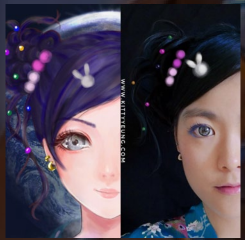
Introduction to Quantum Computing



Kitty Yeung, Ph.D. in Applied Physics

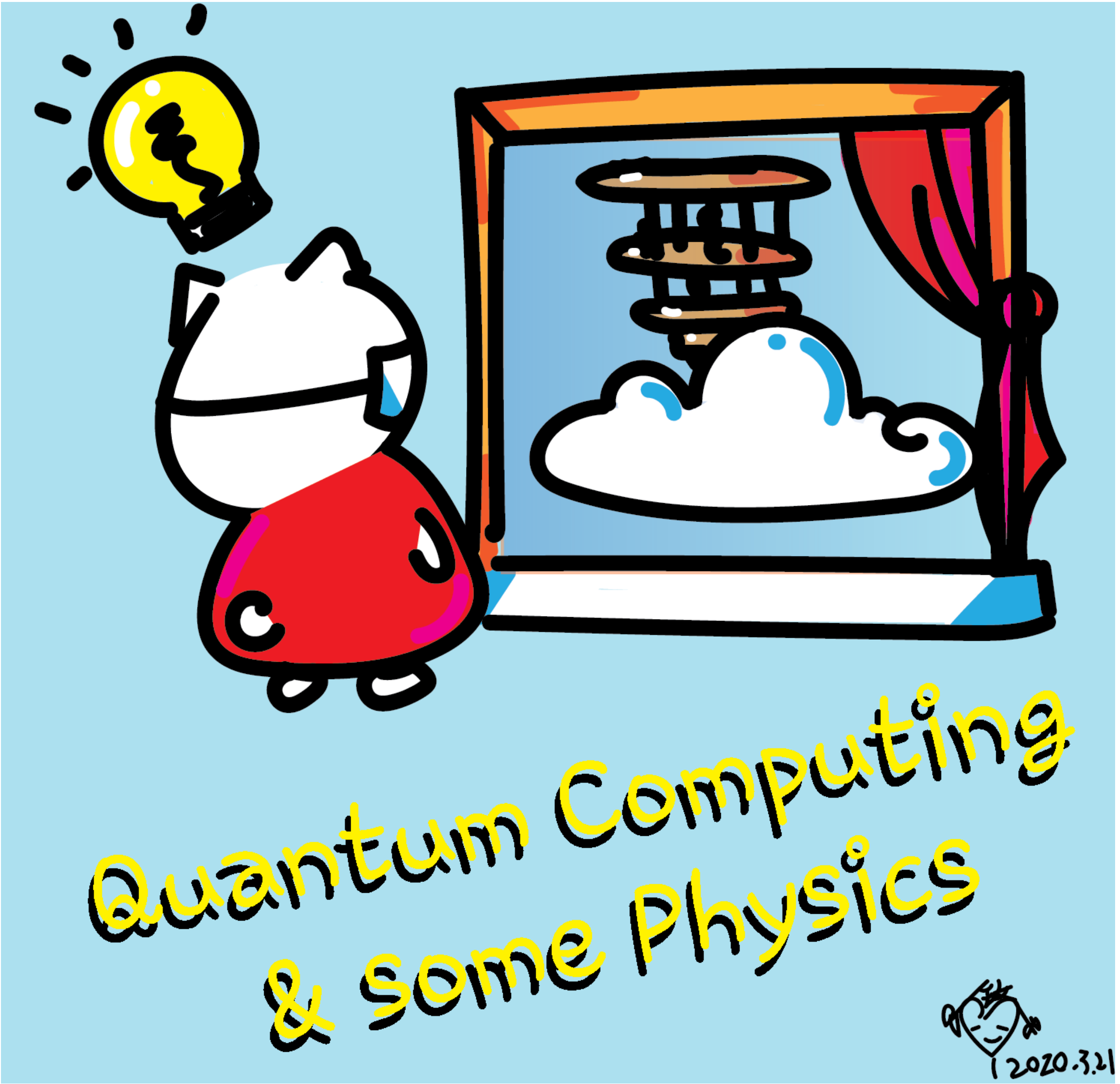
Creative Technologist + Sr. PM
Microsoft

