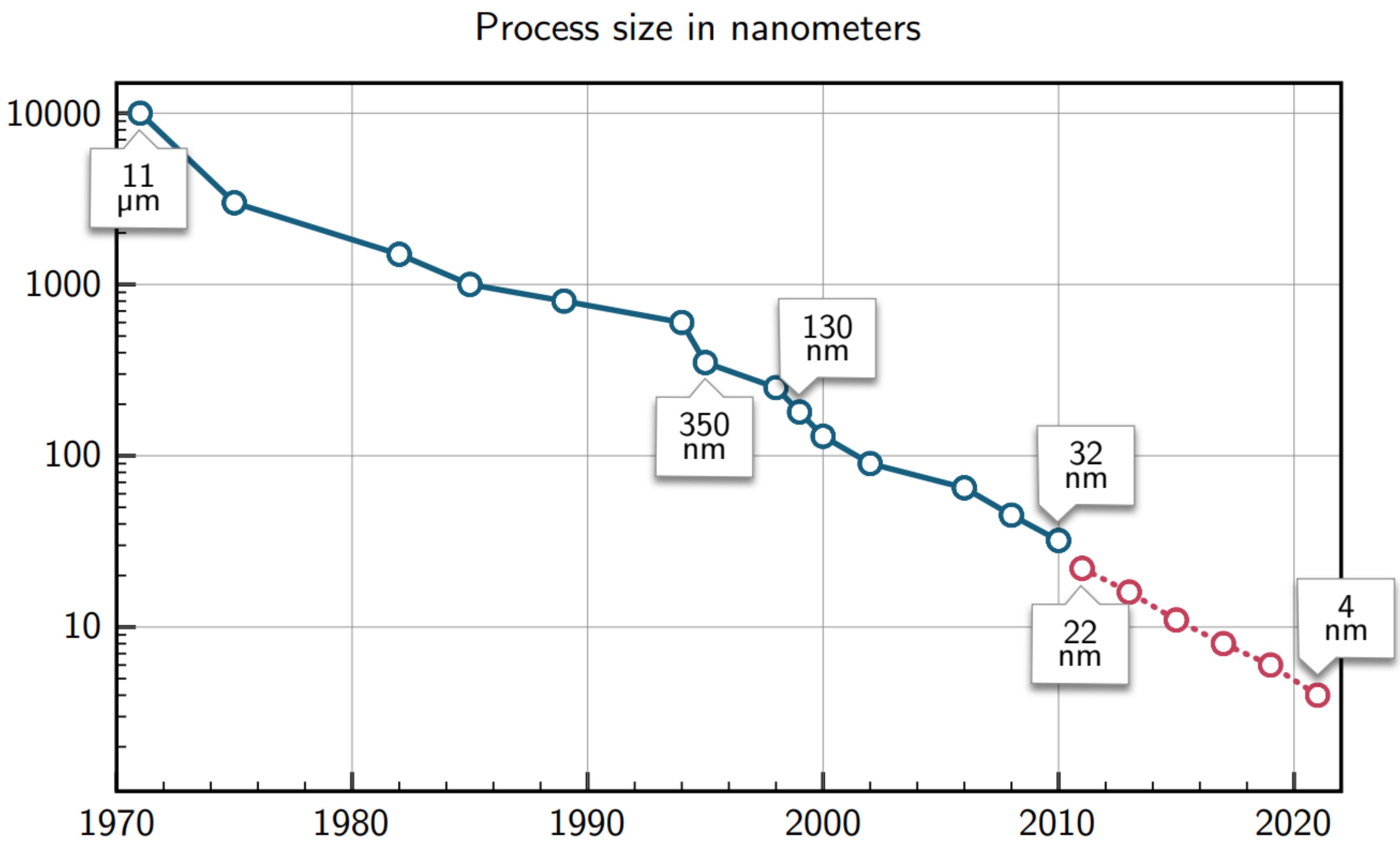
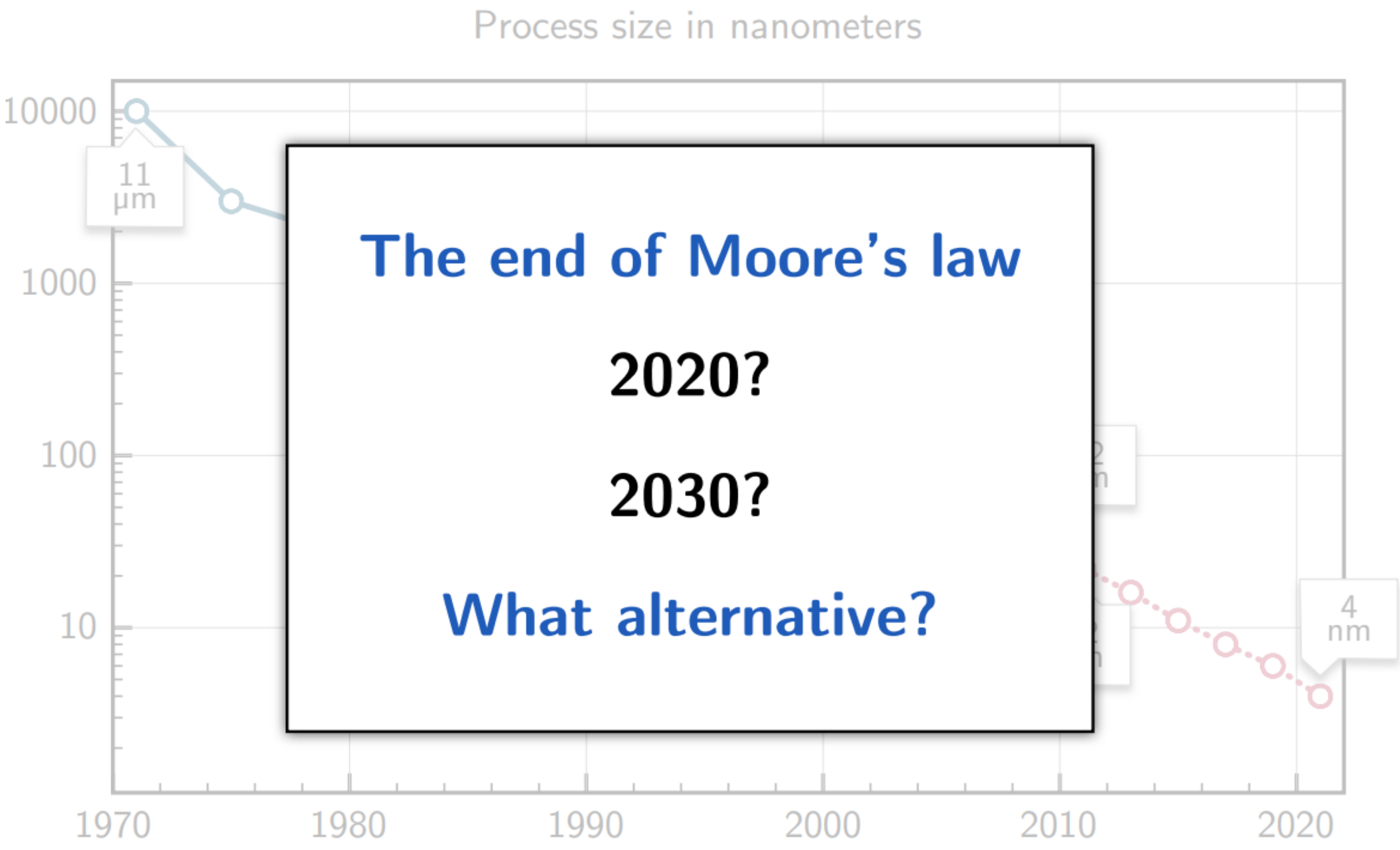
First Quantum Programming

Cheng Lin Hong

Moore’s Law



Process size in nanometers
10000
1000
100
10
1970
1980
1990
2000
2010
2020
11 µm
350 nm
130 nm
32 nm
22 nm
4 nm



The end of Moore's law

2020?

2030?

What alternative?

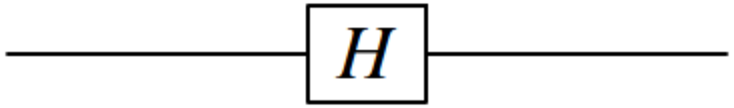
Qubits, Operators and Measurement

- A quantum computation is a collection of three elements
  ① A quantum register or a set of quantum register.
  ② A unitary matrix, which is used to execute a given quantum algorithm.
  ③ Measurement to extract information we need.

