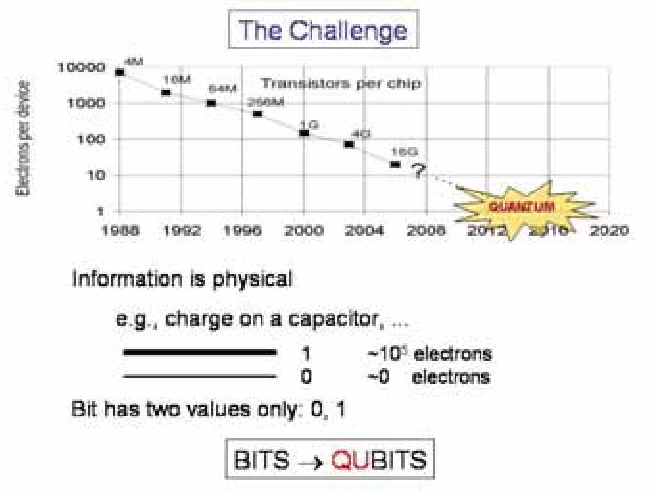
Quantum Computation

By Samuel L. Braunstein
18 April 2005

It’s a pleasure to be here. I’d like to thank the organizers, especially Susan Stepney, and Stephen Emmott Microsoft.

As you all know, this workshop is about “Grand challenges in non-classical computation”, or so it is entitled. In fact, it seems this title already gives the solution away. My own take on this is that we are facing a very clear and imminent grand challenge in computation (or computer science), and that is to build better, faster, more powerful computers to meet our ever-increasing computation and information processing needs. Classical computers have been an immense success. The problem is that the classical notions of computing, as we know them, are going to hit a brick wall.



The Challenge

Information is physical

e.g., charge on a capacitor, ...

	1	~10^5 electrons
	0	~0 electrons

Bit has two values only: 0, 1

BITS → QUBITS

Moore’s law [40 years old this month, April 2005] is telling us that computers are miniaturizing and computer power is increasing.

The expectation is that Moore’s law will break down when components reach the atomic or quantum scale.

The challenge is therefore to break this barrier, using new technologies, leading to new paradigms for computation, perhaps even an altogether new computer science (not to replace the existing one, but to complement and enrich it). And I’m looking forward to hearing all about the different promising approaches to tackle this challenge over the coming two days. In this talk, I’d like to give you a flavor for one of these approaches – the quantum approach, which in some sense is a natural reply to Moore.

The fundamental basis of Moore's law is the realization that ultimately information is represented by physical systems. Let's look at a simple example, say for storage of a single bit we might think of a capacitor:

- charged with many electrons, is a 1
- discharged is a 0.

Or for processing information we might think of the electronic current involved in the switching of a transistor.

Clearly Moore's law, in its present form (i.e., miniaturization in electronics), will break down when a capacitor is so small it can only accommodate a single electron. This problem is occupying an enormous number of engineers, physicists and computer scientists and in particular, there's a growing community thinking hard about various possibilities of non-standard computation. Specifically quantum computing is saying "let's face it, in a few years time we will have the technology to manipulate matter at the atomic scale and rather than letting Moore's stop us, let's find out whether the laws of quantum mechanics can actually be harnessed for powerful information processing."

OK, now we're at the atomic scale, instead of bits, our new quantum computers will use quantum bits or qubits. Simple? Well, not so simple.

This transition doesn't simply involve taking traditional computer science and applying it to these microscopic systems, because that doesn't work. We need to understand how these systems can be harnessed to do computation.

And that's exactly what's being done. Indeed, quantum information science will offer new computational paradigms, new hardware, new machine language, new algorithms …

In particular, we are now developing a new machine language based on quantum rather than ordinary logic, and developing new algorithms and that involves both their design and assessing their performance.

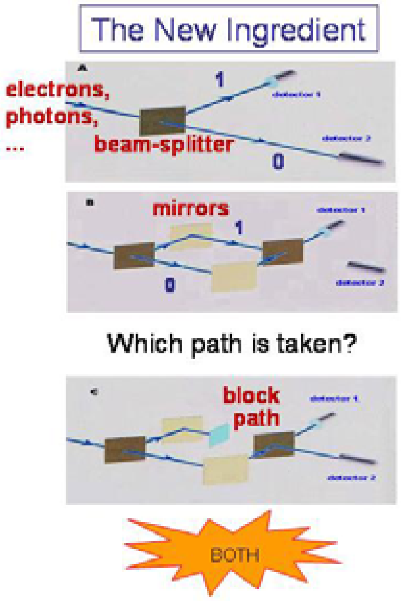
Because we are at the quantum level there are new sources of noise, subject to the laws of quantum mechanics, so we need new error correction protocols – the old ones don't work.

