Quantum Computing Project

Exploring quantum computers and implementing a quantum program with IBMQ

Alejandro Hernandez

Computer Architecture - CS 3650

May 13, 2021

Your final assignment is to build a program that runs on a quantum computer. Then explain how a quantum computer works as if to a child and how your program utilizes the architecture.

1. First you need to understand the architecture of how a quantum computer works in comparison to how a classical computer works like the MIPS based system we learned in this class. GOAL: When trying something new first try to understand what it is you are working with conceptually before diving into coding guides.
2. Next look at the IBM tutorial . I recommend going through the entire youtube series to get started.
3. Build something new! Create a program that runs on the IBM Quantum computer.
4. Create your presentation for someone who does not know or understand quantum computing, think of a child.
    a. You should explain how quantum computing works 5-10 mins
    b. Explain what your program does and why you ran it on a Quantum computer. 10-15 mins
    c. Your demo and explanation should last ~20 min and you will play it in class for credit. I would recommend recording your presentation in advance so that we can get through them quickly.

    Presentations will start Thursday May 6th. NOTE: earlier presentations will be graded more leniently as my expectations will increase for the presentations and projects toward the end.

Classical Computers

- Bits

Quantum Computing

- Qubits

Demonstration

- Quantum program

The Bit

A **binary digit**, or **bit**, can be defined as the **simplest representation of information**. By representing information as bits, we are able to **execute computations** on that information. At the end of the day, **any sort of mathematical calculation can be explained and represented as a computation on bits.**

Binary Digit
0, 1

$$\begin{array}{r} 1 \\ +0 \\ \hline 01 \end{array}$$

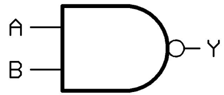
$$\begin{array}{r} 0\ 1\ 0\ 1 \\ +\ 1\ 0\ 1\ 0 \\ \hline 1\ 1\ 1\ 1 \end{array}$$

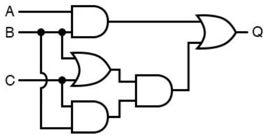
$$f(x) = \int_0^x f'(t)\, dt$$

The Bit

For classical computers, bit computations are actually represented and executed with logic gates, the **basic building blocks** of any digital system. With logic gates, more complex hardware can be engineered to create entire **computer architectures**.

Binary Digit
0, 1





With these complex tools, we can conduct complex computations so as to efficiently solve complex problems:

Complex Problem → Complex Method → Complex Tools → Domain Basics

What the heck is **Quantum Computing?**

Quantum Computing

"Quantum computing is the **exploitation** of collective **properties** of **quantum states**, such as **superposition** and **entanglement**, to **perform computation**. The devices that perform quantum computations are known as quantum computers."

- Wikipedia

The Qubit

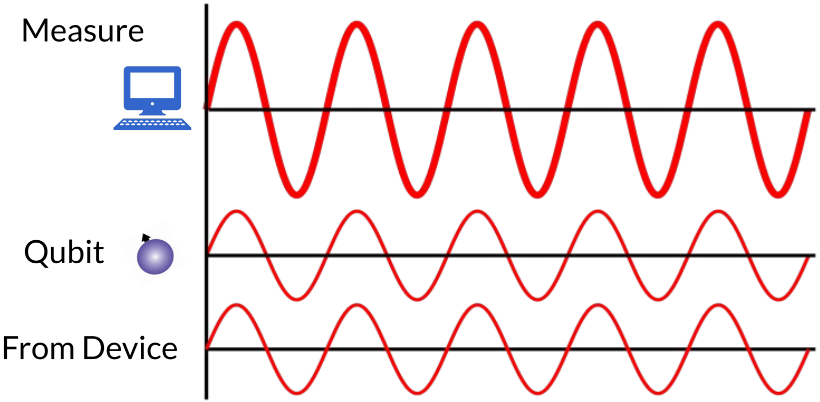
A quantum bit, or qubit, can be very difficult to understand! Let's break down its definition:

