# Core Principles of Quantum

Disclaimer: Work in progress. Portions of these written materials are incomplete.

# Overview

# The quantum tech ecosystem

Quantum Computing

Quantum Communication

Quantum Sensing

# The Big Picture -
## The Three Pillars of Quantum Technologies

### Quantum Communications

- Needed to network together multiple quantum computers
- Provides the ability to conduct communications that are unconditionally secure

### Quantum Computing

- Solves problems that are effectively impossible to solve on classical computers
- Powerful quantum computing machines will pose distinct threat to communications systems that rely on public key cryptography

### Quantum Sensing

- Makes use of quantum effects to get enhanced sensitivity to stimuli

# Quantum Computing

- Idea: making a computer out of quantum-mechanical building blocks in order to better mimic nature’s quantum mechanical properties

- Richard Feynman (1982): *“Nature isn't classical, dammit, and if you want to make a simulation of nature, you'd better make it quantum mechanical.”*

# Quantum Mechanics and Computation

Complexity: quantum computers seem to be fundamentally more powerful than classical computers

Examples of quantum speedup:

- Shor’s algorithm (factoring)
- Grover’s algorithm (search)
- Quantum chemistry simulations
- Etc.

# Possible Quantum Exponential Speedup

QC’s are Turing complete

- Can execute any algorithm a classical computer can
- Does *not* mean one gets a speedup for any given algorithm

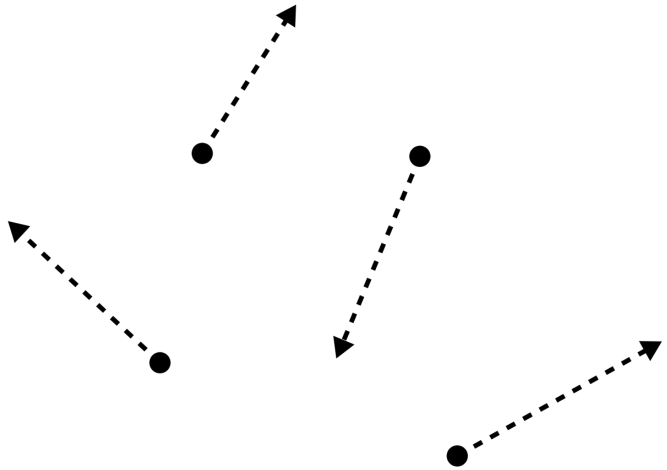
Certain algorithms *do* have a speedup on a QC

- In some cases, exponential speedup such as Shor’s algo
- For other algorithms, speedups are either quadratic or non-existent

# The Vector Space of Quantum States

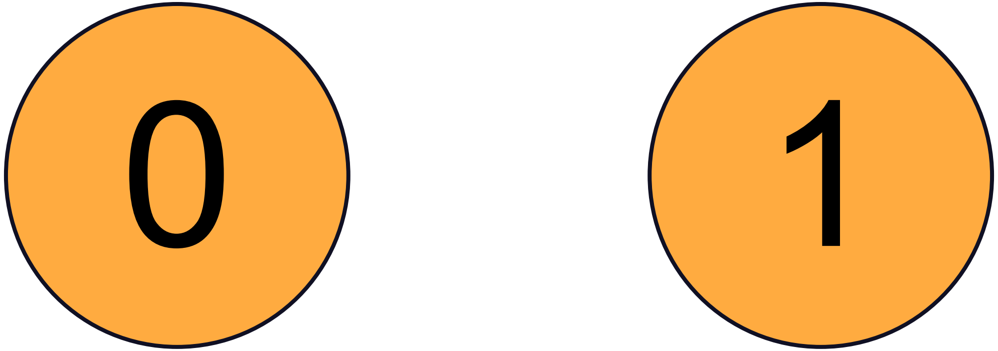
The **state** of a system in physics just means everything you need to know about the system to predict what will happen to it or what you will see when you look at it or probe it with some apparatus.

**Example**: the position and velocity of a bunch of atoms



**Example**: Charge on a capacitor and current through an electric circuit.

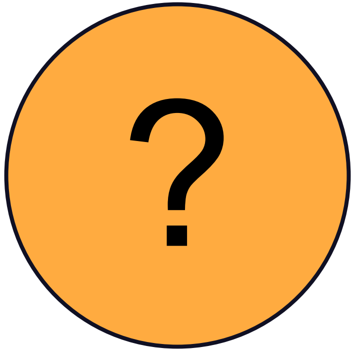
**Very simple example**: A system with only two states, which we can label 0 and 1, or Heads and Tails.



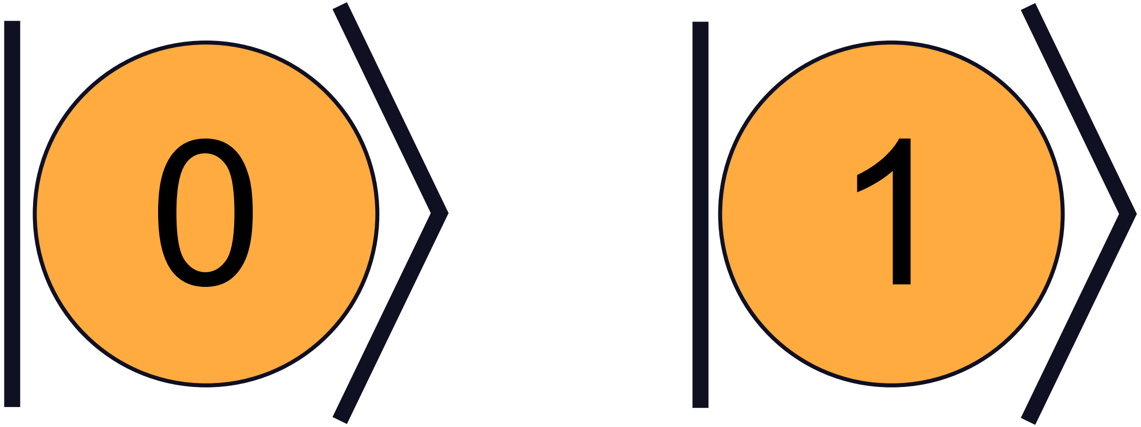
In classical physics, the collection of all the states of a system forms a **set**.

If you want to know which state you're in you just measure all the properties of the system.

