Los Alamos NATIONAL LABORATORY

Foundations of Quantum Computing

Carleton Coffrin
Advanced Network Science Initiative

05/28/2024

With help from Giacomo Nannicini and Marc Vuffray

LA-UR-24-25068

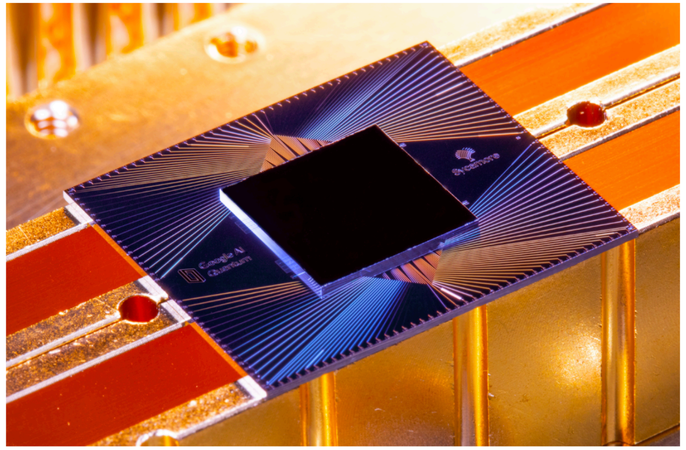
You may have heard, Quantum Computers are here!

The New York Times

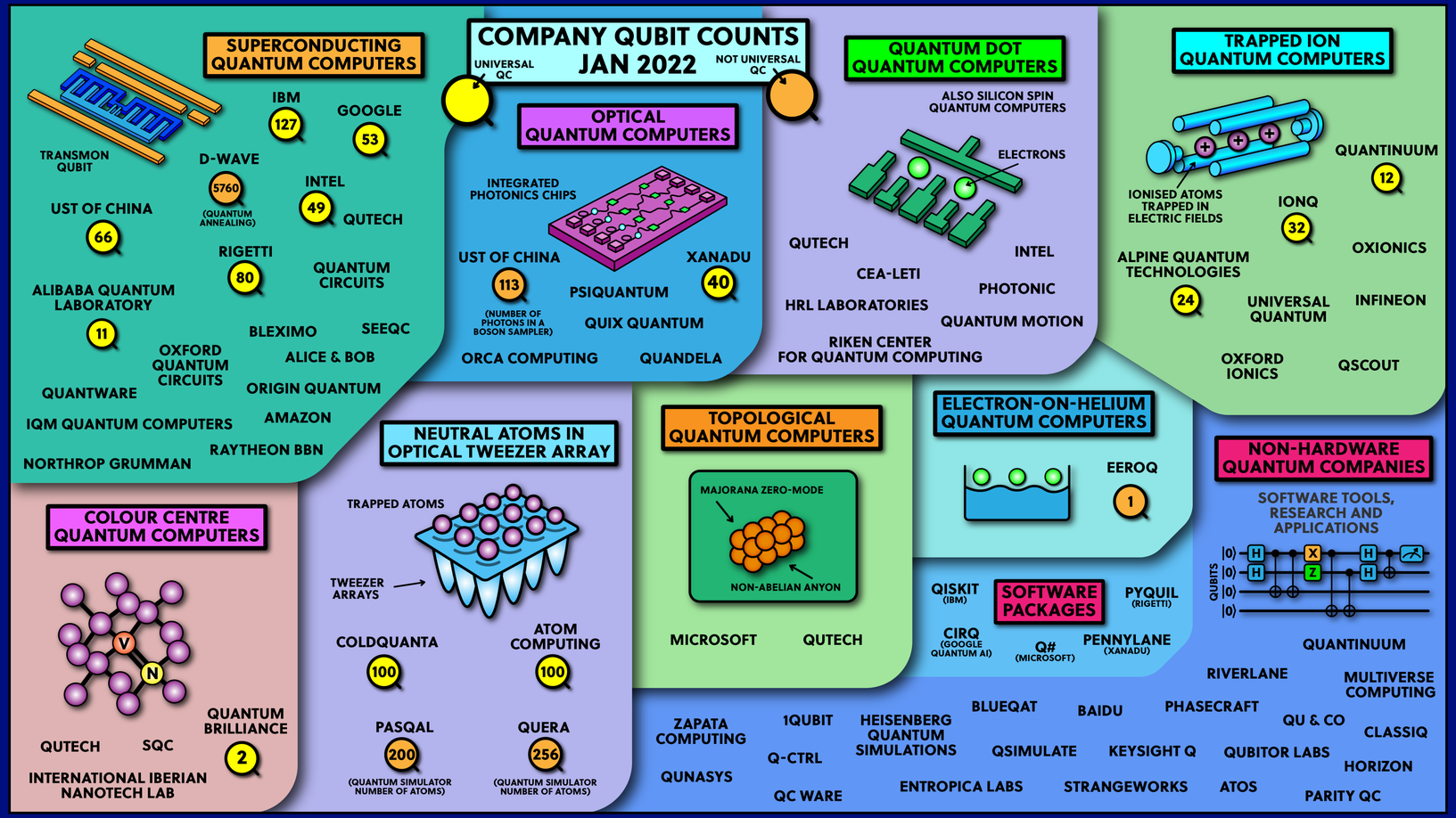
Why Google's Quantum Supremacy Milestone Matters

The company says its quantum computer can complete a calculation much faster than a supercomputer. What does that mean?

Oct. 30, 2019



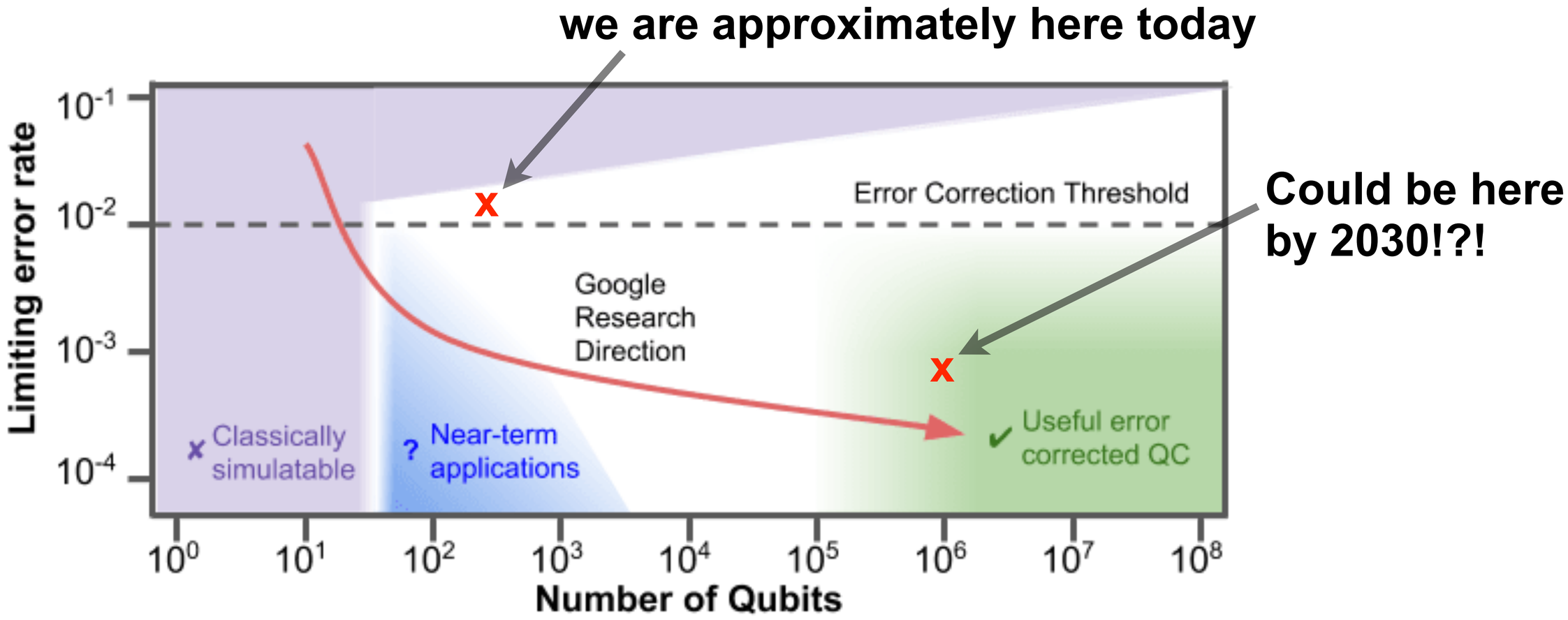
Google A.I. Quantum's Sycamore processor. Erik Lucero/Google



Domain of Science - Dominic Walliman

Era of “Noisy Intermediate Scale Quantum” (NISQ)

Translation: these computers are just barely working



https://www.hpcwire.com/2018/04/26/google-frames-quantum-race-as-two-dimensional/

The State of Quantum Computing

What does it compute?

Will quantum computers be good for anything?

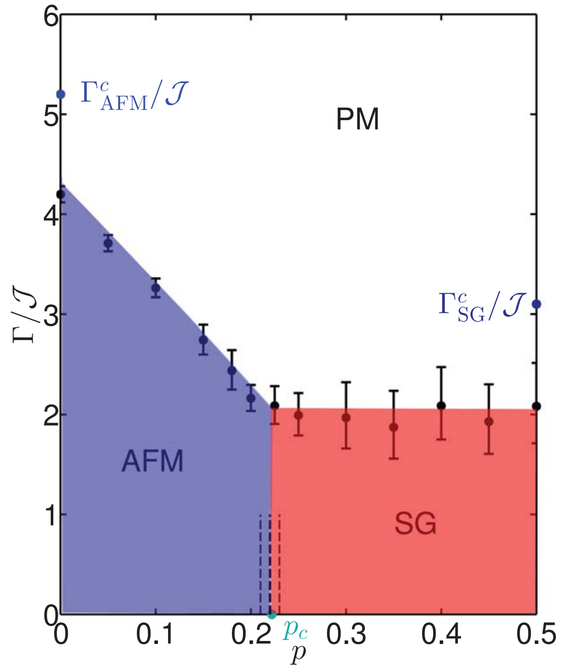
UNIVAC 1960

How fast does it compute?
(esp. compared to state-of-the-art)

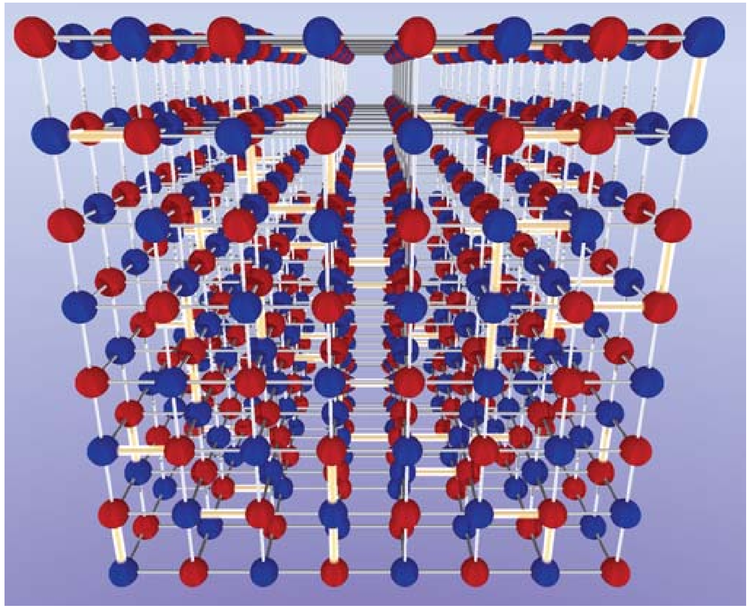
Los Alamos NATIONAL LABORATORY

Killer Apps for Quantum Computers?

