Quantum gates and notation

/tbabej
@tomasbabej

Tomas Babej & Mark Fingerhuth
ProteinQure Inc.

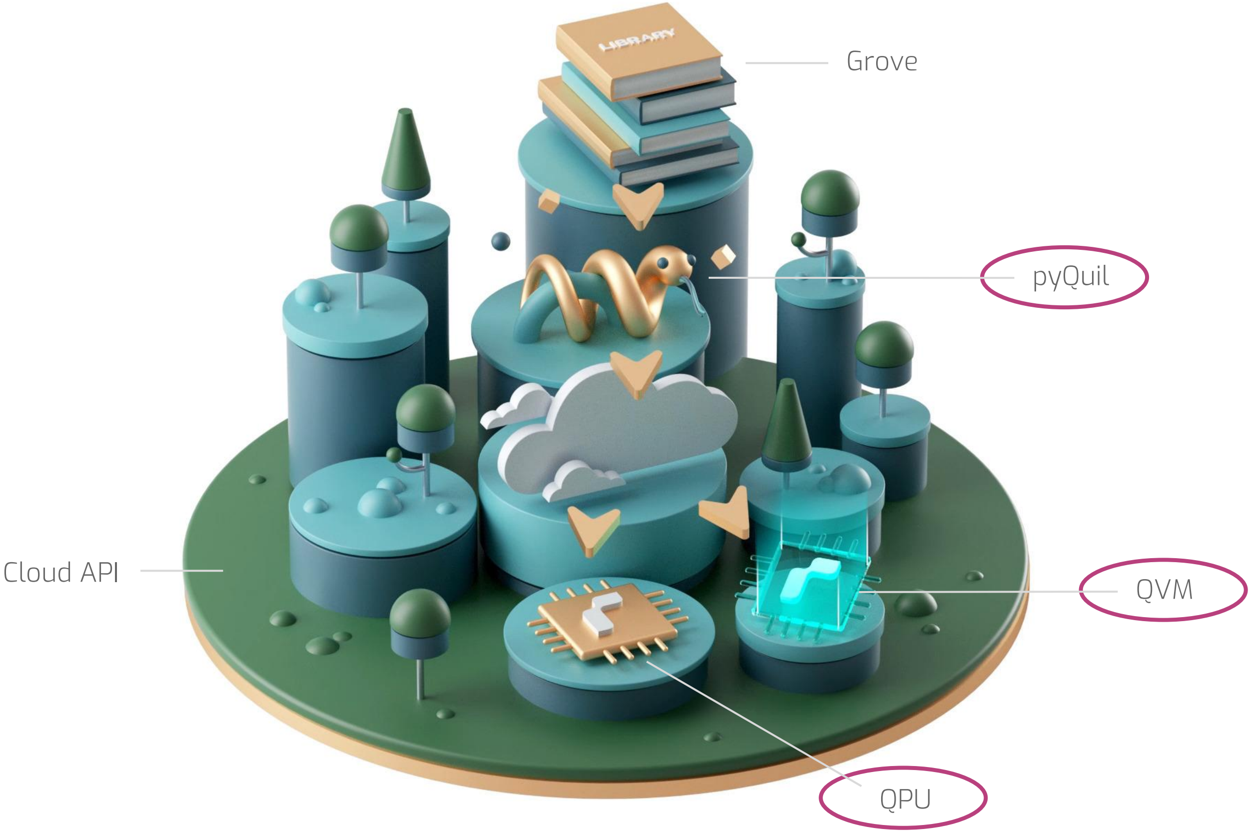
/m-fingerhuth
@mark_fingerhuth

Creative Destruction Lab
Toronto, CA
July 12, 2018

ProteinQure

pyQuil & QUIL: Low-level quantum programming

Bloch sphere | Quantum gates | Quantum circuits | Notation



A two-dimensional quantum bit revisited

Bloch sphere | Quantum gates | Quantum circuits | Notation

We can write a two-dimensional **qubit** as the vector:

Global phase

$$|\psi\rangle = e^{i\gamma}\left(\alpha\,|0\rangle + \beta\,|1\rangle\right) = e^{i\gamma}\begin{bmatrix}\alpha \\ \beta\end{bmatrix}$$

Amplitudes

Ground state

Excited state

A two-dimensional quantum bit revisited

Bloch sphere | Quantum gates | Quantum circuits | Notation

We can **always** drop the **global phase** since it's immeasurable:

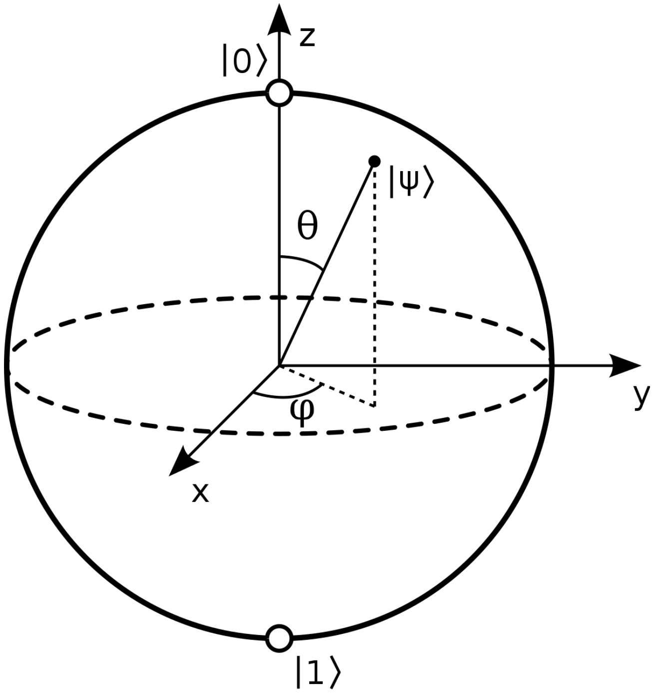
~~Global phase~~

$$|\psi\rangle = \cancel{e^{i\gamma}}\left(\alpha\,|0\rangle + \beta\,|1\rangle \right) = \cancel{e^{i\gamma}} \begin{bmatrix} \alpha \\ \beta \end{bmatrix}$$

