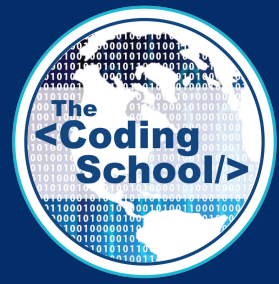


INTRO TO QUANTUM COMPUTING

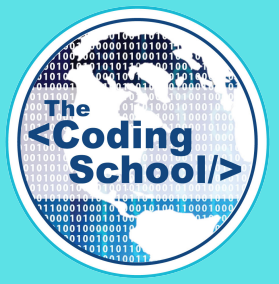
LECTURE #15

QUANTUM COMPUTATION PT. 1 :

THE QUANTUM CIRCUIT MODEL

FRANCISCA VASCONCELOS

2/14/2021



ANNOUNCEMENTS

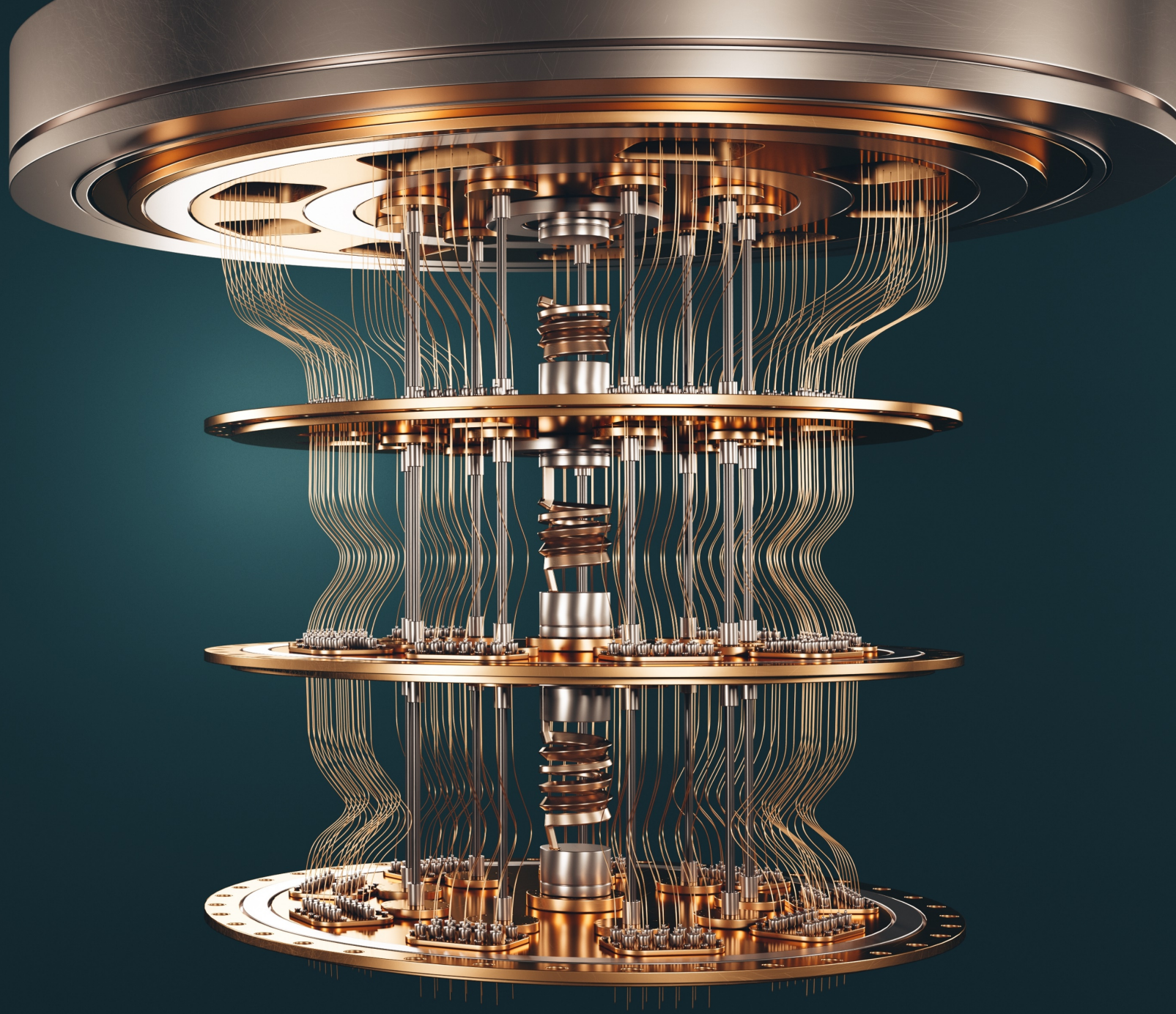
Diversity in {Quantum} Computing Conference

SATURDAY, FEBRUARY 27TH

A free virtual conference for high school students, young professionals, and all who are passionate about ensuring the future of quantum computing is diverse and inclusive.

REGISTER NOW

QubitByQubit.org/conference



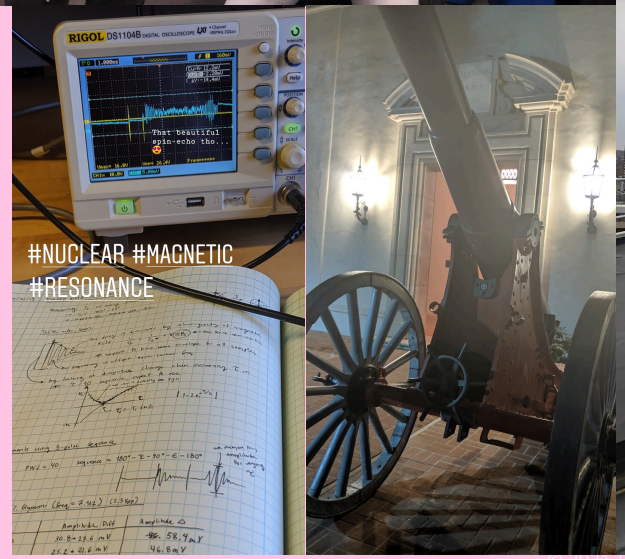
Happy Valentine's Day, QuBaes!

<3|QxQ|ε>

-- with love from the QUBIT X QUBIT team

Belated International Women in Science Day

Fran & Ghada – the OG "QuBaes"



QUANTUM COMPUTATION LECTURE SERIES

Lecture 1 – The Quantum Circuit Model *How can we perform computation with quantum systems?*	CONCEPTS
Lecture 2 – Qiskit Tutorial *How can we program quantum circuits?*	PROGRAMMING
Lecture 3 – Quantum Circuit Mathematics *How can we represent quantum circuits mathematically?*	MATH
Lectures 4-6 – Introductory Quantum Protocols and Algorithms *How can we leverage quantum for cryptography, teleportation, and algorithms?*	APPLICATION

TODAY'S LECTURE

1. What is Quantum Computing?

2. Classical vs Quantum Circuits

    a) Classical vs Quantum States

    b) Classical vs Quantum Gates

    c) Classical vs Quantum Measurement

    d) Quantum Circuits

TODAY'S KEY IDEAS

The quantum circuit is a theoretical model for quantum computation.

It has three key components: states, gates, and measurements.

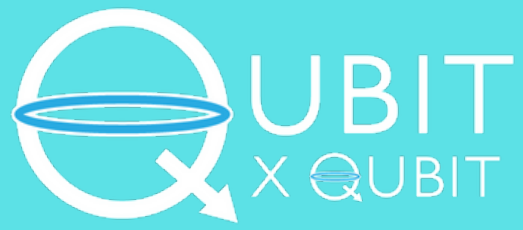
Theoretically, our quantum states simply boil down to: $\alpha|0\rangle + \beta|1\rangle = \begin{pmatrix}\alpha\\ \beta\end{pmatrix}$

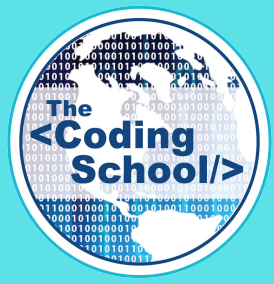
Multiple quantum states are represented as qubit strings: $|0110101\rangle$

Quantum operations (matrices) are the gates of quantum computation.

Unlike deterministic classical measurements, quantum measurements are probabilistic!

If you want to know the underlying distribution of a quantum state, you need to create the state several times and perform repeated measurements!

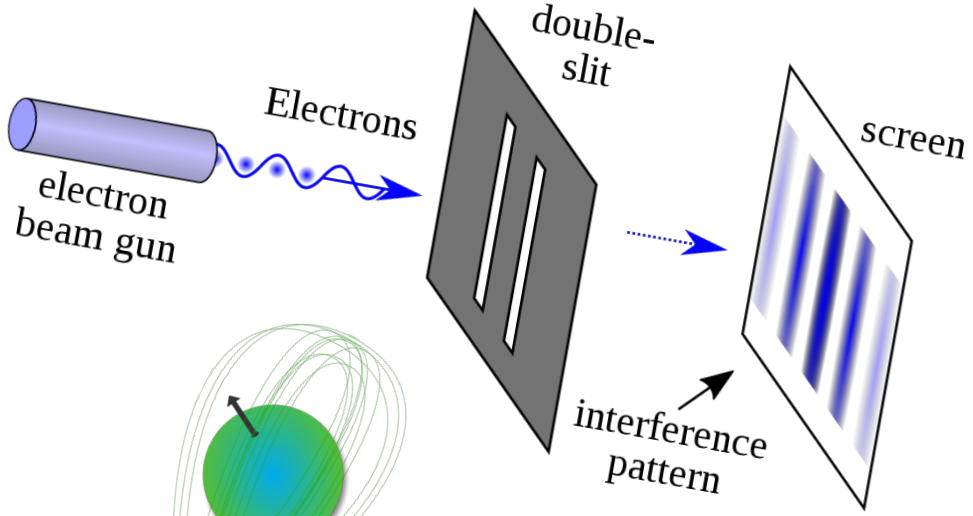


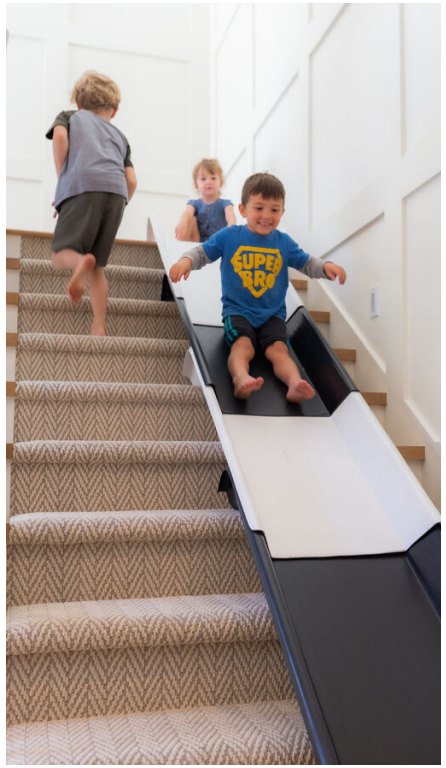


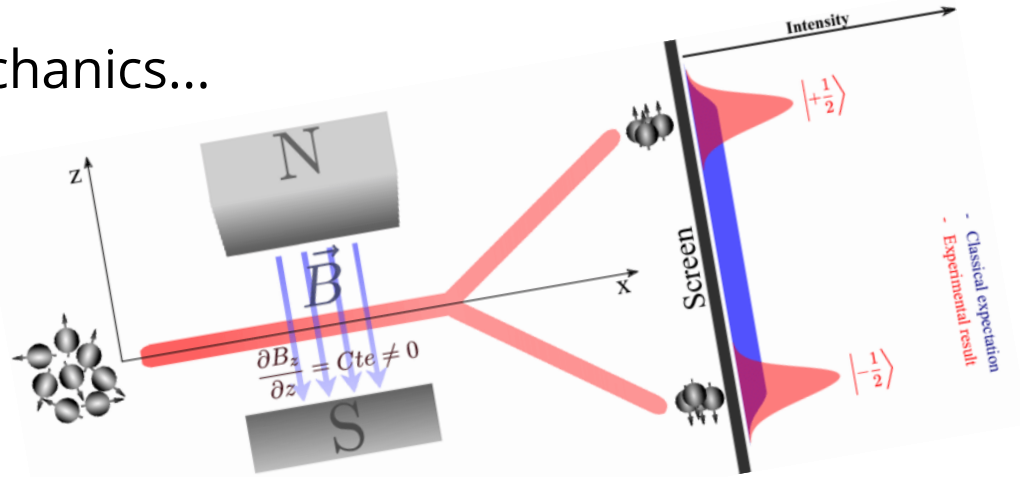
WHAT IS QUANTUM COMPUTING?

QUANTUM MECHANICS

The last few lectures, Amir taught you about quantum mechanics...