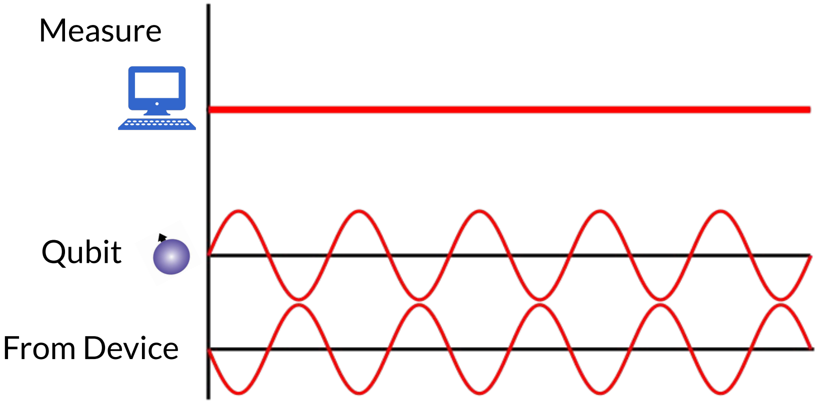
- **A qubit is the smallest unit of quantum information. It is comparable with a classical bit.**
  - In quantum computing, a qubit is the basic unit of quantum information—the quantum version of the classic binary bit physically realized with a two-state device.
- **A qubit has two states AND displays quantum properties.**
  - A qubit is a two-state (or two-level) quantum-mechanical system, one of the simplest quantum systems displaying the peculiarity of quantum mechanics. Examples include the spin of the electron in which the two levels can be taken as spin up and spin down; or the polarization of a single photon in which the two states can be taken to be the vertical polarization and the horizontal polarization.
- **A qubit's quantum properties are fundamental to quantum computing.**
  - However, quantum mechanics allows the qubit to be in a coherent superposition of both states simultaneously, a property that is fundamental to quantum mechanics and quantum computing.

Interference

"**Wave interference** occurs when two **waves** meet while traveling the same medium. The medium takes on a shape of the **net effect** of the **two individual waves**."

In quantum mechanics, inference can be used to **amplify** (**constructive interference**) or **reduce** (**destructive interference**) **the amplitude of desired waves**. This method works, in a way, to **select desired information** for our solution.





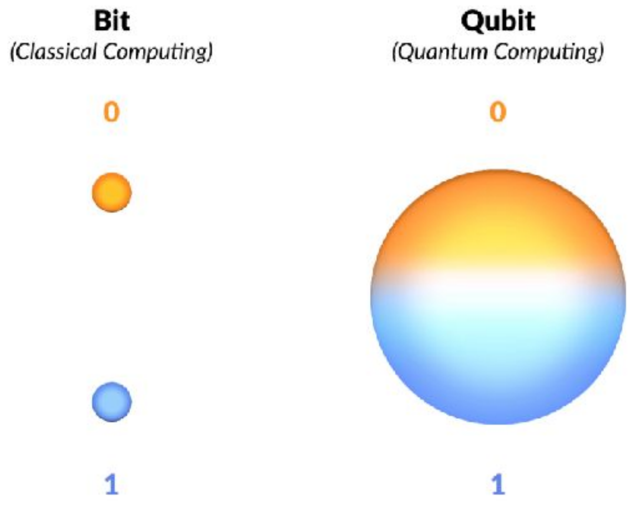
Superposition

Bits and qubits both represent 2 states. When we measure the state of classical bits, we are ALWAYS limited to two options: **the information the bit holds is either 1 or it's 0**. Unlike classical bits, that only have a single state at any time, **a qubit may be put into a superposition of both states**. This means that the probabilities of measuring 0 or 1 for a qubit's state are NOT CERTAIN. The information the qubit holds then is a sum of probabilities: A*0.0 + B*1.0, where A+B = 1.

ANALOGY: Spinning a Coin

An interesting property of a coin is that you can spin or flip it, then letting it land and measure its state as Heads or Tails. A coin can have **two states**, similar to a bit and qubit.

Imagine you have a coin and wanted to **measure its state**. At rest, that is easy. But what state is a **spinning coin** in? You can't definitely call it as Heads or Tails, but rather a **combination of the two states**!



Some Notes on Superposition:

A qubit will (similar to a coin) always measure to be in a 0 or 1 state, because...**we cannot measure superposition**!

In reality, when we measure a qubit that is in superposition, our method of measurement "collapses" the qubit's state into a measured 1 or a measured 0.

Interestingly though, a 1-state qubit put into superposition has a **different state** than a 0-state qubit put into superposition!

There are many other very interesting behaviors of qubits in superposition, but for this presentation, the high-level understanding will be enough.