since

$$|e^{i\gamma}\alpha|^2 = e^{i\gamma}\alpha e^{-i\gamma}\alpha^\dagger = e^{i\gamma - i\gamma}\alpha\alpha^\dagger = \alpha\alpha^\dagger$$

Bloch sphere



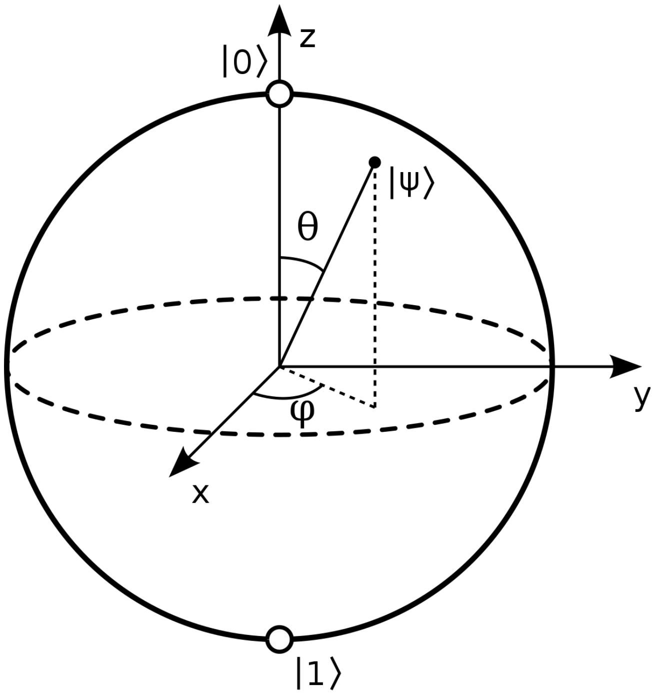
$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$$

$$|\psi\rangle = \cos\left(\frac{\theta}{2}\right)|0\rangle + e^{i\phi}\sin\left(\frac{\theta}{2}\right)|1\rangle$$

where $0 \leq \theta \leq \pi$ and $0 \leq \phi < 2\pi$

$$x = \sin(\theta)\cos(\phi)$$
$$y = \sin(\theta)\sin(\phi)$$
$$z = \cos(\theta)$$

Bloch sphere



Quantum gates rotate the Bloch vector

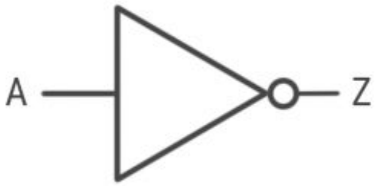
The Bloch sphere provides a **visual tool** to understand quantum gate operations.

The Bloch sphere **only applies to single qubit states**. Visualizing multi-qubit states is out of the scope of this tutorial.

Single-qubit quantum gates

Classical NOT Gate

NOT



Input	Output
0	1
1	0

Single-qubit quantum gates

Bloch sphere | **Quantum gates** | Quantum circuits | Notation

Quantum X (NOT) Gate

Input	Output
0	1
1	0

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}$$

$$? \times \underset{|0\rangle}{\begin{bmatrix} 1 \\ 0 \end{bmatrix}} \rightarrow \underset{|1\rangle}{\begin{bmatrix} 0 \\ 1 \end{bmatrix}}$$

Single-qubit quantum gates

Quantum X (NOT) Gate

Input	Output
0	1
1	0

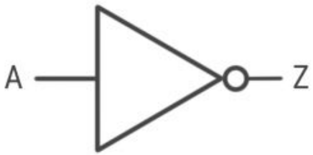
$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}$$

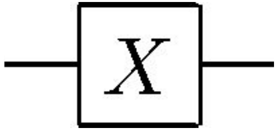
$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \times \underbrace{\begin{bmatrix} 1 \\ 0 \end{bmatrix}}_{|0\rangle} \rightarrow \underbrace{\begin{bmatrix} 0 \\ 1 \end{bmatrix}}_{|1\rangle}$$

Single-qubit quantum gates

X (NOT) Gate

NOT → X





$$\mathbf{X}\,|0\rangle = |1\rangle$$

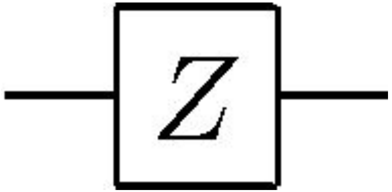
$$\mathbf{X}\,|1\rangle = |0\rangle$$

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \times \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \beta \\ \alpha \end{bmatrix}$$

swaps amplitudes!

Single-qubit quantum gates

Pauli Z



$$Z\left|0\right\rangle = \left|0\right\rangle$$

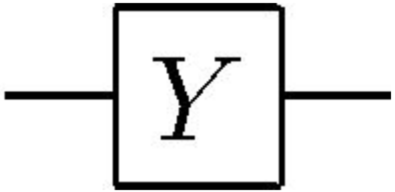
$$Z\left|1\right\rangle = -\left|1\right\rangle$$

$$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \times \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \alpha \\ -\beta \end{bmatrix}$$

adds phase!