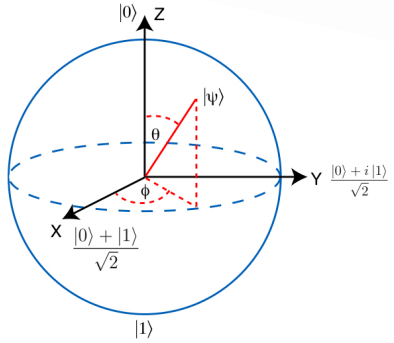






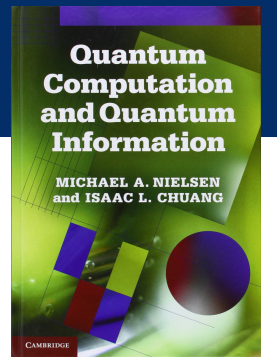
$$H(t)|\psi(t)\rangle = i\hbar\frac{\partial}{\partial t}|\psi(t)\rangle$$

$$\Delta\chi\Delta\rho \geq \frac{\hbar}{2}$$



... However, in today's lecture we are going to focus on ***quantum computation***.

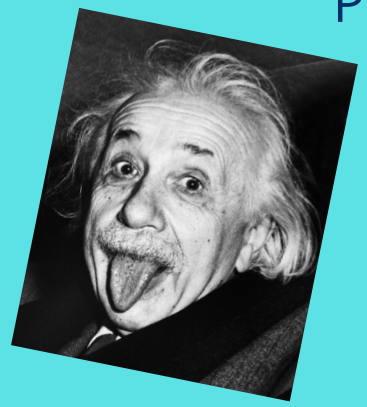
WHY QUANTUM COMPUTING?

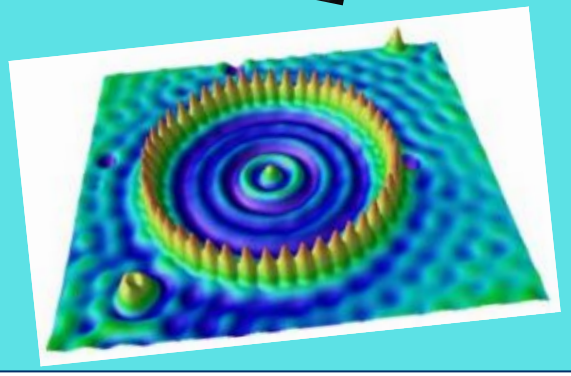


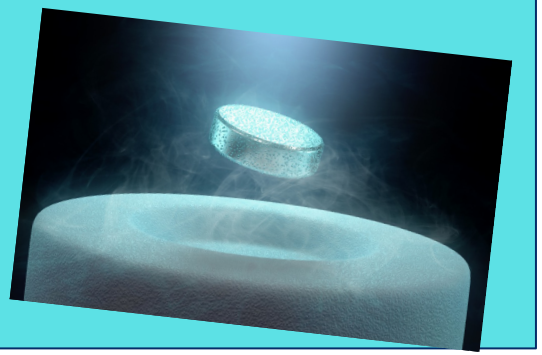
A BRIEF (2-SIDED) HISTORY

as described in Nielson and Chuang's "Introduction to Quantum Computation and Quantum Information"

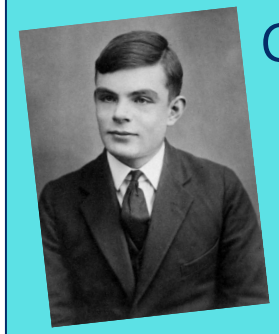
PHYSICS

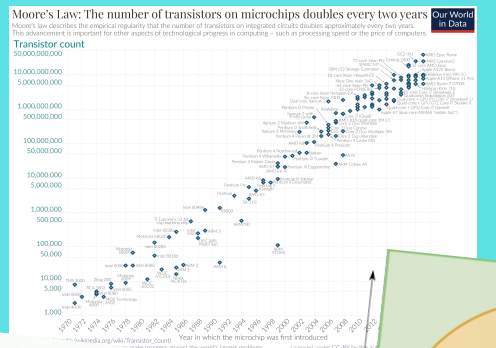


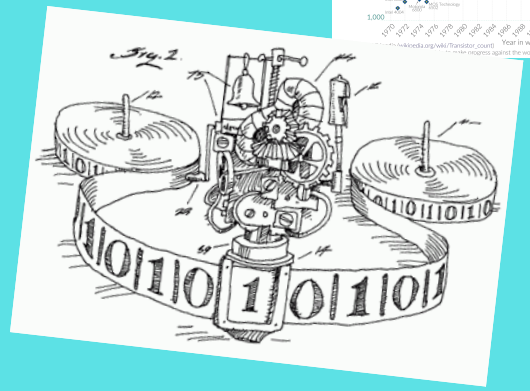


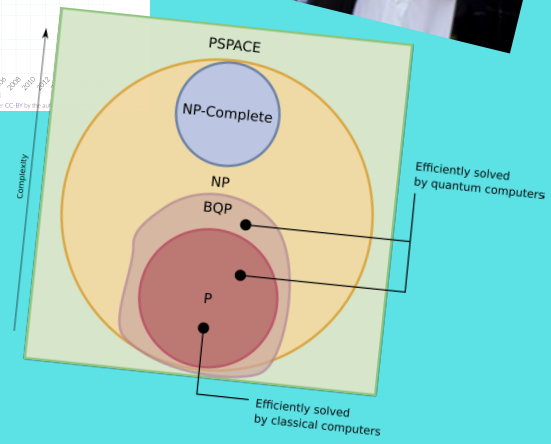


COMPUTER SCIENCE









WHY QUANTUM COMPUTING?

"One of the goals of quantum computation and quantum information is to develop tools which sharpen our intuition about quantum mechanics, and make its predictions more transparent to human minds."

"In the early 1980s, interest arose in whether it might be possible to use quantum effects to signal faster than light – a big no-no according to Einstein's theory of relativity...

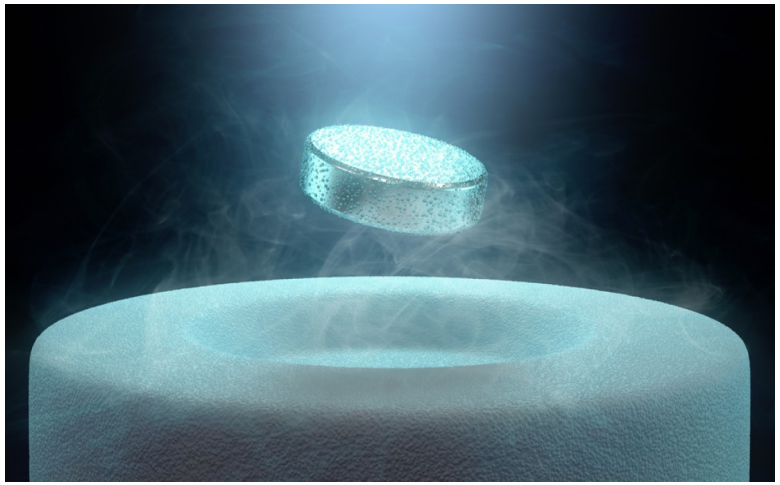
If cloning were possible, then it would be possible to signal faster than light using quantum effects.

However, cloning – so easy to accomplish with classical information – turns out not to be possible in general in quantum mechanics. This ***no-cloning theorem****, discovered in the early 1980s, is one of the earliest results of quantum computation and quantum information."*

WHY QUANTUM COMPUTING?

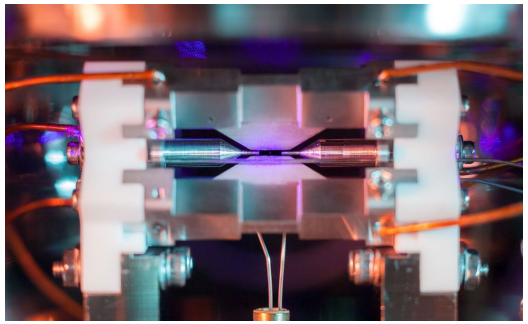
"A related historical strand contributing to the development of quantum computation and quantum information is the interest, dating to the 1970s, of obtaining complete control over single quantum systems."

"Applications of quantum mechanics prior to the 1970s typically involved a gross level of control over a bulk sample containing an enormous number of quantum mechanical systems, none of them directly accessible."

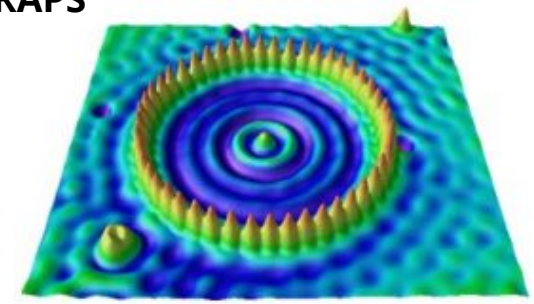


SUPERCONDUCTOR

"Since the 1970s many techniques for controlling single quantum systems have been developed."



ATOM-TRAPS



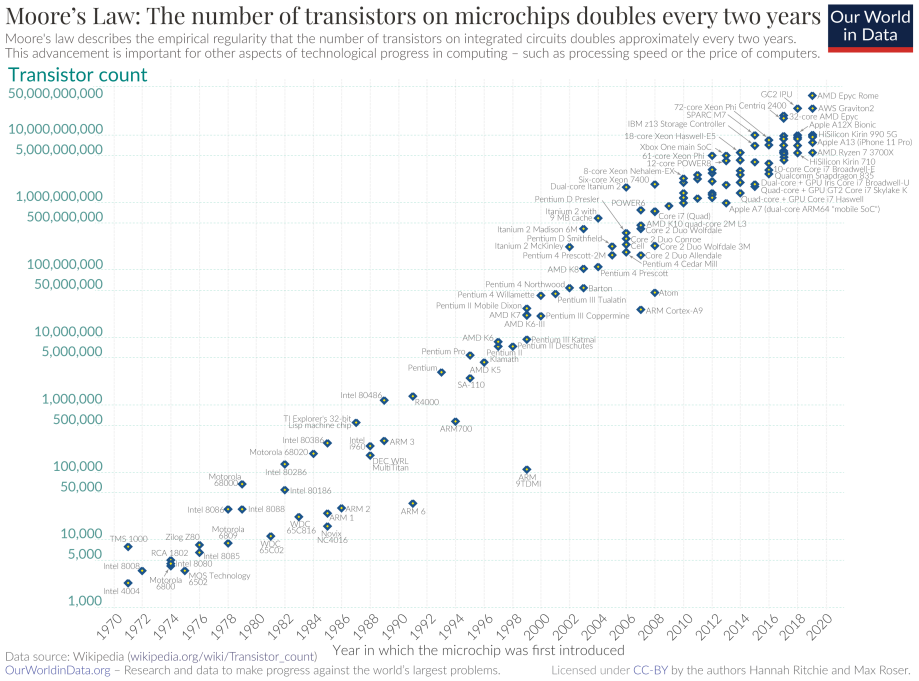
SCANNING TUNNELING MICROSCOPE

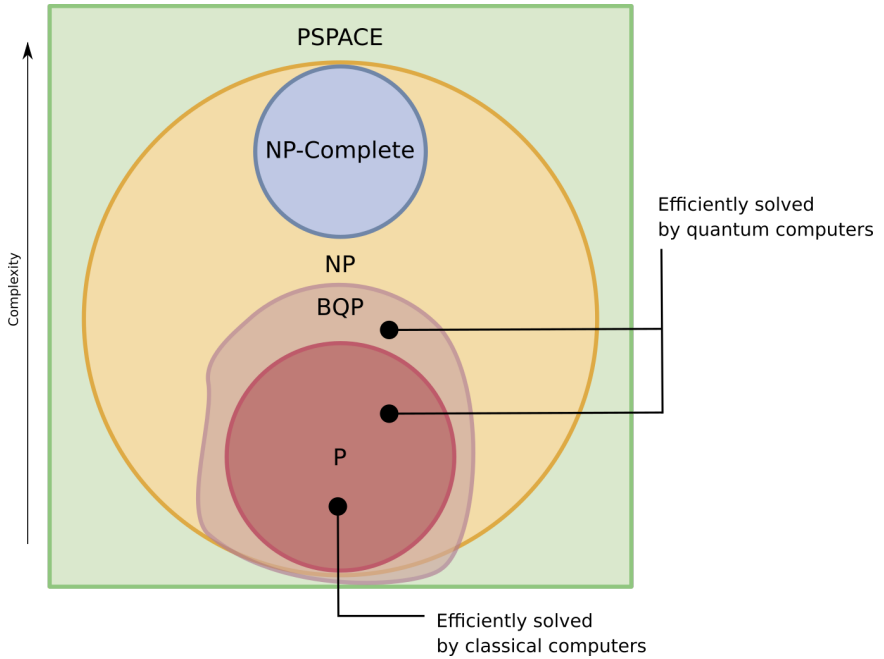
"Quantum computation and quantum information provide a useful series of challenges at varied levels of difficulty for people devising methods to better manipulate single quantum systems, and stimulate the development of new experimental techniques and provide guidance as to the most interesting directions in which to take experiment.

Conversely, the ability to control single quantum systems is essential if we are to harness the power of quantum mechanics for applications to quantum computation and quantum information."

WHY QUANTUM COMPUTING?

