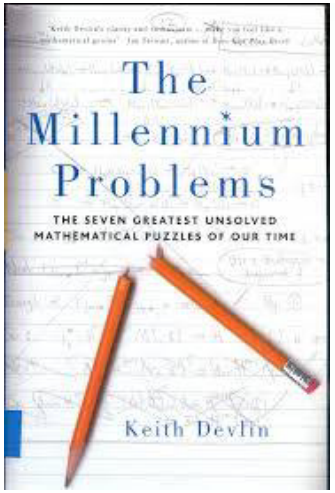
COMP90038
Algorithms and Complexity

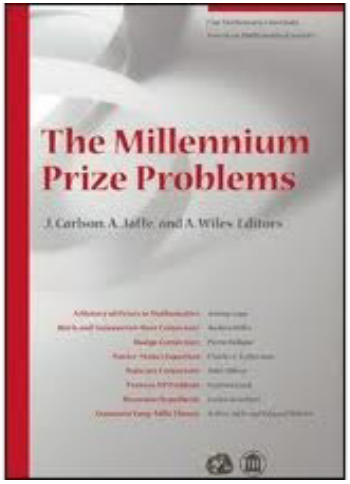
Lecture 23: Special Topics
Quantum Computing

Casey Myers
Casey.Myers@unimelb.edu.au
David Caro Building (Physics) 274

Review from Lecture 22: P vs NP

- The "P versus NP" problem comes from computational complexity theory.
- P means with polynomial time complexity
  - That is, algorithms that have $O(\text{poly}(n))$
  - Sorting is a type of polynomial time problem
- NP means non-deterministic polynomial
  - The answer can be checked in polynomial time, but cannot find the answer in polynomial time for large n.
  - The TSP problem is an NP problem.
- This is the most important question in Computer Science





Review from Lecture 22: NP-Completeness

- The main tool used to determine the class of a problem is reducibility.

- A decision problem P is said to be polynomially reducible to a decision problem Q, if there exists a function t that transforms instances of P to instances of Q such that:
  - t maps all yes instances of P to yes instances of Q and all no instances of P to no instances of Q
  - t is computable by a polynomial time algorithm.

- A decision problem D is said to be NP-complete if:
  - it belongs to class NP
  - Every problem in NP is polynomially reducible to D.

NP-Complete Problems

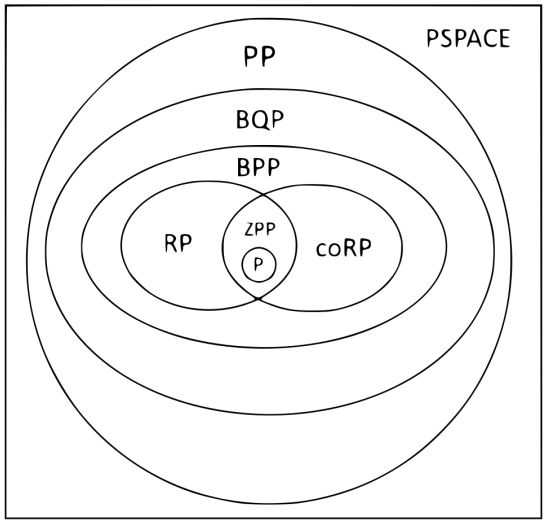
- SAT, SUBSET-SUM, 3COL, LPATH and HAM, as well as thousands of other interesting and practically relevant problems have been shown to be NP-complete.
- This explains the continuing interest in NP-completeness and related concepts from complexity theory.
- For such problems, we do not know of solutions that are essentially better than exhaustive search.
- There are many other interesting complexity classes, including space complexity classes and probabilistic classes.

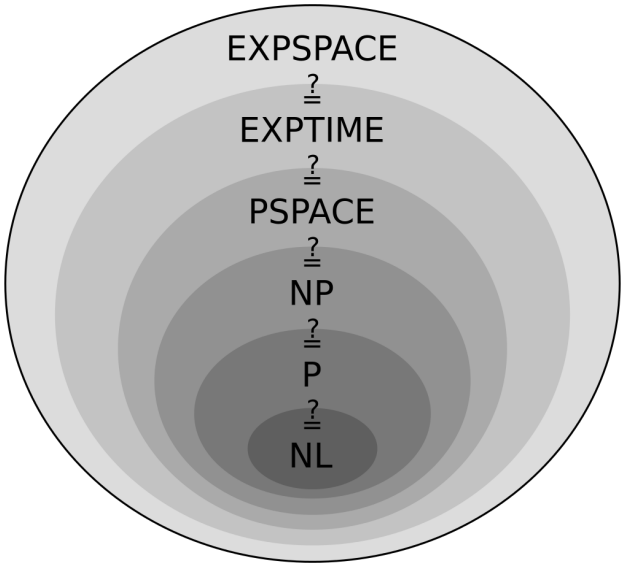
Review from Lecture 22: Open Problems

- We do not know whether P=NP, and there are many other unsolved problems.
- We know that $P \subseteq EXPTIME$, that is, there are problems that can be solved in exponential time but provably not in polynomial time.
- We also know:

$$P \subseteq RP \subseteq NP \subseteq PSPACE \subseteq NPSPACE \subseteq EXPTIME$$

- But which if these inclusions are strict? (Some must be; all could be…)

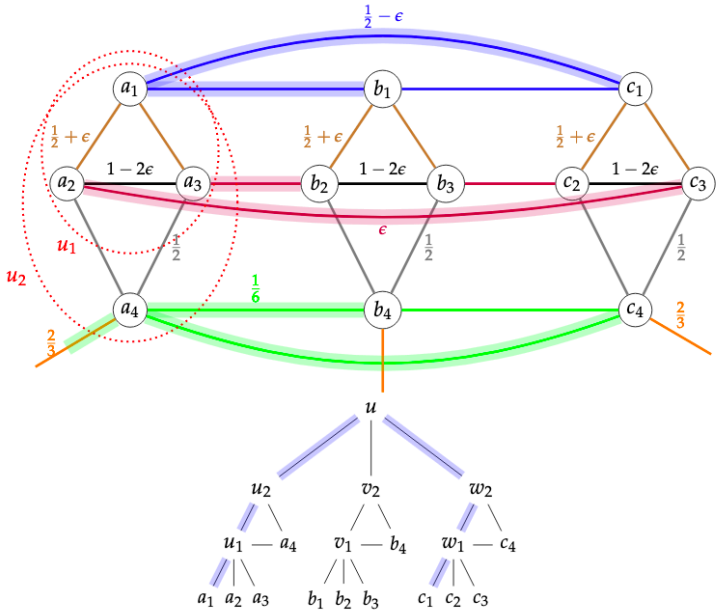




Progress with the Travelling Salesman Problem

- There has been recent incremental improvement to finding approximate solutions to the traveling salesman problem: [Computer Scientists Break Traveling Salesperson Record]

- Computer scientists were able to improve on the best approximate solution from 1976 by 0.2 billionth of a trillionth of a trillionth of a percent.

