Quantum Computing - Lecture Notes

Mark Oskin *
Department of Computer Science and Engineering
University of Washington

Abstract

The following lecture notes are based on the book *Quantum Computation and Quantum Information* by Michael A. Nielsen and Isaac L. Chuang. They are for a math-based quantum computing course that I teach here at the University of Washington to computer science graduate students (with advanced undergraduates admitted upon request). These notes start with a brief linear algebra review and proceed quickly to cover everything from quantum algorithms to error correction techniques. The material takes approximately 16 hours of lecture time to present. As a service to educators, the LaTeXand *Xfig* source to these notes is available online from my home page: `http://www.cs.washington.edu/homes/oskin`. In addition, under the section "course material" from my web page, in the spring quarter/2002 590mo class you will find a sequence of homework assignments geared to computer scientists. Please feel free to adapt these notes and assignments to whatever classes your may be teaching. Corrections and expanded material are welcome; please send them by email to `oskin@cs.washington.edu`.

*The following work is supported in part by NSF CAREER Award ACR-0133188.

Contents

1 Linear Algebra (short review)

The following linear algebra terms will be used throughout these notes.

Z^* - complex conjugate

if $Z = a + b \cdot i$ then $Z^* = a - b \cdot i$

$|\psi\rangle$ - vector, "ket" i.e.

$$\begin{bmatrix} c_1 \\ c_2 \\ \dots \\ c_n \end{bmatrix}$$

$|\psi\rangle$ - vector, "bra" i.e.

$[c_1^*, c_2^*, ..., c_n^*]$

$\langle\varphi|\psi\rangle$ - inner product between vectors $|\varphi\rangle$ and $|\psi\rangle$.

Note for QC this is on $\mathbb{C}^n$ space not $\mathbb{R}^n$!

Note $\langle\varphi|\psi\rangle = \langle\psi|\varphi\rangle^*$

Example: $|\varphi\rangle = \begin{bmatrix} 2 \\ 6i \end{bmatrix}$, $|\psi\rangle = \begin{bmatrix} 3 \\ 4 \end{bmatrix}$

$\langle\varphi|\psi\rangle = [2, -6i] \begin{bmatrix} 3 \\ 4 \end{bmatrix} = 6 - 24i$

$|\varphi\rangle \otimes |\psi\rangle$ - tensor product of $|\varphi\rangle$ and $|\psi\rangle$.

Also written as $|\varphi\rangle|\psi\rangle$

Example: $|\varphi\rangle|\psi\rangle = \begin{bmatrix} 2 \\ 6i \end{bmatrix} \otimes \begin{bmatrix} 3 \\ 4 \end{bmatrix} = \begin{bmatrix} 2 \times 3 \\ 2 \times 4 \\ 6i \times 3 \\ 6i \times 4 \end{bmatrix} = \begin{bmatrix} 6 \\ 8 \\ 18i \\ 24i \end{bmatrix}$

A^* - complex conjugate of matrix A.

if $A = \begin{bmatrix} 1 & 6i \\ 3i & 2+4i \end{bmatrix}$ then $A^* = \begin{bmatrix} 1 & -6i \\ -3i & 2-4i \end{bmatrix}$

A^T - transpose of matrix A.

if $A = \begin{bmatrix} 1 & 6i \\ 3i & 2+4i \end{bmatrix}$ then $A^T = \begin{bmatrix} 1 & 3i \\ 6i & 2+4i \end{bmatrix}$

$A^\dagger$ - Hermitian conjugate (adjoint) of matrix A.

Note $A^\dagger = \left(A^T\right)^*$

if $A = \begin{bmatrix} 1 & 6i \\ 3i & 2+4i \end{bmatrix}$ then $A^\dagger = \begin{bmatrix} 1 & -3i \\ -6i & 2-4i \end{bmatrix}$

$\| |\psi\rangle \|$ - norm of vector $|\psi\rangle$
$\| |\psi\rangle \| = \sqrt{\langle\psi|\psi\rangle}$
Important for normalization of $|\psi\rangle$ i.e. $|\psi\rangle / \| |\psi\rangle \|$

$\langle\varphi|A|\psi\rangle$ - inner product of $|\varphi\rangle$ and $A|\psi\rangle$.
or inner product of $A^\dagger|\varphi\rangle$ and $|\psi\rangle$

2 Postulates of Quantum Mechanics

An important distinction needs to be made between quantum mechanics, quantum physics and quantum computing. Quantum mechanics is a mathematical language, much like calculus. Just as classical physics uses calculus to explain nature, quantum physics uses quantum mechanics to explain nature. Just as classical computers can be thought of in boolean algebra terms, quantum computers are reasoned about with quantum mechanics. There are four postulates to quantum mechanics, which will form the basis of quantum computers:

- **Postulate 1:** Definition of a quantum bit, or *qubit*.
- **Postulate 2:** How qubit(s) transform (evolve).
- **Postulate 3:** The effect of measurement.
- **Postulate 4:** How qubits combine together into systems of qubits.

2.1 Postulate 1: A quantum bit

Postulate 1 (Nielsen and Chuang, page 80):

> "Associated to any *isolated* physical system is a complex vector space with inner product (i.e. a Hilbert space) known as the state space of the system. The system is completely described by its state vector, which is a *unit vector* in the system's state space."

Consider a single qubit - a two-dimensional state space. Let $|\phi_0\rangle$ and $|\phi_1\rangle$ be orthonormal basis for the space. Then a qubit $|\psi\rangle = a|\phi_0\rangle + b|\phi_1\rangle$. In quantum computing we usually label the basis with some boolean name but note carefully that this is *only* a name. For example, $|\phi_0\rangle = |0\rangle$ and $|\phi_1\rangle = |1\rangle$. Making this more concrete one might imagine that "$|0\rangle$" is being represented by an up-spin while "$|1\rangle$" by a down-spin. The key is there is an abstraction between the technology

(spin state or other quantum phenomena) and the logical meaning. This same detachment occurs classically where we traditionally call a high positive voltage "1" and a low ground potential "0".

Note that $|\psi\rangle = a|0\rangle + b|1\rangle$ must be a unit vector. In other words, $\langle\psi|\psi\rangle = 1$ or $|a|^2 + |b|^2 = 1$. For quantum computing $\{a,b\} \in \mathbb{C}$

This formalism for a quantum bit is a direct extension of one way to describe a classical computer. That is, way may write that a classical bit $|\omega\rangle$ is in the state $|\omega\rangle = x|0\rangle + y|1\rangle$. The only difference is x and y are defined not over the complex numbers but rather from the set $\{0,1\}$. That is $\{x,y\} \in \{0,1\}$. The same normalization condition applies $|x|^2 + |y|^2 = 1$. This normalization condition is not a property of quantum mechanics but rather of probability theory.

2.2 Postulate 2: Evolution of quantum systems

Postulate 2 (Nielsen and Chuang, page 81):

> "The evolution of a *closed* quantum system is described by a *unitary transformation*. That is, the state $|\psi\rangle$ of the system at time t_1 is related to the state of $|\psi'\rangle$ of the system at time t_2 by a unitary operator U which depends only on times t_1 and t_2."

I.e. $|\psi'\rangle = U|\psi\rangle$.

The fact that U cannot depend on $|\psi\rangle$ and only on t_1 and t_2 is a subtle and disappointing fact. We will see later that if U could depend on $|\psi\rangle$ then quantum computers could easily solve NP complete problems! Conceptually think of U as something you can apply to a quantum bit but you cannot conditionally apply it. The transform occurs without any regard to the current state of $|\psi\rangle$.

Example:

$$|\psi\rangle = a|0\rangle + b|1\rangle$$
$$U = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$
$$|\psi'\rangle = U|\psi\rangle = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} b \\ a \end{bmatrix} = b|0\rangle + a|1\rangle$$

Example:

$$\text{Let } |\psi\rangle = 1|0\rangle + 0|1\rangle = |0\rangle$$
$$U = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$
$$|\psi'\rangle = U|\psi\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$$

Important: U must be unitary, that is $U^\dagger U = I$

Example:

$$U = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \text{ then } U^\dagger = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$
$$U^\dagger U = \frac{1}{\sqrt{2}} \cdot \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} = I$$

2.3 Postulate 3: Measurement

Postulate 3 (Nielsen and Chuang, page 84):

> "Quantum measurements are described by a collection $\{M_m\}$ of measurement operators. These are operators acting on the state space of the system being measured. The index *m* refers to the measurement outcomes that may occur in the experiment. If the state of the quantum system is $|\psi\rangle$ immediately before the measurement then the probability that result *m* occurs is given by:
> $$p(m) = \langle\psi|M_m^\dagger M_m|\psi\rangle$$
> and the state of the system after measurement is:
> $$\frac{M_m|\psi\rangle}{\sqrt{\langle\psi|M_m^\dagger M_m|\psi\rangle}}$$
> The measurement operators satisfy the *completeness equation*:
> $$\sum_m \langle\psi|M_m^\dagger M_m|\psi\rangle = I$$
> The completeness equation expresses the fact that probabilities sum to one:
> $$1 = \sum_m p(m) = \sum_m \langle\psi|M_m^\dagger M_m|\psi\rangle$$ "

Some important measurement operators are $M_0 = |0\rangle\langle 0|$ and $M_1 = |1\rangle\langle 1|$

$$M_0 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} [1,0] = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$$

$$M_1 = \begin{bmatrix} 0 \\ 1 \end{bmatrix} [0,1] = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$$

Observe that $M_0^\dagger M_0 + M_1^\dagger M_1 = I$ and are thus complete.

Example:

$$|\psi\rangle = a|0\rangle + b|1\rangle$$
$$p(0) = \langle\psi|M_0^\dagger M_0|\psi\rangle$$

Note that $M_0^\dagger M_0 = M_0$, hence

$$p(0) = \langle\psi|M_0|\psi\rangle = [a^*, b^*] \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} =$$

$$= [a^*, b^*] \begin{bmatrix} a \\ 0 \end{bmatrix} = |a|^2$$

Hence the probability of measuring $|0\rangle$ is related to its probability amplitude a by way of $|a|^2$.

It important to note that the state after measurement is related to the outcome of the measurement. For example, suppose $|0\rangle$ was measured, then the state of the system after this measurement is re-normalized as:

$\frac{M_0|\psi\rangle}{|a|} = \frac{a}{|a|}|0\rangle$

As a side note we are forced to wonder if Postulate 3 can be derived from Postulate 2. It seems natural given that measurement in the physical world is just interacting a qubit with other qubits. Thus it seems strange to have measurement be its own postulate. At this point though physicists don't know how derive the measurement postulate from the other three, so we shall just have to be pragmatic and accept it.

2.4 Postulate 4: Multi-qubit systems

Postulate 4 (Nielsen and Chuang, page 94):

> "The state space of a composite physical system is the tensor product of the state spaces of the component physical systems. [sic] e.g. suppose systems 1 through n and system i is in state $|\psi_i\rangle$, then the joint state of the total system is $|\psi_1\rangle \otimes |\psi_2\rangle \otimes \ldots \otimes |\psi_n\rangle$."

Example:

Suppose $|\psi_1\rangle = a|0\rangle + b|1\rangle$ and $|\psi_2\rangle = c|0\rangle + d|1\rangle$, then:

$$|\psi_1\rangle \otimes |\psi_2\rangle = |\psi_1\psi_2\rangle = a \cdot c|0\rangle|0\rangle + a \cdot d|0\rangle|1\rangle + b \cdot c|1\rangle|0\rangle + b \cdot d|1\rangle|1\rangle =$$
$$ac|00\rangle + ad|01\rangle + bc|10\rangle + bd|11\rangle$$

Why the tensor product? This is not a proof, but one would expect some way to describe a composite system. Tensor product works for classical systems (except the restricted definition of the probability amplitudes makes it so that the result is a simple concatenation). For quantum systems tensor product captures the essence of superposition, that is if system A is in state $|A\rangle$ and B in state $|B\rangle$ then there should be some way to have a little of A and a little of B. Tensor product exposes this.

3 Entanglement

Entanglement is a uniquely quantum phenomenon. Entanglement is a property of a multi-qubit state space (multi-qubit system) and can be thought of as a resource. To explain entanglement we'll examine the creation and destruction of an EPR pair of qubits named after Einstein, Podolsky, and Rosen.

Suppose you begin with a qubit $|\psi_1\rangle$ in a zero $|0\rangle$ state.

Let $U = H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$

Then let $|\psi_1'\rangle = H|\psi_1\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$

Now take another qubit $|\psi_2\rangle$ also in the zero $|0\rangle$ state. The joint state-space probability vector is the tensor product of these two:

$$|\psi_1'\rangle \otimes |\psi_2\rangle = |\psi_1'\psi_2\rangle = \frac{1}{\sqrt{2}}|00\rangle + 0|01\rangle + \frac{1}{\sqrt{2}}|10\rangle + 0|11\rangle$$

Now define a new unitary transform:

$$CNot = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

(As an exercise show that *CNot* is unitary), but for now lets just apply *CNot* to our two qubits:

$$|(\psi_1'\psi_2)''\rangle = CNot|\psi_1'\psi_2\rangle = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ \frac{1}{\sqrt{2}} \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ 0 \\ \frac{1}{\sqrt{2}} \end{bmatrix} = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$

The key to entanglement is the property that the state space cannot be decomposed into component spaces. That is, for our example, there does not exists any $|\varphi_1\rangle$ and $|\varphi_2\rangle$ such that $|\varphi_1\rangle \otimes |\varphi_2\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$.