- Quantum circuit model
  - Universal quantum computer

The superposition

Hadamard can be used to create Quantum superposition

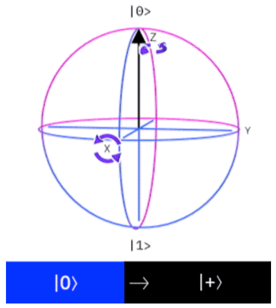



```
q_circ.h(0)
```

Hadamard

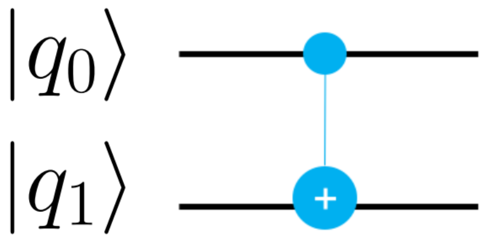
$$\frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}\begin{pmatrix}1\\ 0\end{pmatrix} = \frac{|0\rangle + |1\rangle}{\sqrt{2}}$$

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



Reference: https://quantum-computing.ibm.com/docs/circ-comp/q-gates

- Controlled-NOT

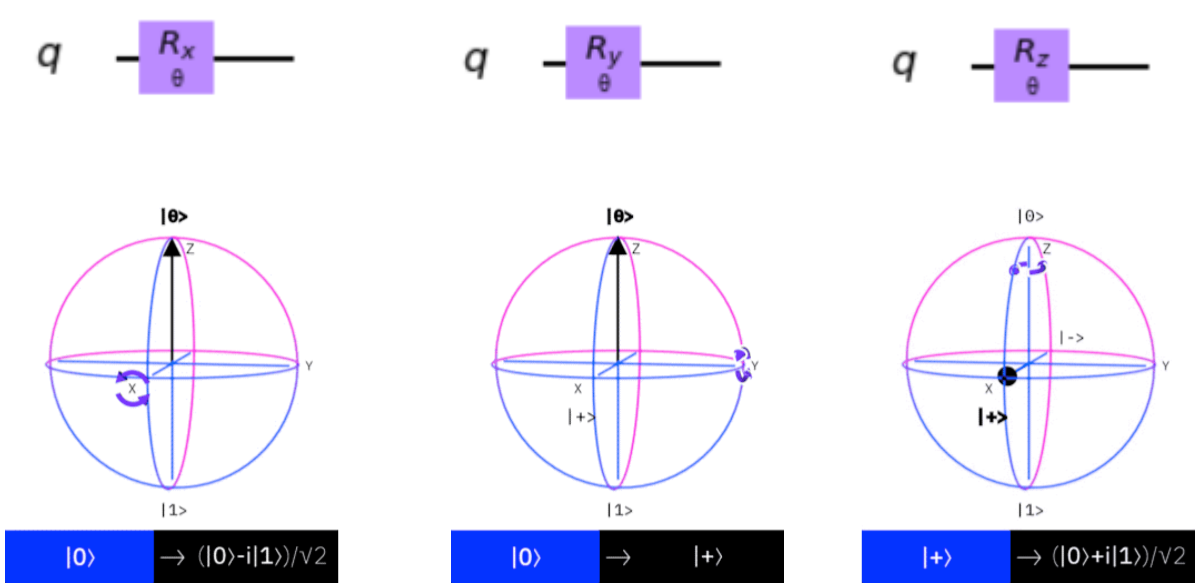


Starting state		Ending State
|00>	→	|00>
|10>	→	|10>
|01>	→	|11>
|11>	→	|01>

$$|q_1, q_0\rangle = |q_1 \oplus q_0, q_0\rangle$$

```
q_circ.cx(0,1)
```

- Single Qubits Rotations



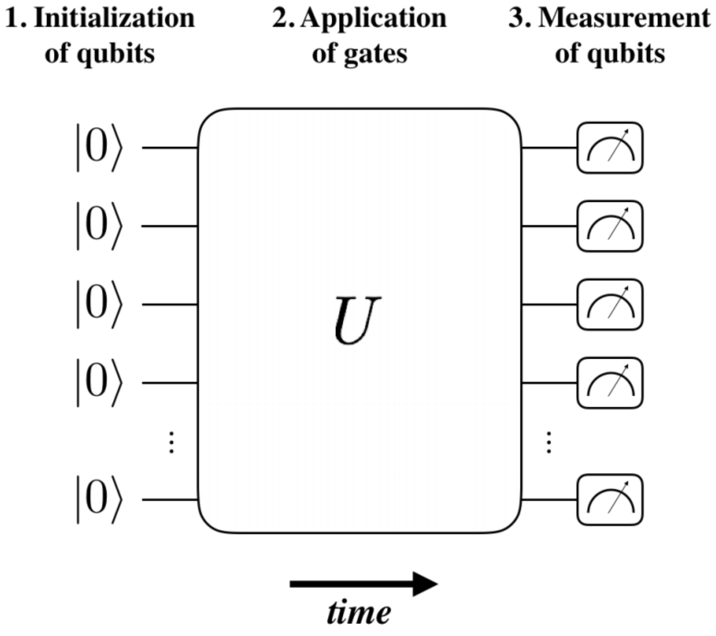

```
q_rot.rx(theta,0)
```

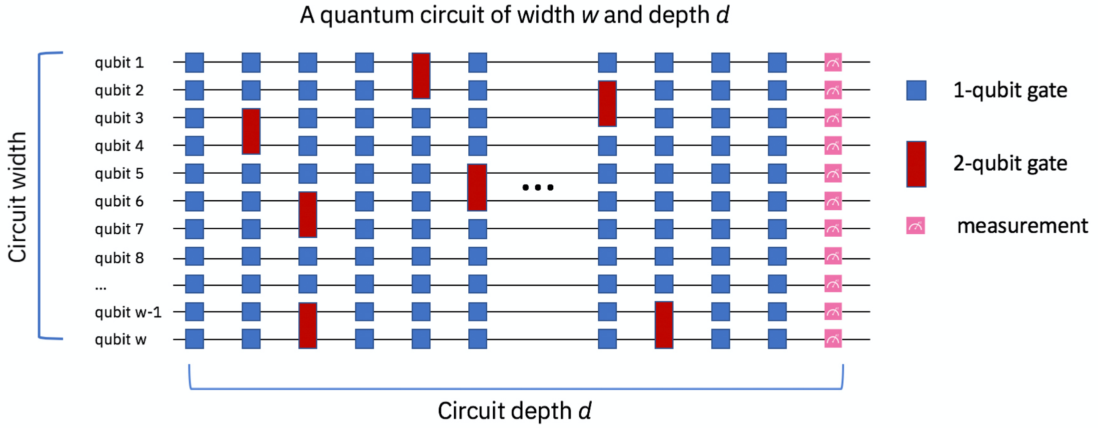
```
q_rot.ry(theta,0)
```

```
q_rot.rz(theta,0)
```