So, for example, you measure all of the positions and all of the velocities of all of the atoms. Or you measure the side of the coin (by looking at it).



In quantum physics, the collection of all the states of a system forms a **complex vector space** (really, a Hilbert space). So we like to use **ket notation** to denote quantum states.



If quantum states are vectors, that means we can take linear combinations of them to get new states.

What does this mean??

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

If quantum states are vectors, that means we can take linear combinations of them to get new states.

What does this mean??

$$2|0\rangle - i|1\rangle$$

The meaning comes from asking about the outcomes of **measurements** and is related to **fundamental randomness** in quantum mechanics.

If we measure the “value” of the qubit (by looking at it) we will only ever see 0 or 1, but the **probability of seeing each of the possible outcomes is determined by the linear combination.**

$$\text{Prob}(0) = |\text{coeff. of } |0\rangle|^2$$

There is an inner product in quantum mechanics, such that distinguishable states like 0 and 1 are **orthogonal** (really **orthonormal**). Then we can say mathematically that

$$\text{Prob}(0 \text{ in the state } |v\rangle) = |\langle 0|v\rangle|^2$$

This definition makes sense if we always agree to **normalize** the quantum states. That is, we always have

$$\langle v | v \rangle = 1$$

**Exercise**: Show that this means the probability of measuring 0 and the probability of measuring 1 sum to one!

# States and Superposition Summary

Instead of 0 and 1 (classical bit), we have |0> and |1> (**qubit**).

**Superpositions** of states are allowed.

$$\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$$

Coefficients/**amplitudes** are complex numbers.

Constructive and destructive **interference** can occur.

Sum of squares of absolute values = 1.

# Computational Basis and Other Bases

We normally expand the wavefunction in terms of a basis of bit strings: the **computational basis**.

$$|\psi\rangle = \frac{i}{\sqrt{3}}|010\rangle + \frac{\sqrt{2}}{\sqrt{3}}|111\rangle$$

$2^n$ amplitudes for n qubits.

Other bases are sometimes convenient, e.g., the X basis

$$|+\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle \qquad |-\rangle = \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle$$

# States as Column Vectors

$$|0\rangle \leftrightarrow \begin{pmatrix} 1 \\ 0 \end{pmatrix} \qquad |1\rangle \leftrightarrow \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

$$\alpha|0\rangle + \beta|1\rangle \leftrightarrow \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$$

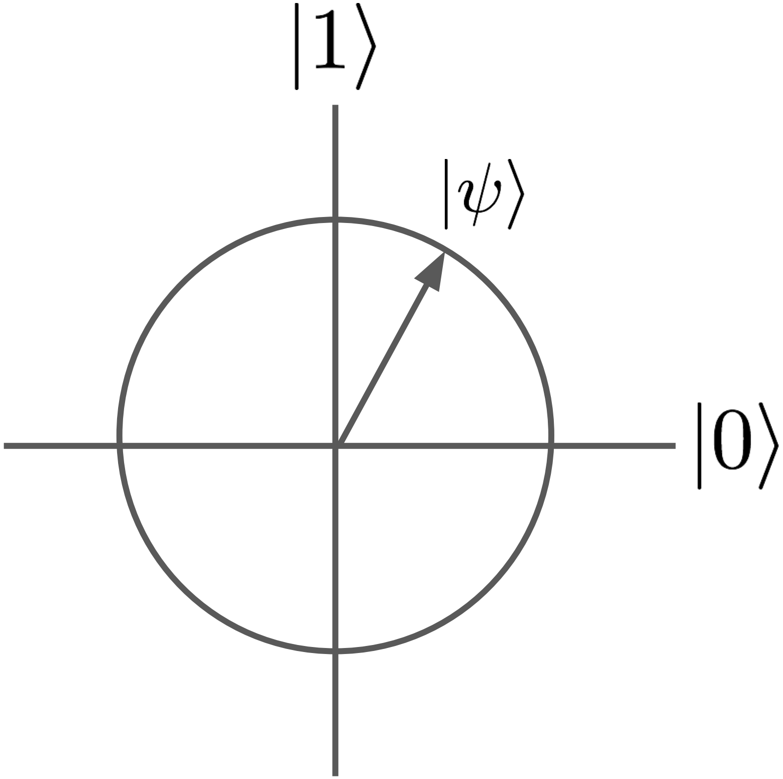
# States as Column Vectors

$$|00\rangle \leftrightarrow \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \qquad |01\rangle \leftrightarrow \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

$$|10\rangle \leftrightarrow \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \qquad |11\rangle \leftrightarrow \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

# Single-Qubit Cartoon

What is this cartoon lacking?



# Bloch Sphere

Antipodal points = **orthogonal** states (perfectly distinguishable)

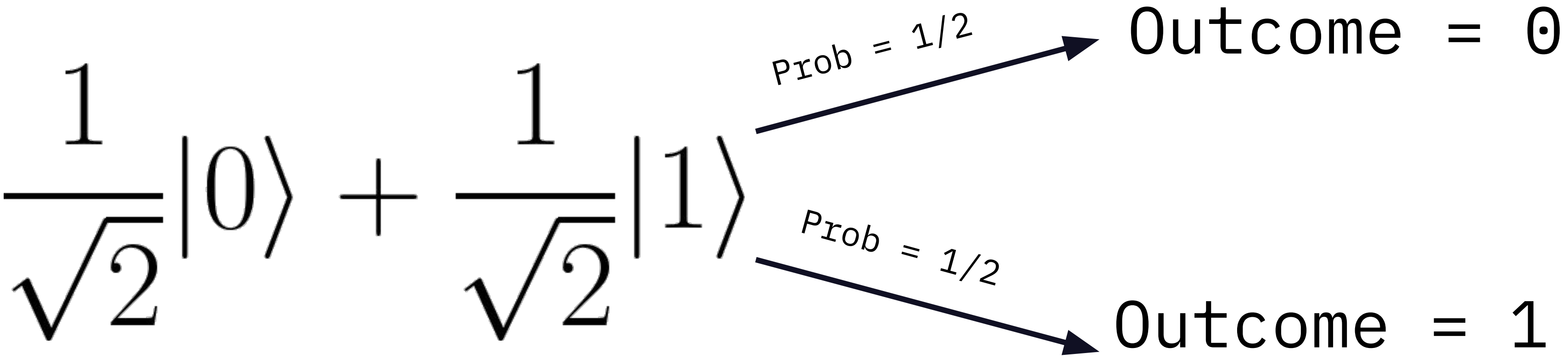
Rotations = **unitary** operations (more on this later)

No convenient analogue for multiple qubits, but still useful for a single qubit

# Measurements

What happens when we measure a property of a quantum system?

