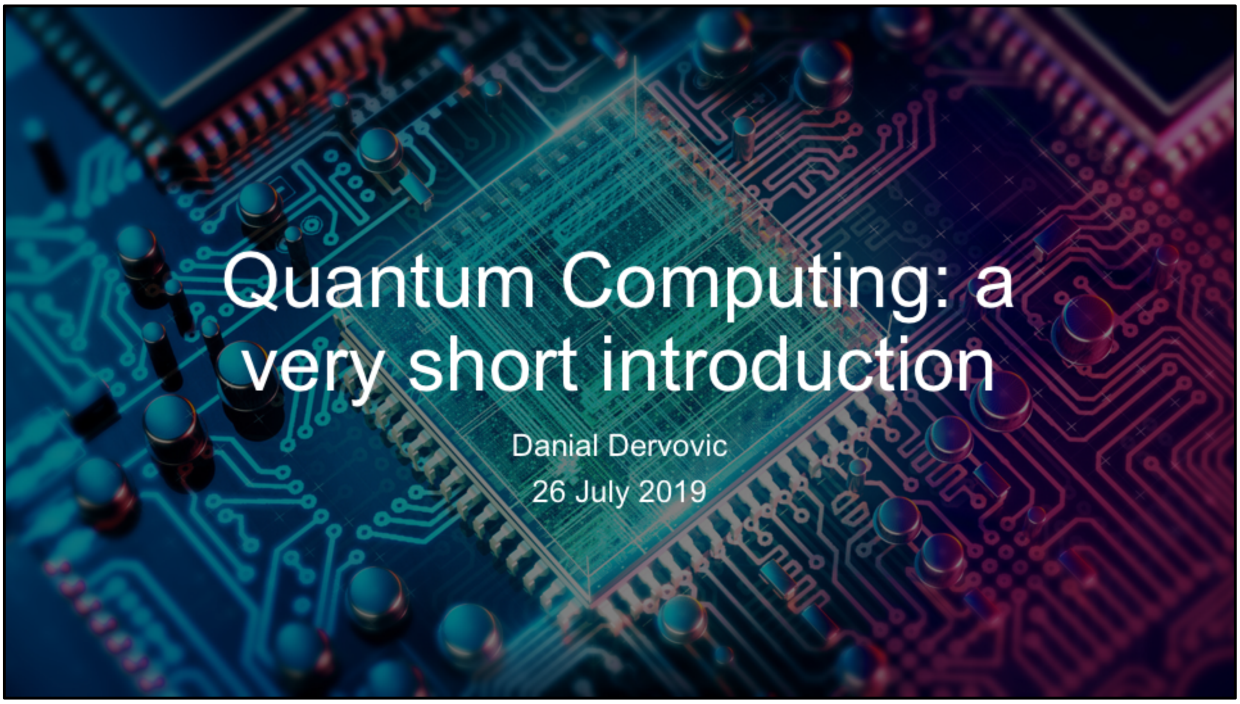
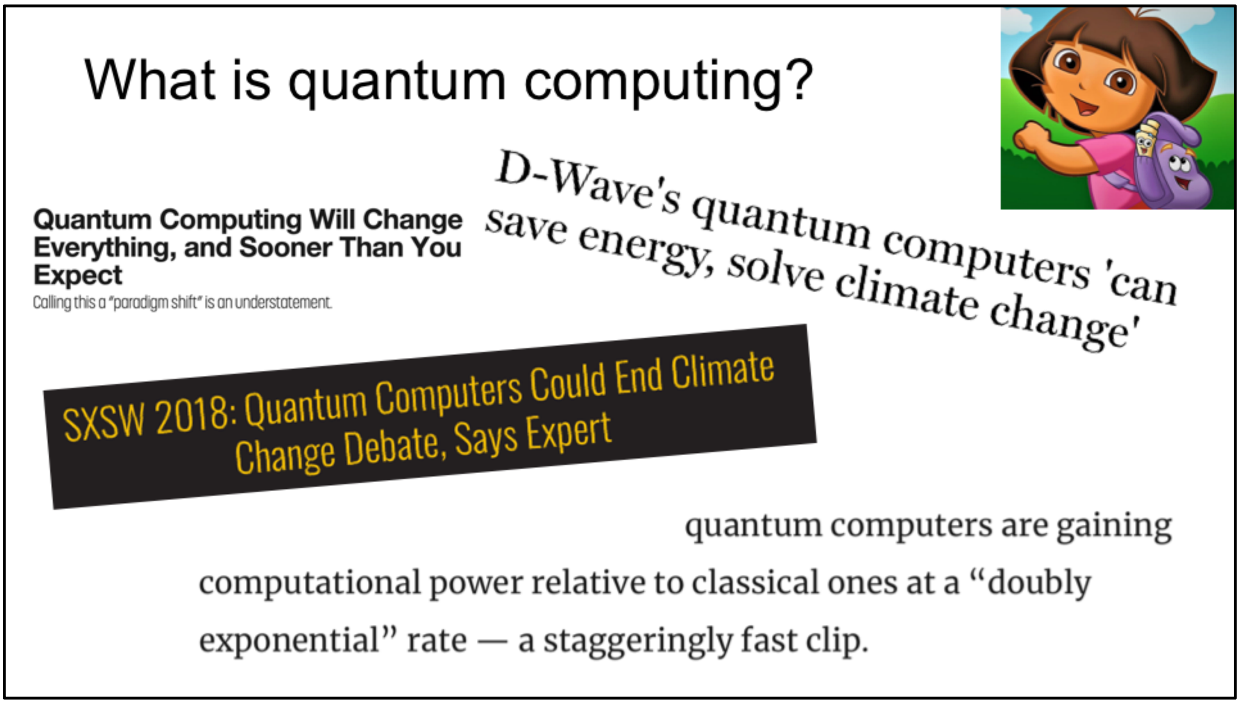


Quantum Computing: a very short introduction

Danial Dervovic

26 July 2019



What is quantum computing?

Quantum Computing Will Change Everything, and Sooner Than You Expect

Calling this a "paradigm shift" is an understatement.

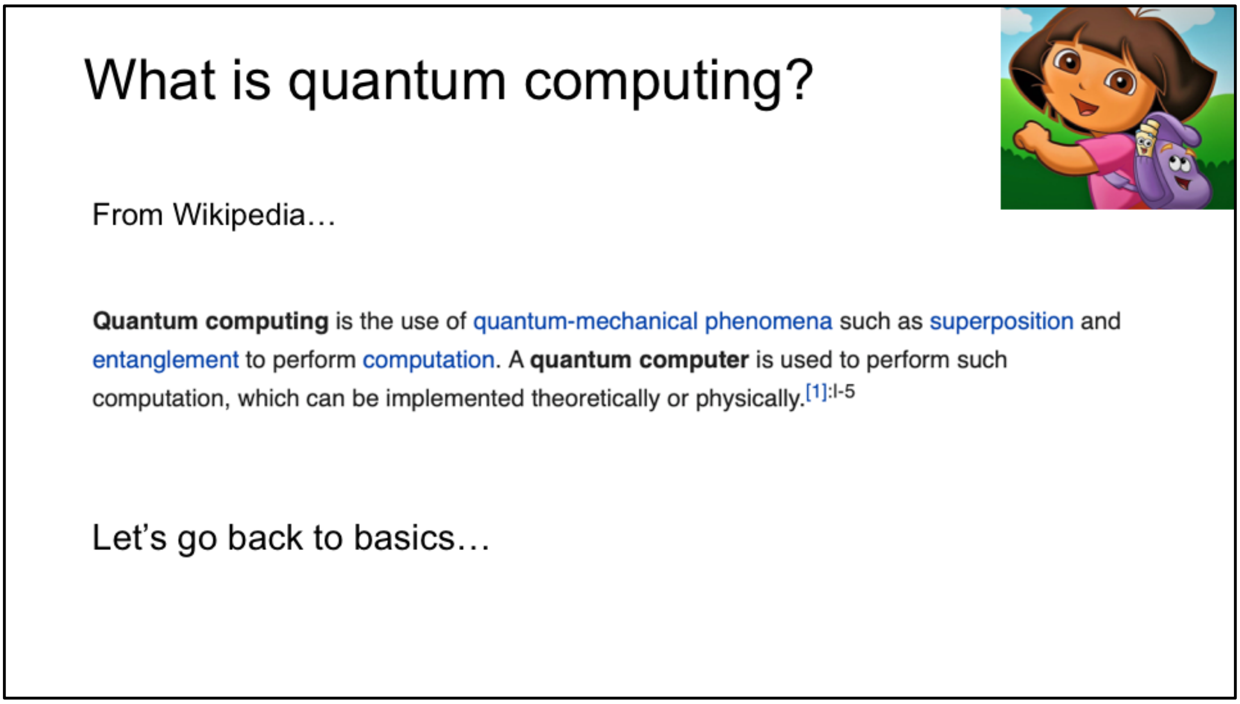
D-Wave's quantum computers 'can save energy, solve climate change'

SXSW 2018: Quantum Computers Could End Climate Change Debate, Says Expert

quantum computers are gaining computational power relative to classical ones at a "doubly exponential" rate — a staggeringly fast clip.

