And now for something completely different.

Quantum Information & Computation.

[refs: Preskill's notes, Nielsen & Chuang]

We've already mentioned that it is often convenient to think of quantum evolution directly in terms of unitary operators \( \hat{U} \), rather than dragging out the Schrödinger equation every time. We've also noted that, especially if we're content to focus on finite-dim. Hilbert spaces, it's convenient to consider qubits 1\(\downarrow\rangle = \alpha|0\rangle + \beta|1\rangle\) as the fundamental thing we act on.
Together, these perspectives suggest that we should consider sequences of unitary operations performed on sets of qubits. This is exactly the world of quantum information theory. Of course these ideas have direct application to building actual quantum computers, but their usefulness is more general—e.g. there is growing interest in the application of quantum information to black holes and quantum gravity.

Let's recall a bit (ha ha) about classical computation. There what we work with are actual bits $\{0,1\}$. 
It's convenient to cast our manipulations on bits in terms of circuits, consisting of wires and gates. A wire just carries a bit down the circuit unchanged, while a gate converts a set of input bits into a set of output bits.

Some famous classical gates:

\[ \neg x = x + 1 \]

(Remember arithmetic on bits is carried out mod 2, so \(1+1=0\).)

\[ x \rightarrow \text{DUPE} \rightarrow x \]

\[ x \rightarrow \text{AND} \rightarrow x \land y = x \cdot y \]

\[ x \rightarrow \text{OR} \rightarrow x \lor y = x + y - x \cdot y \]
In truth-table form:

<table>
<thead>
<tr>
<th>(x, y)</th>
<th>(x \land y)</th>
<th>(x \lor y)</th>
<th>(x \oplus y)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0, 0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0, 1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1, 0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1, 1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Famous Theorem: this set of gates
\{ \text{NOT}, \text{AND}, \text{OR}, \text{DUPE} \} is \underline{universal} — any Boolean function \(\mathbb{0}, \mathbb{1}^n \rightarrow \mathbb{0}, \mathbb{1}^m\) is computable by some circuit constructed from them. (Often "DUPE" is taken for granted and left implicit. But in QM such an operation doesn’t even exist, as we’ll see.)

For example, \(\text{XOR} (x, y) = x \oplus y = "x \text{ or } y \text{ but not both}" \ (\text{exclusive OR}).\)
x\text{OR} may be implemented by the following circuit:

Check with $x, y = 0, 1$.

A gate $G$ is **reversible** if $\exists$ a gate $G^{-1}$ s.t. $GG^{-1} = 1$. Obviously, reversible gates have the same number of inputs as outputs. \text{NOT} is reversible, but \text{AND}, \text{OR}, and \text{xOR} are not. The CNOT (controlled-\text{NOT}) gate is reversible:

$x$ \hspace{1cm} CNOT \hspace{1cm} \text{⊕} y$
So CNOT says "keep $x$ fixed, flip $y$ if $x=1$, keep $y$ fixed if $x=0."$ Often drawn as

\[
\begin{array}{c}
\times \\
\times \\
\times \\
\times \\
\times
\end{array}
\]

\[
\begin{array}{c|c}
\times, y & \text{CNOT}(x, y) \\
\hline
00 & 00 \\
01 & 01 \\
10 & 11 \\
11 & 10
\end{array}
\]

We can perform "irreversible" operations (n bits) $\rightarrow$ (m bits) by adding extra "ancilla" bits to the input, and "garbage" bits to the output.

\[
\begin{array}{c}
\text{n input bits} \\
\rightarrow \\
\text{N-n ancilla bits} \\
\rightarrow \\
\text{N-m garbage bits}
\end{array}
\]

\[
\text{reversible circuit}
\]

\[
\text{m output bits}
\]
Let's generalize to quantum circuits, acting on qubits. Classically, the only deterministic one-bit gates are

- the identity \( \equiv \) reversible
- NOT
- \( \{0,1\} \to 0 \equiv \) irreversible
- \( \{0,1\} \to 1 \equiv \)

Obviously, there's a lot more flexibility when we act on qubits. We generally stick to unitary, and therefore reversible, quantum gates, though of course we might want to do some measurements at the end of the process.
Obvious examples are Pauli Matrices:

\[
\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}
\]
\[
\sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}
\]
\[
\sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}
\]

Another useful one is the Hadamard gate:

\[
H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}
\]
\[
\alpha |0\rangle + \beta |1\rangle - \frac{1}{\sqrt{2}} (\alpha |0\rangle + \beta |1\rangle)
\]

\[
H \text{ rotates } |0\rangle \text{ to } |+\rangle, \text{ and } |1\rangle \text{ to } |-\rangle.
\]

