Quantum Computing:
What is it about?

James Palmer
PHYS 420
Fall 2008

What *is* Quantum Computing about?

- The ideas for QC arose when people began to consider the extreme miniaturization of computer components. (Current transistors are as small as tens of nanometers.)
- R.P. Feynman first considered the idea, and David Deutsch wrote an important paper about it in 1985 (*Quantum Theory, The Church-Turing Principle, and the universal quantum computer)*
- Quantum computing uses the superposition of states to perform operations on many bits at once.
- It is useful for simulating quantum systems, since the quantum computer itself is a quantum system.
- It can do anything a classical computer can do! Knowledge of classical computing concepts lends to an even greater understanding of Q.C.

Compare to classical computing:

Classical Computing	Quantum Computing
·Two representations (binary) • Either 0 or 1 ·Irreversible operations (Increase in entropy. Lost information is given up as heat) • AND Gate: Two inputs, one output. All inputs can't be recovered. ·Serial operations on bits	·Multiple representations (the qubit) ·Reversible operations (No change in entropy) • Toffoli gate: Two inputs, two outputs. All inputs recoverable. ·Parallel operations on qubits by inputing a superposition of qubits to a gate

Representation of Information

What is a qubit?

It is the linear combination of two quantum states:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$$

More than one qubit may be set up a time. (Same as the classical case.) Each qubit is just an independent state. One bit is never enough!

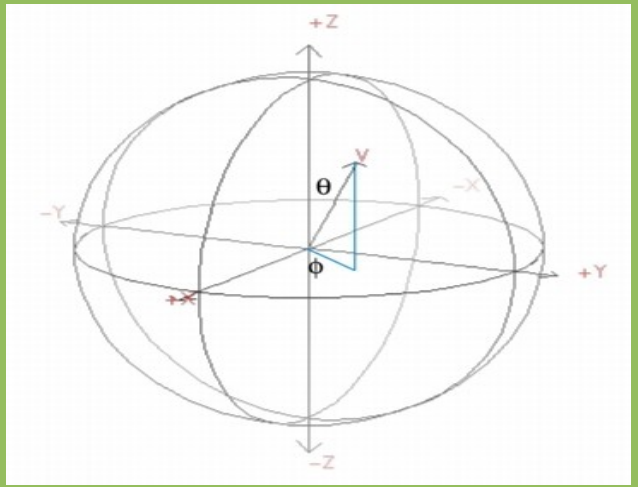
Two qubits:

$$|\psi\rangle = \left[a|0\rangle + b|1\rangle\right]_x \left[c|0\rangle - d|1\rangle\right]_y = ac|0\rangle_x|0\rangle_y + ad|0\rangle_x|1\rangle_y + bc|1\rangle_x|0\rangle_y + bd|1\rangle_x|1\rangle_y$$

In general, the constants are complex.

Any two-level quantum system (or approximately two-level) can realize a qubit. This includes atoms!

The image on the right is a phase space representation of the qubit vector.