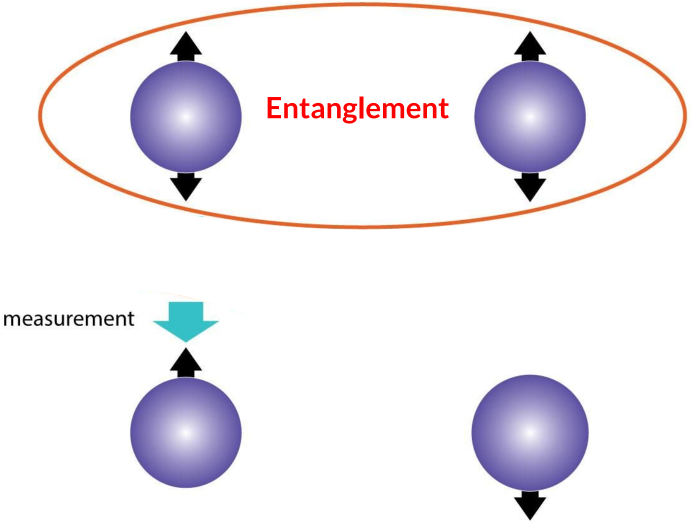
Entanglement

After the **phenomenon of entanglement**, the quantum states of qubits **can no longer be described independently of each other**.

Regardless of **space** or **time**, the state of one entangled qubit will always determine the state of other.

ANALOGY: Spinning 2 Coins
Imagine you have a friend. If you and your pal were to spin two coins enough times, you'd see that there's **no correlation** between one coin's measured state with the other.

However, imagine you were able to **entangle** your two coins. If you spin the coins again, based only on the measured state of one coin, **you would know exactly the state of the other coin**.

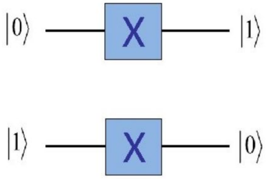


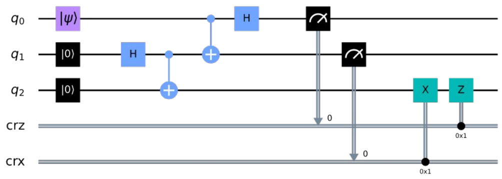
Performing Computations with Qubits

Linear algebra is a great medium for representing quantum computations. While I won't go much into representing these computations with linear algebra in this this presentation, **understanding the math gives greater understanding of the computations**, which makes all of quantum computing much more digestible as things get more complicated.

Note that, similar to classical computing, **we can represent and execute qubit computations with quantum gates**. Additionally, we can also create **circuit diagrams** to demonstrate quantum computations and, eventually, quantum algorithms!

Qubits
$\|0\rangle, \ \|1\rangle$





What you might see as you dive deeper into quantum computing, is that compared to classical computing, there are both striking similarities and alarming differences.

Creating a Program to run on a **Quantum Computer**

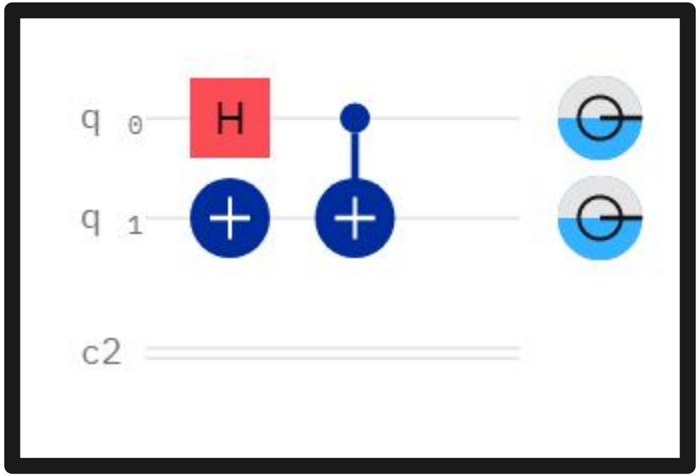
Demonstrating **Quantum Entanglement** with IBMQ

