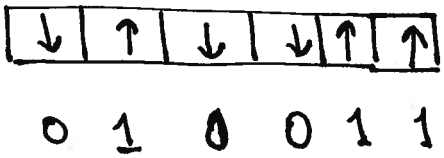
March 11

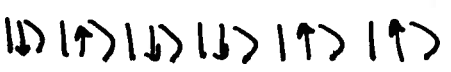
Quantum Computing

In this lecture, I would like to describe a relatively new and quite wonderful application of quantum coherence. This is the idea that quantum coherence can lead to new computer algorithms of higher efficiency.

In a realistic computer, we encode numbers as strings of 0's and 1's represented as the orientation of magnetic domains



In your laptop, these domains are small but nevertheless macroscopic. The macroscopic size of the domain allows the orientation to be read in and out with high fidelity. Still, the domains used for magnetic storage are constantly being made smaller. In the limit, you might imagine that we could encode 0's and 1's in single spins or in single two-level systems of other kinds. Then an N-digit binary value would be represented by a quantum state



For this lecture, I will refer to $|\downarrow\rangle$ and $|\uparrow\rangle$ as $|0\rangle$ and $|1\rangle$, respectively. This encoding of 0 or 1 as a quantum state is called a *qubit*.

In the 1980's, Charles Bennett, Martin Deutsch, and others investigated the special properties of algorithms that preserve the coherence of the quantum state without collapsing it. This field received a tremendous boost in 1994, when Peter Shor discovered that an algorithm that exploits quantum coherence can solve the problem of factoring large number in polynomial time. This factorization problem is a famous example of a problem requiring exponentially long times on a classical computer; in fact, the intractability of this problem is the basis of RSA encryption. There are other problems for which quantum computer algorithms give a special advantage, and undoubtedly there are many more to be discovered. In this lecture, I would like

to give you a taste of this subject and describe a simple example in which the use of quantum coherence can give a dramatic speed-up of a computation.

There are some excellent introductory textbooks on quantumm computing. Two that I particularly recommend are *An Introduction to Quantum Computing*, by Kaye, Laflamme, and Mosca, and *Quantum Computer Science*, by Mermin. This lecture will borrow especially from the first of these texts.

In quantum computing, we solve a computational problem by setting up an appropriate initial state, acting on it by a series of appropriate unitary transformations, and, finally, making a measurement on the state that results. A key part of the strategy is to avoid making any measurements, and to avoid collapsing the wavefunction, until the final step.

In classical computing, we can formalize the transformations on classical bits 0 and 1 as products of a standard set of *gates*. Examples of these are 1-bit gates

NOT: $0 \to 1$ $\qquad$ $1 \to 0$

and 2-bit gates

AND: $(0,0) \to 0$ $\quad$ $(0,1) \to 0$ $\quad$ $(1,0) \to 0$ $\quad$ $(1,1) \to 1$

OR: $(0,0) \to 0$ $\quad$ $(0,1) \to 1$ $\quad$ $(1,0) \to 1$ $\quad$ $(1,1) \to 1$

XOR: $(0,0) \to 0$ $\quad$ $(0,1) \to 1$ $\quad$ $(1,0) \to 1$ $\quad$ $(1,1) \to 0$

The gate XOR is called, in longhand, *exclusive OR*. In theoretical computer science, it is proved that these gates are the basis for a *universal* computer. That is, any computation can be carried out by an appropriate sequence of there operations. To implement a classical computer in hardware, then, we need only find a way to implement each operation of the minimal set.

In quantum computing, gates are implemented by unitary actions on qubits. An implementation of NOT is the operation of σ^x

$$\sigma^x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

$\sigma^x(\uparrow) = \sigma^x \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} = |\downarrow\rangle$, $\sigma^x(|\downarrow\rangle) = |\uparrow\rangle$

For the rest of this lecture, I will write $X = \sigma^x$.

