Intro to Quantum Computing

TRIUMF summer student seminar #2

Olivia Di Matteo
Quantum Information Science Associate, TRIUMF

16 August 2019

Recap of Wednesday

We saw:

- The motivation behind quantum computing
- Single qubit systems
- Common unitary operations
- A very rushed explanation of measurement

Plan for today

We'll go through:

- A still-rushed-but-hopefully-better explanation of measurement
- Multi-qubit systems and entanglement
- Quantum teleportation
- Quantum advantage
- Overview of current-gen quantum hardware

The computational basis

The most common way to express a quantum state is as a linear combination of $|0\rangle$ and $|1\rangle$, or 'in the computational basis':

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \tag{1}$$

You can check that $|0\rangle$ and $|1\rangle$ form a basis by making sure they orthogonal and normalized:

$$\langle 0|0\rangle = 1, \qquad \langle 0|1\rangle = 0 \tag{2}$$

$$\langle 1|0\rangle = 0, \qquad \langle 1|1\rangle = 1 \tag{3}$$

Measuring in the computational basis

When we measure a state

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \tag{4}$$

in the computational basis, we will find that it is in *either* $|0\rangle$ or $|1\rangle$.

The probability of each outcome is related to the amplitudes:

$$\Pr(0) = |\alpha|^2 \tag{5}$$
$$\Pr(1) = |\beta|^2 \tag{6}$$

More formally, to calculate the probability of a given outcome, we take the $|\cdot|^2$ of the inner product with our state:

$$\Pr(0) = |\langle 0|\psi\rangle|^2 = |\alpha|^2 \tag{7}$$
$$\Pr(1) = |\langle 1|\psi\rangle|^2 = |\beta|^2 \tag{8}$$

The Hadamard basis

The choice of basis is not unique - for qubits there are infinitely many sets of 2 orthonormal vectors. We most commonly use $|0\rangle$ and $|1\rangle$, but another set is[1]

$$|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \tag{9}$$

$$|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \tag{10}$$

$$\tag{11}$$

You can check that

$$\langle +|+\rangle = 1, \qquad \langle +|-\rangle = 0 \tag{12}$$

$$\langle -|+\rangle = 0, \qquad \langle -|-\rangle = 1 \tag{13}$$

[1] This notation is a bit confusing, but the labels + and - refer to the sign between the $|0\rangle$ and $|1\rangle$.

Change of basis

You can express qubit states in other bases by performing a *change of basis*, i.e. re-expressing the basis vectors in terms of other basis vectors. For example, from the definitions of $|+\rangle$ and $|-\rangle$, you can work out that

$$|0\rangle = \frac{1}{\sqrt{2}}\left(|+\rangle + |-\rangle\right), \tag{14}$$

$$|1\rangle = \frac{1}{\sqrt{2}}\left(|+\rangle - |-\rangle\right), \tag{15}$$

$$\tag{16}$$

Then we can rewrite a qubit state in the new basis:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \tag{17}$$

$$= \frac{\alpha}{\sqrt{2}}\left(|+\rangle + |-\rangle\right) + \frac{\beta}{\sqrt{2}}\left(|+\rangle - |-\rangle\right) \tag{18}$$

$$= \left(\frac{\alpha+\beta}{\sqrt{2}}\right)|+\rangle + \left(\frac{\alpha-\beta}{\sqrt{2}}\right)|-\rangle \tag{19}$$

Measuring in the Hadamard basis

To measure in the Hadamard basis, we take the inner product with those basis vectors instead:

$$\Pr(+) = |\langle +|\psi\rangle|^2 \tag{20}$$
$$\Pr(-) = |\langle -|\psi\rangle|^2 \tag{21}$$

If we represent our state in the Hadamard basis,

$$|\psi\rangle = \left(\frac{\alpha+\beta}{\sqrt{2}}\right)|+\rangle + \left(\frac{\alpha-\beta}{\sqrt{2}}\right)|-\rangle \tag{22}$$

we can immediately read off the measurement probabilities:

$$\Pr(+) = |\langle +|\psi\rangle|^2 = \frac{1}{2}|\alpha+\beta|^2 \tag{23}$$
$$\Pr(-) = |\langle -|\psi\rangle|^2 = \frac{1}{2}|\alpha-\beta|^2 \tag{24}$$

Multi-qubit systems

Hilbert spaces compose under the *tensor product*.

Example

Let

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}, \quad B = \begin{pmatrix} e & f \\ g & h \end{pmatrix}. \tag{25}$$

The tensor product of A and B, $A \otimes B$ is

$$A \otimes B = \begin{pmatrix} a\begin{pmatrix} e & f \\ g & h \end{pmatrix} & b\begin{pmatrix} e & f \\ g & h \end{pmatrix} \\ c\begin{pmatrix} e & f \\ g & h \end{pmatrix} & d\begin{pmatrix} e & f \\ g & h \end{pmatrix} \end{pmatrix} = \begin{pmatrix} ae & af & be & bf \\ ag & ah & bg & bh \\ ce & cf & de & df \\ cg & ch & dg & dh \end{pmatrix} \tag{26}$$

Multi-qubit systems

Qubit state vectors are also combined using the *tensor product*:

$$|01\rangle = |0\rangle \otimes |1\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \begin{pmatrix} 0 \\ 1 \end{pmatrix} \\ 0 \begin{pmatrix} 0 \\ 1 \end{pmatrix} \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} \tag{27}$$

An n-qubit state is therefore a vector of length 2^n.

The tensor product is linear and distributive, so if we have

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \quad |\varphi\rangle = \gamma|0\rangle + \delta|1\rangle, \tag{28}$$

then they tensor together to form

$$\begin{aligned} |\psi\rangle \otimes |\varphi\rangle &= (\alpha|0\rangle + \beta|1\rangle) \otimes (\gamma|0\rangle + \delta|1\rangle) \\ &= \alpha\gamma|00\rangle + \alpha\delta|01\rangle + \beta\gamma\ |10\rangle + \beta\delta|11\rangle \end{aligned}$$

The states $|00\rangle, |01\rangle, |10\rangle, |11\rangle$ are the computational basis vectors for 2 qubits; we can create arbitrary linear combinations of them as long as the normalization on the coefficients holds.

Single-qubit unitary operations also compose under tensor product.

For example, apply U_1 to qubit $|\psi\rangle$ and U_2 to qubit $|\varphi\rangle$:

$$(U_1 \otimes U_2)(|\psi\rangle \otimes |\varphi\rangle) = (U_1|\psi\rangle) \otimes (U_2|\varphi\rangle) \tag{29}$$

If an n-qubit ket is a vector with length 2^n, then a unitary acting on n qubits has dimension $2^n \times 2^n$.

Exercise: Consider the 2-qubit state

$$|\Psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle). \tag{30}$$

Find

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \quad |\varphi\rangle = \gamma|0\rangle + \delta|1\rangle \tag{31}$$

such that

$$|\Psi\rangle = |\psi\rangle \otimes |\varphi\rangle \tag{32}$$

Exercise: Consider the 2-qubit state

$$|\Psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle). \tag{30}$$

Find

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \quad |\varphi\rangle = \gamma|0\rangle + \delta|1\rangle \tag{31}$$

such that

$$|\Psi\rangle = |\psi\rangle \otimes |\varphi\rangle \tag{32}$$

Solution: This is impossible (sorry!)

Entanglement

The state

$$|\Psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle). \tag{33}$$

is **entangled**.

We cannot describe the two qubits individually, we can only described their combined state.

Paraphrasing from John Preskill: *it's like you're reading a book, but instead of reading the pages sequentially, you have to read it all at the same time in order to understand it.*

Entanglement

Furthermore, the measurement outcomes of

$$|\Psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle). \tag{34}$$

are *perfectly correlated*.

For example, if I measure the first qubit and get 0, I'll get 0 for the second qubit as well!

Entanglement is not limited to two qubits. In principle we can entangle as many as we like:

$$|\Psi\rangle = |00\cdots0\rangle + |11\cdots1\rangle \tag{35}$$

A measurement outcome of 0 on qubit 1 means we'll get 0 on *all other qubits* too.

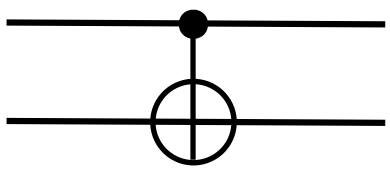
Entangling gates

How do we make an entangled state (in theory)? Previous 2-qubit operations we saw were expressed as tensor products of single-qubit ones.

There exist *entangling gates* that will turn a non-entangled, or separable, state into an entangled one. The most commonly used one is the controlled-NOT, or CNOT:

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

$$\begin{aligned} \text{CNOT}|00\rangle &= |00\rangle \\ \text{CNOT}|01\rangle &= |01\rangle \\ \text{CNOT}|10\rangle &= |11\rangle \\ \text{CNOT}|11\rangle &= |10\rangle \end{aligned}$$



The first qubit is the *control* qubit - it controls whether or not an X (NOT) gate is applied to the second qubit.

Entangling gates: CNOT

Exercise: What happens when we apply a CNOT to qubits in state $|+\rangle \otimes |0\rangle$?

$$\begin{aligned}
\text{CNOT}\left[\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |0\rangle\right] &= \text{CNOT}\left[\frac{1}{\sqrt{2}}(|00\rangle + |10\rangle)\right] \\
&= \frac{1}{\sqrt{2}}(\text{CNOT}|00\rangle + \text{CNOT}|10\rangle) \\
&= \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)
\end{aligned}$$

We've gone from a separable state to an entangled one!