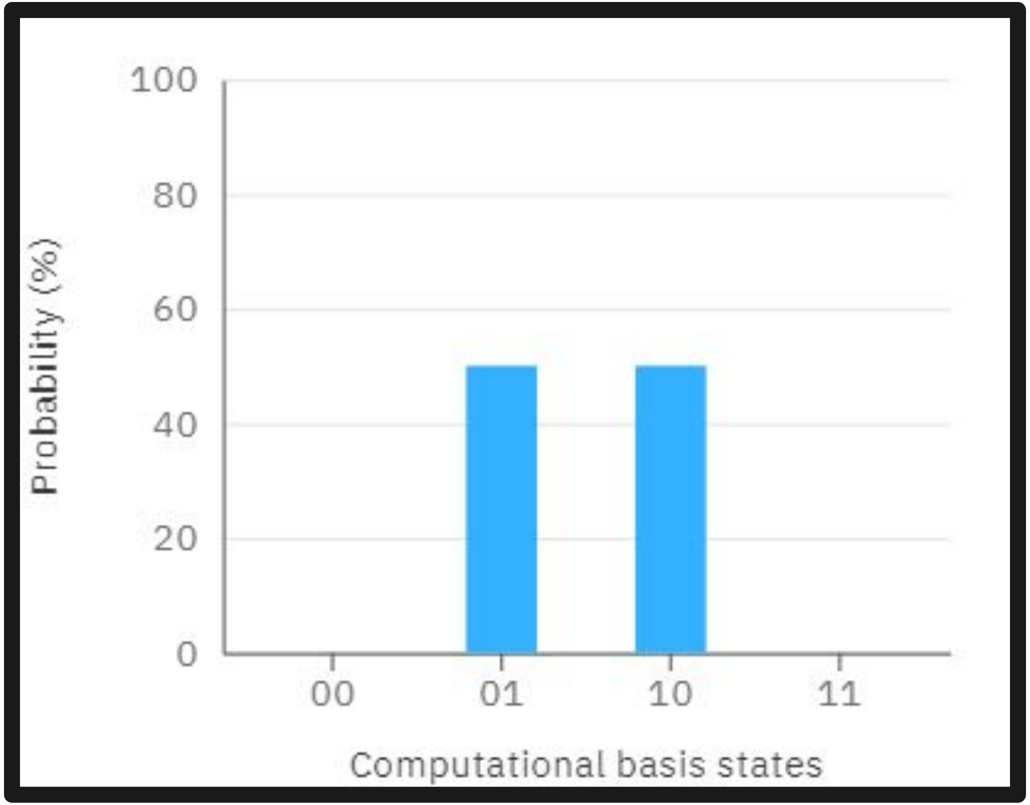
C-Not Gate
Jupyter Notebook

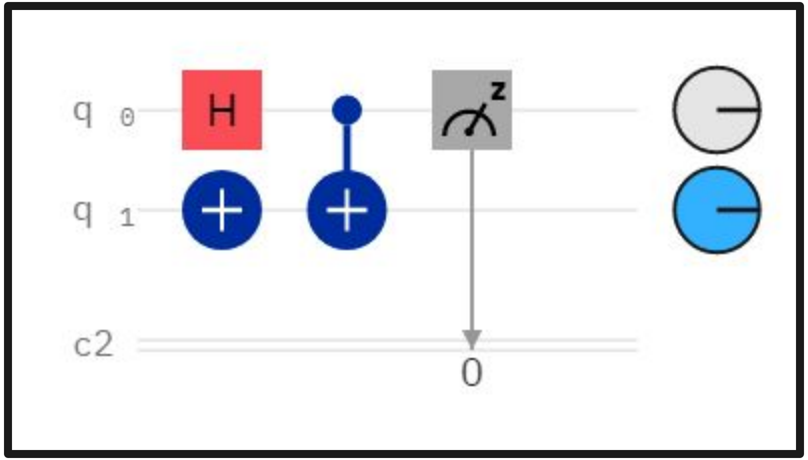
Example Program in QASM (Quantum Assembly)

