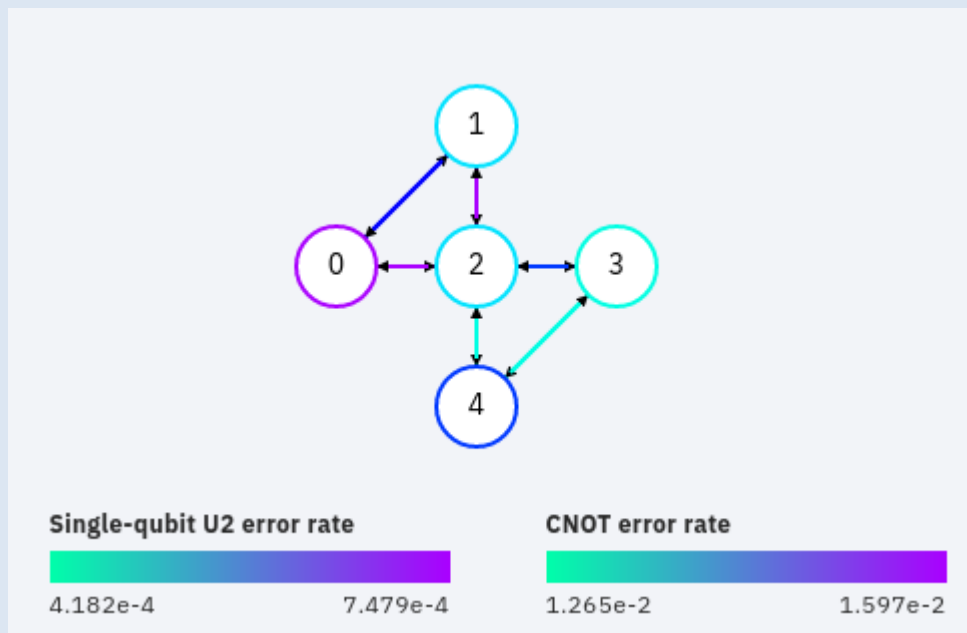
SWAP(q0, q4)

Running 2-Qubit Gates & Connectivity

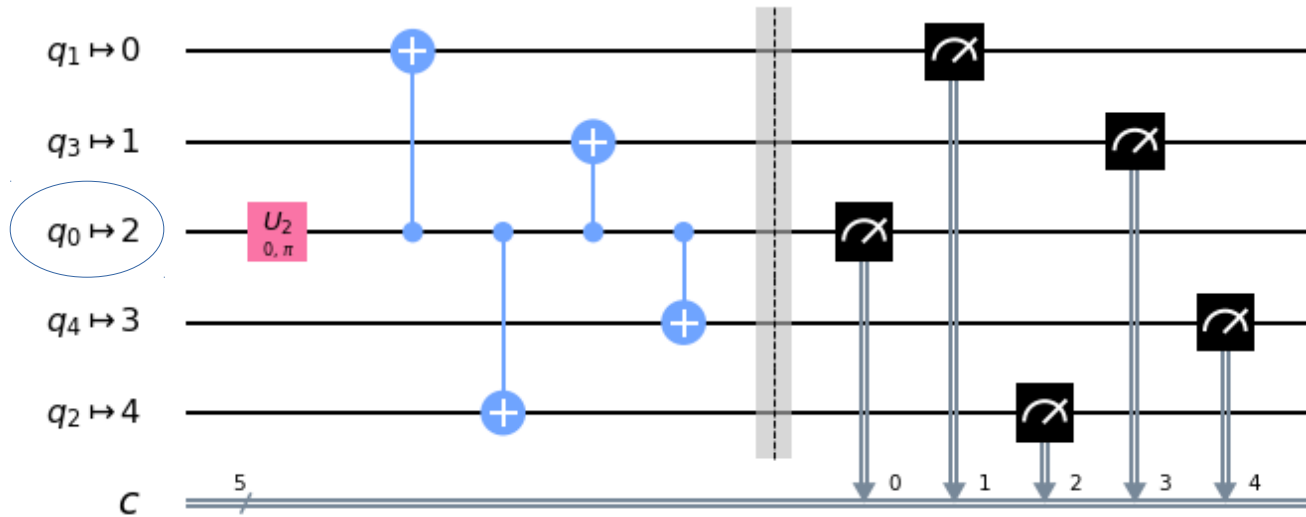
→ Minimize the cost for given quantum algorithm on a given hardware layout

- Software: Compilation/Transpilation
- Hardware: optimized layout for specific algorithms