"One possible solution to the problem posed by the eventual failure of Moore's law is to move to a different computing paradigm. *One such paradigm is provided by the theory of quantum computation, which is based on the idea of using quantum mechanics to perform computations, instead of classical physics.*

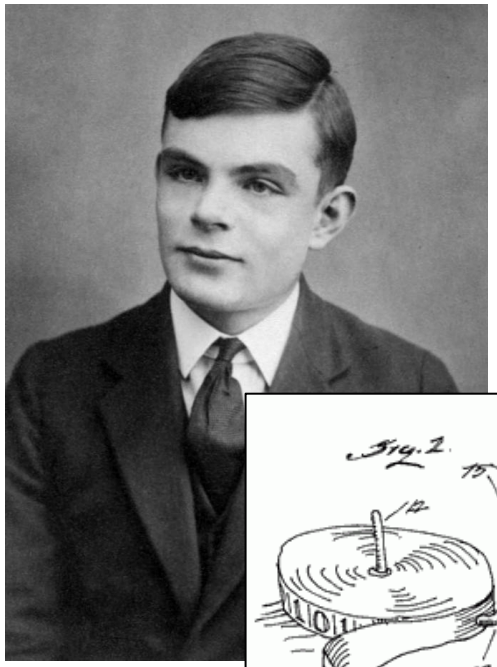


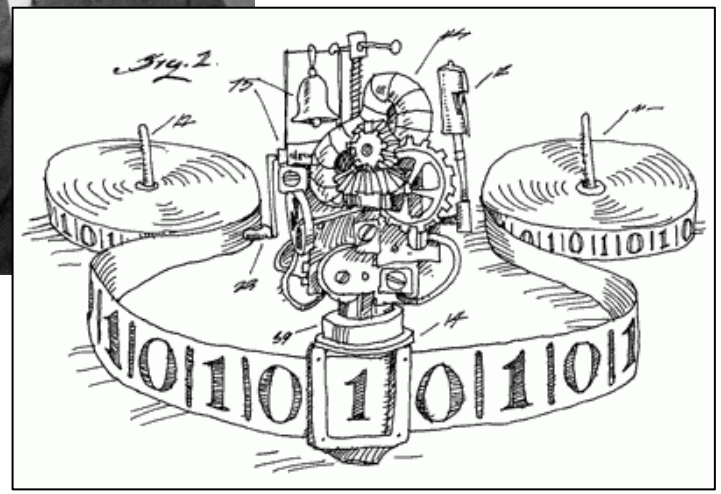


It turns out that while an ordinary computer can be used to simulate a quantum computer, it appears to be impossible to perform the simulation in an efficient fashion. ***Thus quantum computers offer an essential speed advantage over classical computers****. This speed advantage is so significant that many researchers believe that no conceivable amount of progress in classical computation would be able to overcome the gap between the power of a classical computer and the power of a quantum computer."*

WHY QUANTUM COMPUTING?

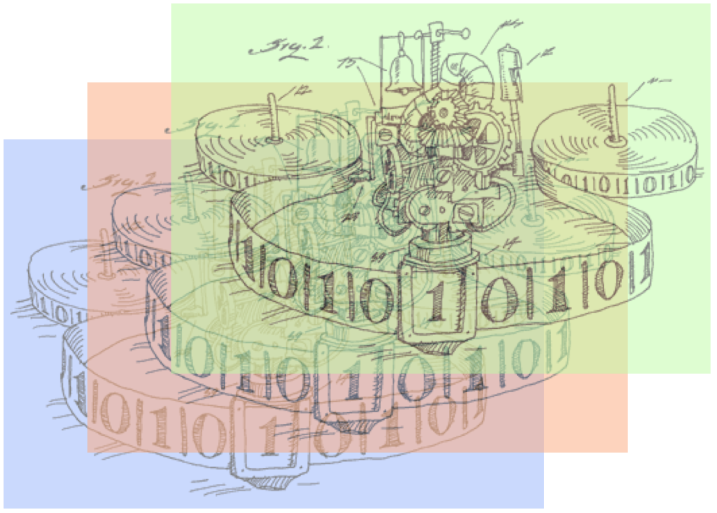
Strengthened Church-Turing Thesis:
Any algorithmic process can be simulated efficiently using a probabilistic Turing machine.





"Motivated by this question, in 1985 David Deutsch asked whether the laws of physics could be use to derive an even stronger version of the Church–Turing thesis… In particular, Deutsch attempted to define a computational device that would be capable of efficiently simulating an arbitrary physical system.

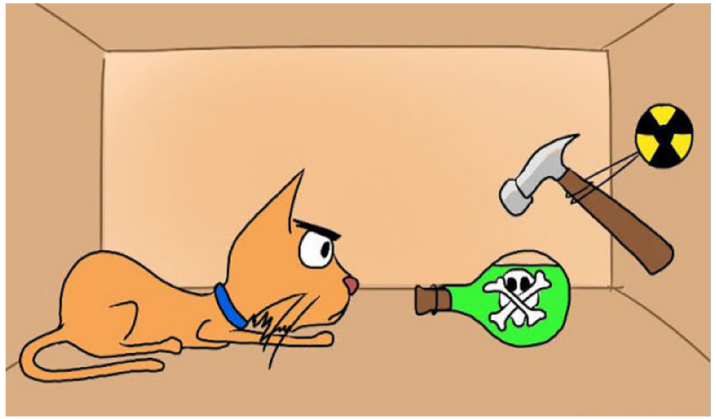
Because the laws of physics are ultimately quantum mechanical, Deutsch was naturally led to consider computing devices based upon the principles of quantum mechanics. ***These devices, quantum analogues of the machines defined forty-nine years earlier by Turing, led ultimately to the modern conception of a quantum computer.****"*



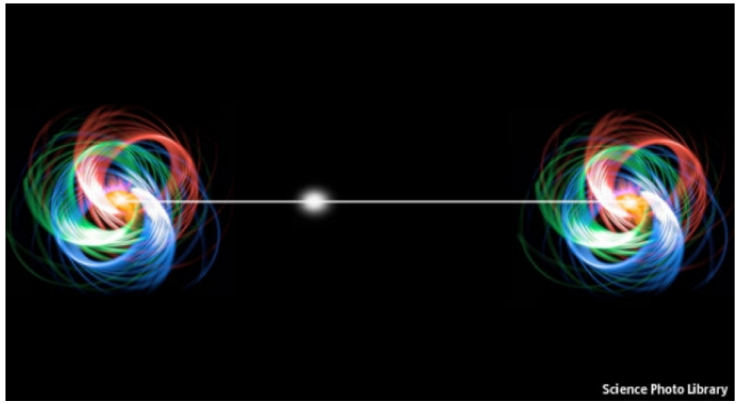
SO, WHAT IS QUANTUM COMPUTING?

Quantum computing is the use of quantum phenomena:

SUPERPOSITION



ENTANGLEMENT



QUANTUM INTERFERENCE

to perform computation (ideally more efficiently than classical computers...) !

CLASSICAL VS QUANTUM CIRCUITS

THE QUANTUM CIRCUIT MODEL

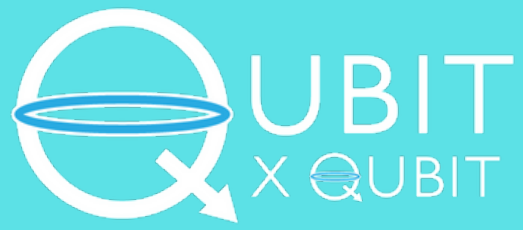
The quantum circuit is a theoretical model for quantum computation. It has three key components:

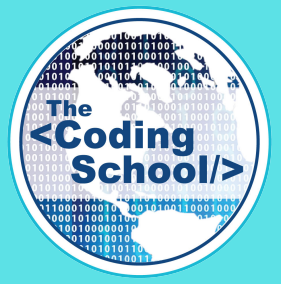
STATES,

GATES,

& MEASUREMENT

Let’s break this down, through analogy to classical computation...





CLASSICAL VS QUANTUM CIRCUITS

STATES

CLASSICAL COMPUTATION – WHY 0s & 1s?

Last semester, we learned about binary representation and Boolean logic.

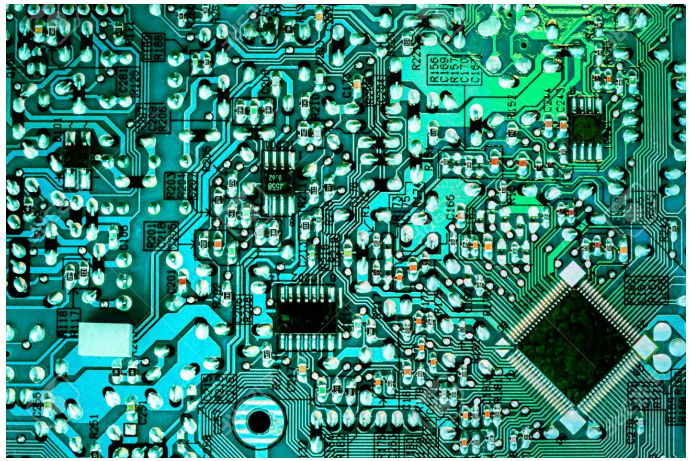
These, however, are abstractions...

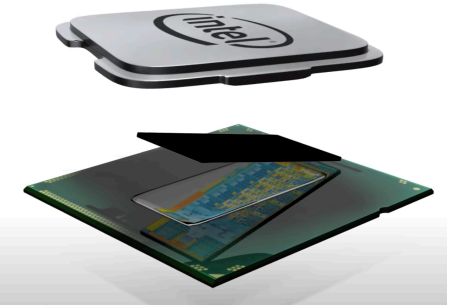
Why do we care about 0s and 1s?

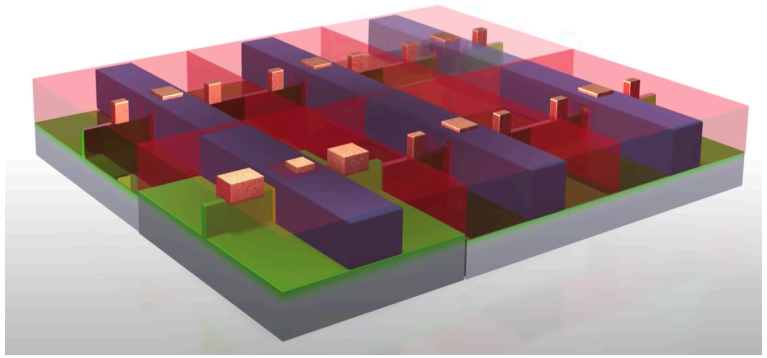
How does a computer *physically* compute?

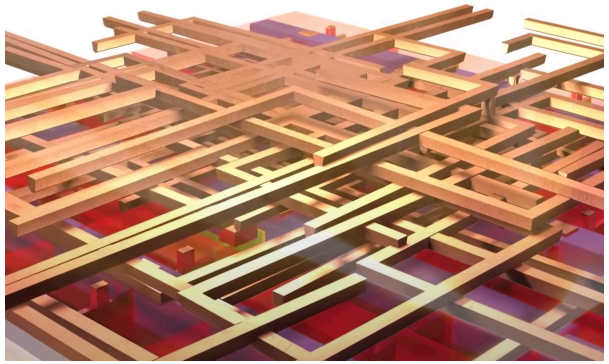
What do these 0s and 1s mean physically?

CLASSICAL HARDWARE





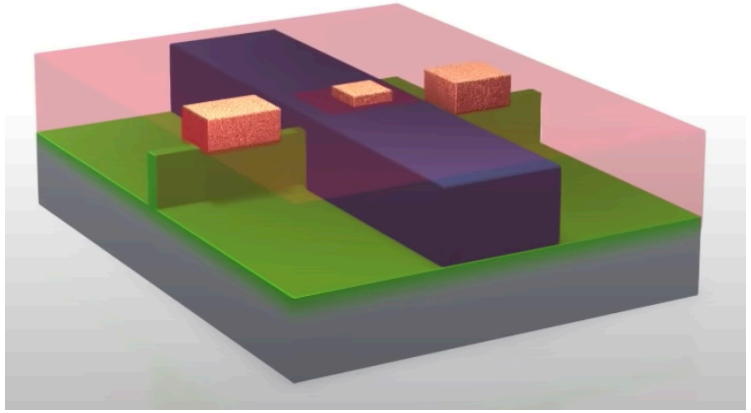




To see how Intel makes their transistors: https://www.youtube.com/watch?v=d9SWNLZvA8g

BITS ARE FUNDAMENTAL!

Classical computers are powered by millions of tiny transistors!



Transistor

=

1

0

A transistor is just a tiny switch, which either allows electricity to pass or not.