And we may need to start from scratch in understanding complexity classes.

And just as hard if not harder – we need to actually build these machines, and are currently exploring different hardware platforms for quantum devices.

But of course, these are still early days. After all, quantum information is a young discipline. The physicist Richard Feynman was the first to seriously think about it in the 1980's. The first substantial breakthrough came in 1993 with the discovery of a quantum algorithm for factoring. Which generated a lot of excitement – but more than excitement, it generated a lot of scepticism and a lot of that scepticism was justified.

A decade and more into the field and there are still very difficult questions and problems (a few of which I'll mention today), but enough fundamental problems have been solved, so that today we know that Quantum Information is here to stay and quantum computing will become a reality sooner or later.


The New Ingredient

Which path is taken?

But before we can talk about all these things, I have to take a step back, to explain what I meant by replacing bits by qubits, this constitutes the essential transition to the language of quantum information processing.

Suppose my bit of information was encoded in the activation of a pair of detectors: one represents a 1, the other a 0.

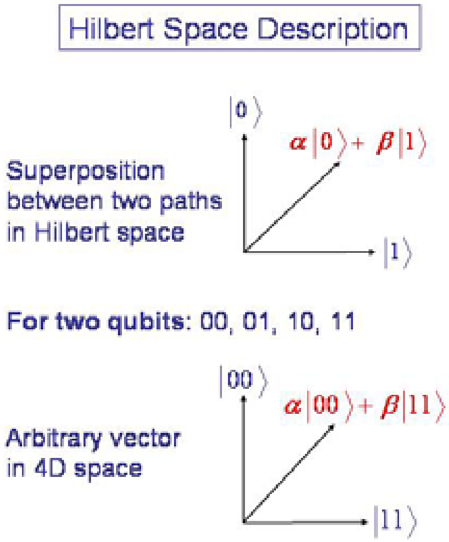
If I shoot a beam of particles (e.g., electrons, photons, even bucky balls) at detector 1, I get a string of ones, at detector 2, I get a string of zeros.

I could introduce a beamsplitter (e.g., coated glass) which splits the beam, with 50% going to detector 1 and 50% to detector 2 in a completely random fashion.

Each particle randomly takes one path or the other. Now we perform an interesting experiment and add a second beamsplitter, which unites the two beams before they reach the detectors. It so happens that we can align this beamsplitter so that all the particles go to detector 1 and none to detector 2.

So now the output from detector 1 no longer tells us which path each particle took, so we want to check which particle took which path (representing a 0 bit or a 1 bit). It turns out that in quantum mechanics we can't do that.

No experiment we try can give us that information. For example, if we block one path we might expect to *only* see a reduction in intensity to detector 1, in fact, we start seeing intensity at detector 2 as well. The only consistent explanation is that each particle is taking *both* paths.


Hilbert Space Description
$|0\rangle$
$\alpha|0\rangle + \beta|1\rangle$
Superposition
between two paths
in Hilbert space
$|1\rangle$
For two qubits: 00, 01, 10, 11
$|00\rangle$
$\alpha|00\rangle + \beta|11\rangle$
Arbitrary vector
in 4D space
$|11\rangle$

The language we use is that the system is in a superposition of representing a zero *and* representing a one.

Whereas a bit is a zero *or* a one, a qubit (a quantum bit) can simultaneously represent any combination of a zero and a one (so it’s not just 50/50 but 40/60 or any combination).

So if we have two qubits we can have a superposition of four states (though I’ve only drawn two here).

So one qubit can represent 2 possibilities simultaneously, and k qubits can represent 2^k possibilities simultaneously. This ability to represent so much information with so few components, this explosion in information, is incredibly promising news.

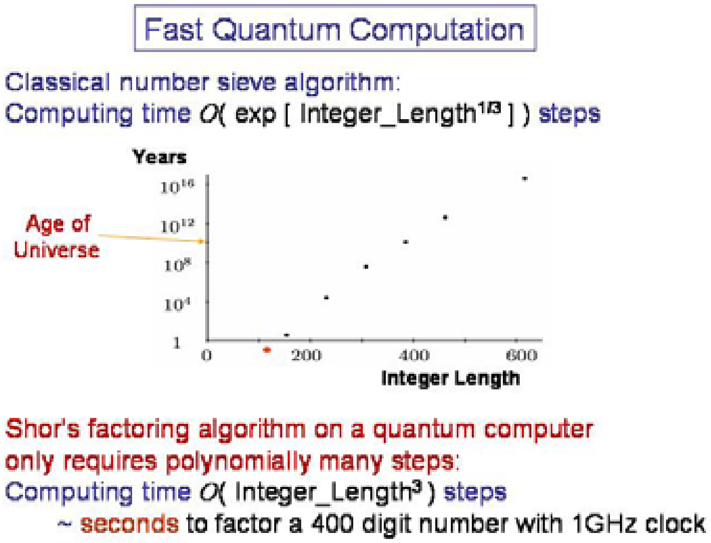
We can think of k particles taking 2^k paths with only polynomial resources – this exponential parallelism can lead to exponential speedup if only we have the right problem and the right algorithm.

