Quantum Computing
The Why and How

Jonathan Baker
Argonne National Lab
7-29-2019

Outline for this Talk (Pt 1.)

1. Why Quantum Computing? What is the current state of quantum computing and what are current challenges? What are algorithms is this paradigm is ideal for?

2. What is different between quantum and classical information? What makes a quantum computer that much different than a classical one?

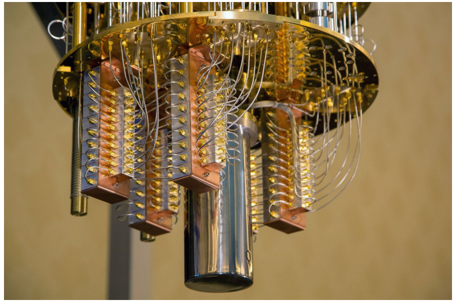
3. How can we use a quantum computer to solve concrete problems? Can we do better than classical computers at the same tasks?

Why Quantum Computing?

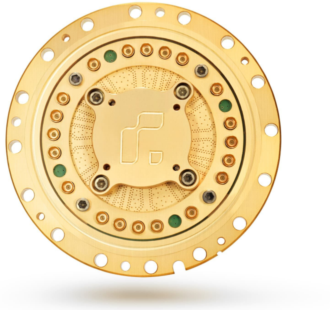
- Fundamentally change what is computable (in a reasonable amount of time)
  - The only known means to potentially scale computation exponentially with the number of devices
  - We can do this by taking advantage of quantum mechanical phenomenon
- Solve currently intractable problems in chemistry, simulation, and optimization
- Moore's Law is ending - quantum computing can act as a replacement in some scientific domains to help continue scaling applications
- Insights in classical computing
  - Many classical algorithms are "quantum-inspired", e.g. in chemistry physics or cryptography
  - Challenges classical algorithms to compete with quantum algorithms

Current State of Quantum Computing: NISQ

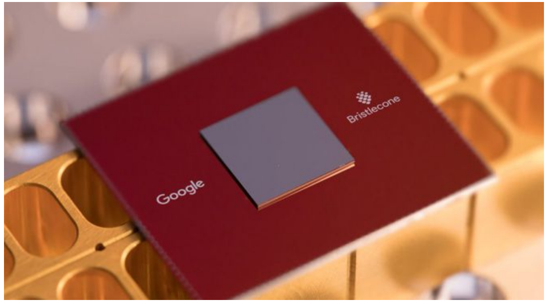
- Noisy-Intermediate Scale Quantum
  - 10s to 100s of qubits
  - Moderate error rates
  - Limited connectivity
  - No error correction