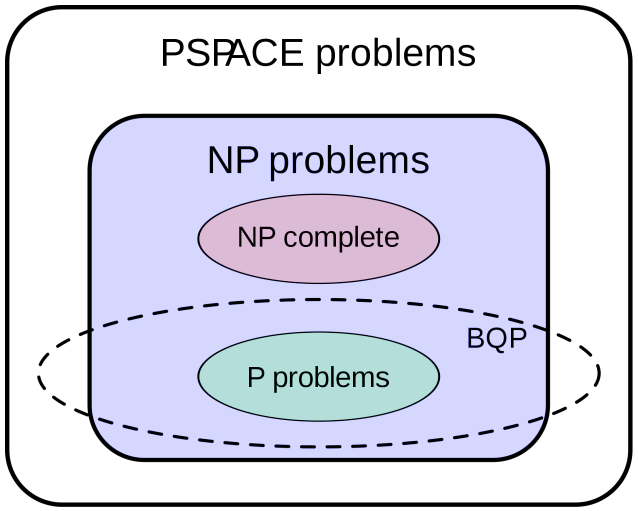


Quantum Computing Complexity

- The class of problems that can be efficiently solved by a quantum computer with bounded error is called BQP ("bounded error, quantum, polynomial time").

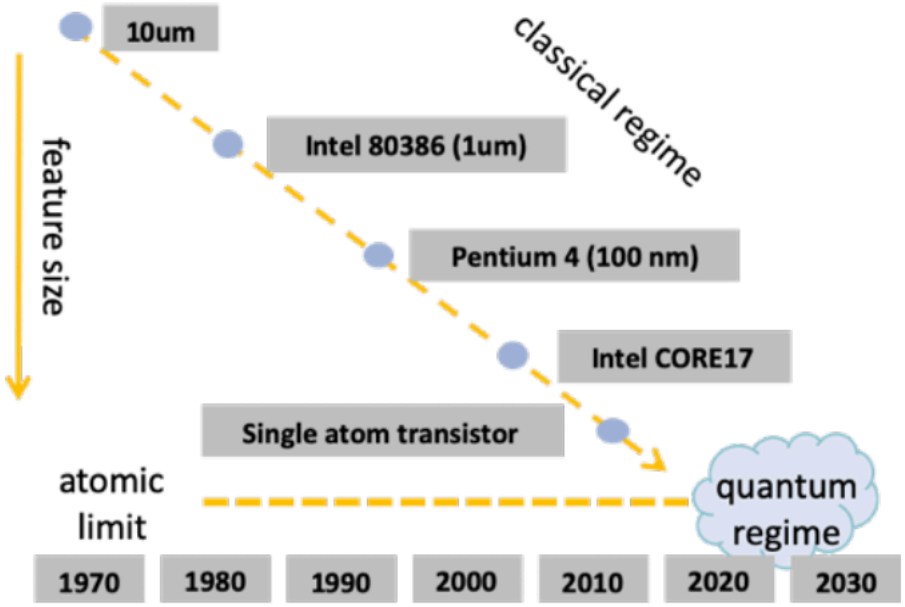
- It is known that $BPP \subseteq BQP$ (BPP="bounded error, probabilistic, polynomial time"), and it is widely suspected that $BQP \nsubseteq BPP$.

- The relationship of BQP to the basic classical complexity classes can be summarised as: P $\subseteq BPP \subseteq BQP \subseteq PP \subseteq PSPACE$, where PP = is a class of decision problems solvable by a "probabilistic Turing machine in polynomial time".

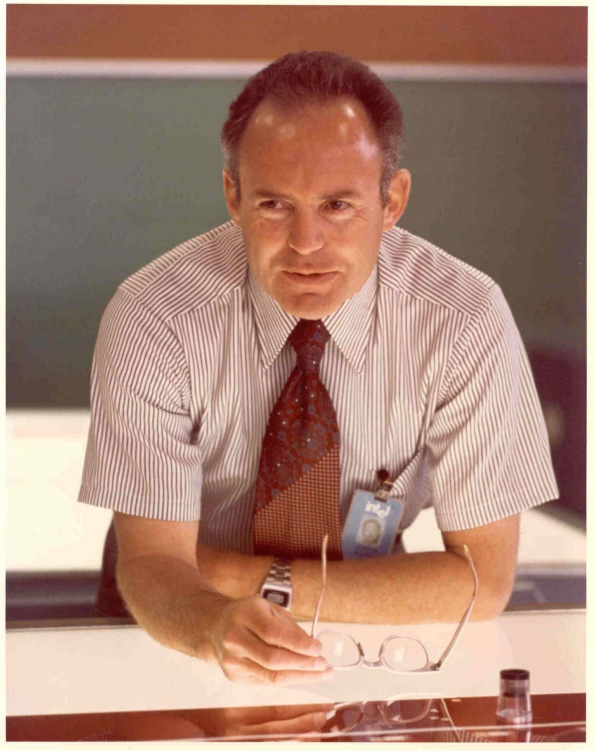


Motivation for Quantum Computing

- Inevitable?
  - As transistor size $< 10nm$, the limit to classical mechanics is quickly approaching (atoms are about $1nm$ or less).



Moore's Law

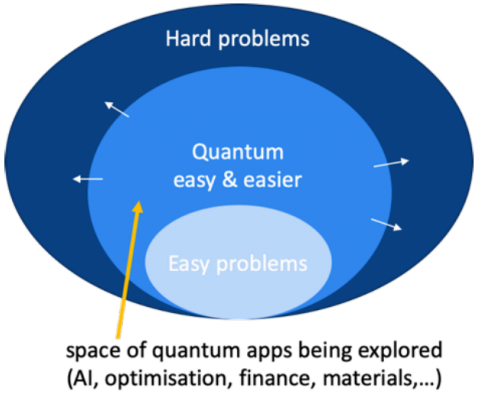


"The number of transistors incorporated in a chip will approximately double every 24 months"

— Gordon Moore, 1965

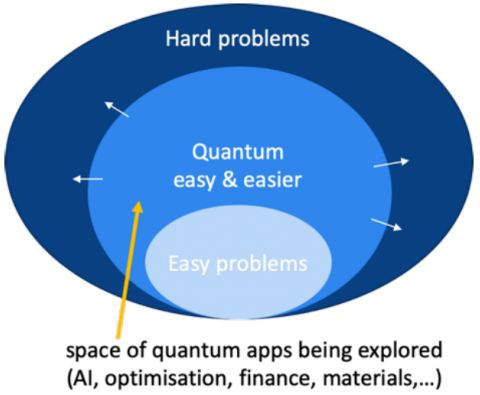
Motivation for Quantum Computing

- More powerful?
  - Quantum computing algorithms have been proposed that solve certain computational problems faster than classical computers.

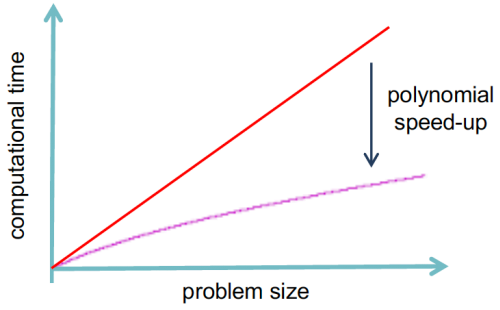


Motivation for Quantum Computing

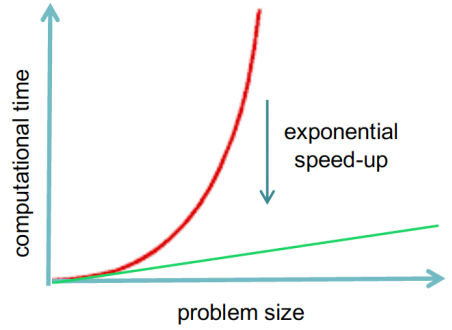
- More powerful?
  - Quantum computing algorithms have been proposed that solve certain computational problems faster than classical computers.