Single-qubit quantum gates

Pauli Y



$$Y\left|0\right\rangle = i\left|1\right\rangle$$

$$Y\left|1\right\rangle = -i\left|0\right\rangle$$

$$\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \times \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} -i\beta \\ i\alpha \end{bmatrix}$$

swaps amplitudes & adds phase!

Properties of single-qubit gates

$$|\langle \Psi \mid \Psi \rangle|^2 = 1$$

The **Bohr rule** states that the inner product **must equal to unity.**

Properties of single-qubit gates

$$|\Psi'\rangle = U\,|\Psi\rangle$$

$$|\,\langle\Psi' \mid \Psi'\rangle\,|^2 = 1$$

Quantum gates **must preserve normalization.**

Properties of single-qubit gates

$$|\Psi'\rangle = U\,|\Psi\rangle \qquad\qquad |\,\langle\Psi' \mid \Psi'\rangle\,|^2 = 1$$

$$|\,\langle\Psi|\,U^\dagger U\,|\Psi\rangle\,|^2 = 1$$

$$U^\dagger U = \mathbb{I}$$

The only requirement for a quantum gate is **unitarity.**

Hence, there are infinitely many quantum gates!

Properties of single-qubit gates

Quantum gates must be **unitary:**

$$UU^\dagger = U^\dagger U = \mathbb{I}$$

Example:

$$S = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix} \qquad S^\dagger = \begin{bmatrix} 1 & 0 \\ 0 & -i \end{bmatrix}$$

Hermitian conjugate

$$SS^\dagger = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & -i \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & -i^2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \mathbb{I}$$

Properties of single-qubit gates

Some quantum gates (e.g. Pauli gates) are **hermitian:**

$$U = U^\dagger$$

Example:

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = X^\dagger = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Elementary single-qubit gates

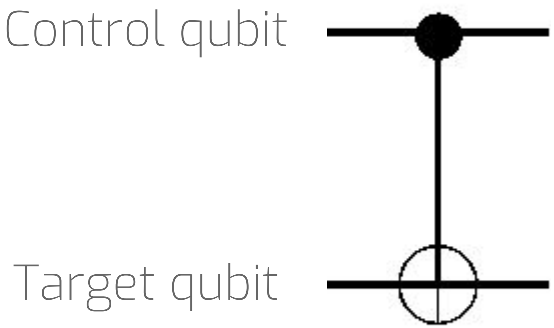
Hadamard	$-\boxed{H}-$	$\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\1 & -1\end{bmatrix}$	`>>> from pyquil.gates import H`
Pauli-X	$-\boxed{X}-$	$\begin{bmatrix}0 & 1\\1 & 0\end{bmatrix}$	`>>> from pyquil.gates import X`
Pauli-Y	$-\boxed{Y}-$	$\begin{bmatrix}0 & -i\\i & 0\end{bmatrix}$	`>>> from pyquil.gates import Y`
Pauli-Z	$-\boxed{Z}-$	$\begin{bmatrix}1 & 0\\0 & -1\end{bmatrix}$	`>>> from pyquil.gates import Z`
Phase	$-\boxed{S}-$	$\begin{bmatrix}1 & 0\\0 & i\end{bmatrix}$	`>>> from pyquil.gates import S`
$\pi/8$	$-\boxed{T}-$	$\begin{bmatrix}1 & 0\\0 & e^{i\pi/4}\end{bmatrix}$	`>>> from pyquil.gates import T`

\+ many more!

A lot of quantum gates are **already implemented** in pyQuil.

Multi-qubit quantum gates

Controlled-NOT (CNOT) gate



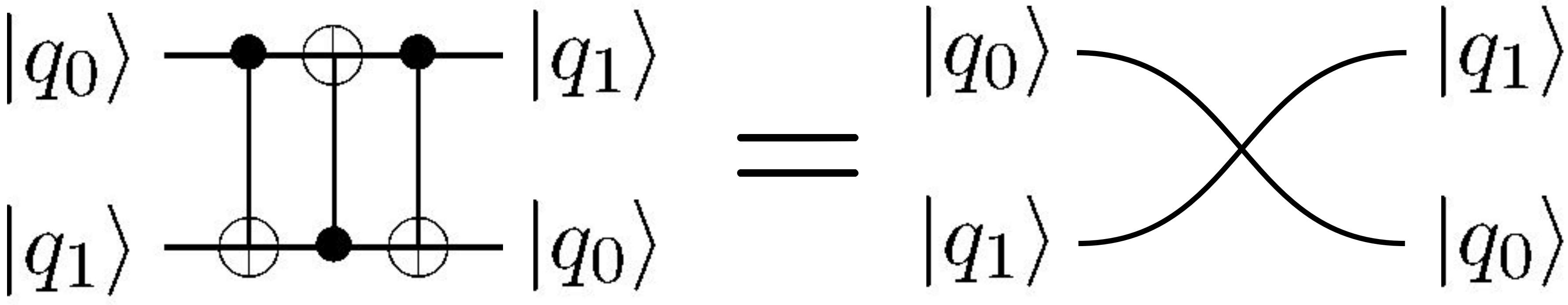

```
>>> from pyquil.gates import CNOT
```

Target qubit	Control qubit	Output
0	0	00
0	**1**	**11**
1	0	10
1	**1**	**01**

Essential for **entangling** two qubits!