Suppose you're an intrepid explorer trying to find out what quantum computing is…

Lots of hype/nonsense going around. This level of hype familiar to everyone in relation to AI
What's it really about?



What is quantum computing?

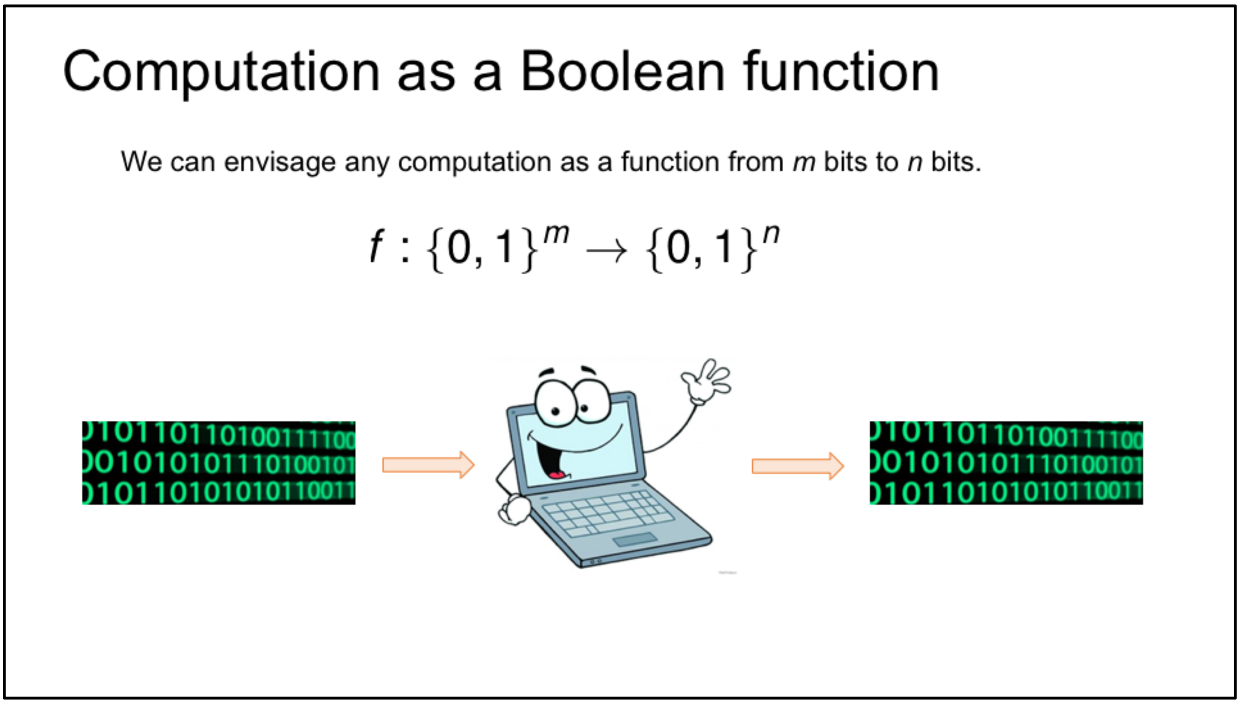
From Wikipedia…

Quantum computing is the use of quantum-mechanical phenomena such as superposition and entanglement to perform computation. A **quantum computer** is used to perform such computation, which can be implemented theoretically or physically.[1]:I-5

Let’s go back to basics…

Our intrepid explorer goes to Wikipedia for a more reasonable viewpoint…

I am going to take an unconventional route. Typically explanations like this Wikipedia one go via physics approach. We’ll take a more mathematical/CS approach here

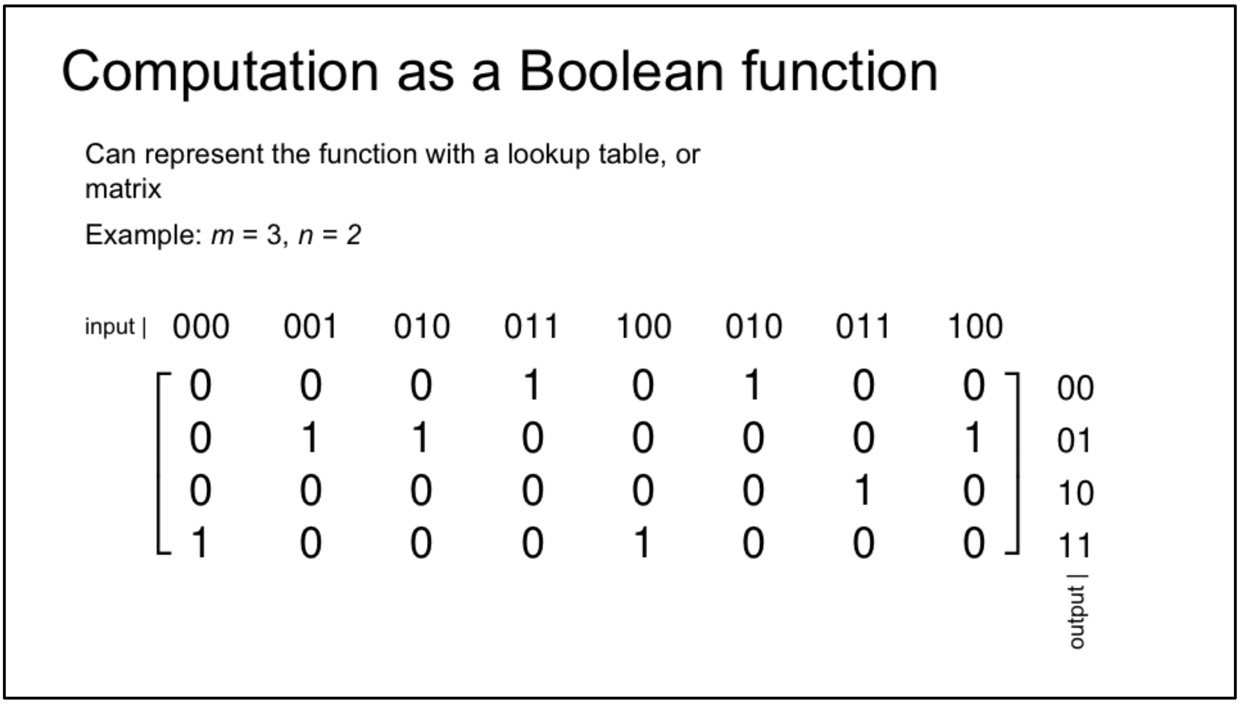


Computation as a Boolean function

We can envisage any computation as a function from *m* bits to *n* bits.

$$f : \{0, 1\}^m \to \{0, 1\}^n$$

I think we can all agree on this simplified model of a computation. Of course this isn’t a universal programmable computer. Programmability gets in the way of the main argument and can be put in later.