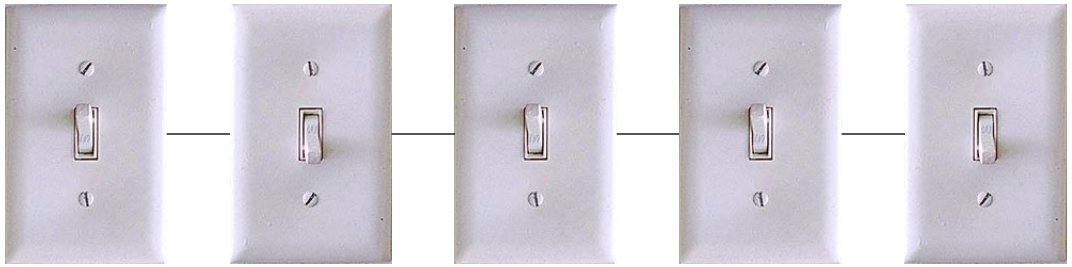
CLASSICAL BIT STRINGS

As computer scientists, we theoretically abstract away the hardware with ***binary representation***!

Mathematically, we abstract the state of a single classical bit as simply 0 or 1:

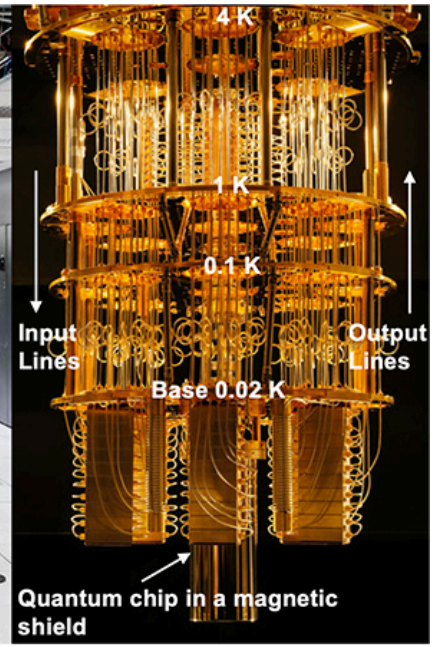
 **= 0**  **= 1**

Mathematically, the state of a group of classical bits is represented by a ***bit-string*** :

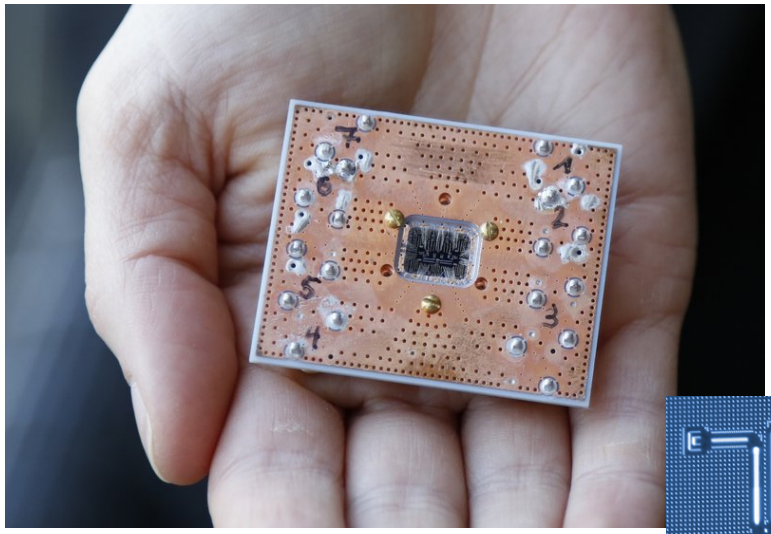
 **= 10110** (= 22)

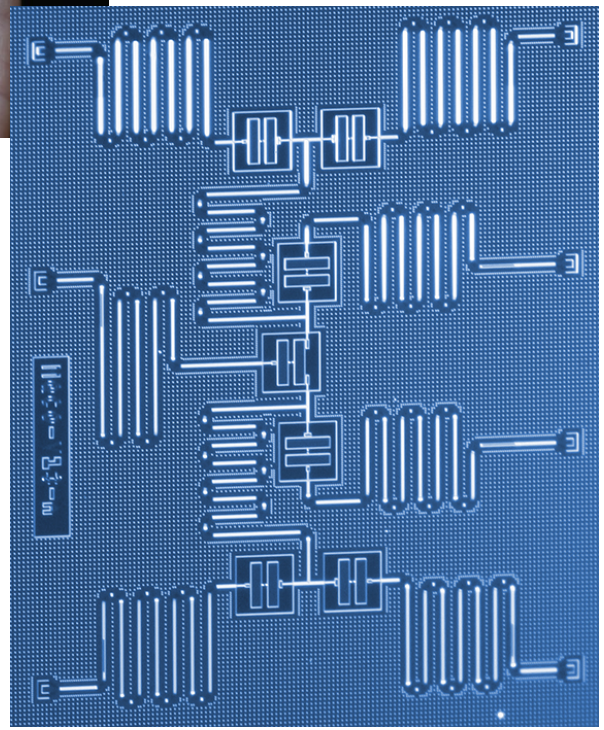
NOW QUANTUM COMPUTATION...

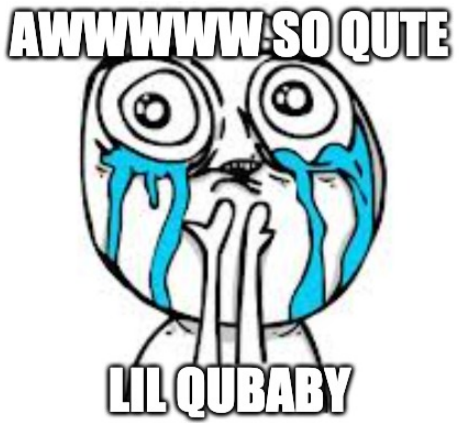
Dilution fridge setup: outside view

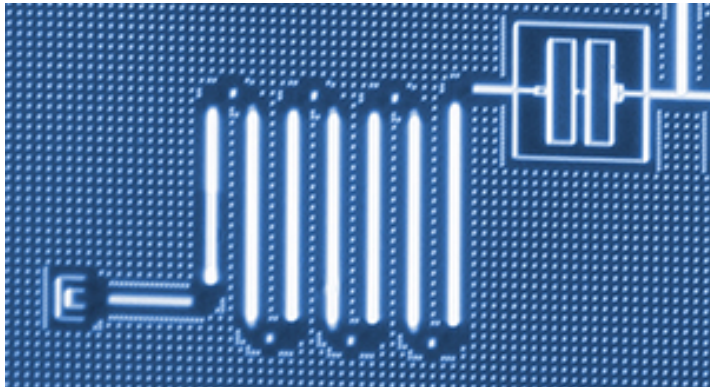


Dilution fridge setup: inside view



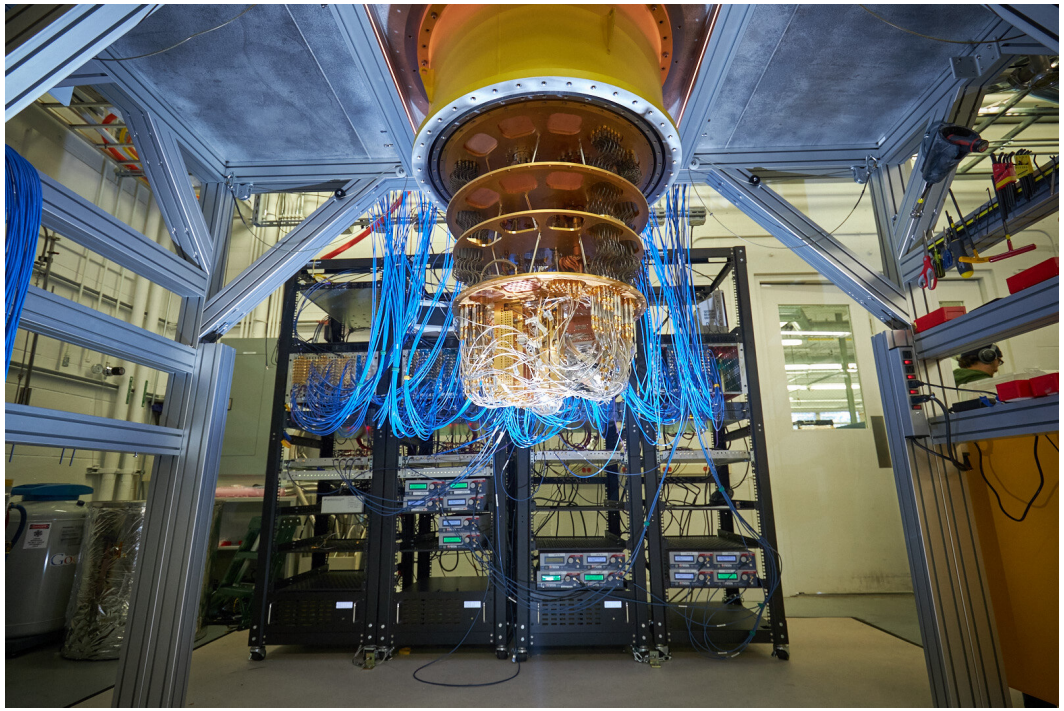




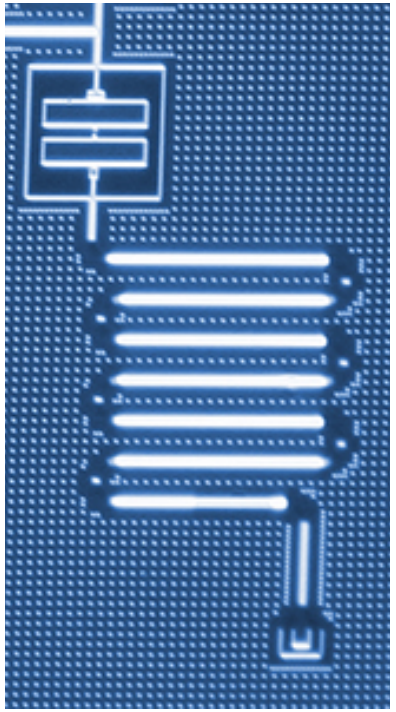


QUBITS ARE FUNDAMENTAL!

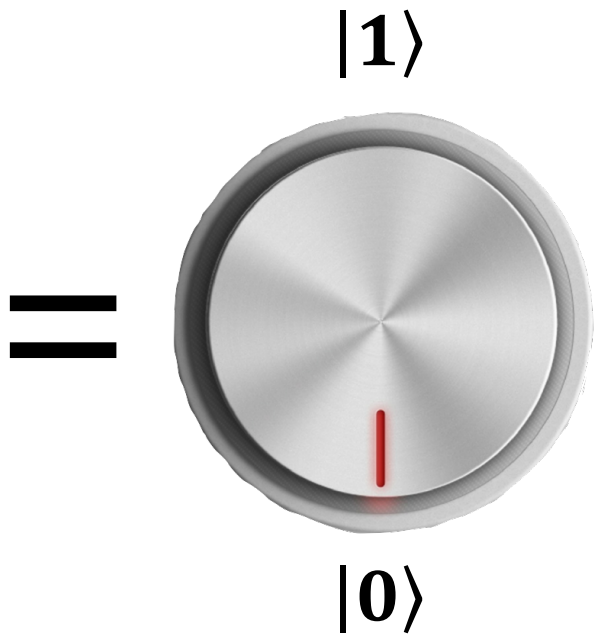
Modern quantum computers are powered by tens of qubits...



(and in the future hopefully millions...)



Qubit

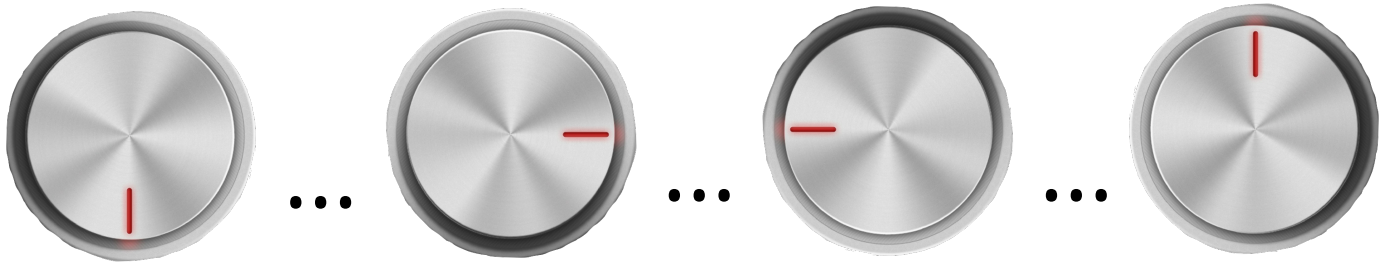


A qubit can be thought of as a dial, in a superposition of the ground and excited energy states.

QUANTUM BIT STRINGS

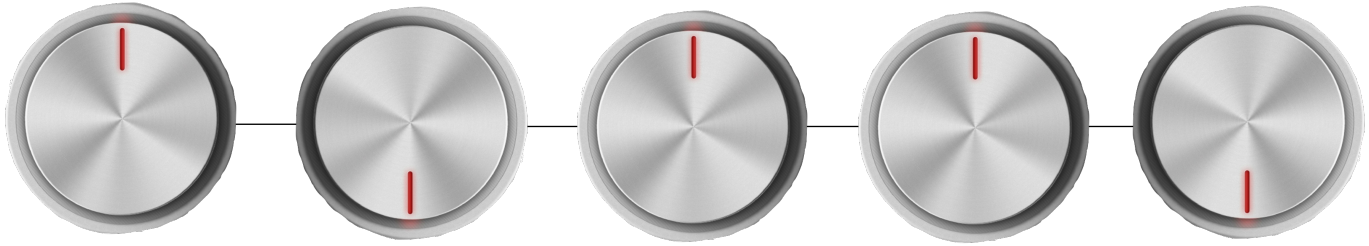
As quantum computer scientists, we theoretically abstract away the quantum hardware with ***vectors*** and ***Dirac notation***!