To illustrate why entanglement is so strange, lets consider performing a measurement just prior to applying the *CNot* gate. The two measurement operators (for obtaining a $|0\rangle$ or a $|1\rangle$) are:

$$M_{0_2} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \text{ and } M_{1_2} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Recall that just prior to the *CNot* the system is in the state $|\psi_1'\psi_2\rangle = \frac{1}{\sqrt{2}}|00\rangle + 0|01\rangle + \frac{1}{\sqrt{2}}|10\rangle + 0|11\rangle$, hence the result of measuring the second qubit will clearly be $|0\rangle$. Note that $M_{0_2}^\dagger M_{0_2} = M_{0_2}$. Therefore:

$$p(0) = \langle\psi_1'\psi_2|M_{0_2}^\dagger M_{0_2}|\psi_1'\psi_2\rangle = \langle\psi_1'\psi_2|M_{0_2}|\psi_1'\psi_2\rangle =$$
$$\left[\tfrac{1}{\sqrt{2}}, 0, \tfrac{1}{\sqrt{2}}, 0\right] \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ \frac{1}{\sqrt{2}} \\ 0 \end{bmatrix} = \left[\tfrac{1}{\sqrt{2}}, 0, \tfrac{1}{\sqrt{2}}, 0\right] \begin{bmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ \frac{1}{\sqrt{2}} \\ 0 \end{bmatrix} = 1$$

After measurement: $\frac{M_m|\psi\rangle}{\sqrt{\langle\psi|M_m^\dagger M_m|\psi\rangle}} = \frac{\begin{bmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ \frac{1}{\sqrt{2}} \\ 0 \end{bmatrix}}{1} = |\psi_1'\psi_2\rangle$

We can see that measurement had no effect on the first qubit. It remains in a superposition of $|0\rangle$ and $|1\rangle$. Now lets consider the same measurement but just after the *CNot* gate is applied. Here:

$$|\psi\rangle = |(\psi_1'\psi_2)''\rangle = \tfrac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$

Now it is not clear whether the second qubit will return a $|0\rangle$ or a $|1\rangle$, both outcomes are equally likely. To see this, lets calculate the probability of obtaining $|0\rangle$:

$$p(0) = \langle\psi|M_{0_2}^\dagger M_{0_2}|\psi\rangle = \langle\psi|M_{0_2}|\psi\rangle =$$
$$= \left[\tfrac{1}{\sqrt{2}}, 0, 0, \tfrac{1}{\sqrt{2}}\right] \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ 0 \\ \frac{1}{\sqrt{2}} \end{bmatrix} = \left[\tfrac{1}{\sqrt{2}}, 0, 0, \tfrac{1}{\sqrt{2}}\right] \begin{bmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ 0 \\ 0 \end{bmatrix} = \tfrac{1}{2}$$

Hence, after the *CNot* gate is applied we have only a $1/2$ chance of obtaining $|0\rangle$. Of particular interest to our discussion, however, is what happens to the state vector of the system:

$$\text{After measurement: } \frac{M_m|\psi\rangle}{\sqrt{\langle\psi|M_m^\dagger M_m|\psi\rangle}} = \frac{1}{\sqrt{1/2}} \times \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ 0 \\ \frac{1}{\sqrt{2}} \end{bmatrix} =$$

$$\frac{1}{\sqrt{1/2}} \times \begin{bmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = |00\rangle$$

This is the remarkable thing about entanglement. By measuring one qubit we can affect the probability amplitudes of the other qubits in a system! How to think about this process in an abstract way is an open challenge in quantum computing. The difficulty is the lack of any classical analog. One useful, but imprecise way to think about entanglement, superposition and measurement is that superposition "is" quantum information. Entanglement links that information across quantum bits, but does not create any more of it. Measurement "destroys" quantum information turning it into classical. Thus think of an EPR pair as having as much "superposition" as an unentangled set of qubits, one in a superposition between zero and one, and another in a pure state. The superposition in the EPR pair is simply linked across qubits instead of being isolated in one.

This, admittedly fuzzy way of thinking about these concepts is useful when we examine teleportation. There we insert an unknown quantum state (carrying a fixed amount of "quantum information") into a system of qubits. We mix them about with additional superposition and entanglement and then measure out the superposition we just added. The net effect is the unknown quantum state remains in the joint system of qubits, albeit migrated through entanglement to another physical qubit.

4 Teleportation

Contrary to its sci-fi counterpart, quantum teleportation is rather mundane. Quantum teleportation is a means to replace the *state* of one qubit with that of another. It gets its out-of-this-world name from the fact that the state is "transmitted" by setting up an entangled state-space of three qubits and then removing two qubits from the entanglement (via measurement). Since the information of the source qubit is preserved by these measurements that "information" (i.e. state) ends up in the final third, destination qubit. This occurs, however, without the source (first) and destination (third) qubit ever directly interacting. The interaction occurs via entanglement.

Suppose $|\psi\rangle = a|0\rangle + b|1\rangle$ and given an EPR pair $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ the state of the entire system is:

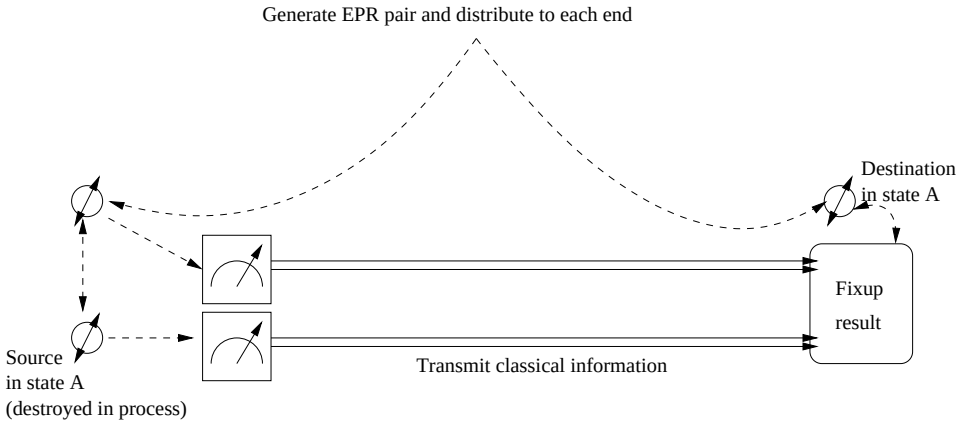

Figure 1: Teleportation works by pre-transmitting an EPR pair to the source and destination. The qubit containing the state to be "teleported" interacts with onehalf of this EPR pair creating a joint state space. It is then measured and only classical information is transmitted to the destination. This classical information is used to "fixup" the destination qubit

$$\frac{1}{\sqrt{2}}\left[a|0\rangle\left(|00\rangle+|11\rangle\right)+b|1\rangle\left(|00\rangle+|11\rangle\right)\right]=\frac{1}{\sqrt{2}}\begin{bmatrix} a\\0\\0\\a\\b\\0\\0\\b\end{bmatrix}$$

Perform the *CNot* operation and you obtain

$$\frac{1}{\sqrt{2}}\left[a|0\rangle\left(|00\rangle+|11\rangle\right)+b|1\rangle\left(|10\rangle+|01\rangle\right)\right]=\frac{1}{\sqrt{2}}\begin{bmatrix} a\\0\\0\\a\\0\\b\\b\\0\end{bmatrix}$$

Next we apply the H gate. However, as an aside, lets examine what happens when we apply the H gate to $|0\rangle$ and to $|1\rangle$. Recall that:

$$H=\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\1 & -1\end{bmatrix}$$

$$H|0\rangle=\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\1 & -1\end{bmatrix}\begin{bmatrix}1\\0\end{bmatrix}=\frac{1}{\sqrt{2}}\begin{bmatrix}1\\1\end{bmatrix}$$

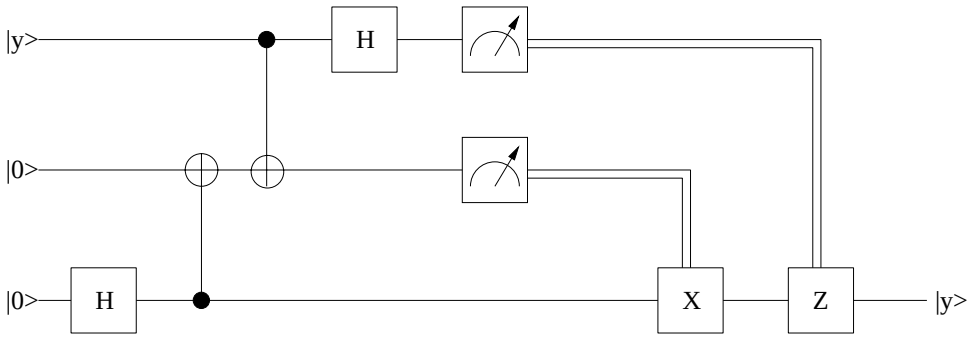

Figure 2: Quantum circuit depicting teleportation. Note that in this diagram single lines represent quantum data while double lines represent classical information.

$$H|1\rangle = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\begin{bmatrix} 0 \\ 1 \end{bmatrix} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

Thus, applying H to our system we have:

$$|\varphi\rangle = \frac{1}{\sqrt{2}}\left[\frac{1}{\sqrt{2}}a\left(|0\rangle+|1\rangle\right)\left(|00\rangle+|11\rangle\right)+\frac{1}{\sqrt{2}}b\left(|0\rangle-|1\rangle\right)\left(|10\rangle+|01\rangle\right)\right] = \frac{1}{2}\begin{bmatrix} a \\ b \\ b \\ a \\ a \\ -b \\ -b \\ a \end{bmatrix}$$

We can rewrite this expression as:

$$\begin{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} \\ \begin{bmatrix} b \\ a \end{bmatrix} \\ \begin{bmatrix} a \\ -b \end{bmatrix} \\ \begin{bmatrix} -b \\ a \end{bmatrix} \end{bmatrix} = \frac{1}{2}\left[|00\rangle\left(a|0\rangle+b|1\rangle\right)+|01\rangle\left(a|1\rangle+b|0\rangle\right)+|10\rangle\left(a|0\rangle-b|1\rangle\right)+|11\rangle\left(a|1\rangle-b|0\rangle\right)\right]$$

Which we can shorten to:

$$\frac{1}{2}\left[|00\rangle\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}|\psi\rangle+|01\rangle\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}|\psi\rangle+|10\rangle\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}|\psi\rangle+|11\rangle i\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}|\psi\rangle\right]$$

These gates are the famous “Pauli” (I,X,Z,Y) operators and this is also written as:

$$\tfrac{1}{2}\left[|00\rangle I|\psi\rangle + |01\rangle X|\psi\rangle + |10\rangle Z|\psi\rangle + |11\rangle iY|\psi\rangle\right]$$

And of interest to us with teleportation:

$$|\varphi\rangle = \tfrac{1}{2}\left[|00\rangle I|\psi\rangle + |01\rangle X|\psi\rangle + |10\rangle Z|\psi\rangle + |11\rangle XZ|\psi\rangle\right]$$

This implies that we can measure the first and second qubit and obtain two classical bits. These two classical bits tell us what transform was applied to the third qubit. Thereby we can "fixup" the third qubit by knowing the classical outcome of the measurement of the first two qubits. This fixup is fairly straightforward, either applying nothing, X, Z or both X and Z. Lets work through an example:

$$M_{10} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$P(10) = \langle\varphi|M_{10}^\dagger M_{10}|\varphi\rangle = \langle\varphi|M_{10}|\varphi\rangle$, since here $M_{10}^\dagger M_{10}$. Thus:

$$M_{10}|\varphi\rangle = \tfrac{1}{2}\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ a \\ -b \\ 0 \\ 0 \end{bmatrix}$$

Therefor: $\langle\varphi|M_{10}|\varphi\rangle = \tfrac{1}{2}[a,b,b,a,a,-b,-b,a]\tfrac{1}{2}\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ a \\ -b \\ 0 \\ 0 \end{bmatrix} = \tfrac{1}{4}[a\cdot a^* + b\cdot b^*]$

Recall that by definition of a qubit we know that $a\cdot a^* + b\cdot b^* = 1$, hence the probability of measuring 01 is $1/4$. The same is true for the other outcomes.

5 Super-dense Coding

Super dense coding is the less popular sibling of teleportation. It can actually be viewed as the process in reverse. The idea is to send two classical bits of information by only sending one quantum bit. The process starts out with an EPR pair that is shared between the receiver and sender (the sender has one half and the receiver has the other).

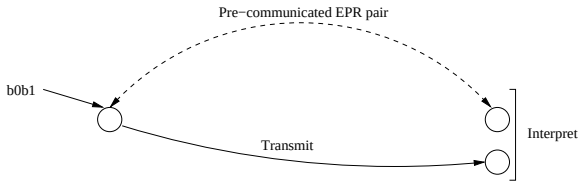

Figure 3: Super-dense coding works by first pre-communicating an EPR pair. To send two bits of classical information one half of this EPR pair (a single qubit) is manipulated and sent to the other side. The two qubits (the one pre-communicated and the one sent) are then interacted and the resulting two bits of classical information is obtained.

$$|\psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$

To send information apply particular quantum gates:

- **00**: apply I (i.e. do nothing)
- **01**: apply Z
- **10**: apply X
- **11**: apply iY (i.e. both X and Z)

- **00**: $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \longrightarrow \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}$
- **01**: $\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \longrightarrow \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle) = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 0 \\ 0 \\ -1 \end{bmatrix}$
- **10**: $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \longrightarrow \frac{1}{\sqrt{2}}(|10\rangle + |01\rangle) = \frac{1}{\sqrt{2}} \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$

- **11**: $i\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}\frac{1}{\sqrt{2}}(|00\rangle+|11\rangle) \longrightarrow \frac{1}{\sqrt{2}}(|01\rangle-|10\rangle) = \frac{1}{\sqrt{2}}\begin{bmatrix} 0 \\ 1 \\ -1 \\ 0 \end{bmatrix}$

These states are known as *bell basis states*. They key to super-dense coding is that they are orthonormal from eachother and are hence distinguishable by a quantum measurement.

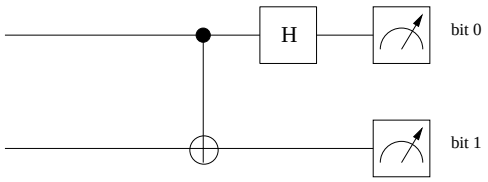

Figure 4: To obtain the two bits of classical information a bell-basis measurement is performed.

Examining this process in more detail:

$|\psi_{00}\rangle = \frac{1}{\sqrt{2}}(|00\rangle+|11\rangle) = \frac{1}{\sqrt{2}}(|0\rangle|0\rangle+|1\rangle|1\rangle)$
apply *CNot* gives us: $\frac{1}{\sqrt{2}}(|0\rangle|0\rangle+|1\rangle|0\rangle)$
apply H gives us: $\frac{1}{\sqrt{2}}\frac{1}{\sqrt{2}}((|0\rangle+|1\rangle)|0\rangle+(|0\rangle-|1\rangle)|0\rangle) =$
$\frac{1}{2}(|00\rangle+|10\rangle+|00\rangle-|10\rangle) = |00\rangle$

$|\psi_{01}\rangle = \frac{1}{\sqrt{2}}(|00\rangle-|11\rangle) = \frac{1}{\sqrt{2}}(|0\rangle|0\rangle-|1\rangle|1\rangle)$
apply *CNot* gives us: $\frac{1}{\sqrt{2}}(|0\rangle|0\rangle-|1\rangle|0\rangle)$
apply H gives us: $\frac{1}{\sqrt{2}}\frac{1}{\sqrt{2}}((|0\rangle+|1\rangle)|0\rangle+(|1\rangle-|0\rangle)|0\rangle) =$
$\frac{1}{2}(|00\rangle+|10\rangle+|10\rangle-|00\rangle) = |10\rangle$

$|\psi_{01}\rangle = \frac{1}{\sqrt{2}}(|10\rangle+|01\rangle) = \frac{1}{\sqrt{2}}(|1\rangle|0\rangle+|0\rangle|1\rangle)$
apply *CNot* gives us: $\frac{1}{\sqrt{2}}(|1\rangle|1\rangle+|0\rangle|1\rangle)$
apply H gives us: $\frac{1}{\sqrt{2}}\frac{1}{\sqrt{2}}((|0\rangle-|1\rangle)|1\rangle+(|0\rangle+|1\rangle)|1\rangle) =$
$\frac{1}{2}(|01\rangle-|11\rangle+|01\rangle+|11\rangle) = |01\rangle$

$\longrightarrow$ Leave the last one for homework.