IBM
50 Superconducting Qubits

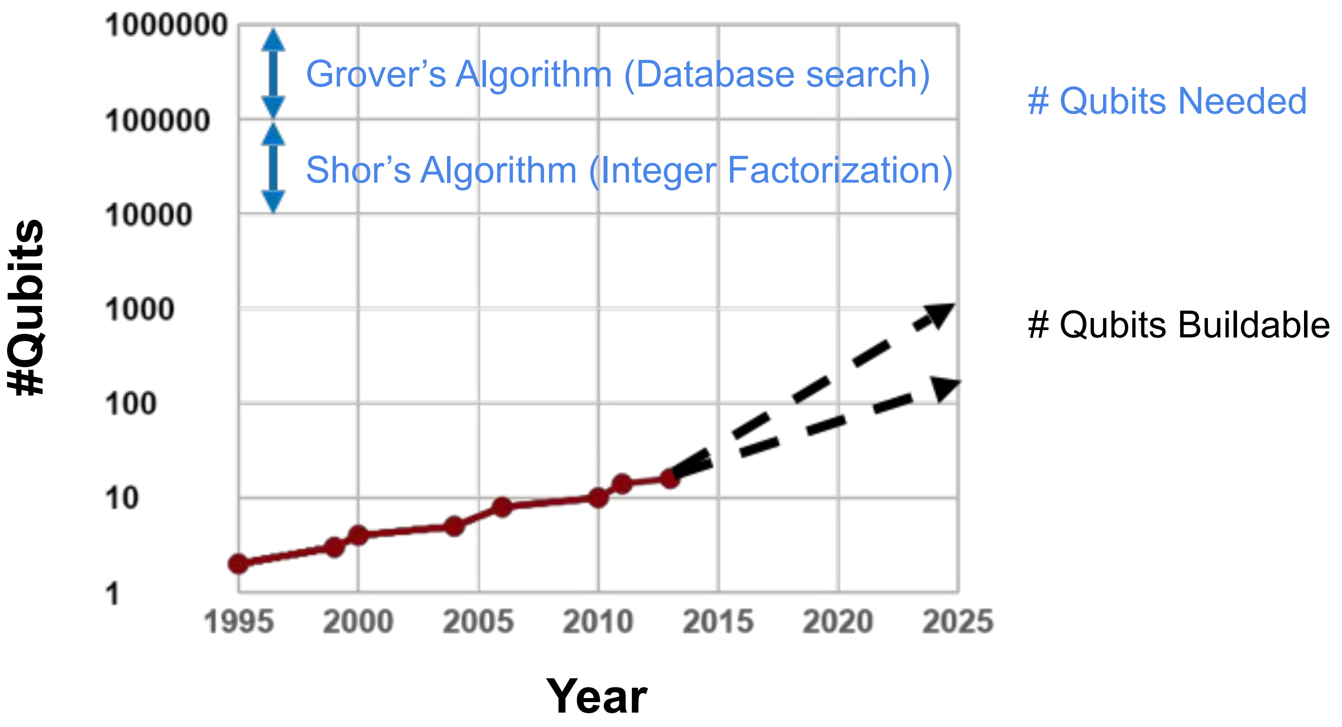


Rigetti
20 Superconducting Qubits

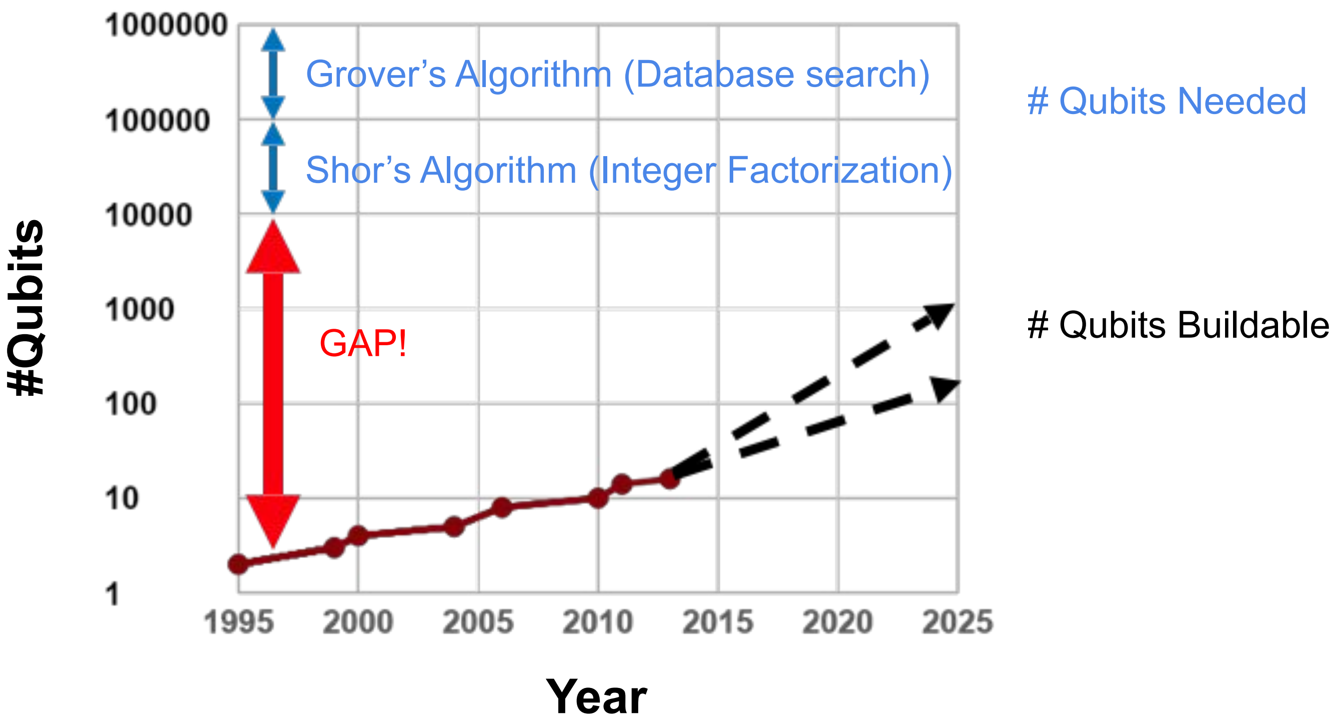


Google
72 superconducting qubits

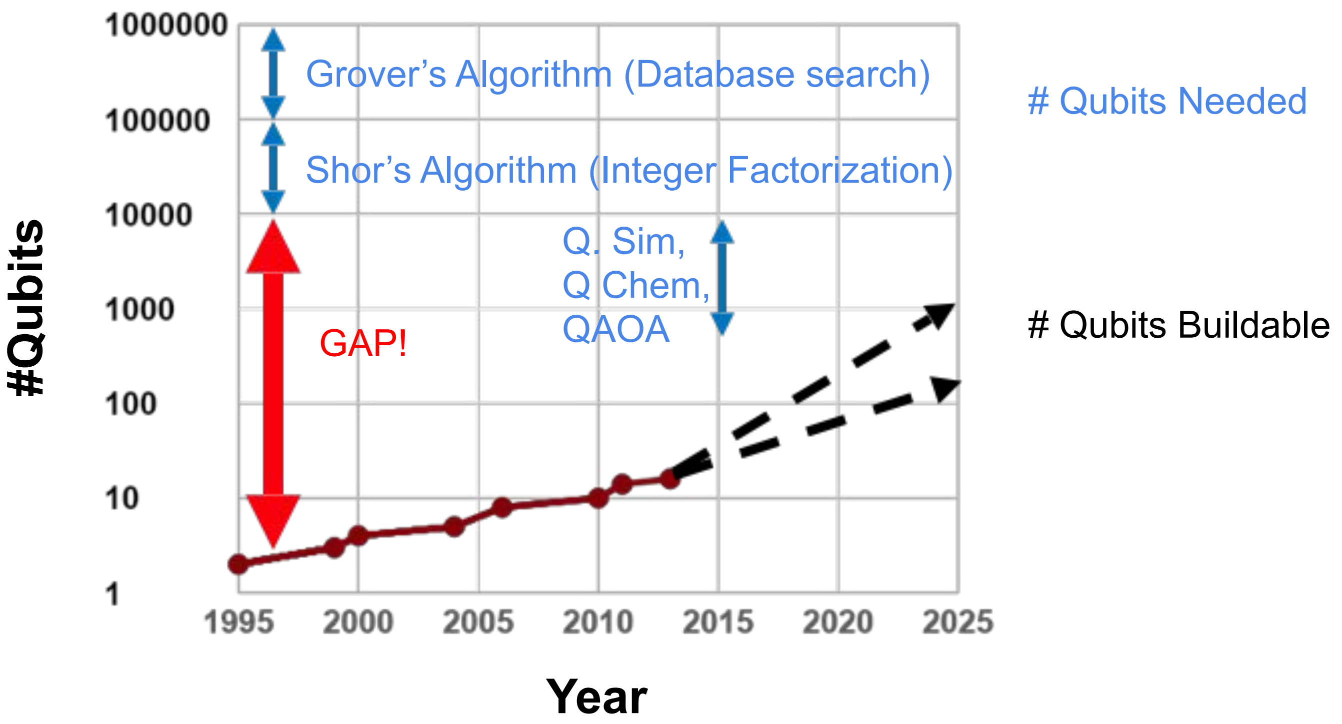
The Algorithms to Machines Gap



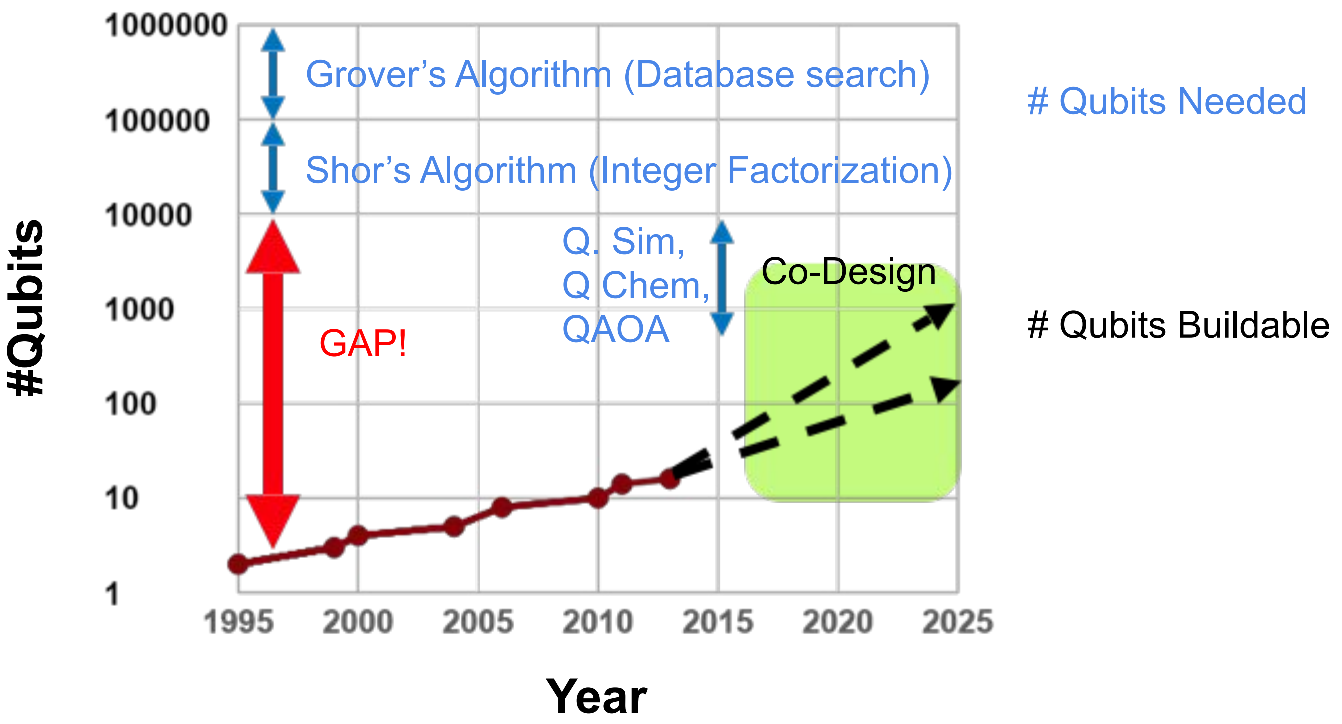
The Algorithms to Machines Gap



The Algorithms to Machines Gap

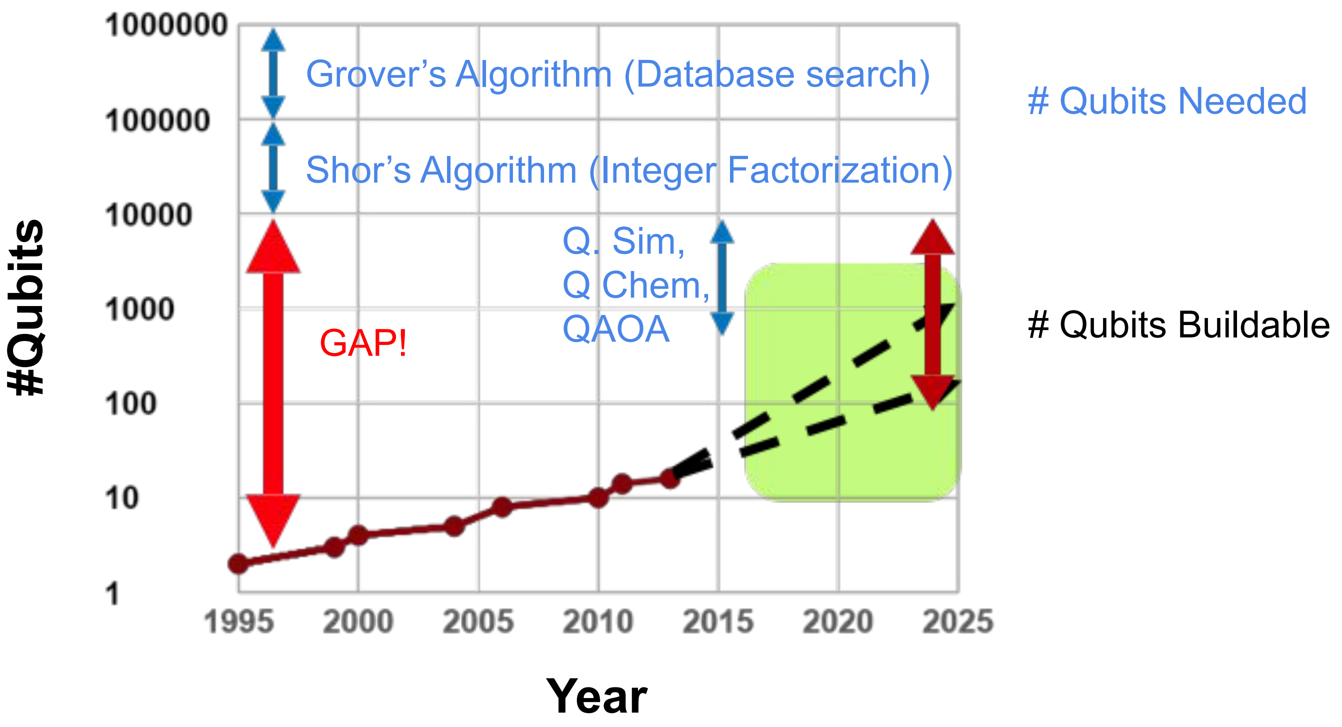


Closing the Gap: Software-Enabled Vertical Integration and Co-Design



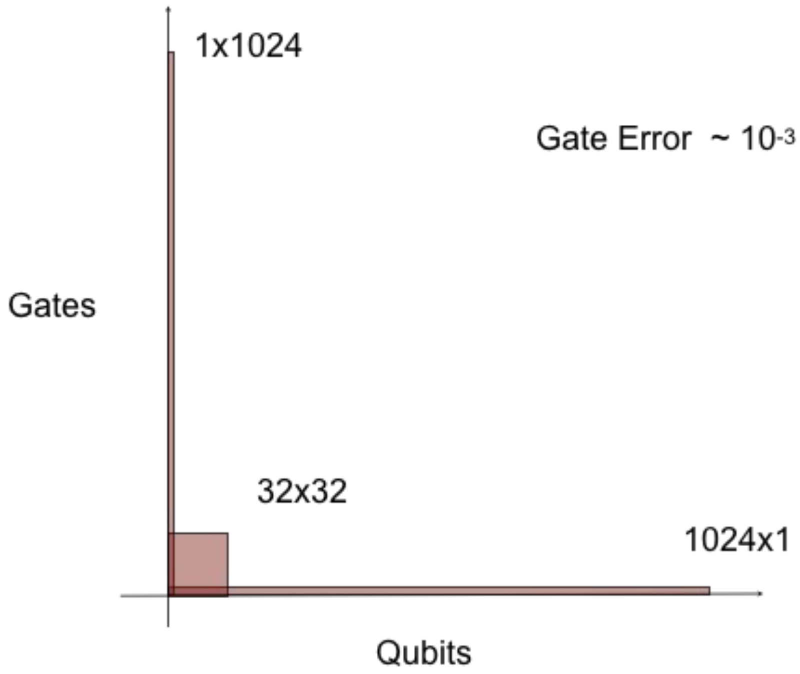
Result: Crossover by 2023!

Develop co-designed algorithms, SW, and HW to close the gap between algorithms and devices by 100-1000X, accelerating QC by 10-20 years.



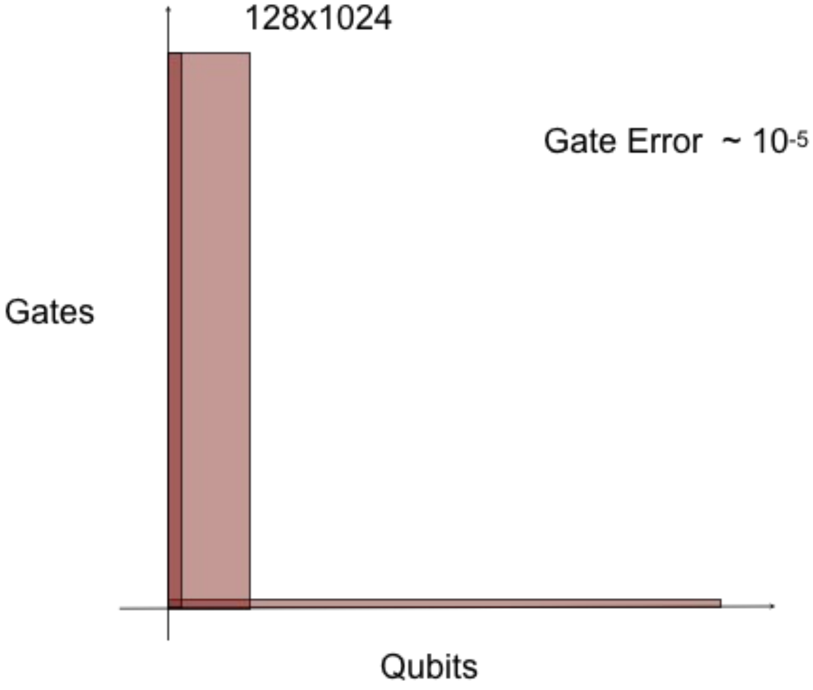
Space-Time Product Limits

Error rates of quantum operations limit what we can accomplish



Space-Time Product Limits

Error rates of quantum operations limit what we can accomplish

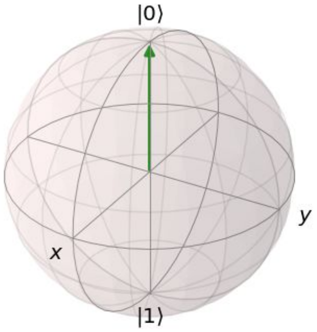