Computation as a Boolean function

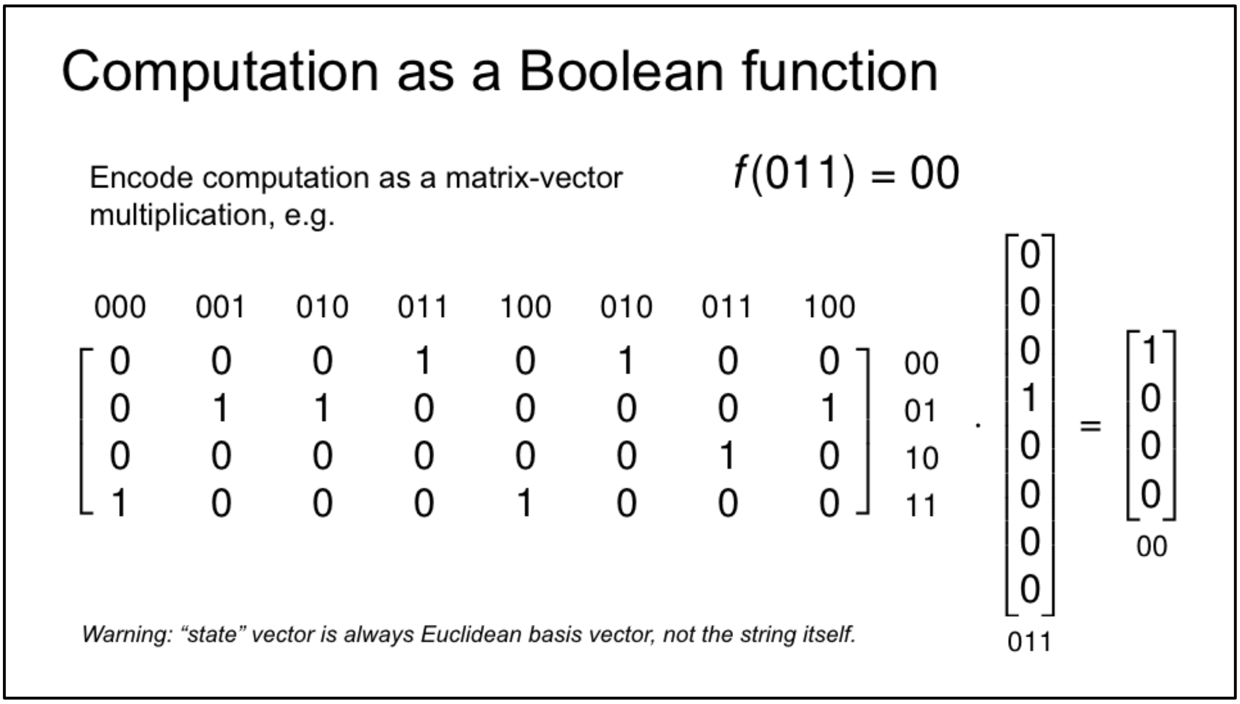
Can represent the function with a lookup table, or matrix

Example: *m* = 3, *n = 2*

$$\begin{array}{c|cccccccc|c}
\text{input |} & 000 & 001 & 010 & 011 & 100 & 010 & 011 & 100 & \\
\hline
 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 00 \\
 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 01 \\
 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 10 \\
 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 11 \\
\hline
 & & & & & & & & & \text{output |}
\end{array}$$

Talk through input and outputs – one iff input and output of function match, zero otherwise
Columns sum to one because every input has one unique output

We’ll show how this matrix model corresponds to something more familiar, and then use the matrix model framework to introduce quantum computing



Computation as a Boolean function

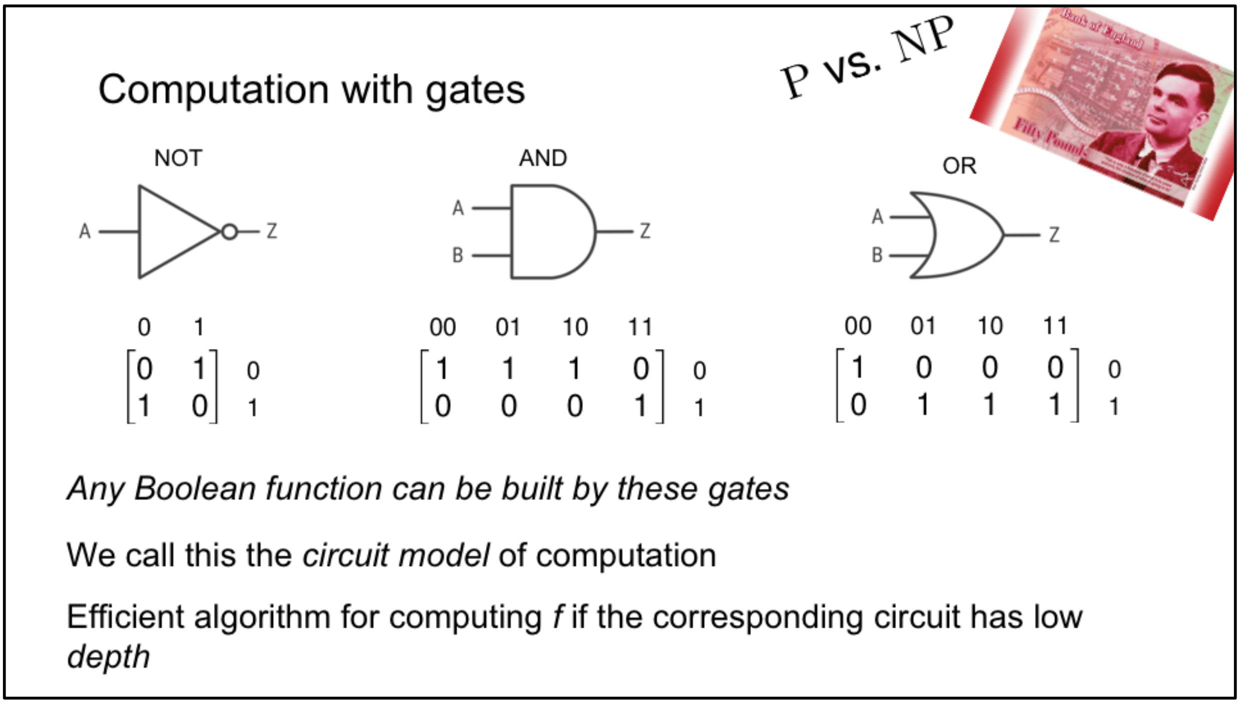
Encode computation as a matrix-vector multiplication, e.g.

$f(011) = 00$

$$\begin{array}{ccccccccc}
000 & 001 & 010 & 011 & 100 & 010 & 011 & 100 & \\
0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 00 \\
0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 01 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 10 \\
1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 11
\end{array}
\cdot
\underset{011}{\begin{bmatrix}0\\0\\0\\1\\0\\0\\0\\0\end{bmatrix}}
=
\underset{00}{\begin{bmatrix}1\\0\\0\\0\end{bmatrix}}$$

Warning: "state" vector is always Euclidean basis vector, not the string itself.

Important things are: representing bit string as a vector – unit vector, not the string itself. And Boolean function as a matrix



Computation with gates

P vs. NP

$$\begin{array}{cc|c} 0 & 1 & \\ \hline 0 & 1 & 0 \\ 1 & 0 & 1 \end{array}$$

$$\begin{array}{cccc|c} 00 & 01 & 10 & 11 & \\ \hline 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{array}$$

$$\begin{array}{cccc|c} 00 & 01 & 10 & 11 & \\ \hline 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 \end{array}$$

Any Boolean function can be built by these gates

We call this the *circuit model* of computation

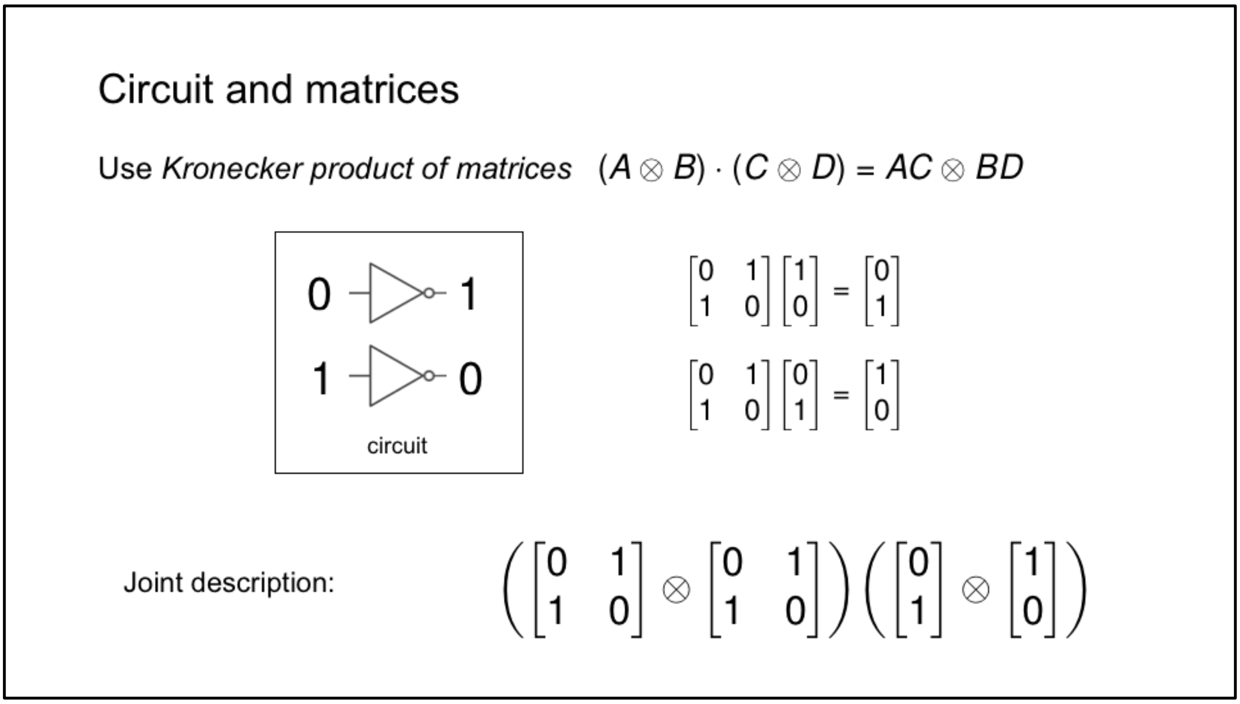
Efficient algorithm for computing *f* if the corresponding circuit has low *depth*

Let’s see the connection with a more traditional model of computation.