Reference: https://quantum-computing.ibm.com/docs/circ-comp/q-gates

Quantum Circuit





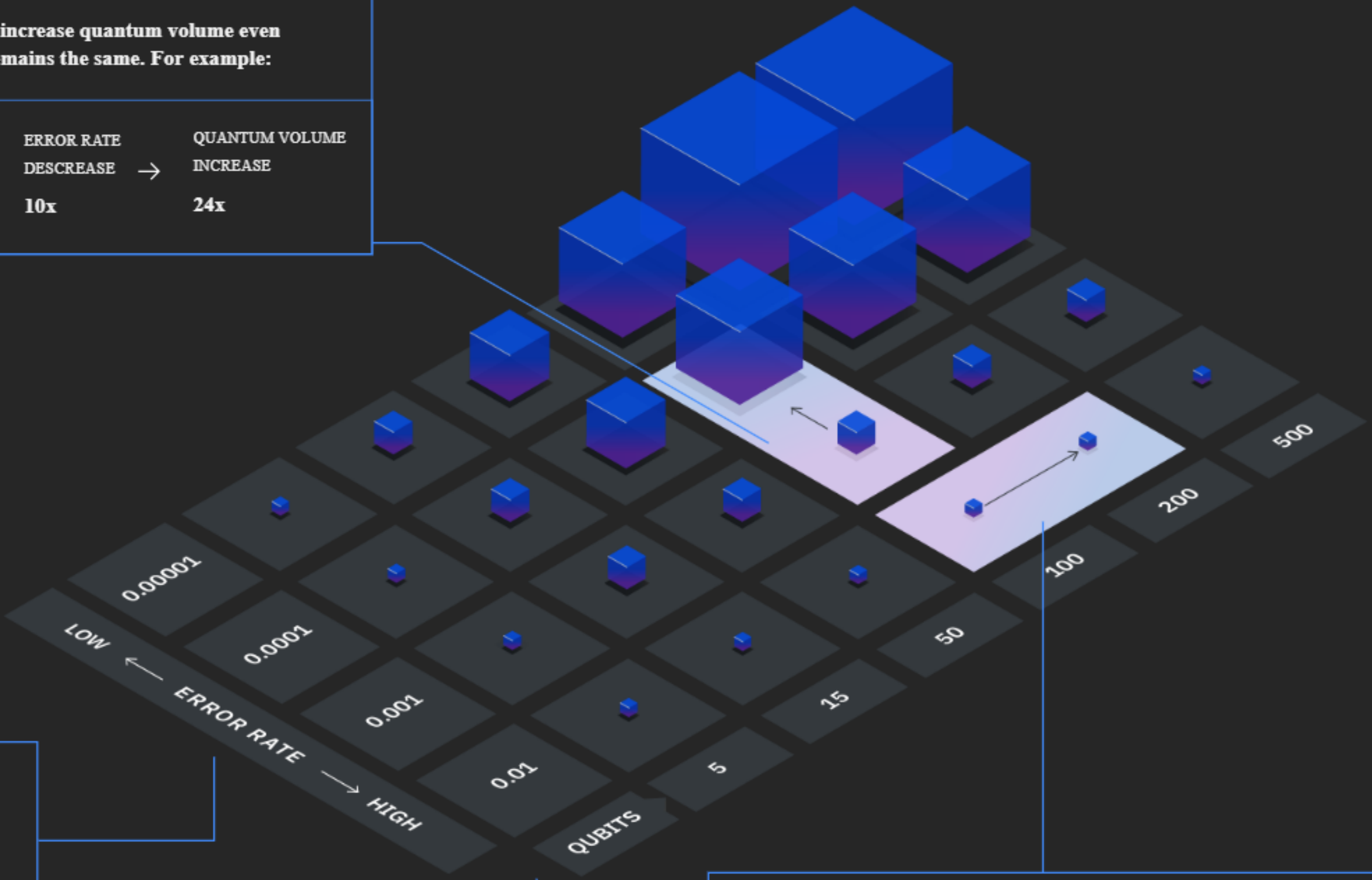
→Broad and profound

Source: https://qiskit.org/textbook/ch-states/introduction.html


iPhone 6
Bigger than bigger
Learn more ›
Watch the film
Experience the keynote
9:41
Tuesday, September 9
9:41
Tuesday, September 9

Improving the error rate can increase quantum volume even when the number of qubits remains the same. For example:

QUBITS EXISTING	→	QUBITS ADDED	→	ERROR RATE DESCREASE	→	QUANTUM VOLUME INCREASE
100 Q		**0 Q**		**10x**		**24x**



Error Rate (Y-Axis)
An expression of how well the device can implement operations between any two qubits

Qubits (X-Axis)
The number of qubits active in the system

If the error rate is high, merely adding qubits will not increase quantum volume.

QUBITS EXISTING	→	QUBITS ADDED	→	ERROR RATE DESCREASE	→	QUANTUM VOLUME INCREASE
100 Q		**100 Q**		**0x**		**0x**

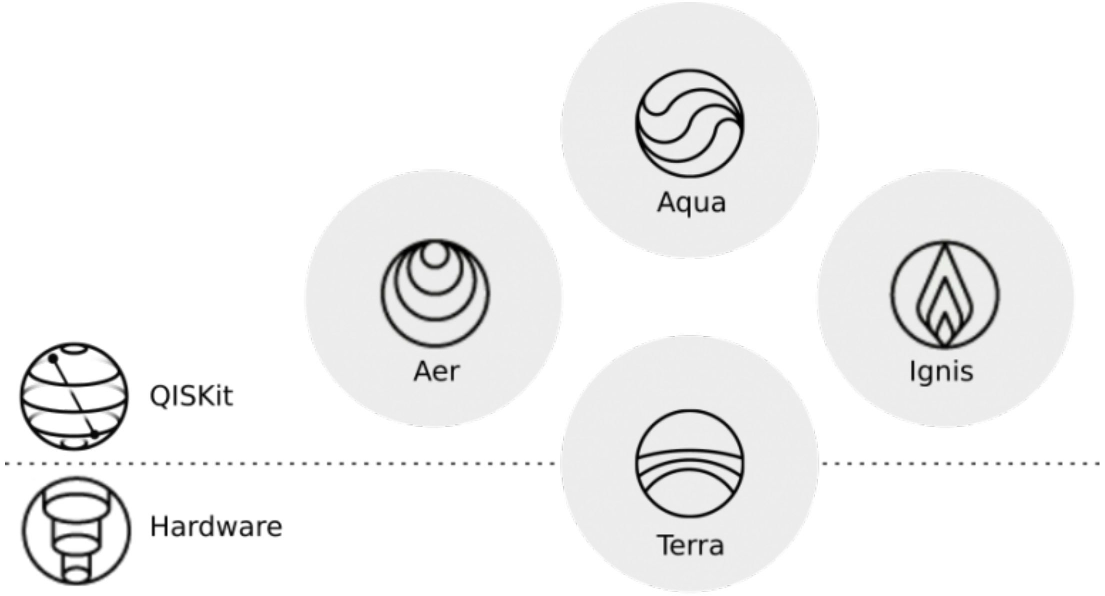
Qiskit Overview

Institution	IBM
First Release	0.1 on March 7, 2017
Open Source	Yes
License	Apache-2.0
HomePage	https://qiskit.org/
Github	https://github.com/Qiskit
Documentation	https://qiskit.org/documentation/
OS	Mac, Windows, Linux
Language	Python
Quantum Language	OpenQASM

Version Information

Qiskit Software	Version
Qiskit	0.17.0
Terra	0.12.0
Aer	0.4.1
Ignis	0.2.0
Aqua	0.6.5
IBM Q Provider	0.6.0

The Qiskit Elements



Terra, the 'earth' element, is the foundation on which the rest of the software lies.

Aer, the 'air' element, permeates all Qiskit elements. For accelerating development via simulators, emulators and debuggers

Aqua, the 'water' element, is the element of life. For building algorithms and applications.

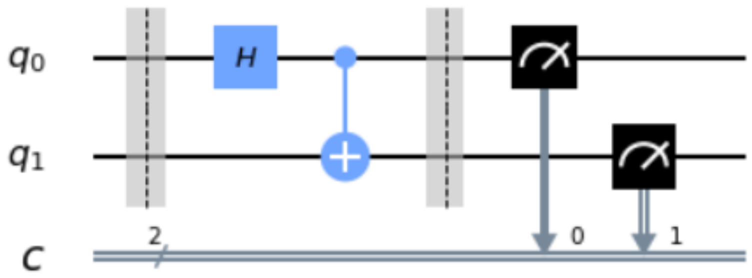
Ignis, the 'fire' element, is dedicated to fighting noise and errors and to forging a new path

Source: https://qiskit.org/documentation/the_elements.html

Qiskit Code Example

In [1]:
```
from qiskit import QuantumCircuit

q_bell = QuantumCircuit(2, 2)
q_bell.barrier()
q_bell.h(0)
q_bell.cx(0, 1)
q_bell.barrier()
q_bell.measure([0, 1], [0, 1])


q_bell.draw(output='mpl',plot_barriers=True)
```

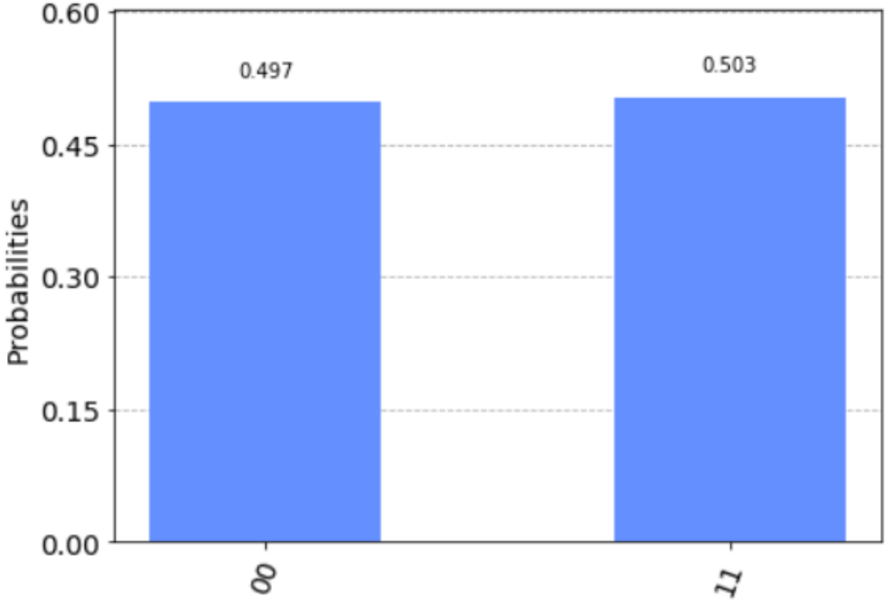
Out[1]:



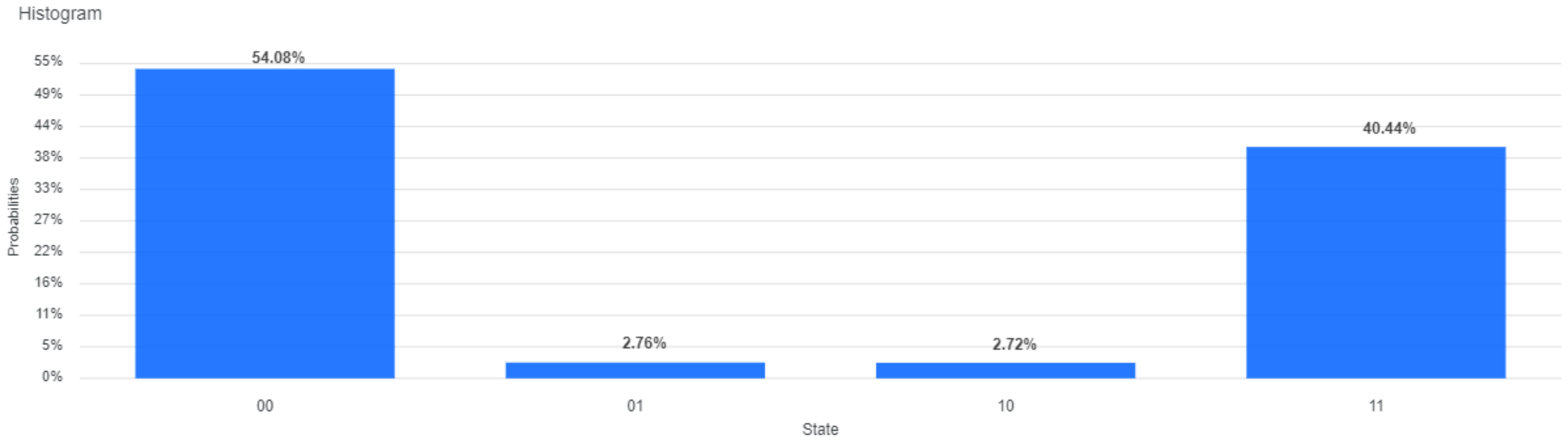
In [2]:
```
from qiskit import Aer, execute
from qiskit.visualization import plot_histogram
backend = Aer.get_backend('qasm_simulator')
job_sim = execute(q_bell, backend,shots=100000)
sim_result = job_sim.result()

print(sim_result.get_counts(q_bell))
plot_histogram(sim_result.get_counts(q_bell))
```

```
{'11': 50254, '00': 49746}
```

Out[2]:



Real QC


Histogram
55%
49%
44%
38%
33%
27%
22%
16%
11%
5%
0%
Probabilities
54.08%
2.76%
2.72%
40.44%
00
01
10
11
State

QA times

Quantum Algorithms 101

The Deutsch Algorithm

- The first to demonstrate quantum over classical computing.

Deutsch Problem:

Given a black box that implement some Boolean function $f: \{0,1\} \to \{0,1\}$.
We are promised that the function is either constant or balanced.

x	f_0	f_1	f_x	$f_{\bar{x}}$
0	0	1	0	1
1	0	1	1	0

constant

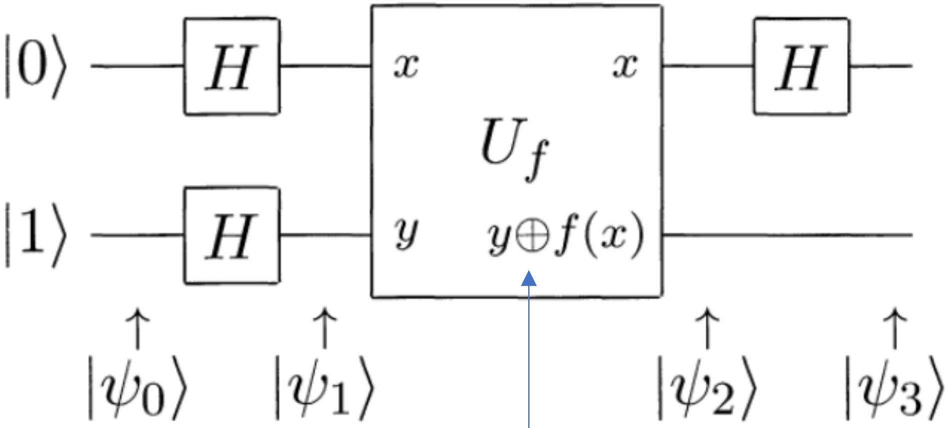
balanced

```
if f(0) = 0:
    if f(1) = 0:
        print("Constant")
    else:
        print("Balanced")
else:
    if f(1) = 0:
        print("Balanced")
    else:
        print("Constant")
```

require two function evaluations to figure out the answer.

The Deutsch Algorithm

- We need only one query on a quantum computer !



Addition module 2

$$|\psi_0\rangle = |0\rangle \otimes |1\rangle$$

$$|\psi_1\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

$$|\psi_2\rangle = \frac{1}{\sqrt{2}}(|0\rangle+|1\rangle)\otimes\frac{1}{\sqrt{2}}(|0 \oplus f(x)\rangle-|1 \oplus f(x)\rangle)$$

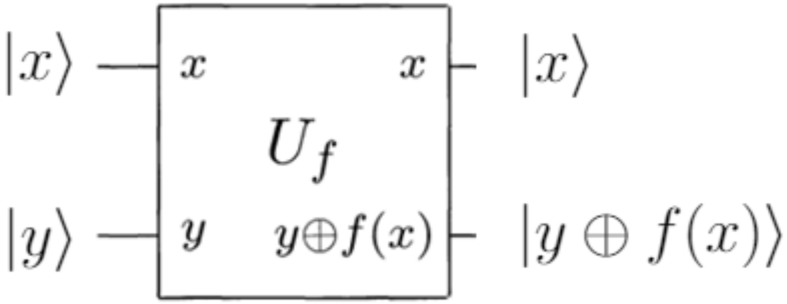
f is Boolean function.

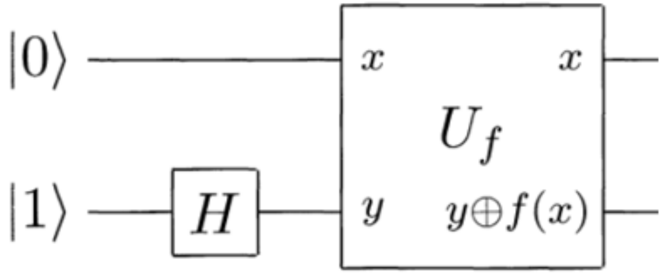
$$|\psi_2\rangle = \frac{1}{\sqrt{2}}(|0\rangle+|1\rangle)\otimes\frac{1}{\sqrt{2}}\left(|f(x)\rangle-\left|\overline{f(x)}\right\rangle\right)$$

Phase Kick Back

- Useful trick in many quantum algorithm.

Consider Quantum Black box function f





$$U_f\left(|x\rangle \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)\right) = |x\rangle \otimes \frac{1}{\sqrt{2}}(|0 \oplus f(x)\rangle - |1 \oplus f(x)\rangle)$$

$$f(x)=0: \quad \frac{|0 \oplus f(x)\rangle - |1 \oplus f(x)\rangle}{\sqrt{2}} = \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

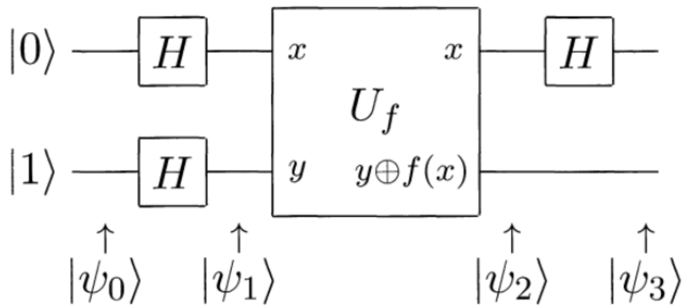
$$f(x)=1: \quad \frac{|0 \oplus f(x)\rangle - |1 \oplus f(x)\rangle}{\sqrt{2}} = \frac{|1\rangle - |0\rangle}{\sqrt{2}} = -\left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right)$$

$$\frac{|0 \oplus f(x)\rangle - |1 \oplus f(x)\rangle}{\sqrt{2}} = (-1)^{f(x)}\left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right)$$

$$U_f\left(|x\rangle \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)\right) = (-1)^{f(x)}\left(|x\rangle \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)\right)$$

$$U_f\left((\alpha_0|0\rangle + \alpha_1|1\rangle) \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)\right) = \left((-1)^{f(0)}\alpha_0|0\rangle + (-1)^{f(1)}\alpha_1|1\rangle\right) \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

The Deutsch Algorithm



$$|\psi_0\rangle = |0\rangle \otimes |1\rangle$$

$$|\psi_1\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

$$|\psi_2\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes \frac{1}{\sqrt{2}}\left(|f(x)\rangle - \left|\overline{f(x)}\right\rangle\right)$$

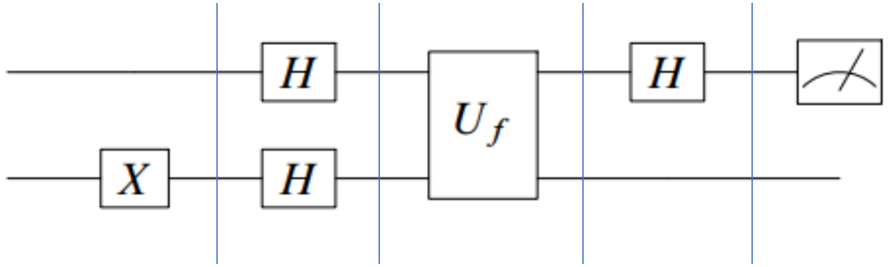
$$U_f\left((\alpha_0|0\rangle + \alpha_1|1\rangle) \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)\right) = \left((-1)^{f(0)}\alpha_0|0\rangle + (-1)^{f(1)}\alpha_1|1\rangle\right) \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