The 2-bit operations above cannot be implemented as unitary transformations, since they contract the dimensionality of the Hilbert space. More generally, we cannot implement a function that carries two qubits to one qubit. A unitary transformation must preserve the size of the Hilbert space, and it must be *reversible*, so each result of the operation must be unique. Here is a unitary implementation of XOR; we carry through the first bit to the first bit of the result, and put the result of XOR into the second bit of the result.

$(0,0) \to (0,0)$ $(0,1) \to (0,1)$ $(1,0) \to (1,1)$

$(1,1) \to (1,0)$

This operation can also be written as *controlled-NOT* or c-NOT

$|a\rangle|b\rangle \to |a\rangle (X)^a |b\rangle$ $X^a = \begin{cases} 1 & a=0 \\ X \equiv \sigma^x & a=1 \end{cases}$

A diagrammatic notation for the flow of the computation (left to right) is

NOT $|a\rangle$ 

c-NOT $|a\rangle$ 

$|b\rangle$ 

Notice that

$\left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right)|1\rangle \xrightarrow{\text{c-NOT}} \frac{1}{\sqrt{2}}\left(|0\rangle|1\rangle + |1\rangle|0\rangle\right)$

That is, a unitary action can transform a product state into a state that is not a product and, in fact, encodes a specific correlation between the two states. In this case, we say that the product state is *entangled*. In the example shown, the result of a measurement on the first state is uncorrelated with the result of a measurement on the second state. Measurement of the first state gives 0 or 1 with 50% probability. With the entangled state, the situation is quite different. There are two possibilities for the measurement of the two spins:

$a \to 0 \qquad b \to 1 \qquad$ 50% probability

$a \to 1 \qquad b \to 0 \qquad$ 50% probability

Measurement of each individual state gives 0 or 1 with 50% probability. However, measurement of the value of the first state predicts the measurement of the value of the second state, or vice versa. This property of entanglement is seen in quantum states that arise from natural physical processes. I will give some examples in the next lecture.

It can be show that the set of general 1-qubit gates, together with one 2-qubit gate that can generate entanglement of a product state $|\psi_1\rangle\,|psi_2\rangle$, gives a basis for a universal quantum computer.

An important 1-qubit gate is the *Hadamard gate H*

$$H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \qquad H|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

To save notation, I will denote

$$|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \qquad |-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

Note that

$$H^2 = 1$$

Here is an interesting elementary application of the Hadamard gate to quantum computing, due to Deutsch. There are four possible functions taking $(0,1)$ to $(0,1)$.

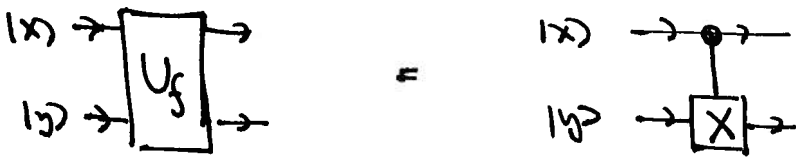
f_0: $0 \to 0$, $1 \to 0$ $\qquad$ f_2: $0 \to 1$, $1 \to 0$

f_1: $0 \to 0$, $1 \to 1$ $\qquad$ f_3: $0 \to 1$, $1 \to 1$

We can implement each of these functions $f_a(x)$ as a linear transformation on a 2-qubit state as by the unitary transformation U_f given by

$$|x\rangle|y\rangle \xrightarrow{f_a} |x\rangle \, | \, \mathrm{XOR}(f(x), y)\rangle$$

The transformation U_f is implemented by action of X on the second qubit, controlled by the first qubit. Diagrammatically,



Two of these functions give equal results when acting on 0 and 1, and two give different results. How easy is it to check that the two results are equal? With a classical computer, we must make two measurements: We measure $f_a(0)$, we measure $f_a(1)$, and we compare the results.