Mathematically, the state of a single quantum bit (qubit) is represented as a superposition of $|0\rangle$ and $|1\rangle$:



$$|0\rangle \quad \frac{|0\rangle + |1\rangle}{2} \quad \frac{|0\rangle - |1\rangle}{2} \quad |1\rangle$$

Mathematically, the state of a group of quantum bits is represented by a ***quantum bit-string*** :



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

$$= |10110\rangle \quad (= |22\rangle)$$

STATES – KEY IDEAS

🔑 Theoretically, our states simply boil down to:

CLASSICAL STATES: 0, 1

QUANTUM STATES: $\alpha|0\rangle + \beta|1\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$

🔑 Multiple states are represented as (qu)bit strings:

CLASSICAL BIT STRING: 0110101

QUANTUM BIT STRING: $|0110101\rangle$

QUBIT X QUBIT

The <Coding School/>

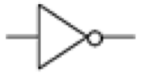
CLASSICAL VS QUANTUM CIRCUITS

GATES

BOOLEAN LOGIC

Last semester, we learned about how ***Boolean logic*** can be used to manipulate classical states.

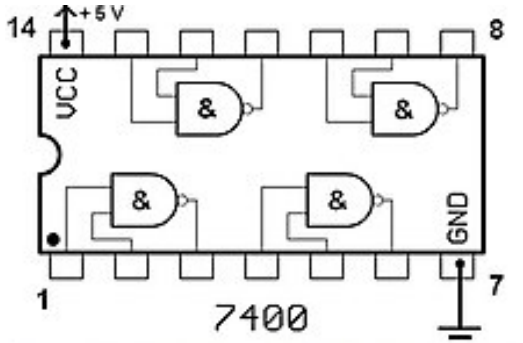
We learned about a few logic gates in particular...

Gate	Symbol	Operator
and		$A \cdot B$
or		$A + B$
not		$\bar{A}$
nand		$\overline{A \cdot B}$
nor		$\overline{A + B}$
xor		$A \oplus B$

BOOLEAN LOGIC

Again, these gates that we draw are a *theoretical representation* of actual hardware in your computer, which is used to perform computation.

Gate	Symbol	Operator
and		$A \cdot B$
or		$A + B$
not		$\bar{A}$
nand		$\overline{A \cdot B}$
nor		$\overline{A + B}$
xor		$A \oplus B$



Further, remembering our discussion of the stack, all programs that you write get compiled down to machine code, which simply performs these Boolean logic operations on bit strings.

BOOLEAN LOGIC

Each classical logic gate performed a specific input-output mapping (represented by a ***truth table***).

There were two possible options, either the bit would remain the same or it would flip:

$$0 \mapsto 0 \quad \text{or} \quad 0 \mapsto 1$$

$$1 \mapsto 1 \quad \text{or} \quad 1 \mapsto 0$$

Mini-Quiz!

Which gate is this the truth table of?

Truth Table

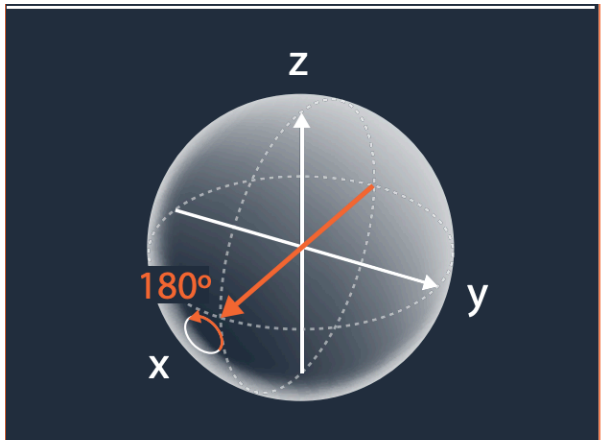
Input A	Input B	Output
0	0	0
1	0	1
0	1	1
1	1	1

QUANTUM LOGIC?

Throughout the course, we have talked about quantum operations, which mathematically are represented as matrices.

The Pauli operations rotate our quantum state vector around the 3 axes of the Bloch sphere!

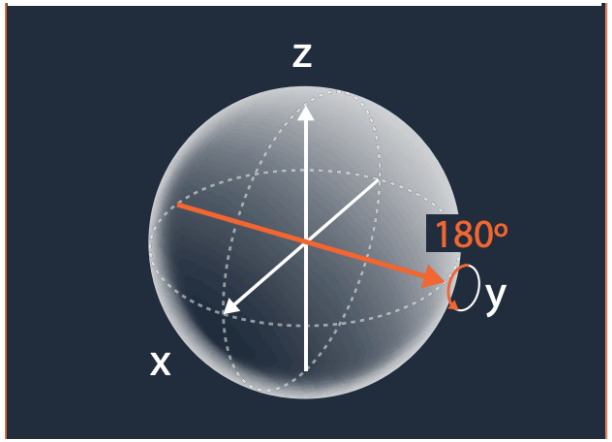
Pauli-X



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

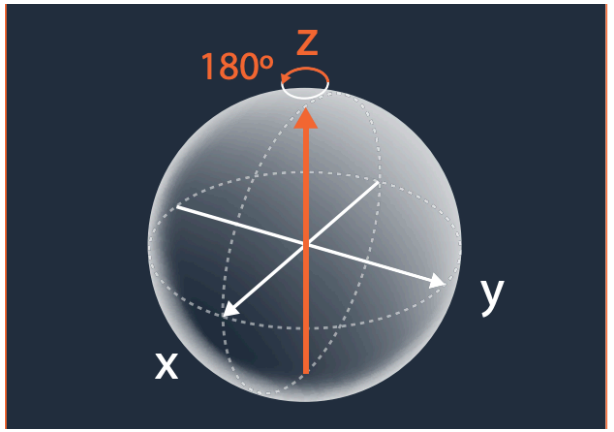
(Bit Flip)

Pauli-Y



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

Pauli-Z

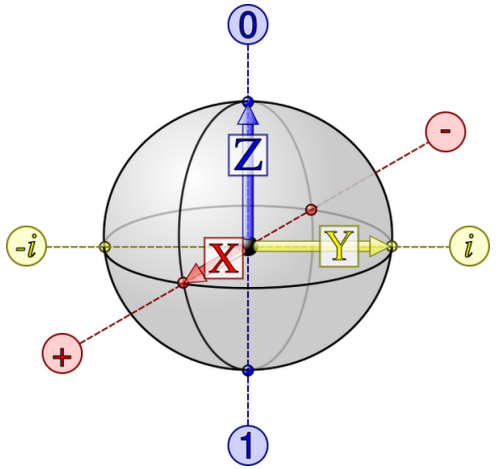


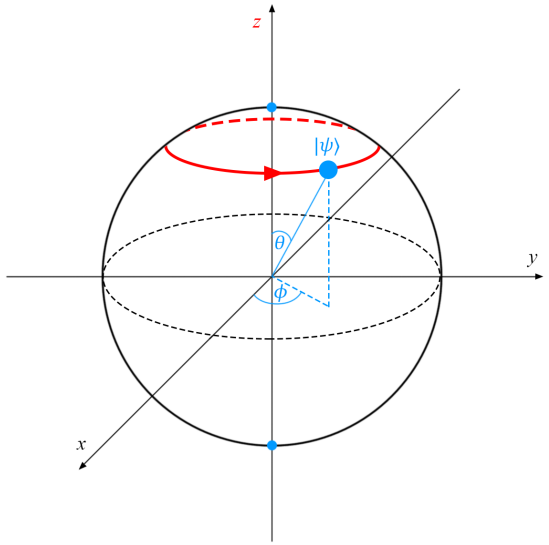
$$\sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

(Phase Flip)

PUTTING ALL THE MATH TOGETHER!!

- Qubits lie in a 2D ***complex Hilbert space***, which we represent with the ***Bloch sphere***.
- ***Eigenvectors*** of the Pauli matrices form ***bases*** for this Hilbert space.
- Thus, every quantum state is a ***linear combination*** of a Pauli's eigenvectors.
- In quantum mechanics, observable operations are ***Hermitian***.
- It turns out that that the Pauli matrices $\{X, Y, Z, I\}$ form a basis that ***span*** the vector space of 2x2 Hermitian matrices.
- ***Any 2x2 Hermitian matrix (observable quantum operator) can be written as a unique linear combination of the Pauli and identity matrices!***
- ***All observable single qubit operations correspond to rotations in the Bloch sphere!***





GATES – KEY IDEAS

We can manipulate the state of a single qubit by rotating it in the Bloch sphere.

Thus, quantum operations (matrices) are the ***gates*** of quantum computation.

Some important single-qubit gates:

HADAMARD

PAULIS

PHASE GATES

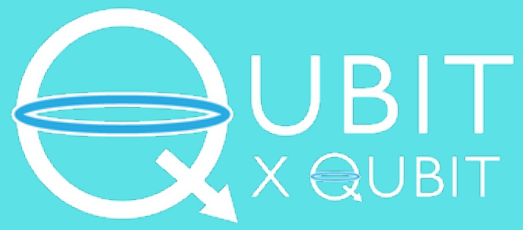
MULTI-QUBIT SYSTEMS

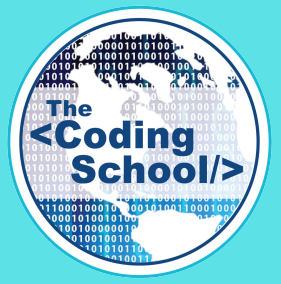
2x2 Hermitian operators can only be used to manipulate single-qubit systems.

However, there are important multi-qubit quantum gates (which are necessary to create entanglement!!):

CNOT **SWAP** **TOFFOLI**

In theory, you can define an n-qubit quantum gate. However, experimentally it is already hard enough implementing a 2-qubit quantum gate...





BREAK TIME!

QUBIT X QUBIT

The <Coding School/>

CLASSICAL VS QUANTUM CIRCUITS

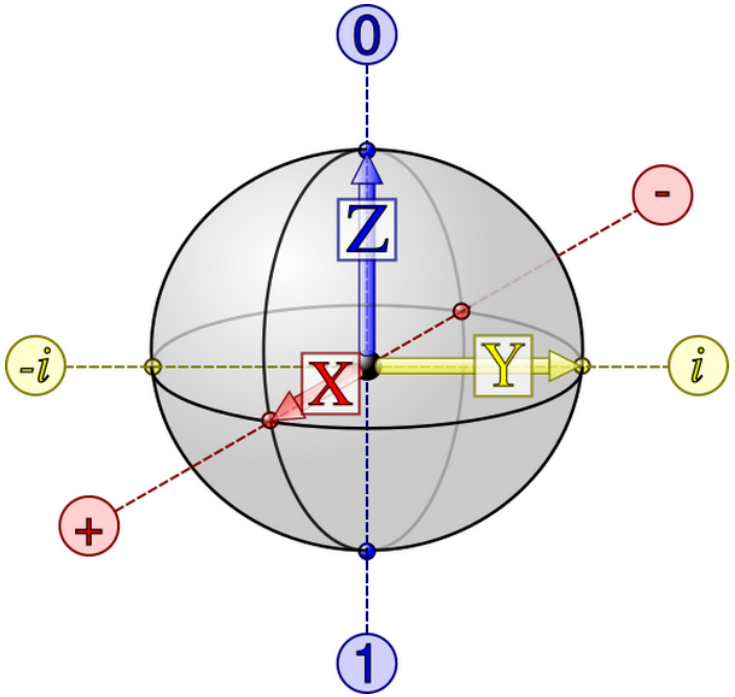
MEASUREMENTS

CLASSICAL MEASUREMENTS?

Last semester, when we discussed Boolean logic, we never really talked about ***measurement***... Why not?

QUANTUM MEASUREMENTS

Unlike deterministic classical measurements, quantum measurements are ***probabilistic***!