Here’s the problem:

In 1903, F.N. Cole got up before an audience of the AMS and without a word wrote down the equation
$2^{67} – 1 = 147573952589676412927 = 193707721 \times 761838257287$
multiplying out the right-hand side to confirm the result. Finding the factors had taken him “three years of Sundays”. He got a standing ovation.

A little over 90 years later, a worldwide consortium of more than 1600 computers (and two FAX machines!) running for eight months solved what was then perhaps the most famous cryptography challenge then in existence: Factoring RSA-129

The alternative quantum algorithm is the famous Shor algorithm. And indeed, with the advent of quantum computers, we can now factor the number 15! So obviously quantum computers are already making their impact!



Fast Quantum Computation

Classical number sieve algorithm:
Computing time O(exp [Integer_Length$^{1/3}$]) steps

Shor's factoring algorithm on a quantum computer only requires polynomially many steps:
Computing time O(Integer_Length^{3}) steps
~ seconds to factor a 400 digit number with 1GHz clock

The problem with the best known classical algorithm for factoring is that it is exponentially slow. So that factoring a 400 digit number would take as long as the age of the universe, even with immense parallelism.

But on a quantum computer (obviously one somewhat larger than that used to factor the number 15) this would potentially take just seconds.

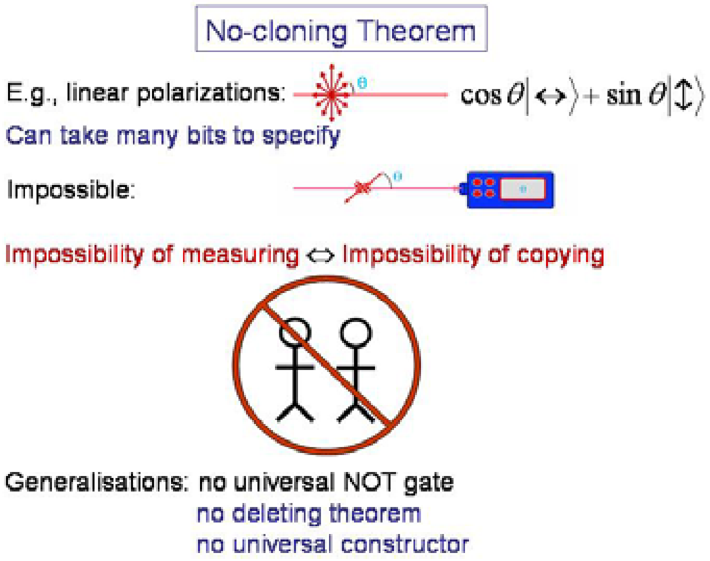
(But at the moment, such a full-scale quantum computer does not yet exist. So banks and the rest of us can certainly rely on RSA for a little longer.)

Here’s how it works … [details, details, details]

So Shor’s algorithm is an example of a problem that can be solved in polynomial time with a bounded probability of error on a quantum computer. This adds a new complexity class BQP to the polynomial hierarchy and naturally brings a whole set of questions, where to draw the line and how to classify specific algorithms.

This was factoring, but quantum computing isn’t just factoring. There are a growing number of algorithms and protocols that rely on quantum information processing, all of them bare some advantages, security in cryptography, secret sharing or data hiding, however, it’s important to note that not all of them deliver a speedup.

In the time that remains, I’d like to mention some of these other forms of quantum information processing that have nothing to do with speedup, but nonetheless are very important for quantum computing, and hopefully by describing how they work I can capture a few more of the truly bizarre properties of quantum information.



No-cloning Theorem

E.g., linear polarizations: $\cos\theta|\leftrightarrow\rangle+\sin\theta|\updownarrow\rangle$

Can take many bits to specify

Impossible:

Impossibility of measuring ⇔ Impossibility of copying

Generalisations: no universal NOT gate
no deleting theorem
no universal constructor

We’ve already encountered the fact, in quantum mechanics, that you can’t make measurements without destroying something about the system (as we’ve seen in the two-path experiment).