6 Deutsch's Algorithm

Deutsch's algorithm is a perfect illustration of all that is miraculous, subtle, and disappointing about quantum computers. It calculates a solution to a problem faster than any classical computer

ever can. It illustrates the subtle interaction of superposition, phase-kick back, and interference. Finally, unfortunately, is solves a completely pointless problem.

Deutsch's algorithm answers the following question: suppose $f(x)$ is either constant or balanced, which one is it? If $f(x)$ were constant then for all x the result is either 0 or 1. However, if $f(x)$ were balanced then for one half of the inputs $f(x)$ is 0 and for the other half it is 1 (which x's correspond to 0 or 1 is completely arbitrary). To answer this question classically, we clearly need to query for both $x = 0$ and $x = 1$, hence two queries are required. Quantum mechanically though we will illustrate how this can be solved in just one query.

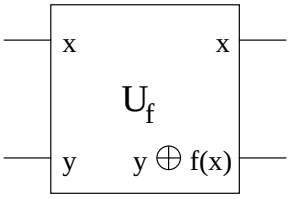

Figure 5: Suppose U_f implements f, x is input as $(|0\rangle + |1\rangle)/\sqrt{2}$ and y as $|0\rangle$, then the remarkable aspect of quantum computing is the output is equal to $(|0, f(0)\rangle + |1, f(1)\rangle)/\sqrt{2}$.

We begin by illustrating how superposition of quantum state creates *quantum parallelism* or the ability to compute on many states simultaneously.

Given a function $f(x) : \{0,1\} \rightarrow \{0,1\}$ using a quantum computer, use two qubits $|x,y\rangle$ and transform them into $|x, y \oplus f(x)\rangle$ (where $\oplus$ represents addition modular two). We use two qubits since we wish to leave the input x or the query register, "un-changed". The second qubit, y, acts as a result register. Let U_f be the unitary transform that implements this. This is illustrated in Figure 5.

Suppose we wish to calculate $f(0)$, then we could input x as $|0\rangle$, and y, our output register, as $|0\rangle$ and apply the U_f transform.

The input is written as $|0\rangle \otimes |0\rangle = |0,0\rangle$.

The output is transformed by U_f to be $|0, 0 \oplus f(0)\rangle$.

Suppose we wish to calculate $f(1)$, then we could input x as $|1\rangle$, and y, our output register, as $|0\rangle$ and apply the U_f transform.

The input is written as $|1\rangle \otimes |0\rangle = |1,0\rangle$.

The output is transformed by U_f to be $|1, 0 \oplus f(1)\rangle$.

But this is not a classical computer – we can actually query the results of 0 and 1 simultaneously using quantum parallelism. For this, let x equal $(|0\rangle + |1\rangle)/\sqrt{2}$ and y equal 0.

The input $|\psi_1\rangle = \frac{|0,0\rangle + |1,0\rangle}{\sqrt{2}}$

The output $|\psi_2\rangle = \frac{|0,f(0)\rangle + |1,f(1)\rangle}{\sqrt{2}}$

$\longrightarrow$ Remarkable: U_f is applied to $|0\rangle$ and $|1\rangle$ simultaneously! This is known as quantum parallelism.

$\longrightarrow$ Problem: sounds good, but measurement produces either $|0, f(0)\rangle$ or $|1, f(1)\rangle$. Hence we need to be clever about what type of question we ask, and how we go about extracting the answer.

The solution is to use another quantum mechanical property: *interference*.

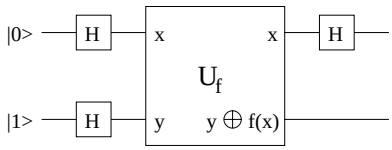

Figure 6: Deutsch's Algorithm uses quantum parallelism and interference to extract information about a global property of the solution space.

Key:
Deutsch's algorithm, as all known quantum algorithms that provide exponential speedup over classical systems do, answers a question about a global property of a solution space. These are often called *promise problems*, whereby the structure of the solution space is promised to be of some form and by carefully using superposition, entanglement and interference we can extract information about that structure. The reason these problems obtain exponential improvement over all known classical algorithms is that classically one has to calculate every point in the solution space in order to obtain full knowledge about this structure. Quantum mechanically we calculate every point using quantum parallelism. Unfortunately this is often **not** how most algorithms are phrased. Usually we work with problems that are phrased of the form "what x gives a value of $f(x)$ with the desired property?" Thus far, quantum computers can only provide square-root improvement to such query-based problems.

Let $|\psi_0\rangle$ be the initial state vector and $|\psi_1\rangle$ be the state of the system prior to applying U_f. Let $|\psi_2\rangle$ be the state of the system after applying U_f and $|\psi_3\rangle$ be the state of the system prior to measurement.

Input: $|\psi_0\rangle = |0,1\rangle$

It may seem strange to start out with a result register of 1 instead of 0, but ignore this for now, we will return to it shortly. Apply the H gate to the query and result registers to obtain:

$$|\psi_1\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

Now, lets examine $y \oplus f(x)$:

Suppose $f(x) = 0$ then $y \oplus f(x) = y \oplus 0 = \frac{1}{\sqrt{2}}(|0 \oplus 0\rangle - |1 \oplus 0\rangle) = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$

Suppose $f(x) = 1$ then $y \oplus f(x) = y \oplus 1 = \frac{1}{\sqrt{2}}(|0 \oplus 1\rangle - |1 \oplus 1\rangle) = \frac{1}{\sqrt{2}}(-|0\rangle + |1\rangle)$

We can compactly describe this behavior in the following formula:

$$y \oplus f(x) = (-1)^{f(x)} \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

Thus, U_f transforms $|x\rangle \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$ into:

$$(-1)^{f(x)}|x\rangle \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

Or we can say that:

$$U_f\left[\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)\right] = \frac{1}{2}\left[(-1)^{f(0)}|0\rangle(|0\rangle - |1\rangle) + (-1)^{f(1)}|1\rangle(|0\rangle - |1\rangle)\right]$$

Suppose f is constant, that is $f(0) = f(1)$, then:

$$\frac{1}{2}\left[(-1)^{f(0)}|0\rangle(|0\rangle - |1\rangle) + (-1)^{f(1)}|1\rangle(|0\rangle - |1\rangle)\right]$$

$$= \frac{1}{2}(-1)^{f(0)}\left[|0\rangle(|0\rangle - |1\rangle) + |1\rangle(|0\rangle - |1\rangle)\right]$$

$$= \pm\frac{1}{2}\left[|0\rangle(|0\rangle - |1\rangle) + |1\rangle(|0\rangle - |1\rangle)\right]$$

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

Suppose instead that f is balanced, that is $f(0) \neq f(1)$, then:

$$\frac{1}{2}\left[(-1)^{f(0)}|0\rangle(|0\rangle - |1\rangle) + (-1)^{f(1)}|1\rangle(|0\rangle - |1\rangle)\right]$$

$$= \frac{1}{2}\left[(-1)^{f(0)}|0\rangle(|0\rangle - |1\rangle) + (-1) \times (-1)^{f(0)}|1\rangle(|0\rangle - |1\rangle)\right]$$

$$= \frac{1}{2}(-1)^{f(0)}\left[|0\rangle(|0\rangle - |1\rangle) - |1\rangle(|0\rangle - |1\rangle)\right]$$

$$= \pm\frac{1}{2}\left[|0\rangle(|0\rangle - |1\rangle) - |1\rangle(|0\rangle - |1\rangle)\right]$$

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

Now run the $|x\rangle$ qubit through an H gate to get $|\psi_3\rangle$

$$|\psi_3\rangle = \begin{array}{ll} \pm\frac{1}{\sqrt{2}}|0\rangle(|0\rangle - |1\rangle) & if \quad f(0) = f(1) \\ \pm\frac{1}{\sqrt{2}}|1\rangle(|0\rangle - |1\rangle) & if \quad f(0) \neq f(1) \end{array}$$

Since in our case $f(0) \oplus f(1) = 0 \Leftrightarrow f(0) = f(1)$ we can write this as

$$|\psi_3\rangle = \pm|f(0) \oplus f(1)\rangle \left[\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right]$$

Hence it is possible to measure x to find $f(0) \oplus f(1)$.

Key:
Note that $f(0) \oplus f(1)$ is a global property of $f(x)$. Classically it would require two evaluations of $f(x)$ to find this answer. Using a quantum computer we are able to evaluate both answers simultaneously and then interfere these answers to combine them together. Another more subtle point is that the phase of the result qubit transfers to the query qubit. This is a special case of phase kick back. In effect, the query qubit acts as a control of whether or not to flip the result qubit. While the result qubit is potentially flipped by the state of the query qubit, the phase of the query qubit is altered by the phase of the result (or target) qubit! We will explore this property in more detail later, since it is the key to Shor's algorithm.

6.1 Deutsch-Jozsa Algorithm

The Deutsch-Jozsa algorithm is a generalization of Deutsch's algorithm.

Suppose $f(x) : \{2^n\} \rightarrow \{0,1\}$ and that f is either constant or balanced. The goal is determine which one it is. Classically it is trivial to see that this would require (in worst case) querying just over half the solution space, or $2^n/2 + 1$ queries. The Deutsch-Jozsa algorithm answers this question with just *one* query!

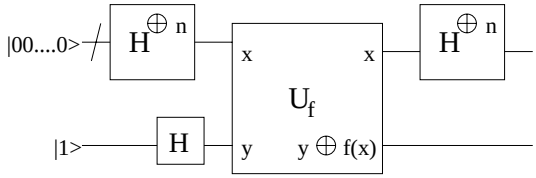

Figure 7: The Deutsch-Jozsa algorithm is simply a multi-qubit generalization of Deutsch's algorithm

The starting state of the system $|\psi_0\rangle$ is fairly straightforward

$$|\psi_0\rangle = |0\rangle^{\otimes n}|1\rangle$$

The symbolic notation $|0\rangle^{\otimes n}$ simply means n consecutive $|0\rangle$ qubits.

We then apply the $H^{\otimes n}$ transform. This symbol means to apply the H gate to each of the n qubits (in parallel, although this does not matter. The key is only that the H gate is applied once to each qubit). One way to define this transform is:

$$H^{\otimes n}|i\rangle = \sum_j \frac{(-1)^{i \cdot j}}{\sqrt{2^n}}|j\rangle$$

This notation is rather terse, but what it is saying is that given any arbitrary state vector, it will be composed of components $|i\rangle$. Each component of this state vector is transformed into a superposition of components $|j\rangle$. For example, lets examine a single qubit:

$$a|0\rangle + b|1\rangle$$

Apply $H^{\otimes 1}$ to get:

$$a\frac{(-1)^{0\cdot 0}}{\sqrt{2}}|0\rangle + a\frac{(-1)^{0\cdot 1}}{\sqrt{2}}|1\rangle + b\frac{(-1)^{1\cdot 0}}{\sqrt{2}}|0\rangle + b\frac{(-1)^{1\cdot 1}}{\sqrt{2}}|1\rangle = \frac{1}{\sqrt{2}}(a+b)|0\rangle + \frac{1}{\sqrt{2}}(a-b)|1\rangle$$

When we look at the actual transform as we have been writing it in the past we find:

$$\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\begin{bmatrix} a \\ b \end{bmatrix} = \frac{a+b}{\sqrt{2}}|0\rangle + \frac{a-b}{\sqrt{2}}|1\rangle$$

Returning back to $|\psi_0\rangle$ we transform it by:

$$|\psi_1\rangle = H^{\otimes n}|0\rangle^{\otimes n}H|1\rangle$$

$$= \sum_{x\in\{0,1\}^n} \frac{1}{\sqrt{2^n}}|x\rangle\left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right]$$

The notation $\{0,1\}^n$ means all possible bit strings of size n. For example, if $n = 2$, this would be "00", "01", "10", and "11".

We then apply the transform U_f that implements $f(x)$ to obtain the state $|\psi_2\rangle$:

$$|\psi_2\rangle = \sum_{x\in\{0,1\}^n} \frac{(-1)^{f(x)}}{\sqrt{2^n}}|x\rangle\left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right]$$

Finally we apply another $H^{\otimes n}$ transform to obtain $|\psi_3\rangle$:

$$|\psi_3\rangle = \sum_{z\in\{0,1\}^n}\sum_{x\in\{0,1\}^n} \frac{(-1)^{x\cdot z + f(x)}}{2^n}|z\rangle\left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right]$$

The key to the Deutsch-Jozsa algorithm is the rather subtle point: Observe the probability ampli-

tude of $z = |0\rangle^{\otimes n}$. Consider when $f(x)$ is constant. Since $z = |0\rangle^{\otimes n}$, we know that $(-1)^{x\cdot z + f(x)}$ is either -1 or $+1$ for all values of x (since z is equal to zero $x \cdot z$ must be equal to zero. Further $f(x)$ was constant). Hence, if $f(x)$ is constant the probability amplitude for $z = |0\rangle^{\otimes n}$ is expressed as:

$$\sum_{x \in \{0,1\}^n} \frac{1}{2^n} = \pm 1$$

Hence when you measure the query register you will obtain a zero. Since postulate one tells that the probabilities must sum to 1, if $f(x)$ is constant, then we must measure a zero.

On the other hand, lets consider if $f(x)$ is balanced. Then $(-1)^{x\cdot z + f(x)}$ will be $+1$ for some x and -1 for other x's. This is where the balanced requirement comes into play. Since all x's are considered, and the function is perfectly balanced, the probability of obtaining $z = |0\rangle^{\otimes n}$ is expressed as:

$$\sum_{x_1} \frac{+1}{2^n} + \sum_{x_2} \frac{-1}{2^n} = 0$$

Where x_1 is the set of x's such that $f(x)$ is equal to 0 and x_2 are those x's where $f(x)$ is equal to 1. Hence you will not measure 0 when $f(x)$ is balanced since the probability amplitudes perfectly destructively interfere to produce a probability of zero for that quantity.

What will be measured if the function is balanced? Anything except 0.

7 Bloch Sphere

The block sphere is a useful visualization tool for single quantum bits. Before discussing it, we need to refresh our complex math a little.