“Good” Quantum Algorithms

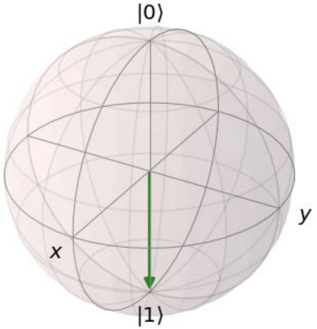
- Compact problem representation
  - Functions, small molecules, small graphs
- High complexity computation
- Compact solution
- Easily-verifiable solution
- Co-processing with classical supercomputers
- Can exploit a small number of quantum kernels

Introduction to the Basics

One Qubit

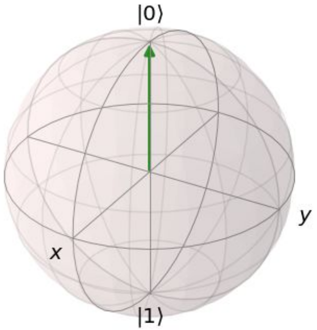


$$|0\rangle$$

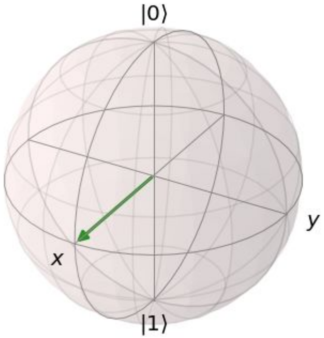


$$|1\rangle$$

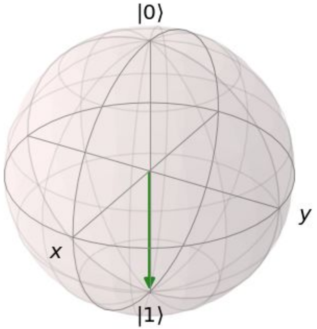
One Qubit



$$|0\rangle$$

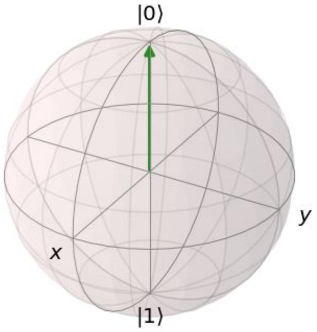


$$|0\rangle + |1\rangle$$

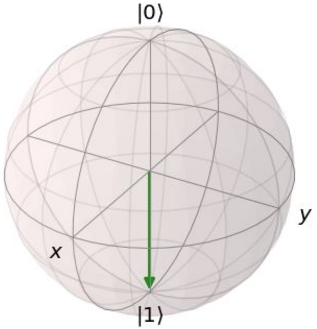


$$|1\rangle$$

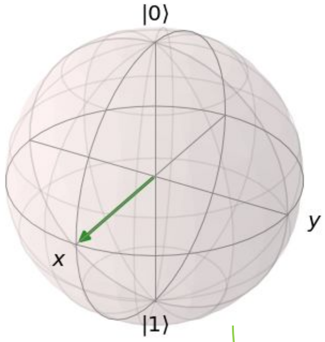
One Qubit



$$|0\rangle$$



$$|1\rangle$$



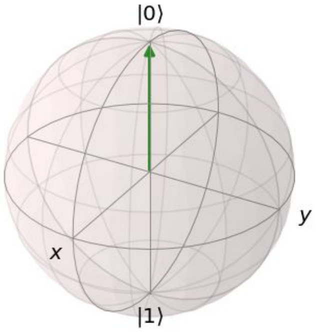
$$|0\rangle + |1\rangle$$

when we try to "see" the state, we can only see 0 or 1.
50% of the time we see 0.
50% of the time we see 1.