- There are broadly two classes of quantum computer.
  - Intermediate quantum computers (NISQ)
    - Medium-scale system (10 ~ 1000 qubits)
    - Potentially polynomial speed-up
    - Optimisation, machine learning, chemistry, finance



  - Full universal quantum computers
    - Large-scale system (≫ 1000 qubits)
    - Potentially exponential speed-up (Shor's)
    - Finance, bio-molecular simulation, database analysis

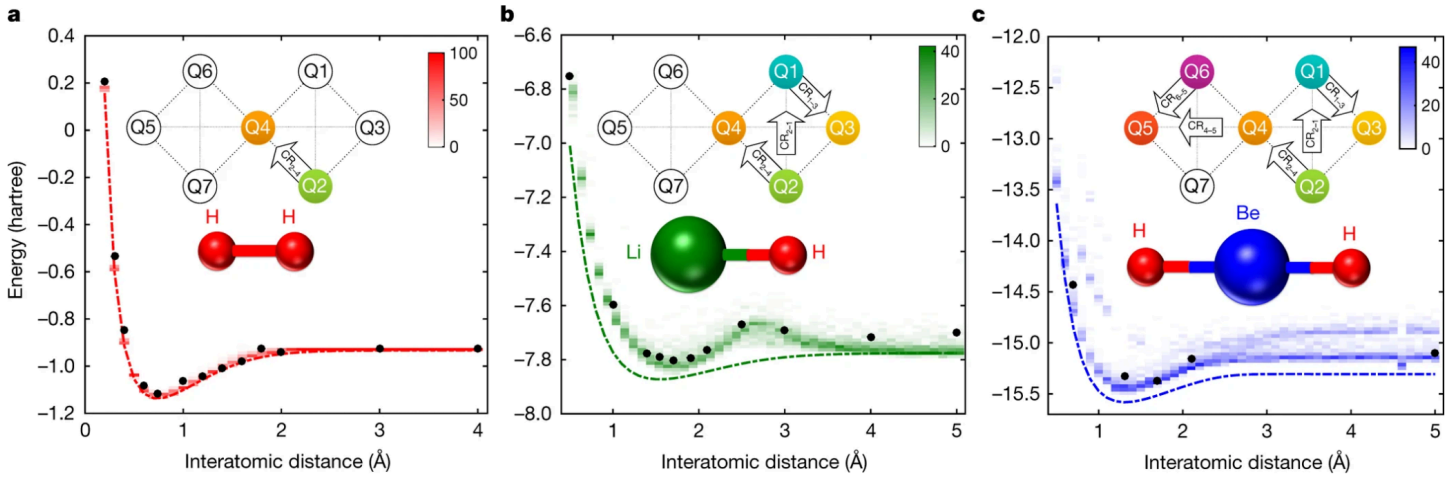


Motivation for Quantum Computing

- Cool?

- Quantum computing could be used to simulate physical systems that the would be impossible on the largest supercomputers.

  - Condensed matter systems
  - Quantum Chemistry
  - Particle physics



Quantum Computing Basics: Qubits

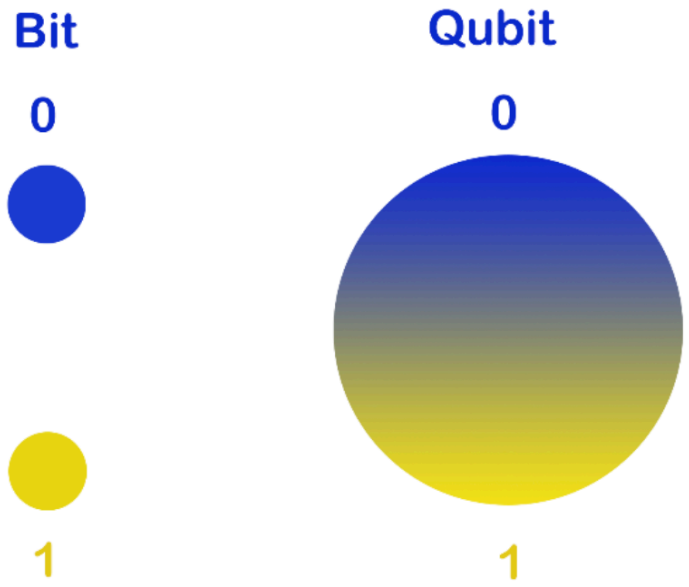
- Quantum computers use quantum phenomenon such as superposition, entanglement and quantum measurement to perform computation.

- **Quantum superposition**: systems can be in indeterminate (multiple) states prior to measurement.

- **Quantum entanglement**: systems can be linked such that measurement of one part correlates to that of another part.

- **Quantum measurement**: result of any given measurement a-priori unknown, system "collapses" to an outcome.

Bits vs Qubits



Quantum Computing Basics: Qubits

- A **bit** can be either a 0 or a 1:  OR 

- Qubits can be in **superpositions:**

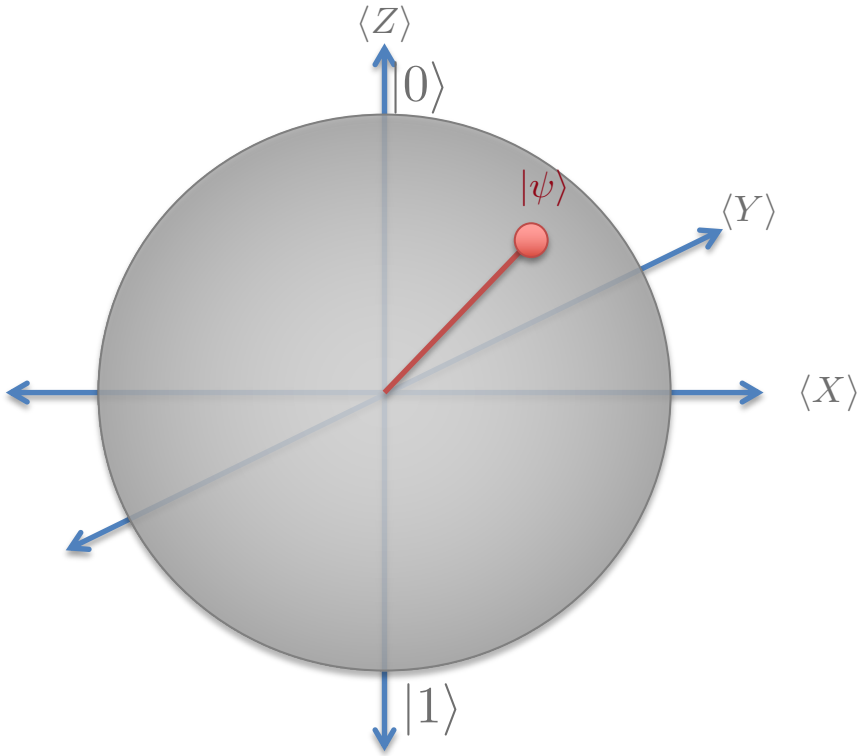
$$\boldsymbol{\alpha}|\mathbf{0}\rangle + \boldsymbol{\beta}|\mathbf{1}\rangle = \boldsymbol{\alpha}|$$  $$\rangle + \boldsymbol{\beta}|$$  $$\rangle$$

where: $|\mathbf{0}\rangle \coloneqq \binom{\mathbf{1}}{\mathbf{0}}$ and $|\mathbf{1}\rangle \coloneqq \binom{\mathbf{0}}{\mathbf{1}}$.