Deutsch's Problem: Quantum Parallelism

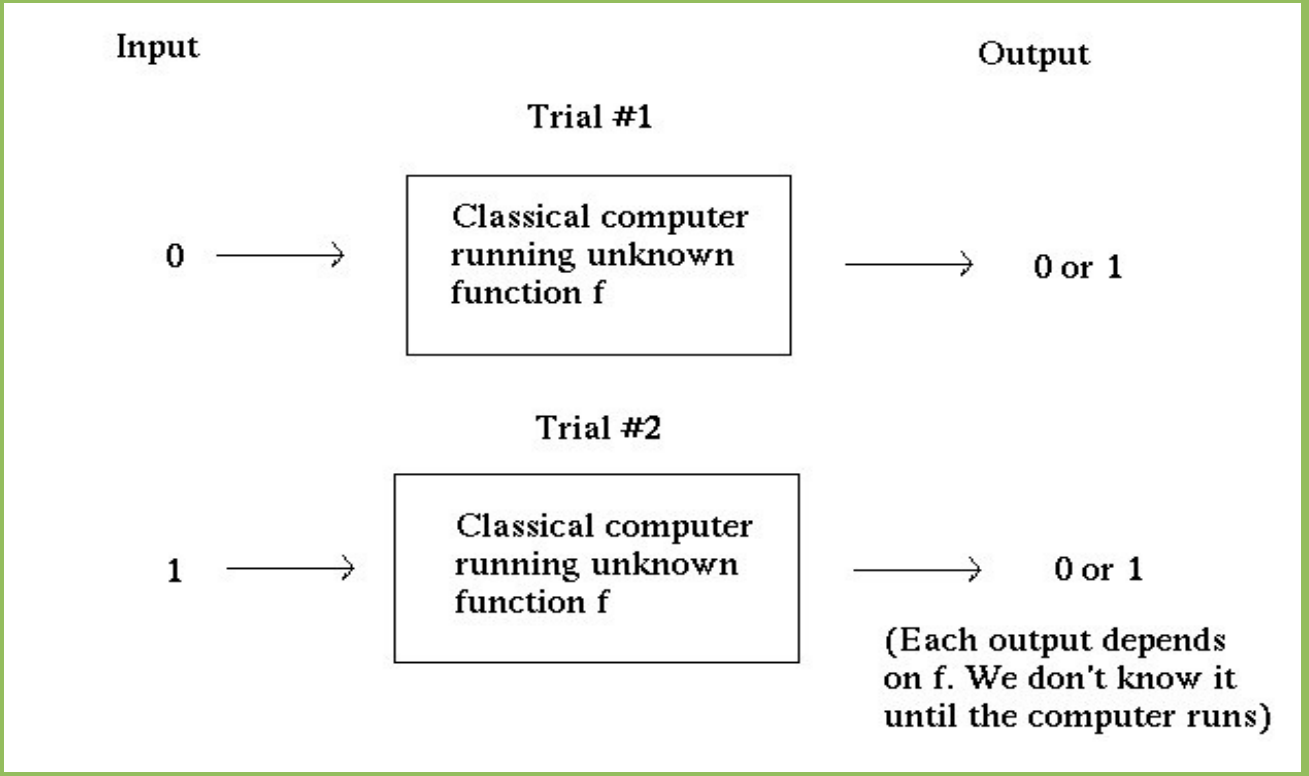
This is one of the simplest quantum algorithms to understand.

We want to know the function *f(x)* acting on one bit. Is it constant (*f(x)* = one value.) or balanced (Each input has a single output.) ?

f(x) has four different possibilities:

f(0)	1	0	0	1
f(1)	1	0	1	0
	Constant		Balanced	

Classical computing requires at least two runs. Run twice, with either input. Is the output different in both trials?



Quantum computing, however, can do it in just **one** run. How?
We could try either $|\Psi\rangle = |0\rangle, |\Psi\rangle = |1\rangle$ but that still requires two runs.

Since f can act on a superposition, let's try a special combination of states, using two qubits, one in x space, one in y space.