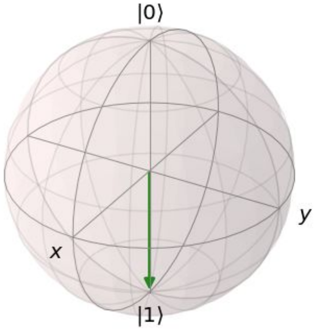
One Qubit

Identically prepared qubits can still behave randomly!

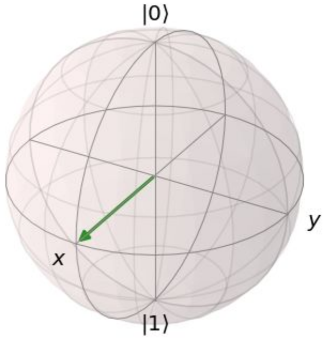
This randomness is inherent in nature, and not a limitation of our observation.



$$|0\rangle$$

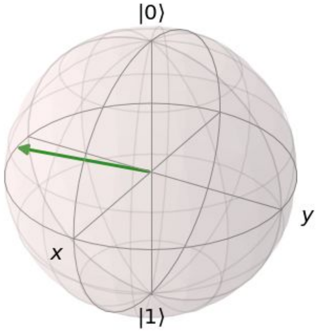


$$|1\rangle$$

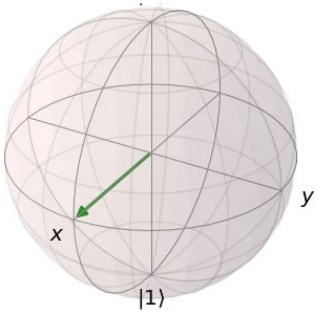


$$|0\rangle + |1\rangle$$

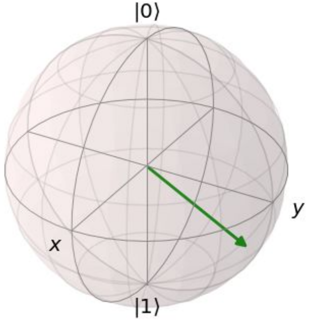
Multiple Qubits



$$(1 + j)\,|0\rangle + |1\rangle$$



$$|0\rangle + |1\rangle$$



$$(-2 + j)\,|0\rangle + (-5j)\,|1\rangle$$

The Power of Quantum Information

Why simulating quantum systems becomes intractable quickly

$$a\,|000\rangle + b\,|001\rangle + c\,|010\rangle + d\,|011\rangle + e\,|100\rangle + f\,|101\rangle + g\,|110\rangle + h\,|111\rangle$$

The Power of Quantum Information

Why simulating quantum systems becomes intractable quickly

$$a\,|000\rangle + b\,|001\rangle + c\,|010\rangle + d\,|011\rangle + e\,|100\rangle + f\,|101\rangle + g\,|110\rangle + h\,|111\rangle$$

- The state of n qubits is described by 2^n coefficients.
- Adding one qubit **doubles** the dimension.
- This is known as **superposition**.

$$\begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \\ g \\ h \end{bmatrix}$$

The Power of Quantum Information

Why simulating quantum systems becomes intractable quickly

classical
probabilistic
bit

The Power of Quantum Information

Why simulating quantum systems becomes intractable quickly

$$(a0 + b1)$$

classical
probabilistic
bit

The Power of Quantum Information

Why simulating quantum systems becomes intractable quickly

$$(a0 + b1)$$

$$(a0 + b1).(c0 + d1)$$

classical
probabilistic
bit

The Power of Quantum Information

Why simulating quantum systems becomes intractable quickly

$$(a0 + b1)$$

$$(a0 + b1).(c0 + d1)$$

$$(a0 + b1).(c0 + d1).(e0 + f1)$$

classical
probabilistic
bit

The Power of Quantum Information

Why simulating quantum systems becomes intractable quickly

Quantum bit
(qubit)

The Power of Quantum Information

Why simulating quantum systems becomes intractable quickly

$$a\,|0\rangle + b\,|1\rangle$$

Quantum bit
(qubit)

The Power of Quantum Information

Why simulating quantum systems becomes intractable quickly

$$a\left|0\right\rangle + b\left|1\right\rangle$$

Quantum bit (qubit)

$$a\left|00\right\rangle + b\left|01\right\rangle + c\left|10\right\rangle + d\left|11\right\rangle$$

The Power of Quantum Information

Why simulating quantum systems becomes intractable quickly

$$a\,|0\rangle + b\,|1\rangle$$

Quantum bit
(qubit)

$$a\,|00\rangle + b\,|01\rangle + c\,|10\rangle + d\,|11\rangle$$

$$a\,|000\rangle + b\,|001\rangle + c\,|010\rangle + d\,|011\rangle + e\,|100\rangle + f\,|101\rangle + g\,|110\rangle + h\,|111\rangle$$

The Power of Quantum Information

Why simulating quantum systems becomes intractable quickly

$$a|0\rangle + b|1\rangle$$

Quantum bit (qubit)

$$a|00\rangle + b|01\rangle + c|10\rangle + d|11\rangle$$

$$a|000\rangle + b|001\rangle + c|010\rangle + d|011\rangle + e|100\rangle + f|101\rangle + g|110\rangle + h|111\rangle$$

The state of an n-qubit system cannot (in general) be written as the state of its individual components.

This is known as ***entanglement****.*

Quantum Information Processing

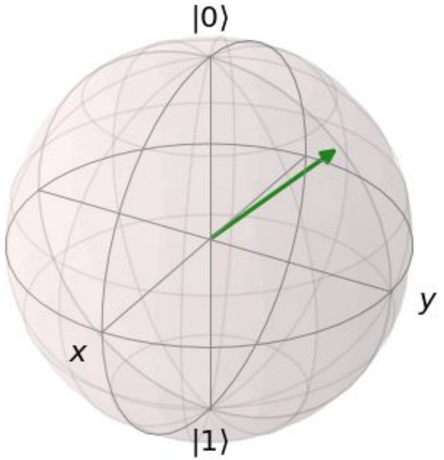
Using Vectors Matrices and Projections

Quantum Information Processing

Using Vectors Matrices and Projections

$$a\,|000\rangle + b\,|001\rangle + c\,|010\rangle + d\,|011\rangle + e\,|100\rangle + f\,|101\rangle + g\,|110\rangle + h\,|111\rangle$$

$$\begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \\ g \\ h \end{bmatrix}$$

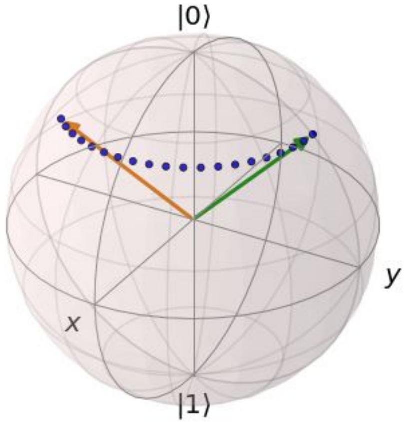


Quantum Information Processing

Using Vectors Matrices and Projections

$$a\,|000\rangle + b\,|001\rangle + c\,|010\rangle + d\,|011\rangle + e\,|100\rangle + f\,|101\rangle + g\,|110\rangle + h\,|111\rangle$$

$$\begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \\ g \\ h \end{bmatrix}$$