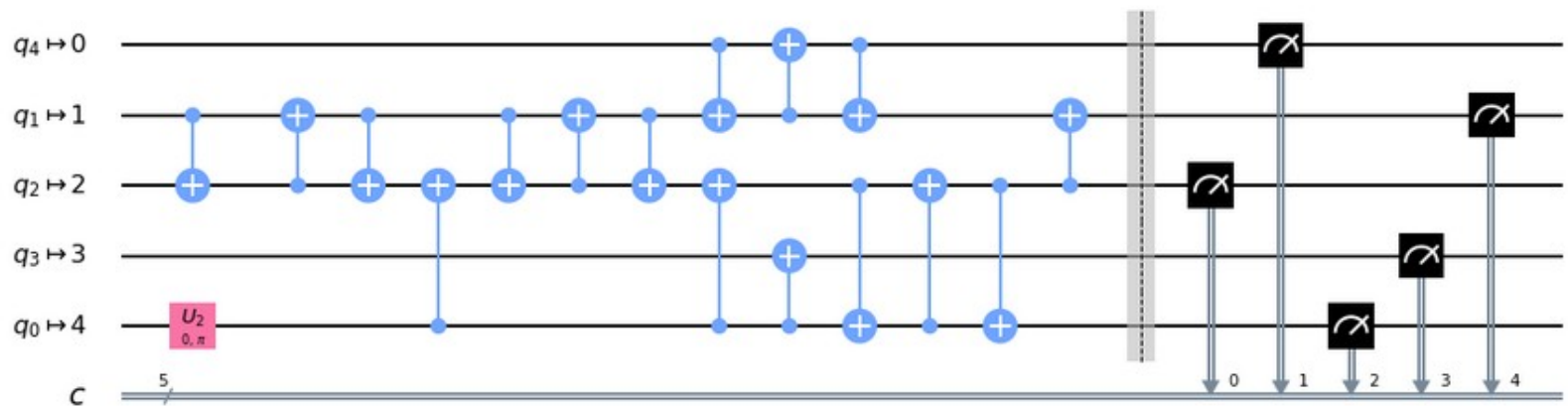
Example:
ibmqx2,
(Yorktown)

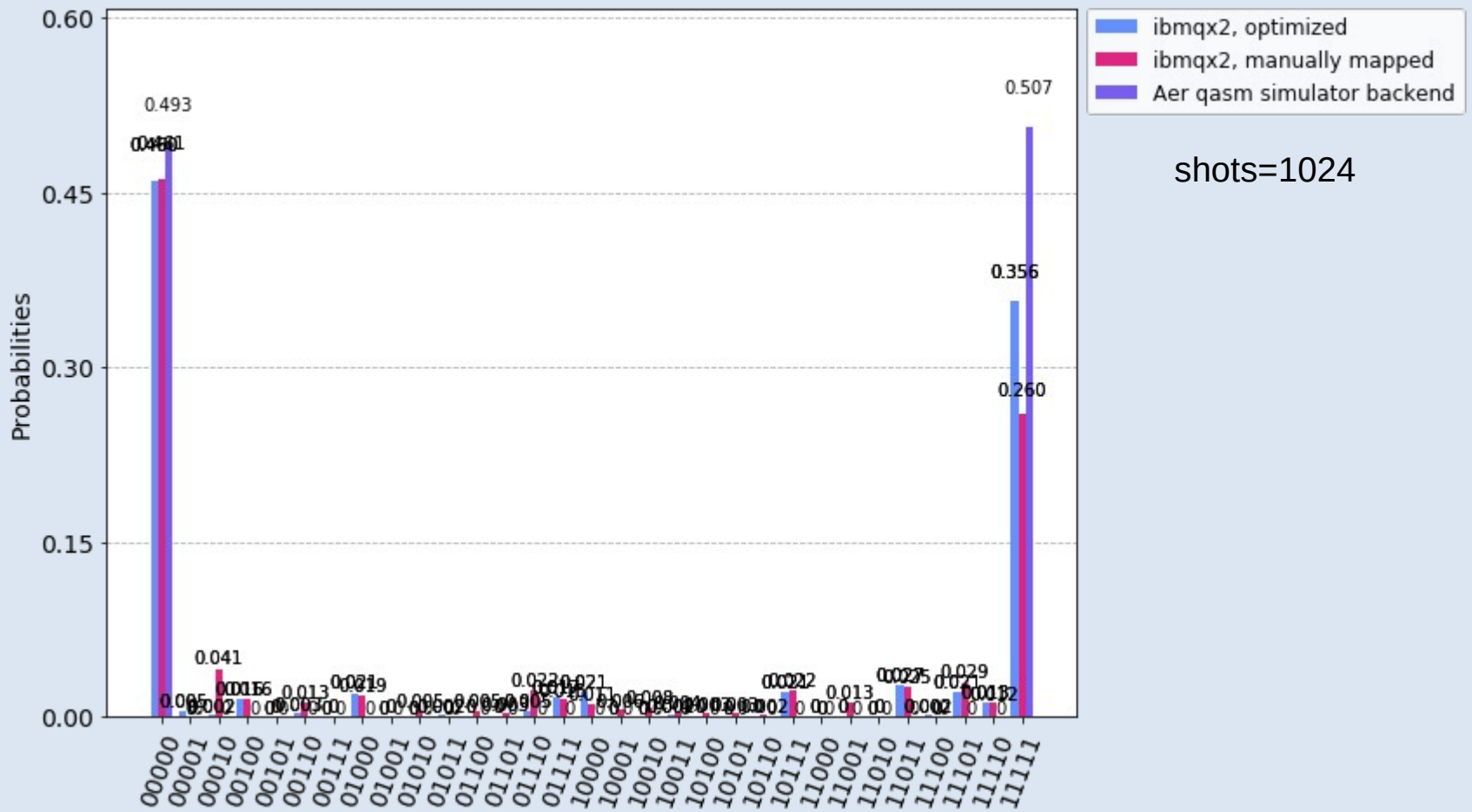


```
auto_mapped_ghz_circ = compiler.transpile(ghz, provider.get_backend('ibmqx2'), optimization_level=3)
auto_mapped_ghz_circ.draw(output='mpl')
```



```
custom_mapped_ghz_circ = compiler.transpile(ghz, provider.get_backend('ibmqx2'),
                                             initial_layout=[4,1,2,3,0], optimization_level=0)
custom_mapped_ghz_circ.draw(output='mpl')
```





→ here: optimized circuit performs better