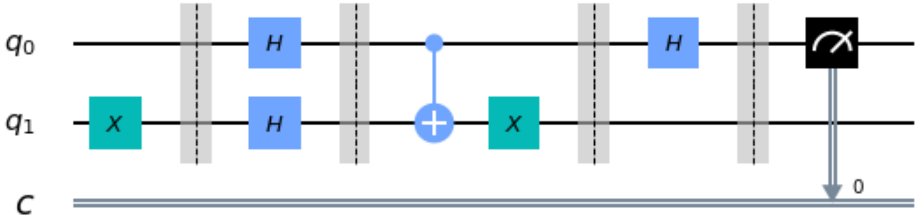
$$\begin{aligned}
|\psi_2\rangle &= \frac{(-1)^{f(0)}}{\sqrt{2}}|0\rangle\left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right) + \frac{(-1)^{f(1)}}{\sqrt{2}}|1\rangle\left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right) \\
&= \left(\frac{(-1)^{f(0)}|0\rangle + (-1)^{f(1)}|1\rangle}{\sqrt{2}}\right)\left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right) \\
&= (-1)^{f(0)}\left(\frac{|0\rangle + (-1)^{f(0)\oplus f(1)}|1\rangle}{\sqrt{2}}\right)\left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right)
\end{aligned}$$

$$|\psi_2\rangle = \begin{cases} \pm\left[\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right]\left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right] & \text{if } f(0) = f(1) \\ \pm\left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right]\left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right] & \text{if } f(0) \neq f(1). \end{cases}$$

$$|\psi_3\rangle = \begin{cases} \pm|0\rangle\left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right] & \text{if } f(0) = f(1) \\ \pm|1\rangle\left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right] & \text{if } f(0) \neq f(1). \end{cases}$$

Qiskit Example



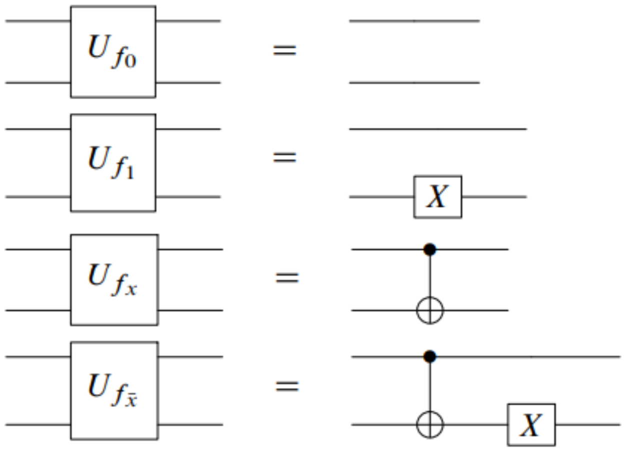


x	f_0	f_1	f_x	$f_{\bar{x}}$
0	0	1	0	1
1	0	1	1	0

constant
output = 0

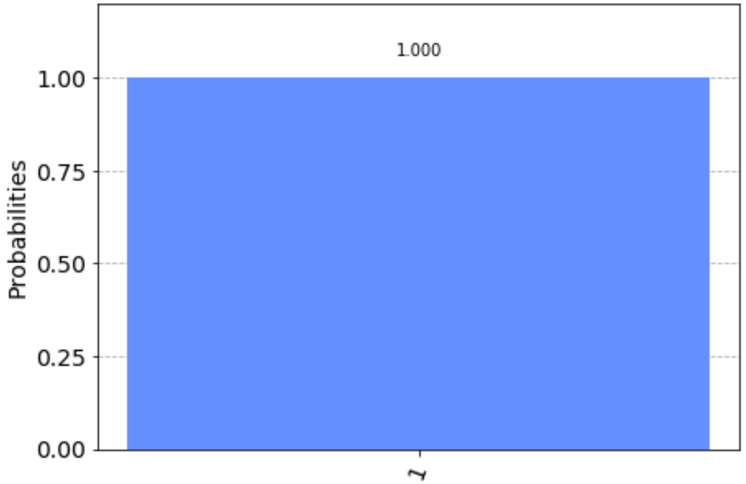
balanced
output = 1

$f_0 \to I$
$f_1 \to -I$
$f_x \to Z$
$f_{\bar{x}} \to -Z$




```python
from qiskit import QuantumCircuit
q_demo=QuantumCircuit(2,1)

q_demo.x(1)
q_demo.barrier()    # psi0
q_demo.h([0,1])
q_demo.barrier()    # psi1
q_demo.cx(0,1)
q_demo.x(1)
q_demo.barrier()    # psi2
q_demo.h(0)
q_demo.barrier()    # psi3
q_demo.measure([0],[0])

q_demo.draw(output='mpl')      # draw your circuit

from qiskit import Aer, execute
from qiskit.visualization import plot_histogram
backend = Aer.get_backend('qasm_simulator')
demo_circ = execute(q_demo, backend,memory=False)
result = demo_circ.result()

#memory = result.get_memory(q_demo)
print(result.get_counts(q_demo))
plot_histogram(result.get_counts(q_demo))
```

```
{'1': 1024}
```



Extension : The Deutsch-Jozsa algorithm

- This time the function f is a function from n bits string to a bit.

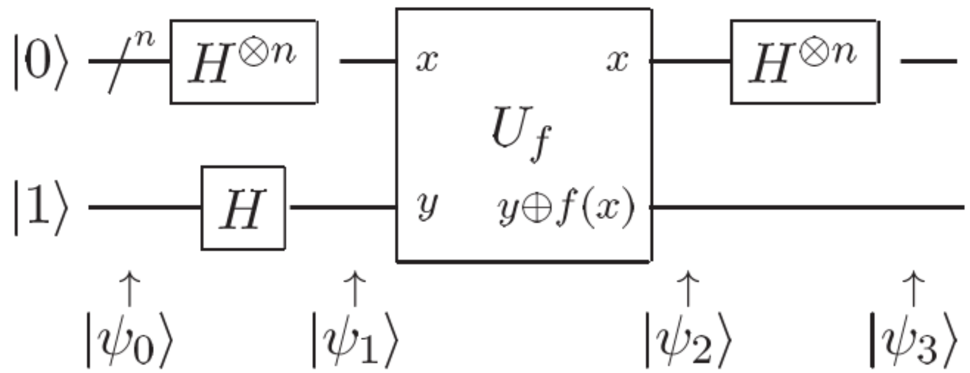
The Deutsch–Jozsa Problem

Input: A black-box for computing an unknown function $f: \{0,1\}^n \to \{0,1\}$.
Promise: f is either a constant or a balanced function.
Problem: Determine whether f is constant or balanced by making queries to f.

The Deutsch-Jozsa algorithm

The Deutsch–Jozsa Problem

Input: A black-box for computing an unknown function $f\colon\{0,1\}^n \to \{0,1\}$.
Promise: f is either a constant or a balanced function.
Problem: Determine whether f is constant or balanced by making queries to f.



$$|\psi_0\rangle = |0\rangle^{\otimes n} \otimes |1\rangle$$

$$|\psi_1\rangle = \frac{1}{\sqrt{2^n}} \sum_x |x\rangle \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

$$|\psi_2\rangle = \frac{1}{\sqrt{2^n}} \sum_x (-1)^{f(x)} |x\rangle \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

$$|\psi_3\rangle = \frac{1}{\sqrt{2^n}} \sum_y \sum_x (-1)^{f(x)+x\cdot y} |y\rangle \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

Summary

Algorithm: Deutsch–Jozsa

Inputs: (1) A black box U_f which performs the transformation $|x\rangle|y\rangle \to |x\rangle|y \oplus f(x)\rangle$, for $x \in \{0, \ldots, 2^n - 1\}$ and $f(x) \in \{0, 1\}$. It is promised that $f(x)$ is either *constant* for all values of x, or else $f(x)$ is *balanced*, that is, equal to 1 for exactly half of all the possible x, and 0 for the other half.

Outputs: 0 if and only if f is constant.

Runtime: One evaluation of U_f. Always succeeds.

