Quantum Computing

Ashley Montanaro
ashley@cs.bris.ac.uk

Department of Computer Science, University of Bristol
Bristol, UK

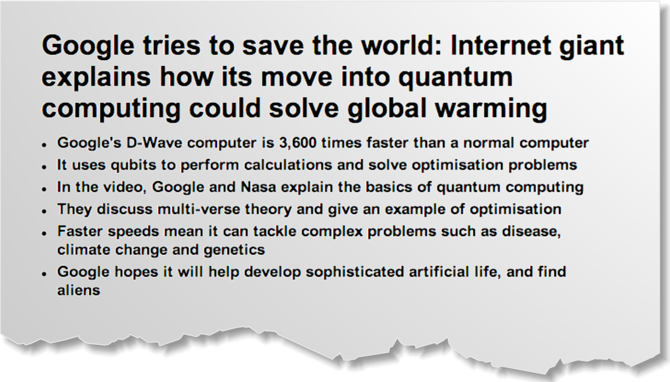
12 February 2014

Quantum computing

A quantum computer is a machine designed to use the principles of quantum mechanics to do things which are fundamentally impossible for any computer built only based on classical physics.

Quantum computing

A quantum computer is a machine designed to use the principles of quantum mechanics to do things which are fundamentally impossible for any computer built only based on classical physics.



Google tries to save the world: Internet giant explains how its move into quantum computing could solve global warming

- Google's D-Wave computer is 3,600 times faster than a normal computer
- It uses qubits to perform calculations and solve optimisation problems
- In the video, Google and Nasa explain the basics of quantum computing
- They discuss multi-verse theory and give an example of optimisation
- Faster speeds mean it can tackle complex problems such as disease, climate change and genetics
- Google hopes it will help develop sophisticated artificial life, and find aliens

Daily Mail, 15 October 2013

Ashley Montanaro
ashley@cs.bris.ac.uk

This talk

1. A brief introduction to the quantum computing model
2. Quantum algorithms: what quantum computers can do
3. Experimental implementations
4. Further reading

The quantum model: qubits

On a normal ("classical") computer, we store information as bits.

The quantum model: qubits

On a normal ("classical") computer, we store information as bits.

- A bit can be either in the state 0, or the state 1.

The quantum model: qubits

On a normal ("classical") computer, we store information as bits.

- A bit can be either in the state 0, or the state 1.
- Physically, we can store a bit in some object that has two states:

Pic: coins-of-the-uk.co.uk

The quantum model: qubits

On a normal ("classical") computer, we store information as bits.

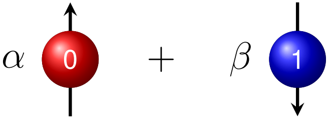
- A bit can be either in the state 0, or the state 1.
- Physically, we can store a bit in some object that has two states:

Pic: coins-of-the-uk.co.uk

- A qubit ("quantum bit") is stored in a tiny physical system like an individual atom that behaves quantum mechanically.

The quantum model: qubits

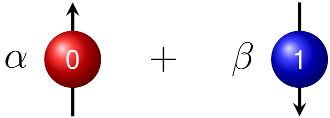
As well as being in states corresponding to 0 or 1, a qubit can be anywhere in between!



- Here α and β are any numbers (in fact, more generally complex numbers...) satisfying $\alpha^2 + \beta^2 = 1$.

The quantum model: qubits

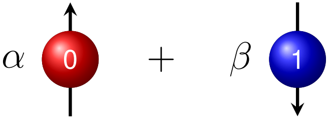
As well as being in states corresponding to 0 or 1, a qubit can be anywhere in between!



- Here α and β are any numbers (in fact, more generally complex numbers...) satisfying $\alpha^2 + \beta^2 = 1$.
- This is called superposition.

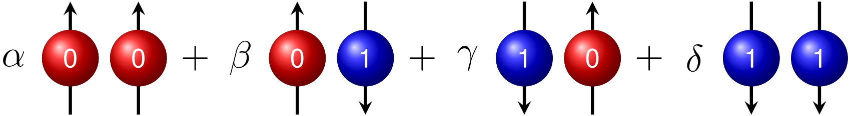
The quantum model: qubits

As well as being in states corresponding to 0 or 1, a qubit can be anywhere in between!