- **Simulation of Quantum Systems**
  - Materials, Chemistry

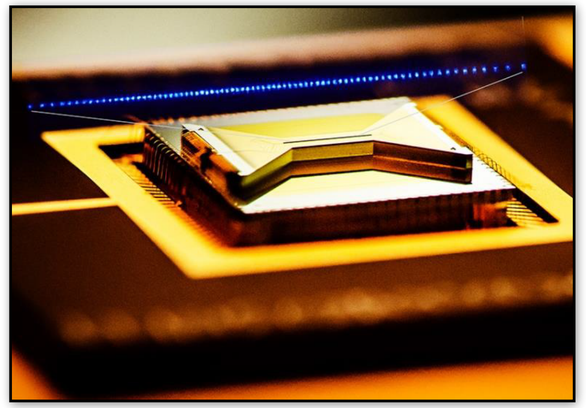


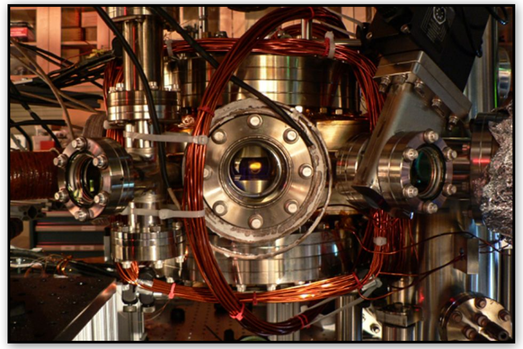
Phase Transition Diagram

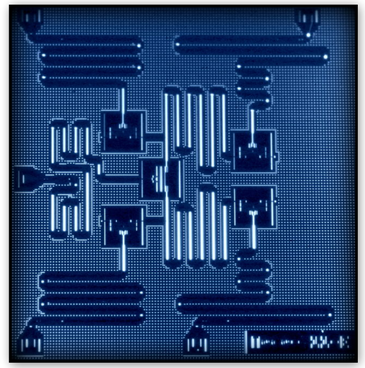


3D Ising Model

Quantum Computer Hardware







But wait, this is an CP/AI/OR conference right?

Will Quantum Computing change how we do in ***optimization?***

I think so! Make up your own mind after hearing from our great lineup of speakers

Master Class Overview (timing)

- **08:45 - 09:00** **Opening Remarks**
- **09:00 - 10:30** **Carleton Coffrin** - Foundations of Quantum Computing
- **10:30 - 11:00** **Break (30m)**
- **11:00 - 12:00** **Ashley Montanaro** - Quantum Computing Algorithms for Operations Research and Constraint Programming
- **12:00 - 13:30** **Lunch (provided; 1h 30m)**

- **13:30 - 14:30** **Tamás Terlaky** - Quantum Computing Algorithms for Interior Point Methods
- **14:30 - 15:00** **Harsha Nagarajan** - Designing Quantum Circuits with Mixed-Integer Polynomial Programming
- **15:00 - 15:30** **Break (30m)**
- **15:30 - 16:00** **Zachary Morrell** - A Brief Introduction to Quantum Annealing
- **16:00 - 16:30** **Andreas Bärtschi** - A Brief Introduction to the Quantum Approximate Optimization Algorithm (QAOA)
- **16:30 - 17:00** **Xiaodi Wu** - Quantum Hamiltonian Descent for Non-convex Continuous Optimization
- **17:00 - 17:30** **David Bernal Neira** - Quantum-Classical Hybrid Methods for Optimization
- **17:30 - 17:45** **Closing Remarks**

- **19:00 - 20:30** **Conference Reception**

Master Class Overview (topics)

Speaker	Topic	Optimization Area	Quantum Computing Model
Carleton Coffrin	QC Foundations	NA	Fault-Tolerant
Ashley Montanaro	Quantum Computing Algorithms for Operations Research and Constraint Programming	QC formal methods, discrete variables (OR/CP)	Fault-Tolerant
Tamás Terlaky	Quantum Computing Algorithms for Interior Point Methods	QC formal methods, continuous variables (IMPs)	Fault-Tolerant
Harsha Nagarajan	Designing Quantum Circuits with Mixed-Integer Polynomial Programming	MINLP for QC, discrete + continuous variables	NISQ and Fault-Tolerant
Zachary Morrell	A Brief Introduction to Quantum Annealing	QC heuristic, discrete variables	Analog
Andreas Bärtschi	A Brief Introduction to the Quantum Approximate Optimization Algorithm (QAOA)	QC meta-heuristic, discrete variables	NISQ
Xiaodi Wu	Quantum Hamiltonian Descent for Non-convex Continuous Optimization	QC heuristic, continuous variables	Analog
David Bernal Neira	Quantum-Classical Hybrid Methods for Optimization	QC hybrid methods, discrete + continuous variables	Analog and NISQ and Fault-Tolerant?

A Brief Introduction to Quantum Computation, Sections

- Introduction (8 sides)

- **Part 1 - The Mathematics of Gate-Based Quantum Computing (26 Slides)**
  - Preliminaries (5)
  - State of a Quantum Computer (8)
  - Quantum Operations (6)
  - Quantum Gate Sets (6)
  - Challenges and Uses of Quantum Computers (2)

- **Part 2 - Quantum Computing and Optimization (5 Slides)**

- **Part 3 - State of Quantum Computing Hardware (8 Slides)**

The Mathematics of *Gate-Based* Quantum Computing (Part 1)

"An Introduction to Quantum Computing, without the Physics"
Giacomo Nannicini, SIAM Review

Most important point
This introduction mostly about notation!

The gate model of quantum computing
is just linear algebra over complex numbers
with *A LOT* of special notation

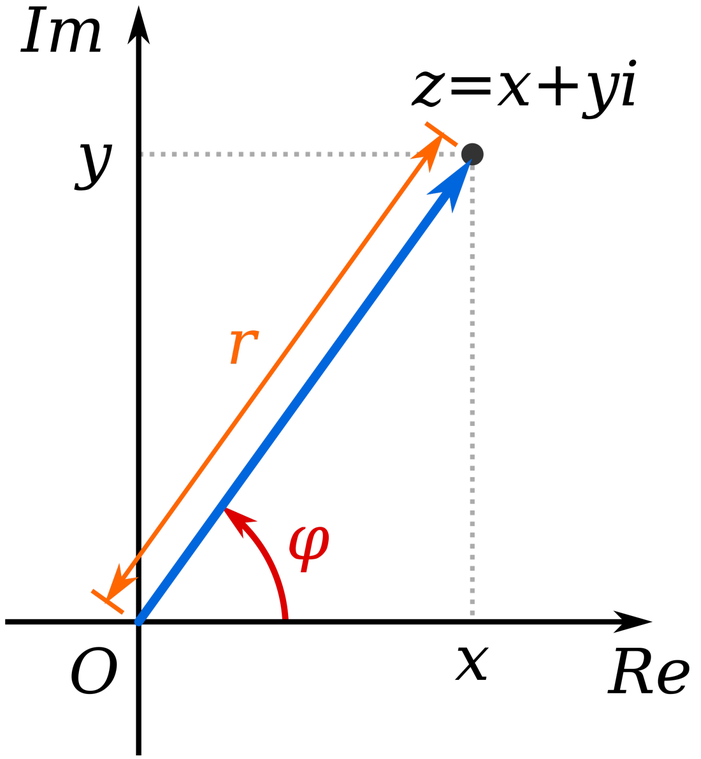
Preliminaries

Brief Refresher on Complex Numbers

- **A complex number has two parts**
  - “Real” x-axis
  - “Imaginary” y-axis

- **Can be represented in “polar form” as a magnitude “r” and an angle “ϕ”**

- **Two important operations**
  - Complex Conjugate (i.e., Z^*)
  - Complex Conjugate and Matrix Transpose, A^{*T} (i.e., $A^{*T}, A^H, A^\dagger$)



$$Z = x + yi = r\angle\phi$$

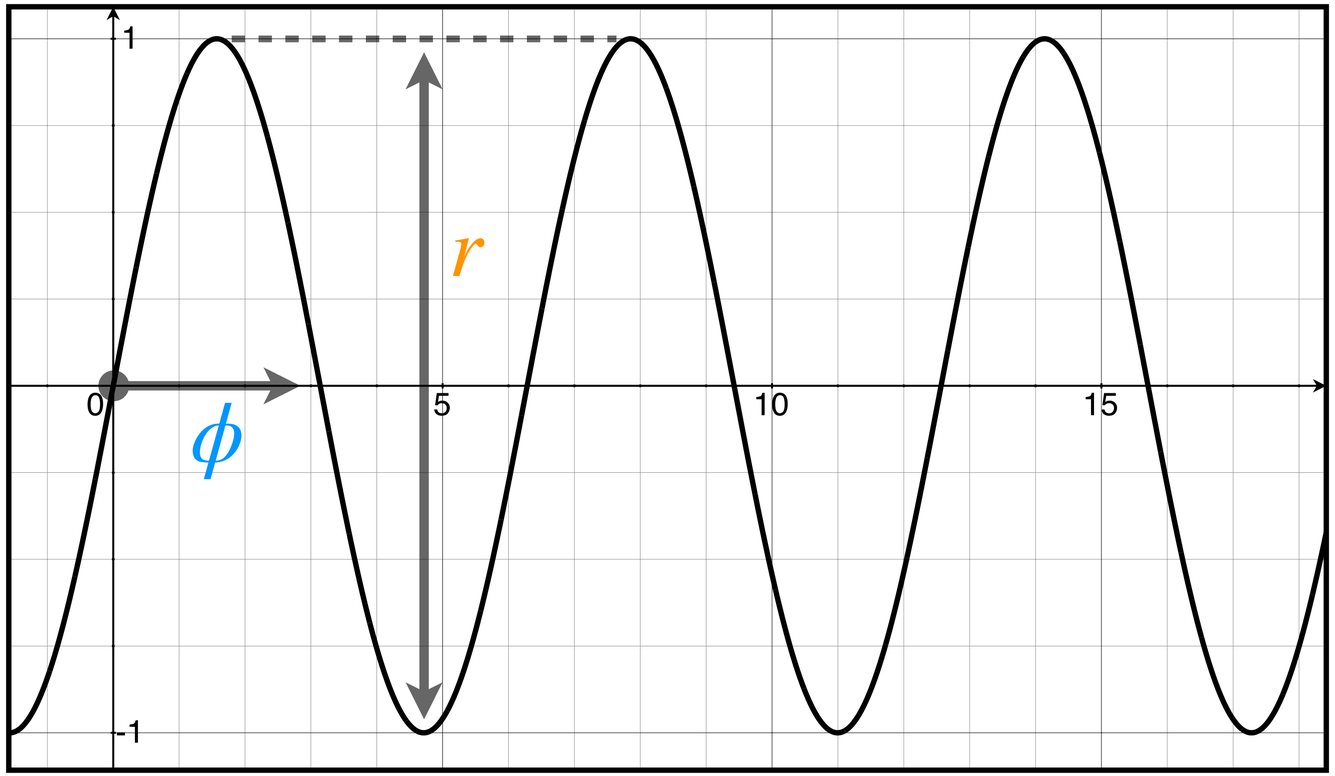
$$Z^* = x - yi = r\angle -\phi$$

https://en.wikipedia.org/wiki/Complex_number

https://en.wikipedia.org/wiki/Conjugate_transpose

Why Complex Numbers for Quantum Computing?