Multi-qubit quantum gates

SWAP gate

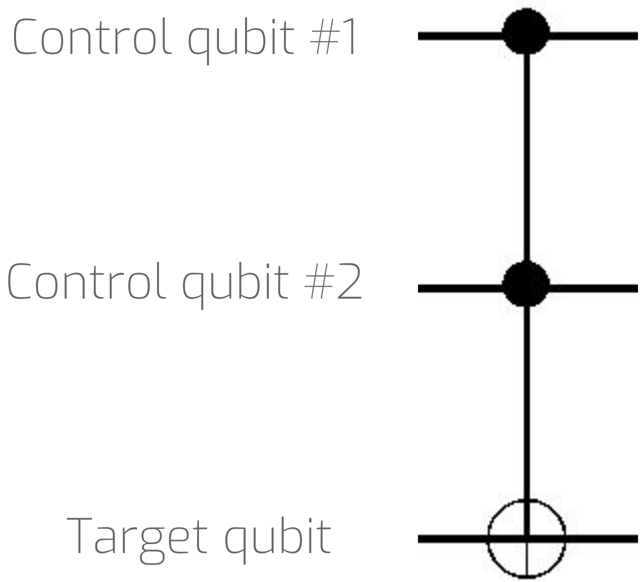



```
>>> from pyquil.gates import SWAP
```

The SWAP operation allows us to **move qubits on the quantum processor**!

Multi-qubit quantum gates

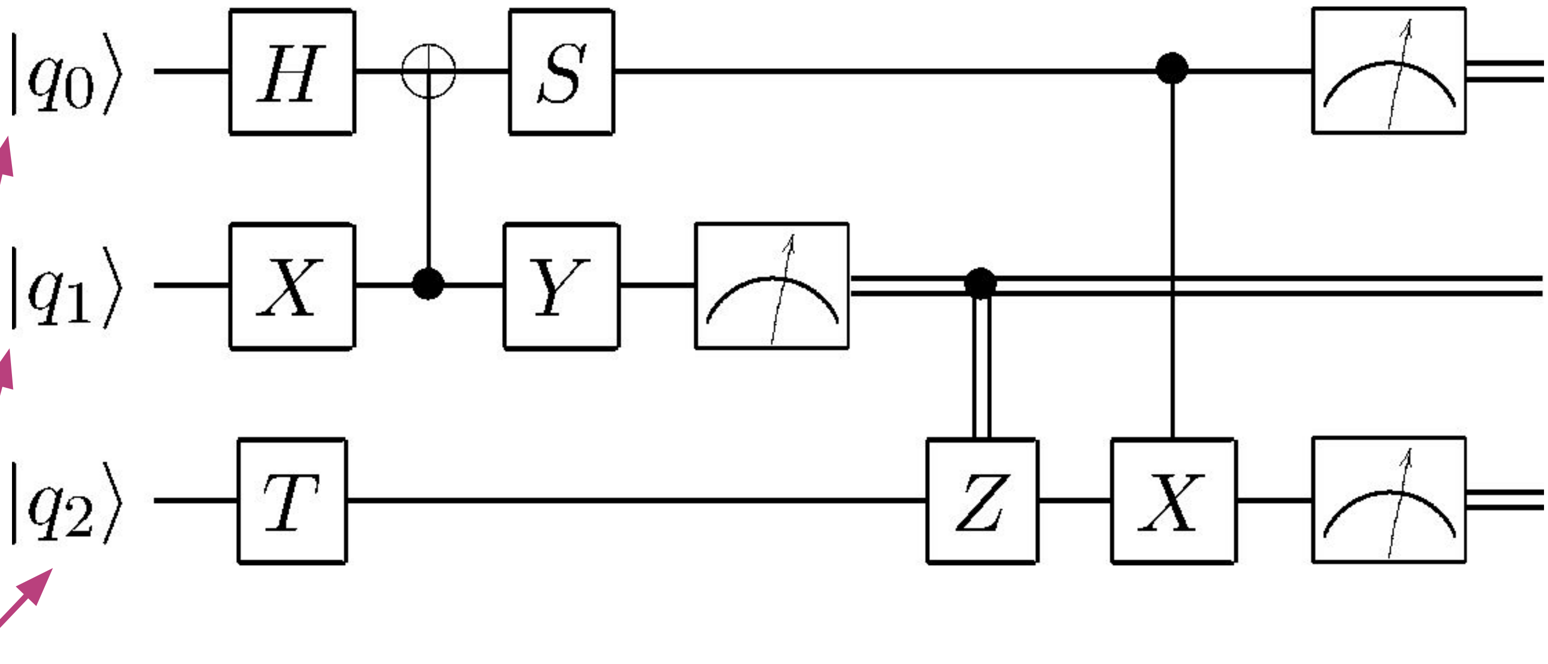
Toffoli (CCNOT) gate



Target qubit	Control qubit #1	Control qubit #2	Output
0	1	0	010
0	0	1	001
0	**1**	**1**	**111**