A complex number $Z = x + y \cdot i$ can be expressed in polar form:

$$x = r\cos(\theta)$$

$$y = r\sin(\theta)$$

$$Z = r(\cos(\theta) + i\sin(\theta)) = re^{i\theta}$$

$$Z^* = re^{-i\theta}$$

Suppose $|\psi\rangle = a|0\rangle + b|1\rangle$ and $|a|^2 + |b|^2 = 1$

Claim: $a = e^{i\gamma}\cos\frac{\theta}{2}$ and $b = e^{i(\gamma+\varphi)}\sin\frac{\theta}{2}$

Clearly $e^{i\gamma}\cos\frac{\theta}{2}$ is a fully general way of expressing all values of a that can still be normalized to 1. Lets examine $|a|^2$:

$$|a|^2 = e^{i\gamma}\cos\tfrac{\theta}{2}\cdot e^{-i\gamma}\cos\tfrac{\theta}{2} = \cos^2\tfrac{\theta}{2}$$

Hence, $|b|^2 = 1-\cos^2\frac{\theta}{2} = \sin^2\frac{\theta}{2}$, therefore:

$$|b|^2 = e^{i(\gamma+\varphi)}\sin\tfrac{\theta}{2}\cdot e^{-i(\gamma+\varphi)}\sin\tfrac{\theta}{2} = \sin^2\tfrac{\theta}{2}$$

Hence, another way to express an arbitrary single qubit is:

$$|\psi\rangle = e^{i\gamma}\cos\tfrac{\theta}{2}|0\rangle + e^{i(\gamma+\varphi)}\sin\tfrac{\theta}{2}|1\rangle$$

$$|\psi\rangle = e^{i\gamma}\left[\cos\tfrac{\theta}{2}|0\rangle + e^{i\varphi}\sin\tfrac{\theta}{2}|1\rangle\right]$$

Claim: $e^{i\gamma}|\psi\rangle \cong |\psi\rangle$ under measurement.

Let M be the measurement operator, then the probability of measuring M is:

$$\langle\psi|e^{-i\gamma}M^\dagger M e^{i\gamma}|\psi\rangle = \langle\psi|M^\dagger M|\psi\rangle$$

Hence, the *global phase* of a quantum bit does not matter. It is not observable, under any measurement. Therefore it is sufficient to write a quantum bit as:

$$|\psi\rangle = \cos\tfrac{\theta}{2}|0\rangle + e^{i\varphi}\sin\tfrac{\theta}{2}|1\rangle$$

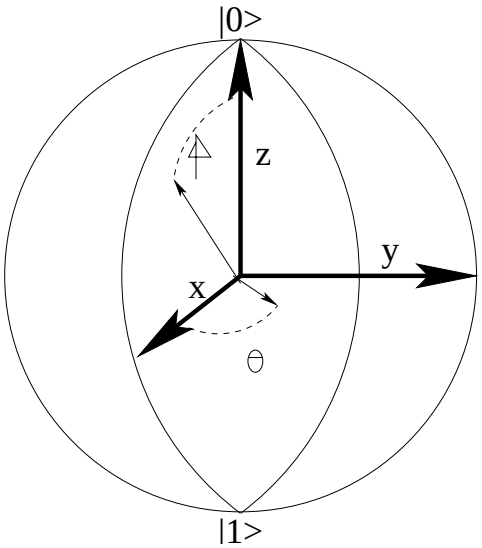

Figure 8: The Bloch sphere is a useful visualization technique for single qubits and unitary transform on them.

I refer you to Examples 4.8 and 4.9 from the book for rigorous mathematical proofs, but here note:

Any single qubit state is a point on the surface of the Bloch sphere.

Any unitary transform becomes a rotation on this sphere: $U = e^{i\gamma}R_{\hat{n}}(\theta)$. This is a rotation across some arbitrary angle $\hat{n}$ and a global phase shift. More usefully this can be broken down into:

$$U = e^{i\gamma}R_z(\beta)R_y(\varphi)R_z(\delta)$$

Rotations about the X, Y, and Z axis are related to the Pauli operators:

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \qquad Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \qquad Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

Before discussing this precisely, lets build up to the R_y rotation. Lets assume that:

$$R_y(\theta) = \begin{bmatrix} \cos\frac{\theta}{2} & -\sin\frac{\theta}{2} \\ \sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{bmatrix}$$

Recall the relations:

$$\sin(A+B) = \sin(A)\cos(B) + \cos(A)\sin(B)$$
$$\cos(A+B) = \cos(A)\cos(B) - \sin(A)\sin(B)$$

Assume a qubit in the state: $\cos\left(\frac{\varphi}{2}\right)|0\rangle + \sin\left(\frac{\varphi}{2}\right)|1\rangle$, then rotating this by $R_y(\theta)$ is:

$$\begin{bmatrix} \cos\frac{\theta}{2} & -\sin\frac{\theta}{2} \\ \sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{bmatrix} \begin{bmatrix} \cos\frac{\varphi}{2} \\ \sin\frac{\varphi}{2} \end{bmatrix}$$

$$= \begin{bmatrix} \cos(\theta/2)\cos(\varphi/2) - \sin(\theta/2)\sin(\varphi/2) \\ \sin(\theta/2)\cos(\varphi/2) + \cos(\theta/2)\sin(\varphi/2) \end{bmatrix} = \cos\left(\tfrac{\theta+\varphi}{2}\right)|0\rangle + \sin\left(\tfrac{\theta+\varphi}{2}\right)|1\rangle$$

Which is what we would expect. Decomposing $R_y(\theta)$ further:

$$R_y(\theta) = \begin{bmatrix} \cos\frac{\theta}{2} & -\sin\frac{\theta}{2} \\ \sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{bmatrix} = \begin{bmatrix} \cos\frac{\theta}{2} & 0 \\ 0 & \cos\frac{\theta}{2} \end{bmatrix} + \begin{bmatrix} 0 & -\sin\frac{\theta}{2} \\ \sin\frac{\theta}{2} & 0 \end{bmatrix}$$

$$= \cos\left(\tfrac{\theta}{2}\right)I - i\cdot\sin\left(\tfrac{\theta}{2}\right)Y = e^{-i\theta Y/2}$$

This $e^{-i\theta Y/2}$ notation relates the Pauli operators back to rotations on the Bloch sphere:

$$R_x(\theta) = e^{-i\theta X/2} = \cos\tfrac{\theta}{2}I - i\cdot sin\tfrac{\theta}{2}X =$$

$$\begin{bmatrix} \cos\frac{\theta}{2} & 0 \\ 0 & \cos\frac{\theta}{2} \end{bmatrix} - \begin{bmatrix} 0 & i\cdot\sin\frac{\theta}{2} \\ i\cdot\sin\frac{\theta}{2} & 0 \end{bmatrix} = \begin{bmatrix} \cos\frac{\theta}{2} & -i\cdot\sin\frac{\theta}{2} \\ -i\cdot\sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{bmatrix}$$

$$R_y(\theta) = e^{-i\theta Y/2} = \cos\tfrac{\theta}{2} I - i \cdot sin\tfrac{\theta}{2} Y =$$

$$\begin{bmatrix} \cos\frac{\theta}{2} & 0 \\ 0 & \cos\frac{\theta}{2} \end{bmatrix} - \begin{bmatrix} 0 & -i \cdot i \cdot \sin\frac{\theta}{2} \\ i \cdot i \cdot \sin\frac{\theta}{2} & 0 \end{bmatrix} = \begin{bmatrix} \cos\frac{\theta}{2} & -\sin\frac{\theta}{2} \\ \sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{bmatrix}$$

$$R_z(\theta) = e^{-i\theta Z/2} = \cos\tfrac{\theta}{2} I - i \cdot sin\tfrac{\theta}{2} Z =$$

$$\begin{bmatrix} \cos\frac{\theta}{2} & 0 \\ 0 & \cos\frac{\theta}{2} \end{bmatrix} - \begin{bmatrix} i \cdot \sin\frac{\theta}{2} & 0 \\ 0 & -i \cdot \sin\frac{\theta}{2} \end{bmatrix} = \begin{bmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{+i\theta/2} \end{bmatrix}$$

Claim: Any single qubit unitary transform can be decomposed into the following:

$$U = e^{-i\alpha} R_z(\beta) R_y(\gamma) R_z(\delta)$$

Consider an arbitrary transform:

$$U = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

Since U is unitary:

$$U^\dagger U = I \text{ or } \begin{bmatrix} a^* & c^* \\ b^* & d^* \end{bmatrix} \begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Which implies:

$$\begin{aligned} a^*a + c^*c &= 1 \\ b^*b + d^*d &= 1 \\ a^*b + c^*d &= 0 \\ b^*a + d^*c &= 0 \end{aligned}$$

Assuming an arbitrary form for $a = e^{-ia'}$ then $a^* = e^{ia'}\cos\left(\frac{\gamma}{2}\right)$ and

$$1 - e^{ia'}\cos\left(\tfrac{\gamma}{2}\right) e^{-ia'}\cos\left(\tfrac{\gamma}{2}\right) = c^*c$$

$$c^*c = 1 - \cos^2\tfrac{\gamma}{2}$$

Which implies that $c = e^{-ic'}\sin\left(\frac{\gamma}{2}\right)$. We can then rewrite our unitary equations:

$$\begin{aligned} &\cos^2\tfrac{\gamma}{2} + \sin^2\tfrac{\gamma}{2} = 1 \\ &b^*b + d^*d = 1 \\ &e^{ia'}\cos\left(\tfrac{\gamma}{2}\right) \cdot b + e^{ic'}\sin\left(\tfrac{\gamma}{2}\right) \cdot d = 0 \\ &b^* \cdot e^{-ia'}\cos\left(\tfrac{\gamma}{2}\right) + d^* \cdot e^{-ic'}\sin\left(\tfrac{\gamma}{2}\right) = 0 \end{aligned}$$

From the last two equations it should be clear that $b = -e^{-ib'}\sin\left(\frac{\gamma}{2}\right)$ and $d = e^{-id'}\cos\left(\frac{\gamma}{2}\right)$ (it is possible to invert where the negative sign is placed, but it is equivalent), which gives us:

$$\begin{aligned}
&\cos^2\tfrac{\gamma}{2} + \sin^2\tfrac{\gamma}{2} = 1\\
&\sin^2\tfrac{\gamma}{2} + \cos^2\tfrac{\gamma}{2} = 1\\
&-e^{ia'}\cos\left(\tfrac{\gamma}{2}\right)\cdot e^{-ib'}\sin\left(\tfrac{\gamma}{2}\right) + e^{ic'}\sin\left(\tfrac{\gamma}{2}\right)\cdot e^{-id'}\cos\left(\tfrac{\gamma}{2}\right) = 0\\
&-e^{ib'}\sin\left(\tfrac{\gamma}{2}\right)\cdot e^{-ia'}\cos\left(\tfrac{\gamma}{2}\right) + e^{id'}\cos\tfrac{\gamma}{2}\cdot e^{-ic'}\sin\left(\tfrac{\gamma}{2}\right) = 0
\end{aligned}$$

Focusing on the last two equations we have:

$$\begin{aligned}
e^{i(a'-b')}\cos\left(\tfrac{\gamma}{2}\right)\sin\left(\tfrac{\gamma}{2}\right) &= e^{i(c'-d')}\cos\left(\tfrac{\gamma}{2}\right)\sin\left(\tfrac{\gamma}{2}\right)\\
e^{i(b'-a')}\cos\left(\tfrac{\gamma}{2}\right)\sin\left(\tfrac{\gamma}{2}\right) &= e^{i(d'-c')}\cos\left(\tfrac{\gamma}{2}\right)\sin\left(\tfrac{\gamma}{2}\right)
\end{aligned}$$

Or quite simply: $a' - b' - c' = -d'$

There are many solutions with three free variables, but a clever one (for our purposes) is:

$$\begin{aligned}
a &= (-\delta - \beta)/2 - \alpha\\
b &= (\delta - \beta)/2 - \alpha\\
c &= (-\delta + \beta)/2 - \alpha\\
d &= (\delta + \beta)/2 - \alpha
\end{aligned}$$

This makes:

$$U = \begin{bmatrix} e^{-i\alpha}e^{-i\delta/2}e^{-i\beta/2}\cos(\gamma/2) & -e^{-i\alpha}e^{i\delta/2}e^{-i\beta/2}\sin(\gamma/2) \\ e^{-i\alpha}e^{-i\delta/2}e^{i\beta/2}\sin(\gamma/2) & e^{-i\alpha}e^{i\delta/2}e^{i\beta/2}\cos(\gamma/2) \end{bmatrix}$$

$$= e^{-i\alpha}\begin{bmatrix} e^{-i\delta/2}e^{-i\beta/2}\cos(\gamma/2) & -e^{i\delta/2}e^{-i\beta/2}\sin(\gamma/2) \\ e^{-i\delta/2}e^{i\beta/2}\sin(\gamma/2) & e^{i\delta/2}e^{i\beta/2}\cos(\gamma/2) \end{bmatrix}$$

$$= e^{-i\alpha}\begin{bmatrix} e^{-i\beta/2} & 0 \\ 0 & e^{i\beta/2} \end{bmatrix}\begin{bmatrix} e^{-i\delta/2}\cos(\gamma/2) & -e^{i\delta/2}\sin(\gamma/2) \\ e^{-i\delta/2}\sin(\gamma/2) & e^{i\delta/2}\cos(\gamma/2) \end{bmatrix}$$

$$= e^{-i\alpha}\begin{bmatrix} e^{-\beta/2} & 0 \\ 0 & e^{\beta/2} \end{bmatrix}\begin{bmatrix} \cos(\gamma/2) & -\sin(\gamma/2) \\ \sin(\gamma/2) & \cos(\gamma/2) \end{bmatrix}\begin{bmatrix} e^{-\delta/2} & 0 \\ 0 & e^{\delta/2} \end{bmatrix}$$

$$= e^{-i\alpha}R_z(\beta)R_y(\gamma)R_z(\delta)$$

Claim: $H \cong R_z\left(\frac{\pi}{2}\right)R_x\left(\frac{\pi}{2}\right)R_z\left(\frac{\pi}{2}\right)$

$$R_z\left(\tfrac{\pi}{2}\right) = \begin{bmatrix} e^{-i\pi/4} & 0 \\ 0 & e^{i\pi/4} \end{bmatrix} \cong e^{i\pi/4}\begin{bmatrix} e^{-i\pi/4} & 0 \\ 0 & e^{i\pi/4} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/2} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$$

$$R_x\left(\tfrac{\pi}{2}\right) = \begin{bmatrix} \cos(\pi/4) & -i\sin(\pi/4) \\ -i\sin(\pi/4) & \cos(\pi/4) \end{bmatrix} = \tfrac{1}{\sqrt{2}} \begin{bmatrix} 1 & -i \\ -i & 1 \end{bmatrix}$$

$$R_z\left(\tfrac{\pi}{2}\right) R_x\left(\tfrac{\pi}{2}\right) R_z\left(\tfrac{\pi}{2}\right) \cong \tfrac{1}{\sqrt{2}} \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix} \begin{bmatrix} 1 & -i \\ -i & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$$

$$= \tfrac{1}{\sqrt{2}} \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix} \begin{bmatrix} 1 & 1 \\ -i & i \end{bmatrix} = \tfrac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Note: is there a similar visualization for two or n qubits?