www.artbyphysicistkittyyeung.com
@KittyArtPhysics
@artbyphysicistkittyyeung

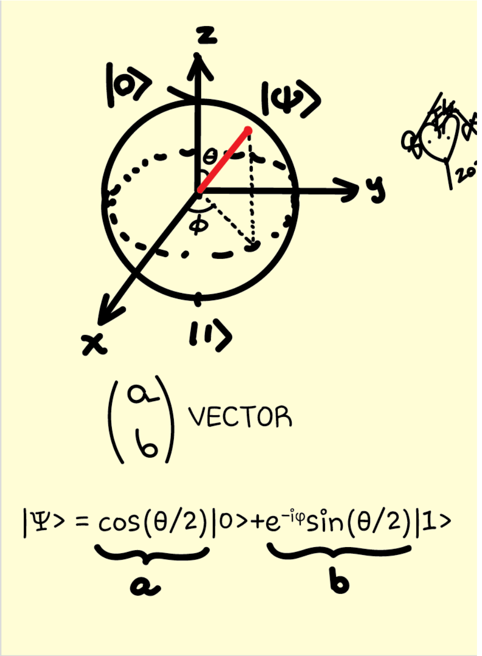
April 26, 2020
Hackaday, session 5

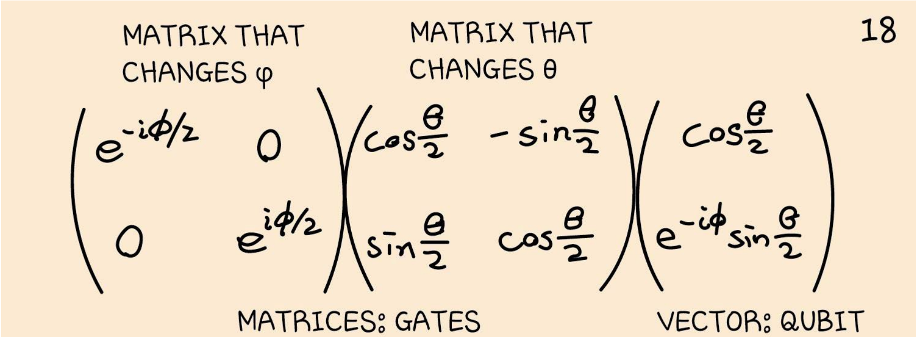
Class structure

- [Comics on Hackaday – Introduction to Quantum Computing](#) every Wed & Sun
- 30 mins every Sun, one concept (theory, hardware, programming), Q&A
- Contribute to Q# documentation http://docs.microsoft.com/quantum
- Coding through Quantum Katas https://github.com/Microsoft/QuantumKatas/
- Discuss in Hackaday project comments throughout the week
- Take notes



Gates (quantum operations)





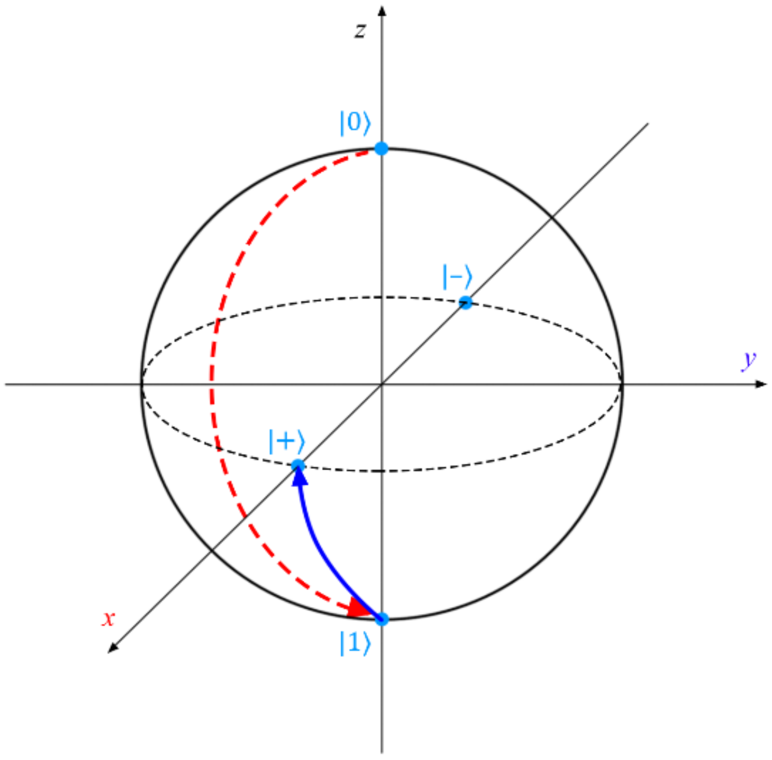
Hadamard H

$$H = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}$$

$$H|0\rangle = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$= \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \frac{1}{\sqrt{2}} \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