Measurements are denoted by

\[
\alpha |0\rangle + \beta |1\rangle - \frac{1}{\sqrt{2}} (\alpha |0\rangle + \beta |1\rangle)
\]

\[
|0\rangle, \text{ probability } |\alpha|^2
\]
\[
|1\rangle, \text{ probability } |\beta|^2.
\]

We could stick measurements anywhere in the circuit, but it always suffices to put them at the end.
There are of course a number of interesting 2-qubit gates. Here we make use of the linearity of QM. We denote a 2-qubit gate as

\[ |x⟩ \rightarrow \text{gate} \rightarrow |y⟩ \text{output} ∈ \mathbb{C}^4 \]

You might think “wait, what if I put an entangled two-qubit state as input?” Because QM is linear, if we know the action of the gate on a complete basis, we know it for any state. And any 2-qubit state has a basis \( |00⟩, |01⟩, |10⟩, |11⟩ \) (and similarly for n-qubit states). So we generally specify our gates by their actions on unentangled basis qubits.
We've already met the quantum CNOT gate:

\[
\begin{array}{c|c}
1 \times \gamma & \text{CNOT}(x, \gamma) \\
\hline
100\gamma & 100\gamma \\
101\gamma & 101\gamma \\
110\gamma & 110\gamma \\
111\gamma & 111\gamma \\
\end{array}
\]

This raises an interesting question. Imagine a classical CNOT, where we fix \( \gamma = 0 \) but leave \( x \) arbitrary. Such a gate simply duplicates \( x \):

\[
\begin{array}{c|c}
\times & \times \\
\hline
0 & X \\
\end{array}
\]  

(classical)

Can we therefore use a quantum CNOT, with \( |\gamma\rangle = 100\gamma \), to duplicate a qubit?
No! Because of entanglement.

\[ |\alpha \psi + \beta \psi'\rangle \otimes |\alpha \phi + \beta \phi'\rangle = (|\alpha \psi\phi\rangle + |\beta \psi'\phi'\rangle) \]

\[ \propto |\alpha \psi\phi\rangle + |\beta \psi'\phi'\rangle. \]

Duplication would be

\[ (|\alpha \psi + \beta \psi'\rangle |\alpha \phi + \beta \phi'\rangle) \]

\[ = |\alpha^2 \psi\phi\rangle + |\alpha \beta \psi'\phi'\rangle + |\beta \alpha \psi\phi'\rangle + |\beta^2 \psi'\phi'\rangle. \]

But it actually goes to \( |\alpha \psi\phi\rangle + |\beta \psi'\phi'\rangle. \)

In fact we have the No-Cloning Theorem (Wootters & Zurek 1982, Dieks 1982).

There exists no quantum circuit acting on n qubits that sends

\[ 1\psi \otimes |\alpha\rangle^{n-1} \rightarrow 1\psi \otimes 1\psi \otimes 1\psi_{n-2}, \]

where \( 1\psi \) is an arbitrary qubit and \( 1\psi_{n-2} \) is any "garbage" output state.
Proof: By contradiction. Assume that a cloning circuit exists, implemented by a unitary $\hat{U}$. Then we know

$$\hat{U} \left( 10^n \right) = 10 \otimes 10 \otimes 1 \Psi(0) \rangle,$$

$$\hat{U} \left( 11 \otimes 10^n \right) = 11 \otimes 11 \otimes 1 \Psi(1) \rangle.$$  

By linearity, acting on $1+\gamma = \frac{1}{\sqrt{2}} \left( 10^n + 11^n \right)$ gives

$$\hat{U} \left( 1+\gamma \otimes 10^n \right) = \frac{1}{\sqrt{2}} \left( 10 \otimes 1 \Psi(0) \rangle + \frac{1}{\sqrt{2}} \left( 11 \otimes 1 \Psi(1) \rangle. \right)$$

But if it really cloned, it should give

$$\hat{U} \left( 1+\gamma \otimes 10^n \right)^2 = \frac{1}{2} \left( 10 \otimes 1 \Psi(0) \rangle \otimes 1 \Psi(0) \rangle + \frac{1}{2} \left( 11 \otimes 1 \Psi(1) \rangle \otimes 1 \Psi(1) \rangle. \right)$$

These are not the same (e.g. there are no 1017 terms in the first expression), therefore no such unitary exists.

This is important! E.g. a quantum computer can't use classical error correction, just by duplicating its internal state many times. Need to be more subtle.
We can use a quantum circuit to entangle two qubits. Add a Hadamard to our CNOT:

\[
\begin{array}{c}
\text{CNOT:} \\
\begin{array}{c}
\frac{1}{\sqrt{2}}(00 + 11) \\
0^1 \quad 0^0 \\
\end{array}
\end{array}
\]

\[
\frac{1}{\sqrt{2}}(00 + 11) = |\Psi^+\rangle
\]

This is one of our Bell states, representing an EPR pair.