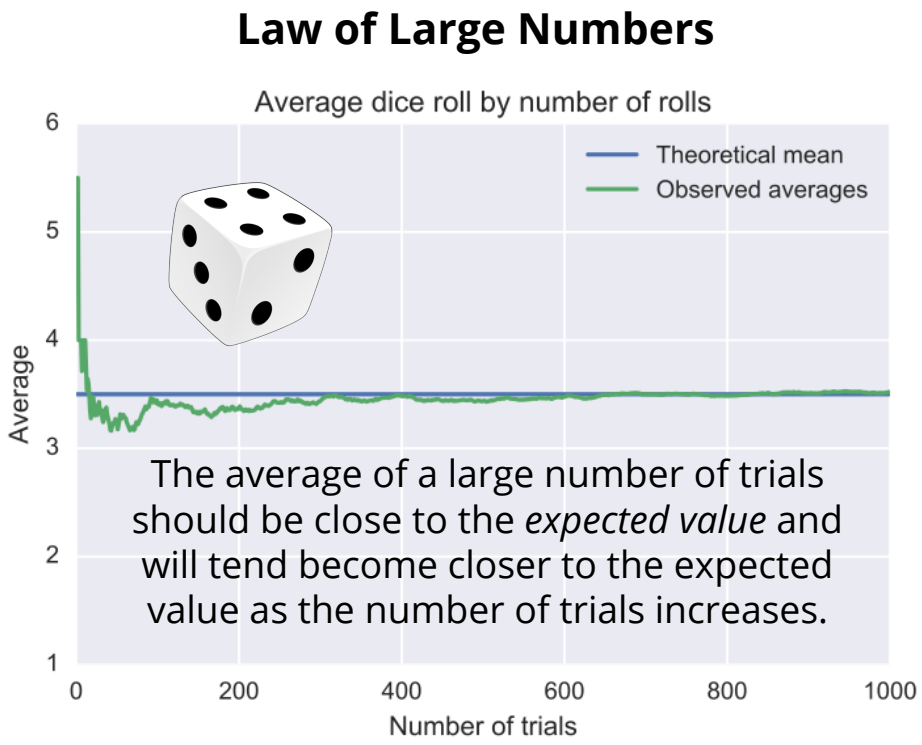


Further, there are different axes along which you can perform these measurements (basis matters)...

Unless otherwise stated, however, we will assume measurements occur along the z-axis.

REPEATED MEASUREMENTS

Recall our discussion of the ***Law of Large Numbers*** last semester...



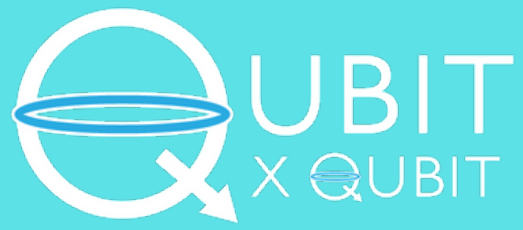
Given the probabilistic nature of quantum measurement, we need to perform ***repeated measurements*** in order to understand the underlying distribution of our state!

MEASUREMENT – KEY IDEAS

Unlike deterministic classical measurements, quantum measurements are ***probabilistic***!

Although you can measure along multiple axes, we usually set the z-axis as our measurement axis in our reference frame.

If you want to know the underlying distribution of a quantum state, you need to create the state several times and perform ***repeated measurements***!



CLASSICAL VS QUANTUM CIRCUITS

PUTTING IT ALL TOGETHER!

THE QUANTUM CIRCUIT MODEL

The quantum circuit is a theoretical model for quantum computation. It has three key components:

STATES,

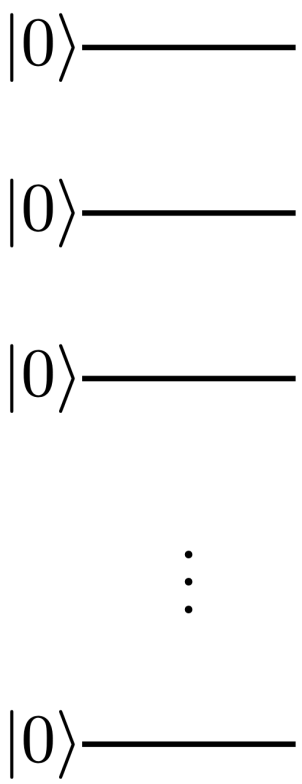
GATES,

& MEASUREMENT

Now, how do we put these all together to create a quantum circuit?

THE QUANTUM CIRCUIT MODEL - STATES

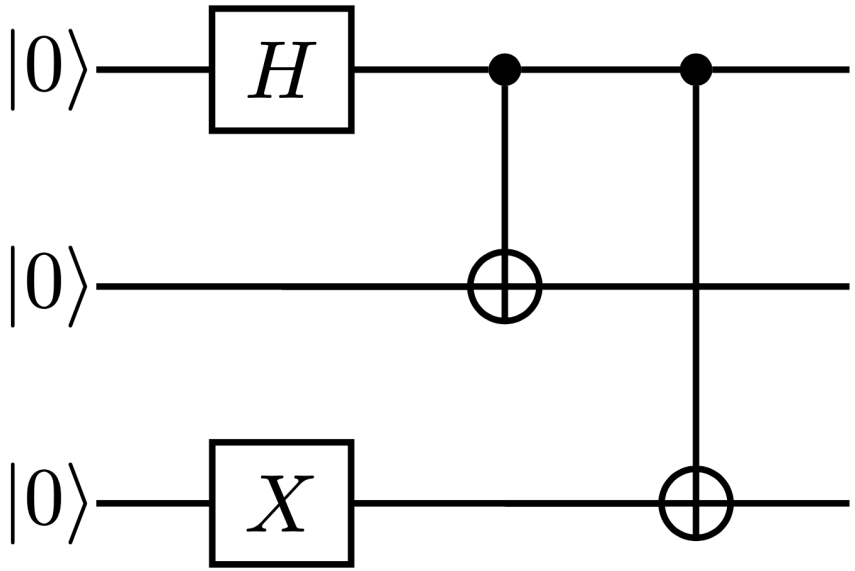
Graphically, our states are represented over time with wires...



THE QUANTUM CIRCUIT MODEL - GATES

We can perform operations (computation!) on those states with gates.

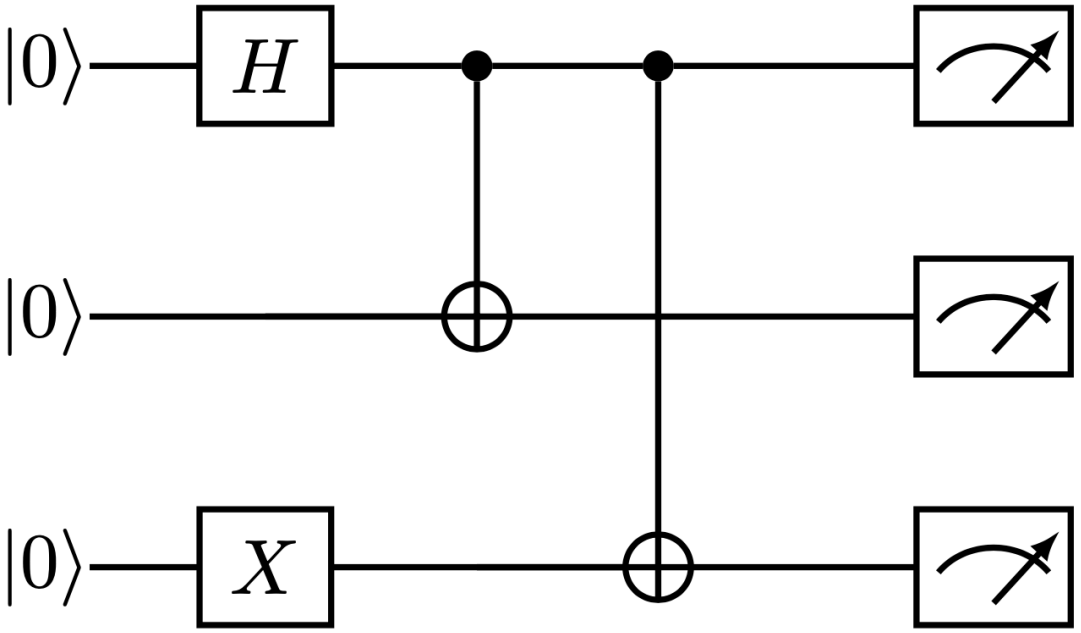
Graphically, our gates are represented as boxes on and connectors between the wires...



THE QUANTUM CIRCUIT MODEL - MEASUREMENT

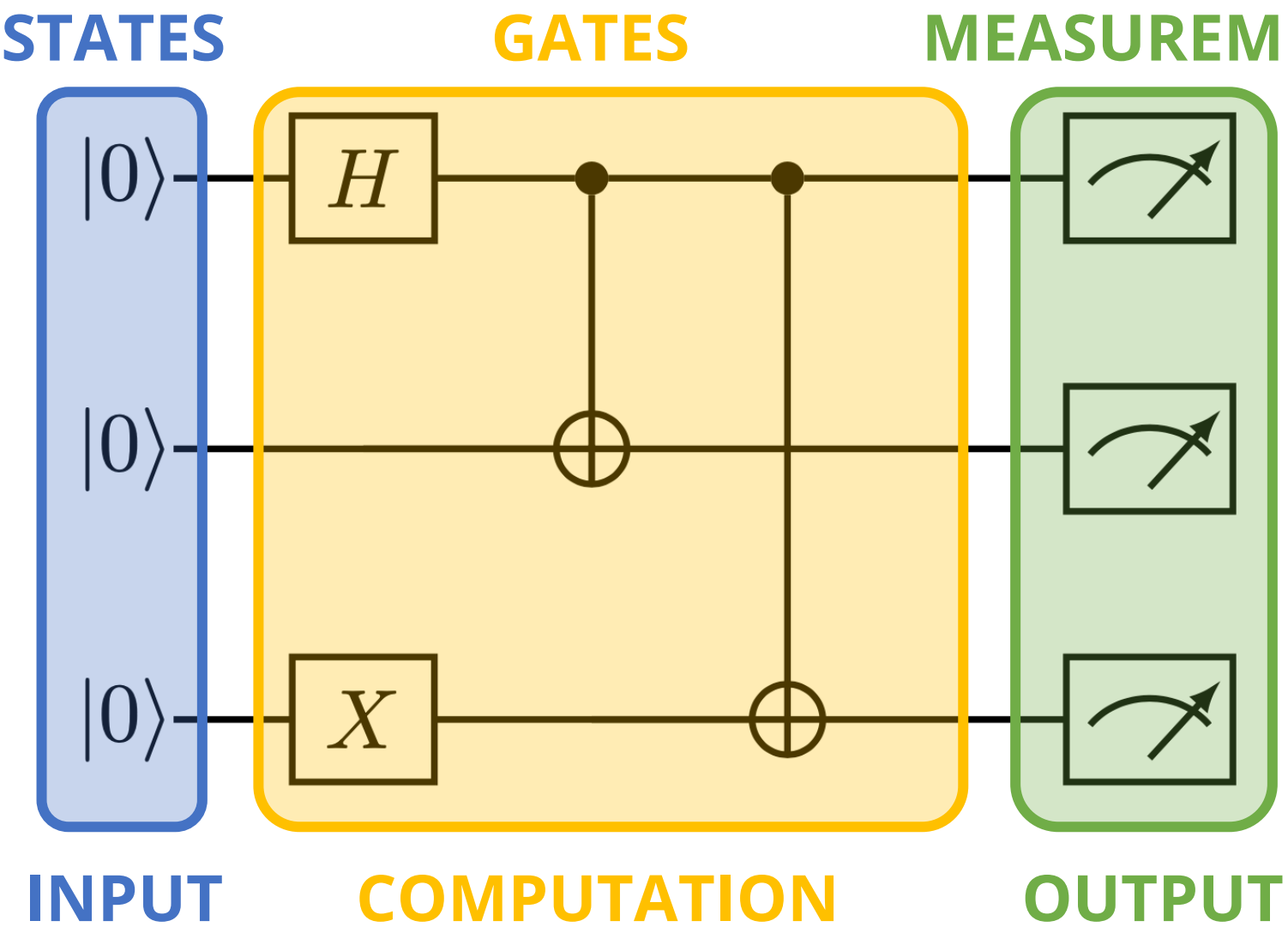
In order to know the output of our computation, we need to perform a measurement!

Graphically, measurement is usually represented with a little dial. **



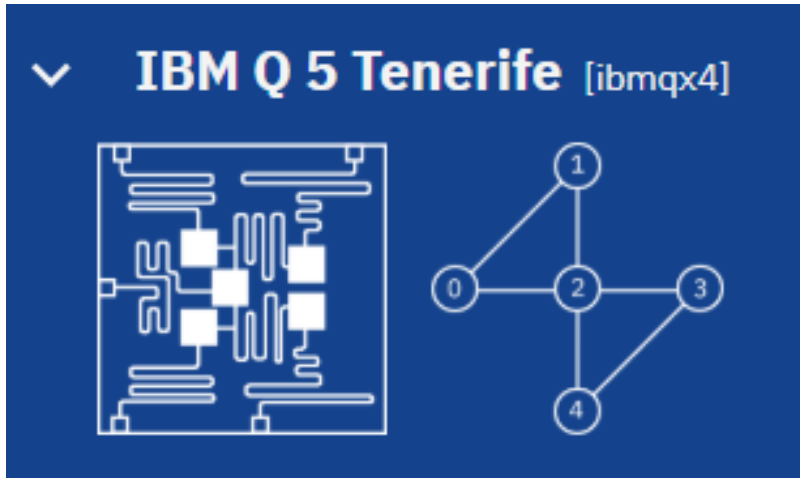
**It is normally implied that the experiment will be repeated multiple times in order to create the distribution of quantum states.

