

Open Source Quantum Computing

Matthew Treinish
Software Engineer - IBM Research

mtreinish@kortar.org

[mtreinish on Freenode](#)

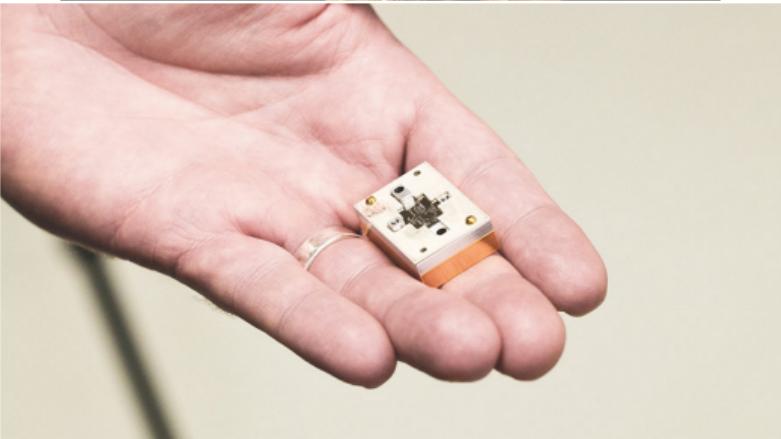
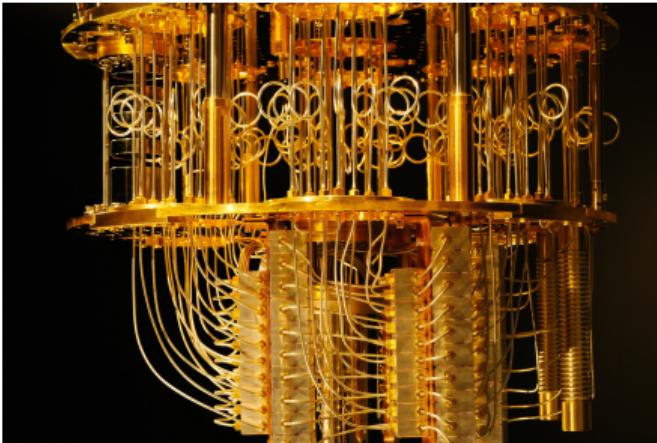
<https://github.com/mtreinish/open-source-quantum-computing>

January 23, 2019

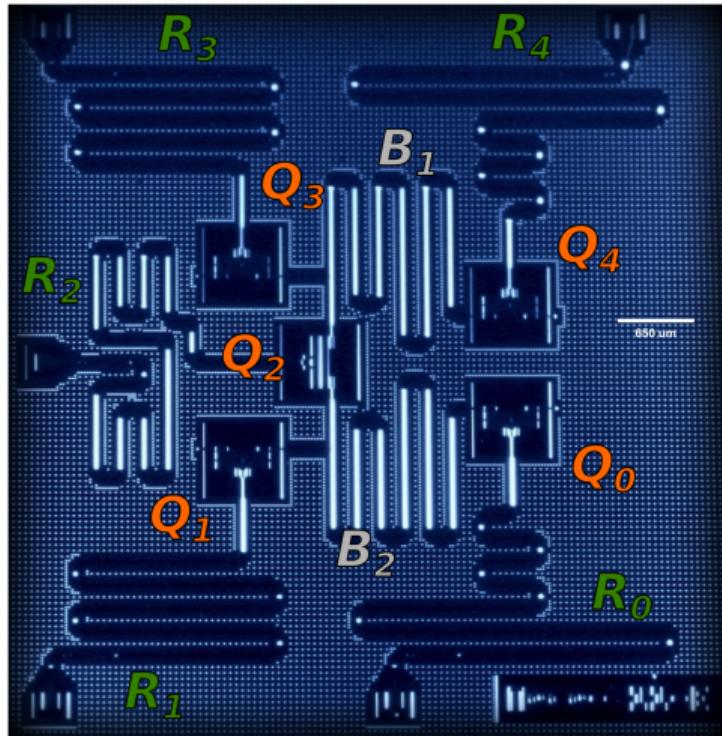
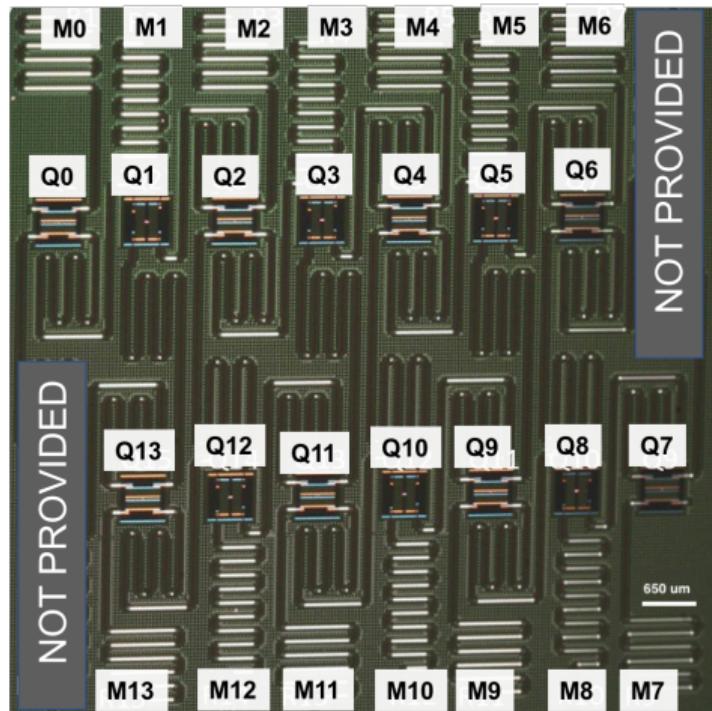