7.1 Phase traveling backwards through control operations

Using our new canonical description of a qubit we can illustrate a fundamental aspect of quantum computing, that is the notion that with controlled operations the target qubit is amplitude flipped on the basis of the control qubit, but the control qubit is phase flipped by the target. This is a key component of the Deutsch-Jozsa algorithm and as we will see shortly Shor's algorithm as well.

Consider two qubits:

$$\left[\cos\theta|0\rangle + e^{i\phi}\sin\theta|1\rangle\right]\left[\cos\sigma|0\rangle + e^{i\lambda}\sin\sigma|1\rangle\right]$$

$$= \cos\theta|0\rangle\left[\cos\sigma|0\rangle + e^{i\lambda}\sin\sigma|1\rangle\right] + e^{i\phi}\sin\theta|1\rangle\left[\cos\sigma|0\rangle + e^{i\lambda}\sin\sigma|1\rangle\right]$$

Perform a *CNot* operation:

$$= \cos\theta|0\rangle\left[\cos\sigma|0\rangle + e^{i\lambda}\sin\sigma|1\rangle\right] + e^{i\phi}\sin\theta|1\rangle\left[e^{i\lambda}\sin\sigma|0\rangle + \cos\sigma|1\rangle\right]$$

Now return this back to "normal form":

$$= \cos\theta|0\rangle\left[\cos\sigma|0\rangle + e^{i\lambda}\sin\sigma|1\rangle\right] + e^{i(\phi+\lambda)}\sin\theta|1\rangle\left[\sin\sigma|0\rangle + e^{-i\lambda}\cos\sigma|1\rangle\right]$$

Recall that:

$$\sin\theta = -\cos\left(\theta + \tfrac{\pi}{2}\right)$$
$$\cos\theta = \sin\left(\theta + \tfrac{\pi}{2}\right)$$

Thus we can write:

$$= \cos\theta|0\rangle\left[\cos\sigma|0\rangle + e^{i\lambda}\sin\sigma|1\rangle\right] + e^{i(\phi+\lambda)}\sin\theta|1\rangle\left[-\cos\left(\sigma+\tfrac{\pi}{2}\right)|0\rangle + e^{-i\lambda}\sin\left(\sigma+\tfrac{\pi}{2}\right)|1\rangle\right]$$

$$= \cos\theta|0\rangle \left[\cos\sigma|0\rangle + e^{i\lambda}\sin\sigma|1\rangle\right] - e^{i(\phi+\lambda)}\sin\theta|1\rangle \left[\cos\left(\sigma+\tfrac{\pi}{2}\right)|0\rangle - e^{-i\lambda}\sin\left(\sigma+\tfrac{\pi}{2}\right)|1\rangle\right]$$

$$= \cos\theta|0\rangle \left[\cos\sigma|0\rangle + e^{i\lambda}\sin\sigma|1\rangle\right] + e^{i(\phi+\lambda+\pi)}\sin\theta|1\rangle \left[\cos\left(\sigma+\tfrac{\pi}{2}\right)|0\rangle + e^{i(\pi-\lambda)}\sin\left(\sigma+\tfrac{\pi}{2}\right)|1\rangle\right]$$

Observe the phase shift that occurs on the first (control qubit). The phase of the target, λ, has now become part of the phase of the control.

7.2 Phaseflips versus bitflips

Observe that:

$$H = \tfrac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \tfrac{1}{\sqrt{2}}\left[\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}\right] = \tfrac{X+Z}{\sqrt{2}}$$

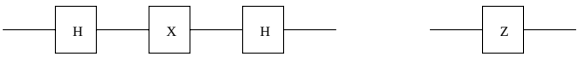

Figure 9: An X rotation surrounded by H gates is a Z rotation

Claim: $HXH = Z$

$$\left(\tfrac{X+Z}{\sqrt{2}}\right) X \left(\tfrac{X+Z}{\sqrt{2}}\right) =$$

$$\left[\tfrac{X+Z}{\sqrt{2}}\right]\left[\tfrac{X^2+XZ}{\sqrt{2}}\right] =$$

$$\left[\tfrac{X+Z}{\sqrt{2}}\right]\left[\tfrac{I+XZ}{\sqrt{2}}\right] =$$

$$\tfrac{XI+XXZ+ZI+ZXZ}{2} =$$

$$\tfrac{X+Z+Z+-X}{2} =$$

$$\tfrac{2Z}{2} = Z$$

Conversely:

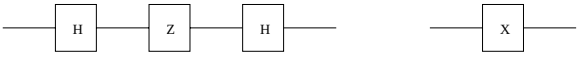

Figure 10: Conversely, a Z rotation surrounded by H gates is an X rotation

Claim: $HZH = X$

$$\left(\tfrac{X+Z}{\sqrt{2}}\right) Z \left(\tfrac{X+Z}{\sqrt{2}}\right) =$$

$$\left[\frac{X+Z}{\sqrt{2}}\right]\left[\frac{ZX+Z^2}{\sqrt{2}}\right] =$$

$$\left[\frac{X+Z}{\sqrt{2}}\right]\left[\frac{ZX+I}{\sqrt{2}}\right] =$$

$$\frac{XZX+XI+ZZX+ZI}{2} =$$

$$\frac{-Z+XI+IX+Z}{2} =$$

$$\frac{2X}{2} = X$$

8 Universal Quantum Gates

Quantum technologies will not be able to directly implement any quantum gate. Fortunately there is a method to synthesize any arbitrary quantum gate from only a small minimal set. This universal set is the quantum analog of the universal gate (NOR or NAND) for classical systems. A universal set of operations are: H, X, T, and $CNot$. In this section we will show how any single qubit gate can be implemented from H, X and T. For multiple qubit gates I refer you to pages 189-191 in your book.

The T gate is sometimes referred to as the $\pi/8$ gate. It is a rotation about the Z axis by $\pi/4$:

$$T = \pi/8 = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix} = e^{i\pi/8}\begin{bmatrix} e^{-i\pi/8} & 0 \\ 0 & e^{i\pi/8} \end{bmatrix} \cong R_z(\pi/4)$$

The essential idea with synthesizing single qubit gates is that any single qubit gate is going to take a qubit which is a point on the Bloch sphere from one position to another. While an arbitrary single qubit gate will do this with arbitrary precision, for computation purposes we can approximate the single qubit gate. So long as we can approximate it to arbitrary precision from only a basic set of gates, then in theory anyway we can synthesize the arbitrary gate. One can view this approximation as dividing up the surface of the Bloch sphere into patches, circles each of which is ε small in size.

The intuition behind approximating an arbitrary rotation is we are going to form a rotation of an irrational amount about some axis. The key is the irrationality of the amount (lets name it Δ). By being irrational we are assured that $k_1 \cdot 2\pi/\Delta \neq k_2 \cdot 2\pi/\Delta$ for all choices of $k_1 \neq -k_2$. To understand the significance of this, imagine a circle. Assuming we are somewhere on the circumference of this circle, if we move Δ amount in one direction we will *never* get back to where we started by continuing to move Δ increments in the same direction (the only way to get back is to move $-\Delta$).

Assuming we are at some point on this circle and we move Δ distance around the circumference then we will move be at some other unique point. Performing this operation again, we will be at yet another unique point. Eventually these unique points *cover* within ε distance the entire

circumference. This covering holds for the surface of the Bloch sphere as well as a circle.

So the big question is how do we move an irrational amount? Lets examine $THTH$

$$THTH = e^{-i\frac{\pi}{8}Z}e^{-i\frac{\pi}{8}X}$$

Note this is using the relation $HZH = X$. This is equal to:

$$= \left[\cos\left(\tfrac{\pi}{8}\right)I - i\cdot\sin\left(\tfrac{\pi}{8}\right)Z\right]\left[\cos\left(\tfrac{\pi}{8}\right)I - i\cdot\sin\left(\tfrac{\pi}{8}\right)X\right]$$

$$= \cos^2\left(\tfrac{\pi}{8}\right)I - i\cdot\cos\left(\tfrac{\pi}{8}\right)\sin\left(\tfrac{\pi}{8}\right)IX - i\cdot\sin\left(\tfrac{\pi}{8}\right)\cos\left(\tfrac{\pi}{8}\right)IZ + i^2\sin^2\left(\tfrac{\pi}{8}\right)ZX$$

$$= \left[\cos^2\left(\tfrac{\pi}{8}\right)I - \sin^2\left(\tfrac{\pi}{8}\right)ZX\right] - i\left[\cos\left(\tfrac{\pi}{8}\right)\sin\left(\tfrac{\pi}{8}\right)IX + \sin\left(\tfrac{\pi}{8}\right)\cos\left(\tfrac{\pi}{8}\right)IZ\right]$$

Recall that $iY = ZX$, hence:

$$= \left[\cos^2\left(\tfrac{\pi}{8}\right)I - i\cdot\sin^2\left(\tfrac{\pi}{8}\right)Y\right] - i\left[\cos\left(\tfrac{\pi}{8}\right)\sin\left(\tfrac{\pi}{8}\right)IX + \sin\left(\tfrac{\pi}{8}\right)\cos\left(\tfrac{\pi}{8}\right)IZ\right]$$

$$= \cos^2\left(\tfrac{\pi}{8}\right)I - i\cdot\sin\left(\tfrac{\pi}{8}\right)\left[\cos\left(\tfrac{\pi}{8}\right)(X+Z) + \sin\left(\tfrac{\pi}{8}\right)Y\right]$$

We are now back to our "canonical form" for a rotation, except we are rotating some amount $\cos\left(\frac{\theta}{2}\right) \equiv \cos^2\left(\frac{\pi}{8}\right)$. The key is θ is irrational.

The angle at which this rotation occurs is along the vector:

$$\hat{n} = \left(\cos\tfrac{\pi}{8}, \sin\tfrac{\pi}{8}, \cos\tfrac{\pi}{8}\right)$$

This irrational rotation of amount θ along the vector $\hat{n}$ we will denote as $R_{\hat{n}}(\theta)$. We are close to finishing, the only thing left is the ability to move along an orthogonal vector $\hat{m}$. This is accomplished using something similar to $HZH = X$. Simply rotate along $HR_{\hat{n}}(\theta)H$.

$$HR_{\hat{n}}(\theta)H = HTHTHH = HTHT = e^{-i\frac{\pi}{8}X}e^{-i\frac{\pi}{8}Z}$$

$$= \left[\cos\left(\tfrac{\pi}{8}\right)I - i\cdot\sin\left(\tfrac{\pi}{8}\right)Z\right]\left[\cos\left(\tfrac{\pi}{8}\right)I - i\cdot\sin\left(\tfrac{\pi}{8}\right)Z\right]$$

Which we note is the same except for the $\sin^2\left(\frac{\pi}{8}\right)XZ$, instead of ZX, hence $-iY$ instead of iY. Hence, a new orthogonal rotation $R_{\hat{m}}$ along the vector:

$$\hat{m} = \left(\cos\tfrac{\pi}{8}, -\sin\tfrac{\pi}{8}, \cos\tfrac{\pi}{8}\right)$$

Hence, using our decomposition idea we can approximate any rotation to arbitrary precision:

$$U = R_{\hat{n}}(\beta\cdot\theta)R_{\hat{m}}(\gamma\cdot\theta)R_{\hat{n}}(\delta\cdot\theta)$$

8.1 More than two qubit controlled operations

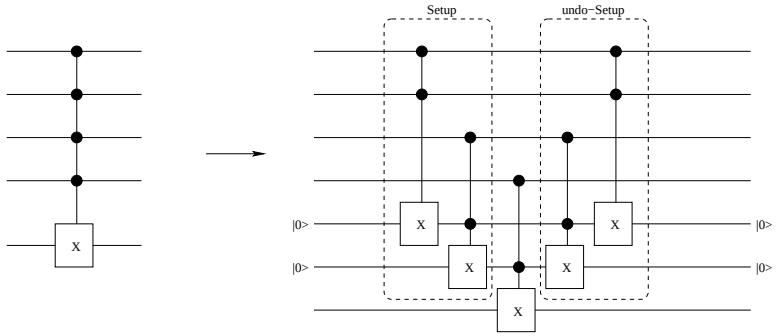

Figure 11: Generalizations of the Toffoli gate are constructed by simply composing a "super control" and then de-composing it.

8.2 Other interesting gates

Claim: The control and target of a *CNot* gate can be swapped when surrounding the input and output with H gates.

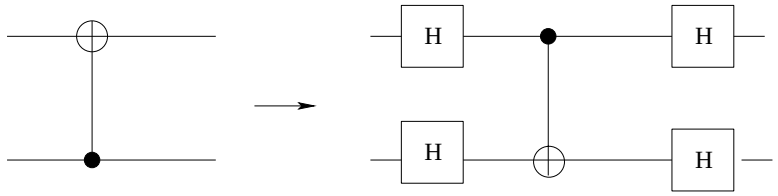

Figure 12: The control and target of a CNOT gate can be swapped when the inputs are surrounded by H gates.

Suppose two qubits:

$$|\psi_1\rangle = a|0\rangle + b|1\rangle$$
$$|\psi_2\rangle = c|0\rangle + d|1\rangle$$

$$|\psi\rangle = |\psi_1\rangle \otimes |\psi_2\rangle$$

$$= ac|00\rangle + ad|01\rangle + bc|10\rangle + bd|11\rangle$$

Now apply the two H gates:

$$\tfrac{1}{2}[ac[(|0\rangle + |1\rangle)(|0\rangle + |1\rangle)] + ad[(|0\rangle + |1\rangle)(|0\rangle - |1\rangle)]$$
$$+bc[(|0\rangle - |1\rangle)(|0\rangle + |1\rangle)] + bd[(|0\rangle - |1\rangle)(|0\rangle - |1\rangle)]]$$

$$= \tfrac{1}{2}[ac[|00\rangle + |01\rangle + |10\rangle + |11\rangle] + ad[|00\rangle - |01\rangle + |10\rangle - |11\rangle]$$
$$+bc[|00\rangle + |01\rangle - |10\rangle - |11\rangle]bd[|00\rangle - |01\rangle - |10\rangle + |11\rangle]]$$

Apply the *CNot* gate:

$$\tfrac{1}{2}[ac[|00\rangle + |01\rangle + |11\rangle + |10\rangle] + ad[|00\rangle - |01\rangle + |11\rangle - |10\rangle]$$
$$+bc[|00\rangle + |01\rangle - |11\rangle - |10\rangle]bd[|00\rangle - |01\rangle - |11\rangle + |10\rangle]]$$

$$= \tfrac{1}{2}[ac[|00\rangle + |01\rangle + |10\rangle + |11\rangle] + ad[|00\rangle - |01\rangle - |10\rangle + |11\rangle]$$
$$+bc[|00\rangle + |01\rangle - |10\rangle - |11\rangle]bd[|00\rangle - |01\rangle + |10\rangle - |11\rangle]]$$

$$= \tfrac{1}{2}[ac[(|0\rangle + |1\rangle)(|0\rangle + |1\rangle)] + ad[(|0\rangle - |1\rangle)(|0\rangle - |1\rangle)]$$
$$+bc[(|0\rangle - |1\rangle)(|0\rangle + |1\rangle)] + bd[(|0\rangle + |1\rangle)(|0\rangle - |1\rangle)]]$$

Apply the H gates:

$$ac|00\rangle + ad|11\rangle + bc|10\rangle + bd|01\rangle$$

$$= ac|00\rangle + bd|01\rangle + bc|10\rangle + ad|11\rangle$$

Which is as if the control and target qubits where swapped!