In quantum computing, it is possible to perform this check with only one measurement. Start from the state

$$H|0\rangle \, H|1\rangle = |+\rangle \, |-\rangle = \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right)\left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right)$$

Act with U_f. Notice that

$$\text{if } f(0)=0 \quad |0\rangle\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right) \xrightarrow{f} |0\rangle\frac{|0\rangle-|1\rangle}{\sqrt{2}} = +|0\rangle|-\rangle$$

$$\text{if } f(0)=1 \quad |0\rangle\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right) \xrightarrow{f} |0\rangle\frac{|1\rangle-|0\rangle}{\sqrt{2}} = -|0\rangle|-\rangle$$

Then

$$f_0 : \left(\frac{|0\rangle+|1\rangle}{\sqrt{2}}\right)|-\rangle \longrightarrow \frac{|0\rangle+|1\rangle}{\sqrt{2}}|-\rangle$$

$$f_1 : \left(\frac{|0\rangle+|1\rangle}{\sqrt{2}}\right)|-\rangle \longrightarrow \frac{|0\rangle-|1\rangle}{\sqrt{2}}|-\rangle$$

$$f_2 : \left(\frac{|0\rangle+|1\rangle}{\sqrt{2}}\right)|-\rangle \longrightarrow \frac{-|0\rangle+|1\rangle}{\sqrt{2}}|-\rangle$$

$$f_3 : \left(\frac{|0\rangle+|1\rangle}{\sqrt{2}}\right)|-\rangle \longrightarrow \frac{-|0\rangle-|1\rangle}{\sqrt{2}}|-\rangle$$

Acting with H on the first qubit gives

$$f_0 : \longrightarrow |0\rangle|-\rangle$$

$$f_1 : \longrightarrow |1\rangle|-\rangle$$

$$f_2 : \longrightarrow -|1\rangle|-\rangle$$

$$f_3 : \longrightarrow -|0\rangle|-\rangle$$

Finally, measure the first qubit. If the result is 0, $f(0) = f(1)$; if the result is 1, $f(0) \neq f(1)$.

Here is a more elaborate and stranger example using similar concepts: Let x be a number of n binary digits, that is a number from 0 to $N-1$, where $N = 2^n$. We can pick one such number w to play a special role. For example, we can have a function $f(x)$ such that

$$f(x) = \begin{cases} 1 & x = w \\ 0 & \text{otherwise} \end{cases}$$

The problem is to find w given the function $f(x)$.

Classically, there is no simple solution to this problem. We would have to evaluate $f(x)$ for each possible value of x until we find a value that gives 1. In this worst case, this requires $N-1$ evaluations.

Grover gave a clever quantum algorithm to find w much more rapidly. This algorithm makes use of an n-qubit Hadamard gate. The action of the 1-qubit Hadamard gate gives, for $z = 0$ or 1,

$$|z\rangle \xrightarrow{H} \frac{1}{\sqrt{2}}\left[|0\rangle + (-1)^z|1\rangle\right]$$

In the same notation, the action of H on both states of a 2-qubit state gives

$$|z_1\rangle|z_2\rangle \xrightarrow{H^{(2)}} H|z_1\rangle H|z_2\rangle = \frac{1}{2}\left[|00\rangle + (-1)^{z_1}|10\rangle + (-1)^{z_2}|01\rangle + (-1)^{z_1+z_2}|11\rangle\right]$$

In general, the action of the H on all states of a n-qubit state gives

$$|z_1 z_2 \cdots z_n\rangle \xrightarrow{H^{(n)}} \frac{1}{2^{n/2}} \sum_{y_1 \cdots y_n = 0,1} (-1)^{z_1 y_1 + z_2 y_2 + \cdots z_n y_n} |y_1 y_2 \cdots y_n\rangle$$

In particular

$$|0\rangle = |00\cdots0\rangle \xrightarrow{H^{(n)}} \frac{1}{2^{n/2}} \cdot \left[|00\cdots0\rangle + |00\cdots01\rangle + \cdots + |11\cdots1\rangle\right]$$