- Here α and β are any numbers (in fact, more generally complex numbers. . .) satisfying $\alpha^2 + \beta^2 = 1$.
- This is called superposition.

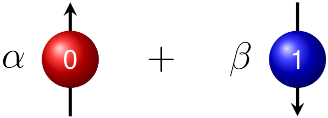
If we have n qubits, they can be in a superposition of 2^n different states:



Ashley Montanaro
ashley@cs.bris.ac.uk

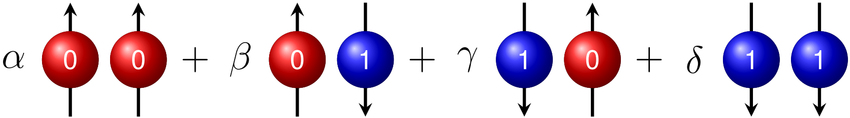
The quantum model: qubits

As well as being in states corresponding to 0 or 1, a qubit can be anywhere in between!



- Here α and β are any numbers (in fact, more generally complex numbers...) satisfying $\alpha^2 + \beta^2 = 1$.
- This is called superposition.

If we have *n* qubits, they can be in a superposition of 2^n different states:



This allows a quantum computer to run an algorithm on many possible inputs simultaneously.

Ashley Montanaro
ashley@cs.bris.ac.uk

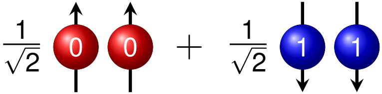
University of BRISTOL

Measurement and entanglement

- If we measure some qubits, we see each outcome with probability equal to its corresponding coefficient squared.

Measurement and entanglement

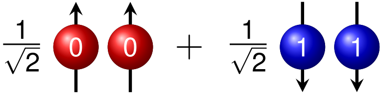
- If we measure some qubits, we see each outcome with probability equal to its corresponding coefficient squared.
- For example, imagine we have two qubits in the state



Ashley Montanaro
ashley@cs.bris.ac.uk

Measurement and entanglement

- If we measure some qubits, we see each outcome with probability equal to its corresponding coefficient squared.
- For example, imagine we have two qubits in the state

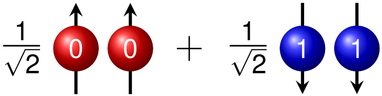


- Then if we measure the qubits, we get outcome 00 with probability $\frac{1}{2}$, and outcome 11 with probability $\frac{1}{2}$.

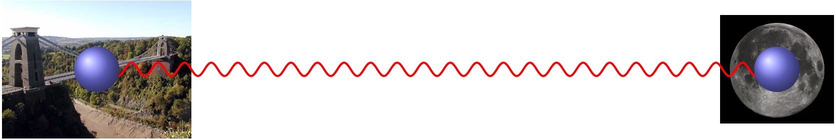
Ashley Montanaro
ashley@cs.bris.ac.uk

Measurement and entanglement

- If we measure some qubits, we see each outcome with probability equal to its corresponding coefficient squared.
- For example, imagine we have two qubits in the state

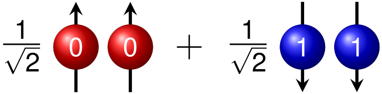


- Then if we measure the qubits, we get outcome 00 with probability $\frac{1}{2}$, and outcome 11 with probability $\frac{1}{2}$.
- But what if the first qubit is in Bristol, and the second is on the Moon?

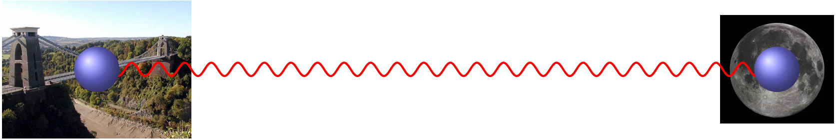


Measurement and entanglement

- If we measure some qubits, we see each outcome with probability equal to its corresponding coefficient squared.
- For example, imagine we have two qubits in the state



- Then if we measure the qubits, we get outcome 00 with probability $\frac{1}{2}$, and outcome 11 with probability $\frac{1}{2}$.
- But what if the first qubit is in Bristol, and the second is on the Moon?



- It seems that the measurement result in Bristol has instantaneously affected the qubit on the Moon...
- This bizarre phenomenon is known as quantum entanglement.