8.3 Swap

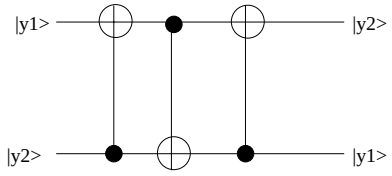

Figure 13: A swap gate exchanges the state of two qubits.

A *Swap* gate is three back to back *CNot* gates as shown in Figure 13.

Suppose two qubits:

$$\begin{bmatrix} a \\ b \end{bmatrix} \otimes \begin{bmatrix} c \\ d \end{bmatrix}$$

$$= ac|00\rangle + ad|01\rangle + bc|10\rangle + bd|11\rangle$$

Apply the first *CNot*:

$$ac|00\rangle + ad|01\rangle + bd|10\rangle + bc|11\rangle$$

Apply the second $CNot$:

$$ac|00\rangle + bc|01\rangle + bd|10\rangle + ad|11\rangle$$

Apply the third $CNot$:

$$ac|00\rangle + bc|01\rangle + ad|10\rangle + bd|11\rangle$$

$$= ca|00\rangle + cb|01\rangle + da|10\rangle + db|11\rangle$$

$$= \begin{bmatrix} c \\ d \end{bmatrix} \otimes \begin{bmatrix} a \\ b \end{bmatrix}$$

9 Shor's Algorithm

Shor's algorithm is used to factor numbers into their components (which can potentially be prime). It does this in roughly $O\left(n^3\right)$ quantum operations, while the best known classical algorithms are exponential. Since the difficulty of factoring is believed to be exponentially hard it forms the basis of most modern crypto systems. Hence, being able to factor in polynomial time on a quantum computer has attracted significant interest.

In this section we are going to begin by constructing a set of tools needed to actually implement Shor's algorithm. Then we put these tools together to actually factor. Our first tool is how factoring is related to another problem: order finding.

9.1 Factoring and order-finding

For positive x, N such that $x < N$, the order of x modulo N is the least positive integer r such that $[x^r (mod N)] = 1$

Why do we care about this problem:

Suppose $N = 77$ and $x = 10$

r:	0	1	2	3	4	5	6	7	8	9	10	11	12	13
$x^r \vert N$:	1	10	23	76	67	54	1	10	23	76	67	54	1	10

Suppose $N = 15$ and $x = 2$

r:	0	1	2	3	4	5	6	7	8	9
$x^r \vert N$:	1	2	4	8	1	2	4	8	1	2

Suppose $N = 15$ and $x = 4$

r :	0	1	2	3	4	5	6	7	8	9
$x^r \vert N$:	1	4	1	4	1	4	1	4	1	4

Suppose $N = 15$ and $x = 11$

r :	0	1	2	3	4	5	6	7	8	9
$x^r \vert N$:	1	11	1	11	1	11	1	11	1	11

Observe that the period of repetition is one less than a factor of N! Shor's algorithm can be summarized in this way:

- Choose some random x (actually x co-prime of N)
- Use quantum parallelism to compute x^r for all r simultaneously.
- Interfere all of the x^r's to obtain knowledge about a global property of the structure (i.e. the period) of the solutions.
- Use this period to find the factor of N

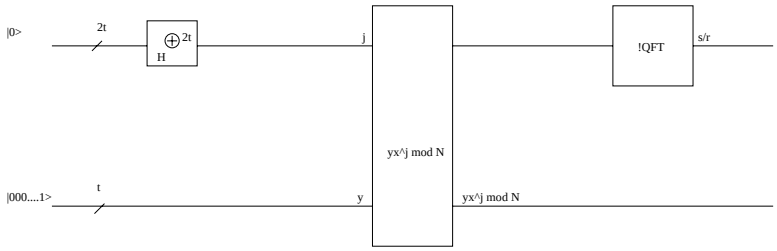

Figure 14: Overview of Shor's algorithm

9.2 Quantum Fourier Transform (QFT)

The Quantum Fourier Transform (QFT) implements the analog of the classical Fourier Transform. It transforms a state space of size 2^n from the amplitude to the frequency domain (just as the Fourier transform can be viewed as a transform from 2^n numbers into a range of size 2^n containing the frequency components from the domain.

The classical Fourier Transform is defined as:

$$y_k \equiv \frac{1}{\sqrt{2^n}} \sum_{j=0}^{2^n-1} x_j e^{2\pi i jk/2^n}$$

The QFT is similarly defined:

$$|j\rangle \longrightarrow \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{2\pi i jk/2^n} |k\rangle$$

Thus an arbitrary quantum state is transformed:

$$\sum_{j=0}^{2^n-1} x_j |j\rangle \longrightarrow \sum_{k=0}^{2^n-1} y_k |k\rangle = \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} \sum_{j=0}^{2^n-1} x_j e^{2\pi i jk/2^n} |k\rangle$$

Example:

$$|00000\rangle + |01000\rangle + |10000\rangle + |11000\rangle$$

is transformed to:

$$|00000\rangle + |00100\rangle + |01000\rangle + |011000\rangle$$
$$+|10000\rangle + |10100\rangle + |11000\rangle + |11100\rangle$$

i.e.:

0 8 16 24

is transformed to:

0 4 8 12 16 20 24 28

So how do we implement the QFT? This derivation is in the book at pages 216-219, but we will expand many of the steps and deviate from it slightly for clarity:

The transform is defined as:

$$|j\rangle \longrightarrow \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} x_j e^{2\pi i jk/2^n} |k\rangle$$

Note that j is a binary number and can be decomposed into the form:

$$j = j_1 2^{n-1} + j_2 2^{n-2} + \ldots + j_n 2^0 = \sum_{i=1}^{n} j_i 2^{n-i}$$

Similarly for k

$$k = \sum_{i=1}^{n} k_i 2^{n-i}$$

Re-express the transform as

$$\frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{2\pi i j \sum_{l=1}^{n} k_l 2^{n-l}/2^n} |k\rangle$$

Canceling the 2^n terms we have:

$$\frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{2\pi i j \sum_{l=1}^{n} k_l 2^{-l}} |k\rangle$$

Now decompose the exponent:

$$\frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{2\pi i j k_1 2^{-1}} \times e^{2\pi i j k_2 2^{-2}} \times \ldots \times e^{2\pi i j k_n 2^{-n}} |k\rangle$$

Similarly, decompose the summation:

$$\frac{1}{\sqrt{2^n}} \sum_{k_1=0}^{1} \sum_{k_2=0}^{1} \ldots \sum_{k_n=0}^{1} e^{2\pi i j k_1 2^{-1}} \times e^{2\pi i j k_2 2^{-2}} \times \ldots \times e^{2\pi i j k_n 2^{-n}} |k_1 k_2 \ldots k_n\rangle$$

Now, pull out the n'th component:

$$\frac{1}{\sqrt{2^n}} \sum_{k_1=0}^{1} \sum_{k_2=0}^{1} \ldots \sum_{k_{n-1}=0}^{1} e^{2\pi i j k_1 2^{-1}} \times e^{2\pi i j k_2 2^{-2}} \times \ldots \times e^{2\pi i j k_n 2^{-n}} |k_1 k_2 \ldots k_{n-1}\rangle \sum_{k_n=0}^{1} e^{2\pi i j k_n 2^{-n}} |k\rangle$$

Similarly for all components and we have:

$$\frac{1}{\sqrt{2^n}} \left(|0\rangle + e^{2\pi i j k_1 2^{-1}} |1\rangle\right) \left(|0\rangle + e^{2\pi i j k_2 2^{-2}} |1\rangle\right) \ldots \left(|0\rangle + e^{2\pi i j k_n 2^{-n}} |1\rangle\right)$$

Next, lets define a new notation:

$$0.j_l j_{l+1} \ldots j_m = \frac{j_l}{2} + \frac{j_{l+1}}{2^2} + \frac{j_m}{2^{m-l+1}}$$

(Suspend your skepticism on why we need this notation for a short while.) Note:

$$e^{2\pi i j 2^{-k}} = e^{2\pi i 2^{-k} \sum_{l=1}^{n} j_l 2^{n-l}} = e^{2\pi i 2^{-k} j_1 2^{n-1}} \times e^{2\pi i 2^{-k} j_2 2^{n-2}} \times \ldots e^{2\pi i 2^{-k} j_2 2^0}$$

$$= e^{2\pi i 2^{n-1-k} j_1} \times e^{2\pi i 2^{n-2-k} j_2} \times \ldots e^{2\pi i 2^{n-n-k} j_n}$$

Suppose that $j_i = 0$ then $e^{2\pi i 2^{n-i-k} j_i} = 1$

Suppose that $j_i = 1$ and $2^{n-i-k} \geq 1$ then the exponent is a multiple of $2\pi i$, hence equal to 1

Suppose that $j_i = 1$ and $2^{n-i-k} < 1$, i.e., $n-i-k < 0$ then lets look at $k = 1$

$$|0\rangle + e^{2\pi i j 2^{-1}}|1\rangle = |0\rangle + e^{2\pi i 0.j_n}|1\rangle$$

For $k = 2$:

$$|0\rangle + e^{2\pi i j 2^{-2}}|1\rangle = |0\rangle + e^{2\pi i 0.j_{n-1} j_n}|1\rangle$$

and similarly for the rest. Hence, can rewrite the transform as:

$$\frac{1}{\sqrt{2^n}} \left(|0\rangle + e^{2\pi i 0.j_n}|1\rangle\right) \left(|0\rangle + e^{2\pi i 0.j_{n-1} j_n}|1\rangle\right) \ldots \left(|0\rangle + e^{2\pi i 0.j_1 j_2 \ldots j_n}|1\rangle\right)$$

which matches what is in your book again.

To see how to actually implement this, lets look at any one of the qubits and how it should be transformed:

$$\frac{1}{\sqrt{2}} \left(|0\rangle + e^{2\pi 0.j_l \ldots j_n}|1\rangle\right)$$

Pull off the first component:

$$\frac{1}{\sqrt{2}} \left(|0\rangle + e^{2\pi i 0.j_l} \times e^{2\pi 0.j_{l-1} \ldots j_n/2}|1\rangle\right)$$

Looking at the first component only:

$$\frac{1}{\sqrt{2}} \left(|0\rangle + e^{2\pi i 0.j_l}|1\rangle\right) = \frac{1}{\sqrt{2}} \left(|0\rangle + e^{2\pi i j_l/2}|1\rangle\right) = \frac{1}{\sqrt{2}} \left(|0\rangle + (-1)^{j_l}|1\rangle\right)$$

This is just an H gate!

What about $e^{2\pi 0.j_{l-1} \ldots j_n/2}$? Use a rotation:

$$R_k = \begin{bmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^k} \end{bmatrix}$$

Note we apply this rotation conditionally on whether or not j_i is equal to 1. We do this by focusing on the least significant digit first. We want to achieve:

$$\left(|0\rangle + e^{2\pi i 0.j_1 j_2 \ldots j_n}|1\rangle\right)$$

Start with $|j_1\rangle|j_2 \ldots j_n\rangle$

Apply H to obtain:

$$\frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i 0.j_1}|1\rangle\right)|j_2\ldots j_n\rangle$$

Apply a controlled R_2 rotation to obtain:

$$\frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i 0.j_1 j_2}|1\rangle\right)|j_2\ldots j_n\rangle$$

Apply controlled R_3:

$$\frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i 0.j_1 j_2 j_3}|1\rangle\right)|j_2\ldots j_n\rangle$$

And so on to obtain:

$$\frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i 0.j_1 j_2\ldots j_n}|1\rangle\right)|j_2\ldots j_n\rangle$$

This is done similarly for the other bits j_2 then j_3, etc, and that's it. The result ends up with the bits in reverse order, but simply swap them and you have the QFT!

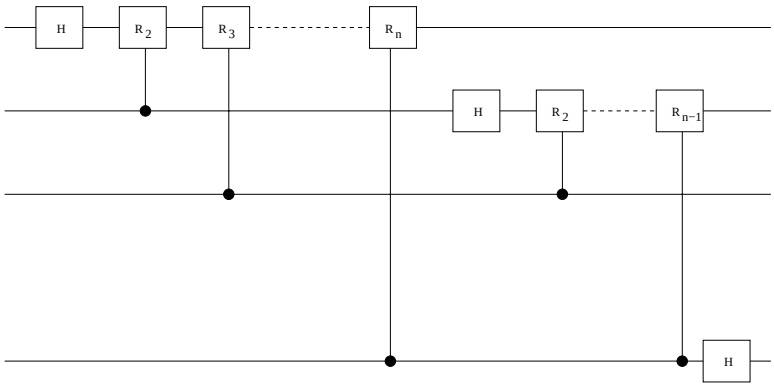

Figure 15: The above circuit implements the Quantum Fourier Transform (QFT). Note the outputs are reversed in their bit-significance.

9.3 Shor's Algorithm – the easy way

A less rigorous, but more intuitive way to think of Shor's algorithm is that you simply compute the function $yx^j mod N$ for all j. One wonders what value of y should be chosen, but an easy value, and one that makes y disappear is to choose the number 1. Then, simply fourier transform the solution space and measure the period, which is the number we are looking for. In some respects, this is exactly how Shor's algorithm works and we give that derivation here. However, a deeper understanding of Shor's algorithm comes from Phase estimation, which we'll get to next. For now, here is the straightforward description of Shor's algorithm:

$$|\psi_1\rangle = |0\rangle^{\otimes t}|00\ldots1\rangle$$

$$|\psi_2\rangle = \frac{1}{\sqrt{2^t}}\sum_j |j\rangle|00\ldots1\rangle$$

$$|\psi_3\rangle = \frac{1}{\sqrt{2^t}}\sum_j |j\rangle|x^j mod N\rangle$$

$$|\psi_4\rangle = \frac{\sqrt{r}}{\sqrt{2^t}}\sum_{j\in\{k,k+r,k+r2,\ldots\}} |j\rangle|x^k mod N\rangle, k\in[0,r-1]$$

$$|\psi_5\rangle = \frac{1}{\sqrt{r}}\sum_{s=0}^{r-1} |s/r\rangle|x^k mod N\rangle$$

$$|\psi_6\rangle = |s'/r\rangle|x^k mod N\rangle$$

9.4 Phase estimation

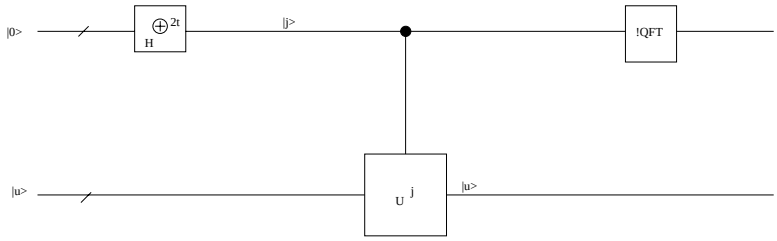

Figure 16: Phase estimation estimates the phase of a particularly formed eigenvalue of the eignvector to a matrix (in the case depicted, U).