Universal gate sets

Now that we've seen the CNOT gate, in principle *I don't need to introduce to you any additional single- or multi-qubit gates*.

That's a good thing - there are an infinite number of unitaries, and there's no way we can individually program each one into our quantum computing hardware.

Thankfully, the CNOT and a few single-qubit gates are all we need for universal quantum computation!

How many is "a few"?

Single-qubit universal gate set

If you have a quantum computer that can perform

$$\{H, \ T\} \tag{36}$$

then you can implement *any* other single-qubit unitary up to an arbitrary precision.

Universal gate sets

Single-qubit universal gate set

If you have a quantum computer that can perform

$$\{H,\ \ T\} \tag{36}$$

then you can implement *any* other single-qubit unitary up to an arbitrary precision.

Multi-qubit universal gate set

If you have a quantum computer that can perform

$$\{H,\ \ T,\ \ \text{CNOT}\} \tag{37}$$

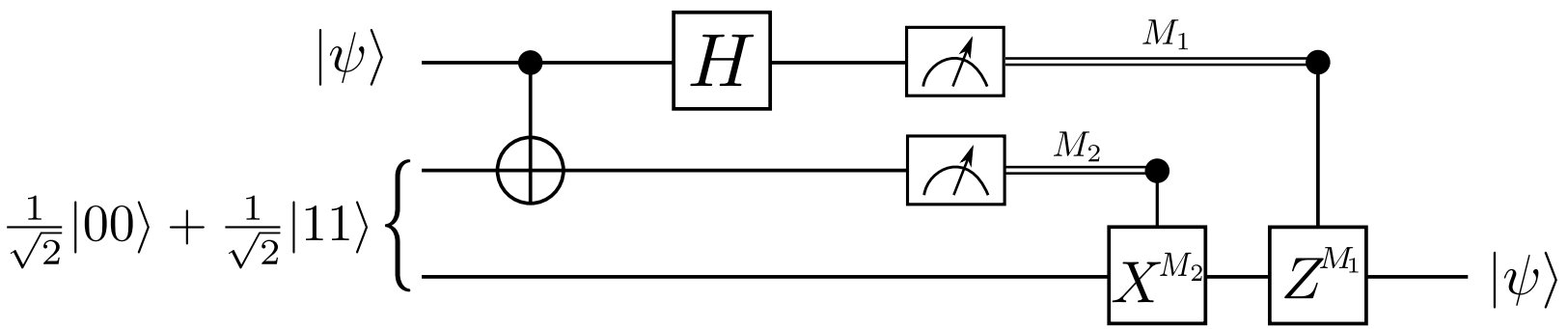
then you can implement *any* other multi-qubit unitary up to an arbitrary precision.

Teleporting quantum states

Quantum teleportation

NOT related to Star Trek, despite the photos in media articles.

Alice has a qubit state she wants to send to Bob. She can do so if they share an entangled pair of qubits.



Wait a minute...

Why does she have to do this crazy thing? Why can't she just make a copy of her state and send it to him?

This is forbidden by the *no-cloning theorem*.

The no-cloning theorem

It is impossible to create a copying circuit that works for arbitrary quantum states.

We can prove this!

Proof of the no-cloning theorem

Suppose we want to clone a state $|\psi\rangle$. We want a unitary operation that sends

$$U(|\psi\rangle \otimes |s\rangle) \rightarrow |\psi\rangle \otimes |\psi\rangle \tag{38}$$

where $|s\rangle$ is some arbitrary state.

Let's suppose we find one. If our cloning machine is going to be universal, then we must also be able to clone some other state, $|\varphi\rangle$.

$$U(|\varphi\rangle \otimes |s\rangle) \rightarrow |\varphi\rangle \otimes |\varphi\rangle \tag{39}$$

Proof of the no-cloning theorem

We purportedly have:

$$U(|\psi\rangle \otimes |s\rangle) = |\psi\rangle \otimes |\psi\rangle \quad (40)$$

$$U(|\varphi\rangle \otimes |s\rangle) = |\varphi\rangle \otimes |\varphi\rangle \quad (41)$$

Take the inner product of the LHS of both equations:

$$(\langle\psi| \otimes \langle s|)U^\dagger U(|\varphi\rangle \otimes |s\rangle) = \langle\psi|\varphi\rangle\langle s|s\rangle = \langle\psi|\varphi\rangle \quad (42)$$

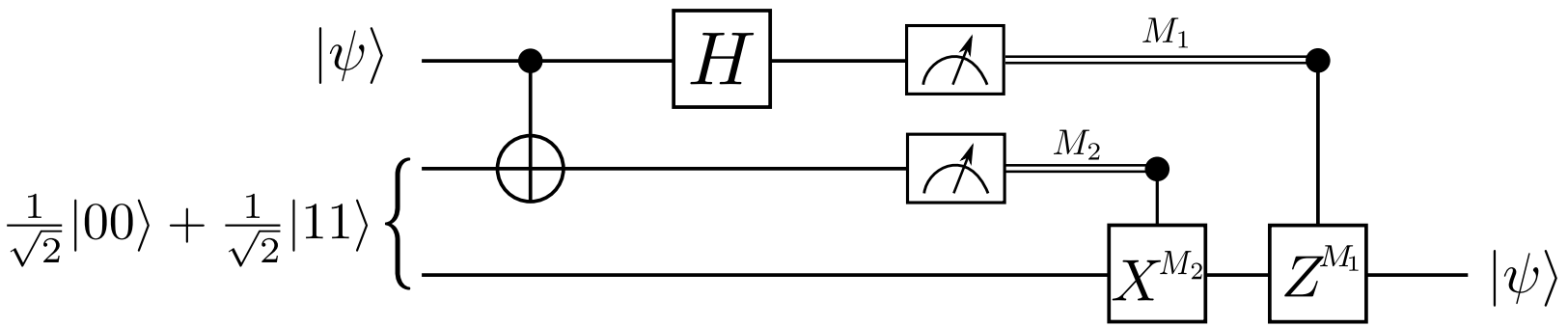
Now take the inner product of the RHS of both equations:

$$(\langle\psi| \otimes \langle\psi|)(|\varphi\rangle \otimes |\varphi\rangle) = (\langle\psi|\varphi\rangle)^2 \quad (43)$$

These two inner products must be equal; but the only numbers that square to themselves are 0 and 1! So either the two states are orthogonal, or are just the same state - they can't be arbitrary!

Back to quantum teleportation

So there is no general protocol for Alice to copy her qubit and send it to Bob; but teleportation allows her to transfer arbitrary *states* from her qubit to the qubit held by Bob.



How does this work?

Quantum teleportation

At the beginning, Alice has two qubits, one of which is part of a shared entangled state with Bob.

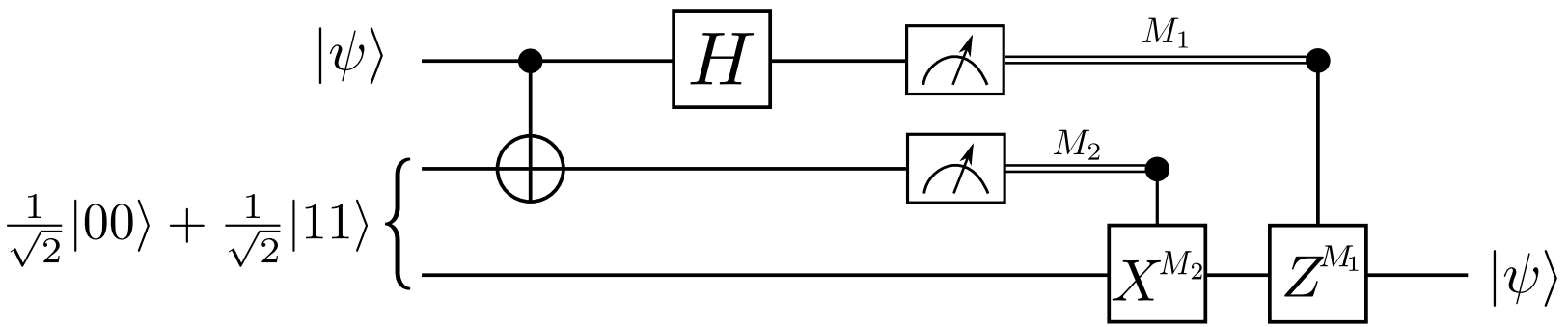
$$|\psi\rangle_A \otimes \frac{1}{\sqrt{2}} (|0\rangle_A |0\rangle_B + |1\rangle_A |1\rangle_B) \tag{44}$$

Alice then entangles her first qubit with the one in the already-entangled state. She then applies a Hadamard to her first qubit, and measures both of her qubits.

Based on the results, Bob will perform a Pauli 'correction', and he will be left with Alice's original state.

I *really* encourage you to work through this one on your own.

Start by setting $|\psi\rangle_A = \alpha|0\rangle_A + \beta|1\rangle_A$, expand out the linear combination, work through the action of the gates, and then try and factor out Bob's qubit before performing the measurement.



Quantum teleportation

If you work through the math, you'll find that before the measurements, the combined state of the system looks like this (removing the $\frac{1}{\sqrt{2}}$ for readability):

$$
\begin{aligned}
(|00\rangle_A + |11\rangle_A) &\otimes (\alpha|0\rangle_B + \beta|1\rangle_B) + \\
(|10\rangle_A + |01\rangle_A) &\otimes (\alpha|1\rangle_B + \beta|0\rangle_B) + \\
(|00\rangle_A - |11\rangle_A) &\otimes (\alpha|0\rangle_B - \beta|1\rangle_B) + \\
(|01\rangle_A - |10\rangle_A) &\otimes (\alpha|1\rangle_B - \beta|0\rangle_B)
\end{aligned}
$$

This is a superposition of 4 distinct terms.