A surprising consequence of this is that you can’t make copies of quantum information, since then you could make many copies and make an ideal measurement leaving the original intact!

The implications for algorithm design can hardly be overstated – we use copying all the time!

In addition, there are related results generalizing no-cloning and those include:

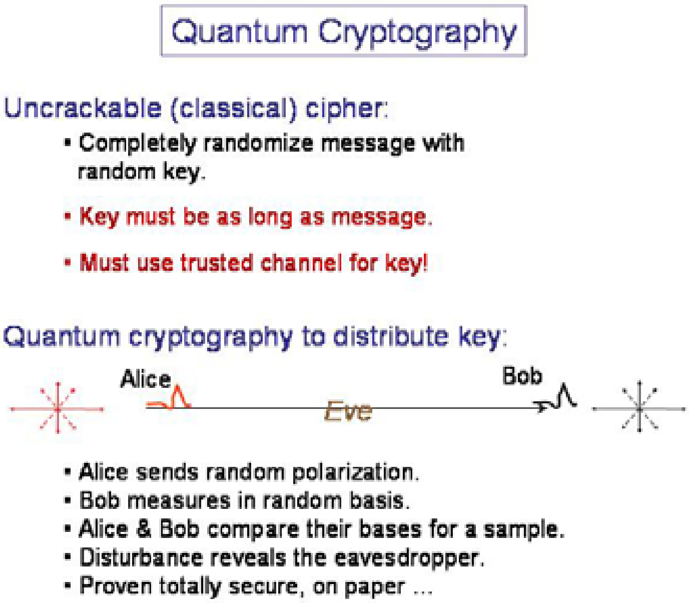
- no universal NOT gate
- the converse of cloning – the no-deleting theorem
- and no universal constructor can be designed or built, so von Neuman’s idea of self-replicating automata cannot be extended to the quantum level.

So all that might sound like really bad news, but instead the approach of quantum computing is to adopt the old software adage “it’s not a bug, it’s a feature.”

Quantum cryptography relies on just this bug, *er* feature.

We already have a perfectly good uncrackable classical cipher, it’s the Vernam cipher or one-time pad. The trouble is that we have to securely transfer the random key. Classically, there’s no guaranteed way to do it.

But the fragility of quantum information provides a way.



Quantum Cryptography

Uncrackable (classical) cipher:

- Completely randomize message with random key.
- Key must be as long as message.
- Must use trusted channel for key!

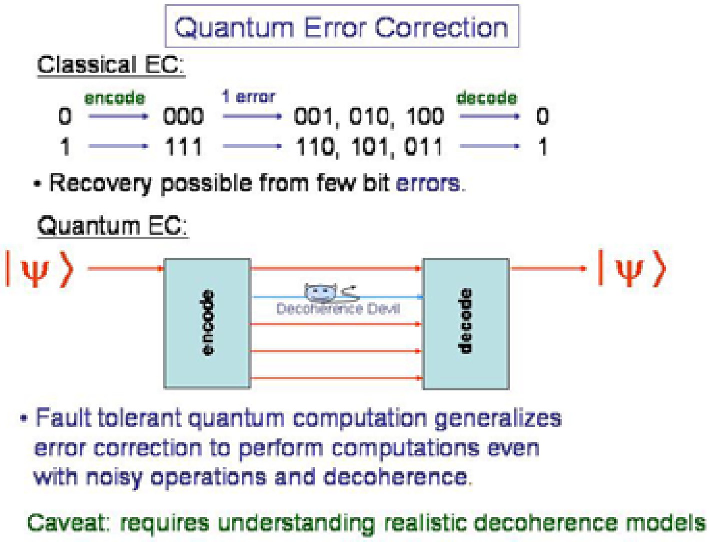
Quantum cryptography to distribute key:

- Alice sends random polarization.
- Bob measures in random basis.
- Alice & Bob compare their bases for a sample.
- Disturbance reveals the eavesdropper.
- Proven totally secure, on paper ...

Alice can send qubits from which Bob can extract some information – enough for them to construct a correlated random key. If, however, Eve comes along and tries to extract information from the quantum channel she in turn will disturb the information being sent and Alice and Bob can discover her and resend.

Indeed, quantum cryptography hardware is already being sold by two companies.

So quantum cryptography is an obvious benefactor of no-cloning.



Quantum Error Correction

Classical EC:

$$0 \xrightarrow{\text{encode}} 000 \xrightarrow{\text{1 error}} 001,\ 010,\ 100 \xrightarrow{\text{decode}} 0$$
$$1 \longrightarrow 111 \longrightarrow 110,\ 101,\ 011 \longrightarrow 1$$

- Recovery possible from few bit errors.