- If we were to measure we would randomly obtain "0" with probability $|\boldsymbol{\alpha}|^2$ or "1" with probability $|\boldsymbol{\beta}|^2$.

- Since probabilities must sum to 1:

$$|\boldsymbol{\alpha}|^2 + |\boldsymbol{\beta}|^2 = 1$$



Bloch Sphere

Exponential Scaling of Computational Space

- 3 bits vs. 3 qubits

  3 bits: **010**

  3 qubits:

  $$\boldsymbol{a|000\rangle + b|001\rangle + c|010\rangle + d|011\rangle + e|100\rangle + f|101\rangle + g|110\rangle + h|110\rangle}$$

- 300 bits vs. 300 qubits

  300 bits: **0100101000 0100101000 0100101000 0100101000 0100101000**
  0100101000 0100101000 0100101000 0100101000 0100101000
  0100101000 0100101000 0100101000 0100101000 0100101000
  0100101000 0100101000 0100101000 0100101000 0100101000
  0100101000 0100101000 0100101000 0100101000 0100101000
  0100101000 0100101000 0100101000 0100101000 0100101000

  300 qubits: ?

  $2^{300} \sim 10^{90}$ dimensional state space!

  Total number of atoms in the universe $\sim 10^{78}$.

Quantum Computing Basics: Entanglement

- Entanglement is a special property of the quantum world with no classical analogue.

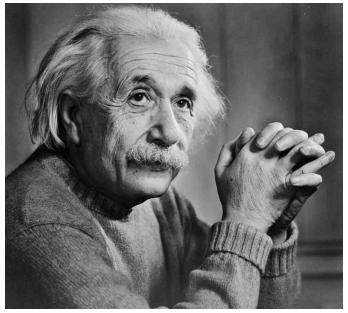
Independent

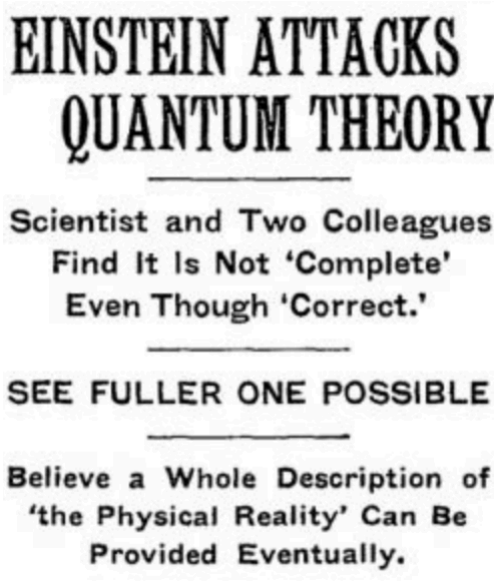
		
	25%	25%
	25%	25%

Entangled

		
	50%	0%
	0%	50%

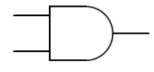
$$\frac{|00\rangle+|11\rangle}{\sqrt{2}} = \frac{1}{\sqrt{2}}\ (|\ \text{  }\ \rangle + |\ \text{  }\ \rangle\ )$$





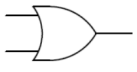
- Even after more than 100 years of research, we still don't fully understand entanglement.
- Entanglement enables
  - Teleportation: transfer of quantum information
  - Dense coding: encoding two classical bit of information per qubit

Gates on Bits Vs. Operations on Qubits



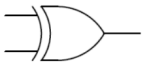
AND

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



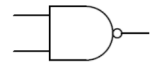
OR

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



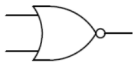
XOR

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



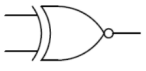
NAND

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



NOR

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



XNOR

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

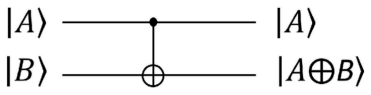
- **Logical Gates**
  - AND, OR, NOT, XOR, NAND, …
  - Acts on bit(s).
  - Final output is a well-defined bit (single digit).

- **Quantum Gates = Operations**
  - X, Z, CNOT, H, Toffoli, …
  - Acts in qubit(s): can change amplitude and/or phase [complex numbers].
  - Final output can be a superposition state and measurement is done to determine the value.

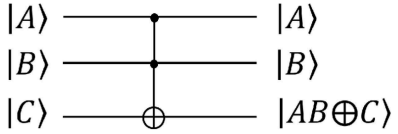
NOT Gate $\alpha|0\rangle + \beta|1\rangle \; —[X]— \; \alpha|1\rangle + \beta|0\rangle$

Z Gate $\alpha|0\rangle + \beta|1\rangle \; —[Z]— \; \alpha|0\rangle - \beta|1\rangle$

CNOT Gate



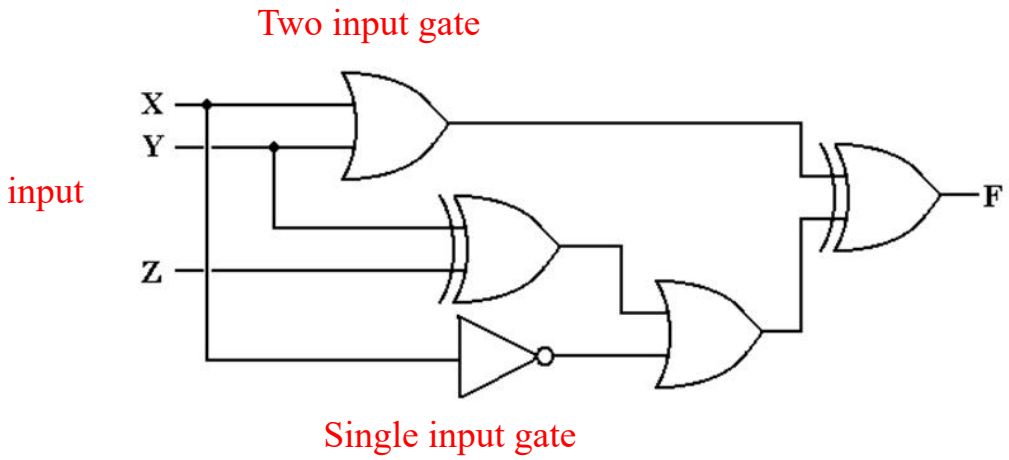
Toffoli Gate



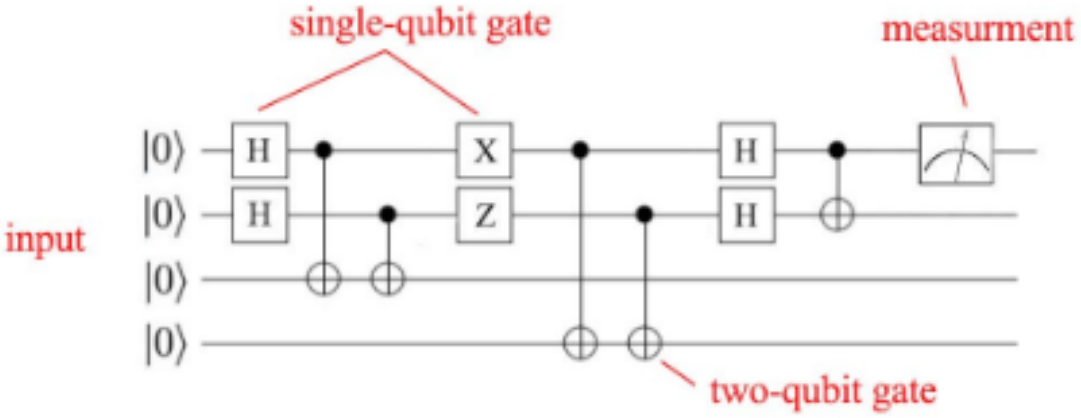
Hadamard Gate $\alpha|0\rangle + \beta|1\rangle \; —[H]— \; \alpha\frac{(|0\rangle + |1\rangle)}{\sqrt{2}} + \beta\frac{(|0\rangle - |1\rangle)}{\sqrt{2}}$