You can see that Bob's state is always some variation on the original state of Alice:

$$
\begin{aligned}
(|00\rangle_A + |11\rangle_A) &\otimes (\alpha|0\rangle_B + \beta|1\rangle_B) + \\
(|10\rangle_A + |01\rangle_A) &\otimes (\alpha|1\rangle_B + \beta|0\rangle_B) + \\
(|00\rangle_A - |11\rangle_A) &\otimes (\alpha|0\rangle_B - \beta|1\rangle_B) + \\
(|01\rangle_A - |10\rangle_A) &\otimes (\alpha|1\rangle_B - \beta|0\rangle_B)
\end{aligned}
$$

The possible states of Alice's qubits are an orthonormal 2-qubit basis of entangled states called the Bell basis.

$$
\begin{aligned}
(|00\rangle_A + |11\rangle_A) &\otimes (\alpha|0\rangle_B + \beta|1\rangle_B) + \\
(|10\rangle_A + |01\rangle_A) &\otimes (\alpha|1\rangle_B + \beta|0\rangle_B) + \\
(|00\rangle_A - |11\rangle_A) &\otimes (\alpha|0\rangle_B - \beta|1\rangle_B) + \\
(|01\rangle_A - |10\rangle_A) &\otimes (\alpha|1\rangle_B - \beta|0\rangle_B)
\end{aligned}
$$

So Alice can measure in the Bell basis, and send her results to Bob.

$$\begin{aligned}
(|00\rangle_A + |11\rangle_A) &\otimes (\alpha|0\rangle_B + \beta|1\rangle_B) + \\
(|10\rangle_A + |01\rangle_A) &\otimes (\alpha|1\rangle_B + \beta|0\rangle_B) + \\
(|00\rangle_A - |11\rangle_A) &\otimes (\alpha|0\rangle_B - \beta|1\rangle_B) + \\
(|01\rangle_A - |10\rangle_A) &\otimes (\alpha|1\rangle_B - \beta|0\rangle_B)
\end{aligned}$$

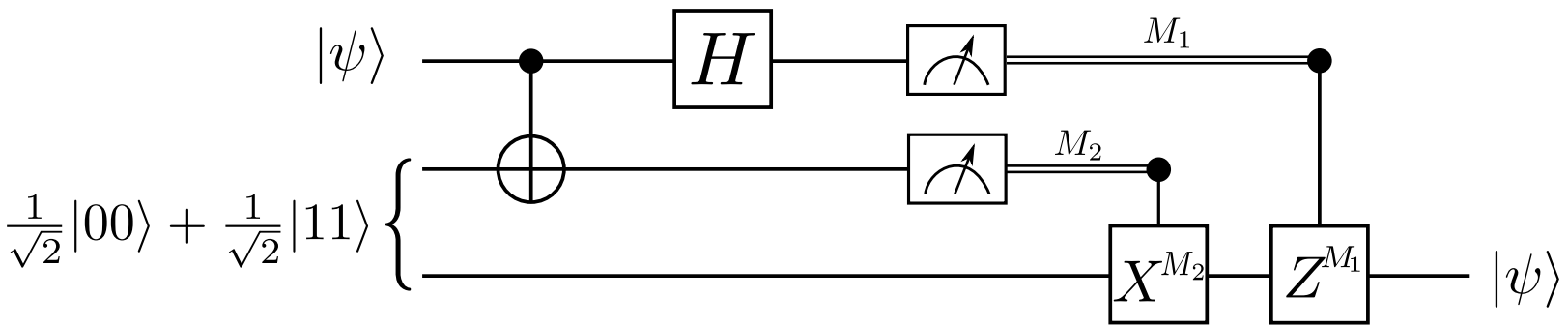
Once Bob knows the results, he knows exactly what term of the superposition they had, and can adjust his state accordingly.

Hands-on with Quirk

Go back to

`https://algassert.com/quirk`

and implement the circuit for quantum teleportation:



You will need to:

- Prepare a state to teleport (e.g. define some rotation)
- Prepare the shared entangled state

Quantum advantage

Simulating a quantum computer

You might be thinking: *"if everything is just linear algebra, why bother building a quantum computer at all?"*

For an n-qubit computation, we'd need to:

- Store 2^n complex numbers
- Store $2^n \times 2^n$ unitary matrices
- Multiply the two together repeatedly

This has **insane** time and memory requirements.

The best *full* quantum simulators running on a laptop can manage about 30 qubits with 16 GB RAM. The top supercomputers have managed ~ 50 qubits on circuits of depth ~ 40.

Quantum speedup

Important question: What kinds of problems can quantum computers do better than classical computers?

There are 3 important classes of quantum algorithms:

- Algorithms based on the *quantum Fourier transform*
- Algorithms based on *amplitude amplification*
- Hamiltonian simulation

Polynomial speedup

Quantum algorithms based on amplitude amplification typically obtain a *polynomial* speedup over the best-known classical algorithms. i.e. for n qubits, if a classical algorithm takes $\sim 2^n$ time steps to run, the improved quantum version will take $\sim \sqrt{2^n} = 2^{\frac{n}{2}}$ time steps.

Exponential speedup

Many quantum algorithms that use the quantum Fourier transform as an underlying subroutine obtain a *superpolynomial*, or *exponential* speedup. If the classical algorithm takes $\sim 2^n$ time steps, the quantum algorithm takes $\sim n^k$, for some integer k (ideally n^2 or n^3).

What can we do with a quantum computer?

There are many potential uses for quantum hardware:

- Use the quantum Fourier transform to solve problems in discrete mathematics (discrete logarithm, order-finding, factoring)
- Random number generation
- Quantum key distribution and cryptography for secure quantum communication
- Using Grover's algorithm to search large spaces
- Speeding up linear algebra operations; as 'accelerators' for machine learning algorithms
- *Tons of things we haven't even thought of yet!* Designing quantum algorithms is really hard.

Will quantum computers be better at everything?

Not every quantum algorithm provides the desired 'superpolynomial' speedup over classical computers.

There will still be problems that even quantum computers can't solve efficiently.[2]

So, more important question: what kinds of problems are complex enough to make it worth pulling out a quantum computer in the first place?

[2] For those of you who are interested in computational complexity, check out the complete problems in the complexity class QMA `https://arxiv.org/abs/1212.6312`.

Quantum advantage

At what point will quantum computers be able to solve a useful problem that is intractable for a classical computer? How large of a problem, or how many qubits, do we need before we see a *quantum advantage*?[3]

Need things that are exponentially hard on a classical computer but that quantum computers can solve efficiently.

This is a moving, problem-dependent target! A few years ago, people were estimating we would need around 50 qubits for this, but classical simulators have caught up.

[3] See: https://medium.com/@wjzeng/clarifying-quantum-supremacy-better-terms-for-milestones-in-quantum-computation-d15ccb53954f

Candidate problem: sampling random circuits

Given a random circuit, sample from the probability distribution given by its output.

Candidate problem: sampling random circuits

Given a random circuit, sample from the probability distribution given by its output.

Classical:
Do the matrix multiplication to work through the entire circuit, get the final amplitudes, then sample. Exponentially hard!

Candidate problem: sampling random circuits

Given a random circuit, sample from the probability distribution given by its output.

Classical:
Do the matrix multiplication to work through the entire circuit, get the final amplitudes, then sample. Exponentially hard!

Quantum:
Run the circuit many times and keep track of the distribution of measured outputs.

How large a circuit do we need before we can no longer achieve this with classical computers?

Candidate problem: sampling random circuits

Quantum Physics

Quantum Supremacy Circuit Simulation on Sunway TaihuLight

Riling Li, Bujiao Wu, Mingsheng Ying, Xiaoming Sun, Guangwen Yang

(Submitted on 13 Apr 2018 (v1), last revised 13 Aug 2018 (this version, v3))

With the rapid progress made by industry and academia, quantum computers with dozens of qubits or even larger size are being realized. However, the fidelity of existing quantum computers often sharply decreases as the circuit depth increases. Thus, an ideal quantum circuit simulator on classical computers, especially on high-performance computers, is needed for benchmarking and validation. We design a large-scale simulator of universal random quantum circuits, often called 'quantum supremacy circuits', and implement it on Sunway TaihuLight. The simulator can be used to accomplish the following two tasks: 1) Computing a complete output state-vector; 2) Calculating one or a few amplitudes. We target the simulation of 49-qubit circuits. For task 1), we successfully simulate such a circuit of depth 39, and for task 2) we reach the 55-depth level. To the best of our knowledge, both of the simulation results reach the largest depth for 49-qubit quantum supremacy circuits.

(When this paper was written, the TaihuLight was the top supercomputer in the world.)

Candidate problem: sampling random circuits

Quantum Physics

Massively parallel quantum computer simulator, eleven years later

Hans De Raedt, Fengping Jin, Dennis Willsch, Madita Nocon, Naoki Yoshioka, Nobuyasu Ito, Shengjun Yuan, Kristel Michielsen

(Submitted on 12 May 2018 (v1), last revised 11 Dec 2018 (this version, v2))

A revised version of the massively parallel simulator of a universal quantum computer, described in this journal eleven years ago, is used to benchmark various gate-based quantum algorithms on some of the most powerful supercomputers that exist today. Adaptive encoding of the wave function reduces the memory requirement by a factor of eight, making it possible to simulate universal quantum computers with up to 48 qubits on the Sunway TaihuLight and on the K computer. The simulator exhibits close-to-ideal weak-scaling behavior on the Sunway TaihuLight,on the K computer, on an IBM Blue Gene/Q, and on Intel Xeon based clusters, implying that the combination of parallelization and hardware can track the exponential scaling due to the increasing number of qubits. Results of executing simple quantum circuits and Shor's factorization algorithm on quantum computers containing up to 48 qubits are presented.