$$= \frac{1}{\sqrt{2}} (|0\rangle + |1\rangle) \equiv |+\rangle$$



$H|0\rangle = |+\rangle$

$$H|1\rangle = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

$$= \frac{1}{\sqrt{2}} (|0\rangle - |1\rangle) \equiv |-\rangle \, .$$

Pauli gates

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

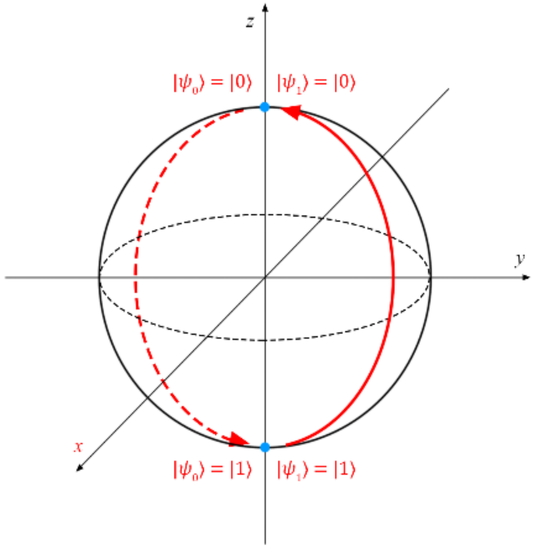
$$Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$$

$$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

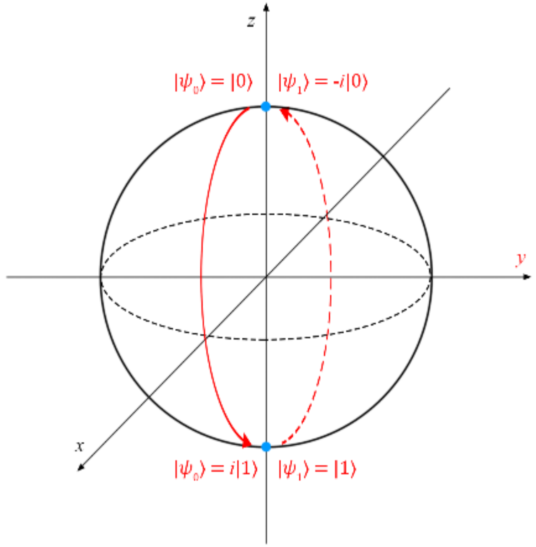
$$X \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} \beta \\ \alpha \end{pmatrix}$$

$$Y \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = i \begin{pmatrix} -\beta \\ \alpha \end{pmatrix}$$

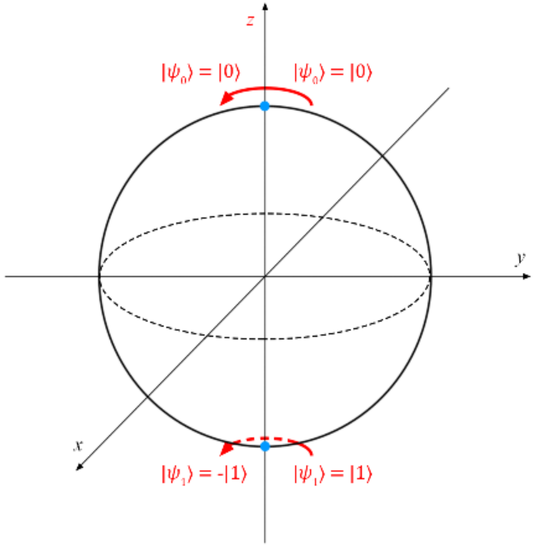
$$Z \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} \alpha \\ -\beta \end{pmatrix}$$



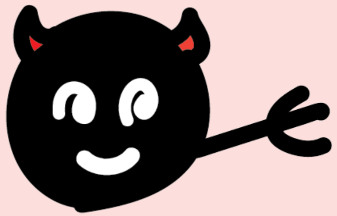
Pauli *X*



Pauli *Y*



Pauli *Z*



CONTROL QUBIT:
YOU STAY THE SAME IF I'M $|0\rangle$;
YOU CHANGE IF I'M $|1\rangle$.