The first thing to know is that the outcome is **random with probability determined by the Born rule**.



$$\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$$

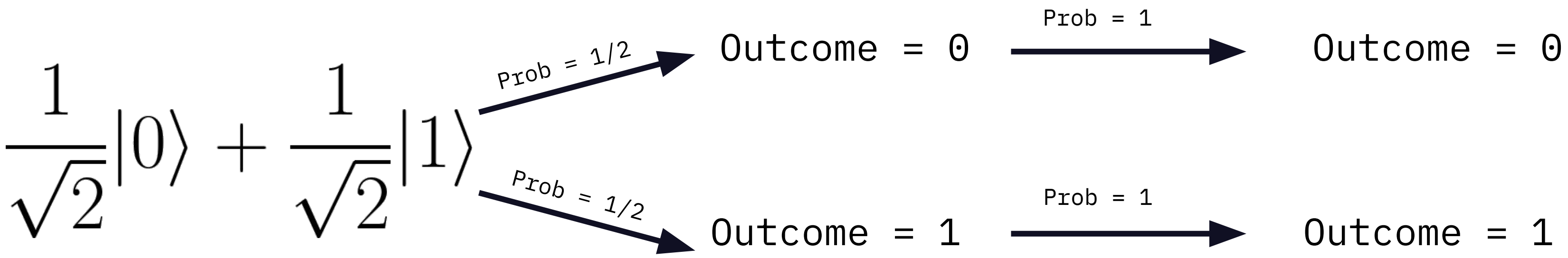
Prob = 1/2

Outcome = 0

Prob = 1/2

Outcome = 1

Experimental fact: Repeatedly measuring a system gives consistent, repeatable results.



$$\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$$

This fact (together with others) can be encapsulated in the **collapse rule** for quantum states. All it says is that a quantum state should be updated based on new information, similar to an ordinary probability distribution.

Superpositions are “destroyed” by collapse!

$$\begin{tikzcd}
 & {|0\rangle} \arrow[r, "\text{Prob = 1}"] & {|0\rangle} \\
\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle \arrow[ur, "\text{Prob = 1/2}"] \arrow[dr, "\text{Prob = 1/2}"'] & & \\
 & {|1\rangle} \arrow[r, "\text{Prob = 1}"] & {|1\rangle}
\end{tikzcd}$$

You don’t have to just measure the “label” of a basis state!

Consider two qubits and imagine measuring the **parity** of the qubits.

Parity = +1

$$|00\rangle \quad |11\rangle$$

Parity = -1

$$|01\rangle \quad |10\rangle$$

Parity = +1

$$|00\rangle \ |11\rangle$$

Parity = -1

$$|01\rangle \ |10\rangle$$

We can think of the positive and negative parity **subspaces** spanned by the positive and negative parity vectors, respectively.

Here’s a state with probability ⅔ to have parity +1 and probability ⅓ to have parity -1. After measuring the parity, the state “collapses” onto the **subspace** consistent with the measurement outcome.

$$\begin{tikzcd}
 & \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle \\
\frac{1}{\sqrt{3}}|00\rangle + \frac{1}{\sqrt{3}}|01\rangle + \frac{1}{\sqrt{3}}|11\rangle \arrow[ur, "\text{Prob = 2/3}"] \arrow[dr, "\text{Prob = 1/3}"] & \\
 & |01\rangle
\end{tikzcd}$$

# Measurement and Born Rule

Quantum state is not directly observable --- sampling from the Born distribution is all we can do.

Quantum computer outputs 1s and 0s.

Probability = Absolute Value Squared of Amplitude.

Repeating a measurement **immediately** returns the **same** answer.

Must repeat the whole experiment to resample from the distribution.

# Break

(10 min.)

# Entanglement and Quantum Key Distribution

**Entanglement** refers to correlations that exist due to superpositions of multi-qubit states.

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

The **Bell state** (also known as a **Bell pair** or **EPR pair**) has the property that either qubit is measured as 0 or 1 with **equal probability**, but **both qubits always come out the same**.

# Quantum Key Distribution

## Entanglement-based: E91

Ekert (1991):

- Uses entanglement (Bell pairs) to create shared key
- Security guaranteed by physics

Protocol:

1. Alice, Bob pick two (mutually unbiased) bases. E.g.. the computational basis and the X basis.
2. Alice creates a Bell Pair of 2 qubits, sends one qubit to Bob.
3. Alice, Bob measure their qubits in one of the two bases independently and at random.
4. Alice, Bob classically communicate to share basis choices. They keep measurement results from coinciding basis choices.
5. Estimate error rates, and either:
    a. Abort if too high
    b. Correct errors, perform privacy amplification to distill a shared random private (symmetric) key

| | |
|---|---|
| | |
| | |
| | |
| | |
| | |
| | |

| | |
|---|---|
| | |
| | |
| | |
| | |
| | |
| | |

| | |
|---|---|
| 0,1 | |
| 0,1 | |
| +,- | |
| 0,1 | |
| +,- | |
| +,- | |

Alice and Bob start with many shared Bell pairs.

They independently choose bases to measure their respective qubits in.

| | |
|---|---|
| 0,1 | |
| +,- | |
| +,- | |
| 0,1 | |
| +,- | |
| 0,1 | |

| | |
|---|---|
| 0,1 | 0 |
| 0,1 | 1 |
| +,- | + |
| 0,1 | 1 |
| +,- | + |
| +,- | - |

They measure their qubits, each obtaining a random sequence of 0s and 1s. (“+” can be registered as 0 and “-” as 1).

| | |
|---|---|
| 0,1 | 0 |
| +,- | + |
| +,- | + |
| 0,1 | 1 |
| +,- | + |
| 0,1 | 1 |

| | |
|---|---|
| 0,1 | 0 |
| ~~0,1~~ | ~~1~~ |
| +,- | + |
| 0,1 | 1 |
| +,- | + |
| ~~+,-~~ | ~~-~~ |

The basis choices are communicated and only instances where the same basis was chosen are retained. This should be about half of the time.