Real Quantum Computer

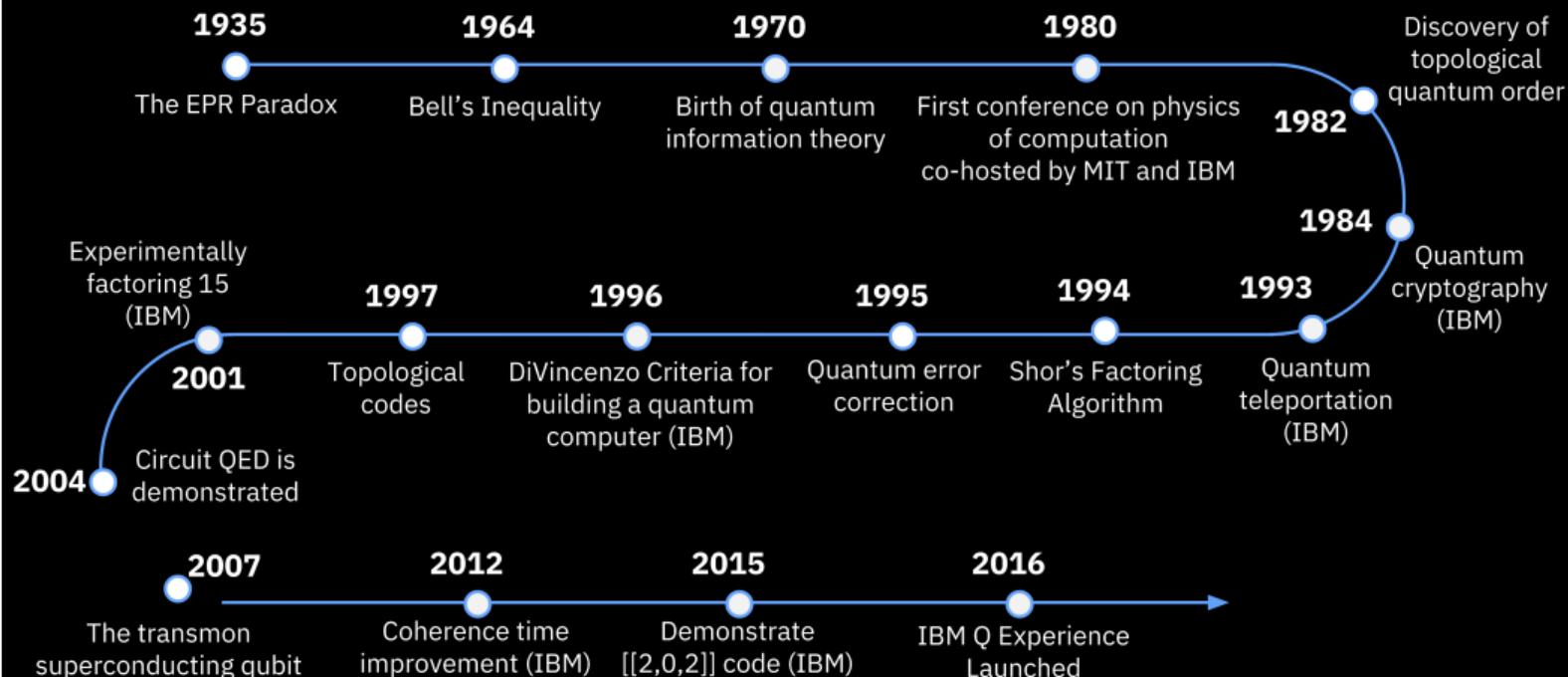


Quantum Chips



<https://github.com/Qiskit/ibmq-device-information>

History of Quantum Computing

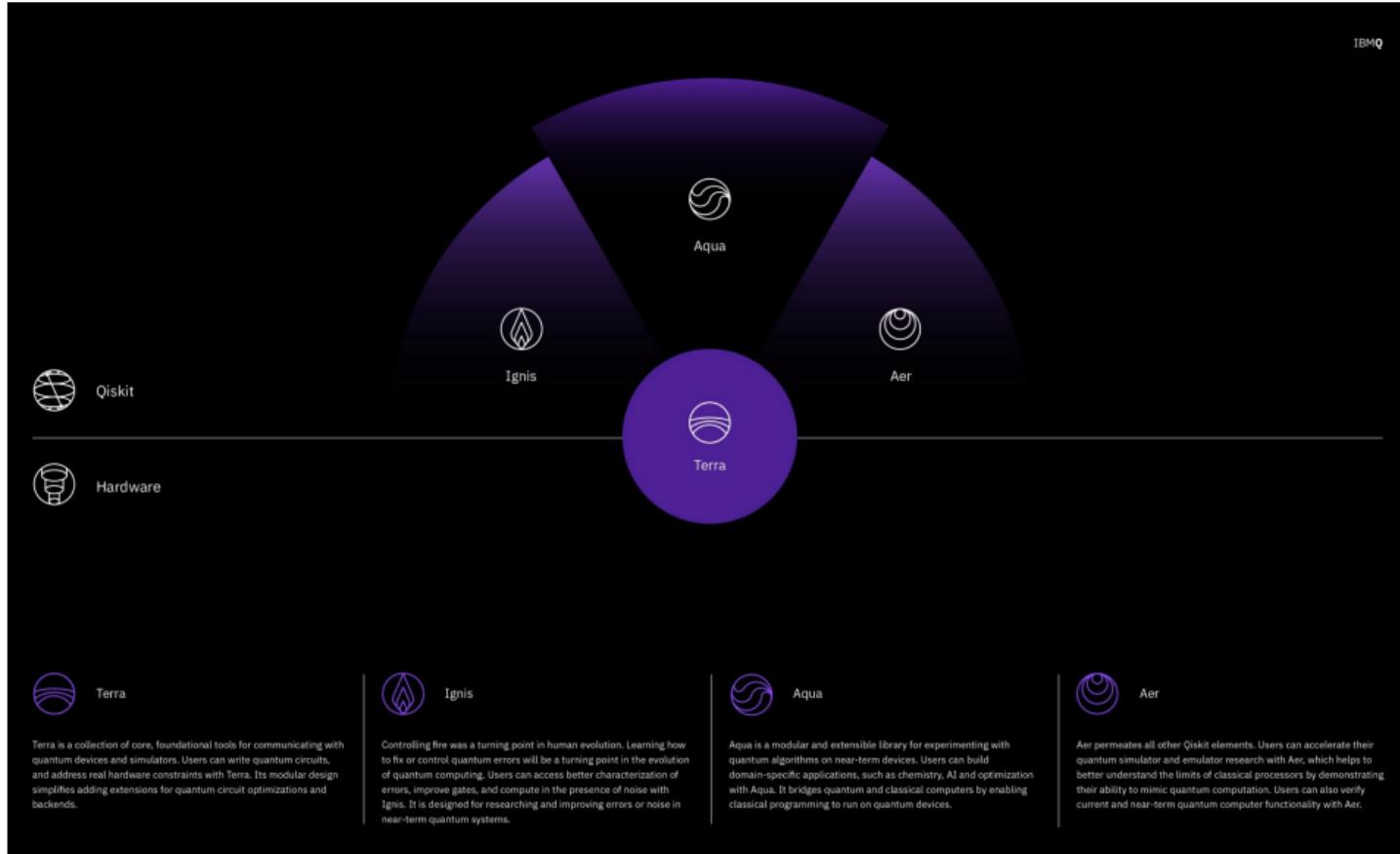


What is Qiskit?

- ▶ SDK for working with Noisy Intermediate-Scale Quantum (NISQ) computers
- ▶ Apache 2.0 License
- ▶ Designed to be backend agnostic
- ▶ Includes out-of-the-box local simulators and support for running on IBMQ



Qiskit Elements



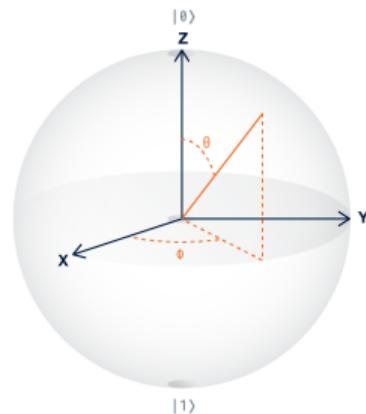