Circuit Model

Classical Digital Circuit



Quantum Circuit



Complex numbers

Superposition

Probabilistic amplitudes

Entanglement

Measurement

Time steps

Example 1: Deutsch-Jozsa Algorithm

- One of the first quantum algorithms proposed (1992) that showed a quantum advantage over classical computing.
- No well-defined real-world application.
- Inspiration for the inspiration for Shor's famous algorithm.
- Problem statement:

Given a Boolean function, $f(x)$, determine if:
$f(x)$ is constant (always gives the same result), or
$f(x)$ is balanced (gives equal numbers of 0's and 1's).

Example 1: Deutsch-Jozsa Algorithm

- Examples: for an input "x" of "n" bits, the function $f(x)$ is:

n=1

CONSTANT	f(0) = 0 f(1) = 0
	f(0) = 1 f(1) = 1
BALANCED	f(0) = 0 f(1) = 1
	f(0) = 1 f(1) = 0

n=2

CONSTANT	f(00) = 0 f(01) = 0 f(10) = 0 f(11) = 0
	f(00) = 1 f(01) = 1 f(10) = 1 f(11) = 1
BALANCED	f(00) = 0 f(01) = 0 f(10) = 1 f(11) = 1
	f(00) = 1 f(01) = 1 f(10) = 0 f(11) = 0

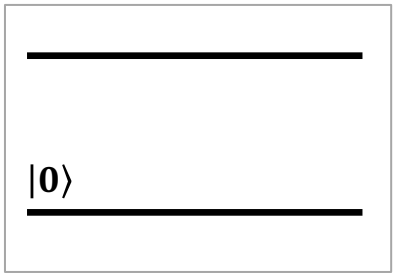
- **Elementary operation**: evaluation of $f(x)$ for x [depends on n].
- **Classical algorithm** (worst case) needs $(2^n/2)+1$ queries
  Time complexity = $O(2^n)$
- **Quantum algorithm** needs just 1 query
  Time complexity = $O(1)$.

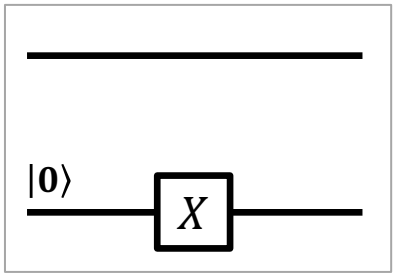
Example 1: Deutsch-Jozsa Algorithm

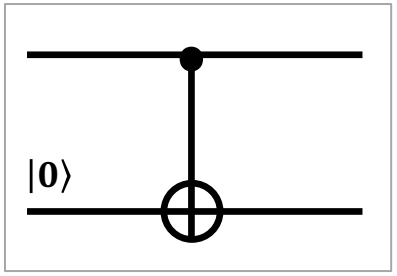
- For the one qubit input case, the implementations for 4 choices of function are:

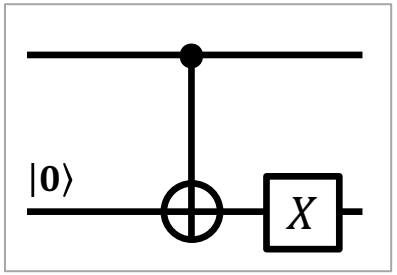
n=1

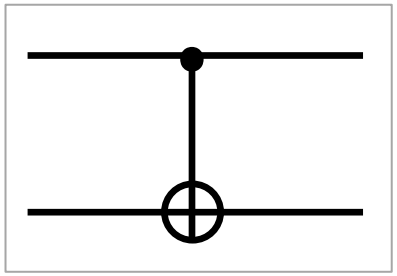
CONSTANT	f(0) = 0 f(1) = 0
	f(0) = 1 f(1) = 1
BALANCED	f(0) = 0 f(1) = 1
	f(0) = 1 f(1) = 0







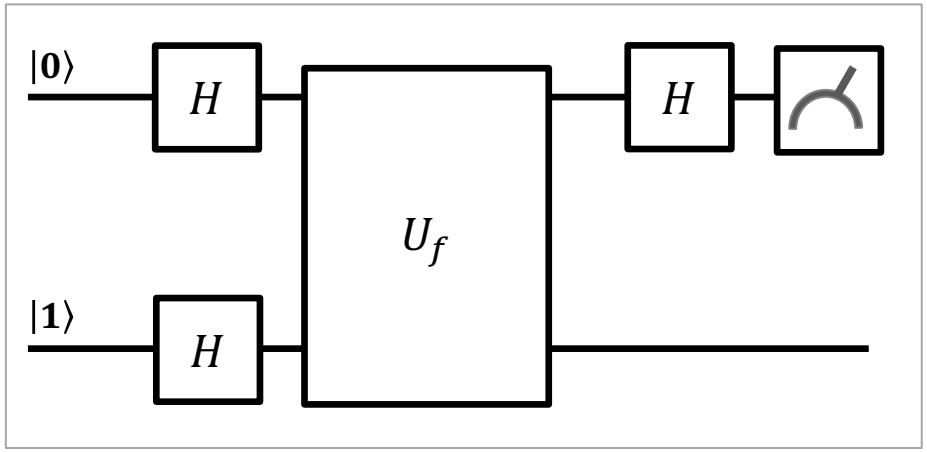


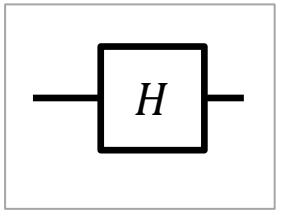


$$= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

$$|00\rangle \rightarrow |00\rangle$$
$$|01\rangle \rightarrow |01\rangle$$
$$|10\rangle \rightarrow |11\rangle$$
$$|11\rangle \rightarrow |10\rangle$$