Candidate problem: sampling random circuits

Quantum Physics

Quantum Supremacy Is Both Closer and Farther than It Appears

Igor L. Markov, Aneeqa Fatima, Sergei V. Isakov, Sergio Boixo

(Submitted on 27 Jul 2018 (v1), last revised 26 Sep 2018 (this version, v3))

As quantum computers improve in the number of qubits and fidelity, the question of when they surpass state-of-the-art classical computation for a well-defined computational task is attracting much attention. The leading candidate task for this milestone entails sampling from the output distribution defined by a random quantum circuit. We develop a massively-parallel simulation tool Rollright that does not require inter-process communication (IPC) or proprietary hardware. We also develop two ways to trade circuit fidelity for computational speedups, so as to match the fidelity of a given quantum computer --- a task previously thought impossible. We report massive speedups for the sampling task over prior software from Microsoft, IBM, Alibaba and Google, as well as supercomputer and GPU-based simulations. By using publicly available Google Cloud Computing, we price such simulations and enable comparisons by total cost across hardware platforms. We simulate approximate sampling from the output of a circuit with 7x8 qubits and depth 1+40+1 by producing one million bitstring probabilities with fidelity 0.5%, at an estimated cost of $35184. The simulation costs scale linearly with fidelity, and using this scaling we estimate that extending circuit depth to 1+48+1 increases costs to one million dollars. Scaling the simulation to 10M bitstring probabilities needed for sampling 1M bitstrings helps comparing simulation to quantum computers. We describe refinements in benchmarks that slow down leading simulators, halving the circuit depth that can be simulated within the same time.

Quantum advantage

There has yet to be a concrete demonstration of quantum advantage. How close are we?

Google and NASA signed an agreement last July saying they would prove it in 12 months...

Quantum simulation on Summit

Instead they have just pushed forward the frontier.

Quantum Physics

Establishing the Quantum Supremacy Frontier with a 281 Pflop/s Simulation

Benjamin Villalonga, Dmitry Lyakh, Sergio Boixo, Hartmut Neven, Travis S. Humble, Rupak Biswas, Eleanor G. Rieffel, Alan Ho, Salvatore Mandrà

(Submitted on 1 May 2019)

Noisy Intermediate-Scale Quantum (NISQ) computers aim to perform computational tasks beyond the capabilities of the most powerful classical computers, thereby achieving "Quantum Supremacy", a major milestone in quantum computing. NISQ Supremacy requires comparison with a state-of-the-art classical simulator. We report HPC simulations of hard random quantum circuits (RQC), sustaining an average performance of 281 Pflop/s (true single precision) on Summit, currently the fastest supercomputer in the world. In addition, we propose a standard benchmark for NISQ computers based on qFlex, a tensor-network-based classical high-performance simulator of RQC, which are considered the leading proposal for Quantum Supremacy.

Summit is currently the top supercomputer....

Quantum simulation on Summit

They used the whole thing!!!

			PFlop/s		Efficiency (%)	
Circuit Size	**Nodes Used**	**Runtime (h)**	**Peak**	**Sust.**	**Peak**	**Sust.**
$7 \times 7 \times (1 + 40 + 1)$	2300	4.84	191	142	92.0	68.5
$7 \times 7 \times (1 + 40 + 1)$	4600	2.44	381	281	92.1	68.0
$11 \times 11 \times (1 + 24 + 1)$	4550	0.278	368	261	89.8	63.7

TABLE I: Performance of the simulation of our three runs on Summit. For the

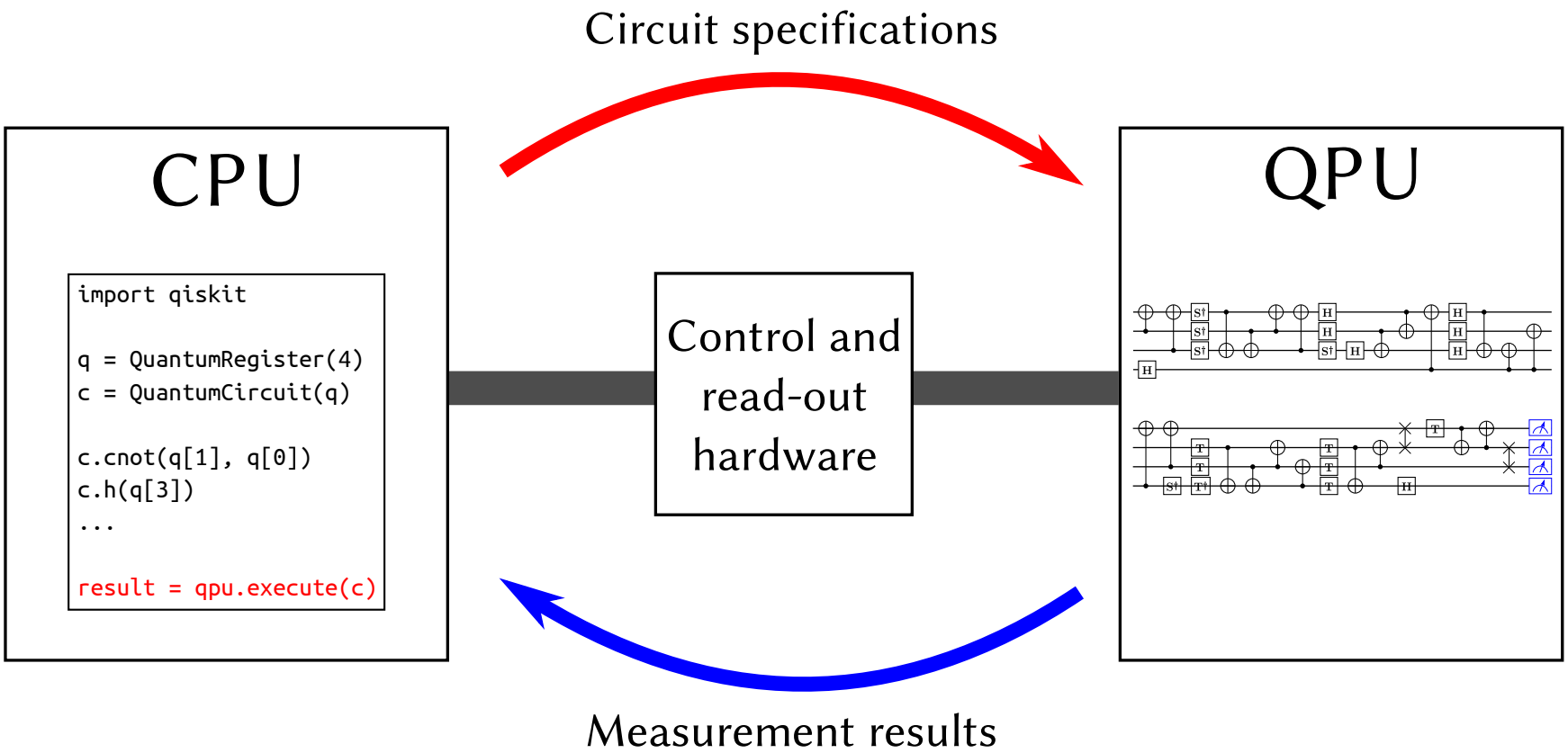
Simulated circuits for qubits in a 7×7 grid with depth 42, and 11×11 grid with depth 26.

Image credit: http://arxiv.org/abs/1905.00444

Quantum computing in practice: Hardware and the NISQ era

How do I use a quantum computer?

Now, and likely in the future, quantum computers are being used like special-purpose *accelerators* (think how GPUs are used today).



How do we build a *quantum processing unit*, or QPU?

The DiVincenzo criteria

1. A *scalable* physical system with well-characterized qubits.
2. The ability to *initialize* the state of the qubits to a simple fiducial state.
3. Long relevant decoherence times, much longer than the gate operation times.
4. A *universal* set of quantum gates.
5. A qubit-specific *measurement* capability.

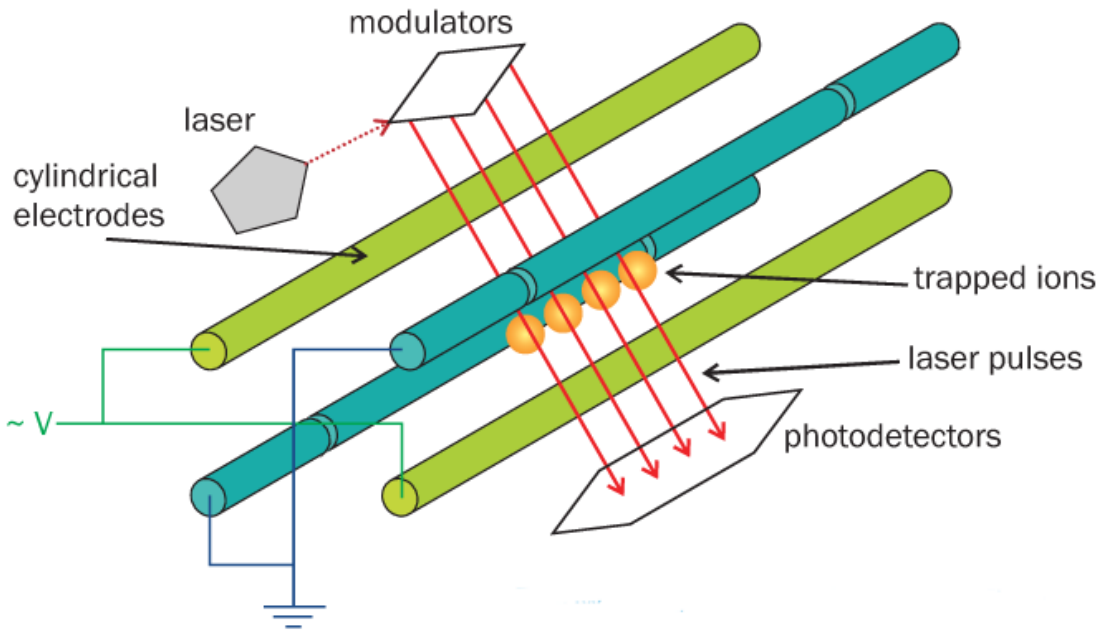
Candidate systems for qubits

- Superconducting qubits
- Trapped ions
- Spin qubits (neutral atoms, diamond NV centres, silicon spins qubits)
- Photons
- Topological qubits
- ... many more.