Remember Boolean logic gates from school. In our matrix vector language we can write them in the following ways.

<Walk through the gates>

Aside: million-dollar question of computer science, P vs NP. Everyone attacks this problem with circuit depth lower bounds



Circuit and matrices

Use *Kronecker product of matrices* $(A \otimes B) \cdot (C \otimes D) = AC \otimes BD$

0
1
1
0
circuit

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

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

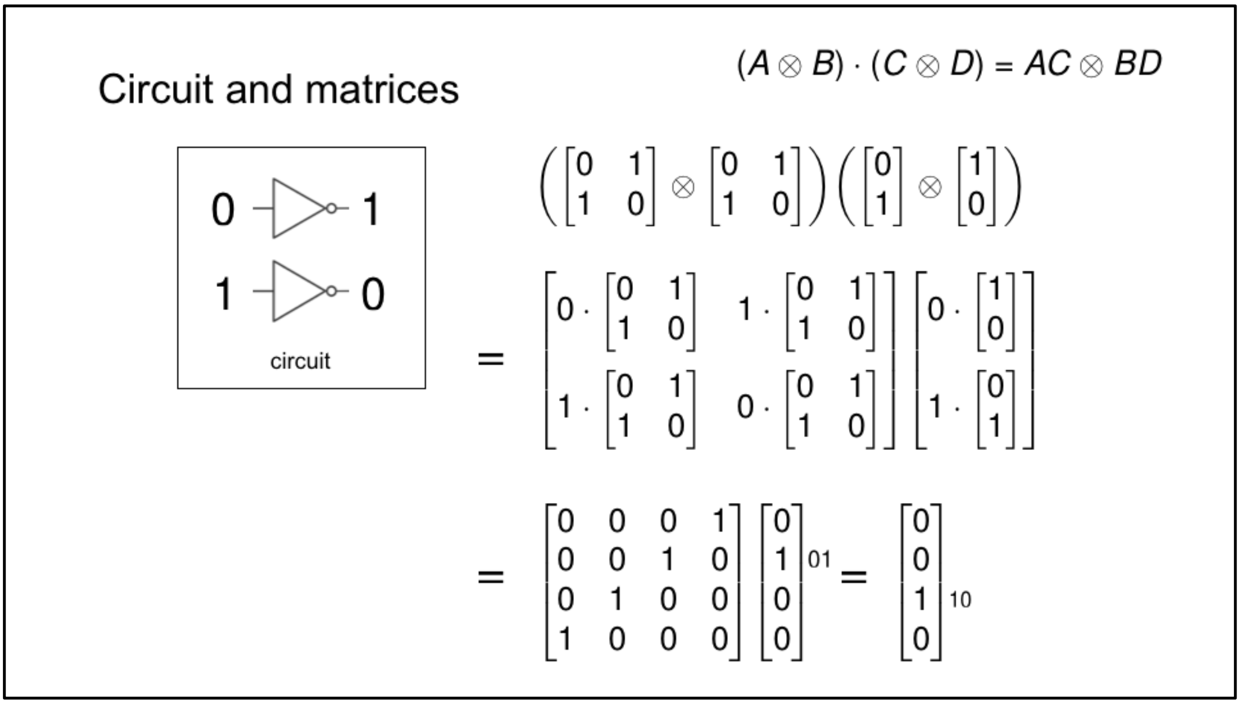
Joint description: $$\left(\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \otimes \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \right) \left(\begin{bmatrix} 0 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix} \right)$$

Kronecker product has this nice property. Only when A and C, B and D have complementary dimensions

Circuit on the left in our matrix language

First consider the two legs of the circuit separately

We then consider them at the same time with Kronecker product



Circuit and matrices

$$(A \otimes B) \cdot (C \otimes D) = AC \otimes BD$$

$$\left(\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \otimes \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}\right)\left(\begin{bmatrix} 0 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix}\right)$$

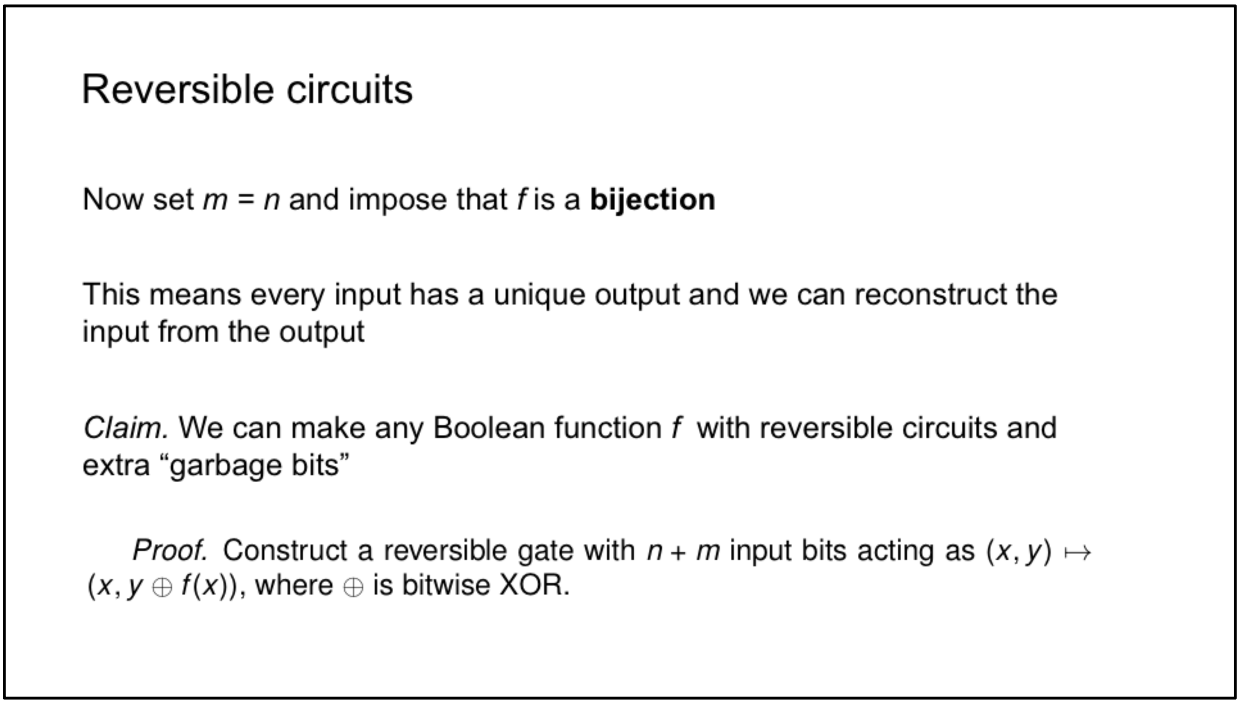
$$= \begin{bmatrix} 0 \cdot \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} & 1 \cdot \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \\ 1 \cdot \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} & 0 \cdot \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \end{bmatrix} \begin{bmatrix} 0 \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} \\ 1 \cdot \begin{bmatrix} 0 \\ 1 \end{bmatrix} \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}_{01} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}_{10}$$

MAIN TAKE HOME: correspondence between Boolean functions, circuits, matrix representation

We can build the big matrices from little ones, which correspond to elementary operations on bits. Universal set of gates corresponds to universal set of matrices

Converse, given any matrix we can decompose into gates



Reversible circuits

Now set $m = n$ and impose that f is a **bijection**

This means every input has a unique output and we can reconstruct the input from the output