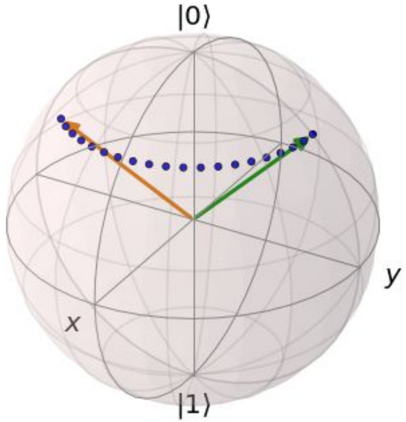


Quantum Information Processing

Using Vectors Matrices and Projections

$$a\,|000\rangle + b\,|001\rangle + c\,|010\rangle + d\,|011\rangle + e\,|100\rangle + f\,|101\rangle + g\,|110\rangle + h\,|111\rangle$$

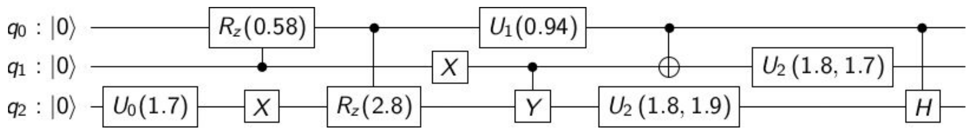
$$\begin{pmatrix} g_{00} & g_{01} & g_{02} & g_{03} & g_{04} & g_{05} & g_{06} & g_{07} \\ g_{10} & g_{11} & g_{12} & g_{13} & g_{14} & g_{15} & g_{16} & g_{17} \\ g_{20} & g_{21} & g_{22} & g_{23} & g_{24} & g_{25} & g_{26} & g_{27} \\ g_{30} & g_{31} & g_{32} & g_{33} & g_{34} & g_{35} & g_{36} & g_{37} \\ g_{40} & g_{41} & g_{42} & g_{43} & g_{44} & g_{45} & g_{46} & g_{47} \\ g_{50} & g_{51} & g_{52} & g_{53} & g_{54} & g_{55} & g_{56} & g_{57} \\ g_{60} & g_{61} & g_{62} & g_{63} & g_{64} & g_{65} & g_{66} & g_{67} \\ g_{70} & g_{71} & g_{72} & g_{73} & g_{74} & g_{75} & g_{76} & g_{77} \end{pmatrix} \begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \\ g \\ h \end{bmatrix}$$



Quantum Information Processing

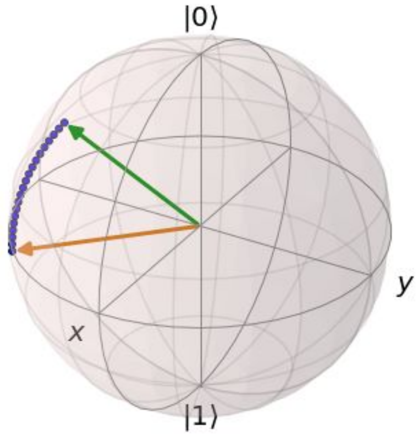
Using Vectors Matrices and Projections

$$a\,|000\rangle + b\,|001\rangle + c\,|010\rangle + d\,|011\rangle + e\,|100\rangle + f\,|101\rangle + g\,|110\rangle + h\,|111\rangle$$



Quantum Circuit Model

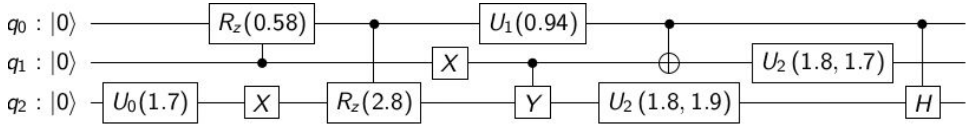
$$\begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \\ g \\ h \end{bmatrix}$$



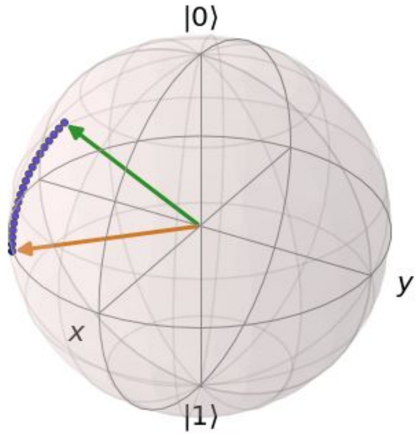
Quantum Information Processing

Using Vectors Matrices and Projections

$$a\,|000\rangle + b\,|001\rangle + c\,|010\rangle + d\,|011\rangle + e\,|100\rangle + f\,|101\rangle + g\,|110\rangle + h\,|111\rangle$$



$$\begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \\ g \\ h \end{bmatrix}$$



Important Result:
1-qubit & 2-qubit gates (i.e. local operations) are sufficient for universal computation [Barenco et al. 95].

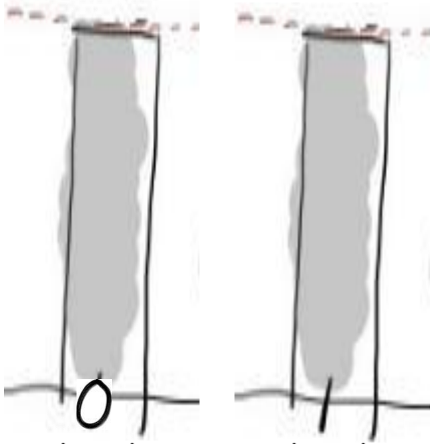
Example Quantum Gates: The Hadamard Gate

Example Quantum Gates: The Hadamard Gate

H Creates Superposition!

Example Quantum Gates: The Hadamard Gate

H Creates Superposition!

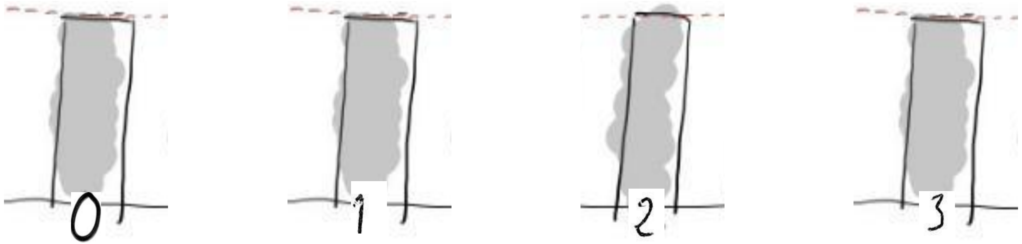


$$|0\rangle + |1\rangle$$

Example Quantum Gates: The Hadamard Gate

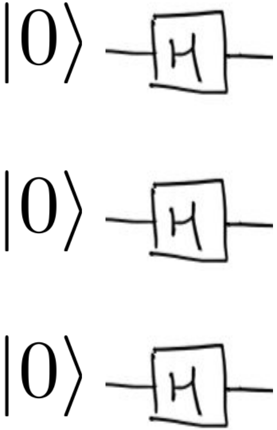
$$|0\rangle \; -\boxed{H}-$$

$$|0\rangle \; -\boxed{H}-$$



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

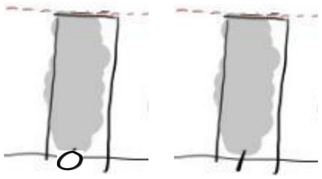
Example Quantum Gates: The Hadamard Gate



$$|000\rangle + |001\rangle + |010\rangle + |011\rangle + |100\rangle + |101\rangle + |110\rangle + |111\rangle$$

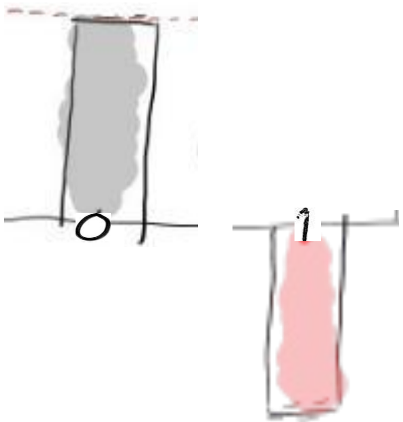
Example Quantum Gates: The Hadamard Gate

$$|0\rangle \quad -\boxed{H}- \quad |0\rangle + |1\rangle \quad -\boxed{H}- \quad |0\rangle$$



Example Quantum Gates: The Hadamard Gate

$$|1\rangle \;\; -\boxed{H}- \;\; |0\rangle - |1\rangle \;\; -\boxed{H}- \;\; |1\rangle$$