- ▶ Is the base layer for applications, provides interface to hardware and simulators
- ▶ Provides an SDK for working with quantum circuits
- ▶ Compiles circuits to run on different backends
- ▶ Written in Python



The Qubit

- ▶ The bloch sphere provides a representation of qubit state
- ▶ State can be at any point along surface of sphere
- ▶ Measuring a qubit occurs along the Z axis. (also called basis states)
- ▶ Measuring a qubit is irreversible and will either be 0 or 1
- ▶ Qubits initialized to $|0\rangle$

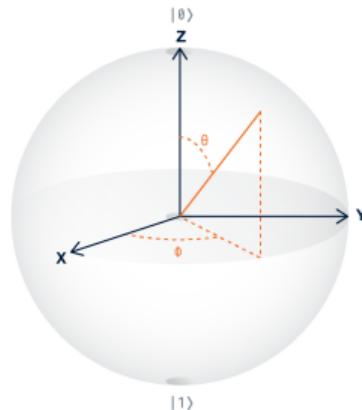
Bloch Sphere:



Qubit Phase

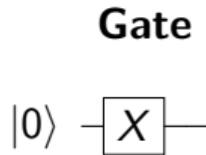
Phase is ϕ :

- ▶ While qubits are read along the basis vectors you can still use the other dimensions
- ▶ The phase can be leveraged to encode more information in the qubit



Quantum Gates

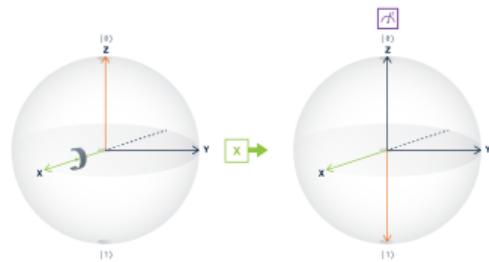
- ▶ Quantum Logic Gates are used perform operations on qubits
- ▶ Gates are reversible
- ▶ Gates can be represented as unitary matrices