THE QUANTUM CIRCUIT MODEL

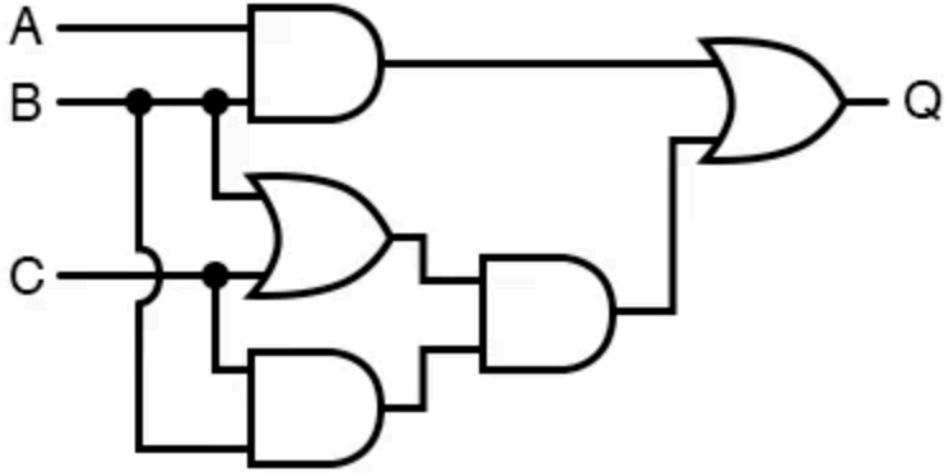


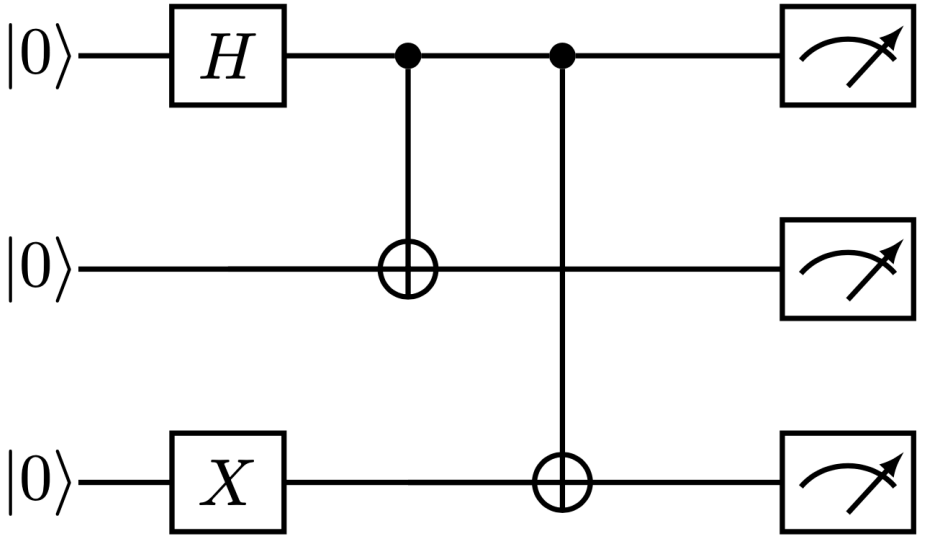
NOTE ON CHIP CONNECTIVITY

On actual quantum devices (especially large ones), it is very rare that all qubits are connected pairwise. Instead, qubits are usually only connected to neighboring qubits. Thus for 2-qubit gates (and larger) you should keep chip connectivity in mind...



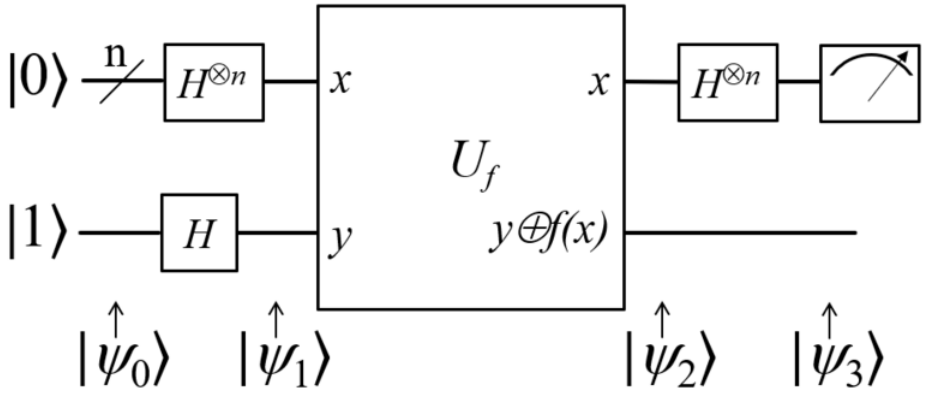
CLASSICAL VS QUANTUM CIRCUITS



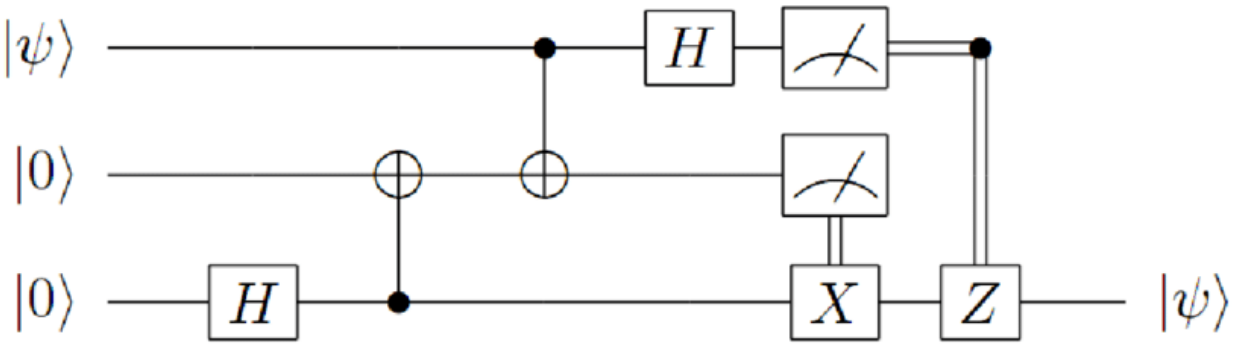


SOME QUANTUM CIRCUITS TO COME...

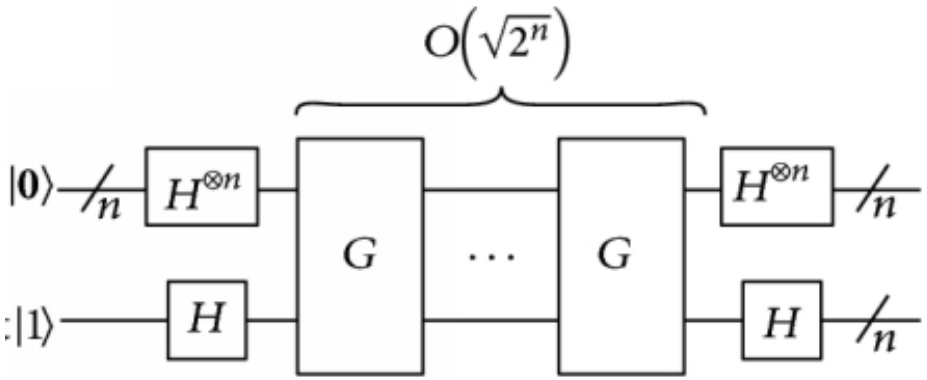
DEUTSCH-JOZSA



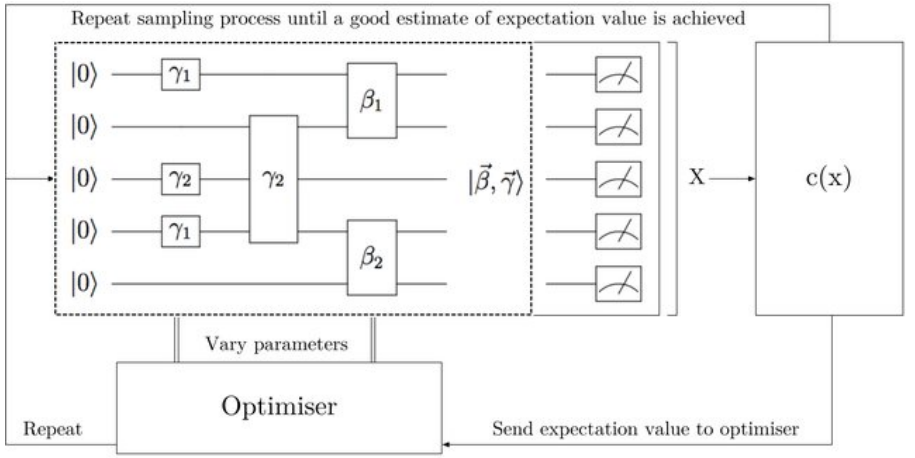
QUANTUM TELEPORTATION



GROVER'S

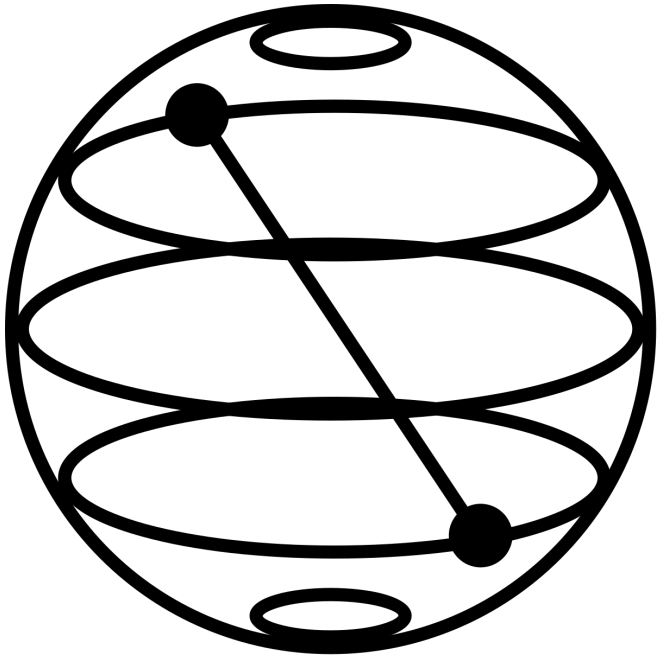


QAOA



QISKIT

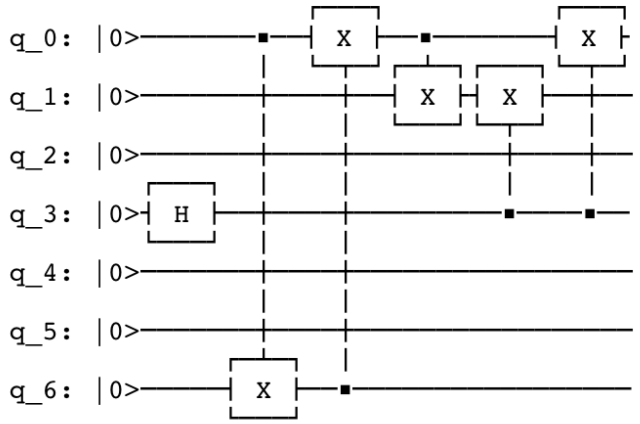
Next class, you will learn how to construct quantum circuits via programming, with IBM's ***Qiskit*** software!



In [164]:

```
1  qr = QuantumRegister(7, 'q')
2  qr = QuantumRegister(7, 'q')
3  circuit = QuantumCircuit(qr)
4  circuit.h(qr[3])
5  circuit.cx(qr[0], qr[6])
6  circuit.cx(qr[6], qr[0])
7  circuit.cx(qr[0], qr[1])
8  circuit.cx(qr[3], qr[1])
9  circuit.cx(qr[3], qr[0])
10 circuit.draw()
```

Out[164]:

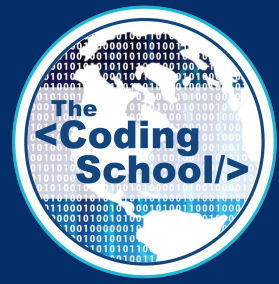


With Qiskit, you can run your quantum circuits on IBM's actual superconducting quantum computers!!!

PROGRAMMING LANGUAGES

In classical computing, there are several programming languages available...

Although we use Qiskit in this class, there are tons of quantum programming languages out there as well...