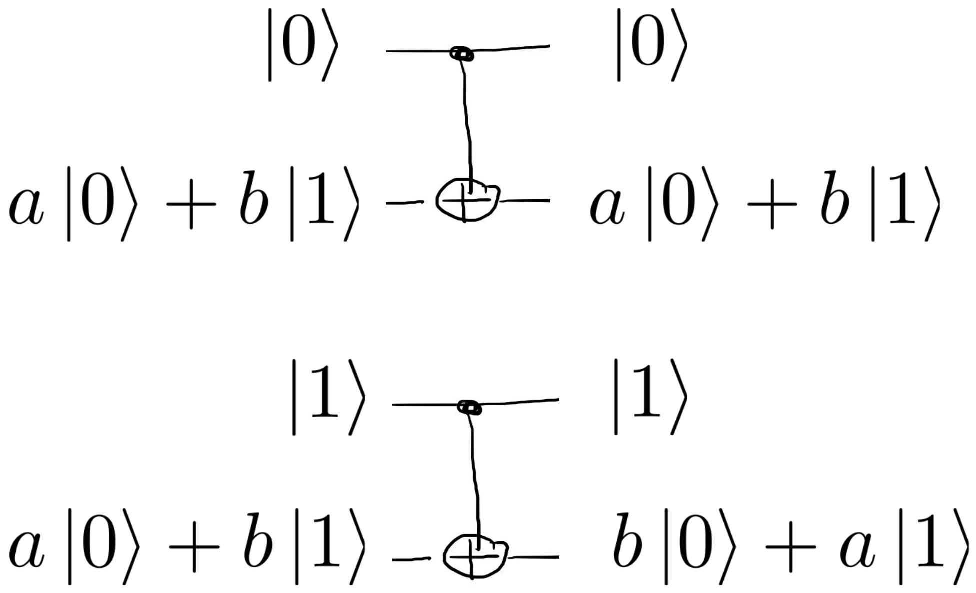


Example Quantum Gates: The Controlled-NOT Gate

The Quantum “If” (CNOT)

Example Quantum Gates: The Controlled-NOT Gate

The Quantum “If” (CNOT)

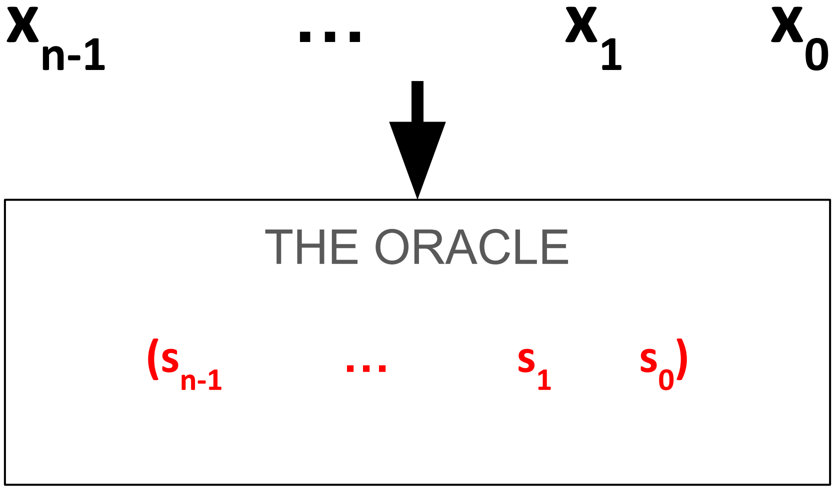


A CNOT flips the target bit if the control bit is 1

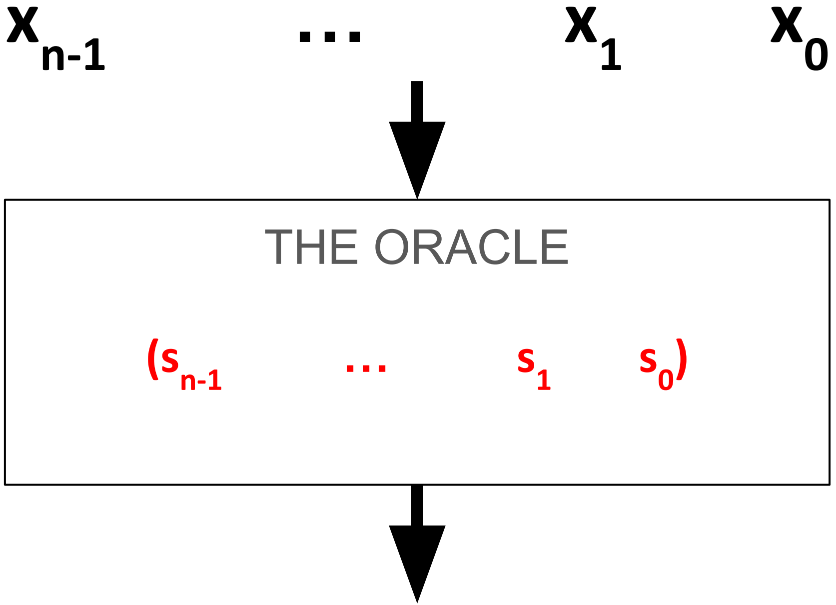
Quantum Algorithm: Bernstein-Vazirani

$\mathbf{x_{n-1}}$ **…** $\mathbf{x_1}$ $\mathbf{x_0}$

Quantum Algorithm: Bernstein-Vazirani



Quantum Algorithm: Bernstein-Vazirani



$$x_{n-1}s_{n-1} \oplus \ldots \oplus x_1 s_1 \oplus x_0 s_0$$

Quantum Algorithm: Bernstein-Vazirani

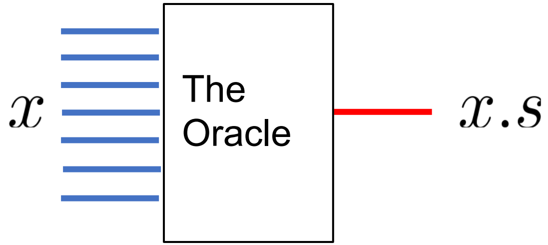
Classically, we need *n* tries.

$$\textbf{Optimal classical strategy:}\left\{\begin{array}{l} X = 1\ \ 0\ \ \dots\ 0\ \ 0\ \ (2^{n-1}) \\ X = 0\ \ 1\ \ \dots\ 0\ \ 0\ \ (2^{n-2}) \\ . \\ . \\ X = 0\ \ 0\ \ \dots\ 1\ \ 0\ \ (2) \\ X = 0\ \ 0\ \ \dots\ 0\ \ 1\ \ (1) \end{array}\right\}\ \textbf{n tries}$$

Quantumly, we need *1* try.

Quantum Algorithm: Bernstein-Vazirani

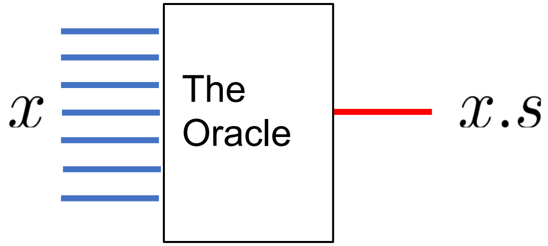
Classical Vs. Quantum Oracle



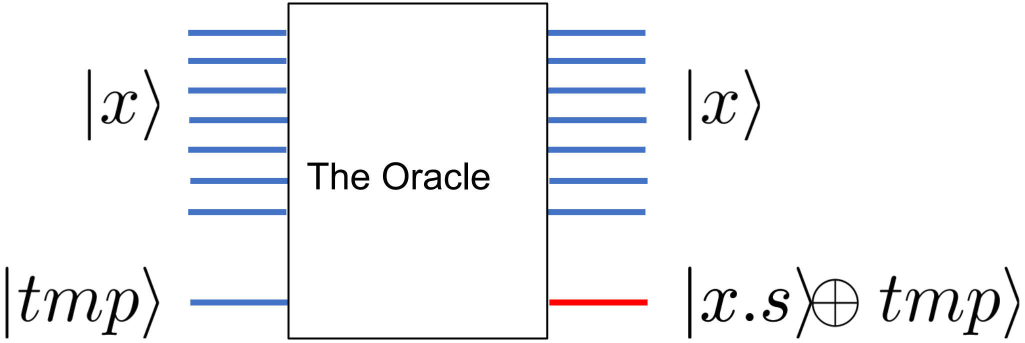
Classical Dot-Product Oracle

Quantum Algorithm: Bernstein-Vazirani

Classical Vs. Quantum Oracle



Classical
Dot-Product
Oracle



Quantum
Dot-Product Oracle

Difference? Must be reversible!