Ashley Montanaro
ashley@cs.bris.ac.uk

Shor's algorithm

Integer factorisation

Given an integer N such that $N = p \times q$ for prime numbers p and q, find p and q.

For example: given 15 as input, the output should be 3 and 5.

Shor's algorithm

Integer factorisation

Given an integer N such that $N = p \times q$ for prime numbers p and q, find p and q.

For example: given 15 as input, the output should be 3 and 5.

Shor's algorithm

- In 1994, Peter Shor described a quantum algorithm which can factorise large integers efficiently.
- No efficient classical algorithm is known for this problem.

Pic: physik.uni-graz.at

Factorisation and cryptography

Why should we care about integer factorisation?

Factorisation and cryptography

Why should we care about integer factorisation?

- The RSA cryptosystem which underlies Internet security relies on the hardness of integer factorisation.

Factorisation and cryptography

Why should we care about integer factorisation?

- The RSA cryptosystem which underlies Internet security relies on the hardness of integer factorisation.
- If we could factorise large numbers efficiently, we could break this cryptosystem.

Factorisation and cryptography

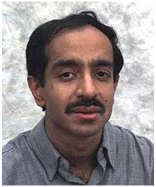
Why should we care about integer factorisation?

- The RSA cryptosystem which underlies Internet security relies on the hardness of integer factorisation.
- If we could factorise large numbers efficiently, we could break this cryptosystem.

In 2009, a 232-digit number was factorised using hundreds of computers over a period of 2 years... by comparison, a large quantum computer could factorise a number with thousands of digits in a matter of minutes.

Grover's algorithm

- One of the most basic problems in computer science is unstructured search.



Pic: Bell Labs

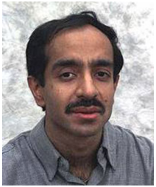
Grover's algorithm

- One of the most basic problems in computer science is unstructured search.

- Imagine we have n boxes, each containing a 0 or a 1. We can look inside a box at a cost of one query.

0	0	1	0	0	0	1	0

- We want to find a box containing a 1.



Pic: Bell Labs

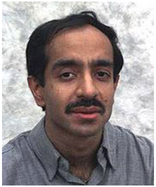
Ashley Montanaro
ashley@cs.bris.ac.uk

Grover's algorithm

- One of the most basic problems in computer science is unstructured search.
- Imagine we have n boxes, each containing a 0 or a 1. We can look inside a box at a cost of one query.

0	0	1	0	0	0	1	0

- We want to find a box containing a 1.
- On a classical computer, this task could require n queries in the worst case.



Pic: Bell Labs

Ashley Montanaro
ashley@cs.bris.ac.uk

Grover's algorithm

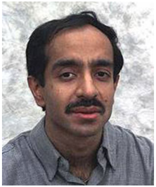
- One of the most basic problems in computer science is unstructured search.

- Imagine we have n boxes, each containing a 0 or a 1. We can look inside a box at a cost of one query.

0	0	1	0	0	0	1	0

- We want to find a box containing a 1.

- On a classical computer, this task could require n queries in the worst case. But on a quantum computer, Grover's algorithm can solve the problem with roughly $\sqrt{n}$ queries.



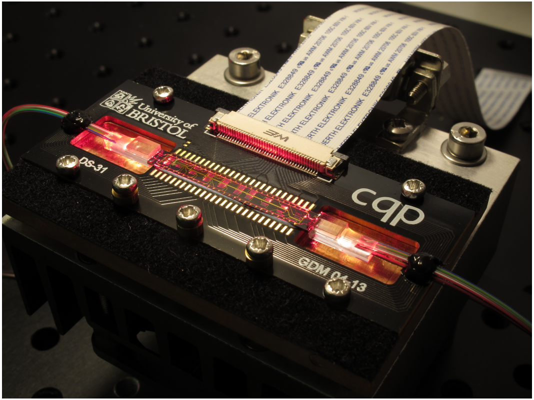
Pic: Bell Labs

Experimental implementations

There are a number of different technologies which could be used to implement a quantum computer.