Superconducting qubits and trapped ions are currently the most well-developed.

Candidate systems for qubits: Trapped ions

Qubits are ground state hyperfine levels of ions suspended between electrodes.



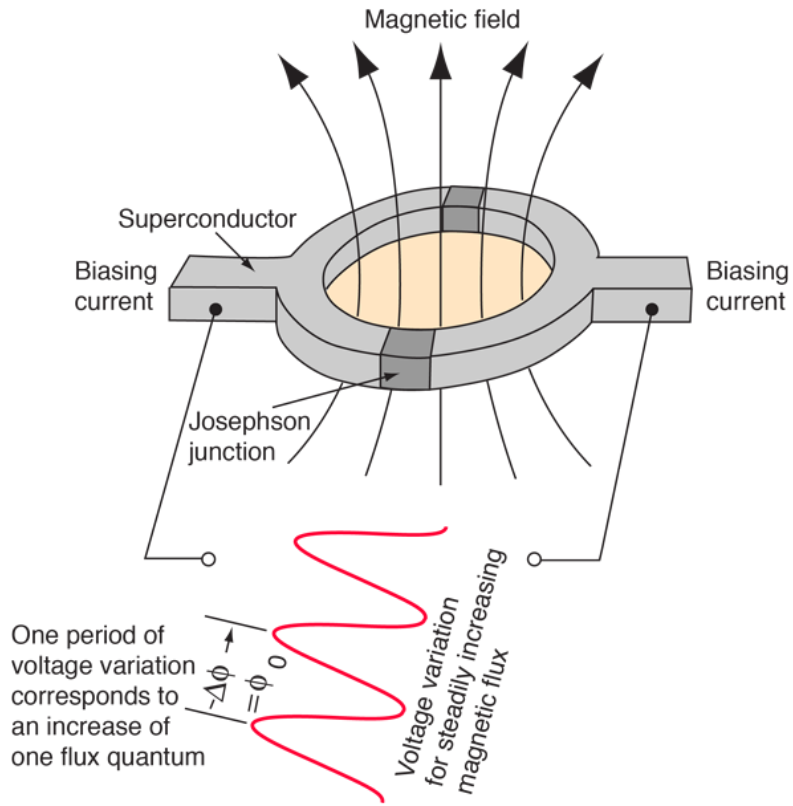
Qubits are individually addressable by applying EM fields of varying frequency and duration. Entangling gates are applied by coupling with a common vibrational mode of the whole chain.

Image credit: Wikimedia commons

Candidate systems for qubits: superconducting qubits

Multiple ways to make qubits:

- Charge qubits: uses the number of Cooper pairs sitting between a capacitor and a Josephson junction in an electrical circuit
- Flux qubits: uses the direction of magnetic flux induced by current in a superconducting loop

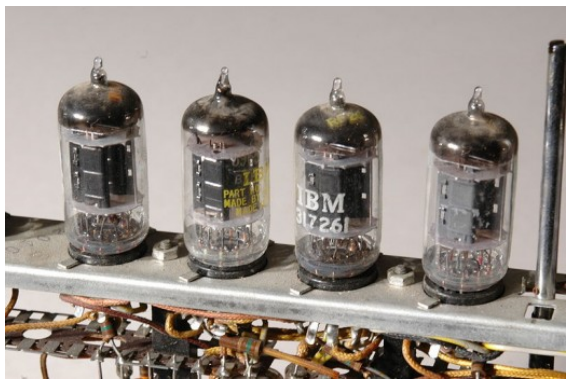


Qubits are individually addressable by applying microwave pulses; qubits are coupled via electric circuit connections.

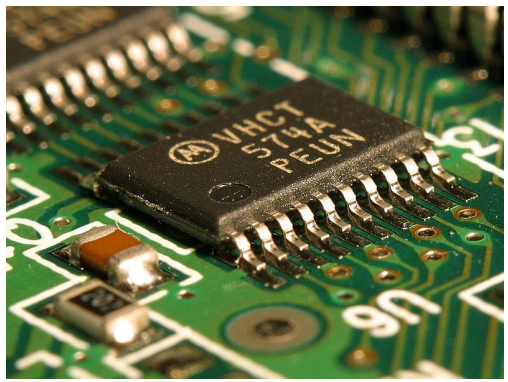
Image credit: `http://hyperphysics.phy-astr.gsu.edu/hbase/Solids/Squid.htm`

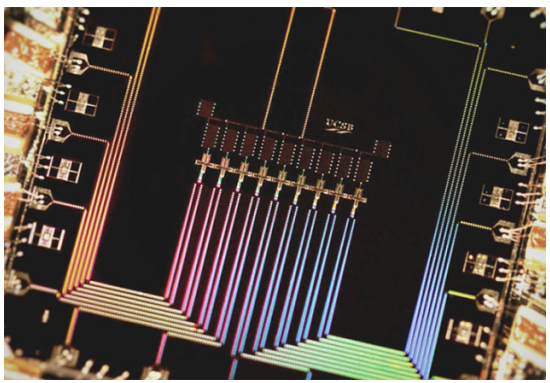
Candidate systems for qubits

However, it's important to remember that the "real" qubit of the future might not have even been invented yet!



vs.





vs.

???

Image credits:
https://hubpages.com/business/What-Is-a-Transistor-and-Why-is-it-Important
https://en.wikipedia.org/wiki/Solid-state_electronics
https://physicsworld.com/a/google-gains-new-ground-on-universal-quantum-computer/

NISQ-era quantum computing

We are in the era of 'Noisy, intermediate-scale quantum', or **NISQ** devices: medium-sized machines, but they have a number of limitations.

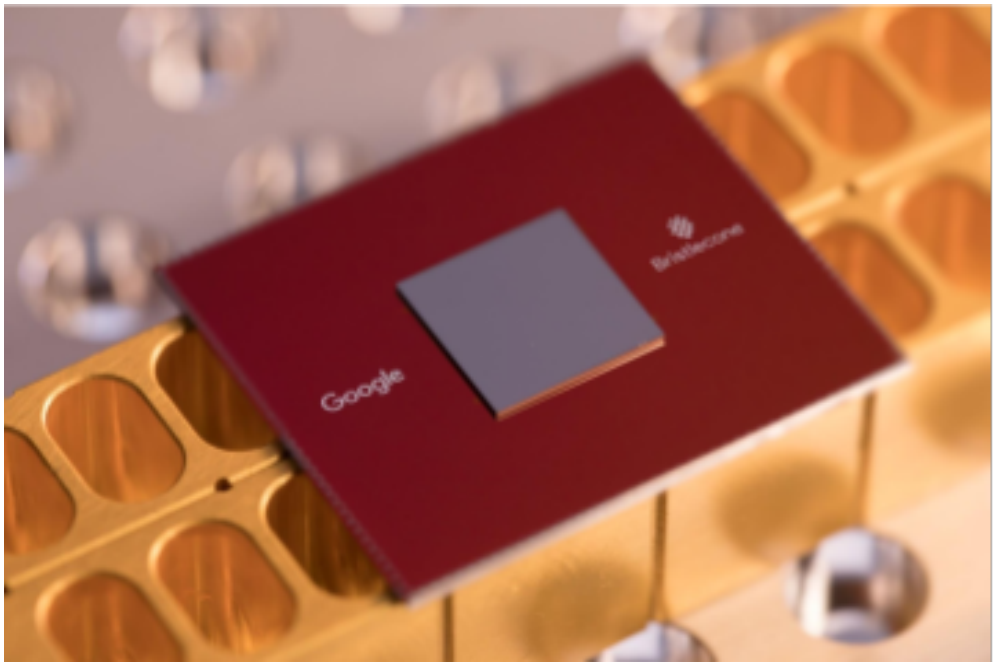
- Qubit count
- Qubit connectivity
- Error rates, coherence times

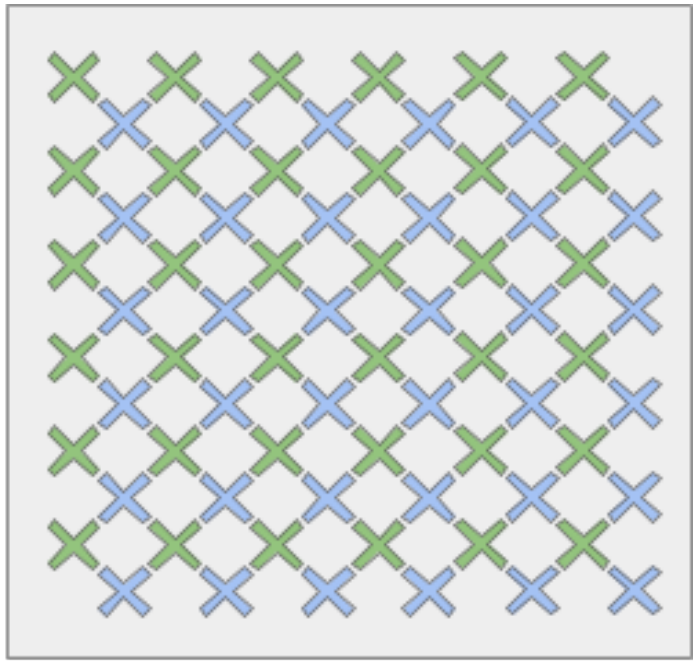
Useful site:
`https://quantumcomputingreport.com/scorecards/qubit-count/`

Important note: things are changing fast; no guarantees any of these numbers will be the same a month from now.