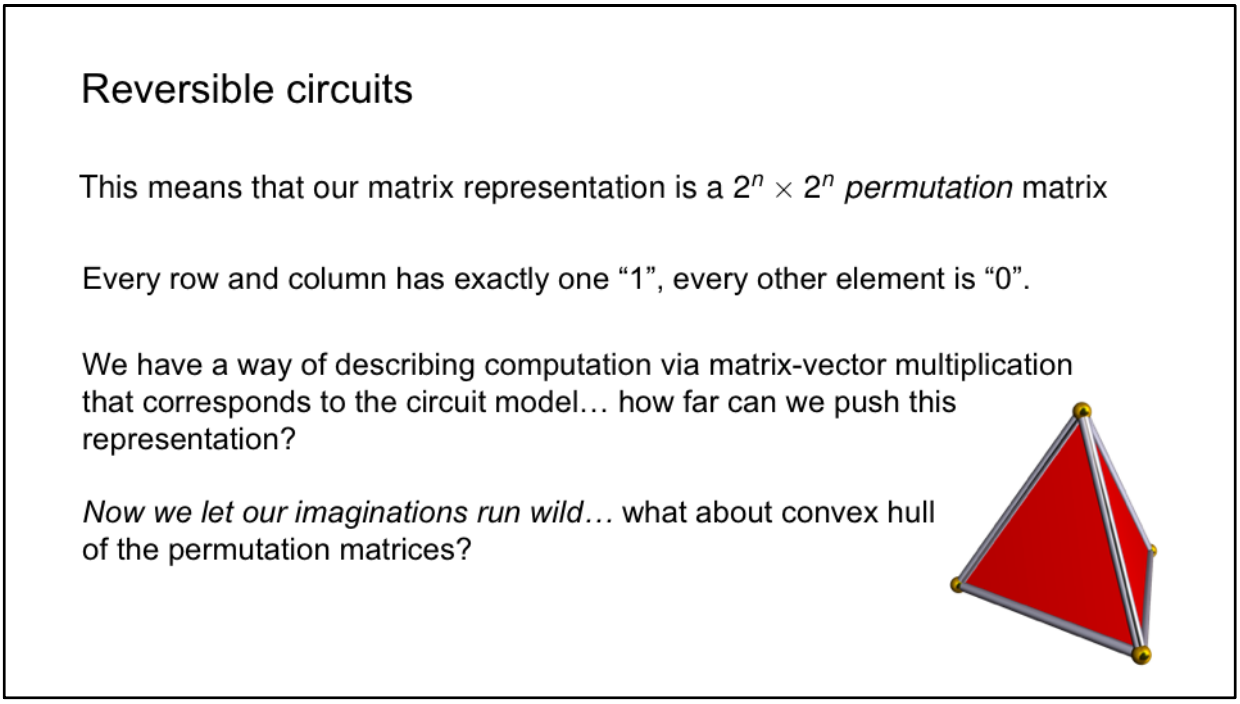
Claim. We can make any Boolean function f with reversible circuits and extra “garbage bits”

Proof. Construct a reversible gate with $n + m$ input bits acting as $(x, y) \mapsto (x, y \oplus f(x))$, where $\oplus$ is bitwise XOR.

Called reversible because you can reverse the output

Take home is: making the circuit reversible doesn’t remove computational power

We’re just doing this to make the maths easier



Reversible circuits

This means that our matrix representation is a $2^n \times 2^n$ *permutation* matrix

Every row and column has exactly one “1”, every other element is “0”.

We have a way of describing computation via matrix-vector multiplication that corresponds to the circuit model… how far can we push this representation?

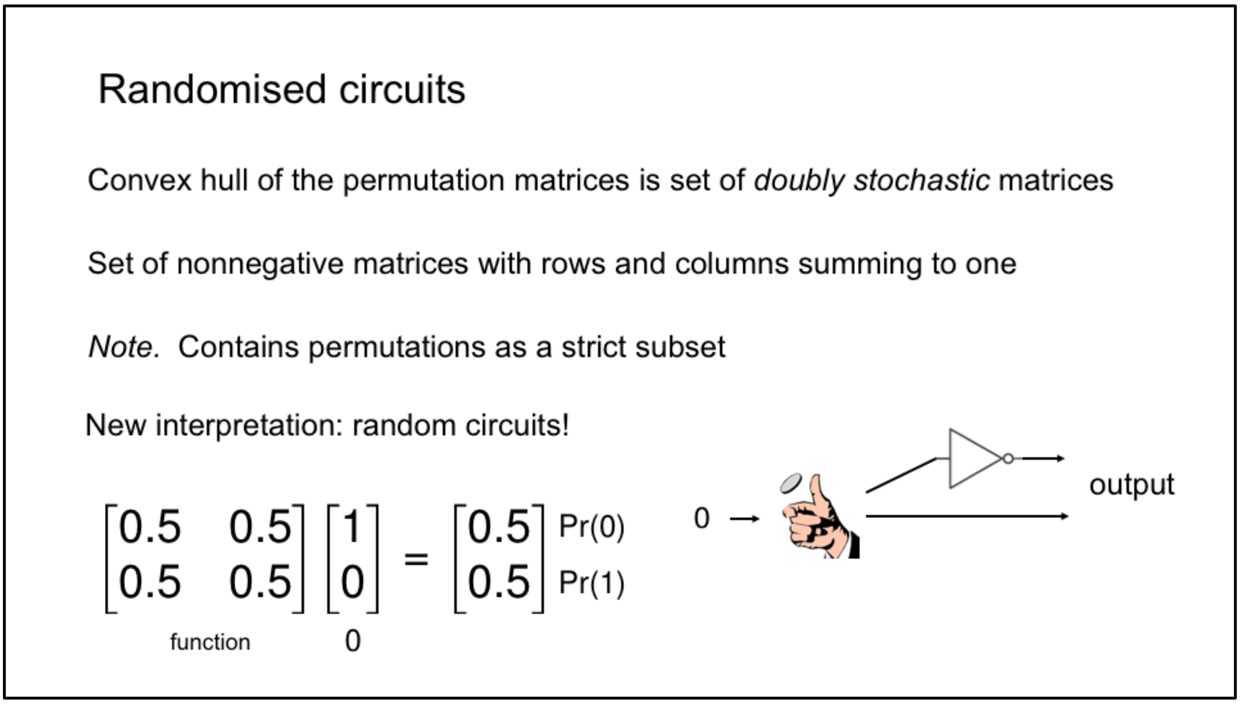
Now we let our imaginations run wild… what about convex hull of the permutation matrices?

Representation is weird but might be useful now

Reversible circuits have their own universal set – either TOFFOLI == CCNOT or Fredkin==CSWAP

Let’s change the representation and see if it implies a model of computation

Tetrahedron is the convex hull of any 4 points in 3D



Randomised circuits

Convex hull of the permutation matrices is set of *doubly stochastic* matrices

Set of nonnegative matrices with rows and columns summing to one

Note. Contains permutations as a strict subset

New interpretation: random circuits!

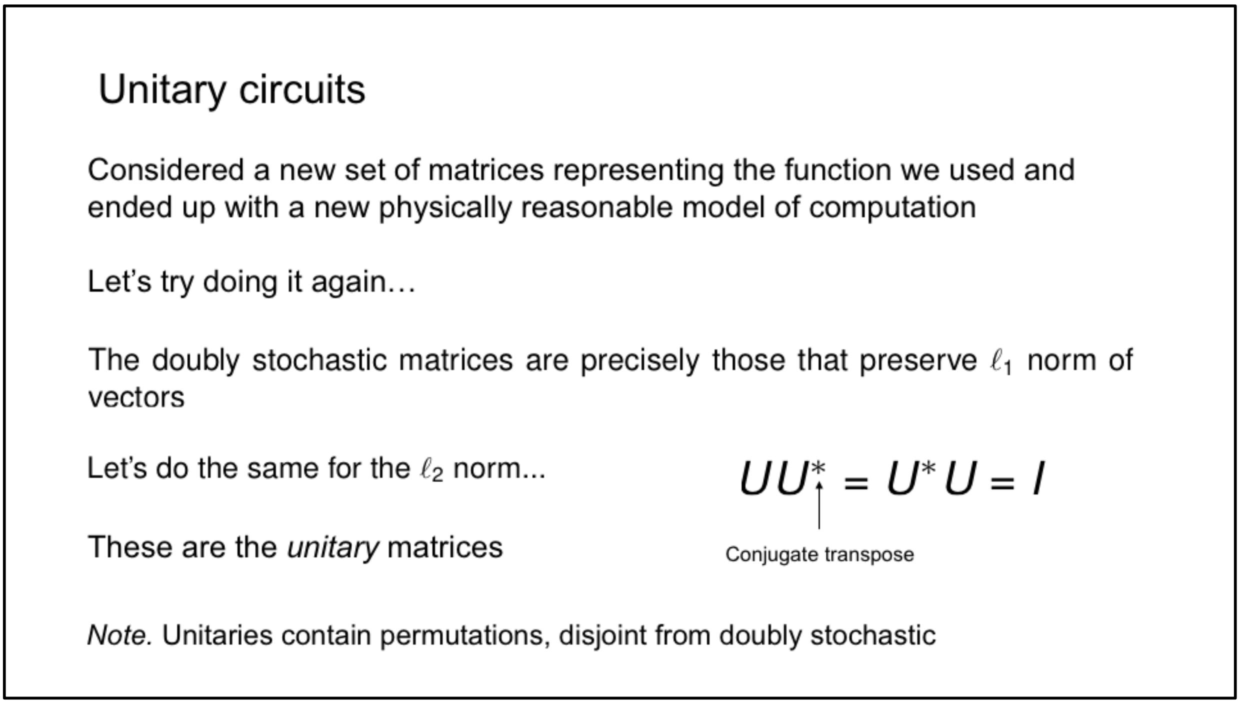
$$\underset{\text{function}}{\begin{bmatrix}0.5 & 0.5\\0.5 & 0.5\end{bmatrix}}\underset{0}{\begin{bmatrix}1\\0\end{bmatrix}}=\begin{bmatrix}0.5\\0.5\end{bmatrix}\begin{matrix}\Pr(0)\\\Pr(1)\end{matrix}$$