TARGET QUBIT:
OKAY~

PRESERVE

$$\mathrm{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

A 4X4 MATRIX

SWITCH

The controlled-not gate manipulates the target qubit based on the state of the control qubit.

$\mathrm{CNOT}|00\rangle=|00\rangle$
$\mathrm{CNOT}|01\rangle=|01\rangle$
$\mathrm{CNOT}|10\rangle=|11\rangle$
$\mathrm{CNOT}|11\rangle=|10\rangle$

There are other controlled gates for multiple qubits you should look up.
We highlight CNOT as it will be used in every(?) algorithm (sounds familiar?!)

CNOT

$$CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Math insert - Matrix multiplication --

Gates are N by N matrices that multiply to state with 2^N vector elements. They follow the rules such that

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} ax + by \\ cx + dy \end{pmatrix},$$

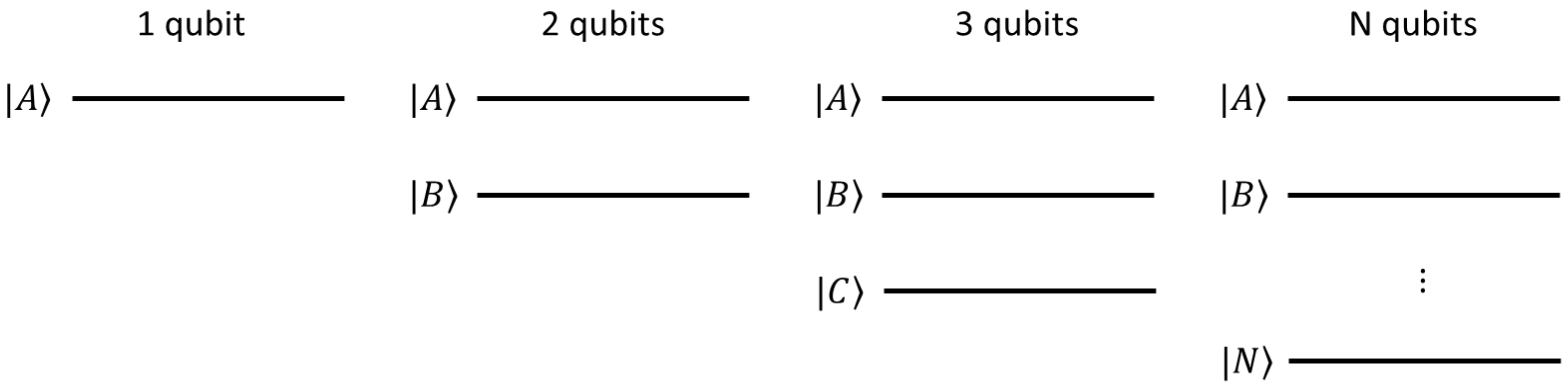
$$\begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix}\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} ax + by + cz \\ dx + ey + fz \\ gx + hy + iz \end{pmatrix},$$

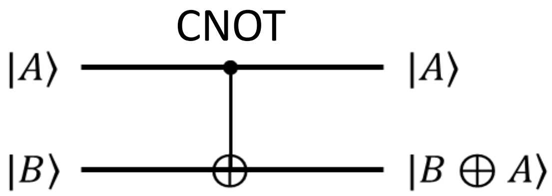
and so on.

$$CNOT|10\rangle = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}\begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = |11\rangle\,.$$

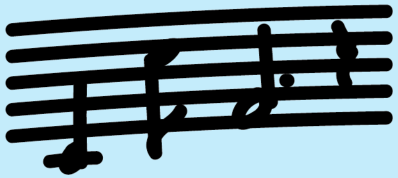
Similarly, $C|00\rangle = |00\rangle, C|01\rangle = |01\rangle$ and $C|11\rangle = |10\rangle$.