| | |
|---|---|
| 0,1 | 0 |
| ~~+,-~~ | ~~0~~ |
| +,- | + |
| 0,1 | 1 |
| +,- | + |
| ~~0,1~~ | ~~1~~ |

| | |
|---|---|
| 0,1 | 0 |
| ~~0,1~~ | ~~1~~ |
| +,- | + |
| 0,1 | 1 |
| +,- | + |
| ~~+,-~~ | ~~-~~ |

Alice and Bob can confirm that they agree on a subset of the qubits that were kept, and they can confirm that they only agree about half the time on qubits that were thrown out.

| | |
|---|---|
| 0,1 | 0 |
| ~~+,-~~ | ~~0~~ |
| +,- | + |
| 0,1 | 1 |
| +,- | + |
| ~~0,1~~ | ~~1~~ |

# Bell Pairs and Basis Changes

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

An equal superposition in the computational basis is also an equal superposition in the X basis! This is a key fact in the QKD protocol.

# Suppose A and B choose the same basis...

$$\frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle \xrightarrow[\text{MEASUREMENT}]{} |00\rangle \text{ OR } |11\rangle$$

$$\frac{1}{\sqrt{2}}|++\rangle + \frac{1}{\sqrt{2}}|--\rangle \xrightarrow[\text{MEASUREMENT}]{} |++\rangle \text{ OR } |--\rangle$$

Alice and Bob get the same result, but the result is random!

# Suppose A and B choose different bases...

$$\frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle \xrightarrow[\text{MEASUREMENT}]{} |00\rangle \text{ OR } |11\rangle$$

$$|00\rangle = \frac{1}{\sqrt{2}}|0+\rangle + \frac{1}{\sqrt{2}}|0-\rangle$$

Alice and Bob get random, uncorrelated results!

# Quantum Key Distribution

## Prepare-and-measure: BB84

### Bennett & Brassard (1984):

-originally used photon polarization as qubit

-generalizes to other types of qubits

-Security guaranteed by physics

### Protocol:

1. Pick two (mutually unbiased) bases
2. Alice encodes bit in a qubit in one of two random bases
3. Bob projects received qubits in one of two random bases
4. Alice and Bob classically use an authenticated classical channel to share basis choices
    a. Keep results from coinciding basis choices
5. Shared randomness (symmetric key) distilled
    a. any eavesdropping is detectable in the statistics

# How Quantum States Change with Time

There are **two rules** determining how quantum states change with time in the absence of measurement or decoherence.

1) Linearity

$$|v_0\rangle \mapsto |v_t\rangle \text{ and } |w_0\rangle \mapsto |w_t\rangle$$

$$\text{implies}$$

$$a|v_0\rangle + b|w_0\rangle \mapsto a|v_t\rangle + b|w_t\rangle$$

There are **two rules** determining how quantum states change with time in the absence of measurement or decoherence.

2) Conservation of information - orthonormal states map onto orthonormal states

$$|v_0\rangle \text{ and } |w_0\rangle \text{ orthonormal}$$

$$\text{implies}$$

$$|v_t\rangle \text{ and } |w_t\rangle \text{ orthonormal}$$

Putting these rules together means that time evolution is determined by **unitary matrices**.

$$|v_t\rangle = U|v_0\rangle$$

# Unitary Operators

A unitary operator U satisfies

$$UU^\dagger = U^\dagger U = I$$

where $U^\dagger$ denotes the Hermitian conjugate, i.e. the conjugate transpose of U.

# Unitary Operators Preserve Distinguishability

$$\langle\psi|\phi\rangle = \langle\psi|U^\dagger U|\phi\rangle = \left(\langle\psi|U^\dagger\right)\left(U|\phi\rangle\right)$$

Inner products measure distinguishability.

Not going to review in detail here.

# Break

(10 min.)

# Quantum Computing

# A Brief Early History of Quantum Computing

- 1979 - Paul Benioff details the principles of quantum computing
- 1980 - Manin publishes Computable and Non-Computable, a book which describes quantum computers
- 1981 - Feynman proposed the idea of creating machines based on the laws of quantum mechanics instead of the laws of classical physics.
- 1985 - David Deutsch developed the quantum Turing machine, showing that quantum circuits are universal.
- 1994 - Peter Shor develops a quantum algorithm to factor very large numbers in polynomial time.
- 1997 - Lov Grover develops a quantum search algorithm with O(N^.5) complexity.
- 1999 -  Yasunobu Nakamura realizes the first superconducting qubit.
- 2000 - David DiVincenzo publishes the basic requirements of a full-fledged quantum computer.
- 2001 - Isaac Chuang et. al. demonstrate the first realization of Shor’s algorithm on a nuclear magnetic resonance platform.

# DiVincenzo 5 Principles of QC

# Quantum Circuits

# Picture of a Quantum Algorithm

# Quantum Gates

# Quantum Circuit = Time Evolution

We construct the time evolution operator from simple building blocks.

Those building blocks are the quantum gates.

# Operators as Matrices

Just like how we represent states as column vectors, we can represent operators as matrices which act on those column vectors.

For example, the X operator:

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

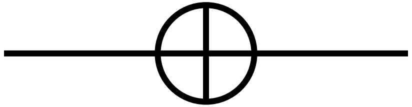
$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

# Single-Qubit Gates

# Pauli-X (NOT)

$$|0\rangle \mapsto |1\rangle$$
$$|1\rangle \mapsto |0\rangle$$

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



# Applying an operator to a state

If we represent the state |0> as a vector (1 0) and then wish to apply an operator to it, we can represent the operator as a matrix and simply perform matrix-vector multiplication.  For example, to apply NOT to the state |0>:

$$\mathbf{X} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

.

# X Operator on the Bloch Sphere

Rotates around the X axis by
180°

Clockwise or counterclockwise?