Example 1: Deutsch-Jozsa Algorithm





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

$$|0\rangle \rightarrow \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

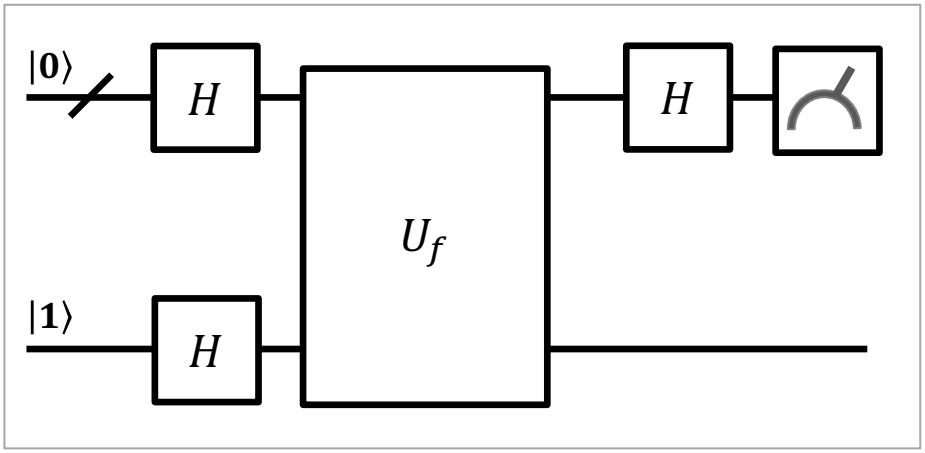
$$|1\rangle \rightarrow \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

		Measured
CONSTANT	f(0) = 0 f(1) = 0	0
	f(0) = 1 f(1) = 1	0
BALANCED	f(0) = 0 f(1) = 1	1
	f(0) = 1 f(1) = 0	1

The Deutsch-Jozsa algorithm determines in just *one query* whether the function is constant or balanced.

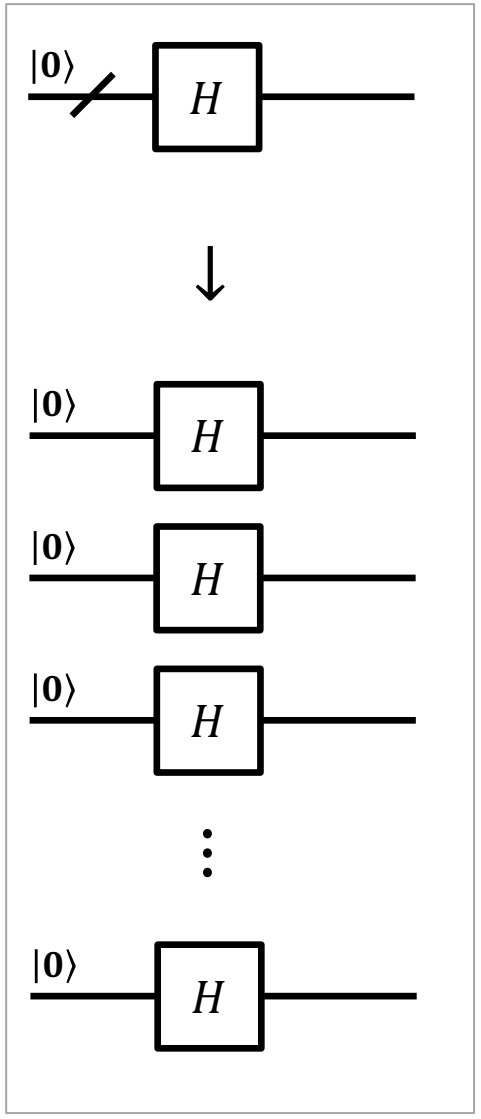
Classically, this would require *two queries*.

Example 1: Deutsch-Jozsa Algorithm



This multi-qubit Deutsch-Jozsa algorithm will still require only *one query* to determine whether the function is constant or balanced.

Note: 0's in the output register → constant, any other value → balanced.

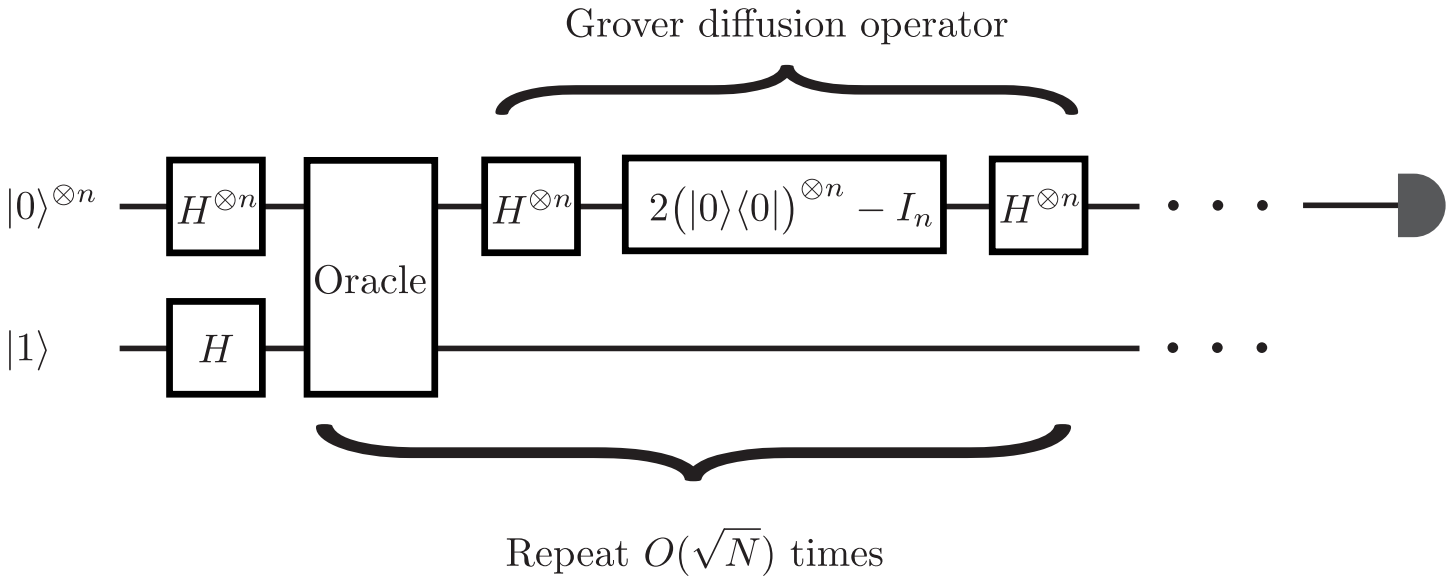


Example 2: Grover's Search Algorithm

- One of the first (1996) quantum algorithms to show a speed-up compared to it's classical counterpart for a useful problem.

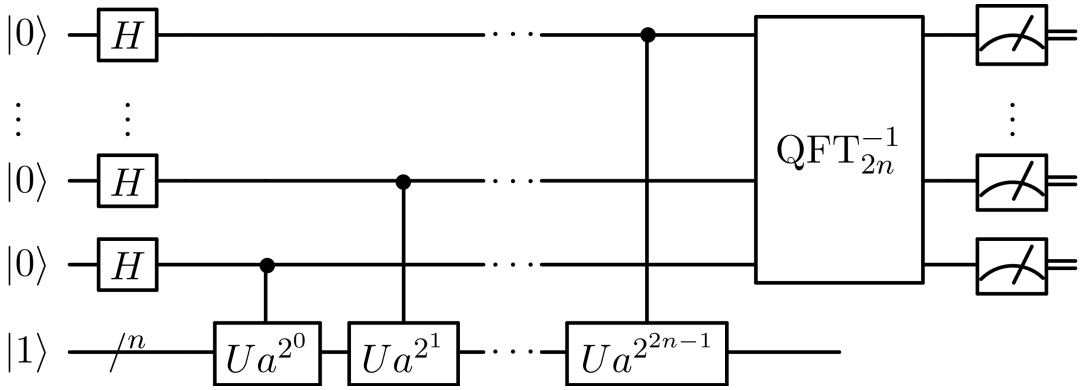
- Based on amplitude amplification, something that is uniquely available to quantum systems. Has formed the basis for many future quantum algorithms.