Circuit representation



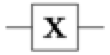
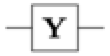
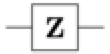
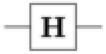
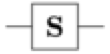
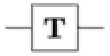
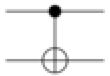
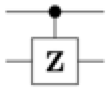
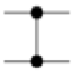
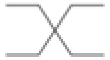
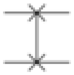
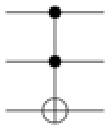


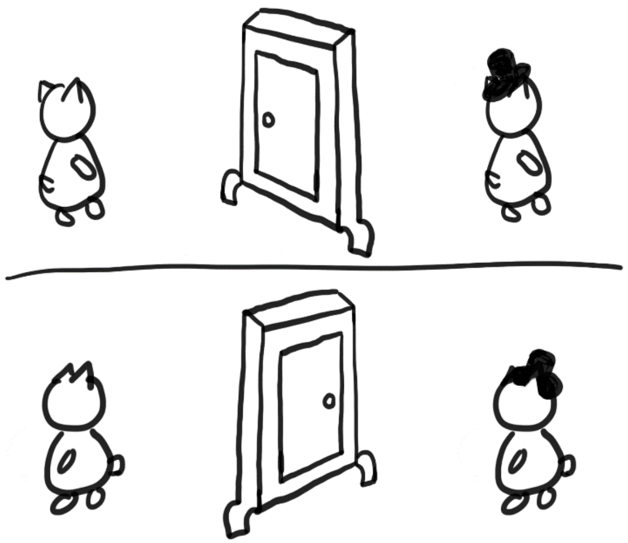
Target B controlled by A



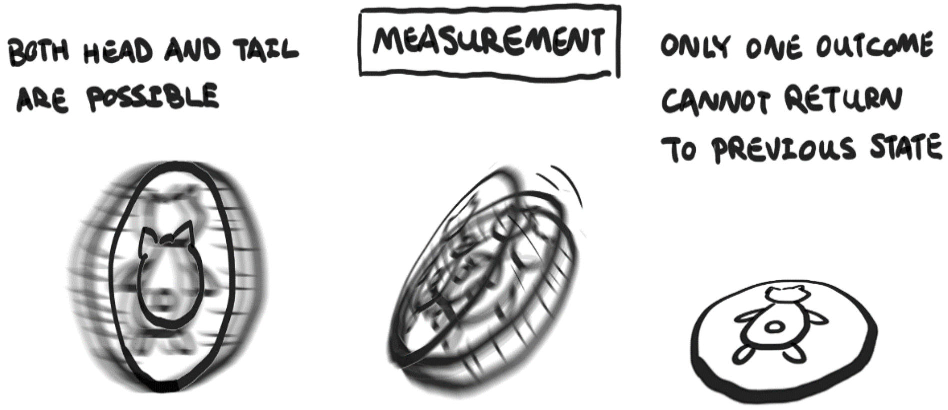
The Bloch sphere is no longer useful when we look at more than one qubit. But we have another graphic representation to use for multi-qubit systems.

Similar to how the lines in music scores denote the time-evolving music, we can use lines to represent the time-evolving qubit states:

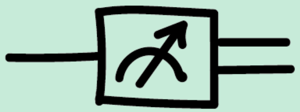
Operator	Gate(s)		Matrix
Pauli-X (X)			$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$
Pauli-Y (Y)			$\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$
Pauli-Z (Z)			$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$
Hadamard (H)			$\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$
Phase (S, P)			$\begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$
$\pi/8$ (T)			$\begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}$
Controlled Not (CNOT, CX)			$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$
Controlled Z (CZ)			$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$
SWAP			$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
Toffoli (CCNOT, CCX, TOFF)			$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$



Reversible

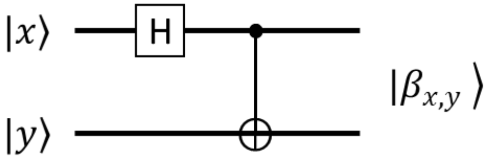


Not reversible



This is the circuit representation for measurement. It is not a gate. The output is a classical result, denoted by a double line.