# Pauli-Y

$$|0\rangle \mapsto i|1\rangle$$
$$|1\rangle \mapsto -i|0\rangle$$

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

# Y Operator on the Bloch Sphere

Rotates around the Y axis by 180°

# Pauli-Z (Phase Flip)

Diagonal in the computational basis

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

$$|0\rangle \mapsto |0\rangle$$

$$|1\rangle \mapsto -|1\rangle$$

# Z Operator on the Bloch Sphere

Rotates around the Z axis by
180°

# Hadamard Gate

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

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

$$H = \frac{1}{\sqrt{2}}(X + Z) = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

# H Operator on the Bloch Sphere

Rotates around the “X+Z” axis by 180°

Exchanges X with Z

$$HZH = X$$

# Arbitrary Rotations of the Bloch Sphere

Exponentiate the Z operator to rotate by an arbitrary angle around the Z axis.

$$e^{-i\pi x Z} = \begin{pmatrix} e^{-i\pi x} & 0 \\ 0 & e^{i\pi x} \end{pmatrix}$$

# Arbitrary Rotations of the Bloch Sphere

What about a ½ rotation?

Shouldn’t that just be Z?

$$e^{-i\pi Z/2} = \begin{pmatrix} -i & 0 \\ 0 & i \end{pmatrix}$$

# Arbitrary Rotations of the Bloch Sphere

Quarter-rotation and eighth-rotation have names (up to overall phase).

$$
\begin{aligned}
e^{-i\pi Z/4} &= \begin{pmatrix} e^{-i\pi/4} & 0 \\ 0 & e^{i\pi/4} \end{pmatrix} \\
&= e^{-i\pi/4} \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} \\
&= e^{-i\pi/4} S
\end{aligned}
$$

$$
\begin{aligned}
e^{-i\pi Z/8} &= \begin{pmatrix} e^{-i\pi/8} & 0 \\ 0 & e^{i\pi/8} \end{pmatrix} \\
&= e^{-i\pi/8} \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix} \\
&= e^{-i\pi/8} T
\end{aligned}
$$

# Arbitrary Rotations of the Bloch Sphere

Trick for exponentiating certain operators. Works because $Z^2 = 1$.

Similar formula for other Pauli matrices.

$$\begin{aligned} e^{-i\pi x Z} &= cos \pi x - i(\sin \pi x) Z \\ &= \begin{pmatrix} e^{-i\pi x} & 0 \\ 0 & e^{i\pi x} \end{pmatrix} \end{aligned}$$

# Arbitrary Rotations of the Bloch Sphere

Exponentiate the X operator to rotate by an arbitrary angle around the X axis, similarly with Y

$$e^{-i\pi x X} = \begin{pmatrix} \cos \pi x & -i \sin \pi x \\ -i \sin \pi x & \cos \pi x \end{pmatrix}$$

# Two-Qubit Gates

# Controlled Gates

Acts as unitary operator U on the target qubit when the control qubit is in the |1> state.

$$\mathrm{CU} = \begin{array}{cc} \begin{array}{cccc} 00 & 01 & 10 & 11 \end{array} & \\ \left(\begin{array}{cccc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \multicolumn{2}{c}{\multirow{2}{*}{$U$}} \\ 0 & 0 & & \end{array}\right) & \begin{array}{c} 00 \\ 01 \\ 10 \\ 11 \end{array} \end{array}$$

# Controlled NOT (CNOT)

If the control bit is |0>, the target bit is left unchanged.

If the control bit is |1> then the target bit is flipped.

CNOT is commonly used to entangle two qubits.

$$\mathrm{CNOT} = \begin{array}{cc} \begin{matrix} 00 & 01 & 10 & 11 \end{matrix} & \\ \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} & \begin{matrix} 00 \\ 01 \\ 10 \\ 11 \end{matrix} \end{array}$$

# Controlled Z (CZ)

Acts as Z on the target qubit when the control bit is |1>.

CZ is symmetric between the two qubits --- it doesn’t matter which bit is the control!

$$\mathrm{CZ} = \begin{array}{cc} \begin{array}{cccc} 00 & 01 & 10 & 11 \end{array} & \\ \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} & \begin{array}{c} 00 \\ 01 \\ 10 \\ 11 \end{array} \end{array}$$

# SWAP

Exchanges the states of two qubits.

Equivalent to “crossing the wires.”

$$\mathrm{SWAP} = \begin{array}{cccccc} & 00 & 01 & 10 & 11 & \\ \left(\begin{matrix} \\ \\ \\ \\ \end{matrix}\right. & \begin{matrix} 1 \\ 0 \\ 0 \\ 0 \end{matrix} & \begin{matrix} 0 \\ 0 \\ 1 \\ 0 \end{matrix} & \begin{matrix} 0 \\ 1 \\ 0 \\ 0 \end{matrix} & \begin{matrix} 0 \\ 0 \\ 0 \\ 1 \end{matrix} & \left.\begin{matrix} \\ \\ \\ \\ \end{matrix}\right) \begin{matrix} 00 \\ 01 \\ 10 \\ 11 \end{matrix} \end{array}$$

# Not all gates created equal...

A given hardware platform might perform single-qubit gates and controlled phase rotations (or some other choice of two-qubit gates.

All others must be compiled from these.

Solovay-Kitaev Theorem: A small “universal” gate set can be used to compile any quantum operation with low overhead!

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & e^{i\pi x} \end{pmatrix}$$

# Fredkin operator (CSWAP)

Controlled swap - trinary operator

In this operator we look to the control qubit to determine whether we swap two target qubits.

# Toffoli operator

This is a trinary operator.  It uses the boolean AND on the two control qubits.  We apply X in two of the eight cases

|000> → |000>

|001> → |001>

|010> → |010>

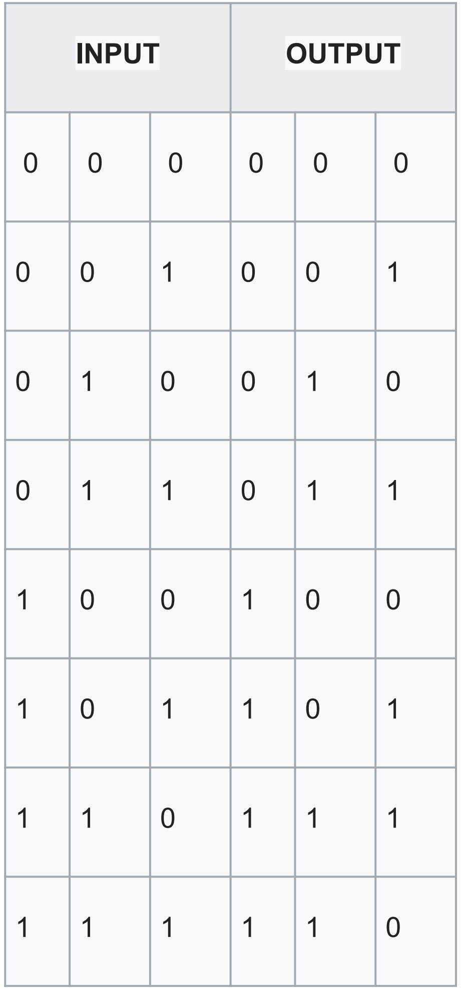
|011> → |011>

|100> → |100>

|101> → |101>

|110> → |111>

|111> → |110>



| INPUT | | | OUTPUT | | |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 | 1 |
| 1 | 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 0 |