Google

Currently Google's *Bristlecone* is the largest device. It has 72 superconducting qubits arranged in a grid pattern.

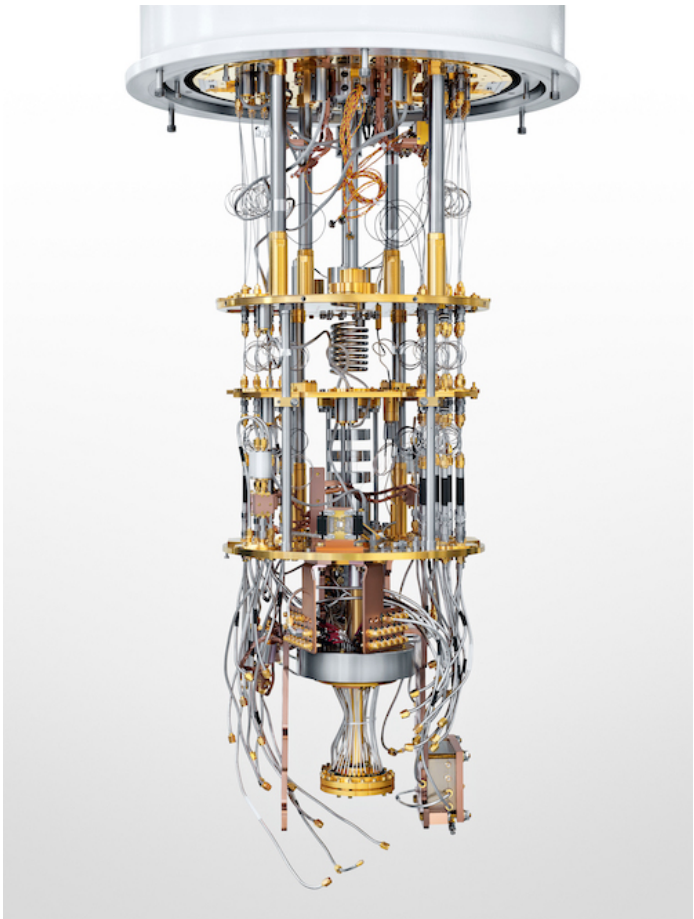




https://ai.googleblog.com/2018/03/a-preview-of-bristlecone-googles-new.html

Rigetti

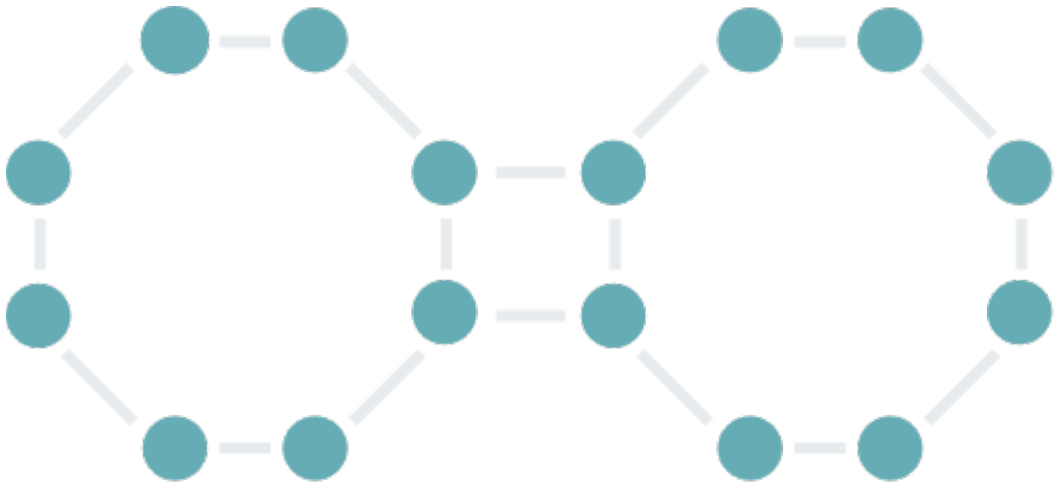
A California-based startup. Full-stack (hardware + software), superconducting qubit chip accessible via the cloud.



"Image courtesy Rigetti Computing. Photo by Justin Fantl."

Hardware graph - Rigetti

16-qubit chip in use



128-qubit chip in development; 8 x 16 of the unit cell above.

https://medium.com/rigetti/the-rigetti-128-qubit-chip-and-what-it-means-for-quantum-df757d1b71ea

IonQ

Startup out of U. Maryland. 79 qubits individually addressible; can do pairwise 2-qubit gates on up to 11 qubits. Currently accessible by request only.

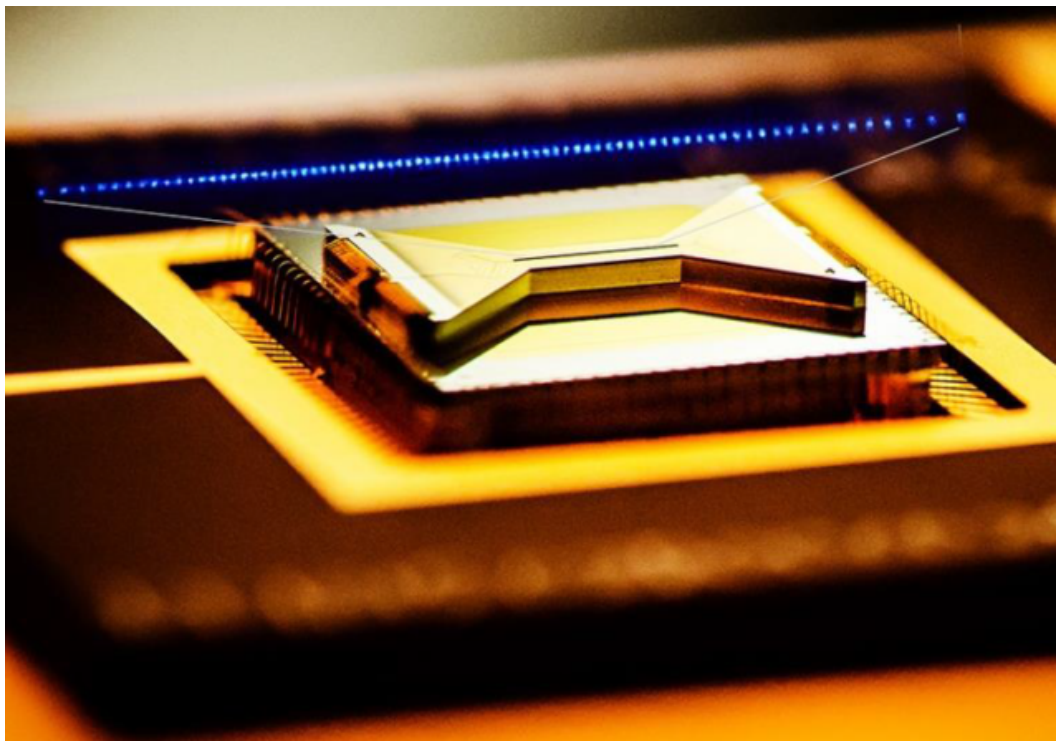


Image credit: `https://physicsworld.com/a/ion-based-commercial-quantum-computer-is-a-first/`

IBM

Full-stack, superconducting qubits, available in the cloud.