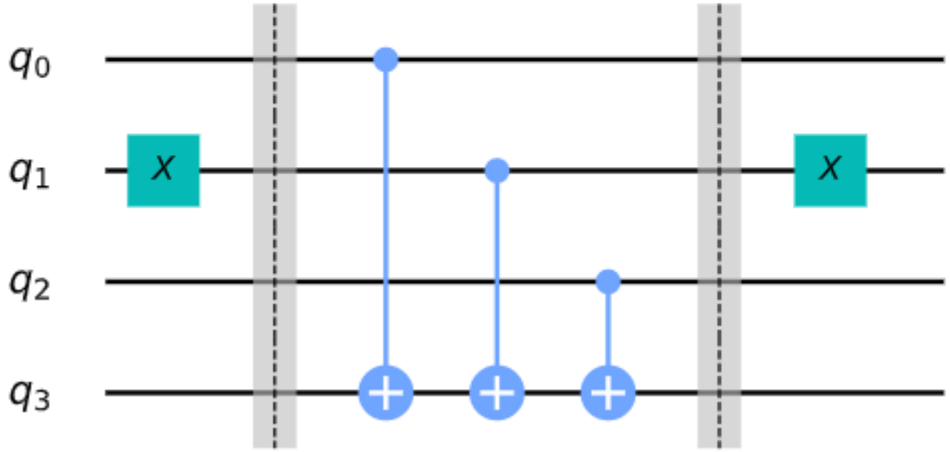
Procedure:

1.	$\|0\rangle^{\otimes n}\|1\rangle$	initialize state
2.	$\to \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} \|x\rangle \left[\frac{\|0\rangle - \|1\rangle}{\sqrt{2}}\right]$	create superposition using Hadamard gates
3.	$\to \sum_x (-1)^{f(x)} \|x\rangle \left[\frac{\|0\rangle - \|1\rangle}{\sqrt{2}}\right]$	calculate function f using U_f
4.	$\to \sum_z \sum_x \frac{(-1)^{x \cdot z + f(x)} \|z\rangle}{\sqrt{2^n}} \left[\frac{\|0\rangle - \|1\rangle}{\sqrt{2}}\right]$	perform Hadamard transform
5.	$\to z$	measure to obtain final output z

Qiskit Example

Outputs 0	Outputs 1
001	000
010	011
100	101
111	110

Balanced Oracle



Now, your turn

Improvement?

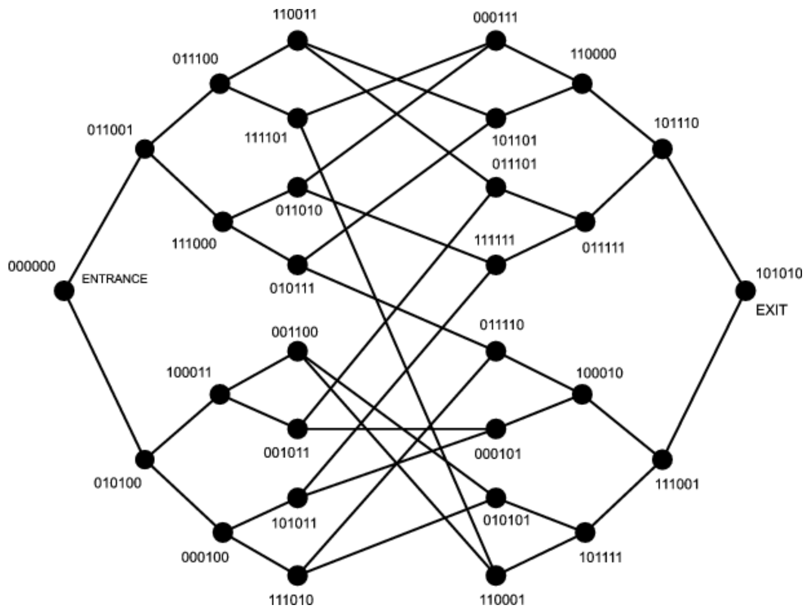
- The worst case in classical take us $\frac{2^n}{2} + 1$
- In quantum we need only one query ….. But, if there exist error ? How about randomized algorithm ?

- That’s why we have Bernstein-Vazirani, Simon’s algorithm.