Anyone who likes Markov chains will recognize this matrix-vector multiplication (with everything transposed)

Random circuit: flip a coin and decide what circuit to use. Simple example here

This example: flip a coin and do a NOT if heads, nothing if tails

Vector elements correspond to probability of obtaining the corresponding bit



Unitary circuits

Considered a new set of matrices representing the function we used and ended up with a new physically reasonable model of computation

Let's try doing it again…

The doubly stochastic matrices are precisely those that preserve ℓ_1 norm of vectors

Let's do the same for the ℓ_2 norm...

$$U\underset{\substack{\uparrow\\ \text{Conjugate transpose}}}{U^*} = U^* U = I$$

These are the *unitary* matrices

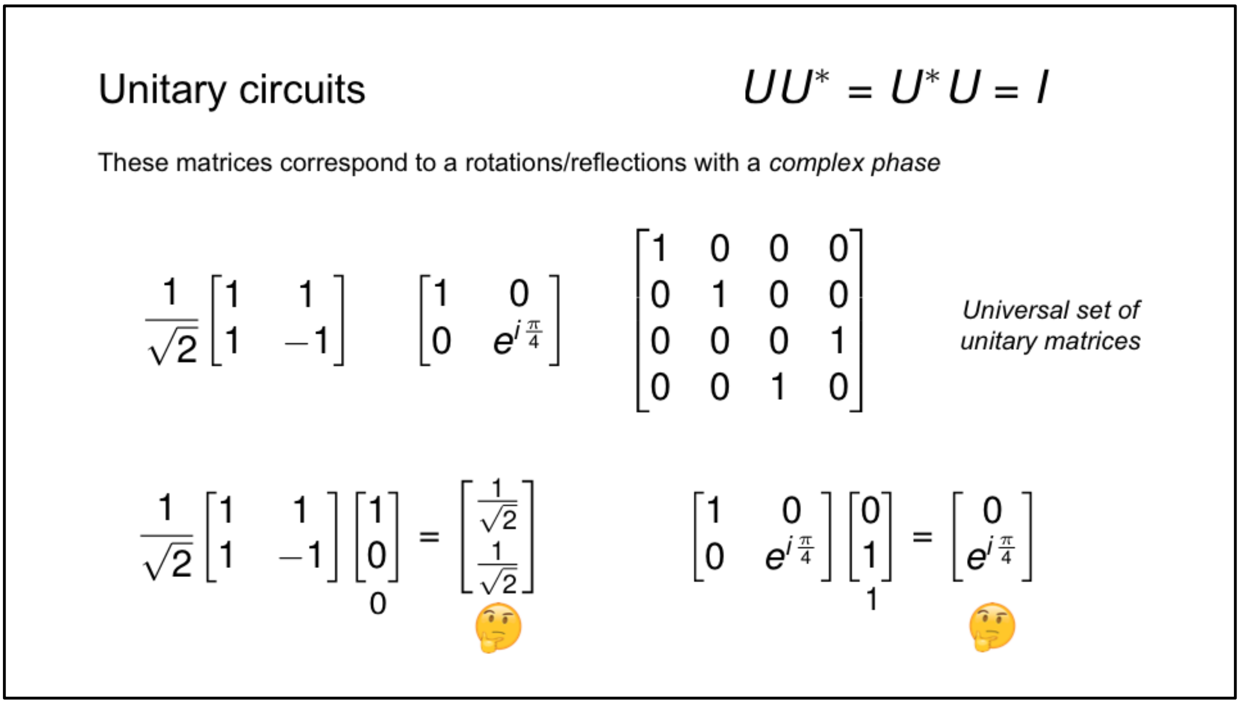
Note. Unitaries contain permutations, disjoint from doubly stochastic

Really cool that we can change the type of matrix and get a physically plausible model

Data Scientists like the l_1 norm, what about their other favourite norm, the l_2 norm?

Might remember unitary matrices from spectral decomposition or an SVD

Star is conjugate transpose



Unitary circuits

$$UU^* = U^* U = I$$

These matrices correspond to a rotations/reflections with a *complex phase*

$$\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix} \quad \begin{bmatrix}1 & 0\\ 0 & e^{i\frac{\pi}{4}}\end{bmatrix} \quad \begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & 1 & 0\end{bmatrix}$$

Universal set of unitary matrices

$$\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}\underset{0}{\begin{bmatrix}1\\ 0\end{bmatrix}} = \begin{bmatrix}\frac{1}{\sqrt{2}}\\ \frac{1}{\sqrt{2}}\end{bmatrix} \qquad \begin{bmatrix}1 & 0\\ 0 & e^{i\frac{\pi}{4}}\end{bmatrix}\underset{1}{\begin{bmatrix}0\\ 1\end{bmatrix}} = \begin{bmatrix}0\\ e^{i\frac{\pi}{4}}\end{bmatrix}$$

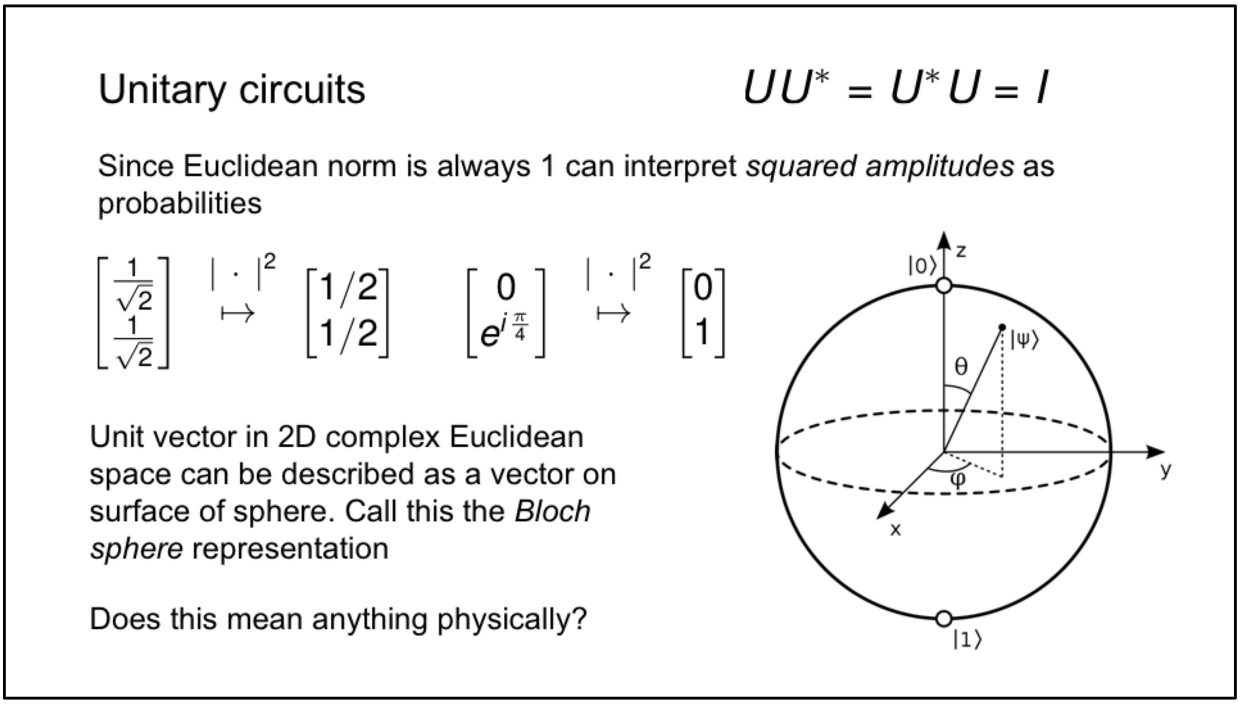
🤔 🤔

We have a set of universal unitary “gates” here, in that we can build any unitary matrix to arbitrary accuracy with these, in much the same way as with AND, OR and NOT

Let’s apply these matrices to the matrix representation of the bits 0 and 1.