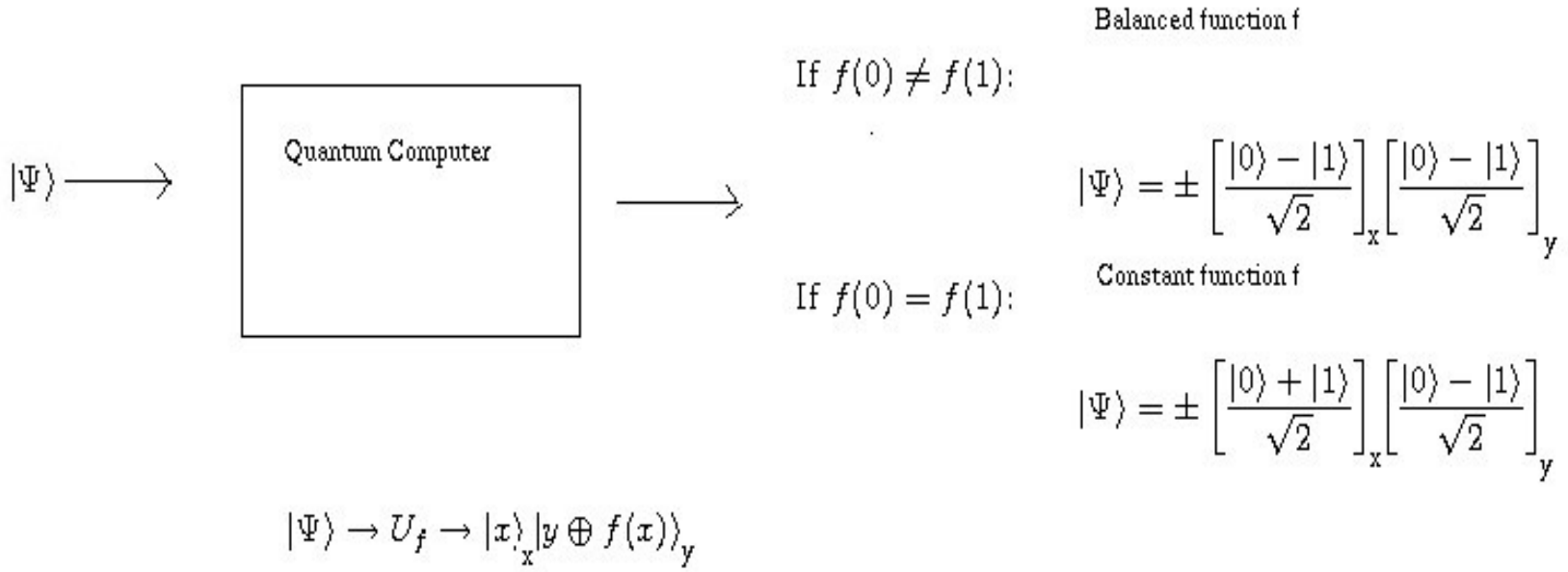
$$|\Psi\rangle = \left[\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right]_x \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right]_y = \frac{1}{2}\left[|0\rangle_x|0\rangle_y - |0\rangle_x|1\rangle_y + |1\rangle_x|0\rangle_y - |1\rangle_x|1\rangle_y\right]$$

Acting transformation U on the state, which takes the exclusive OR (XOR) operation of the y qubit and f acting on x:

y	*f(x)*	*y* XOR *f(x)*
0	0	0
0	1	1
1	0	1
1	1	0

Why do this?

Trial #1: The *only* trial!



The phase between 0 and 1 is different for f(x) being a constant (positive) or being balanced (negative). We learned this by running the program only once!

We see that the output is definitely different for either form of f(x). The relative phase in the first qubit is different for either situation. That means that the computer probes both possible inputs (0 or 1) to the function at the same time! This mode of parallel computing can be extended to other (more useful) problems.

Other Quantum Algorithms

- Shor's Algorithm
  - Used to factor large numbers. It's much quicker than classical methods. Problems that would take a classical computer longer than the age of the universe to solve can be done in perhaps a few years by a quantum computer.
  - Implications for security
- Grover's Algorithm
  - Fast database search. Already implemented on a small scale.
  - Put the bits to be searched in a superposition. Conditionally rotate each one except the solution. Measure to find solution.

All Quantum Algorithms can only be run through one iteration, unlike classical systems which allow many interations. (No “for” loops, in the parlance of programmers)