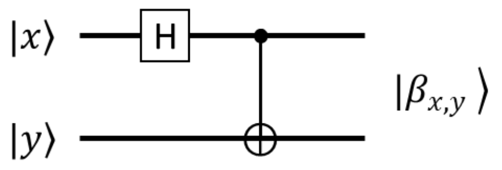
Creating Bell states (entanglement)



In	Out
$\|00\rangle$	$(\|00\rangle + \|11\rangle)/\sqrt{2} \equiv \|\beta_{00}\rangle$
$\|01\rangle$	$(\|01\rangle + \|10\rangle)/\sqrt{2} \equiv \|\beta_{01}\rangle$
$\|10\rangle$	$(\|00\rangle - \|11\rangle)/\sqrt{2} \equiv \|\beta_{10}\rangle$
$\|11\rangle$	$(\|01\rangle - \|10\rangle)/\sqrt{2} \equiv \|\beta_{11}\rangle$

Try proving this table

Creating Bell states (entanglement)



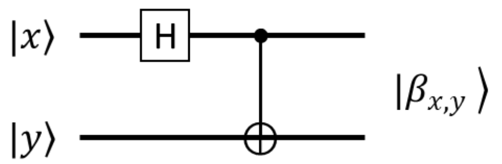
$$H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

$$H|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

In	Out
$\|00\rangle$	$(\|00\rangle + \|11\rangle)/\sqrt{2} \equiv \|\beta_{00}\rangle$
$\|01\rangle$	$(\|01\rangle + \|10\rangle)/\sqrt{2} \equiv \|\beta_{01}\rangle$
$\|10\rangle$	$(\|00\rangle - \|11\rangle)/\sqrt{2} \equiv \|\beta_{10}\rangle$
$\|11\rangle$	$(\|01\rangle - \|10\rangle)/\sqrt{2} \equiv \|\beta_{11}\rangle$

Try proving this table

Creating Bell states (entanglement)



In	Out
$\|00\rangle$	$(\|00\rangle + \|11\rangle)/\sqrt{2} \equiv \|\beta_{00}\rangle$
$\|01\rangle$	$(\|01\rangle + \|10\rangle)/\sqrt{2} \equiv \|\beta_{01}\rangle$
$\|10\rangle$	$(\|00\rangle - \|11\rangle)/\sqrt{2} \equiv \|\beta_{10}\rangle$
$\|11\rangle$	$(\|01\rangle - \|10\rangle)/\sqrt{2} \equiv \|\beta_{11}\rangle$

Try proving this table

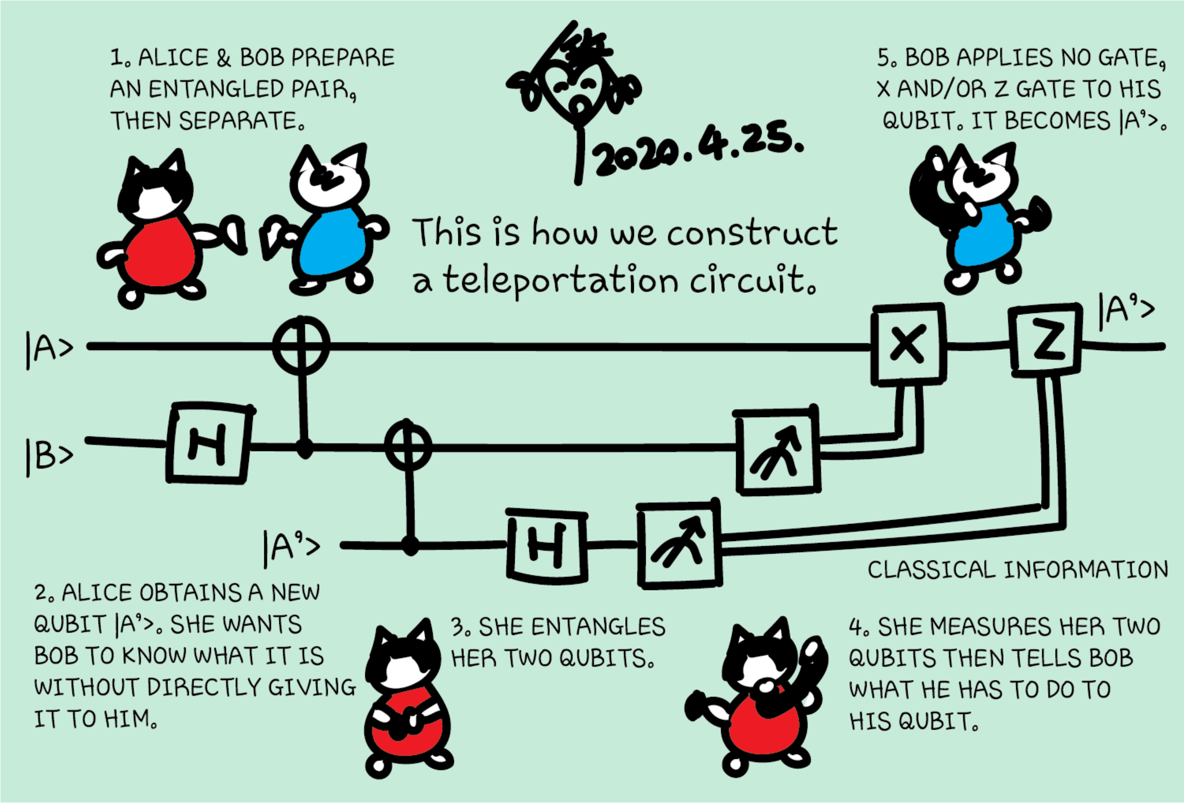
$$H|0\rangle\,|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)|0\rangle$$