Weird results… no obvious interpretation. Don’t add up to one and are complex-valued!

State of a qubit described by a Bloch sphere



Unitary circuits

$$UU^* = U^*U = I$$

Since Euclidean norm is always 1 can interpret *squared amplitudes* as probabilities

$$\begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} \overset{|\cdot|^2}{\mapsto} \begin{bmatrix} 1/2 \\ 1/2 \end{bmatrix} \qquad \begin{bmatrix} 0 \\ e^{i\frac{\pi}{4}} \end{bmatrix} \overset{|\cdot|^2}{\mapsto} \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Unit vector in 2D complex Euclidean space can be described as a vector on surface of sphere. Call this the *Bloch sphere* representation

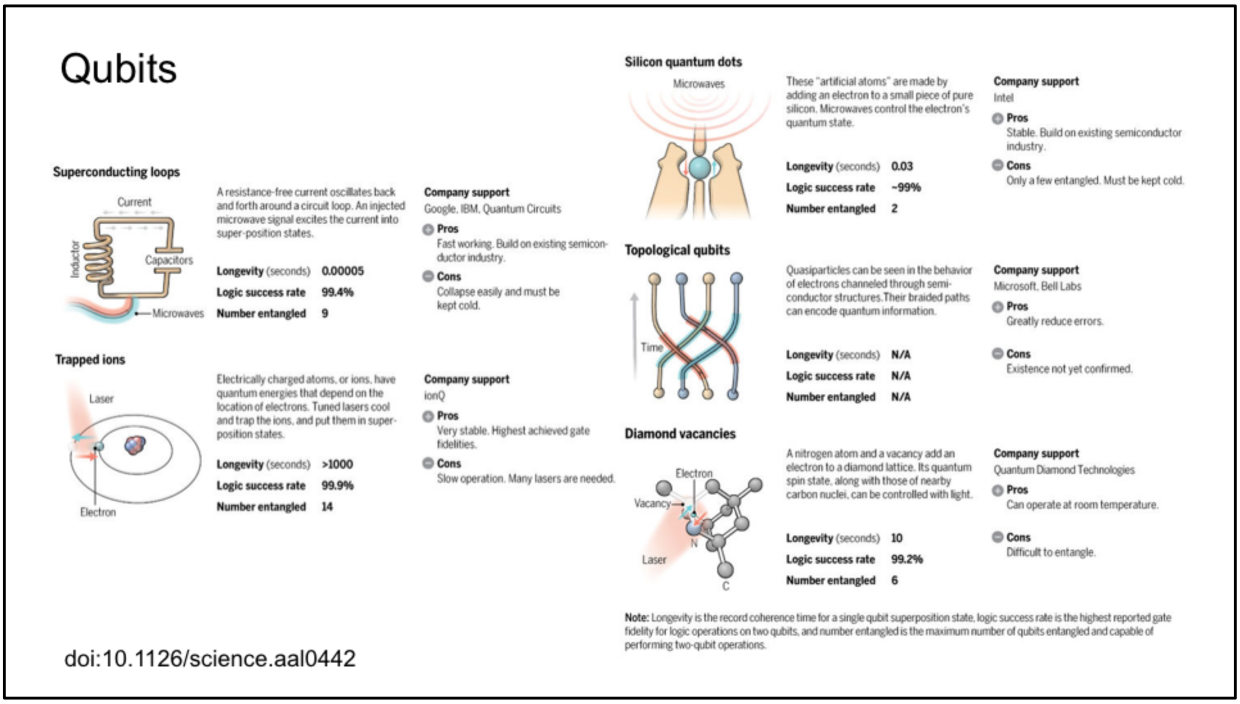
Does this mean anything physically?

If we look at the final states from previous slide, with this new interpretation of squared modulus we recover probabilities

State of these vectors described by a Bloch sphere. Action of unitary is as a rotation/reflection of the sphere

This is interesting since in our circuits we can have paths cancel out with each other

This is all well and good, but what does it actually physically mean anything or have we firmly ventured into la-la-land?



Qubits

Superconducting loops

A resistance-free current oscillates back and forth around a circuit loop. An injected microwave signal excites the current into super-position states.

Longevity (seconds)	**0.00005**
Logic success rate	**99.4%**
Number entangled	**9**

Company support
Google, IBM, Quantum Circuits

⊕ **Pros**
Fast working. Build on existing semiconductor industry.

⊖ **Cons**
Collapse easily and must be kept cold.

Trapped ions

Electrically charged atoms, or ions, have quantum energies that depend on the location of electrons. Tuned lasers cool and trap the ions, and put them in super-position states.

Longevity (seconds)	**>1000**
Logic success rate	**99.9%**
Number entangled	**14**

Company support
ionQ

⊕ **Pros**
Very stable. Highest achieved gate fidelities.

⊖ **Cons**
Slow operation. Many lasers are needed.

Silicon quantum dots

These "artificial atoms" are made by adding an electron to a small piece of pure silicon. Microwaves control the electron's quantum state.

Longevity (seconds)	**0.03**
Logic success rate	**~99%**
Number entangled	**2**

Company support
Intel

⊕ **Pros**
Stable. Build on existing semiconductor industry.

⊖ **Cons**
Only a few entangled. Must be kept cold.

Topological qubits

Quasiparticles can be seen in the behavior of electrons channeled through semiconductor structures.Their braided paths can encode quantum information.

Longevity (seconds)	**N/A**
Logic success rate	**N/A**
Number entangled	**N/A**

Company support
Microsoft, Bell Labs

⊕ **Pros**
Greatly reduce errors.

⊖ **Cons**
Existence not yet confirmed.

Diamond vacancies

A nitrogen atom and a vacancy add an electron to a diamond lattice. Its quantum spin state, along with those of nearby carbon nuclei, can be controlled with light.

Longevity (seconds)	**10**
Logic success rate	**99.2%**
Number entangled	**6**

Company support
Quantum Diamond Technologies

⊕ **Pros**
Can operate at room temperature.

⊖ **Cons**
Difficult to entangle.

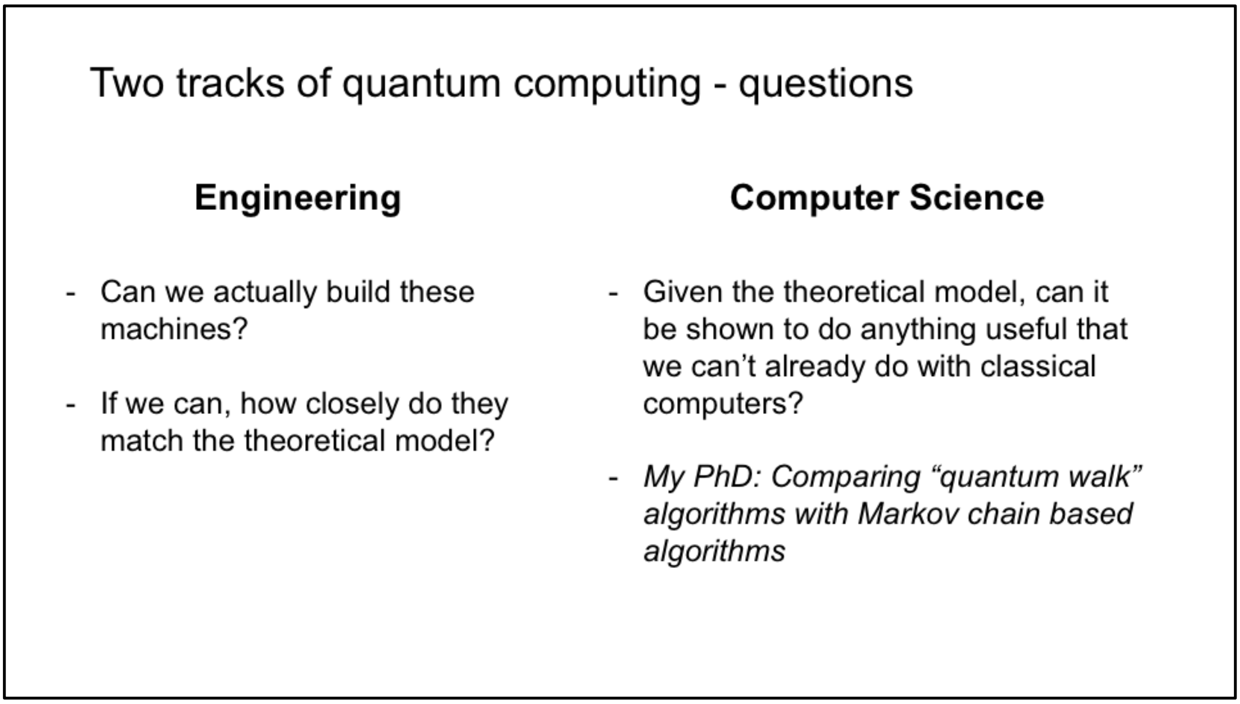
Note: Longevity is the record coherence time for a single qubit superposition state, logic success rate is the highest reported gate fidelity for logic operations on two qubits, and number entangled is the maximum number of qubits entangled and capable of performing two-qubit operations.

doi:10.1126/science.aal0442

Each of these physical systems behaves as a qubit, i.e. a Bloch sphere acted on by unitaries. Unitaries carried out via electrical and/or optical signals

I'm not an experimentalist so very much not an expert here

These devices have been built successfully, albeit on a small scale



Two tracks of quantum computing - questions

Engineering

- Can we actually build these machines?
- If we can, how closely do they match the theoretical model?