Quantum Algorithm: Bernstein-Vazirani

Implementing the Oracle

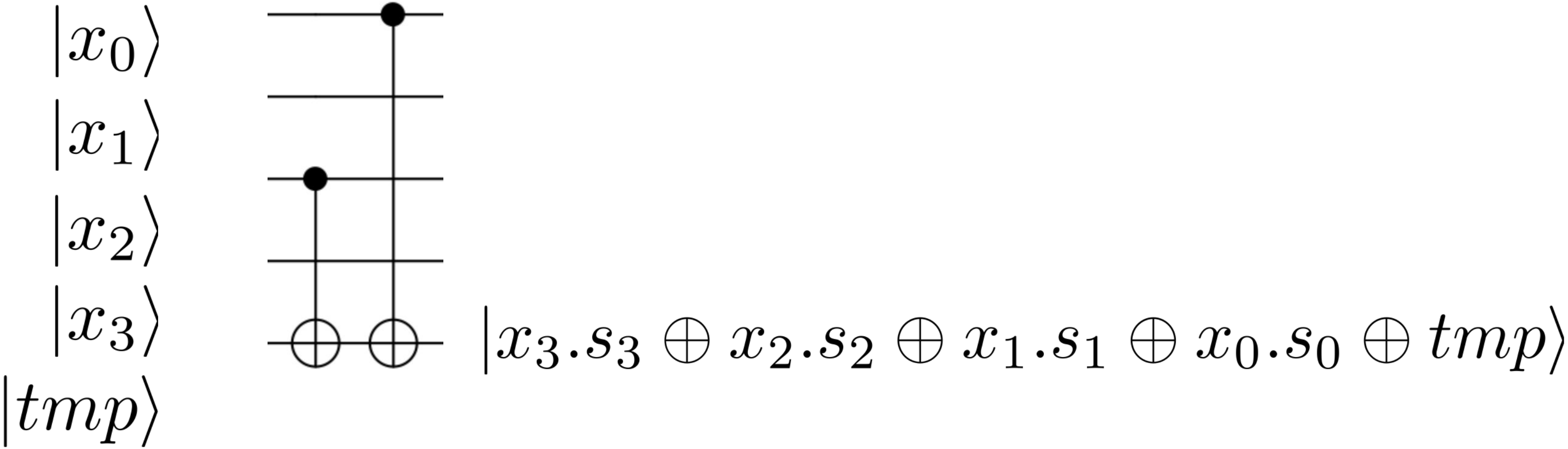
The control pattern for the oracle depends on the hidden bitstring.

Quantum Algorithm: Bernstein-Vazirani

Implementing the Oracle

The control pattern for the oracle depends on the hidden bitstring.

s = 0101

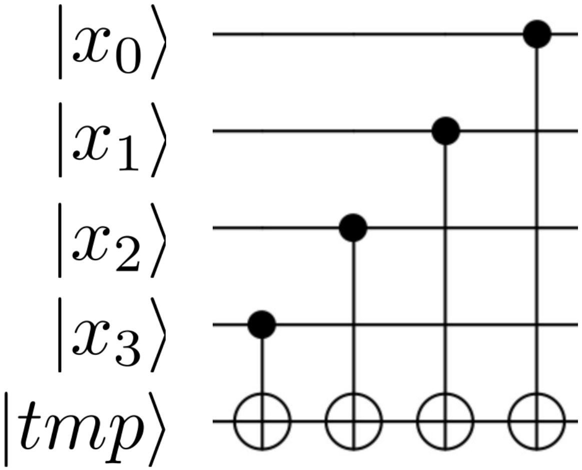


Quantum Algorithm: Bernstein-Vazirani

Implementing the Oracle

The control pattern for the oracle depends on the hidden bitstring.

S = 1111



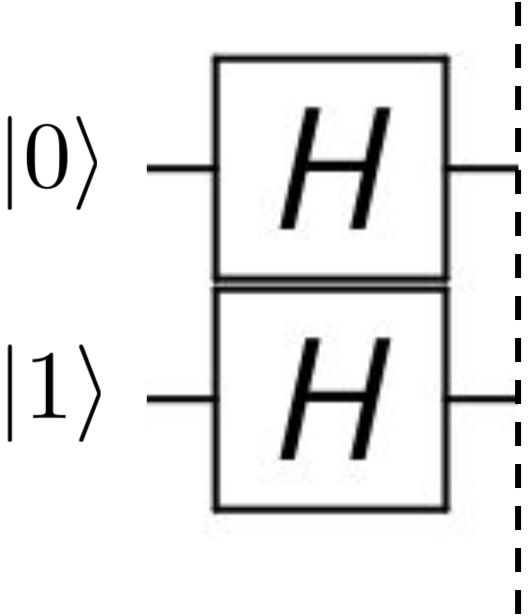
$$|x_3.s_3 \oplus x_2.s_2 \oplus x_1.s_1 \oplus x_0.s_0 \oplus tmp\rangle$$

Quantum Algorithm: Bernstein-Vazirani

The Key Trick - Phase Kickback

Quantum Algorithm: Bernstein-Vazirani

The Key Trick - Phase Kickback



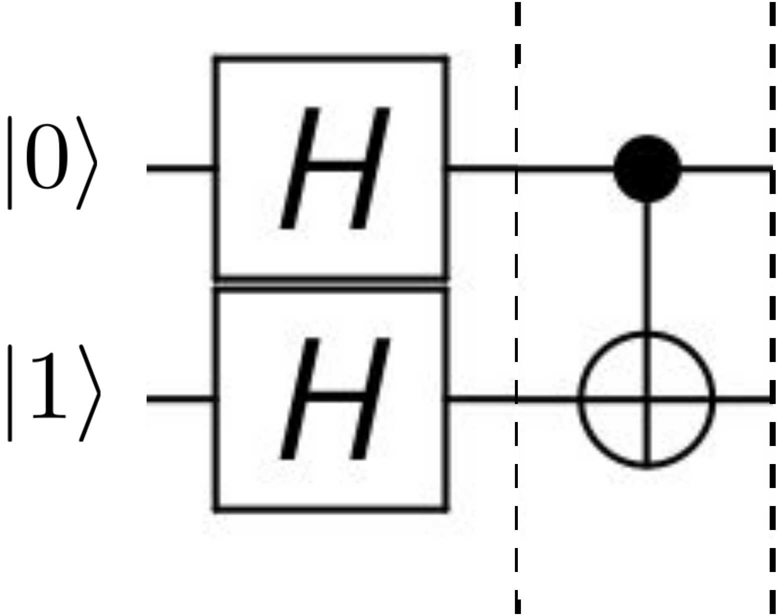
$$(|0\rangle + |1\rangle)(|0\rangle - |1\rangle) = |00\rangle - |01\rangle + |10\rangle - |11\rangle$$

Quantum Algorithm: Bernstein-Vazirani

The Key Trick - Phase Kickback

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

Phase kickback!