Suppose $|u\rangle$ is an eigenvector of U with eigenvalue $e^{2\pi i\varphi}$

This means:

$$\left[e^{2\pi i\varphi}I - U\right]|u\rangle = 0 \text{ from } (\lambda I - A)x = 0$$

$$e^{2\pi i\varphi}I|u\rangle = U|u\rangle$$

Or more generally

$$e^{2\pi i 2^t\varphi}I|u\rangle = U^{2^t}|u\rangle$$

The idea with phase estimation is to apply the controlled U^{2^k} operation with $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ as the control, for $k = \{0,1,\ldots,n\}$. Then apply inverse QFT to n qubits and obtain $|\tilde{\varphi}\rangle$, an estimate of φ. In summary:

Start with: $|0\rangle^{\otimes t}|u\rangle$

Apply $H^{\otimes t}$ to get:

$$\frac{1}{\sqrt{2^t}}\sum_j |j\rangle|u\rangle$$

Apply controlled-U^{2^k} operation for each of the k qubits of j

$$\frac{1}{\sqrt{2^t}}\sum_j e^{2\pi i j\varphi_u}|j\rangle|u\rangle$$

Apply inverse QFT:

$$|\tilde{\varphi}\rangle|u\rangle$$

9.5 Shor's Algorithm – Phase estimation method

- Use phase estimation on U where $U|y\rangle = |xy(modN)\rangle$
- find $|u_s\rangle$
- Estimate phase s/r
- Use continued fraction expansion to find r

Given $U|y\rangle \equiv |xy(modN)\rangle$ claim that $|u_s\rangle$, the eigenvectors of U are:

$$|u_s\rangle \equiv \frac{1}{\sqrt{r}}\sum_{k=0}^{r-1} e^{-2\pi i\frac{s}{r}k}|x^k modN\rangle$$

Suppose $e^{2\pi i\frac{s}{r}}$ for $0 \leq s \leq r-1$ are eigenvalues, then:

$$U|u_s\rangle = e^{2\pi i\frac{s}{r}}|u_s\rangle$$

$$e^{2\pi i\frac{s}{r}}\frac{1}{\sqrt{r}}\sum_{k=0}^{r-1} e^{-2\pi i\frac{s}{r}k}|x^k modN\rangle$$

$$= \frac{1}{\sqrt{r}}\sum_{k=0}^{r-1} e^{-2\pi i\frac{s}{r}(k-1)}|x^k modN\rangle$$

$$= \frac{1}{\sqrt{r}}\sum_{k=0}^{r-1} e^{-2\pi i\frac{s}{r}k}|x^{k+1} modN\rangle$$

$$\cong \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} e^{-2\pi i \frac{s}{r} k} |x^k mod N\rangle$$

$$U|u_s\rangle = \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} e^{-2\pi i \frac{s}{r} k} U|x^k mod N\rangle$$

$$= \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} e^{-2\pi i \frac{s}{r} k} |x^{k+1} mod N\rangle$$

$$\cong \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} e^{-2\pi i \frac{s}{r} k} |x^k mod N\rangle$$

So how to prepare $|u_s\rangle$ for Shor's algorithm? We can't for a single eigenvalue, but note, we can for a superposition of them!

$$\frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} |u_s\rangle$$

$$= \frac{1}{\sqrt{r}} \frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} \sum_{k=0}^{r-1} e^{-2\pi i \frac{s}{r} k} |x^k mod N\rangle$$

Look carefully at $k = 0$

$$\frac{1}{r} \sum_{s=0}^{r-1} e^0 |x^0 mod N\rangle = \frac{r}{r} |00\ldots1\rangle = |00\ldots1\rangle$$

So, Shor's algorithm starts out with the state:

$$|0\rangle^{\otimes t} \frac{1}{\sqrt{r}} \frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} \sum_{k=0}^{r-1} e^{-2\pi i \frac{s}{r} k} |x^k mod N\rangle = |0\rangle^{\otimes t} |00\ldots1\rangle$$

The $H^{\otimes t}$ transform is applied:

$$\frac{1}{\sqrt{2^t}} \sum_j |j\rangle \frac{1}{\sqrt{r}} \frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} \sum_{k=0}^{r-1} e^{-2\pi i \frac{s}{r} k} |x^k mod N\rangle$$

Modular exponentiation is applied:

$$\frac{1}{\sqrt{2^t}} \sum_j |j\rangle \frac{1}{\sqrt{r}} \frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} \sum_{k=0}^{r-1} e^{-2\pi i \frac{s}{r} j + (-2\pi i \frac{s}{r} k)} |x^k mod N\rangle$$

Measure the second register to obtain some k

$$\frac{1}{\sqrt{2^t}}\sum_j |j\rangle \frac{1}{\sqrt{r}}\sum_{s=0}^{r-1} e^{-2\pi i \frac{s}{r}(j+k)}|x^k mod N\rangle$$

$$= \frac{1}{\sqrt{2^t}}\sum_j \frac{1}{\sqrt{r}}\sum_{s=0}^{r-1} e^{-2\pi i \frac{s}{r}(j+k)}|j\rangle|x^k mod N\rangle$$

Inverse QFT to obtain:

$$= \frac{1}{\sqrt{r}}\sum_{s=0}^{r-1} |\widetilde{s/r}\rangle|x^k mod N\rangle$$

Measure first register to obtain a particular $\widetilde{s/r}$, and then use continued fraction expansion to find r.

9.6 Continuous fraction expansion

(See page 230 from the book)

$$[a_0, a_1, \ldots, a_m] = a_0 + \frac{1}{a_1 + \frac{1}{a_2 + \frac{1}{\ddots}}}$$

Examples:

$$\frac{13}{64} = \frac{1}{\frac{64}{13}} = \frac{1}{4+\frac{12}{13}} = \frac{1}{4+\frac{1}{\frac{13}{12}}} = \frac{1}{4+\frac{1}{1+\frac{1}{12}}} \approx \frac{1}{5}$$

$$\frac{18}{64} = \frac{1}{\frac{64}{18}} = \frac{1}{3+\frac{10}{18}} = \frac{1}{3+\frac{1}{\frac{18}{10}}} = \frac{1}{3+\frac{1}{1+\frac{8}{10}}} = \frac{1}{3+\frac{1}{1+\frac{1}{\frac{10}{8}}}} = \frac{1}{3+\frac{1}{1+\frac{1}{1+\frac{2}{8}}}} \approx \frac{2}{7}$$

9.7 Modular Exponentiation

$$|z\rangle|y\rangle \longrightarrow |z\rangle|x^z y mod N\rangle$$

Note that:

$$Z = z_t 2^{t-1} + z_{t-1}2^{t-2} + \ldots + z_1 2^0$$

$$x^Z = x^{z_t 2^{t-1}} \times x^{z_{t-1}2^{t-2}} \times \ldots \times x^Z = x^{z_1 2^0}$$

$$\left[x^{2^{t-1}}\right]^{z_t} \times \left[x^{2^{t-2}}\right]^{z_{t-1}} \times \ldots \times \left[x^{2^0}\right]^{z_0}$$

Can compute x^{2^i} classically. Note that z_i is binary, hence this is really a sequence of conditional multiplies, almost. The tricky bit is mod, but that distributes, i.e.:

$$[(iN+\alpha)k] \, mod N = [iN \cdot k + \alpha \cdot k] \, mod N = \alpha k \, mod N$$

Thus, compute:

$$\left[x^{2^{t-1}} mod N\right]^{z_t} \times \left[x^{2^{t-2}} mod N\right]^{z_{t-1}} \times \ldots \times \left[x^{2^0} mod N\right]^{z_0} mod N$$

Note that this is only t conditional modular multiplications.

10 Grover's Algorithm

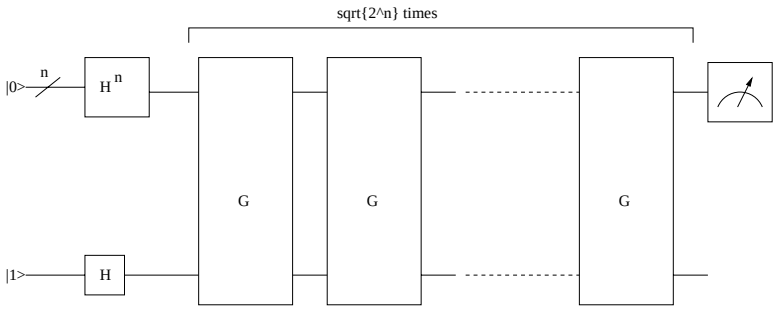

Figure 17: Grover's algorithm

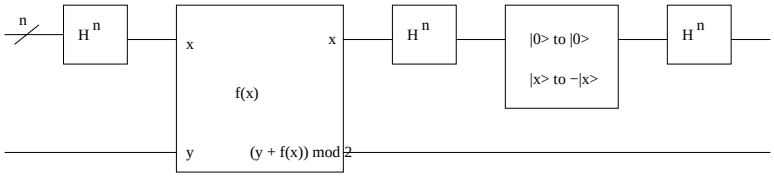

Figure 18: A Grover iteration

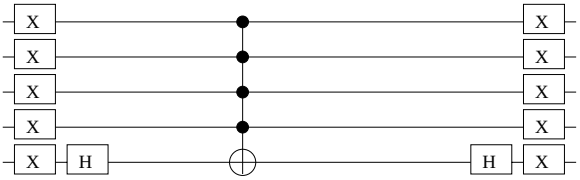

Figure 19: A Grover operator taking $|0\rangle \longrightarrow |0\rangle$ and $|x\rangle \longrightarrow -|x\rangle$ for $x \neq 0$. Note, this actually does the reverse, but that is the same up to a global phase of -1, which isn't observable.

Examine the first iteration of Grover's algorithm:

Apply H gates:

$$|0\rangle^{\otimes n}|1\rangle \longrightarrow \frac{1}{\sqrt{2^n}}\sum_j |j\rangle \frac{(|0\rangle-|1\rangle)}{\frac{1}{\sqrt{2}}}$$

Apply oracle:

$$\longrightarrow \frac{1}{\sqrt{2^n}}\sum_j (-1)^{f(j)}|j\rangle \frac{(|0\rangle-|1\rangle)}{\frac{1}{\sqrt{2}}}$$

$$= \frac{1}{\sqrt{2^n}}\left[\sum_{j\neq m} |j\rangle - |m\rangle\right] \frac{(|0\rangle-|1\rangle)}{\frac{1}{\sqrt{2}}}$$

$$= \frac{1}{\sqrt{2^n}}\left[\sum_j |j\rangle - 2|m\rangle\right] \frac{(|0\rangle-|1\rangle)}{\frac{1}{\sqrt{2}}}$$

Apply $H^{\otimes n}$

$$\longrightarrow \left[|0\rangle^{\otimes n} - \frac{1}{\sqrt{2^n}}\frac{1}{\sqrt{2^n}}2\sum_j (-1)^{m\cdot j}|j\rangle\right] \frac{(|0\rangle-|1\rangle)}{\frac{1}{\sqrt{2}}}$$

$$= \left[\left(1-\frac{2}{2^n}\right)|0\rangle^{\otimes n} - \frac{2}{2^n}\sum_{j\neq 0}(-1)^{m\cdot j}|j\rangle\right] \frac{(|0\rangle-|1\rangle)}{\frac{1}{\sqrt{2}}}$$

Apply Grover operator transforming $|0\rangle \longrightarrow |0\rangle$ and $|x\rangle \longrightarrow -|x\rangle$

$$\longrightarrow \left[\left(1-\frac{2}{2^n}\right)|0\rangle^{\otimes n} + \frac{2}{2^n}\sum_{j\neq 0}(-1)^{m\cdot j}|j\rangle\right] \frac{(|0\rangle-|1\rangle)}{\frac{1}{\sqrt{2}}}$$

$$= \left[\left(\frac{2^n-4}{2^n}\right)|0\rangle^{\otimes n} + \frac{2}{2^n}\sum_j (-1)^{m\cdot j}|j\rangle\right] \frac{(|0\rangle-|1\rangle)}{\frac{1}{\sqrt{2}}}$$

Apply $H^{\otimes n}$

$$= \left[\left(\frac{2^n-4}{2^n}\right)\sum_j |j\rangle + \frac{2}{2^n}|m\rangle\right] \frac{(|0\rangle-|1\rangle)}{\frac{1}{\sqrt{2}}}$$

So observe that the "other" part has a slightly lower probability, and the correct answer (m) has a slightly more likely chance of being measured. Lets generalize this for an arbitrary iteration.

$$\frac{W}{\sqrt{2^n}}\sum_j |j\rangle + \frac{R}{\sqrt{2^n}}|m\rangle$$

Apply oracle:

$$\longrightarrow \frac{W}{\sqrt{2^n}}\sum_j (-1)^{f(j)}|j\rangle - \frac{R}{\sqrt{2^n}}|m\rangle$$

$$= \frac{W}{\sqrt{2^n}}\sum_j |j\rangle - \frac{R+2W}{\sqrt{2^n}}|m\rangle$$

Apply $H^{\otimes n}$

$$\longrightarrow W|0\rangle - \frac{R+2W}{\sqrt{2^n}}\frac{1}{\sqrt{2^n}}\sum_j (-1)^{j\cdot m}|j\rangle$$

Apply Grover operator:

$$\longrightarrow W - 1\left[\frac{R+2W}{2^n}\right]|0\rangle + \frac{R+2W}{\sqrt{2^n}}\frac{1}{\sqrt{2^n}}\sum_{j\neq 0}(-1)^{j\cdot m}|j\rangle$$

$$= W - 2\left[\frac{R+2W}{2^n}\right]|0\rangle + \frac{R+2W}{\sqrt{2^n}}\frac{1}{\sqrt{2^n}}\sum_j (-1)^{j\cdot m}|j\rangle$$

Apply $H^{\otimes n}$

$$\longrightarrow W - 2\left[\frac{R+2W}{2^n}\right]\frac{1}{\sqrt{2^n}}\sum_j |j\rangle + \frac{R+2W}{2^n}|m\rangle$$

Thus, for each iteration the probability of obtaining a random answer "W" changes from:

$$W_i = W_{i-1} - 2\left[\frac{R_{i-1}+2W_{i-1}}{2^n}\right]$$

and the probability of obtaining the correct answer changes from:

$$R_i = R_{i-1} + 2W_{i-1}$$

(these probability amplitudes are all over $\sqrt{2^n}$

Note that the sequence $\{W_1, W_2, \ldots, W_n\}$ is non-increasing and the seqeuence $\{R_1, R_2, \ldots, R_n\}$ is non-decreasing. This (and the above equation) implies that the sequence $\{R_1, R_2 - R_1, \ldots, R_n - R_{n-1}\}$ is non-increasing. We can use this fact to prove the correctness of Grover's algorithm.