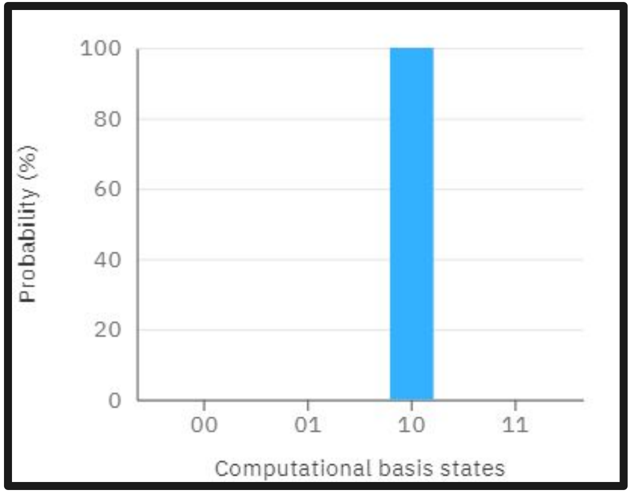
OpenQASM 2.0

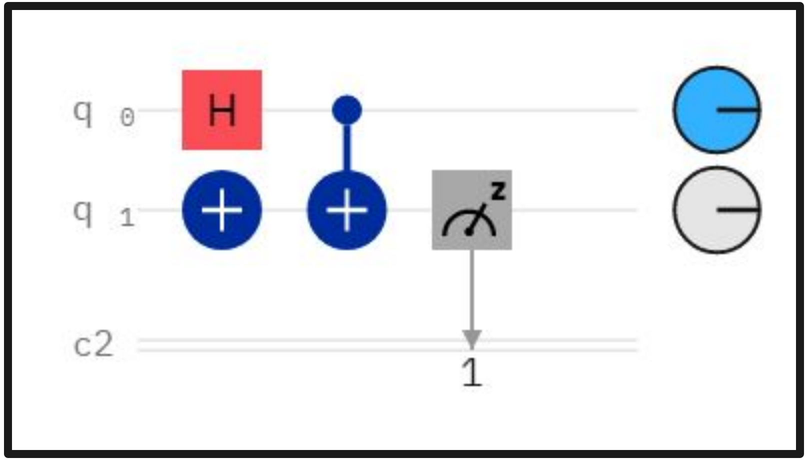
```
OPENQASM 2.0;
include "qelib1.inc";
qreg q[2]; //Q1 & Q0 states = 0
creg c[2];
h q[0]; //Q0 is in superposition (two states)
x q[1]; //Q1 state = 1
cx q[0],q[1]; //C:Q0 & T:Q1. Because Q0 is in superposition, the qubits are entangled!

// Recall that the CX gate means the state of Q0 will determine the state of Q1. When we measure either qubit, we will reveal both that state AND the other qubit's state!

// MEASURING Q0:
//  If Q0 is in 0 state, that meant the CX gate left Q1 alone. So we know Q1 state is 1...Final Output [01].
//  If Q0 is in state 1, that meant the CX gate inverted the state of Q0...Final Output: [10].
```

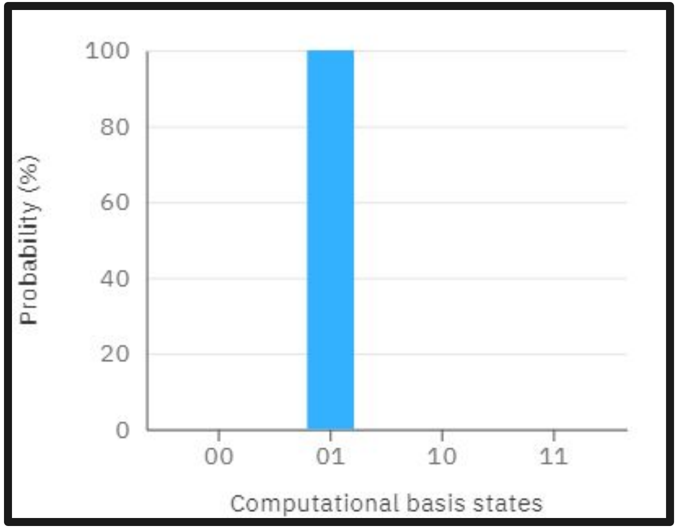

q 0
q 1
H
c2


Probability (%)
100
80
60
40
20
0
00
01
10
11
Computational basis states


q 0
H
z
q 1
c2
0


100
80
60
40
20
0
Probability (%)
00
01
10
11
Computational basis states


q 0
H
z
q 1
c2
1


100
80
60
40
20
0
Probability (%)
00
01
10
11
Computational basis states

Demonstrating **Quantum Phase Estimation** with IBMQ

[T-Gate]
[Jupyter Notebook]

What is a Quantum Phase Estimation?