- **Quantum physics is made of waves!**



$$Z = r \angle \phi$$

https://en.wikipedia.org/wiki/Wave_function

The Kronecker Product ⊗ (a.k.a. Tensor Product)

- **A recursive matrix operation**
- **Allows one to efficiently work with exponentially large structures**

$$u \otimes v = \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} \otimes \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \begin{pmatrix} u_1 v_1 \\ u_1 v_2 \\ u_2 v_1 \\ u_2 v_2 \\ u_3 v_1 \\ u_3 v_2 \end{pmatrix}$$

$$\mathbf{A} \otimes \mathbf{B}$$

$$\mathbf{A} \otimes \mathbf{B} = \begin{pmatrix} a_{11}\mathbf{B} & \cdots & a_{1n}\mathbf{B} \\ \vdots & \ddots & \vdots \\ a_{m1}\mathbf{B} & \cdots & a_{mn}\mathbf{B} \end{pmatrix}$$

$$A \otimes B = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{pmatrix} \otimes \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix} = \begin{pmatrix} a_{11}b_{11} & a_{11}b_{12} & a_{12}b_{11} & a_{12}b_{12} \\ a_{11}b_{21} & a_{11}b_{22} & a_{12}b_{21} & a_{12}b_{22} \\ a_{21}b_{11} & a_{21}b_{12} & a_{22}b_{11} & a_{22}b_{12} \\ a_{21}b_{21} & a_{21}b_{22} & a_{22}b_{21} & a_{22}b_{22} \\ a_{31}b_{11} & a_{31}b_{12} & a_{32}b_{11} & a_{32}b_{12} \\ a_{31}b_{21} & a_{31}b_{22} & a_{32}b_{21} & a_{32}b_{22} \end{pmatrix}$$

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \otimes \begin{pmatrix} 0 & 5 \\ 6 & 7 \end{pmatrix} = \begin{pmatrix} 1\begin{pmatrix} 0 & 5 \\ 6 & 7 \end{pmatrix} & 2\begin{pmatrix} 0 & 5 \\ 6 & 7 \end{pmatrix} \\ 3\begin{pmatrix} 0 & 5 \\ 6 & 7 \end{pmatrix} & 4\begin{pmatrix} 0 & 5 \\ 6 & 7 \end{pmatrix} \end{pmatrix} = \begin{pmatrix} 0 & 5 & 0 & 10 \\ 6 & 7 & 12 & 14 \\ 0 & 15 & 0 & 20 \\ 18 & 21 & 24 & 28 \end{pmatrix}$$

Dirac Notation (Braket) $\langle | \rangle$

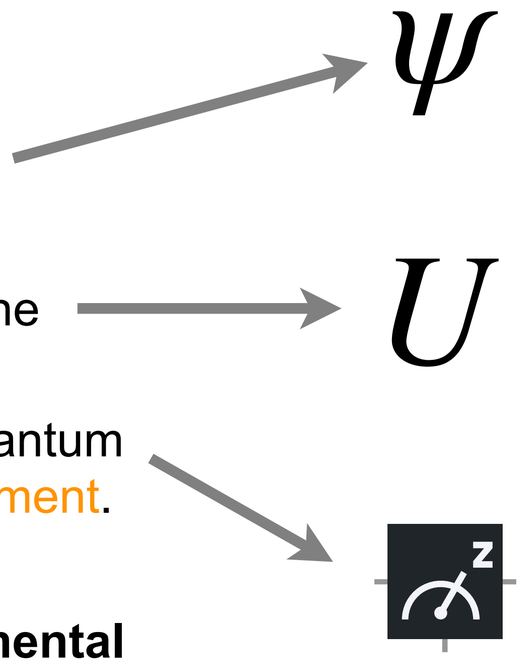
- **Given a complex Euclidean space $\mathbb{S} \equiv \mathbb{C}^n$,**
  - $|\psi\rangle \in \mathbb{S}$ denotes a column vector
  - $\langle\psi| \in \mathbb{S}$ denotes a row vector that is the conjugate transpose of $|\psi\rangle \in \mathbb{S}$
    - i.e., $\langle\psi| = |\psi\rangle^\dagger$

- $\langle\psi|$ **is called a *bra***
- $|\psi\rangle$ **is called a *ket***
- **A sequence of matrix operations (applied right to left) looks like,**
  - $\langle\psi_2|\ldots|C|B|A|\psi_1\rangle$

- **What does this have to do with Quantum Computing?**
  - Quantum computers works with large complex Euclidean spaces $(\mathbb{C}^n)$

What is a Gate-Based Quantum Computer?

- **Three Key Components**
  - The quantum computer has a state (ψ) that is contained in a quantum register and is initialized in a predefined way.
  - The state evolves by applying operations (U) specified in advance in the form of an algorithm.
  - At the end of the computation, some information on the state of the quantum register is obtained by means of a special operation, called a measurement.



- **This is the circuit model, and it is similar to a Turing machine. Fundamental results regarding universal quantum computers are presented in [Deutsch, 1985, Yao, 1993, Bernstein and Vazirani, 1997]**

- **There exists an alternative model of computation, called the adiabatic model. It is equivalent to the circuit model [Aharonov et al., 2008]**

State of a Quantum Computer

(Superposition and Entanglement)

The Standard Basis of *One Qubit*

- **The qubit's state is a *unit vector* of 2 complex numbers (i.e., $\mathbb{C}^2$)**
- **The standard basis is $|0\rangle$ for the first entry and $|1\rangle$ for the second entry**

standard basis

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

In {0} state

$$|\psi\rangle = \begin{matrix} |0\rangle \\ |1\rangle \end{matrix} \begin{pmatrix} 1.0 + 0.0i \\ 0.0 + 0.0i \end{pmatrix}$$

split between {0,1} states

$$|\psi\rangle = \begin{matrix} |0\rangle \\ |1\rangle \end{matrix} \begin{pmatrix} \sqrt{0.7} + 0.0i \\ \sqrt{0.3} + 0.0i \end{pmatrix}$$

unit vector requirement

$$\langle \psi | \psi \rangle = 1$$

Reading a Quantum State

- **When you "read" a qubit, you get a basis state 0 or 1** ***(not the $\mathbb{C}^2$ state vector!)***
- **The amplitudes of the Quantum State (ψ) determine's the probability distribution of the basis states that you will observe**

$$p_s = \text{diagonal}(|\psi\rangle\langle\psi|)$$

$$|\psi\rangle = \begin{matrix}|0\rangle\\|1\rangle\end{matrix}\begin{pmatrix}\sqrt{0.5}+0.0i\\\sqrt{0.5}+0.0i\end{pmatrix} \quad p_s = \begin{matrix}|0\rangle\\|1\rangle\end{matrix}\begin{pmatrix}0.5\\0.5\end{pmatrix} \qquad |\psi\rangle = \begin{matrix}|0\rangle\\|1\rangle\end{matrix}\begin{pmatrix}0.5+0.25i\\\sqrt{2}/4+0.75i\end{pmatrix} \quad p_s = \begin{matrix}|0\rangle\\|1\rangle\end{matrix}\begin{pmatrix}5/16\\11/16\end{pmatrix}$$

The goal of a quantum algorithm is most often to
put 100% of the state probability into the "right" basis vector
(e.g., optimal solution to a combinatorial problem)
Then reading the solution out of the quantum computer is easy

How does this generalize to Multiple Qubits?

- **The "Power" of Quantum Computing**
  - The quantum state vector grows exponentially with the number of qubits (this is why the Kronecker product is so useful in quantum computing)

- **Compute over exponentially large vectors!**

$$2^{|Qubits|} \downarrow \quad |\psi\rangle = \underset{q_1}{\begin{pmatrix} 1.0+0.0i \\ 0.0+0.0i \end{pmatrix}} \underset{q_1,q_2}{\begin{pmatrix} 1.0+0.0i \\ 0.0+0.0i \\ 0.0+0.0i \\ 0.0+0.0i \end{pmatrix}} \underset{q_1,q_2,q_3}{\begin{pmatrix} 1.0+0.0i \\ 0.0+0.0i \\ 0.0+0.0i \\ 0.0+0.0i \\ 0.0+0.0i \\ 0.0+0.0i \\ 0.0+0.0i \\ 0.0+0.0i \end{pmatrix}}$$

Two Qubit Example

- **The quantum state space grows as $(\mathbb{C}^2)^{\otimes q}$ (q is the number of qubits)**
- **For two qubits we have $(\mathbb{C}^2)^{\otimes 2} = \mathbb{C}^2 \otimes \mathbb{C}^2 = \mathbb{C}^4$ with the following 4 basis elements**

Recall these are Complex values

$$|0\rangle = |0\rangle \otimes |0\rangle = |00\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

$$|1\rangle = |0\rangle \otimes |1\rangle = |01\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

$$|2\rangle = |1\rangle \otimes |0\rangle = |10\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$

$$|3\rangle = |1\rangle \otimes |1\rangle = |11\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

2 qubit quantum state with basis vectors

$$|\psi\rangle = \begin{matrix} |00\rangle \\ |01\rangle \\ |10\rangle \\ |11\rangle \end{matrix} \begin{pmatrix} 0.0 + 0.0i \\ 1.0 + 0.0i \\ 0.0 + 0.0i \\ 0.0 + 0.0i \end{pmatrix}$$

What is a *Product State*?

- **Special quantum states that can be written as the Kronecker product of other states**
  - Intuition - even though these two quantum states are part of one large quantum system, they are separable and do not impact one another.
  - i.e., measurement outcomes of each subsystem are independent

General Form

$$|\psi\rangle = |\psi_1\rangle \otimes |\psi_2\rangle$$

Where is the special quantum stuff that we hear about?

Prepare one Specific Basis State

$$\overset{q_1}{|1\rangle} \otimes \overset{q_2}{|0\rangle} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$

Like, Superposition and Entanglement!

Prepare the uniform probably state

$$\overset{q_1}{\left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right)} \otimes \overset{q_2}{\left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right)} = \begin{pmatrix} \frac{1}{\sqrt{2}} + 0.0i \\ \frac{1}{\sqrt{2}} + 0.0i \end{pmatrix} \otimes \begin{pmatrix} \frac{1}{\sqrt{2}} + 0.0i \\ \frac{1}{\sqrt{2}} + 0.0i \end{pmatrix} = \begin{pmatrix} 0.5 + 0.0i \\ 0.5 + 0.0i \\ 0.5 + 0.0i \\ 0.5 + 0.0i \end{pmatrix}$$

What is *Superposition*?

- **The quantum system is in a linear combination of basis states**
  - i.e., if you observe the system sometimes you will see one state and other times you will see a different state

$$p_s = \text{diagonal}(|\psi\rangle\langle\psi|)$$

The "uniform" superposition state on 1 qubit

$$|\psi\rangle = \begin{matrix}|0\rangle \\ |1\rangle\end{matrix}\begin{pmatrix}\sqrt{0.5}+0.0i \\ \sqrt{0.5}+0.0i\end{pmatrix} \quad p_s = \begin{matrix}|0\rangle \\ |1\rangle\end{matrix}\begin{pmatrix}0.5 \\ 0.5\end{pmatrix}$$

A more complex superposition state on 2 qubits

$$|\psi\rangle = \begin{matrix}|00\rangle \\ |01\rangle \\ |10\rangle \\ |11\rangle\end{matrix}\begin{pmatrix}0.25+0.50i \\ 0.25-0.25i \\ 0.50+0.50i \\ 0.00-0.25i\end{pmatrix} \quad p_s = \begin{matrix}|00\rangle \\ |01\rangle \\ |10\rangle \\ |11\rangle\end{matrix}\begin{pmatrix}0.3125 \\ 0.1250 \\ 0.5000 \\ 0.0625\end{pmatrix}$$

What is *Entanglement*?