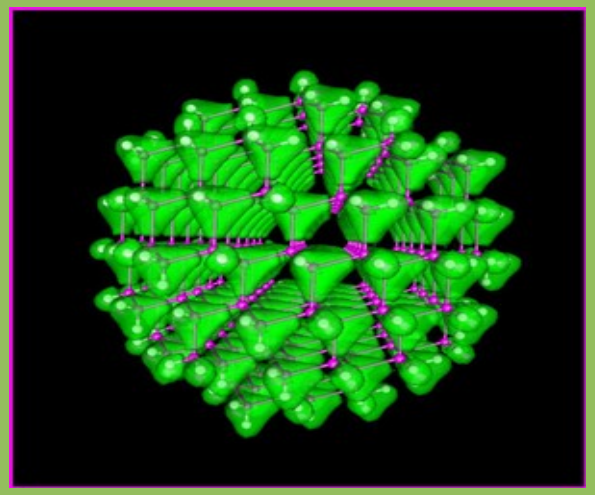
Physical Realization

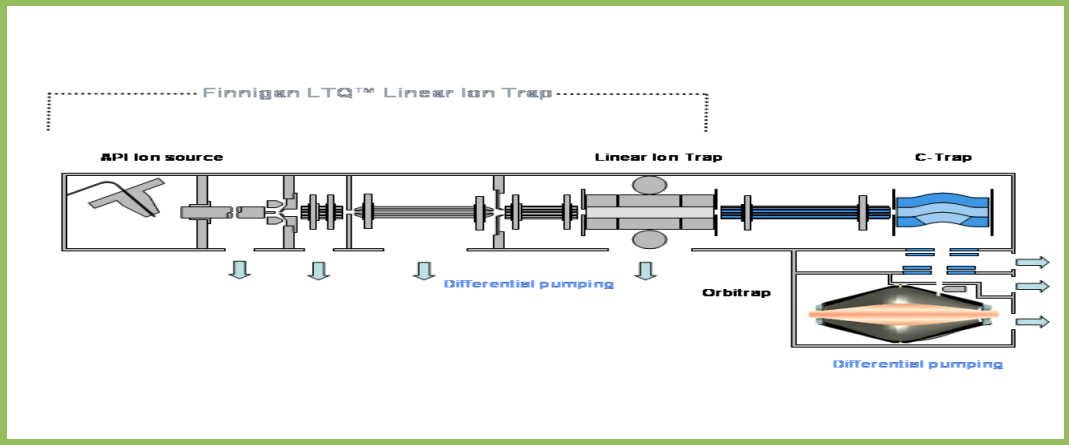
·Any two-level system could do it, in theory
·Nuclear Magnetic Resonance provides a venue
- Qubits are stored in the orientation of the nuclear spins
- Operations could be such a thing as a pulse of radio waves.

·Ion traps: Trap ions in electric and magnetic fields. Use of lasers to do operations by exciting ions.

·Polarization of Photons: Mainly difficult to interact with the photon

·Quantum Dots: An electron trapped by an array of atoms.
- Operations through laser exciting the electron
- Half the transition time puts it in a superposition of states.





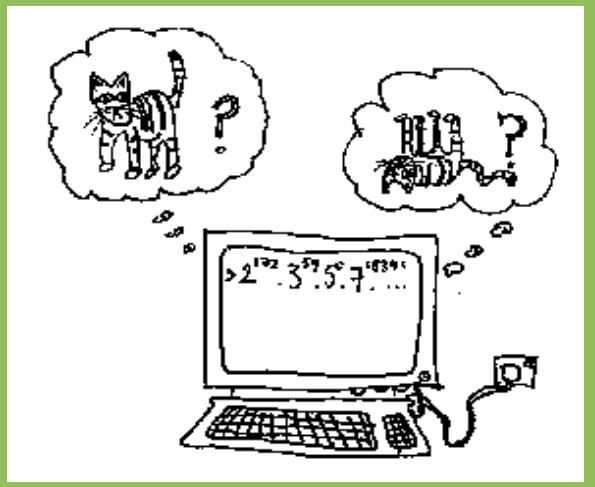
Difficulties

- Current systems can at most have several qubits. (For example, NMR can handle around 10 qubits at most)

- Interactions with the environment lead to *decoherence*, whereby the system is no longer in a well defined state. Again, this limits the number of qubits current technology can handle.

- The lifetime of a system's excited state is often too short to be of much use. (Quantum dots stay excited for about a microsecond.)

References:



- *The Temple of Quantum Computing*, By Riley Perry
- http://www.cs.caltech.edu/~westside/quantum-intro.html
- http://www.theory.caltech.edu/~preskill/ph229/
- **http://alumni.imsa.edu/~matth/quant/473/473proj/node9.html**
- **http://alumni.imsa.edu/~matth/**
- **http://www.doc.ic.ac.uk/~nd/surprise_97/journal/vol4/spb3/**