$$H|0\rangle\,|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)|1\rangle$$

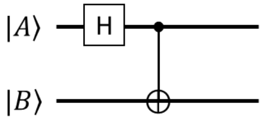
$$H|1\rangle\,|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)|0\rangle$$

$$H|1\rangle\,|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)|1\rangle$$

Apply CNOT

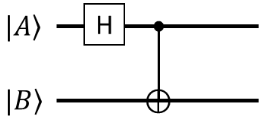


First two qubits	Third qubit	Alice tells Bob to
00	$[\alpha\|0\rangle + \beta\|1\rangle]$	do nothing
01	$[\alpha\|1\rangle + \beta\|0\rangle]$	apply X
10	$[\alpha\|0\rangle - \beta\|1\rangle]$	apply Z
11	$[\alpha\|1\rangle - \beta\|0\rangle]$	apply X and Z



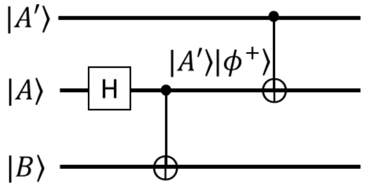
$$|\phi^+\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}}$$

Teleportation



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

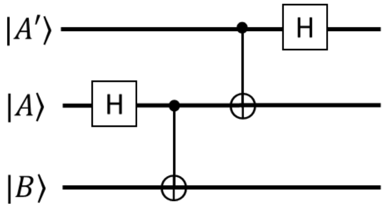
Let $|A'\rangle = \alpha|0\rangle + \beta|1\rangle$



$$|A'\rangle|\phi^+\rangle = (\alpha|0\rangle + \beta|1\rangle)\,\frac{|00\rangle+|11\rangle}{\sqrt{2}}$$

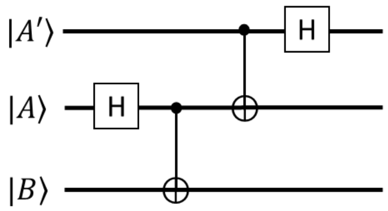
$$= \frac{1}{\sqrt{2}}(\alpha|000\rangle + \alpha|011\rangle + \beta|100\rangle + \beta|111\rangle).$$

$$CNOT|A'\rangle|\phi^+\rangle = \frac{1}{\sqrt{2}}(\alpha|000\rangle + \alpha|011\rangle + \beta|110\rangle + \beta|101\rangle)$$

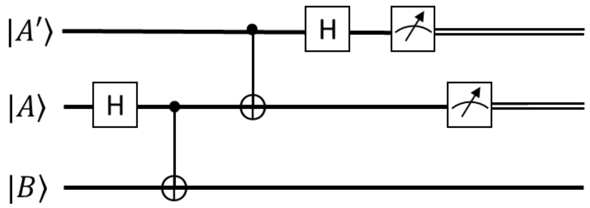


$$\frac{1}{2}[|00\rangle(\alpha|0\rangle + \beta|1\rangle) + |01\rangle(\alpha|1\rangle + \beta|0\rangle) + |10\rangle(\alpha|0\rangle - \beta|1\rangle) + |11\rangle(\alpha|1\rangle - \beta|0\rangle)]$$

$$\frac{1}{\sqrt{2}}\left[\alpha\left(\frac{|0\rangle+|1\rangle}{\sqrt{2}}\right)|00\rangle + \alpha\left(\frac{|0\rangle+|1\rangle}{\sqrt{2}}\right)|11\rangle + \beta\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right)|10\rangle + \beta\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right)|01\rangle\right]$$



$$\frac{1}{2}[|00\rangle(\alpha|0\rangle + \beta|1\rangle) + |01\rangle(\alpha|1\rangle + \beta|0\rangle) + |10\rangle(\alpha|0\rangle - \beta|1\rangle) + |11\rangle(\alpha|1\rangle - \beta|0\rangle)]$$