- **A quantum system state that cannot be expressed as product state is entangled**
  - i.e., the observed values of each qubit (when measured) are correlated
  - requires at least 2 qubits

"Bell Pair"
a.k.a., maximally entangled state

$$|\psi\rangle = \begin{matrix} \\ |00\rangle \\ |01\rangle \\ |10\rangle \\ |11\rangle \end{matrix} \begin{pmatrix} \sqrt{0.5} \\ 0 \\ 0 \\ \sqrt{0.5} \end{pmatrix} \qquad p_s = \begin{matrix} q_1, q_2 \\ |00\rangle \\ |01\rangle \\ |10\rangle \\ |11\rangle \end{matrix} \begin{pmatrix} 0.5 \\ 0 \\ 0 \\ 0.5 \end{pmatrix}$$

If you read "0" on qubit 1, then you will also read "0" on qubit 2

There is no setting of the constants c_1 to c_8 such that the Kronecker product produces the Bell state!

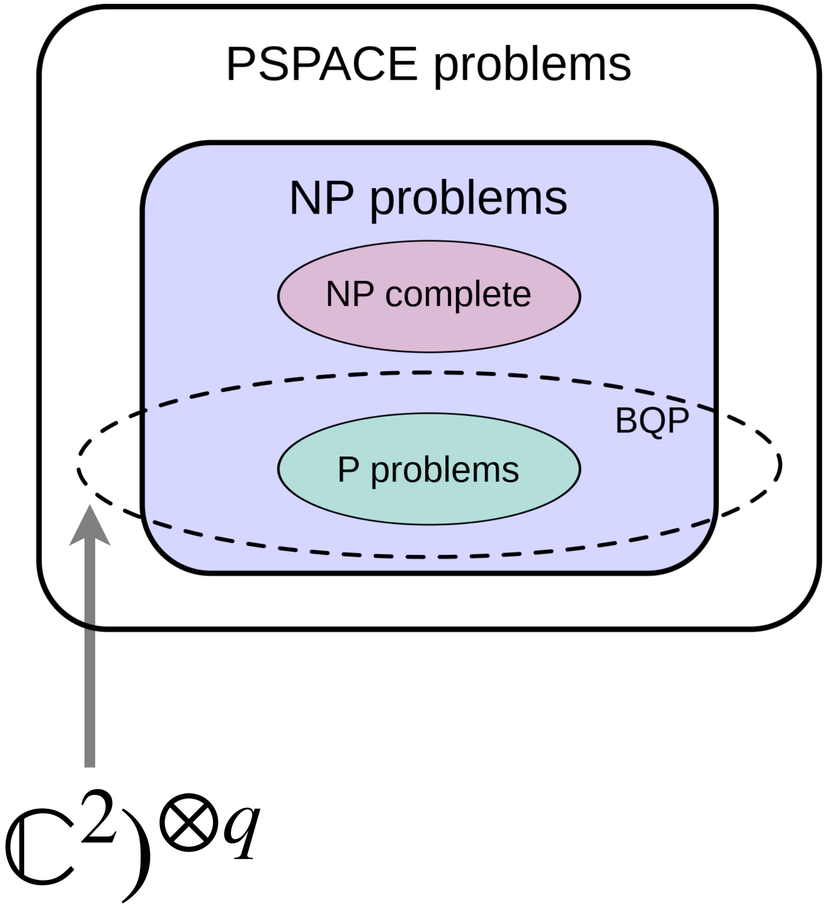
$$\begin{matrix} q_1 \\ \begin{pmatrix} c_1 + c_2 i \\ c_3 + c_4 i \end{pmatrix} \end{matrix} \otimes \begin{matrix} q_2 \\ \begin{pmatrix} c_5 + c_6 i \\ c_7 + c_8 i \end{pmatrix} \end{matrix} \neq \begin{pmatrix} \sqrt{0.5} + 0.0i \\ 0.0 + 0.0i \\ 0.0 + 0.0i \\ \sqrt{0.5} + 0.0i \end{pmatrix}$$

Superposition and Entanglement, so what?

- **The key ingredients for a gate-based quantum algorithm to be faster than a classical algorithm**
  - Leverage the exponentially large space that is available (i.e., $(\mathbb{C}^2)^{\otimes q}$)
  - Manage a delicate interplay of superposition and entanglement in the quantum state

- **If these are not leveraged, most likely a classical computer can do the job at hand**

https://en.wikipedia.org/wiki/BQP



Quantum Operations
(Gates)

$$U$$

Quantum Gates

- **Quantum Operations (U) are called *Gates***

Quantum Gate

$$U$$

  - A square matrix of complex numbers that are applied to the Quantum State (ψ)
  - Theory of Quantum Mechanics says this operation must be unitary

Unitary Property

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

  - Given the state vector has size $\mathbb{C}^{2^q}$ the gates have size $\mathbb{C}^{2^q \times 2^q}$, where q is the number of qubits

- **Important Consequences**
  - Quantum operations are linear
  - Quantum operations are reversible

While these properties may seem to be extremely restrictive, a quantum computer is Turing-complete! [Deutsch, 1985]

Quantum Gates, An 1 Qubit Example

$$\psi_0 = \begin{pmatrix} 0 \\ -1 \end{pmatrix}$$

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

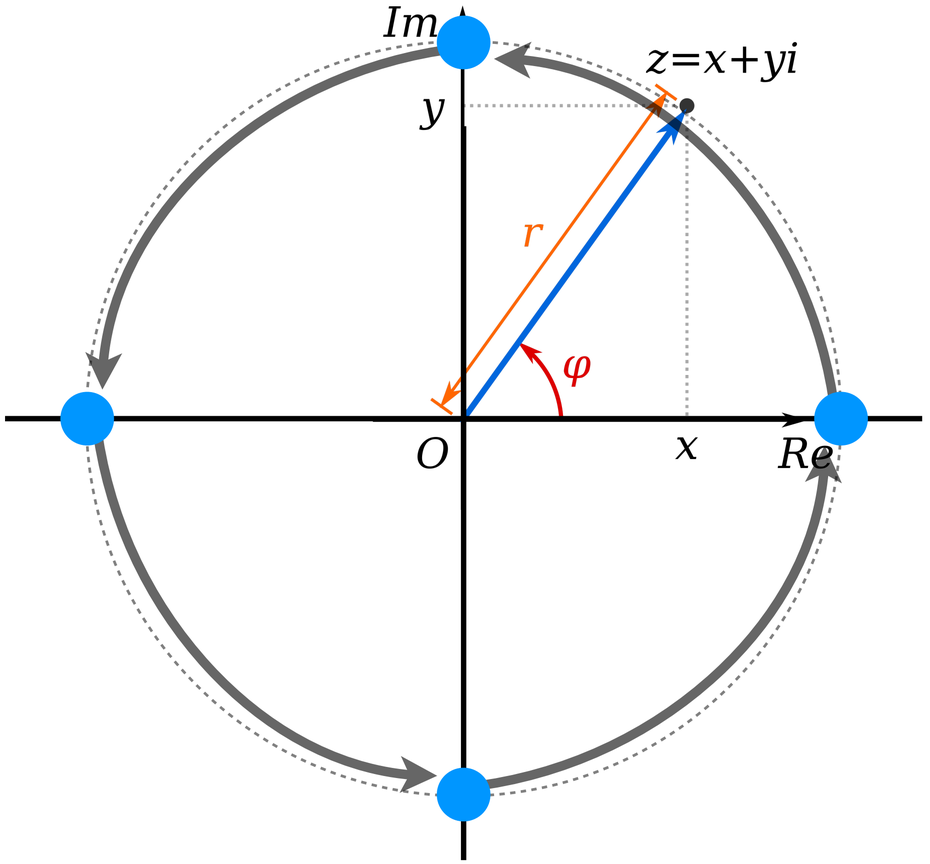
$$|\psi_0\rangle = \begin{pmatrix} 0 \\ -1 \end{pmatrix}$$

$$U|\psi_0\rangle = \begin{pmatrix} 0 \\ -i \end{pmatrix}$$

$$U|U|\psi_0\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

$$U|U|U|\psi_0\rangle = \begin{pmatrix} 0 \\ i \end{pmatrix}$$

$$U|U|U|U|\psi_0\rangle = \begin{pmatrix} 0 \\ -1 \end{pmatrix}$$



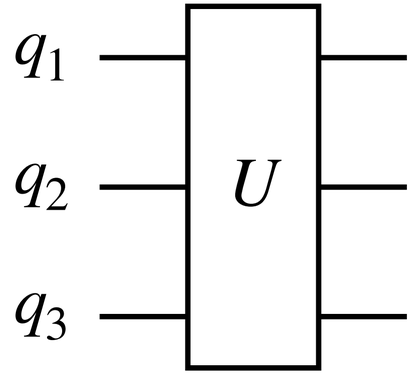
Quantum Gates, Notation

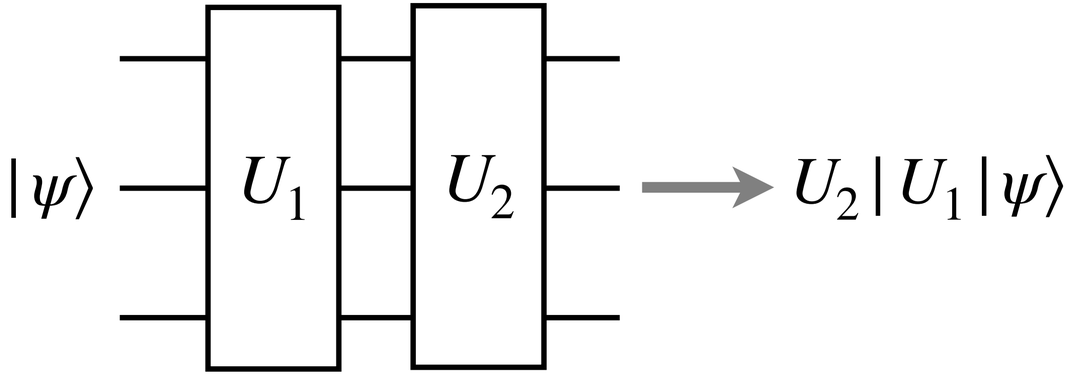
- **Quantum programs are often drawn as a score over a number of qubits. Gates appear as blocks in the score**
  - Note, in this 3 qubit example the quantum state has size $\mathbb{C}^8$ the gate U is an $\mathbb{C}^{8\times8}$ matrix

- **Scores are read from the left to right**
  - *But written right to left*

- **A gate-based quantum program consists of repeated applications of unitary gates**
  - You can also have measurements, but details are outside the scope of this introduction

Quantum Score





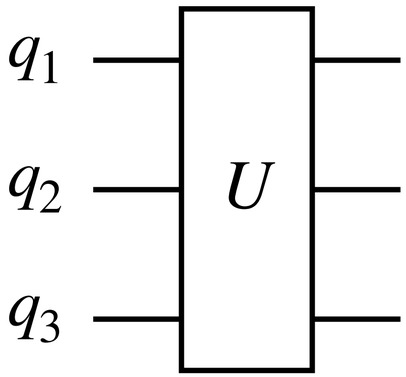
Quantum Program

$$U_n|\ldots|U_3|U_2|U_1|\psi\rangle$$

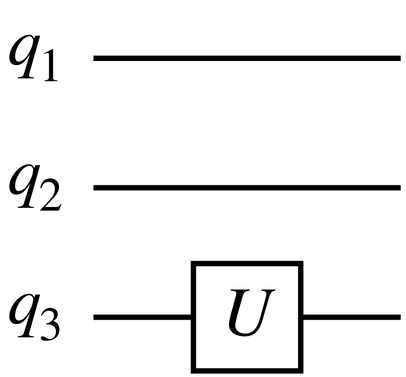
Quantum Gates on Few Qubits

- **Gates can be applied to subsets of qubits**
  - Identity matrices expand the operation to be the right size ($\mathbb{C}^{2^q \times 2^q}$)