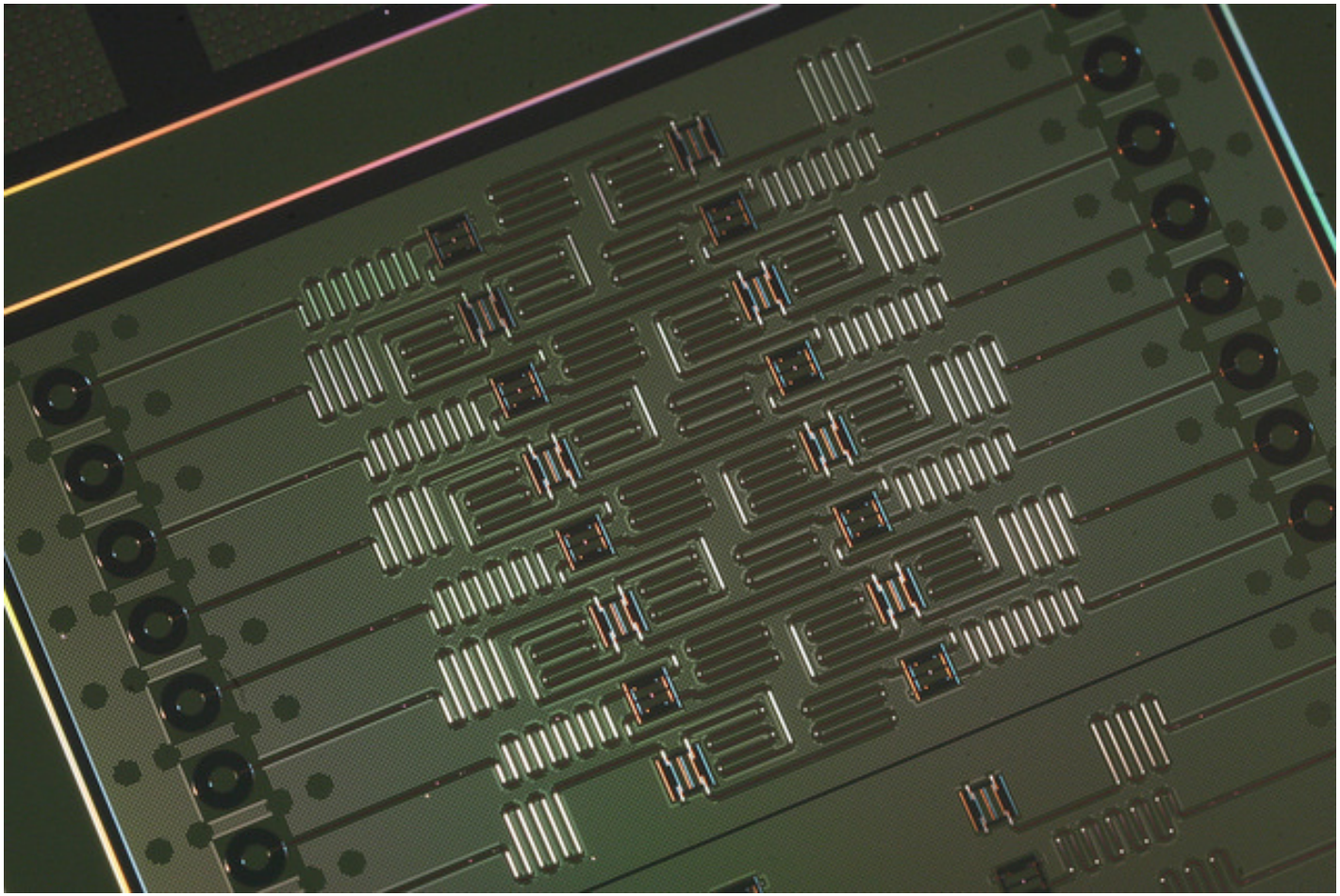
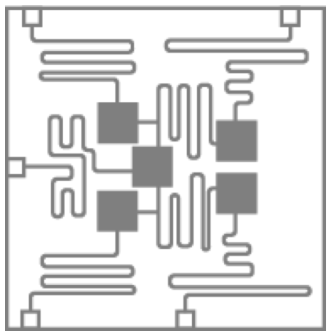
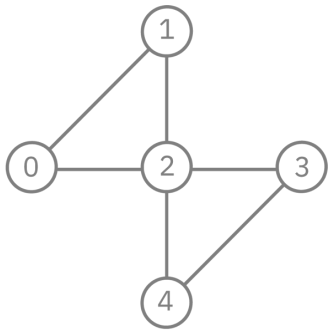


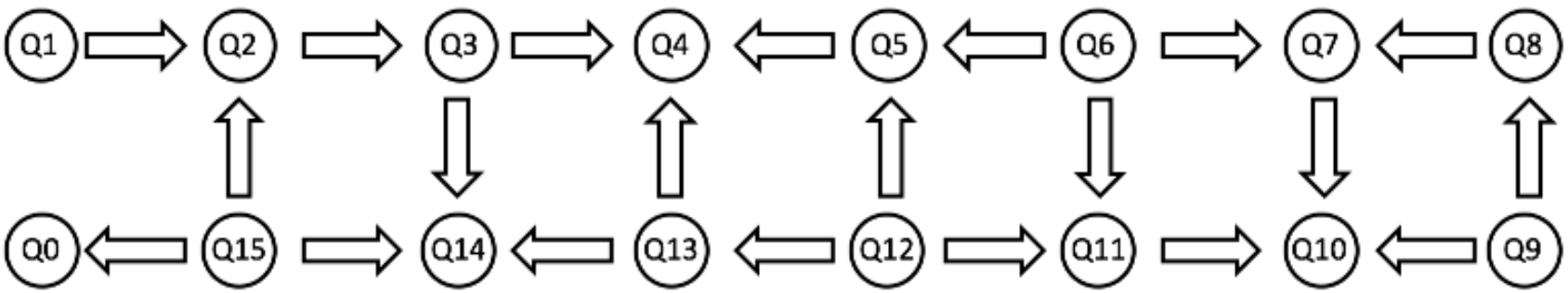
Image credit: https://www.ibm.com/blogs/research/2018/09/next-gen-qubits/

Hardware graph - IBM

Small machines



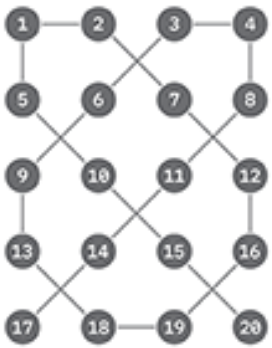


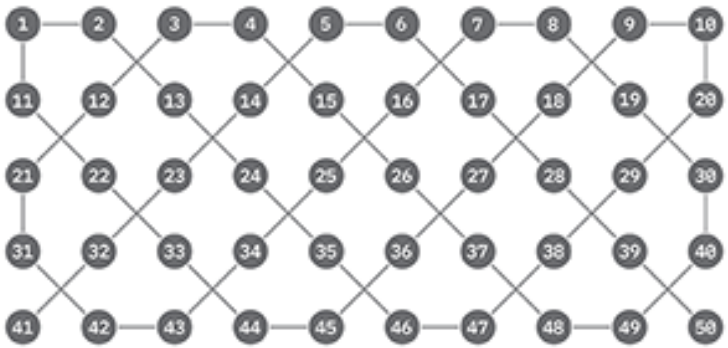


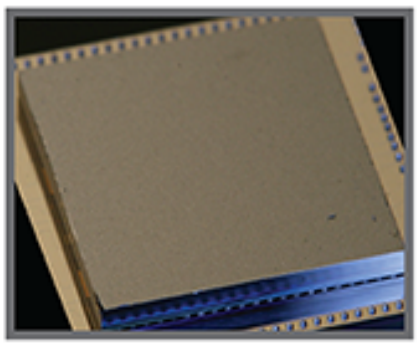
https://www.research.ibm.com/ibm-q/technology/devices/

Hardware graph - IBM

Larger machines







https://www.ibm.com/blogs/research/2017/11/the-future-is-quantum/

Qubit counts

Other players (non-exhaustive):

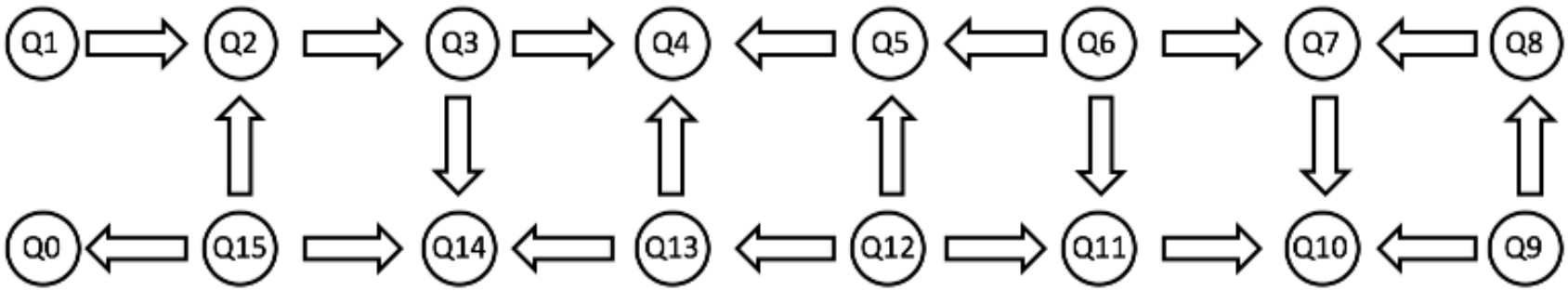
- Superconducting: Intel (49), U. of Science and Technology China (10), Alibaba (11)
- Ion traps: IonQ (11)[4], Inst. for Quantum Optics and Quantum Information (20)
- Spin qubits[5]: Intel (26)
- Neutral atoms: U. Wisconsin (49)
- Photonic computing: Xanadu, PsiQuantum
- Topological qubits: Microsoft

[4] 160 atoms, single-qubit gates on 79, two-qubit gates on all pairs of 11

[5] Being worked on at UBC and SFU!