Quantum EC:

- Fault tolerant quantum computation generalizes error correction to perform computations even with noisy operations and decoherence.

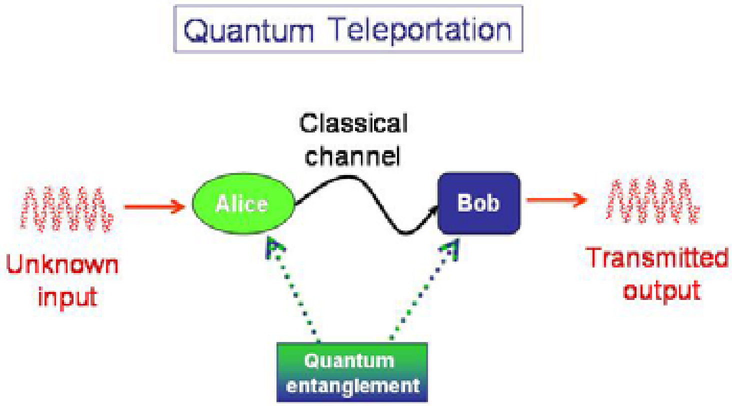
Caveat: requires understanding realistic decoherence models

In contrast, quantum error correction seemed for a long time as the Achilles heel of the entire field.

All physical processes involve noise, it was quite clear that without robust fault-tolerant error correction there was no future for the field.

A simple redundancy code requires copying to make the code, and measurement to decode it. It looks bad!

Fortunately, it turns out that although we can't clone or measure without destroying information we can nonetheless make this work – an alternative way of thinking about error correction is that the information is encoded on multi-bit correlations – this generalizes nicely to qubits.



Yet another apparent consequence of no-cloning is the impossibility of communicating quantum information through classical channels (since the classical channel can be copied, many Bob's could then each have a copy of the original state).

Which is not possible by the no-cloning theorem.

However, if we share a special quantum resource, which we call entanglement, between Alice and Bob this process becomes possible. Now there's something singling out this Bob and this Alice from anybody else: So now, copying the classical channel can't complete the process of quantum information transfer.

This protocol is given the amusing name of Quantum Teleportation.

A pair of systems will be in an entangled state whenever it can't be written as a product.

There's a nice metaphor invented by Charlie Bennett of IBM of how to picture such entangled states in terms of a pair of perfect lovers. Alice can tell you what Bob would answer to any question – even on subjects they've never heard of before, and even when they are separated over very large distances.

You can take this metaphor further: In fact, it turns out that entanglement is monogamous: if Alice is entangled with Bob she can’t be entangled with Charlie; if she becomes entangled with Charlie she loses her entanglement with Bob.

This resource makes quantum teleportation possible, and in many ways teleportation is a great name. It’s even weirder than its science fiction’s name sake!!

Quantum teleportation was first demonstrated in 1998 when photons and beams of light were teleported in a lab about a distance of one meter in a number of different experiments. This made a big impact in the field and was recognised by the journal Science as one of the top 10 scientific achievements (in all fields) of the year.

Since then, the states of atoms have been teleported very short distances, but light has now been teleported even tens of kilometres.

Before I finish, I would like to come back to the point that information is physical. In quantum communication, where progress in theory and experiment are going hand-in-hand the physical representations are straightforward. But for quantum computation in this talk I’ve concentrated on the theory, but to end I’d like to just give you a flavor of the types of implementations that are being actively pursued.

The key problem with building a quantum computer is that there is a fundamental tension in the design. We would like the qubits to strongly interact with each other for conditional operations, but yet be very well isolated from everything else. There are many proposals and prototype devices with 0, 1 and up to 7 qubits.

First were ion-traps, high-temp NMR then shot ahead and even factored 15 for us, but has severe scaling problems. Some of the approaches based on semiconductors, which naively seem most promising, are relying on manipulating matter at the atomic scale and the technology is not quite there yet.

There’s actually value in this diversity, since each one is teaching us new lessons in state preparation, readout or qubit coupling. At least for now I think it’s worth preserving that diversity.

We have no idea which of these implementations (if any of those listed) will before the hardware platform of choice. But the fundamental obstacles have been overcome – at least theoretically – it’s now down to beating on the technology.

The exciting questions remain: how will this fit into computer science and with what scope? How much computational power can we get out of these systems? What are the fundamental limitations to quantum information processing?

These are the questions and these are the challenges: How far can we push this envelope.

Thank you.