Matrix Form

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

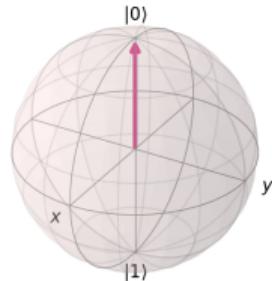
Bloch Sphere



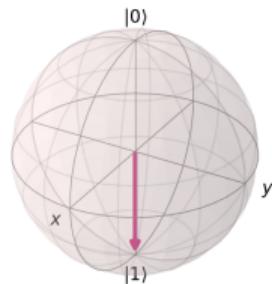
Superposition

- ▶ Identically prepared qubits can still behave randomly
- ▶ The randomness is inherent in nature

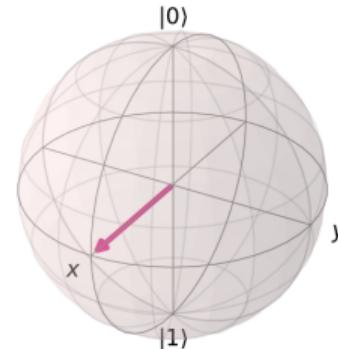
$|0\rangle$



$|1\rangle$



$|0\rangle + |1\rangle$

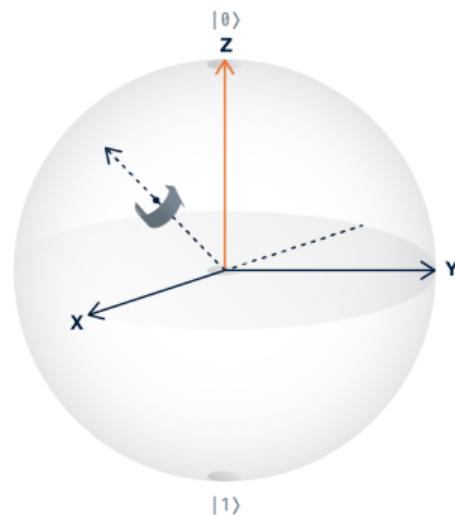


**50/50 chance of being
 $|0\rangle$ or $|1\rangle$**

Hadamard Gate

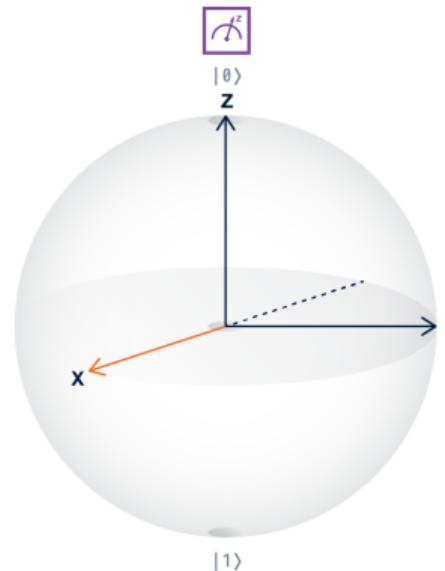
Gate

$|0\rangle \xrightarrow{H} |0\rangle$



Matrix Form

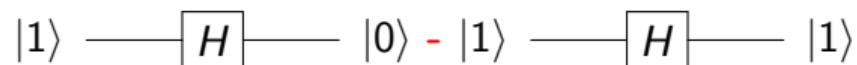
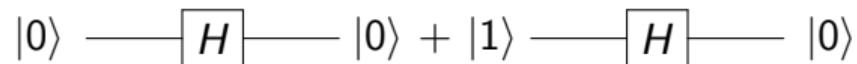
$$\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$



π rotation around
X+Z axis:
exchanges X and Z

Hadamard and Phase

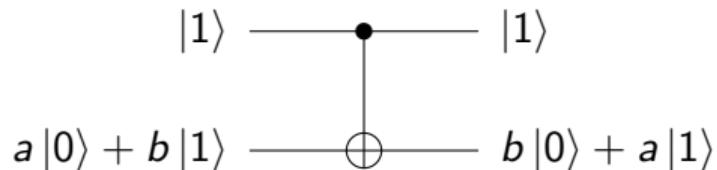
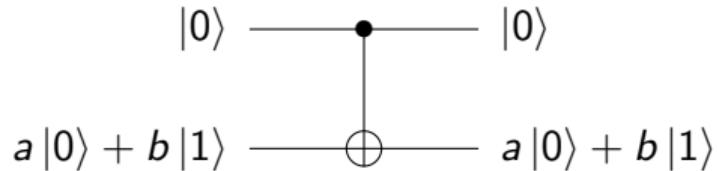
Hadamard gates are self inverses:



Phase

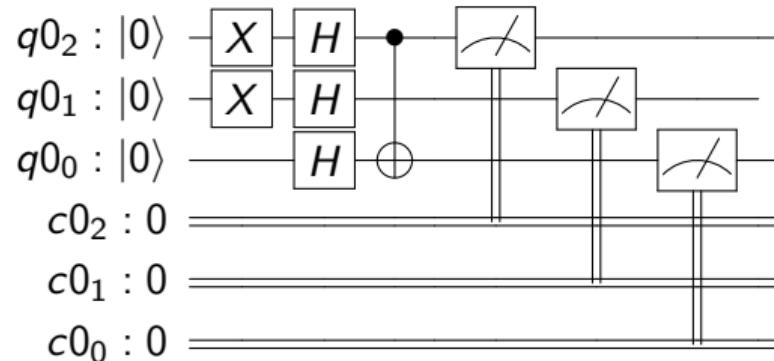
Controlled Not Gate

CNOT flips the *target* bit if the *control* bit is 1



Quantum Circuits

Putting it together you build a circuit like:



- ▶ Each row represents a bit, either quantum or classical
- ▶ The operations are performed each qubit left to right
- ▶ Shows dependencies of operations

Bernstein-Vazirani Algorithm¹



Input (query)
 $\Leftarrow X_{n-1} \dots X_1 X_0$

Secret Bitstring
 $S_{n-1} \dots S_1 S_0$

Output (result)
 $\Rightarrow X_{n-1}S_{n-1} \oplus \dots X_1S_1 \oplus X_0S_0$

The Oracle

¹E. Bernstein & U. Vazirani, STOC, 93

Optimal Classical Oracle

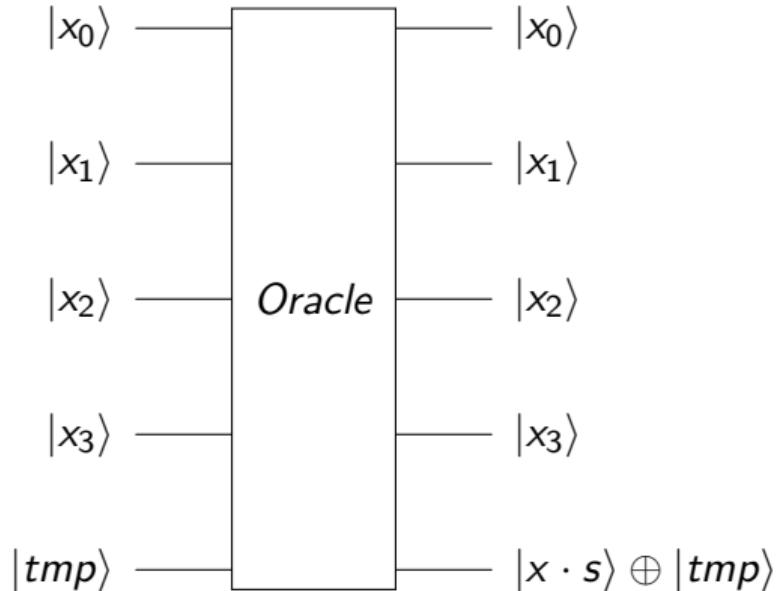


Loop over each bit!

$$\left\{ \begin{array}{ll} X = 1 0 \cdots 0 0 & (2^{n-1}) \\ X = 0 1 \cdots 0 0 & (2^{n-2}) \\ \vdots & \\ X = 0 0 \cdots 1 0 & (2) \\ X = 0 0 \cdots 0 1 & (1) \end{array} \right.$$