- **Problem:** An unsorted list of N entries could be searched to find a marked item with $O(\sqrt{N})$ steps instead of the best classical result of $O(N)$.



Example 3: Shor's Factoring algorithm

- The first (1994) quantum algorithms to show an exponential speed-up compared to it's classical counterpart.
- Is a generalisation of the quantum phase estimation algorithm: $U|\phi\rangle=e^{2i\theta}|\phi\rangle,\ 0 \leq \theta < 1$
- **Problem**: given an integer N as an input find the integers $1 < N_1, N_2 < N$ such that $N = N_1 N_2$.
- Key part of the algorithm is to find the order of $x\ mod\ N$: find r such that $x^r = 1(mod\ N)$.
- Classically the running time is: $O\left(exp\left(\sqrt[3]{\frac{64}{9}n\ (\log n)^2}\right)\right)$, where $n = \log N$
- Quantum runs in $O(n^3 \log n)$ and uses $O(n^2 \log n \log\log n)$ gates.



Example 4: Solve Linear Systems of Equations

- Solving linear systems is central to a majority of science, engineering, finance and economics applications.
- Problem: given a system $A\vec{x} = \vec{b}$, find $\vec{x}$.
- Equivalent to: $A|x\rangle = |b\rangle$ with solution $|x\rangle = \frac{A^{-1}|b\rangle}{||A^{-1}|b\rangle||}$.
- Provided the linear systems is sparse and has a low condition number κ, the quantum algorithm has a runtime of $O(\log(N)\,\kappa^2)$.
- Classically, the fastest algorithm runs in $O(N\kappa)$, an exponential speedup.

Example 5: Linear Algebra

- Many linear algebra techniques can be performed exponentially faster on a quantum computer.
- Assume $N = 2^n$, so $n = \log N$.

Technique	Classically	Quantum
FFT	$O(N \log(N))$	$O(\ (\log(N))^2\ )$
Eigenvectors/eigenvalues	$O(N^3)$ to $O(N^2)$	$O(\ (\log(N))^2\ )$
Matrix inversion	$O(N^3)$ to $O(N \log(N))$	$O(\ (\log(N))^3\ )$

- These are the basic operations required for most machine learning protocols.

Example 6: Quantum Supremacy

- Google recently achieved quantum computational supremacy with a 53-qubit superconducting device.
- Circuit depths up to 20 layers.
- Classical simulations predicted to take 10,000 years.
- Quantum solution takes about 200 seconds.

Article

Quantum supremacy using a programmable superconducting processor

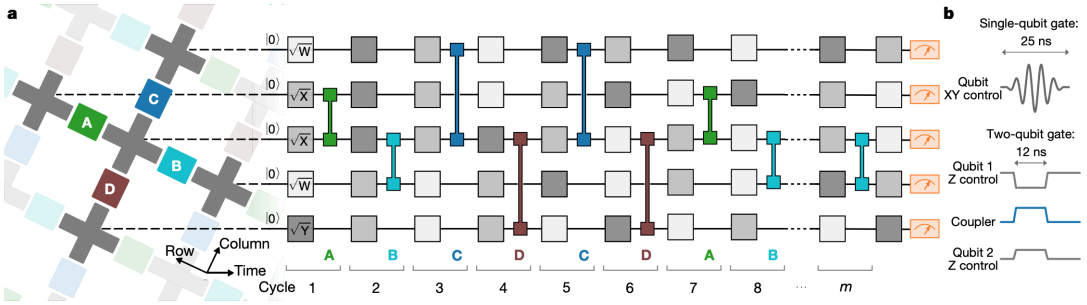
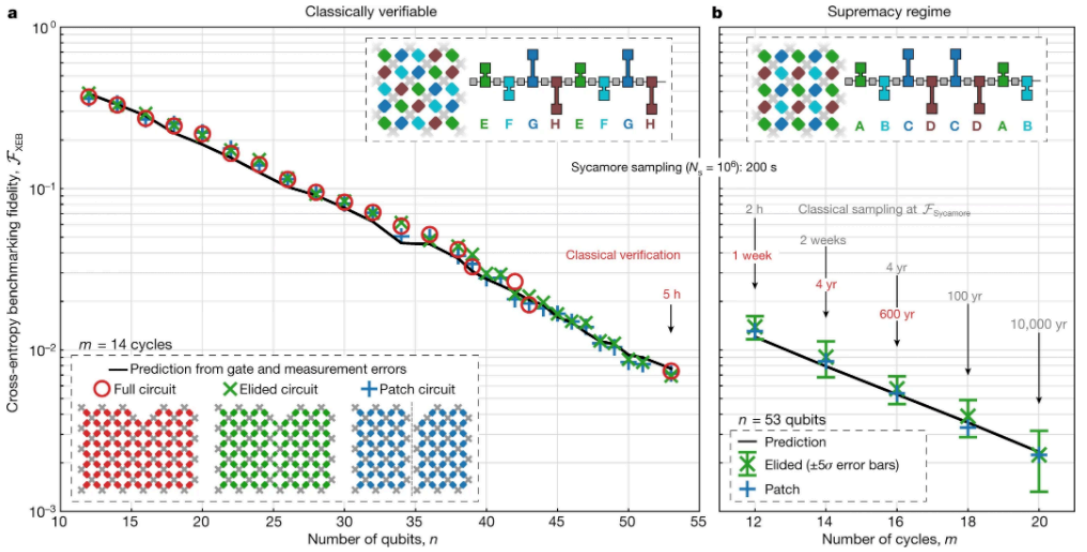
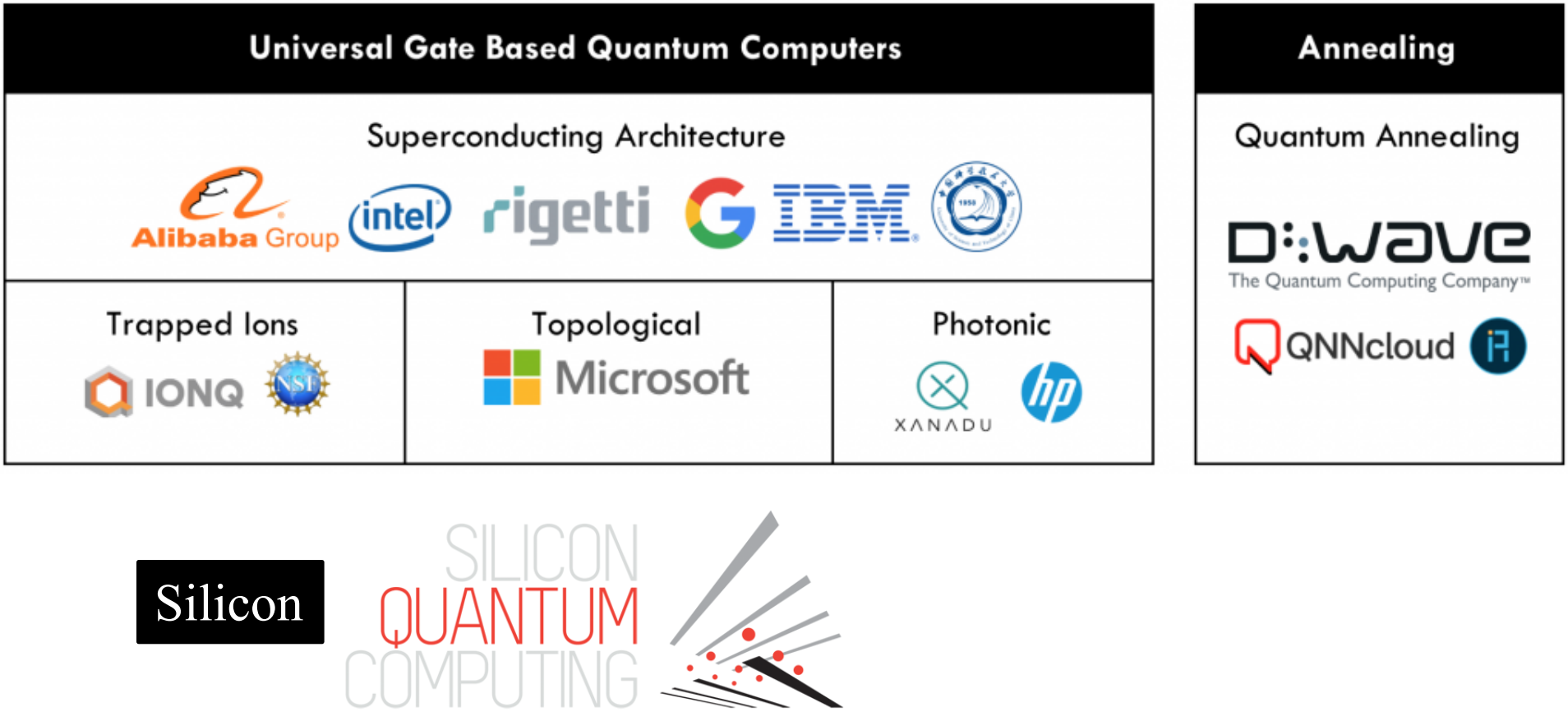