To set up the problem for a quantum algorithm, we need a unitary implementation of the function $f(x)$. For this, introduce another state $|r\rangle$ whose initial value is $|-\rangle$. Let U_f be the action of X on this state controlled by x; that is, we evaluate $f(x)$ from the properties of the n qubits, and the apply $X^{f(x)}$ to $|r\rangle$. As we saw in the example of Deutsch's problem, this gives

$$x^f|-\rangle = \begin{cases} +|-\rangle & f = 0 \\ -|-\rangle & f = 1 \end{cases}$$

The sign is an overall sign for the state, and we can equally well consider it as a sign $(+1)$ or (-1) applied to $|x\rangle$. In the rest of the argument, I will take this point of view, and I will not write $|r\rangle$ explicitly.

In addition, it will be useful to define another unitary transformation U_ϕ, corresponding to a function $F_\phi(x)$ with the values

$$F_\phi(x) = \begin{cases} 0 & x = 0 \\ 1 & \text{otherwise} \end{cases}$$

This sends

$|0\rangle = |00\cdots0\rangle \rightarrow +|0\rangle$ $\qquad$ $|x\rangle \rightarrow -|0\rangle$ for $x \neq 0$

Grove showed that it is possible to solve the problem of finding w by starting from the state

$$H^{(n)}|0\rangle$$

and applying repeatedly the set of unitary transformations

$$H^{(n)}|0\rangle \rightarrow \left(\boxed{U_f} \rightarrow \boxed{H^{(n)}} \rightarrow \boxed{U_\phi} \rightarrow \boxed{H^{(n)}} \right)$$

To understand this iteration, we can perform one step in a rough way. The operation U_f modifies the state $H^{(n)}|0\rangle$ by changing the sign in front of the term involving $|w\rangle$

$$H^{(n)}|0\rangle \xrightarrow{U_f} \frac{1}{2^{n/2}}\left[|000\cdots0\rangle + |100\cdots0\rangle + \cdots - |w\rangle + \cdots |11\cdots1\rangle\right]$$

We can rewrite the resulting state as

$$\frac{1}{2^{n/2}}\left[|00\cdots0\rangle + \cdots + |11\cdots1\rangle\right] - \frac{2}{2^{n/2}}|w\rangle$$

These states are not quite orthogonal, but the effect of orthogonalizing these states is only of order $1/\sqrt{N}$. Now act with $H^{(n)}U_0H^{(n)}$. This reverses the sign of the term involving $|w\rangle$, again, up to terms proportional to the overlap between $|w\rangle$ and $H^{(n)}|0\rangle$. The final result is approximately

$$\frac{1}{2^{n/2}}\left[|00\cdots0\rangle + \cdots + |11\cdots1\rangle\right] + \frac{2}{2^{n/2}}|w\rangle$$

The amplitude of $|w\rangle$ has been amplified. The idea of Grover's iteration is that, if we maintain the quantum coherence of the state, each amplitude of the $|w\rangle$ term will be further amplified until this term is dominant.

We can analyze the iteration more carefully. Let

$$|w\rangle = \frac{1}{\sqrt{N-1}} \sum_{x \neq w} |x\rangle$$

and

$$H^{(n)}|0\rangle = |\psi\rangle$$

Then

$$|\psi\rangle = \frac{1}{\sqrt{N}}|\omega\rangle + \sqrt{\frac{N-1}{N}}|\psi\rangle$$

A useful state orthogonal to this one is

$$|\bar{\psi}\rangle = \sqrt{\frac{N-1}{N}}|\omega\rangle - \frac{1}{\sqrt{N}}|\psi\rangle$$