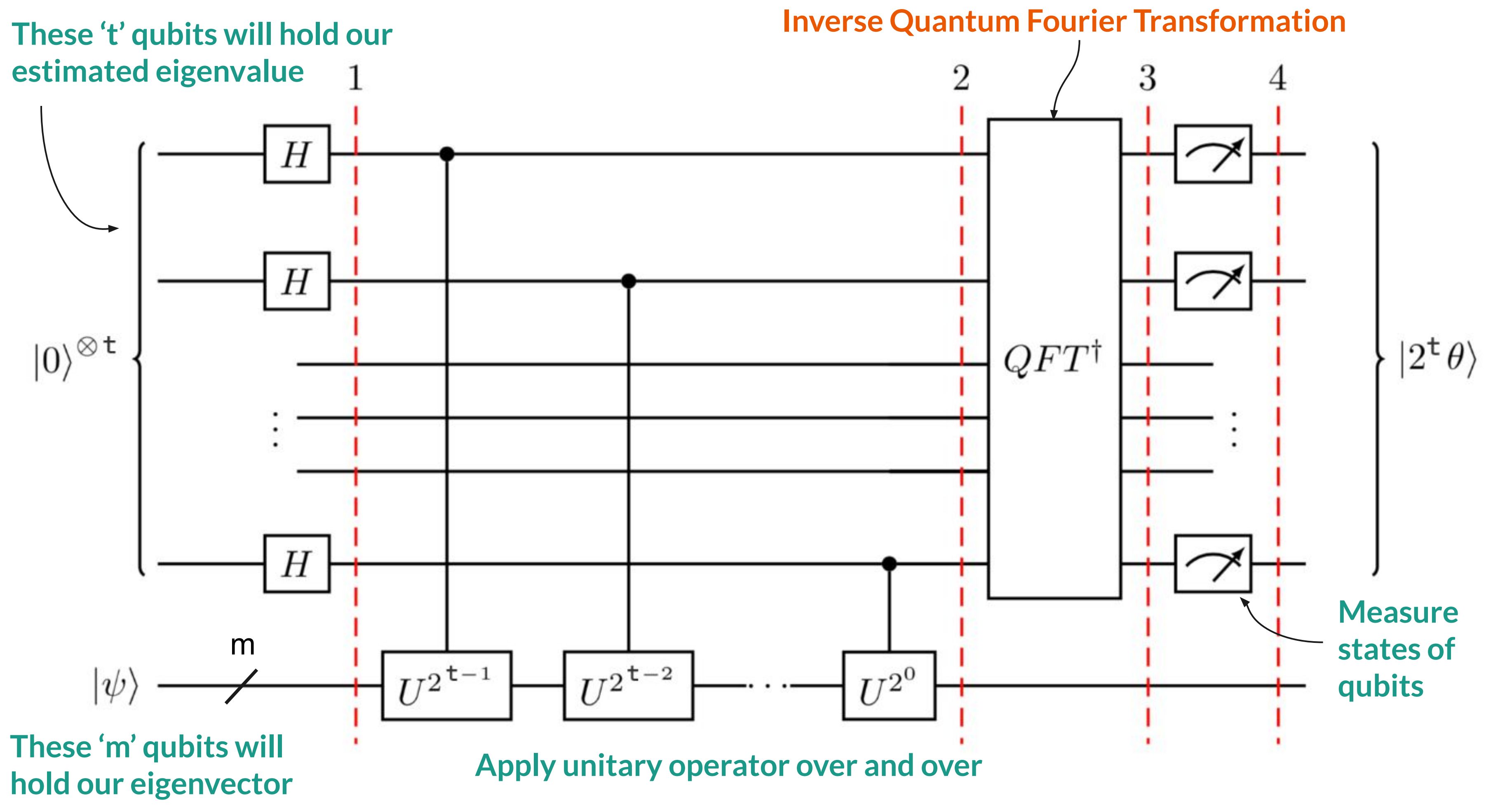
Unitary Operation: For matrices U, A, and B: If $UA = B$ and $U^TB = A$, then U is a unitary operation.
For any unitary operation U, $UU^T = I$ or $UU^TA = A$.

Most quantum gates can be understood as unitary operations! One such gate is the **T-gate**. Put simply, this gate performs a **rotation of π/4 around the Z-axis** direction. In matrix form, it looks and functions with |1⟩as so:

$$T = \begin{bmatrix} 1 & 0 \\ 0 & e^{\frac{i\pi}{4}} \end{bmatrix}, \quad T^{\dagger} = \begin{bmatrix} 1 & 0 \\ 0 & e^{-\frac{i\pi}{4}} \end{bmatrix} \qquad T|1\rangle = \begin{bmatrix} 1 & 0 \\ 0 & e^{\frac{i\pi}{4}} \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = e^{\frac{i\pi}{4}}|1\rangle$$

Quantum Phase Estimation (QPE) is one of the most important subroutines in quantum computation. It serves as a central **building block** for many quantum algorithms. The objective of the algorithm is the following:

Given a unitary operator U, the algorithm estimates θ in $U|\psi\rangle = e^{\boldsymbol{2\pi i\theta}}|\psi\rangle$. Here, $|\psi\rangle$ is the eigenvector and $e^{\boldsymbol{2\pi i\theta}}$ is the corresponding eigenvalue.


These 't' qubits will hold our estimated eigenvalue
Inverse Quantum Fourier Transformation
1
2
3
4
H
H
H
$|0\rangle^{\otimes t}$
$QFT^{\dagger}$
$|2^t\theta\rangle$
m
$|\psi\rangle$
$U^{2^{t-1}}$
$U^{2^{t-2}}$
U^{2^0}
Measure states of qubits
These 'm' qubits will hold our eigenvector
Apply unitary operator over and over

Quantum Phase Estimation of the T-Gate

[1] $U|\psi\rangle = e^{\boldsymbol{2\pi i\theta}}|\psi\rangle$

[2] $T|1\rangle = e^{2i\pi\theta}|1\rangle$

[3] $T|1\rangle = \begin{bmatrix} 1 & 0 \\ 0 & e^{\frac{i\pi}{4}} \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = e^{\frac{i\pi}{4}}|1\rangle \Rightarrow \theta = \frac{1}{8}$