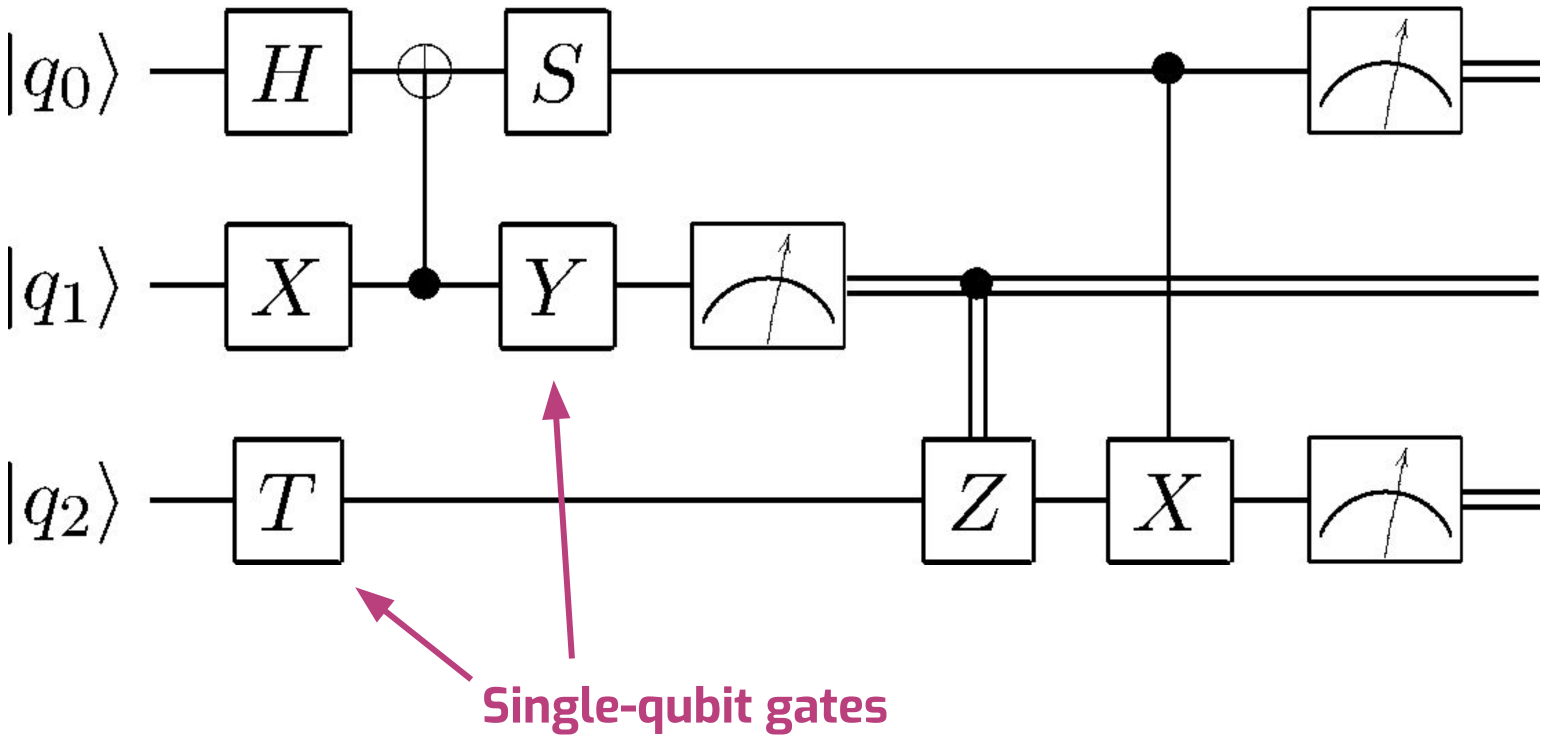
```
>>> from pyquil.gates import CCNOT
```