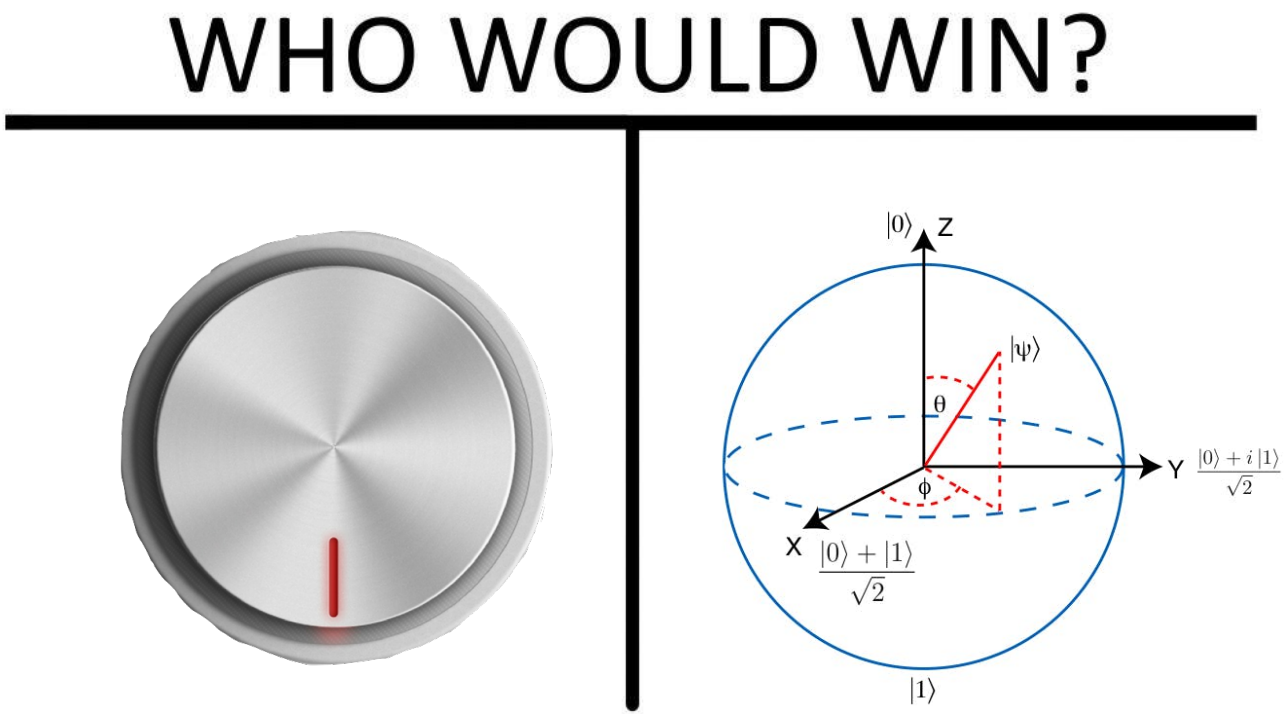
WHY CIRCUITS?

When we program in Python, we are not writing programs to create Boolean logic circuits...

So why do we program to create quantum circuits???

DIALS, WHAT ABOUT THE BLOCH SPHERE?

At the beginning of the year, we said you could think of a qubit as a dial. However, over time we showed that quantum states actually lie in the Bloch sphere.

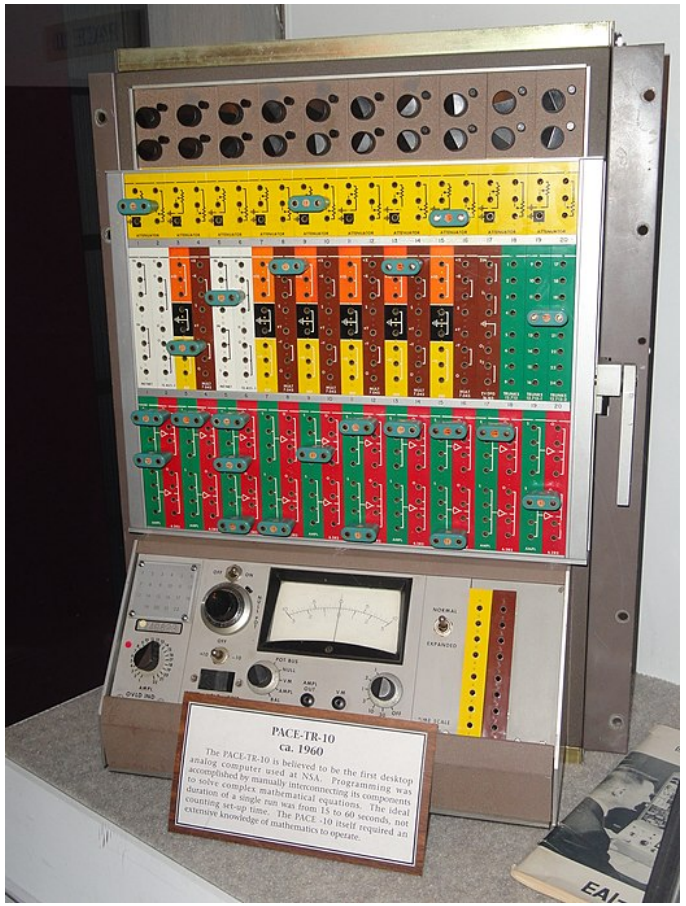


What's the difference? Why the added complication?

DIALS -- ANALOG COMPUTING

In the early 1900s, researchers were very excited by the prospects of *analog computers*, in which bits were actually continuous (or analog) – literally dials!

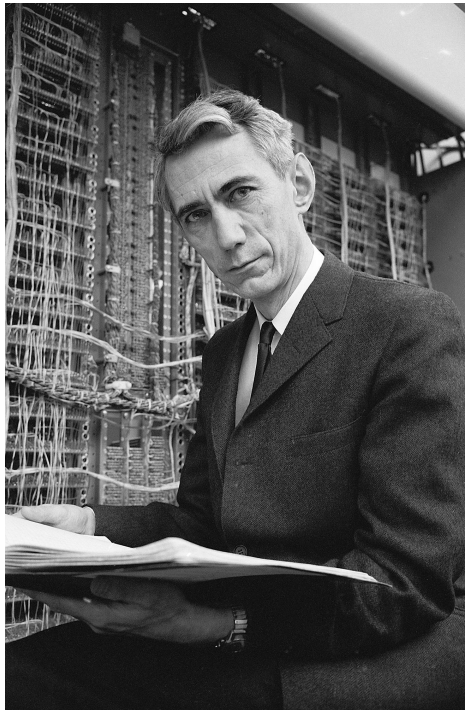
However, by the 1960s, researchers largely gave up on the technology, mostly because of the inability to correct for ***noise***.



Desktop analog computer 1960.

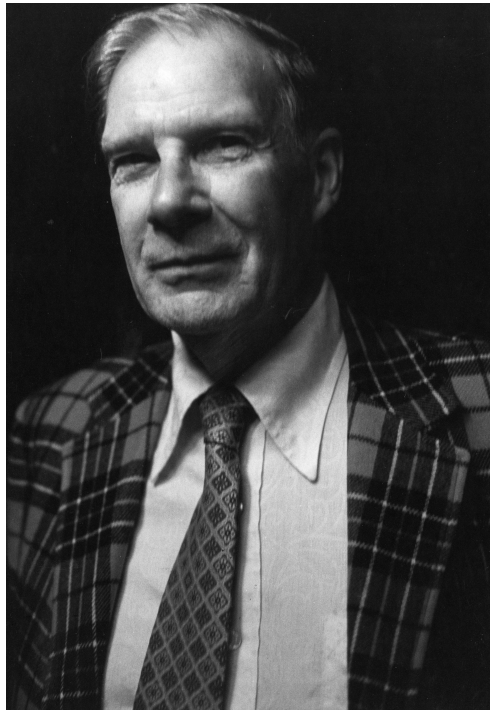
TANGENT: NOISE & ERROR CORRECTION

The problem of noise was largely resolved in the 1940-50s for classical digital computers, with the introduction of ***information theory*** and classical ***error detection/correction***.



Claude Shannon

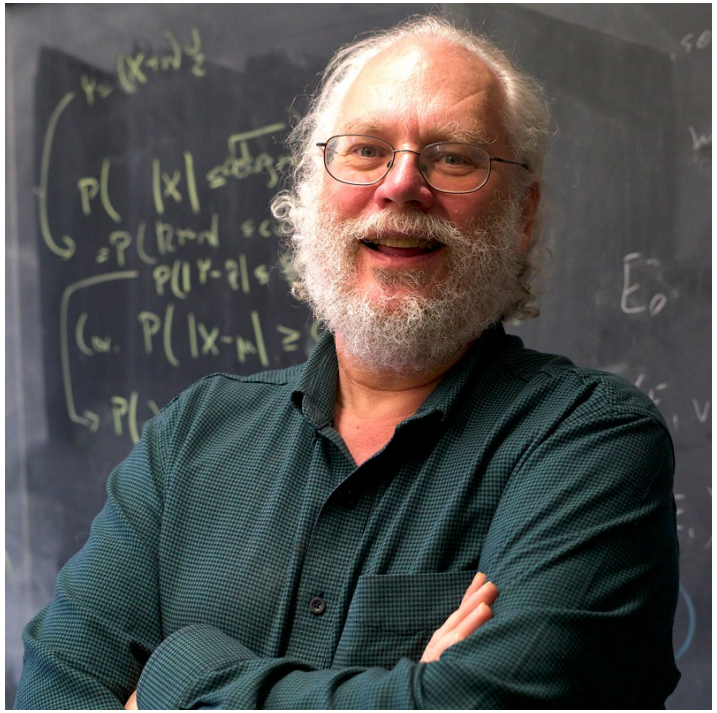
Father of Information Theory



Richard Hamming

Developed 1st Modern Error Correcting Codes

The problem of noise was similarly addressed for quantum computers by Peter Shor in the 1950s, with the introduction of ***quantum error correction***.



Peter Shor

Developed Shor's Algorithm & the 1st Quantum Error Correction Scheme

TANGENT: PETER SHOR

Peter Shor worked at Bell Labs, before becoming an MIT Professor of Mathematics.

Peter Shor is most famous and widely regarded for his development of ***Shor's algorithm***, which provides an exponential speedup for prime factorization. This has really important implications in the field of cryptography, since RSA (the main crypto scheme in use today) relies on the fact that factorizing large prime numbers is *really* hard. Thus, the algorithm got a lot of people – researchers and governments – interested in quantum computing.

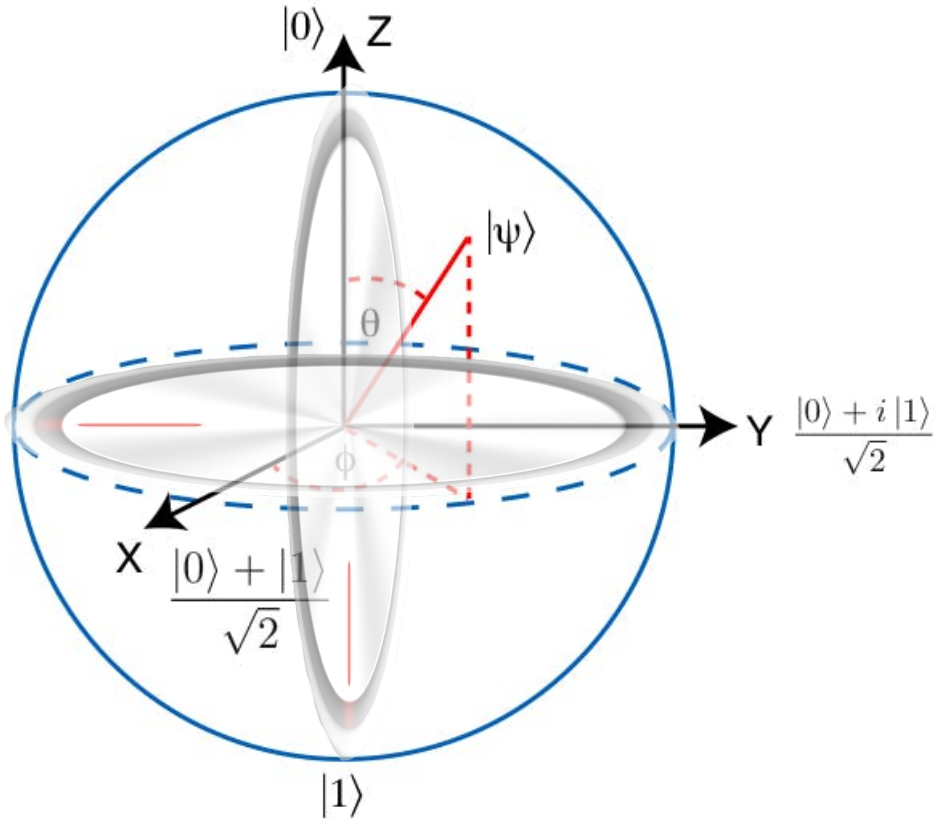
Shor's algorithm popularized quantum computing!

However, it was actually Shor's work on quantum error correction – including the introduction of the ***Shor Code*** – which ensured quantum computers would work in practice. Quantum error correction enables qubits to be resilient to certain types of noise. Without the development of quantum error correction theory, quantum computers would have probably died out like analog computers.

Shor's pioneering work in quantum error correction ensured that quantum computers would work in the presence of noise!**

**assuming we can reach the error thresholds necessary for fault-tolerance

BLOCH SPHERE -- QUANTUM COMPUTING



If you like thinking of qubits as dials, you still can!

However, you should think of a qubit as being two dials:

(1) The first dial controls the level of superposition between the ground and excited states!

(2) The second dial controls the *phase difference* between the states in your superposition!