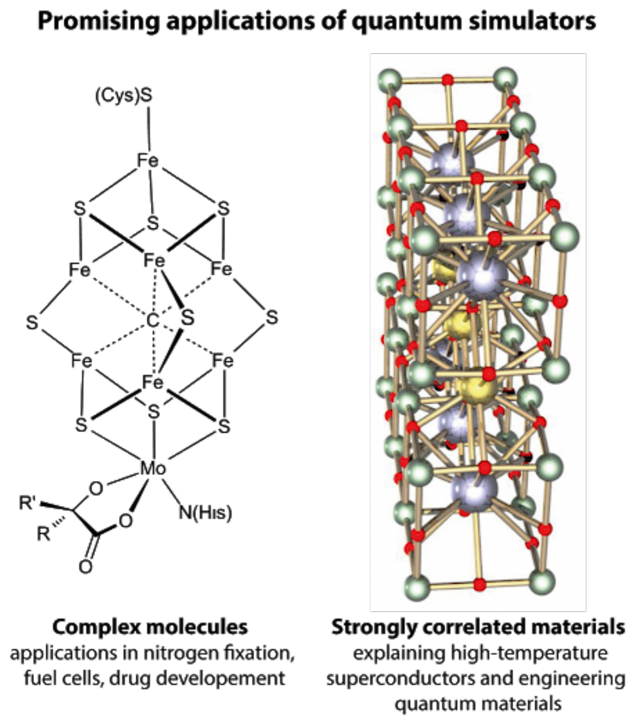
BQP class

- BQP: Bounded-error quantum polynomial time
  - What we focus on for QC

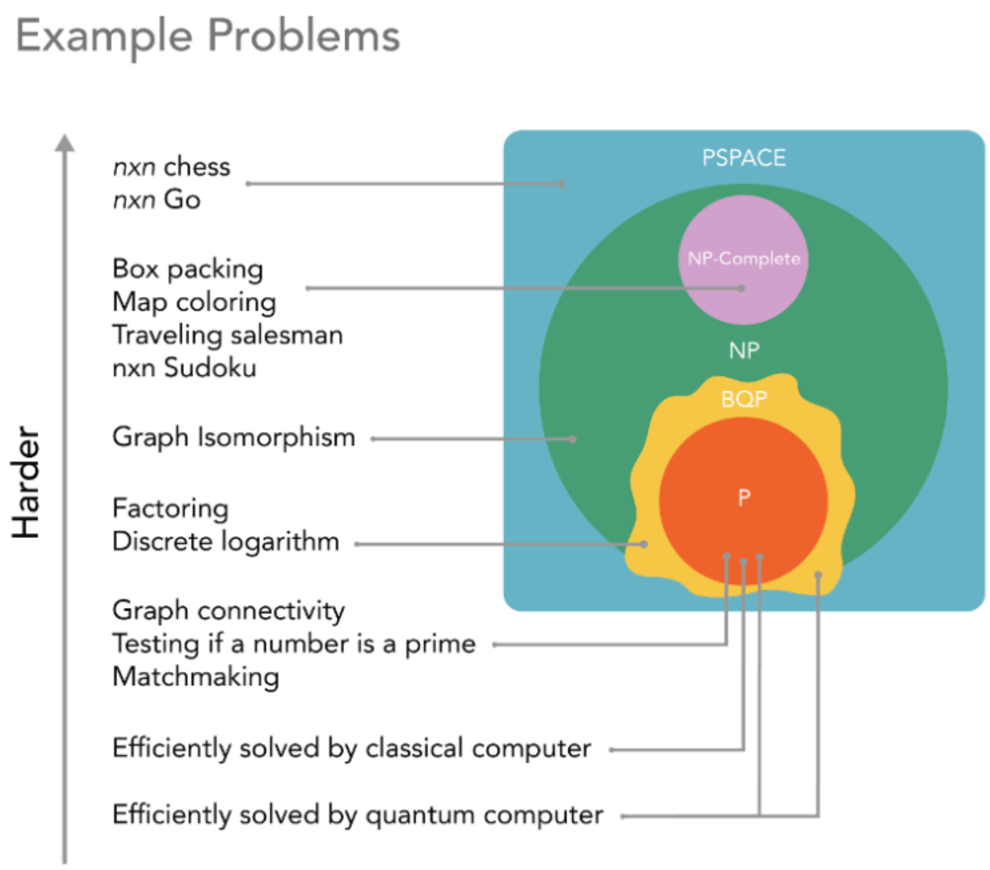
Quantum Walks

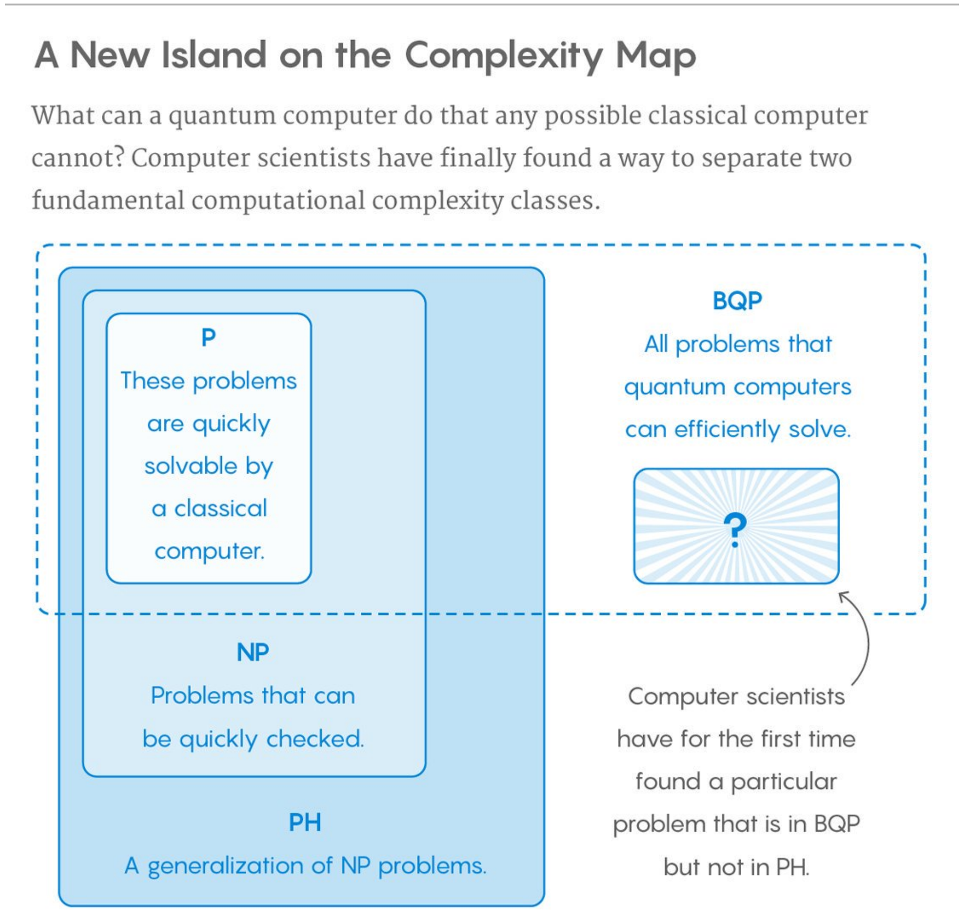


Quantum Simulation



Where Quantum Computers Fits In





https://complexityzoo.uwaterloo.ca/Complexity_Zoo

The Zoo of Quantum Algorithms

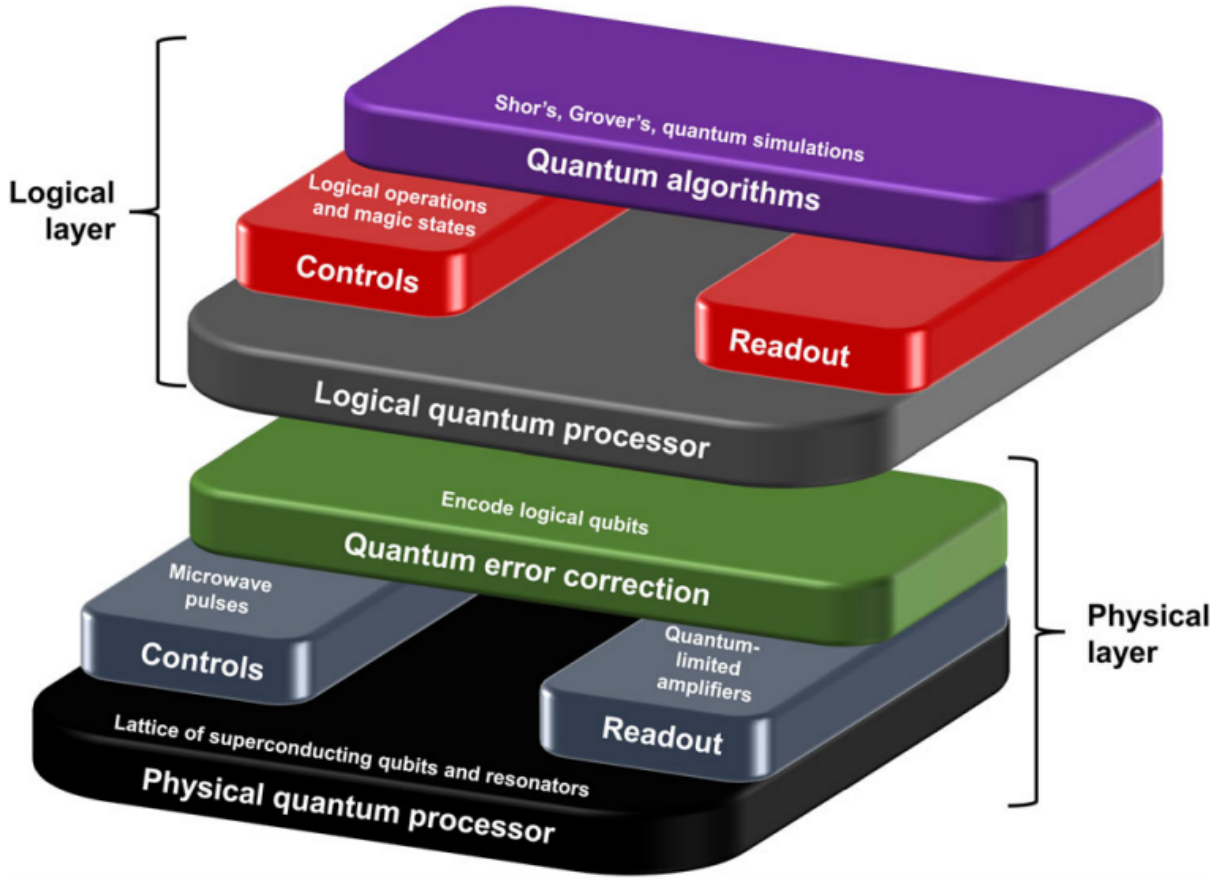
Algebraic and Number Theoretic Algorithms	Oracular Algorithms	Approximation and Simulation Algorithms	Optimization, Numerics, and Machine Learning
• **Algorithm:** Factoring (Super-polynomial) • **Algorithm:** Discrete-log (Super-polynomial) • **Algorithm:** Verifying Matrix Products (Polynomial)	• **Algorithm:** Searching (Polynomial) • **Algorithm:** Hidden Shift (Super-polynomial) • **Algorithm:** Counterfeit Coins (Polynomial)	• **Algorithm:** Quantum Simulation (Super-polynomial) • **Algorithm:** Quantum Approximate Optimization (Super-polynomial)	• **Algorithm:** Machine Learning (Varies) • **Algorithm:** Adiabatic Algorithms (Unknown) • **Algorithm:** Quantum Dynamic Programming (Polynomial)

https://quantumalgorithmzoo.org/

Why have we not shown that?

- All of above QA require a quantum circuit implemented with near zero error rates
  - → Need Quantum Error Correction

Quantum Computing Stack



Source: [Gambetta, J.M., Chow, J.M. & Steffen, M. Building logical qubits in a superconducting quantum computing system. *npj Quantum Inf* **3,** 2 (2017).]

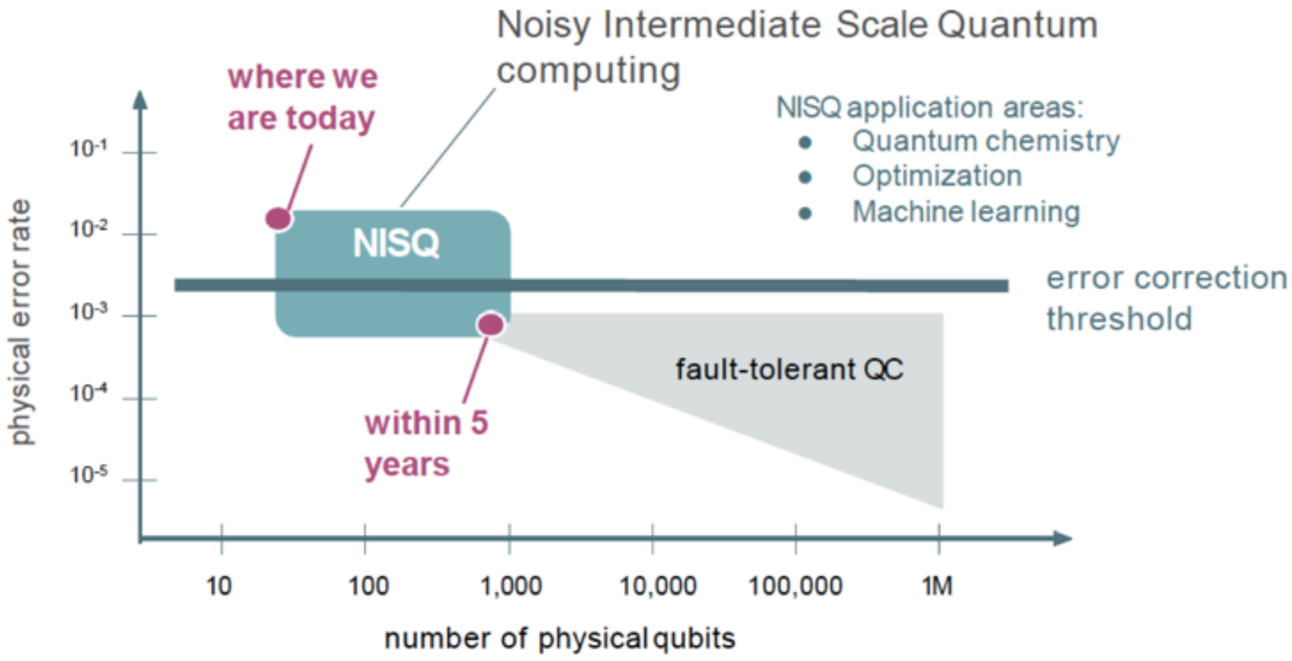
Where are we now?