Quantum circuits

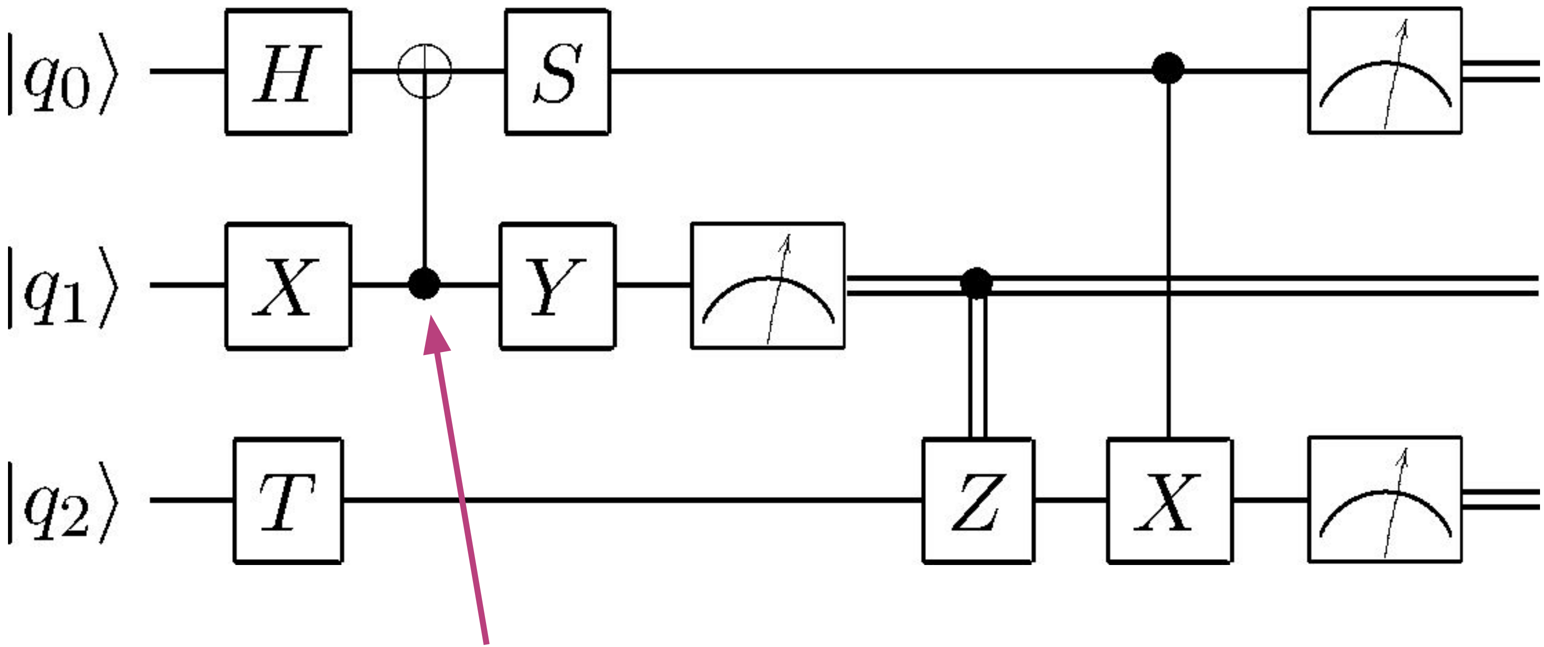


Initial qubit state ($|000\rangle$)