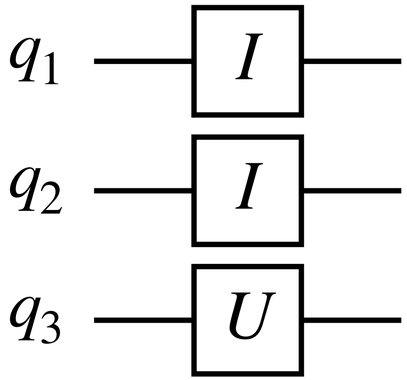
General Quantum Gate



Single-Qubit Gate (viz)



Single-Qubit Gate (math)



Generic Single-Qubit Gate

$$U(\theta, \phi, \lambda) =$$

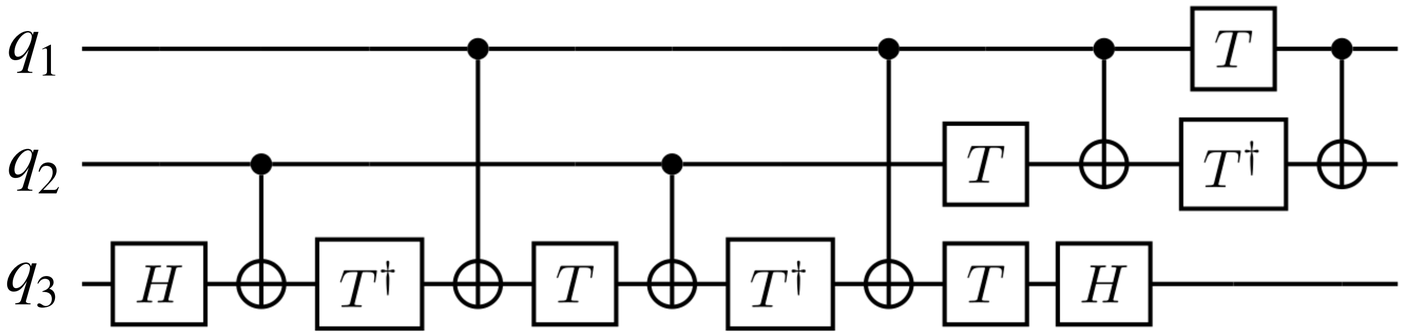
$$\begin{pmatrix} e^{-i(\phi+\lambda)/2}\cos(\theta/2) & -e^{-i(\phi-\lambda)/2}\sin(\theta/2) \\ e^{i(\phi-\lambda)/2}\sin(\theta/2) & e^{i(\phi+\lambda)/2}\cos(\theta/2) \end{pmatrix}$$

Math

$$(I \otimes I \otimes U)\,|\psi\rangle$$

Quantum Computer Program

- **Quantum Program (score)**
  - Most commonly shown as a sequence of one and two qubit gates



Common Single Qubit Gates

Operator	Gate(s)	Matrix
Pauli-X (X)	X	$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$
Pauli-Y (Y)	Y	$\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$
Pauli-Z (Z)	Z	$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$
Hadamard (H)	H	$\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$
Phase (S, P)	S	$\begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$
$\pi/8$ (T)	T	$\begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}$

Common Two-Qubit Gate

Controlled Not (CNOT, CX)		$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$

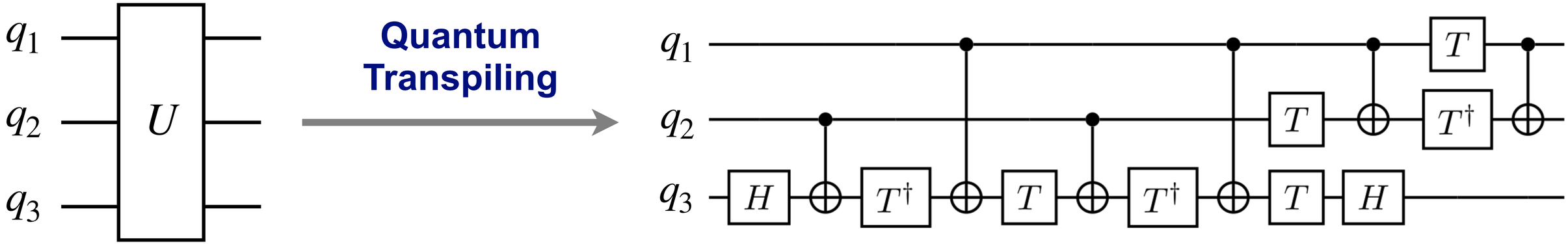
https://en.wikipedia.org/wiki/Quantum_logic_gate

https://en.wikipedia.org/wiki/Quantum_computing

Quantum Computer Program Transpiling

- **In practice, one cannot directly implement one big unitary transform (U)**
  - How would you write down a matrix with $\mathbb{C}^{2^q \times 2^q}$ entries?
  - Quantum computers only implement a limited set of gates (usually only on 1 or 2 qubits)

- **This necessitates a process of transpiling the unitary that one wants to implement into a sequence of gates that are supported by the hardware**



Quantum Gate Sets

Some Common Quantum Gates

- **Quantum programs will be represented as a sequence of gates operating on 1 or 2 qubits**
  - What are common gates?
  - What is a sufficiently expressive gate set?

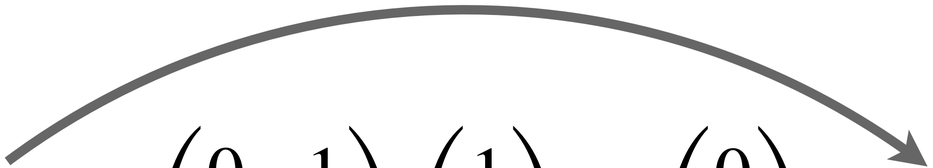
- **Pauli Gates (I, X, Y, Z)**
  - Single qubit gates $\mathbb{C}^{2\times 2}$, also are a basis for single qubits

$$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

The X Gate, One Example

- **The X gate swaps the entries of the quantum state and is sometimes called the quantum equivalent of a classical NOT gate.**

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



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

What about *Superposition*?!?

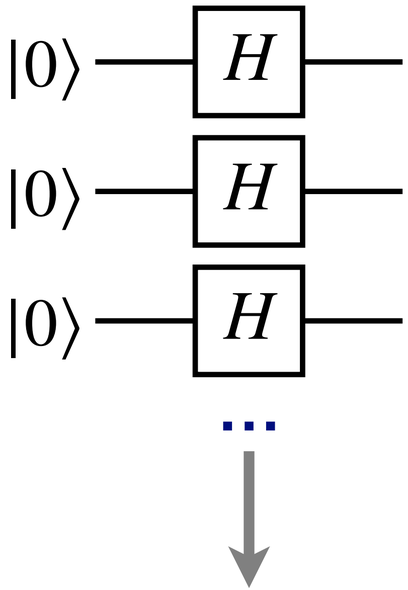
- **The Hadamard Gate (H) splits probability across basis states**

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

$$H|0\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{matrix} |0\rangle \\ |1\rangle \end{matrix} \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix} = \frac{1}{\sqrt{2}} \left(|0\rangle + |1\rangle \right)$$

$$H|1\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{matrix} |0\rangle \\ |1\rangle \end{matrix} \begin{pmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{pmatrix} = \frac{1}{\sqrt{2}} \left(|0\rangle - |1\rangle \right)$$

Layer of Hadamard Gates



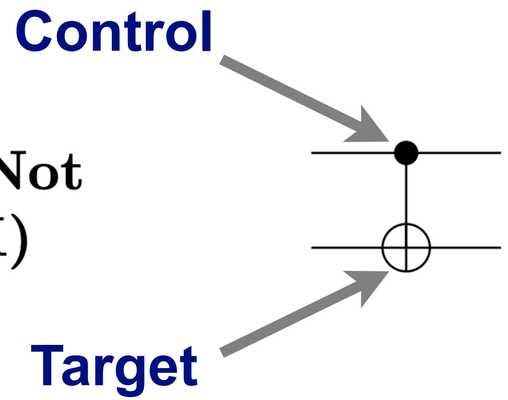
Uniform Superposition State

$$\frac{1}{\sqrt{2^q}} \sum_{x_1,\ldots,x_q} |x_1\rangle \otimes \ldots \otimes |x_q\rangle$$

What about *Entanglement*?!?

- **Single qubit gates are not sufficient to generate entanglement between quantum states!**
  - We need a two-qubit gate for that! $\mathbb{C}^{4\times4}$

- **Controlled Not (CNOT) is one of the most common**
  - Also known as CX, C for control, X is the quantum versions of NOT
  - If the control qubit is 0 then I is applied to the target qubit
  - If the control qubit is 1 then X is applied to the target qubit



Controlled Not (CNOT, CX)

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

Control Not on Basis States

$$CX_{12}|00\rangle = |00\rangle \qquad CX_{12}|01\rangle = |01\rangle$$

$$CX_{12}|10\rangle = |11\rangle \qquad CX_{12}|11\rangle = |10\rangle$$

Los Alamos NATIONAL LABORATORY

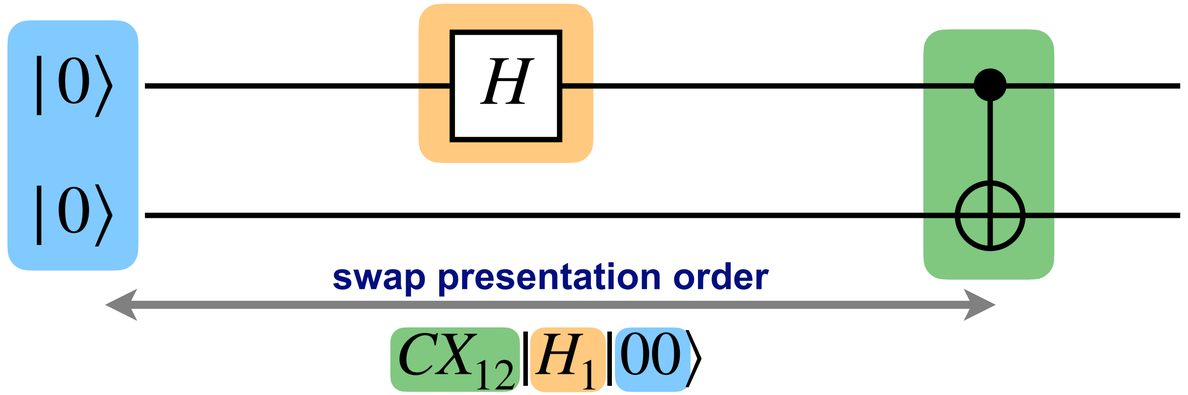
Preparing a Bell Pair, Maximally Entangled State

- **How to program (i.e., prepare) this quantum state using quantum gates?**

Bell State

$$|\psi\rangle = \begin{matrix} \\ |00\rangle \\ |01\rangle \\ |10\rangle \\ |11\rangle \end{matrix} \begin{matrix} q_1, q_2 \\ \begin{pmatrix} \sqrt{0.5} \\ 0 \\ 0 \\ \sqrt{0.5} \end{pmatrix} \end{matrix}$$

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

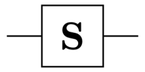
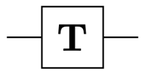
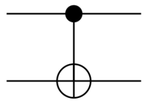


$$|\psi\rangle = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1/\sqrt{2} & 0 & 1/\sqrt{2} & 0 \\ 0 & 1/\sqrt{2} & 0 & 1/\sqrt{2} \\ 1/\sqrt{2} & 0 & -1/\sqrt{2} & 0 \\ 0 & 1/\sqrt{2} & 0 & -1/\sqrt{2} \end{pmatrix} \begin{matrix} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \\ H_1 \otimes I_2 \end{matrix} \begin{matrix} |00\rangle \\ |01\rangle \\ |10\rangle \\ |11\rangle \end{matrix}$$