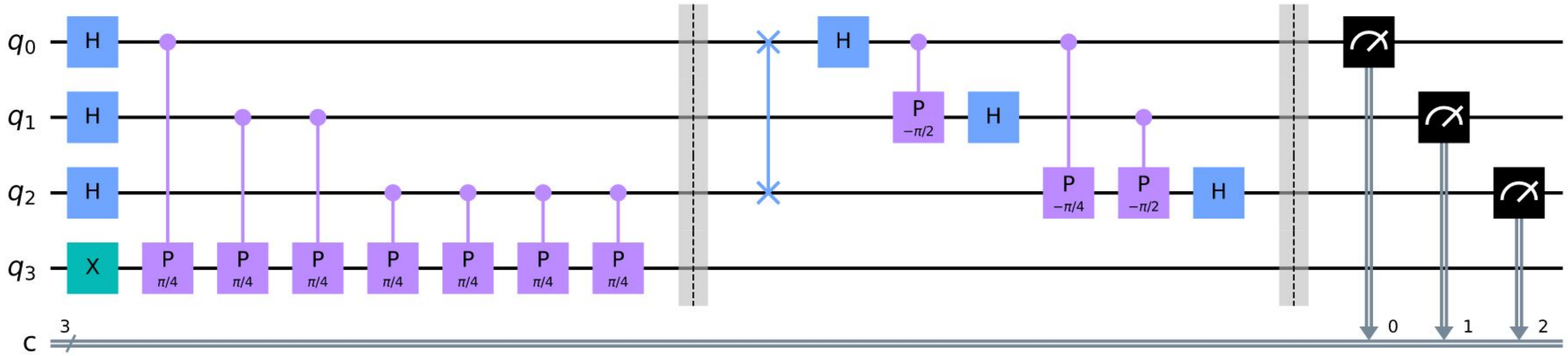
Recall that QPE estimates θ with [1]. We can apply this to the T-gate (a unitary operator) in [2]. Also recall the result with a 1-state qubit, shown in [3]. Solving for θ in [2] , we find θ = ⅛.

Now that we've seen Quantum Phase Estimation with the T-gate, let's examine this process on a circuit diagram and a subsequent quantum program!

QPE of the T-Gate: Circuit Diagram

These 3 qubits will hold our estimated eigenvalue

Inverse Quantum Fourier Transformation



Fourth qubit holds eigenvector

Apply unitary operations over and over

Measure states of qubits onto classical bits

Demonstration on Qiskit

What's the point?

If you're wondering, "This seems kinda pointless: **we have to know the unitary operator to perform the operations on the estimated qubits**. How is this helpful?"

QPE is an integral component for much more complex and larger quantum algorithms!

If fact, **it is possible to create circuits/quantum computations/unitary operations for which we don't know the eigenvalues!** And, if we learn the eigenvalues of the operation, it can allow us to do very impressive computations!

Shor's Algorithm (the pride and joy of quantum computing) requires number factorization- QPE is integral to that process!

Quantum Computers: Advantages

Our system is quantum in nature! The particles are quantum in nature. **They handle the true complexity**. And we'd expect these "basics" to be complex!

Entanglement: between 2 superposition particles; state of one determines state of the other.
Can store AND manipulate much more information.

Great at small input & output, with MANY possibilities.

- **EXAMPLES**: Prime factors of large numbers ⇒ Shor's Algorithm, Longest distance between A and B (NP-Complete), Gene-analysis, AI training, chemical reactions (drug building), stock market behavior. Finding patterns!

LINK: 1.5 The Case for Quantum

Quantum Computers: Shortcomings

We don't actually KNOW the states of qubits, unlike classical computer's bits. Why? Because we actually can't measure superposition! Observing states of qubits collapses them into a single state and OUT of superposition (which is their true complexity). We only know probability. And we can't just throw more qubits at the problem! With more and more qubits, we multiply the uncertainty of our output and increase the instability of our computer.

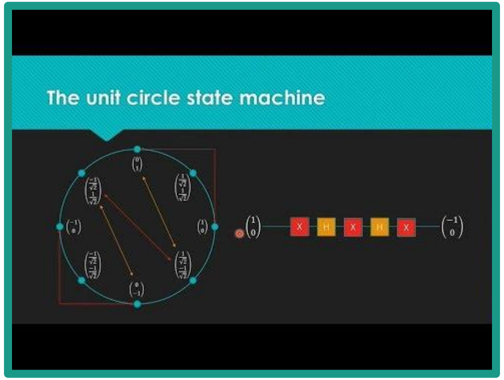
We don't get a trillion possible answers. We get one.