Qubit connectivity

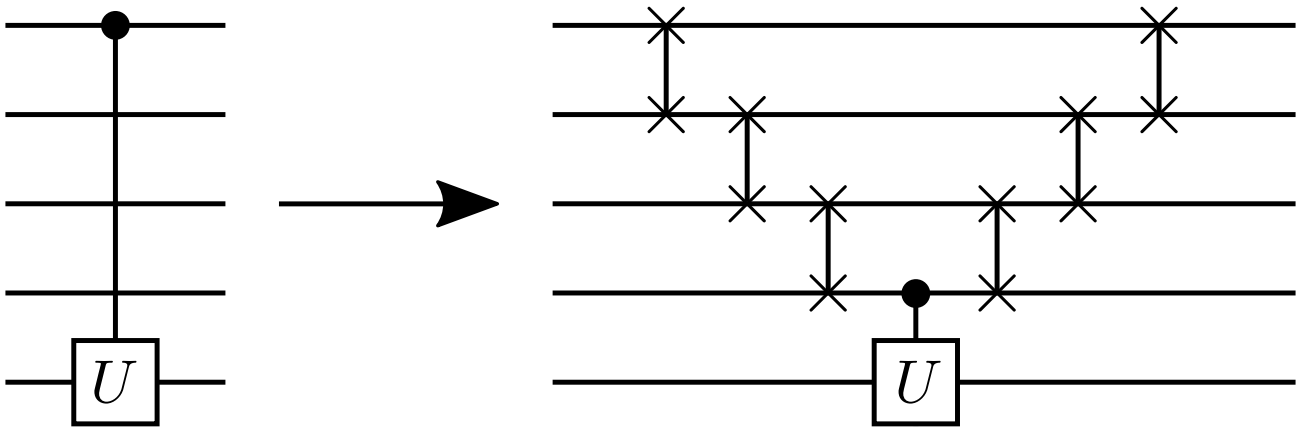
Look closely at the structure of this graph:



The arrows indicate where, and in what direction we can perform CNOT gates. What happens when we need to apply a CNOT on qubits that are far apart?

Qubit connectivity

Naive solution: when you need to operate on two non-adjacent qubits, perform SWAP gates until they are adjacent, perform the operation, then undo all the SWAPs.



Better solution: Heuristic techniques for compiling down to circuits over the universal gate set that fit the topology. For example, 're-allocation' of qubit indexing to minimize the number of SWAPs.

Noise

This is the 20-qubit IBM Q System One. It's in a dilution refrigerator cooled to $\sim 10 - 20$mK, suspended in a 9-foot cube.

Image credit:
`https://www.hpcwire.com/2019/01/10/ibm-quantum-update-q-system-one-launch-new-collaborators-and-qc-center-plans/`

Why so cold? Energy levels are very close together; need thermal excitations to be small enough that they don't cause transition between the qubit states.

Quantifying noise in quantum computing

Why so cold? Energy levels are very close together; need thermal excitations to be small enough that they don't cause transition between the qubit states.

Any amount of interaction with the outside world can cause *decoherence* of our quantum system.

Quantifying noise in quantum computing

Why so cold? Energy levels are very close together; need thermal excitations to be small enough that they don't cause transition between the qubit states.

Any amount of interaction with the outside world can cause *decoherence* of our quantum system.

We judge the quality of our qubits using a variety of metrics, for example:

- Gate fidelity (1- and 2-qubit gates)
- Spin relaxation time (time it takes for $|1\rangle$ to 'relax' to $|0\rangle$)
- Decoherence time

NISQ-device qubit quality

All values are averaged over the qubits:

Org.	Machine	1-qubit fid.	2-qubit fid.
IBM	Q System One	99.96	98.31
Rigetti	16Q Aspen-4	95.5	90.35
IonQ	11 Qubit	99.64	97.5

Other news:

- 2018: Silicon qubits with gate fidelity 99.96 (`https://arxiv.org/abs/1807.09500`)
- 2018: IonQ reports single-qubit fidelities of $> 99\%$, two-qubit of $> 98\%$ (`https://ionq.co/news/december-11-2018#appendix`)
- APS March Meeting 2019: Google and Rigetti claim $> 99\%$ fidelity two-qubit gates

For relatively up-to-date numbers:
`https://quantumcomputingreport.com/scorecards/qubit-quality/`

NISQ-device coherence times

All times are averages:

Org.	**Machine**	$\mathbf{T_1}$ (μ s)	$\mathbf{T_2}$ (μ s)
IBM	Q System One	73.9	69.1
Rigetti	16Q Aspen-4	25.24	19.89
IonQ	11 Qubit	$> 10^{10}$	$\sim 3 \cdot 10^6 \mu$s

T_1 is the relaxation time, T_2 is the decoherence time.

Other news:

- 2018: Silicon qubits have seen $T_2 \approx 9.4ms$
- 2018: IonQ reports $T_1 \approx 10^{10}\mu$s, $T_2 \approx 10^6\mu$s
- 2019: IBM reports lifetimes up to 500μs lifetime for individual superconducting qubits (`https://www.ibm.com/blogs/research/2019/03/power-quantum-device/`)
- 2019: TU Delft reports lifetimes up to 63s in diamond NV-center qubit `http://arxiv.org/abs/1905.02094`

Prospects for NISQ devices

We're getting there.

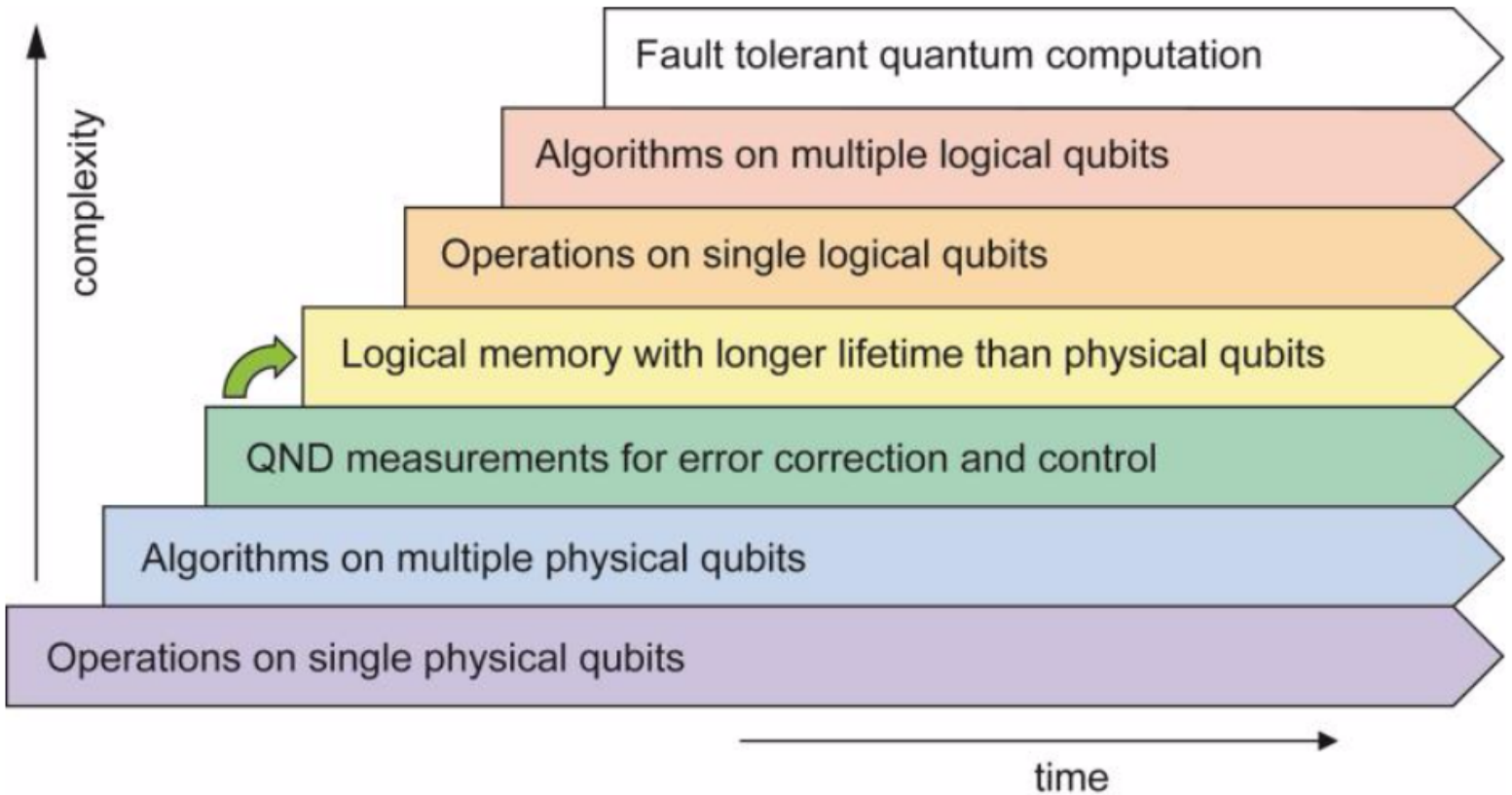


Image credit: 'Superconducting circuits for quantum information: An Outlook', Devoret and Schoelkopf, 2013

Prospects for NISQ devices

Today's qubits are too noisy to run algorithms 'at scale'.

However, there is a well-developed theory of *quantum error correction* and *fault-tolerance*. Multiple physical qubits can 'work together' in an error-correcting code as a single 'logical qubit'.

The problem is that there is typically an overhead of *thousands* of physical qubits required to make a single logical qubit. We don't have that many qubits!

This is beyond the scope of these lectures. For a cute, intuitive description: `https://www.youtube.com/watch?v=0U9_mrxLl3g`

Applications of NISQ devices

... can we still do interesting things with NISQ devices?

Yes! There is a steady stream of proof-of-concept applications and small/toy problem instances.

A few things that have been done so far:

- Calculations of molecular energies; largest to date has been water on the IonQ platform
  http://arxiv.org/abs/1902.10171
- Supervised classification (ML) on a toy dataset
  https://www.nature.com/articles/s41586-019-0980-2
- Simulation of lattice gauge theories
  http://stacks.iop.org/1367-2630/19/i=10/a=103020

Summary

Today we saw:

- Multi-qubit operations and entanglement
- How quantum state teleporation works
- What quantum advantage is, and a candidate problem
- What 'real' quantum computers currently look like, and what their limitations are

Questions? Comments? Interested in research? Feel free to come bug me any time (down the hall in MOB 284).