Experimental implementations

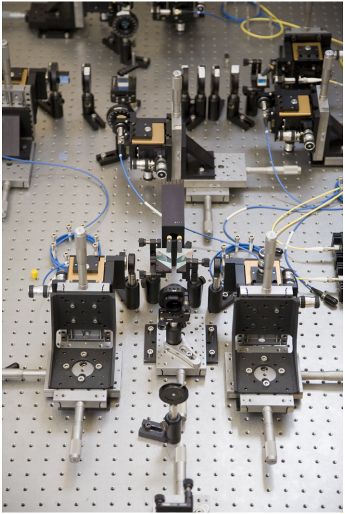
There are a number of different technologies which could be used to implement a quantum computer.



Photonic quantum circuits on silicon (University of Bristol)

Pic: University of Bristol

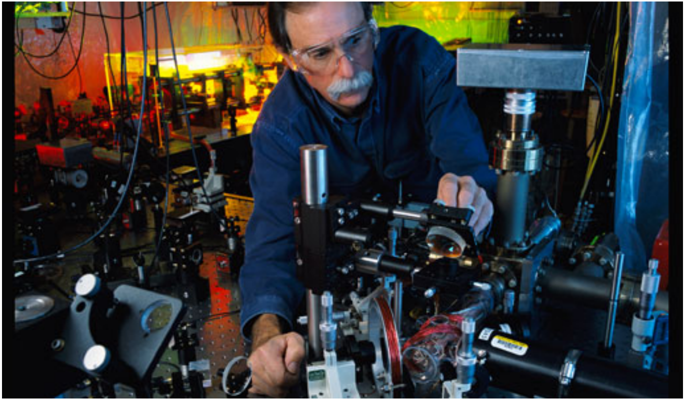
Experimental implementations



"Bulk" optics (University of Bristol)

Pic: Carmel King

Experimental implementations



Ion trap (David Wineland group, NIST)

Pic: nobelprize.org

Quantum computing FAQs

1. When can I have one?

Quantum computing FAQs

1. When can I have one?
2. Will I have one on my desk?

Quantum computing FAQs

1. When can I have one?
2. Will I have one on my desk?
3. Can they help discover aliens?

Quantum computing FAQs

1. When can I have one?
2. Will I have one on my desk?
3. Can they help discover aliens?

To summarise:

- Quantum computing is a new and exciting model of computation which can do things that classical computing simply cannot.
- A massive international effort is ongoing to build a large-scale quantum computer, including here at Bristol.
- There are still many fascinating open problems to address.

Further reading

- Winning a Game Show with a Quantum Computer
  Ashley Montanaro
  `http://www.cs.bris.ac.uk/~montanar/gameshow.pdf`
- Quantum Computing Since Democritus
  Scott Aaronson
  `http://www.scottaaronson.com/democritus/`
- Introduction to Quantum Computing, University of Waterloo
  John Watrous
  `https://cs.uwaterloo.ca/~watrous/LectureNotes.html`

Partial timeline: Theory of quantum computing

	⋮
1984	Quantum cryptographic key distribution invented [Bennett+Brassard]
1985	General quantum computational model proposed [Deutsch]
1992	First exponential quantum speed-up discovered [Deutsch and Jozsa]
1993	Quantum teleportation invented [Bennett et al.]
1994	Shor’s algorithm rewrites the rulebook of classical cryptography
1995	Quantum error-correcting codes invented [Shor]
1996	Quantum simulation algorithm proposed [Lloyd]
1996	Quantum speed-up for unstructured search problems [Grover]
1998	Efficient quantum communication protocols [Buhrman et al.]
2003	Exponential speed-ups by quantum walks invented [Childs et al.]
	⋮

Partial timeline: Quantum computing experiments

	⋮
1997-8	Quantum teleportation demonstrated [Innsbruck, Rome, Caltech, . . .]
1998	Quantum error-correction demonstrated [MIT]
2001	Shor's algorithm factorises $15 = 3 \times 5$ using NMR [IBM]
2005	8 qubits controlled in ion trap [Innsbruck]
2008	Photonic waveguide quantum circuits demonstrated [Bristol]
2010	Entangled states of 14 qubits created in ion trap [Innsbruck]
2012	$21 = 3 \times 7$ factorised using quantum optics [Bristol]
2012	100μs coherence for superconducting electronic qubits [IBM]
2013	First publicly-accessible "quantum cloud" [Bristol]
	⋮