Universal Quantum Gate Sets

- **How many types of gates do you need so that your quantum computer can implement any quantum computation efficiently?**
  - There are many possible options
  - All commercial gate-based quantum computers implement one

Clifford + T

Hadamard (H)		$\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}$
Phase (S, P)		$\begin{bmatrix}1 & 0\\ 0 & i\end{bmatrix}$
$\pi/8$ (T)		$\begin{bmatrix}1 & 0\\ 0 & e^{i\pi/4}\end{bmatrix}$
Controlled Not (CNOT, CX)		$\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & 1 & 0\end{bmatrix}$

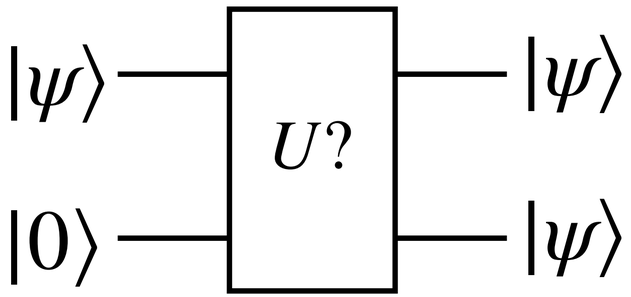
https://en.wikipedia.org/wiki/Quantum_logic_gate#Universal_quantum_gates

Challenges and Uses of Quantum Computing

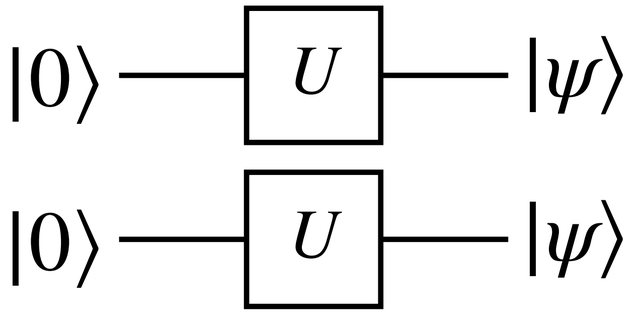
Quantum States Cannot be Copied

- **Imagine you have an *arbitrary* quantum state ψ and you would like a copy of it.**
  - You need to design unitary matrix U that takes in one state and outputs two

- **This is not possible! No such U exists. See the "no-cloning theorem"**



What can you do in practice?



The Challenge of Quantum Computing

- **What useful computations can you do with unitary transforms on exponentially large matrices?**

- **Only a few great answers… *so far!***
  - Simulate Quantum Systems (solve Schrödinger Equation)
  - Factor Integers (Shor's Algorithm)
  - Unstructured Search (Grover's Algorithm)
  - Solving Linear Systems of Equations (HHL)
  - Simulating Large Collections of Oscillating Springs (arXiv:2303.13012)

"Quantum Algorithm Implementations for Beginners"

arXiv:1804.03719

Quantum Computing and Optimization (Part 2)

Some Quantum Algorithms (there are not that many…)

- **Solving Linear Systems (solve A*x* = b)**
  - Maybe can accelerate IPMs? Lehigh leads the way (speaker Tamás Terlaky, arXiv:2205.01220)

- **Solving SDPs** (arXiv:1710.02581)
  - Will not be covered in this masterclass (but ask speaker Xiaodi Wu about it during the breaks!)

- **Combinatorial Search (e.g., BnB, CP)** (arXiv:1810.05582)
  - Speaker Ashley Montanaro has been a trail blazer in this topic

Briefly touch on what we can expect in NP-Hard use cases

- **Quantum Annealing (QA)**
  - ***Heuristic*** for solving ***combinatorial optimization***; no hyper parameters (in theory)
  - Covered by speaker Zachary Morrell

- **Quantum Approximate Optimization Algorithm (QAOA)**
  - ***Meta-heuristic*** for solving ***combinatorial optimization***; hyper parameter search is hard
  - Covered by speaker Andreas Bärtschi

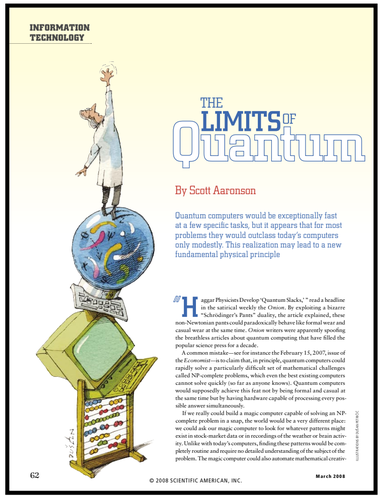
What can we hope for from Quantum-Accelerated Combinatorial Optimization?

- ***"The Limits of Quantum Computers"***
  Scott Aaronson, Scientific American, 2008

Quantum computers *do not solve* NP-Complete problems in polynomial time!

1) Provable quadratic speedups of exponential runtimes are usually possible. Maybe more, under special circumstances…

2) Similar to classical heuristics, there is no good theories for the performance of Quantum heuristics. Performance benchmarking on real hardware is essential.


INFORMATION TECHNOLOGY

THE LIMITS OF Quantum

By Scott Aaronson

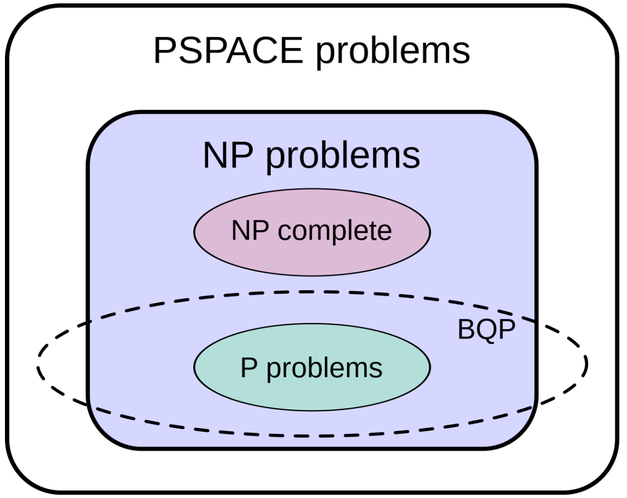
Quantum computers would be exceptionally fast at a few specific tasks, but it appears that for most problems they would outclass today's computers only modestly. This realization may lead to a new fundamental physical principle

"Harvard Physicists Develop 'Quantum Slacks,'" read a headline in the satirical weekly the *Onion*. By exploiting a bizarre "Schrödinger's Pants" duality, the article explained, these non-Newtonian pants could paradoxically behave like formal wear and casual wear at the same time. *Onion* writers were apparently spoofing the breathless articles about quantum computing that have filled the popular science press for a decade.

A common mistake—see for instance the February 15, 2007, issue of the *Economist*—is to claim that, in principle, quantum computers could rapidly solve a particularly difficult set of mathematical challenges called NP-complete problems, which even the best existing computers cannot solve quickly (so far as anyone knows). Quantum computers would supposedly achieve this feat not by being formal and casual at the same time but by having hardware capable of processing every possible answer simultaneously.

If we really could build a magic computer capable of solving an NP-complete problem in a snap, the world would be a very different place: we could ask our magic computer to look for whatever patterns might exist in stock-market data or in recordings of the weather or brain activity. Unlike with today's computers, finding these patterns would be completely routine and require no detailed understanding of the subject of the problem. The magic computer could also automate mathematical creativ-

62 © 2008 SCIENTIFIC AMERICAN, INC. March 2008

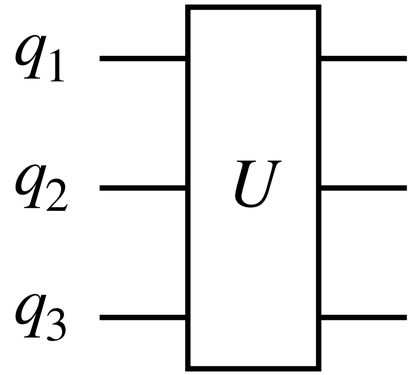


https://en.wikipedia.org/wiki/BQP

Why Unconstrained Combinatorial comes up a lot in Quantum Computing?

convert this *classical function* $E(x)$ into a unitary matrix

$$\min_x: E(x) = \sum_i c_i x_i + \sum_{i,j} c_{ij} x_i x_j + \sum_{i,j,k} c_{ijk} x_i x_j x_k + \ldots$$



subject to: $x_i \in \{0,1\}, \forall\, i \in \{1,\ldots,n\}$

Solution space of the optimization problem, maps directly onto the basis vectors of the quantum state

$$|\psi\rangle = \begin{matrix} |00\rangle \\ |01\rangle \\ |10\rangle \\ |11\rangle \end{matrix} \begin{pmatrix} 0.0 + 0.0i \\ 1.0 + 0.0i \\ 0.0 + 0.0i \\ 0.0 + 0.0i \end{pmatrix}$$

Why QUBOs for Quantum Computing?

- **Quadratic Unconstrained Binary Optimization (QUBO)**

QUBO

$$\min_x: E(x) = \sum_i x_i + \sum_{i,j} x_i x_j$$

subject to: $x_i \in \{0,1\}, \forall\, i \in \{1,\ldots,n\}$

Can be implemented with 1 and 2 qubit gates

These are “easy” for current quantum hardware platforms

Why Ising Models for Quantum Computing?

- **The Ising Model is a popular model in Classical Physics**
  - The Traverse Ising Model is a popular Quantum Physics

- **The classical Ising Model is equivalent to Quadratic Unconstrained Binary Optimization (QUBO)**

QUBO

$$\min : \sum_{i,j \in \mathcal{E}} \boldsymbol{c}_{ij} x_i x_j + \sum_{i \in \mathcal{N}} \boldsymbol{c}_i x_i$$

s.t.