The ideal classical oracle is $\mathcal{O}(n)$

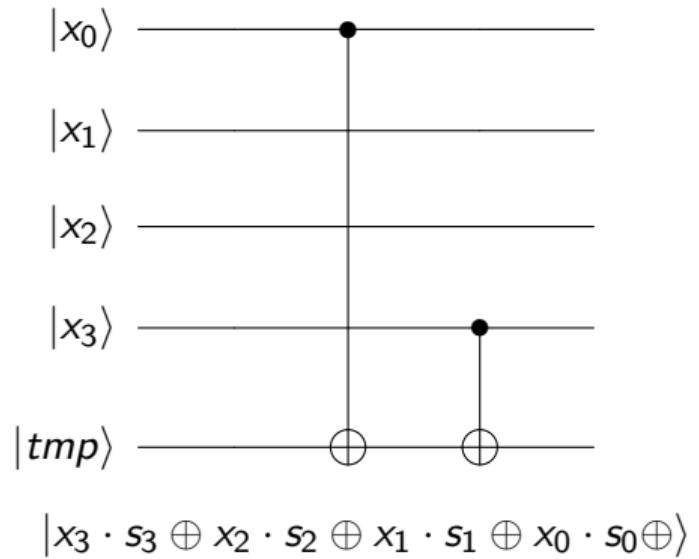
Quantum Oracle



A quantum oracle is $\mathcal{O}(1)$

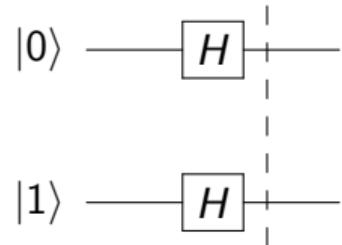
Quantum Oracle Implementation

$$S = 1001$$



Phase Kickback

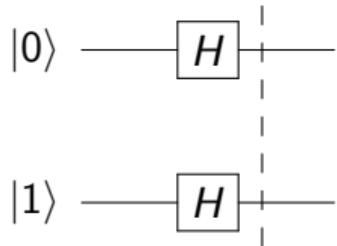
$$(|0\rangle + |1\rangle)(|0\rangle - |1\rangle)$$



Phase Kickback

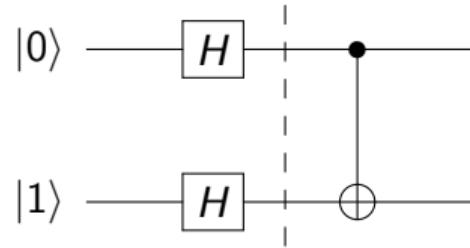
$$(|0\rangle + |1\rangle)(|0\rangle - |1\rangle) = |00\rangle - |01\rangle + |10\rangle - |11\rangle$$

Expand it



Phase Kickback

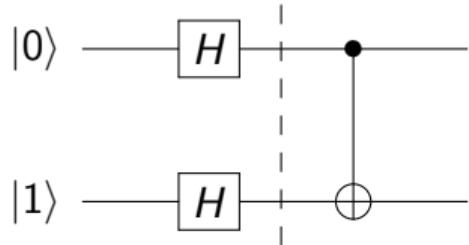
$$(|0\rangle + |1\rangle)(|0\rangle - |1\rangle) = |00\rangle - |01\rangle + |10\rangle - |11\rangle$$



$$|00\rangle - |01\rangle - |10\rangle + |11\rangle$$

Phase Kickback

$$(|0\rangle + |1\rangle)(|0\rangle - |1\rangle) = |00\rangle - |01\rangle + |10\rangle - |11\rangle$$

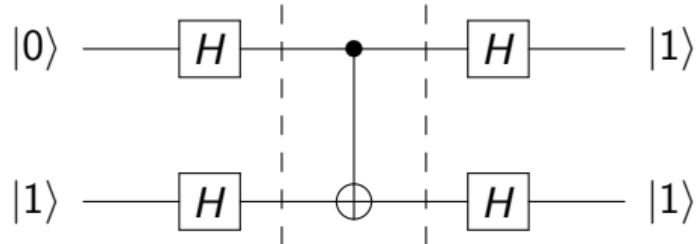


$$|00\rangle - |01\rangle - |10\rangle + |11\rangle = (|0\rangle - |1\rangle)(|0\rangle - |1\rangle)$$