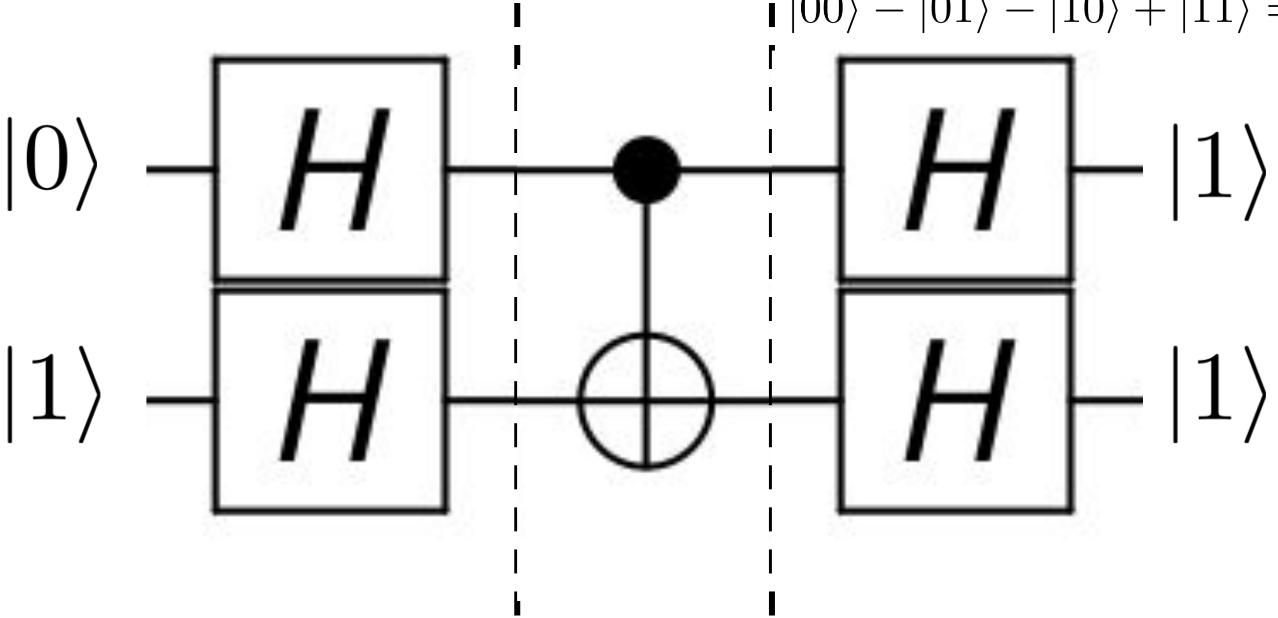


$$(|0\rangle + |1\rangle)(|0\rangle - |1\rangle) = |00\rangle - |01\rangle + |10\rangle - |11\rangle$$

Quantum Algorithm: Bernstein-Vazirani

The Key Trick - Phase Kickback

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



Phase kickback!

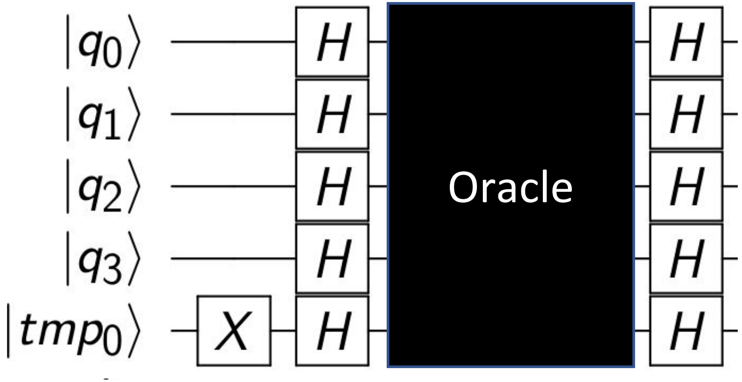
$$(|0\rangle + |1\rangle)(|0\rangle - |1\rangle) = |00\rangle - |01\rangle + |10\rangle - |11\rangle$$

Quantum Algorithm: Bernstein-Vazirani

Putting it all together

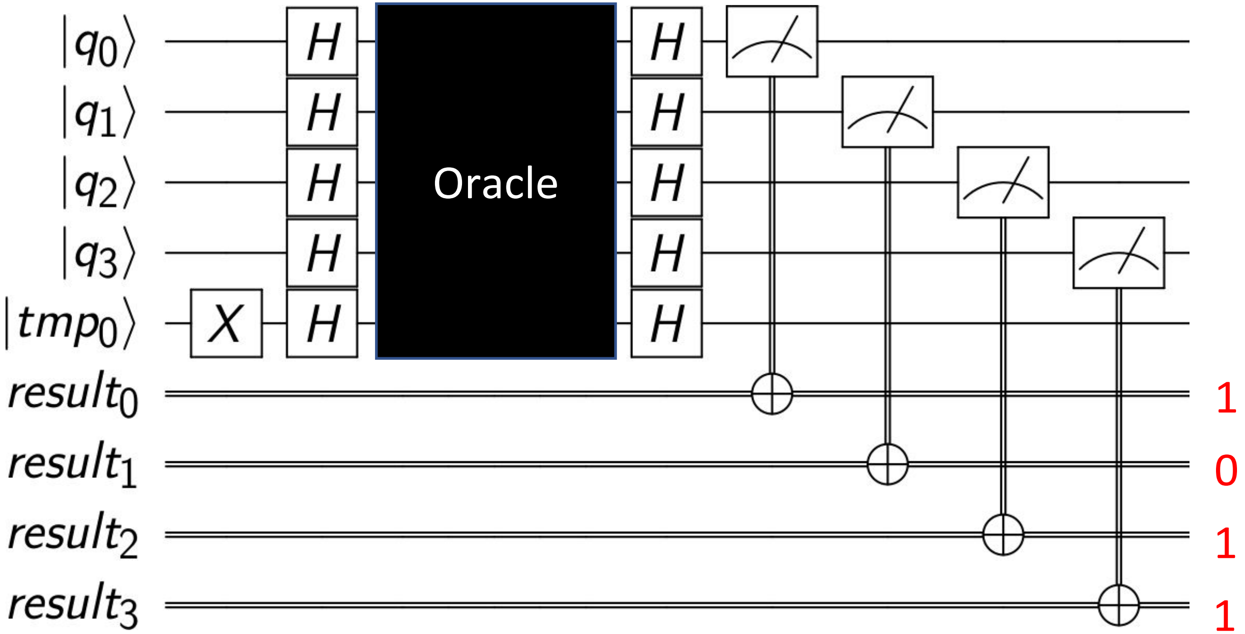
Quantum Algorithm: Bernstein-Vazirani

Putting it all together



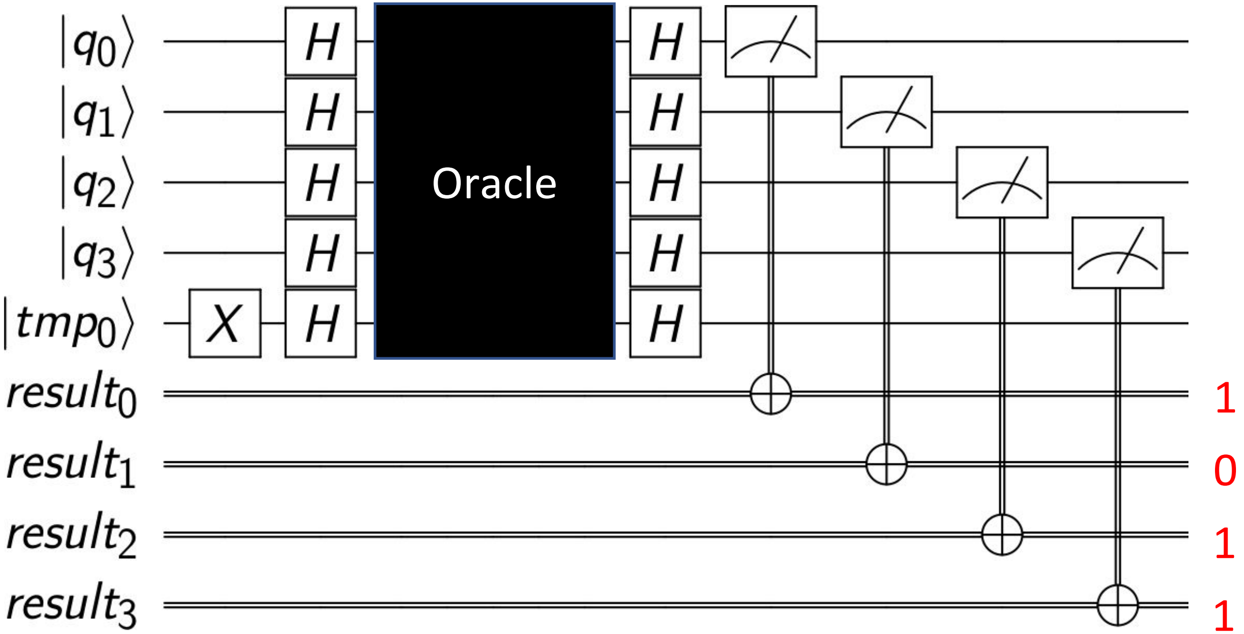
Quantum Algorithm: Bernstein-Vazirani

Putting it all together



Quantum Algorithm: Bernstein-Vazirani

Putting it all together



Wherever there’s CNOT, phase kickback puts that control qubit in state |1>.

Quantum Algorithm: Bernstein-Vazirani

Why did it work?

1. Classical oracles can only be queried with a single number at a time. Quantum oracles can be queried in superposition.

2. We don’t just “try every answer simultaneously”. The problem had a structure that we could exploit using qubits: **encode information in phases**