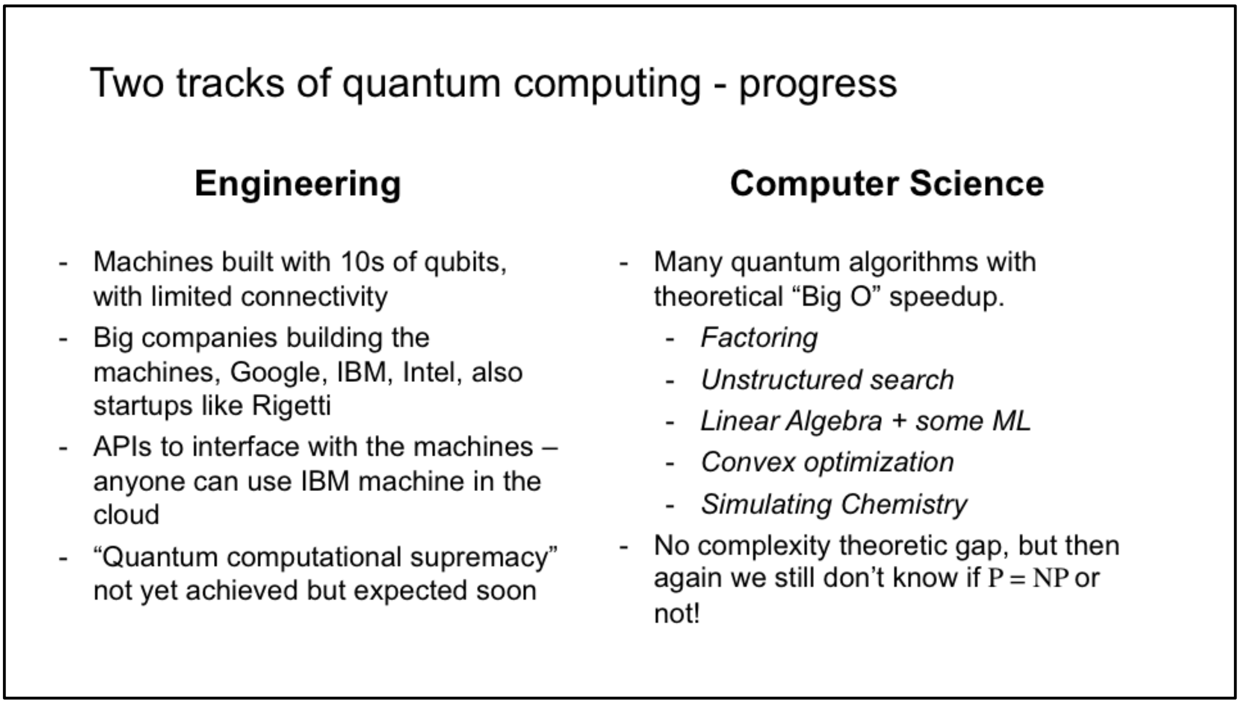
Computer Science

- Given the theoretical model, can it be shown to do anything useful that we can’t already do with classical computers?
- *My PhD: Comparing “quantum walk” algorithms with Markov chain based algorithms*

I like to think of the field as having two broad tracks, engineering and computer science

Matrix representation shows the increased space of functions accessible

My PhD, quantum walks are a “quantum” version of random walks on a graph. I have been comparing the mixing properties of quantum and classical random walks, as these are used in certain algorithms



Two tracks of quantum computing - progress

Engineering

- Machines built with 10s of qubits, with limited connectivity
- Big companies building the machines, Google, IBM, Intel, also startups like Rigetti
- APIs to interface with the machines – anyone can use IBM machine in the cloud
- “Quantum computational supremacy” not yet achieved but expected soon

Computer Science

- Many quantum algorithms with theoretical “Big O” speedup.
  - *Factoring*
  - *Unstructured search*
  - *Linear Algebra + some ML*
  - *Convex optimization*
  - *Simulating Chemistry*
- No complexity theoretic gap, but then again we still don’t know if $\mathrm{P} = \mathrm{NP}$ or not!

Current state of the art we have big companies like Google and IBM building small machines.

They’re not useful to anyone other than physicists for now, but they do implement the model we’ve talked about, so promising

Can log in to the IBM quantum experience and use a few different machines in the cloud

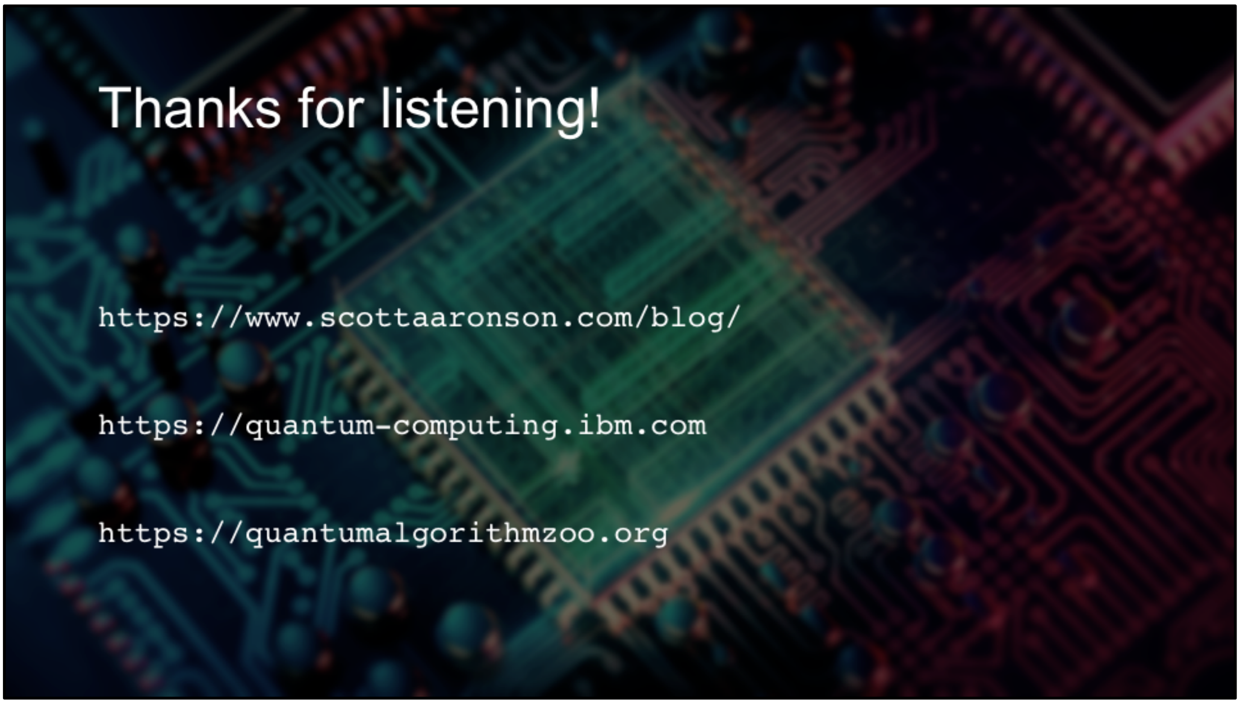
Quantum computational supremacy is the idea of carrying out a computational task that no classical machine (including a server farm) can do in a reasonable time, and doing it on QC

Algorithms show theoretical speedups asymptotically (Big O) but we don’t know if this will translate to real speedup

Some interesting use cases, factoring most famous. Some linear algebra ops used -> into ML applications. Convex Optimisation

Most compelling case is Quantum chemistry. Simulating quantum requires exponential resources, but QC can natively implement this

No unconditional complexity theory separation, but we can’t even do this for P and NP!



Thanks for listening!

https://www.scottaaronson.com/blog/

https://quantum-computing.ibm.com

https://quantumalgorithmzoo.org

If you’re interested in learning more, here are some good resources