$$x_i \in \{0, 1\} \; \forall i \in \mathcal{N}$$

$\sigma_i = 2x_i - 1$

$x_i = \frac{\sigma_i + 1}{2}$

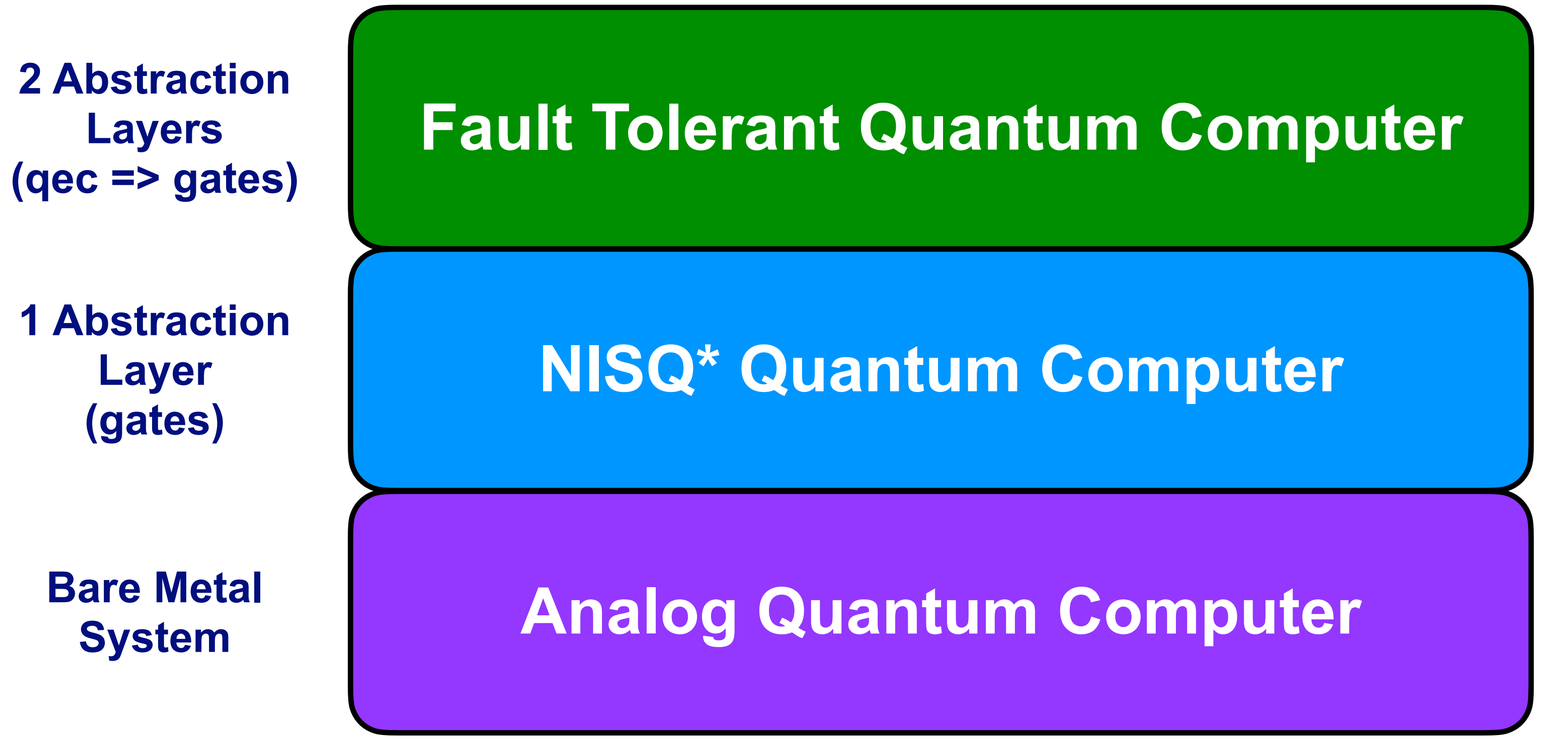
Ising Model

$$\min : \sum_{i \in \mathcal{N}} \boldsymbol{c}_i \sigma_i + \sum_{i,j \in \mathcal{E}} \boldsymbol{c}_{ij} \sigma_i \sigma_j$$

$$\text{s.t.:} \; \sigma_i \in \{-1, 1\} \; \forall i \in \mathcal{N}$$

State of Quantum Computing Hardware (Part 3)

Types of Quantum Computation



* NISQ = Noisy Intermediate Scale Quantum
arXiv:1801.00862

Analog Quantum Computation

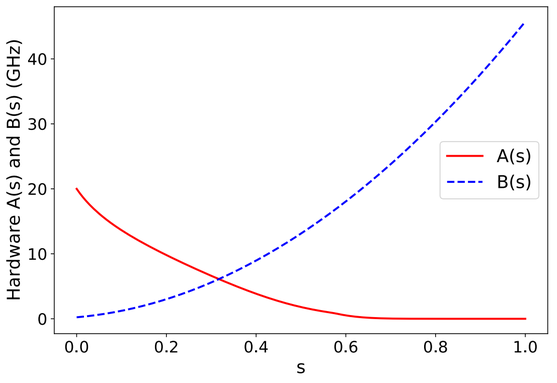
- **What does it do?**
  - Apply control parameters to evolve a real-world quantum system in time
- **Natural use cases**
  - Quantum Simulation (open and closed), Optimization
- **Arguments For**
  - Very efficient use of quantum hardware resources
  - Very fast (no abstraction layers)
- **Arguments Against**
  - Highly specialized to specific applications (i.e. limited Hamiltonian options)
  - Hardware noise can be limiting

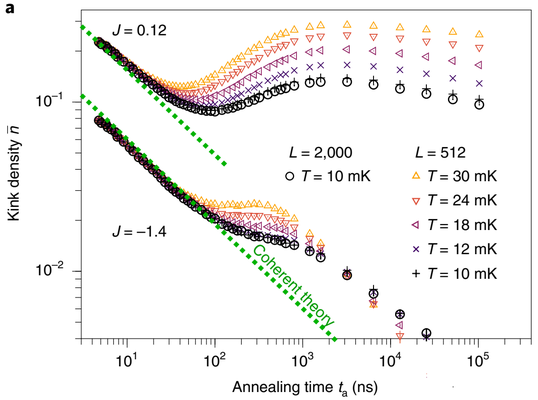
Solve this ODE

$$t \in [0,T], \left|\Psi_0\right\rangle$$

Evolution Time **Initial State**

$$i\frac{d}{dt}\left|\Psi(t)\right\rangle = H(t)\left|\Psi(t)\right\rangle$$





$$H_{dwave}(t) = A(t)\left(\sum_i \hat{\sigma}_i^x\right) + B(t)\left(\sum_i h_i\hat{\sigma}_i^z + \sum_{i,j} J_{ij}\hat{\sigma}_i^z\hat{\sigma}_j^z\right)$$

"Coherent quantum annealing in a programmable 2000-qubit Ising chain" arXiv:2202.05847

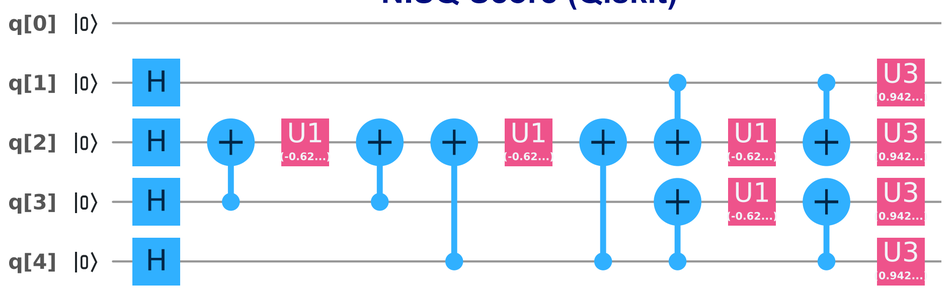
NISQ Gate-Based Computation

- **What does it do?**
  - Apply *imperfect quantum gates* to implement a noisy *universal* quantum computation
  - Note: these usually include arbitrary rotation gates
- **Natural use cases**
  - Quantum Simulation (closed), Variational Quantum Algorithms, Quantum Machine Learning (?)
- **Arguments For**
  - Flexible (all types of quantum computations are in scope)
  - Fast (just 1 abstraction layer)
- **Arguments Against**
  - Hardware noise can be limiting
  - Unclear how to extend beyond qubit coherence time

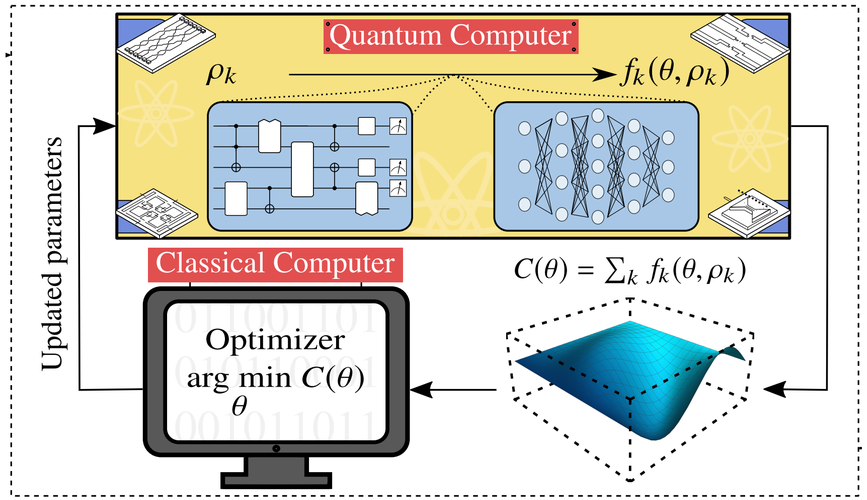
Apply Unitary Matrices (U) to a quantum state (ψ)

$$U_n | \ldots | U_3 | U_2 | U_1 | \psi \rangle$$

NISQ Score (Qiskit)



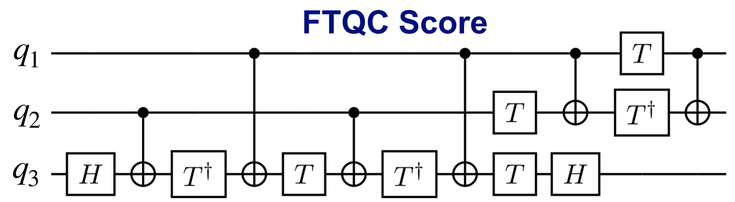
Variational Quantum Algorithm Structure

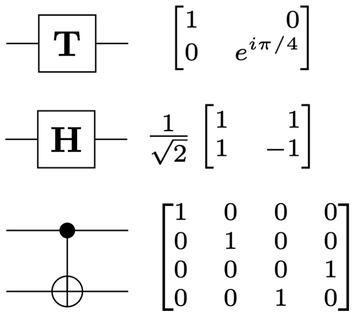


"Variational quantum algorithms" Cerezo et. al.
arXiv:2012.09265

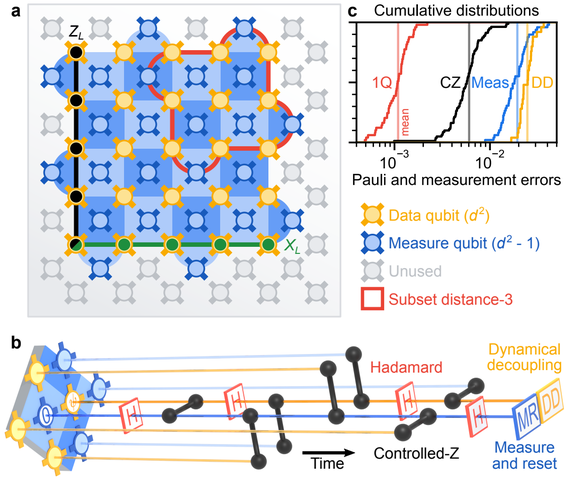
Fault-Tolerant Gate-Based Computation

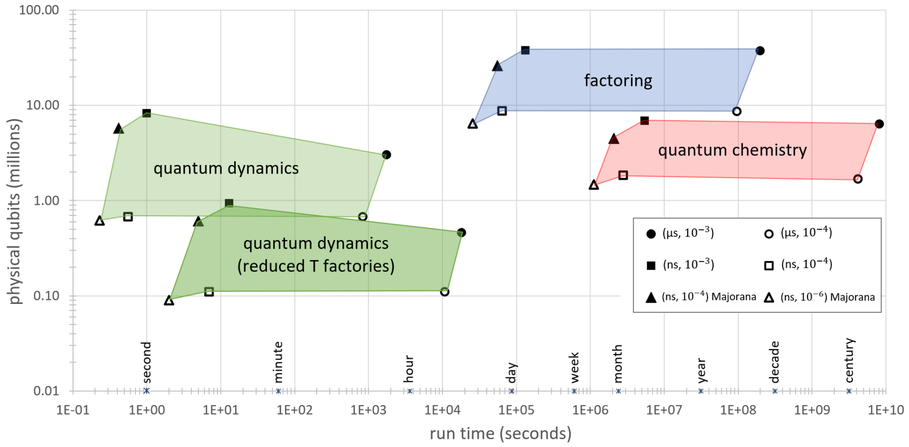
- **What does it do?**
  - Apply quantum *error corrected gates* to implement *perfect universal* quantum computation
  - Note: a restrictive gate set (e.g., Clifford+T)
- **Natural use cases**
  - Quantum Simulation (high-accuracy), Factoring, Linear Systems, Nonlinear Systems (?)
- **Arguments For**
  - Reliable, Algorithms with Proofs
  - Quantum Error Correction (QEC) enables going beyond the limit of qubit coherence time
- **Arguments Against**
  - Requires a LOT of physical qubits (e.g., >10^5)
  - Algorithms require a LOT of gates (e.g., >10^6)
  - Slow, QEC overheads take time (2 abstraction layers)





"Suppressing quantum errors by scaling a surface code logical qubit"
arXiv:2207.06431





"Assessing requirements to scale to practical quantum advantage"
arXiv:2211.07629

Who is doing what?