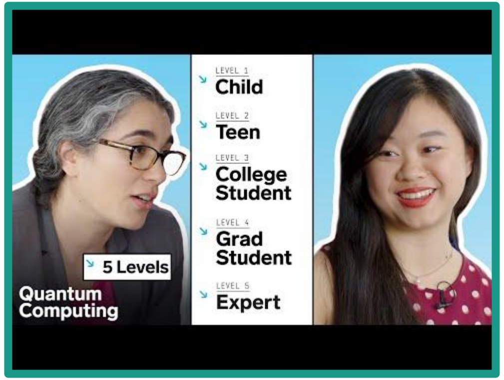
Maintaining a quantum environment is HARD: need isolation and cold.

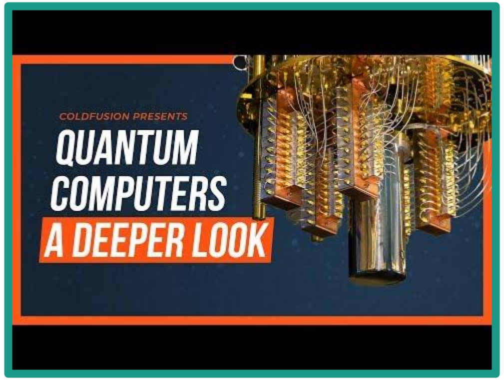
Only faster for utilization of superposition- it's not like speeding up each individual operation, it's like cutting down on number of operations necessary

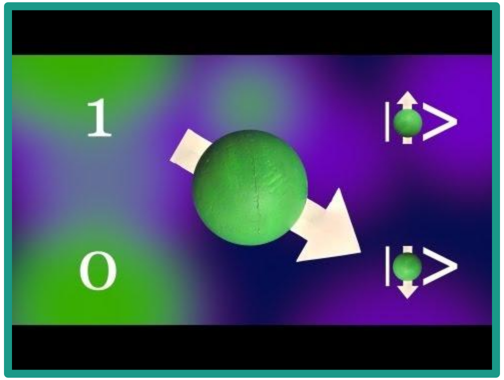
Decoherence: our qubits don't hold information (aka their state) forever! We may have a time limit some information as after a while it will disappear!

Additional Resources & Materials







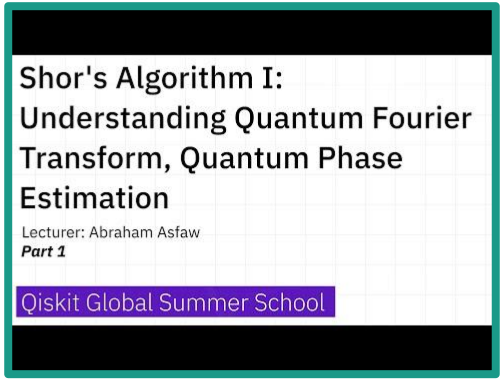


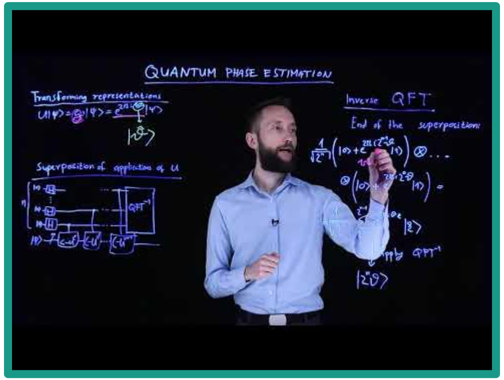
Demystifying Quantum Gates

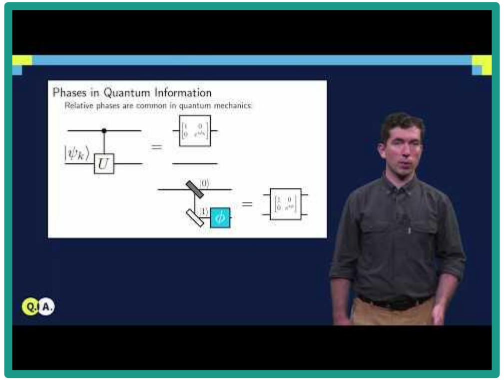
Quantum Phase Estimation

Eigenvectors/Eigenvalues

Additional Resources & Materials







[GitHub Repository](#) GitHub

[Learn with Qiskit](#)

Thank you!