# Quantum Teleportation

# Quantum Teleportation

## Quantum Teleportation

-Method to leverage entanglement on a network to transfer **qubits**

-Can be chained in a sequence

-Can be used to transfer quantum data between quantum computers

## Protocol:

1. Alice and Bob share a Bell pair. Alice has an input state to transfer to Bob
2. Alice performs a Bell measurement on her qubits
3. Alice relays measurement results (classical data) to Bob
4. Bob applies corresponding correction to his qubit
5. Quantum state transfer complete

# Break

(10 min.)

# Quantum Error Correction

# The Challenge: Quantum Error Correction

The key challenge with QC is to build one with lots of qubits and maintain all the qubits in a superposition while we do a calculation.  This is called maintaining coherence.

# Quantum Error Correction

Want to detect that errors have occurred without measuring the state, and be able to correct those errors.

Problem: Aren’t the number of possible errors infinite and continuous?

Answer: Yes, but you can decompose any error in terms of fundamental errors! Similar to gate decomposition. On a single qubit, errors can always be decomposed in terms of combinations of bit flips (X) and phase flips (Z)!

# Bit Flip Correction

$$|0_L\rangle = |000\rangle$$
$$|1_L\rangle = |111\rangle$$

Notice that the parity of neighboring qubits is even in both of these states. So if we measure both of those parities, we can tell if a single qubit has flipped (and which one has flipped) without saying whether we are in $|0_L>$ or $|1_L>$.

$Z_1Z_2$ and $Z_2Z_3$ are the operators to check.

# Phase Flip Correction

$$|0_L\rangle = |+++\rangle$$
$$|1_L\rangle = |---\rangle$$

We can measure “X basis parity” in order to detect phase flip errors.

$X_1X_2$ and $X_2X_3$ are the operators to check.

# Shor Code

Combine bit flip and phase flip codes to get a code that can correct against any single-qubit error!

$$|0_L\rangle = (|000\rangle + |111\rangle)(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)/\sqrt{8}$$

$$|1_L\rangle = (|000\rangle - |111\rangle)(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)/\sqrt{8}$$

This is a very simple 9 qubit code, but it is not powerful enough to be practical. More powerful codes can detect and correct errors on multiple qubits simultaneously. Modern codes can have 1000:1 physical-to-logical ratios.

# Fault Tolerance

Implementing an error correcting code requires overhead not only in number of qubits, but also in terms of circuit size (logical operations are built out of many physical operations.

Increasing the size of the circuit introduces more opportunities for errors. Doesn’t this make things worse?

As long as the fundamental error rate in the hardware is below a certain threshold, we will win (theorem). Then we can concatenate our code with itself to decrease the effective error rate as much as we want, as long as we have a large enough computer.

# Spotlight Algorithms

# Shor’s Algorithm

Shor’s Algorithm on Wikipedia

# Shor’s Algorithm

Five steps for factoring a number N:

1) Check that N is not prime, not even, and not a perfect power ($a^b$).
2) Choose random integer a < N. If gcd(a,N) is not 1, we are finished (unlikely!), else continue.
3) Compute smallest integer r such that $a^r$ = 1 mod N. r is called “the order of a mod N,” and r > 0 because gcd(a,N) = 1.
4) If r is odd or $a^{r/2}$ = -1 mod N, go to (2).
5) At least one of gcd($a^{r/2}$ + 1,N) and gcd($a^{r/2}$ - 1,N) is a factor of N (likely both).

# Factoring 15

Following Shor's algorithm:

1) Check
2) Choose a = 8
3) $a^2 = 64 = 4 \bmod 15$, $a^3 = 2 \bmod 15$, $a^4 = 1 \bmod 15$
4) Check
5) $a^2 - 1 = 3 \bmod 15$, $a^2 + 1 = 5 \bmod 15$

# What about the quantum part?

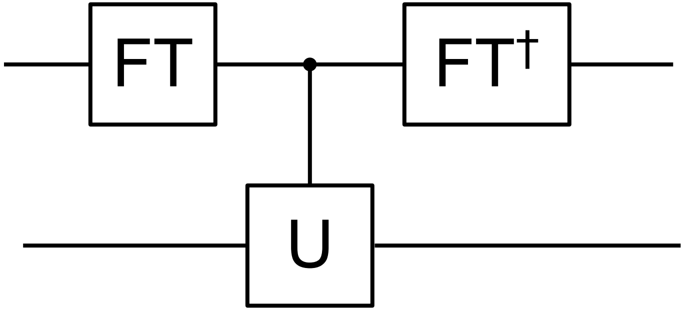
r is the period of the function

$$f(x) = a^x \text{mod } N$$

If we had a unitary that could do controlled multiplication,

$$|x\rangle|b\rangle \mapsto |x\rangle|ba^x\rangle$$

then we can do Phase Estimation like in this circuit (plus some additional post-processing).



# QFT: Quantum Fourier transform

The quantum Fourier transform is used in many quantum algorithms, notably Shor's algorithm for factoring and computing the discrete logarithm, the quantum phase estimation algorithm for estimating the eigenvalues of a unitary operator, and algorithms for the hidden subgroup problem.

# Grover’s Algorithm - Quadratic Speedup

Solves the ”needle in a haystack” problem --- how do you find a special item among many?

You know the special one when you see it, but there is no good way to hunt for it.