Fig. 3 | Control operations for the quantum supremacy circuits. a, Example quantum circuit instance used in our experiment. Every cycle includes a layer each of single- and two-qubit gates. The single-qubit gates are chosen randomly from $\{\sqrt{X}, \sqrt{Y}, \sqrt{W}\}$, where $W=(X+Y)/\sqrt{2}$ and gates do not repeat sequentially. The sequence of two-qubit gates is chosen according to a tiling pattern, coupling each qubit sequentially to its four nearest-neighbour qubits. The couplers are divided into four subsets (ABCD), each of which is executed simultaneously across the entire array corresponding to shaded colours. Here we show an intractable sequence (repeat ABCDCDAB); we also use different coupler subsets along with a simplifiable sequence (repeat EFGHEFGH, not shown) that can be simulated on a classical computer. **b**, Waveform of control signals for single- and two-qubit gates.



Emerging Quantum Computing Industry

- Companies building the hardware are using six main physical systems:

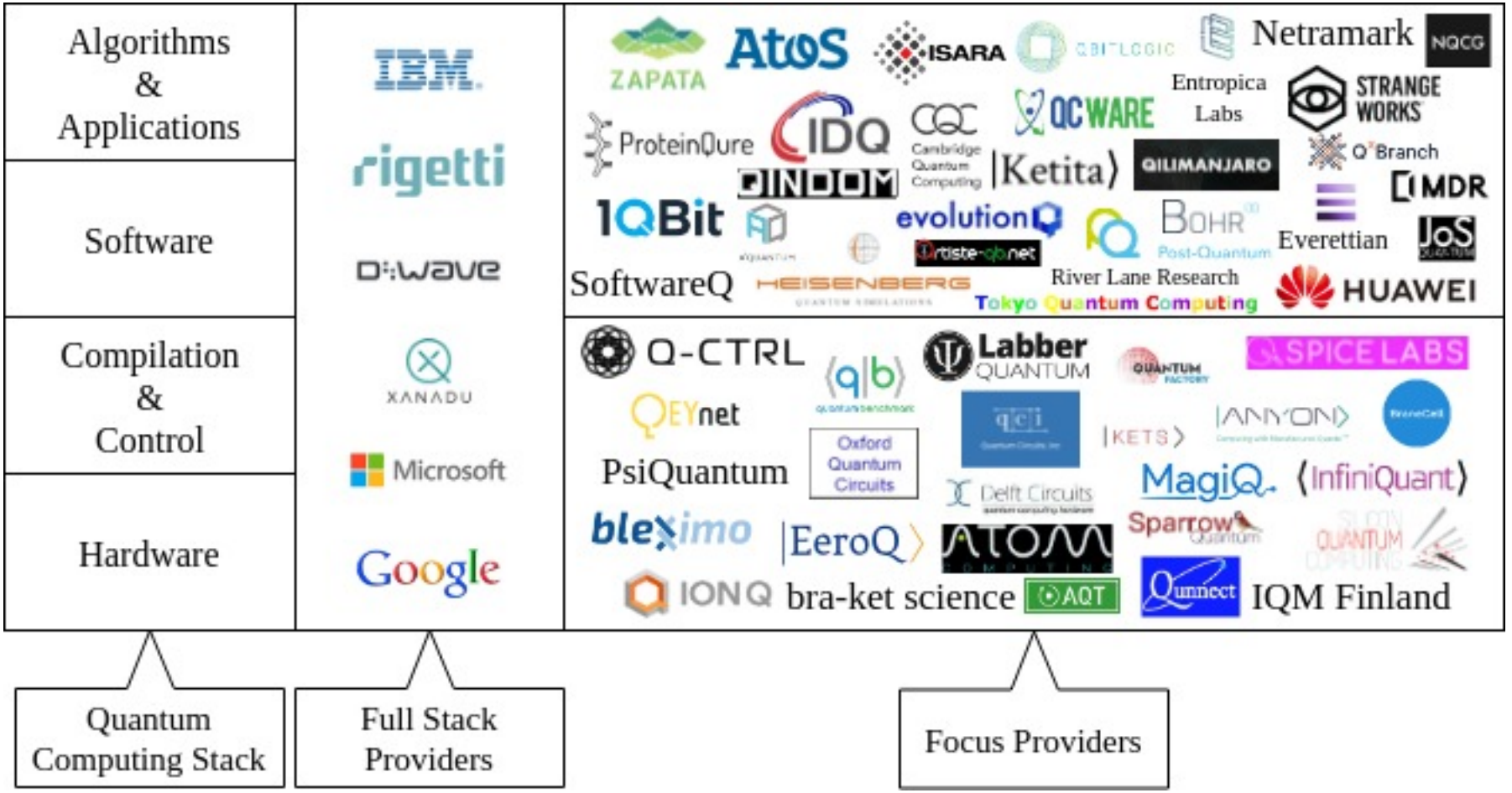


Silicon

SILICON QUANTUM COMPUTING

Emerging Quantum Computing Industry

- There are now a lot of quantum computing companies.



Coming Up Next

- Review Lecture