If the first qubit is 0, the state after measurement becomes

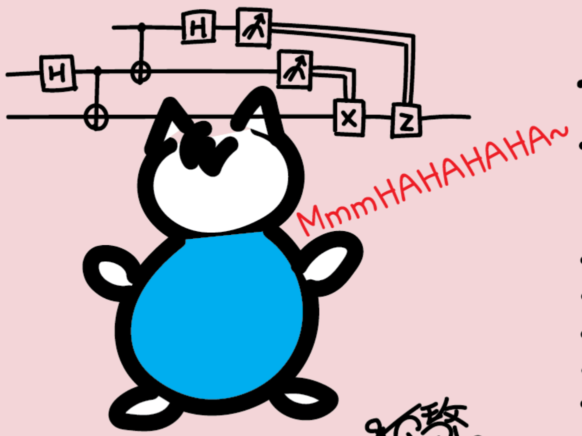
$$\frac{1}{2}[|00\rangle(\alpha|0\rangle + \beta|1\rangle) + |01\rangle(\alpha|1\rangle + \beta|0\rangle)] .$$

If then another measurement is done on the second qubit and it is 0, the state becomes

$$\frac{1}{2}[|00\rangle(\alpha|0\rangle + \beta|1\rangle)] .$$

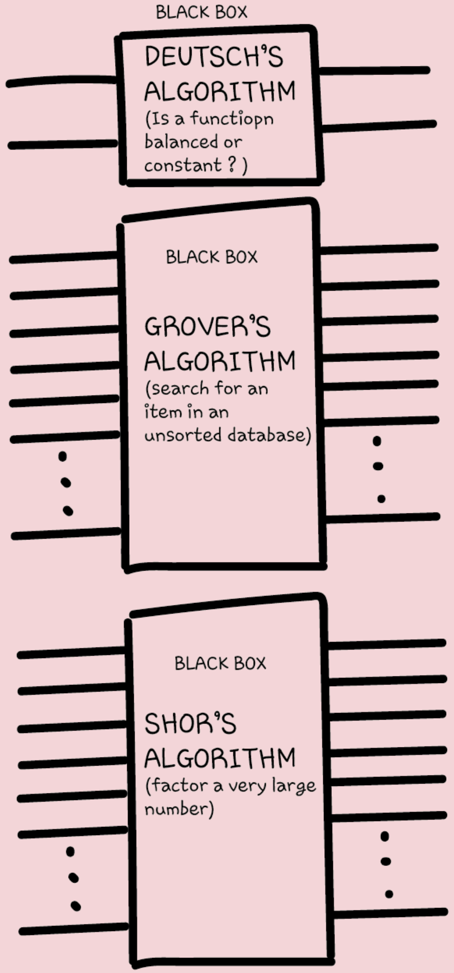
This also tells us that the third qubit is in state $[\alpha|0\rangle + \beta|1\rangle]$.

First two qubits	Third qubit	Alice tells Bob to
00	$[\alpha|0\rangle + \beta|1\rangle]$	do nothing
01	$[\alpha|1\rangle + \beta|0\rangle]$	apply X
10	$[\alpha|0\rangle - \beta|1\rangle]$	apply Z
11	$[\alpha|1\rangle - \beta|0\rangle]$	apply X and Z



Some algorithms are neatly presented with circuits.

But as the number of qubits and complexity of algorithms grow, it's not feasible to always write down each qubit and construct circuits.



Just like we wouldn't describe each transistor in a classical computer when we write a program. We need a high-level quantum computing language to program quantum computers.

Q# exercise: option 1

No installation, web-based Jupyter Notebooks

- The Quantum Katas project (tutorials and exercises for learning quantum computing) https://github.com/Microsoft/QuantumKatas

- Tutorials

- BasicGates
- Superposition
- Measurements
- Teleportation

- SuperdenseCoding
- DeutschJozsaAlgorithm
- GroversAlgorithm
- SimonsAlgorithm

Coming next

- Hardware
- Twitch?
- Q# and Algorithms