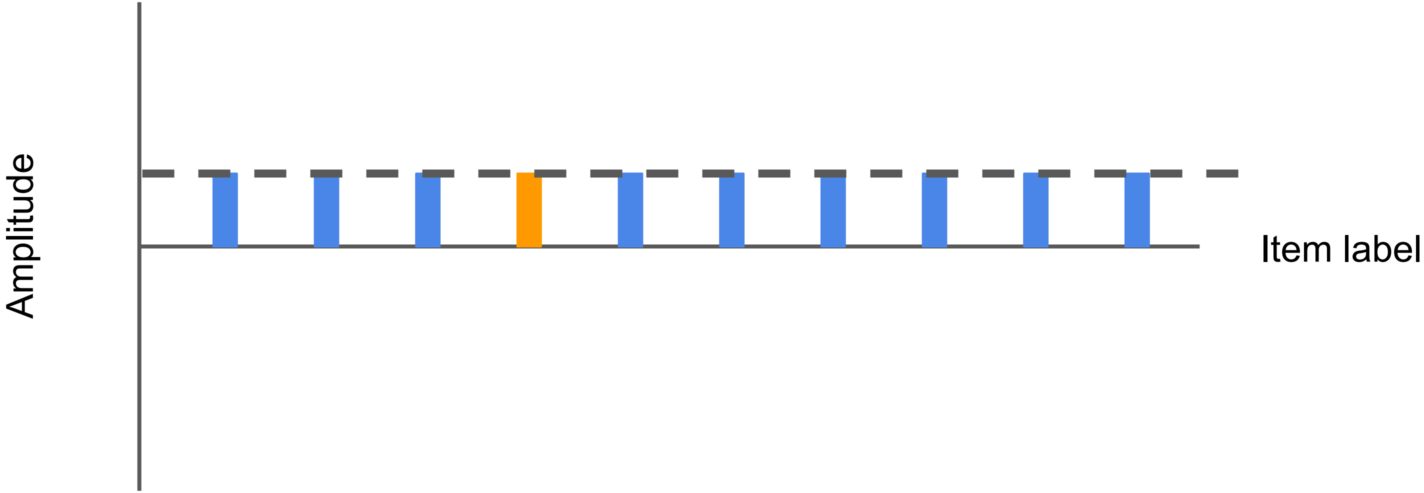
The best you can do classically is check each possibility one by one. Time O(N).

Grover’s algorithm takes time $O(N^{1/2})$.

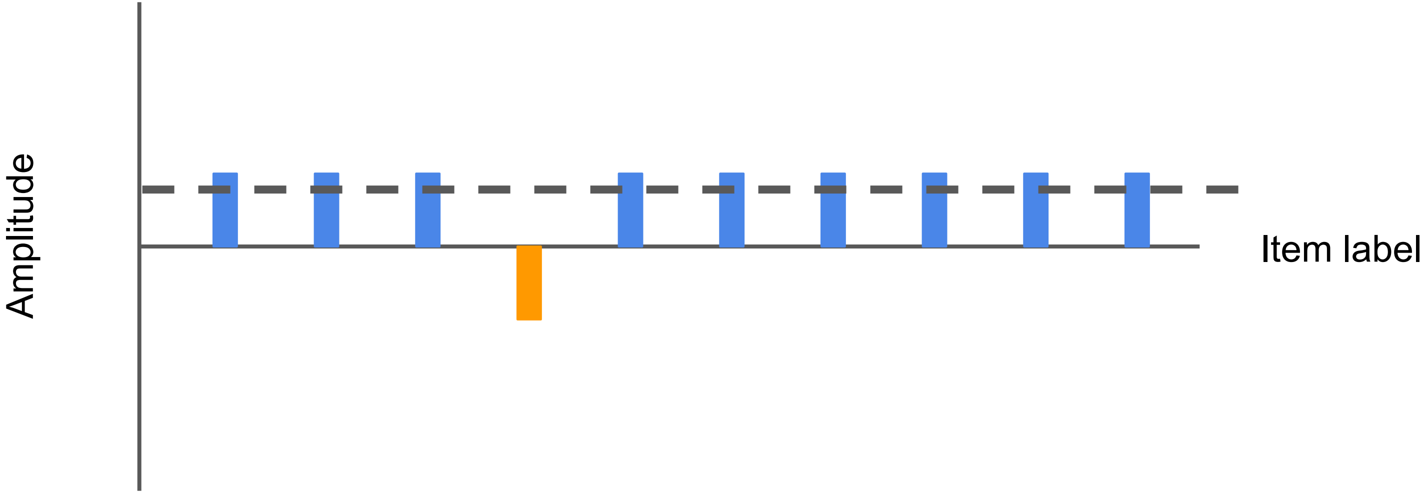
Grover's Algorithm on Wikipedia

# Grover’s Algorithm

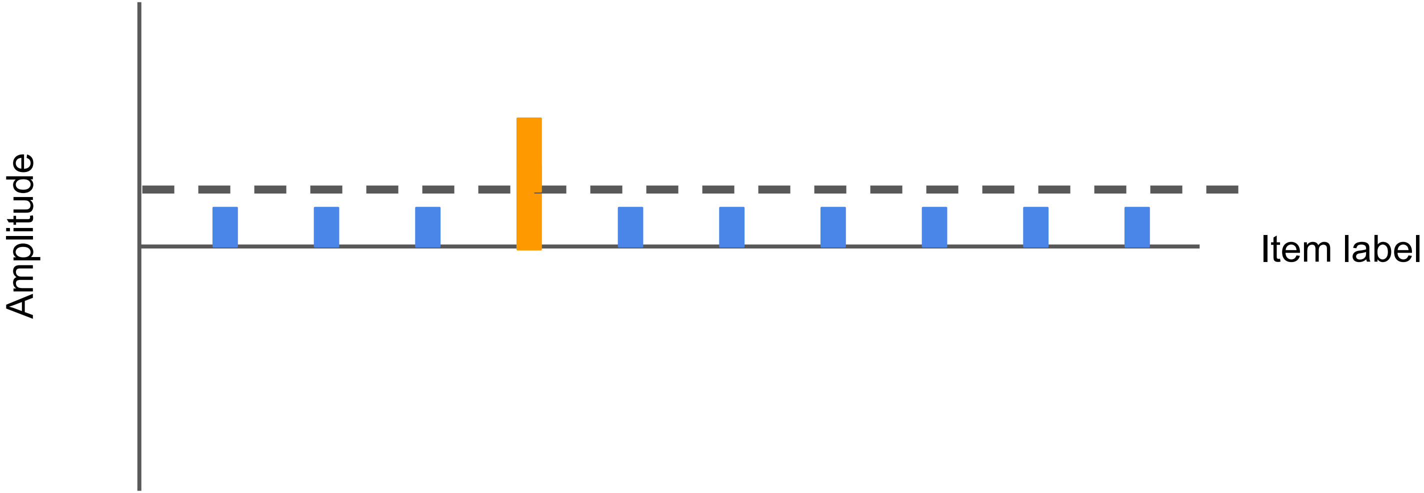
# Grover’s Algorithm



# Grover’s Algorithm



# Grover’s Algorithm



quantumai.google/cirq

# Break

(10 min.)