Phase Kickback

Phase Kickback

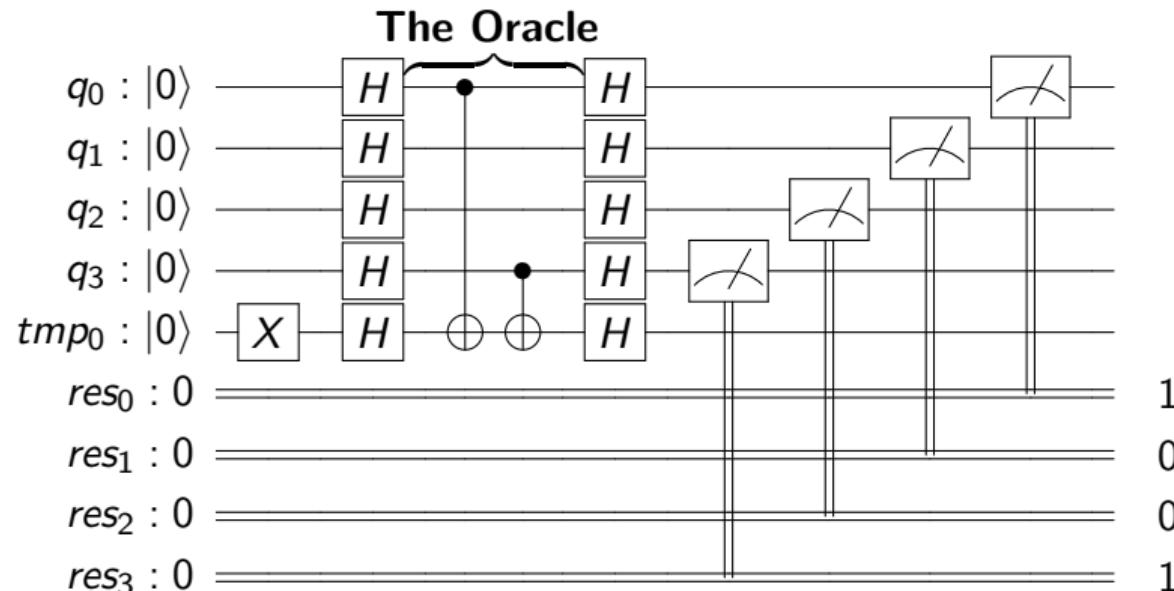
$$(|0\rangle + |1\rangle)(|0\rangle - |1\rangle) = |00\rangle - |01\rangle + |10\rangle - |11\rangle$$



$$|00\rangle - |01\rangle - |10\rangle + |11\rangle = (|0\rangle - |1\rangle)(|0\rangle - |1\rangle)$$

Phase Kickback

Full Circuit for a Quantum Oracle



Where there is a CNOT phase kickback will set the control qubit to state |1>

Live Demo

Open Source in Quantum Computing

- ▶ The tools for developing quantum programs are almost all open
- ▶ Foster collaboration while the field is just taking off
- ▶ Learn and build on history of development of classical computers

Other Open Source Tools

- ▶ <https://github.com/rigetticomputing/pyquil>
- ▶ <https://github.com/ProjectQ-Framework/ProjectQ>
- ▶ <https://github.com/quantumlib/Cirq>
- ▶ <https://github.com/qutip/qutip>
- ▶ <https://github.com/XanaduAI/strawberryfields>

A lot more out there: <https://github.com/topics/quantum-computing>

Conclusions

- ▶ Quantum Computing is about solving problems that we can't with classical computers
- ▶ It's not just in labs anymore, quantum computing is accessible by everyone now
- ▶ It's still very early for quantum computers
- ▶ Open source software is playing a key role early on

Where to get more information

- ▶ Qiskit: <https://qiskit.org/>
- ▶ IBM Q Experience: <https://quantumexperience.ng.bluemix.net/qx>
- ▶ Tutorials on Quantum Computing and Qiskit:
<https://github.com/Qiskit/qiskit-tutorials>
- ▶