Define

$$\sin\theta = \frac{1}{\sqrt{N}} \qquad \cos\theta = \sqrt{\frac{N-1}{N}}$$

Then the relationship among these four states is

$$|\psi\rangle = \cos\theta|\psi\rangle + \sin\theta|\omega\rangle \qquad |\psi\rangle = \cos\theta|\psi\rangle - \sin\theta|\bar{\psi}\rangle$$

$$|\bar{\psi}\rangle = -\sin\theta|\psi\rangle + \cos\theta|\omega\rangle \qquad |\omega\rangle = \sin\theta|\psi\rangle + \cos\theta|\bar{\psi}\rangle$$

In Grover's iteration, we start from $|\psi\rangle$. The first step of the iteration is

$$|\psi\rangle \xrightarrow{U_f} \cos\theta|\psi\rangle - \sin\theta|\omega\rangle$$

We can rewrite this state in terms of $|\psi\rangle$ and $|\overline{\psi}\rangle$,

$$U_f|\psi\rangle = (\cos^2\theta - \sin^2\theta)|\psi\rangle - 2\sin\theta\cos\theta|\overline{\psi}\rangle$$

$$= \cos 2\theta\,|\psi\rangle - \sin 2\theta\,|\overline{\psi}\rangle$$

Now apply $H^{(n)}$,

$$\xrightarrow{H^{(n)}} \quad \cos 2\theta\,|0\rangle - \sin 2\theta\; H^{(n)}|\overline{\psi}\rangle$$

$$\xrightarrow{U_\phi} \quad \cos 2\theta\,|0\rangle + \sin 2\theta\; H^{(n)}|\overline{\psi}\rangle$$

The state $H^{(n)}|\overline{\psi}\rangle$ is orthogonal to $|0\rangle$, so it gets a (-1) under the application of U_ϕ. Finally, applying $H^{(n)}$, we find

$$\xrightarrow{H^{(n)}} \quad \cos 2\theta\,|\psi\rangle + \sin 2\theta\,|\overline{\psi}\rangle$$

$$= (\cos 2\theta\cos\theta - \sin 2\theta\sin\theta)|\psi\rangle + (\cos 2\theta\sin\theta + \sin 2\theta\cos\theta)|w\rangle$$

$$= \cos 3\theta\,|\psi\rangle + \sin 3\theta\,|w\rangle$$

The complete effect of Grover's iteration is

$$\cos\theta\,|\psi\rangle + \sin\theta\,|w\rangle \xrightarrow{(H^{(n)}U_\phi H^{(n)}U_f)} \cos 3\theta\,|w\rangle + \sin 3\theta\,|w\rangle$$

In a similar way, a second iteration gives

$$H^{(n)}U_\phi H^{(n)}U_f \quad \cos 5\theta\,|\psi\rangle + \sin 5\theta\,|w\rangle$$

After k iterations

$$\longrightarrow \quad \cos(2k+1)\theta \; |\psi\rangle + \sin(2k+1)\theta \; |w\rangle$$

If we can find k such that

$$(2k+1)\theta = \frac{\pi}{2}$$

the term with $|\psi\rangle$ disappears and we find

$$|\psi\rangle \xrightarrow{(H^{(n)} U_\phi H^{(n)} U_\xi)^k} \approx |w\rangle$$

If N is large, the angles in successive iterations are very closely spaced. We can approximate

$$\theta \approx \frac{1}{\sqrt{N}}$$

so value of k that gives the best approximation to $|w\rangle$ is

$$k = \frac{\pi}{4}\sqrt{N}$$

After this number of iterations, we measure the n bits, and, with very high probably, we can read off correctly the binary expansion of w.

There are many more surprises in quantum computing. I encourage you to look further into the textbooks mentioned at the beginning of this lecture to learn more about them.

The leading problem today in quantum computing is the question of how a quantum computer might be realized in a physical system. Quantum computers have been built with handfuls of qubits, but ideally we would like to have systems with millions or billions of qubits. This is what we would need to allow quantum computers to compete with the current generation of computers.