Quantum circuits



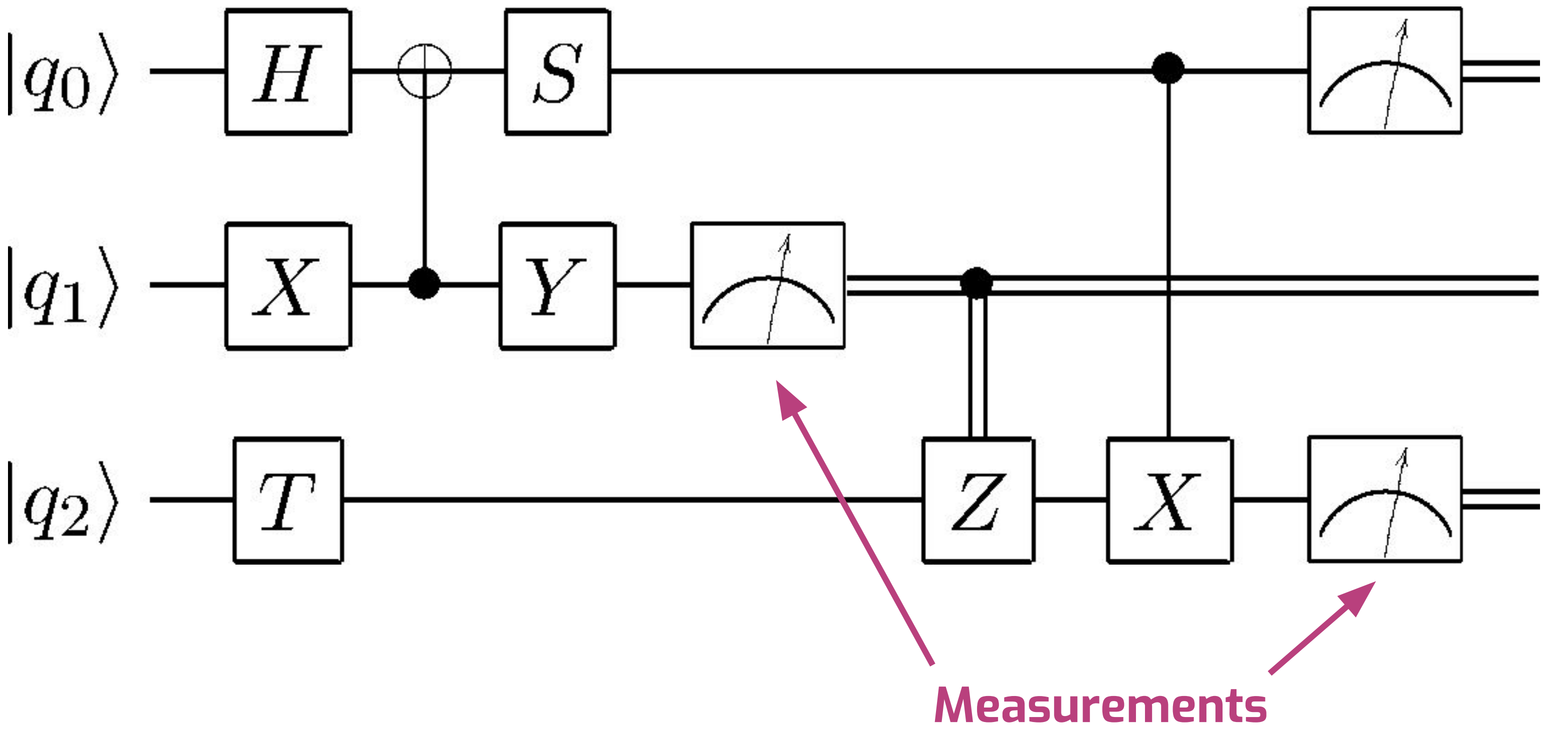
Quantum circuits



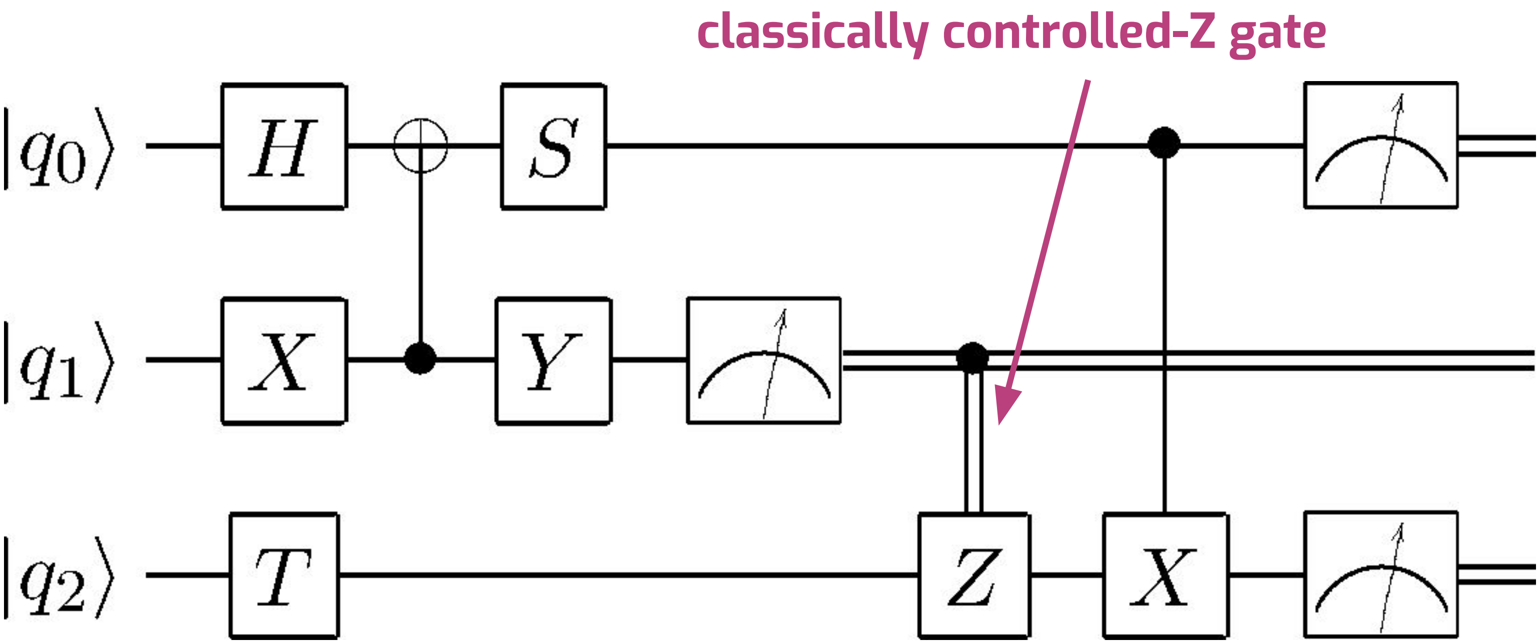
controlled-NOT (CNOT)

Quantum circuits

Bloch sphere | Quantum gates | **Quantum circuits** | Notation



Quantum circuits



Rigetti's shared memory makes such operations very easy!

Notation

$$|000\rangle$$

Is the **super lazy** physicist's version of

$$|0\rangle\,|0\rangle\,|0\rangle$$

which is the **lazy** physicist's version of

$$|0\rangle \otimes |0\rangle \otimes |0\rangle$$

Notation

$$|000\rangle$$

Qubit **2** Qubit **1** Qubit **0**

This is a **common convention** in quantum computing.
You will explore the reasoning behind this in the next tutorial.