Three Types of Commercial Quantum Computers

Available Today

Available Soon?

	Noisy Analog	Noisy Gate-Based	Fault-Tolerant Gate-Based
Who?	D-Wave QuEra PASQAL	IBM Q Quantum AI IONQ rigetti XANADU QUANTINUUM qci	PsiQuantum atom computing D-Wave Quantum AI qci ALICE & BOB

Public Roadmaps

IBM Q, PASQAL, Infleqtion, QuEra

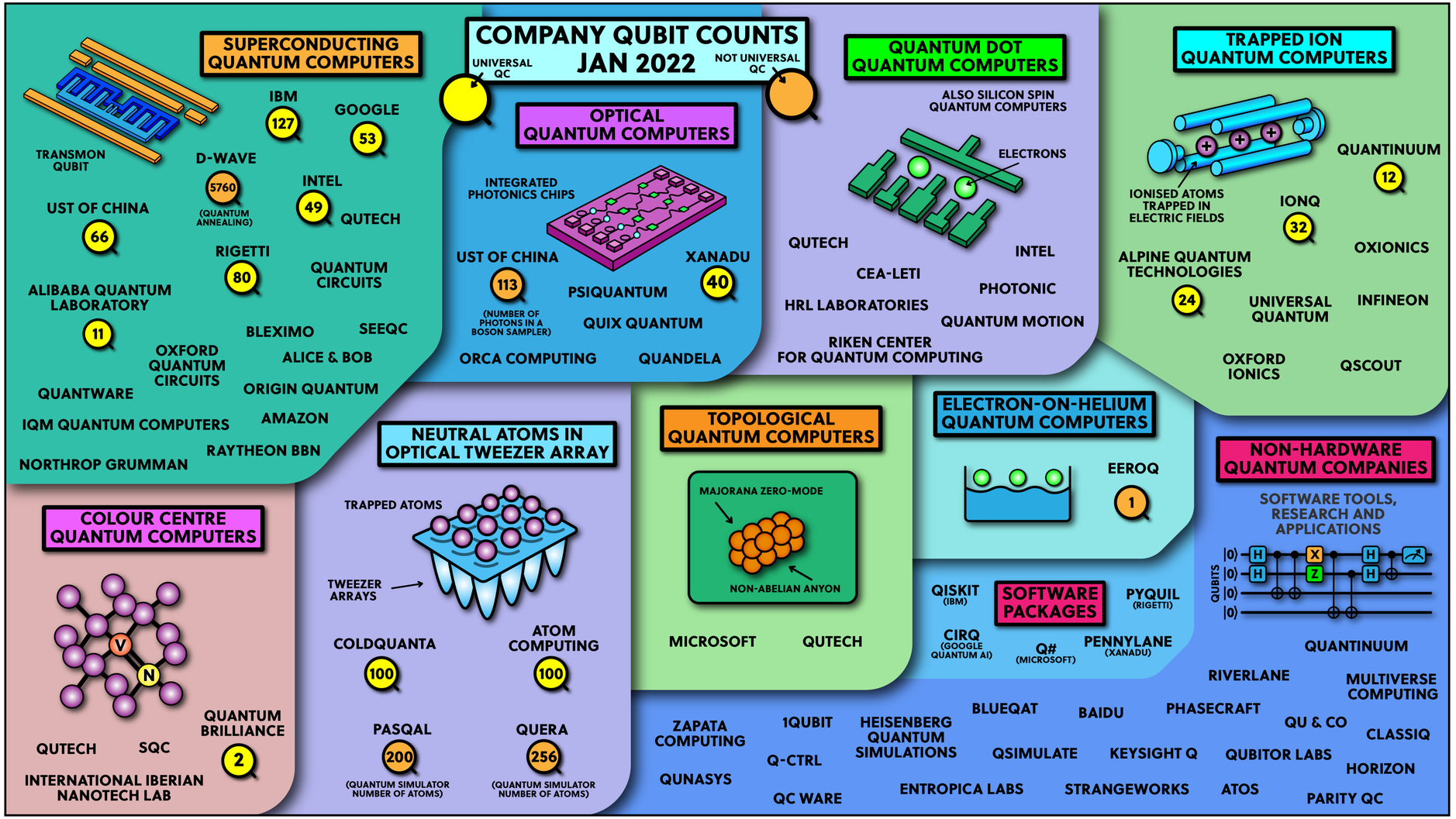
And many others…

FTQC proposed by 2026-2030

Why so many?

There is no “transistor” for quantum computing… yet

Qubit Technologies



Domain of Science - Dominic Walliman

Pros and Cons of *NISQ Hardware* Platforms

Just the vendors I know the best…

IBM Q™ rigetti D:wave IONQ QUANTINUUM IQuEra XANADU

Feature	Superconducting Circuits	Trapped Ions	Neutral Atoms	Photonic	Topological / Quantum Dot / Silicon Spin
Speed	Fast	Slow	Slow	?	??
Noise	Medium	Low	Low	?	??
Scale (qubits demonstrated)	High 500-5000	Low 10-50	Medium 100-1000	?	??
Connectivity	Sparse	High	Sparse	?	??

No clear winner,
co-design of hardware and application maximizes performance

Who will win the Quantum Computing race?

I have no idea!

Everyone has a plausible pitch for why their approach is best

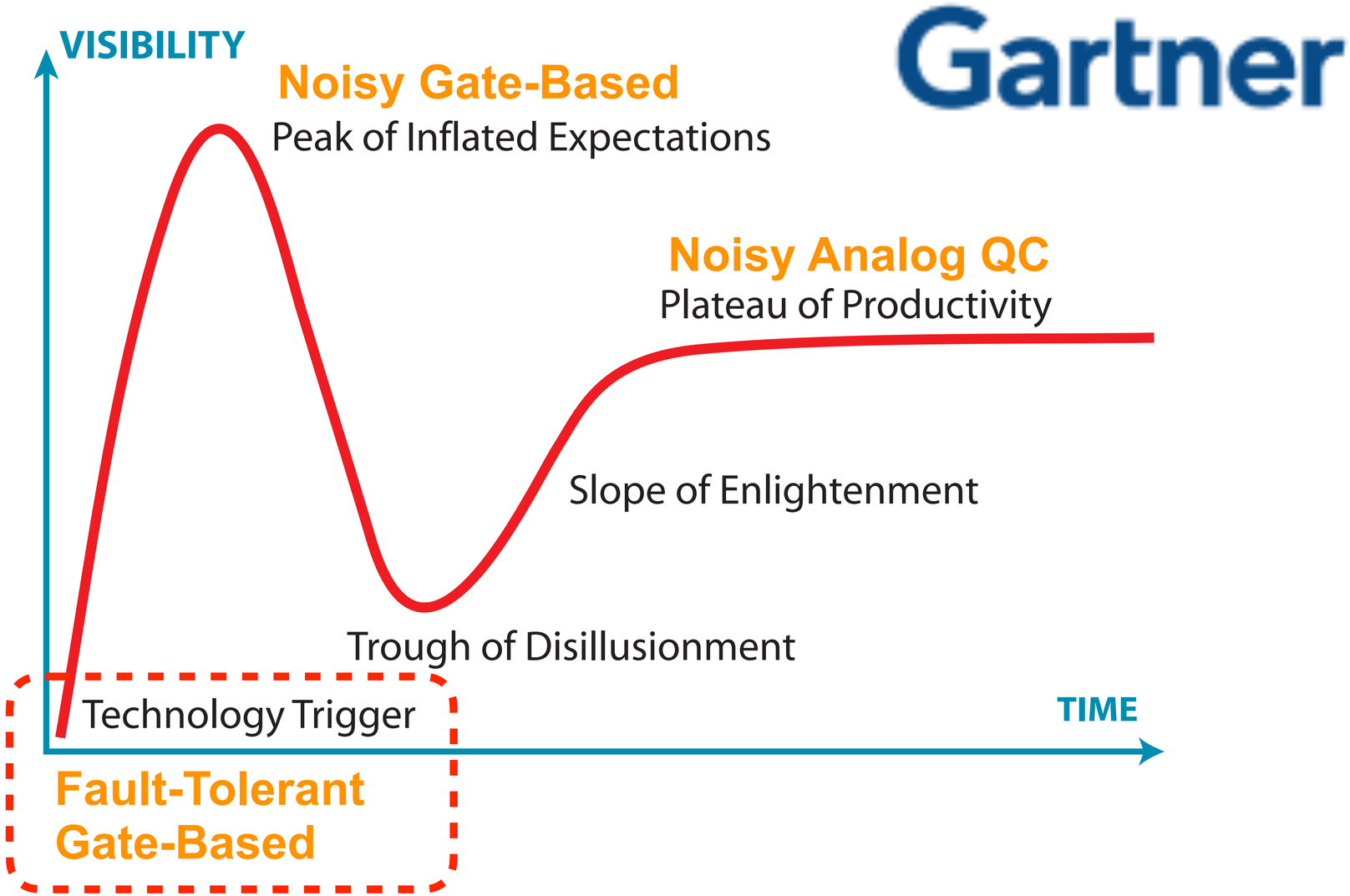
Technology Stage of Quantum Computing Models

Gartner's Technology Hype Cycle

How long before we have a very large and useful Fault-Tolerant quantum computer?

12 months ago most folks would say "10-15 years"

Today it seems much sooner (2026-2030)



Master Class Overview (topics)

Speaker	Topic	Optimization Area	Quantum Computing Model
Carleton Coffrin	QC Foundations	NA	Fault-Tolerant
Ashley Montanaro	Quantum Computing Algorithms for Operations Research and Constraint Programming	QC formal methods, discrete variables (OR/CP)	Fault-Tolerant
Tamás Terlaky	Quantum Computing Algorithms for Interior Point Methods	QC formal methods, continuous variables (IMPs)	Fault-Tolerant
Harsha Nagarajan	Designing Quantum Circuits with Mixed-Integer Polynomial Programming	MINLP for QC, discrete + continuous variables	NISQ and Fault-Tolerant
Zachary Morrell	A Brief Introduction to Quantum Annealing	QC heuristic, discrete variables	Analog
Andreas Bärtschi	A Brief Introduction to the Quantum Approximate Optimization Algorithm (QAOA)	QC meta-heuristic, discrete variables	NISQ
Xiaodi Wu	Quantum Hamiltonian Descent for Non-convex Continuous Optimization	QC heuristic, continuous variables	Analog
David Bernal Neira	Quantum-Classical Hybrid Methods for Optimization	QC hybrid methods, discrete + continuous variables	Analog and NISQ and Fault-Tolerant?

Wrapping Up

I hope I have convinced you that the Quantum Computing model is not magic, and is in fact quite hard to make good use of

Never the less, the rest of this Master Class will show you several interesting use cases for quantum computing in optimization!

Thanks!

Some Public FTQC Roadmaps

- **IBM** IBM Q™
  - 200 logical qubits by 2029
  - 100M Gates
- **Infleqtion**
  - 100 logical qubits by 2028
  - 1-100M Gates
- **QuEra**
  - 100 logical qubits by ***2026***
  - Gates?