Resource	Current Capability	Estimate for 2048-bit Shor's factorization*
Operational Fidelity	~99%	99.99%
# of physical gate operations	100s	1.5×10^{21}
# of physical qubits	53	98,000,000
# of FT logical operations (perfect operations)	0	450,000,000,000
# of FT logical qubits (ideal qubits)	0	12000

*N. C. Jones, R. Van Meter, A. G. Fowler, P. L. McMahon, J. Kim, T. D. Ladd, and Y. Yamamoto, Layered Architecture for Quantum Computing, Phys. Rev. X 2, 031007 (2012).

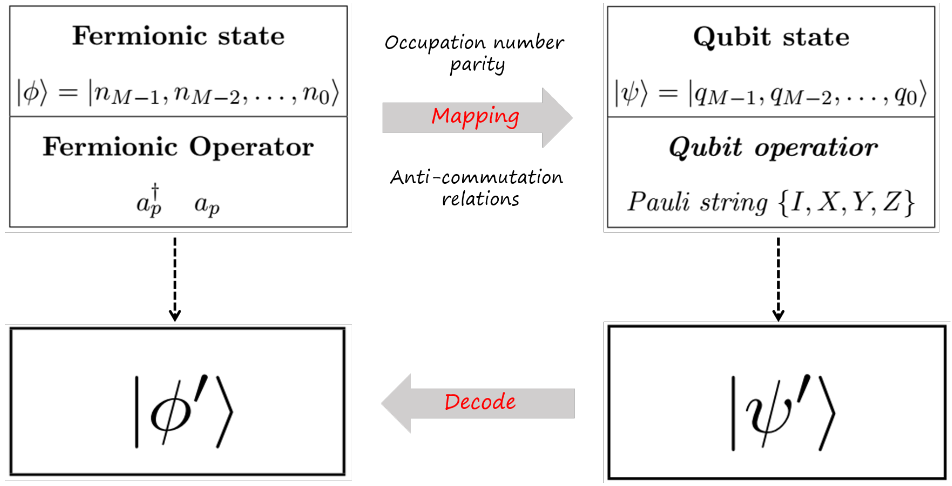
Near term

- Noisy Intermediate-Scale Quantum (NISQ)
  - we don’t expect to be able to execute a circuit that contains many more than about 1000 gates



Application: Quantum meets CS

- When simulating Fermionic Problem



PHYSICAL REVIEW A **95**, 032332 (2017)

Operator locality in the quantum simulation of fermionic models

Vojtěch Havlíček

Institute for Theoretical Physics and Station Q Zurich, ETH Zurich, 8093 Zurich, Switzerland
and Department of Computer Science, University of Oxford, Wolfson Building, Parks Road, Oxford OX1 3QD, United Kingdom

Matthias Troyer

Institute for Theoretical Physics and Station Q Zurich, ETH Zurich, 8093 Zurich, Switzerland
and Quantum Architectures and Computation Group, Microsoft Research, Redmond, Washington 98052, USA

James D. Whitfield

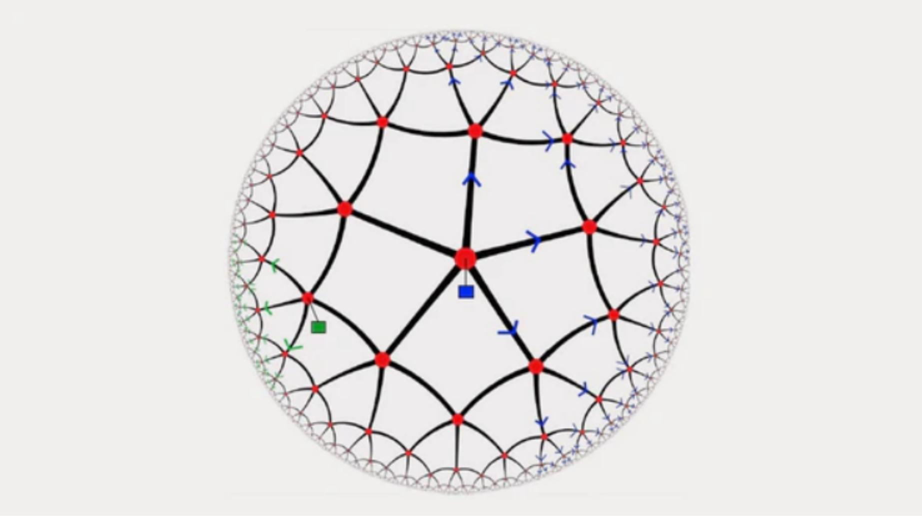
Department of Physics and Astronomy, Dartmouth College, 6127 Wilder Laboratory, Hanover, New Hampshire 03755, USA

(Received 6 February 2017; published 29 March 2017)

Simulating fermionic lattice models with qubits requires mapping fermionic degrees of freedom to qubits. The simplest method for this task, the Jordan-Wigner transformation, yields strings of Pauli operators acting on an extensive number of qubits. This overhead can be a hindrance to implementation of qubit-based quantum simulators, especially in the analog context. Here we thus review and analyze alternative fermion-to-qubit mappings, including the two approaches by Bravyi and Kitaev and the auxiliary fermion transformation. The Bravyi-Kitaev transform is reformulated in terms of a classical data structure and generalized to achieve a further locality improvement for local fermionic models on a rectangular lattice. We conclude that the most compact encoding of the fermionic operators can be done using ancilla qubits with the auxiliary fermion scheme. Without introducing ancillas, a variant of the Bravyi-Kitaev transform provides the most compact fermion-to-qubit mapping for Hubbard-like models.

Application: Doing Physics with Quantum Computers

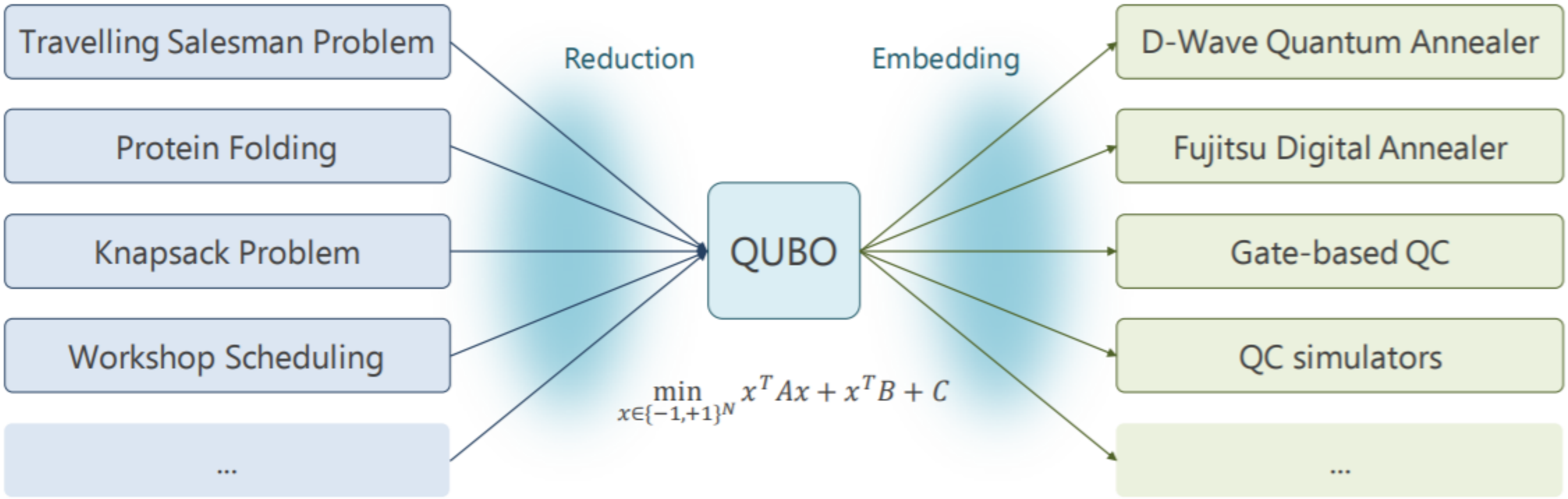
- It from Qubit: Projects



Developments over the past ten years have shown that major advances in our understanding of quantum gravity, quantum field theory and other aspects of fundamental physics can be achieved by bringing to bear insights and techniques from quantum information theory. Nonetheless, fundamental physics and quantum information theory remain distinct disciplines and communities, separated by significant barriers to communication and collaboration. Funded by a grant from the Simons Foundation, "It from Qubit" is a large-scale effort by some of the leading researchers in both communities to foster communication, education and collaboration between them, thereby advancing both fields and ultimately solving some of the deepest problems in physics. The overarching scientific questions motivating the collaboration include:

- Does spacetime emerge from entanglement?
- Do black holes have interiors?
- Does the universe exist outside our horizon?
- What is the information-theoretic structure of quantum field theories?
- Can quantum computers simulate all physical phenomena?
- How does quantum information flow in time?

Application: Quantum + NP = NO problems?



QUBO reduction from NP-hard problems