Notation

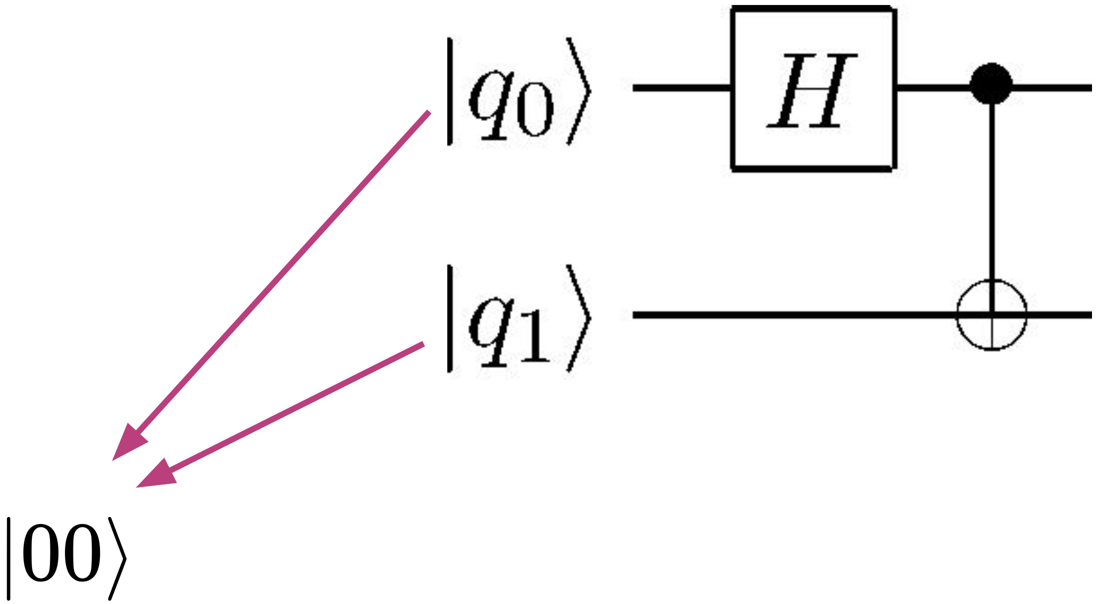
$$Z_0 \otimes Z_1 \otimes Z_2$$

is the **lazy** physicist's version of

$$\sigma_0^Z \otimes \sigma_1^Z \otimes \sigma_2^Z$$

Pen & Paper quantum computing

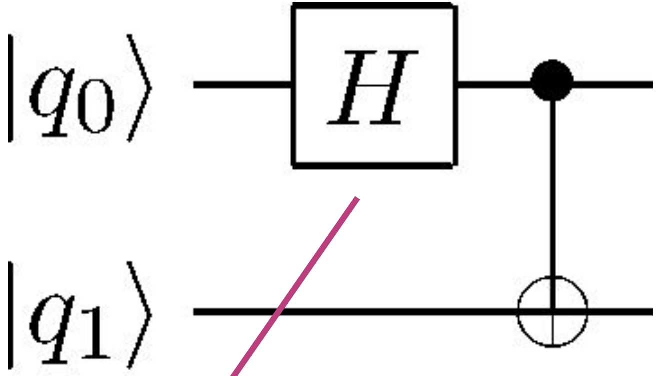
Bloch sphere | Quantum gates | Quantum circuits | Notation



To finish it up, let's put it all together and
walk through a **quantum circuit with pen & paper!**

Pen & Paper quantum computing

Bloch sphere | Quantum gates | Quantum circuits | Notation

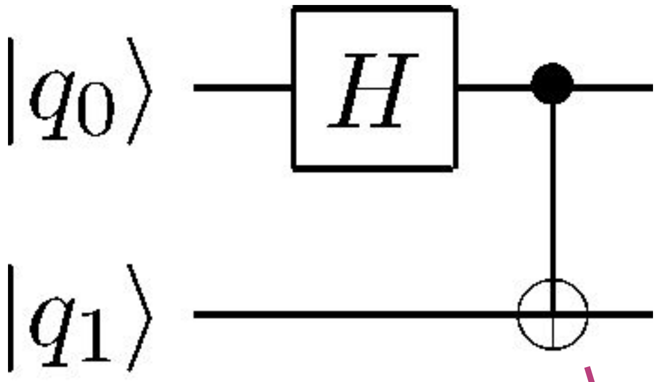


$$|00\rangle \quad \rightarrow \quad (\mathbb{I} \otimes H)\,|00\rangle$$

In a multi-qubit system don't forget the **tensor product with the identity matrix!**

Pen & Paper quantum computing

Bloch sphere | Quantum gates | Quantum circuits | Notation

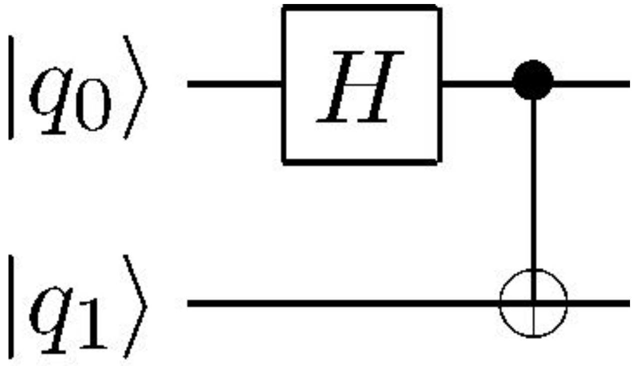


$$|00\rangle \quad \rightarrow \quad (\mathbb{I} \otimes H)\,|00\rangle \;\rightarrow\; \mathrm{CNOT}(0,1)(\mathbb{I} \otimes H)\,|00\rangle$$

Pay attention to the **order of operations!**

Pen & Paper quantum computing

Bloch sphere | Quantum gates | Quantum circuits | Notation

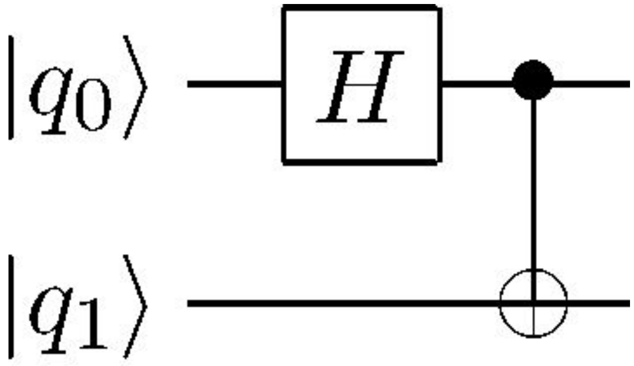


$$\mathrm{CNOT}(0,1)(\mathbb{I}\otimes H)\,|00\rangle = \mathrm{CNOT}(0,1)\left[\frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|01\rangle\right]$$

Apply the Hadamard to the 0th qubit only and don't apply CNOT yet.

Pen & Paper quantum computing

Bloch sphere | Quantum gates | Quantum circuits | Notation



$$\mathrm{CNOT}(0,1)(\mathbb{I} \otimes H)\,|00\rangle = \mathrm{CNOT}(0,1)\left[\frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|01\rangle\right]$$

$$= \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle = |\Phi^+\rangle$$

Applying the CNOT results in a maximally entangled state.
It's often called the **first Bell state.**

rigetti

Now, please start working through the exercises in the Jupyter Notebook for **Tutorial 2**

Brought to you by

in /tbabej
@tomasbabej

in /m-fingerhuth
@mark_fingerhuth

ProteinQure