# NISQ Algorithms: QAOA

# Quantum Approximate Optimization Algorithm

Farhi, Goldstone, Gutmann (1411.4028)

Objective: Find a bit string with low energy (energy = cost)

More complicated energy functions will be more difficult in practice, but the recipe is general

Variational: there are parameters to tune (Quantum Machine Learning!)

Reminiscent of adiabatic theorem/algorithm, but different in practice

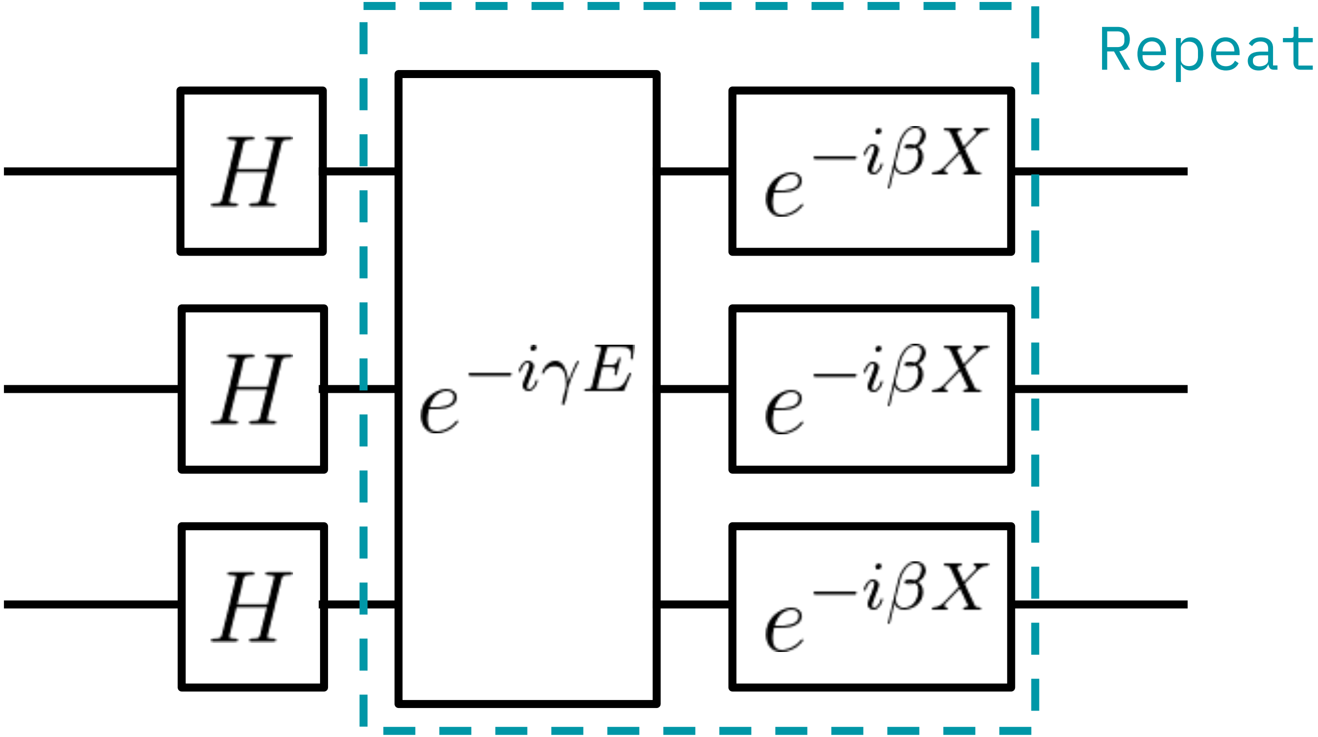
# Quantum Approximate Optimization Algorithm

$$|\psi\rangle = \exp(-i\beta_p X)\exp(-i\gamma_p E)\cdots\exp(-i\beta_1 X)\exp(-i\gamma_1 E)H^{\otimes n}|0\rangle$$

2p parameters to tune, try to get |ψ> to be a low energy state

Really: try to get |ψ> to have a large amplitude in low-energy states. You don’t actually care if |ψ> itself is low-energy.

# Quantum Approximate Optimization Algorithm

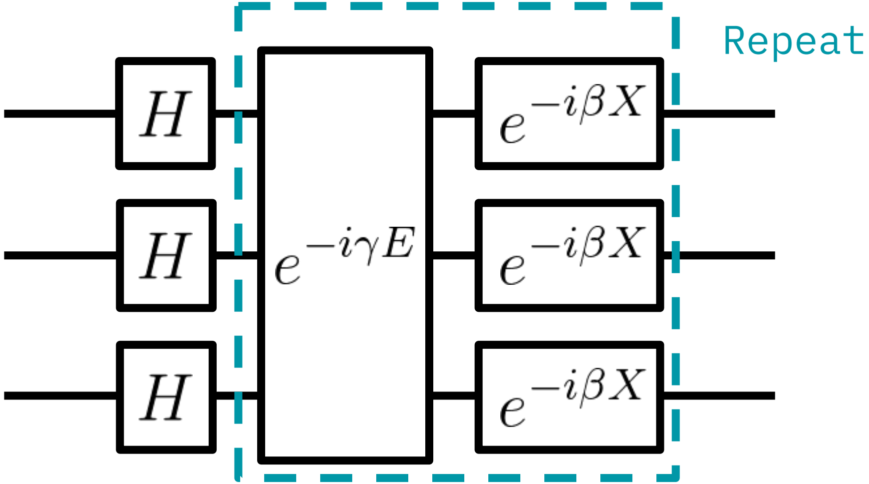


# How does the QAOA work?

Start with equal superposition over all bitstrings (H operations)

Add phases that depend on the energy of the bitstring

Mix the bitstrings around, creating constructive/destructive interference that depends on the energy



Thank you!