Suppose that $W_{i-1} \geq 1/2$ after $\sqrt{2^n}$ iterations. Then:

$$R_i - R_{i-1} = 2 \times W_{i-1} \geq 2 \times 1/2 = 1$$

Hence, $W_{\sqrt{2^n}} \geq \sqrt{2^n}$, hence we must measure the correct answer, which is a direct contradiction to $W_{i-1} \geq 1/2$! Therefore after $\sqrt{2^n}$ iterations $W_{i-1} < 1/2$. This means that probability of obtaining the wrong answer is less than $1/4$, which means with 75% chance we will measure the correct answer m after $\sqrt{2^n}$ iterations.

11 Error Correction

Let's suppose there was no theory of error correction for quantum computers. Then a single quantum bit de-coheres (randomizes) at the same rate as a classical bit: $e^{-\lambda t}$, except for quantum systems they tend to be so error prone that λ is quite large.

Given a system if n qubits the probability that there is no error in the entire system is the probability of no-error on each qubit multiplied together. That is:

$$e^{-\lambda t} \times e^{-\lambda t} \times \ldots \times e^{-\lambda t} = e^{-\lambda t \cdot n}$$

As an example of how faulty quantum systems are, lets compute the largest number of bits we can factor using Shor's algorithm without error correction.

Discounting the inverse QFT, Shor's algorithm is roughly $t = 65n^3$ complexity, when factoring n qubits. Suppose $\varepsilon = 0.05$ is the probability of failure that we want (this says that with 5% chance we will fail, but that is acceptable since if we fail we will just measure random data, which we can check and then re-run the algorithm anew). This implies that:

$$1 - \varepsilon > e^{-\lambda t \cdot n}$$

Let $p(n) = t \cdot n$ ($p(n)$ is the space-time complexity of an algorithm)

$$1 - \varepsilon > e^{-\lambda \cdot p(n)}$$

An appropriate value of λ is 10^{-6}, this is far to aggressive for current demonstration systems (which are more like 10^{-3}), but is around what we expect future quantum systems to be. Furthermore, it is above a critical threshold – more on this later. So for $\lambda = 10^{-6}$, how many bits can we factor with Shor's algorithm?

$$1 - \varepsilon > e^{-\lambda \cdot p(n)}$$

$$log(1 - \varepsilon) = -\lambda p(n)$$

$$65n^3 n = p(n) = -log(1 - \varepsilon)/\lambda$$

$$\longrightarrow n = \left[\left(log(1-0.05)/10^{-6} \right) /65 \right]^{\frac{1}{4}} = [342]^{\frac{1}{4}} \approx 4 \text{ qubits!}$$

Clearly, this is not good. Fortunately, there are methods to apply essentially classical error correction techniques to quantum systems.

Error correction overview:

- Errors are caused by decoherence (entanglement with the environment).
- We "fight" this entanglement with more entanglement.
- Create codes that allow for the measurement of error, without the measurement of the value. I view this as information loss, we create codes whereby we can gain knowledge about the error, but not the state, and thereby we "leak" the knowledge about the errors out of the system.
- Based on the ideas of classical codes (well almost). Classical codes such as TMR, i.e., $0 = 000$ and $1 = 111$. Just repeat the classical bit and then take a majority vote.

11.1 Shor's 3 qubit bit-flip code

Shor's code for protecting against bit-flip errors is the most basic of quantum codes and is a direct translation of TMR. That is, given a qubit:

$$|\psi\rangle = a|0\rangle + b|1\rangle$$

encode it to be:

$$|\psi_c\rangle = a|000\rangle + b|111\rangle$$

To see how to correct errors on this code, lets write it as:

$$|\psi_c\rangle = a|z_1 z_2 z_3\rangle + b|\bar{z_1}\bar{z_2}\bar{z_3}\rangle$$

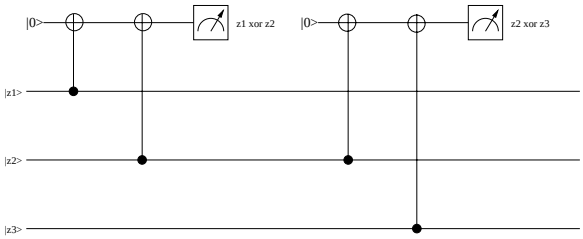

Figure 20: Measurement of errors in the 3 qubit bit flip code.

Error correction is based around the idea of using an ancilla set of qubit(s) that are entangled with the code word. The ancilla is carefully entangled in such a way that it's state is entangled with the error on the qubits (if there is any) and not the state. To see how to do this, lets return to the classical TMR codes again.

Suppose we have a classical code 010. Clearly majority voting would lead us to conclude that the code should really be 000. However, lets arrive at this result using the restriction that we cannot gain knowledge about the logical state of the code. That is, lets arrive not at the knowledge that the code should be 000, but that only there is a bit-flip error on the 2nd qubit. Thus we won't be able to know whether it is 010 or 101. This knowledge, while incomplete, does tell us where the error is and how to correct it. We can use it to know whether to apply a NOT gate to the 2nd's qubit. To gain this knowledge (and this knowledge only!), measure the following:

$s_1 = z_1 \oplus z_2$ (that is $s_1 = z_1$ *xor* z_2)

$s_2 = z_2 \oplus z_3$

Lets examine the outcome possibilities:

code-word	s_1	s_2
000	0	0
001	0	1
010	1	1
011	1	0
100	1	0
101	1	1
110	0	1
111	0	0

Focus in on the 010 and 101 code words:

code-word	s_1	s_2
010	1	1
101	1	1

There in lies the key: s_1 and s_2 do not tell us whether the code word should be 000 or 111, but they do tell us that the 2nd bit has been flipped.

This is the key concept behind quantum error correction (one of two). We devise a measurement that gives us partial knowledge, chiefly in this case, the xor. Here is how to do this in the quantum world:

Start with a code word:

$$|\psi_c\rangle = a|z_1 z_2 z_3\rangle + b|\bar{z_1}\bar{z_2}\bar{z_3}\rangle$$

And use an ancilla qubit:

$$[a|z_1z_2z_3\rangle + b|\bar{z_1}\bar{z_2}\bar{z_3}\rangle]\,|0\rangle = a|z_1z_2z_3\rangle|0\rangle + b|\bar{z_1}\bar{z_2}\bar{z_3}\rangle|0\rangle$$

Apply a CNot with the code-word bit 1 as the control and the ancilla the target.

$$\longrightarrow a|z_1z_2z_3\rangle|z_1\rangle + b|\bar{z_1}\bar{z_2}\bar{z_3}\rangle|\bar{z_1}\rangle$$

Apply a 2nd CNot with the code-word bit 2 as the control and the ancilla the target.

$$\longrightarrow a|z_1z_2z_3\rangle|z_1 \oplus z_2\rangle + b|\bar{z_1}\bar{z_2}\bar{z_3}\rangle|\bar{z_1} \oplus \bar{z_2}\rangle$$

Suppose that $z_1 = z_2$ (thus $\bar{z_1} = \bar{z_2}$), then:

$$z_1 \oplus z_2 = 0 \text{ and } \bar{z_1} \oplus \bar{z_2} = 0$$

Suppose that $z_1 \neq z_2$ (thus, $\bar{z_1} \neq \bar{z_2}$), then:

$$z_1 \oplus z_2 = 1 \text{ and } \bar{z_1} \oplus \bar{z_2} = 1$$

Hence, the ancilla qubit above does not gain knowledge on whether z_1 or z_2 is equal to 1 or 0, but it does gain knowledge about the parity of z_1 and z_2. Doing the same for bits z_2 and z_3 we have:

$\|z_1 \oplus z_2\rangle$	$\|z_2 \oplus z_3\rangle$	error ?
0	0	None
0	1	qubit 3 has flipped
1	0	qubit 1 has flipped
1	1	qubit 2 has flipped

This is understating the power of quantum error correction. Unlike classical bits, quantum bits can be in a continuous range between 0 and 1. Furthermore, so are the errors! The true power of quantum error correction is that it transform continuous errors to either no error at all, or a discrete error. Here's an example of how:

Suppose $|\psi_c\rangle = a|000\rangle + b|111\rangle$ and some random rotation about the x-axis $R_x(\theta)$ occurs on the third qubit. This is expressed as:

$$\longrightarrow a|0\rangle|0\rangle R_x(\theta)|0\rangle + b|1\rangle|1\rangle R_x(\theta)|1\rangle$$

Recall that:

$$R_x(\theta) = \begin{bmatrix} \cos\frac{\theta}{2} & -i\cdot\sin\frac{\theta}{2} \\ -i\cdot\sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{bmatrix}$$

Thus,

$$= a|0\rangle|0\rangle\left[\cos\tfrac{\theta}{2}|0\rangle - i\sin\tfrac{\theta}{2}|1\rangle\right] + b|1\rangle|1\rangle\left[-i\sin\tfrac{\theta}{2}|0\rangle + \cos\tfrac{\theta}{2}|1\rangle\right]$$

Now parity check the 2nd and 3rd qubits. First add the ancilla qubit:

$$a|0\rangle|0\rangle\left[\cos\tfrac{\theta}{2}|0\rangle|0\rangle - i\sin\tfrac{\theta}{2}|1\rangle|0\rangle\right] + b|1\rangle|1\rangle\left[-i\sin\tfrac{\theta}{2}|0\rangle|0\rangle + \cos\tfrac{\theta}{2}|1\rangle|0\rangle\right]$$

Apply the first CNot:

$$\longrightarrow a|0\rangle|0\rangle\left[\cos\tfrac{\theta}{2}|0\rangle|0\rangle - i\sin\tfrac{\theta}{2}|1\rangle|0\rangle\right] + b|1\rangle|1\rangle\left[-i\sin\tfrac{\theta}{2}|0\rangle|1\rangle + \cos\tfrac{\theta}{2}|1\rangle|1\rangle\right]$$

Apply the 2nd CNot:

$$\longrightarrow a|0\rangle|0\rangle\left[\cos\tfrac{\theta}{2}|0\rangle|0\rangle - i\sin\tfrac{\theta}{2}|1\rangle|1\rangle\right] + b|1\rangle|1\rangle\left[-i\sin\tfrac{\theta}{2}|0\rangle|1\rangle + \cos\tfrac{\theta}{2}|1\rangle|0\rangle\right]$$

Measure the ancilla qubit. What is the probability of measuring 0?

$$p(0) = a^2\cos^2\tfrac{\theta}{2} + b^2\cos^2\tfrac{\theta}{2} = \left[a^2+b^2\right]\cos^2\tfrac{\theta}{2} = \cos^2\tfrac{\theta}{2}$$

Similarly for 1:

$$p(1) = a^2\sin^2\tfrac{\theta}{2} + b^2\sin^2\tfrac{\theta}{2} = \left[a^2+b^2\right]\sin^2\tfrac{\theta}{2} = \sin^2\tfrac{\theta}{2}$$

It doesn't matter which one we pick, but lets choose 0, after measurement the qubits are re-normalized to become:

$$\frac{a\cos\frac{\theta}{2}|000\rangle|0\rangle + b\cos\frac{\theta}{2}|111\rangle|0\rangle}{\sqrt{\cos^2\frac{\theta}{2}}} = \left[a|000\rangle + b|111\rangle\right]|0\rangle$$

Key:
The very fact of measuring a continuous error has made it discrete, and in this case has made the error go away! Why did it go away? Think of it this way, the error was some superposition / entanglement with the environment, and measurement has clasped that entanglement and superposition. Alternatively, just think of the error as additional "information" in the state vector and measurement has observed (and made classical) that information. The fact that that information was not in a single qubit, but in a joint system of the two qubits is subtle.

11.2 Protecting phase

Quantum bits have a phase as well as an amplitude. This too must be protected. However, recall that the phase flip Z surrounded by two H gates is a bit flip X. More generally, a phase rotation surrounded by two H gates is an amplitude rotation. Thus a phase error is just a bit-flip error if the code is setup to protect for bit flips but then the qubits of the code are passed through H gates prior

to use. Starting with the 3 bit Shor bit flip code we can derive the 3 bit Shor phase flip code in this way. The encoder is shown in Figure 21

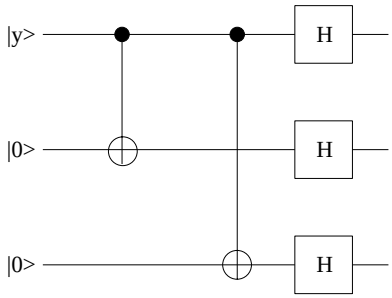

Figure 21: Encoder that takes an arbitrary state $|\psi\rangle$ and encodes it in the 3 qubit Shor phase-flip protecting code

However we need to correct for both phase and amplitude to be fully fault tolerant. For this, we need an additional tool. The tool we use is concatenation. This is exactly like the classical use of concatenation. For example, suppose we have the classical bit "1", and we TMR encode this to be "111". This can protect for one of the bits being in error. However, suppose the error rate is so high that it is possible for two bits to be in error. What we can do is take each bit of the "1-1-1" and re-encode that with TMR to obtain: "111-111-111". Then suppose we have the value "111-001-111". We first perform error correction on the lowest layer to obtain "111-000-111". Next we perform error correction on the upper layer (in logical space) to obtain "111-111-111". This is not the most efficient classical code, but it is easy to reason and work with. It can also be translated directly into quantum codes.

To protect for both phase and amplitude on a qubit we first encode it with a bit-flip code, and pass those qubits through H gates. This protects the upper layer for phase. We then re-encode each of those qubits again with a bit flip code. This protects the lower layer for amplitude. All told, the entire code (known as the Shor 9 qubit code, shown in Figure 22) protects for a single phase or a single amplitude error on any of the 9 qubits and logically holds one 1 quantum bit. Figure 23 illustrates how to measure the error in a phase.

11.3 7 Qubit Steane code

Now that we have the 9 qubit Shor code we can protect quantum state. But we also want compute on it. To compute in a fault tolerant manner we cannot decode the code word (which we'll call the logical qubit) into a single qubit (which we'll call a physical qubit), transform it, and then re-encode it. The reason is the physical qubit is susceptible to errors, and if an error does occur to it, in this unencoded state, then the computation is lost. The solution is to transform a basic set of quantum operations that would ordinarily be applied to physical qubits such that they can be applied to logical qubits in a fault tolerant manner.

Doing this for the Shor code is not easy. Logical X and Z operations can be performed simply

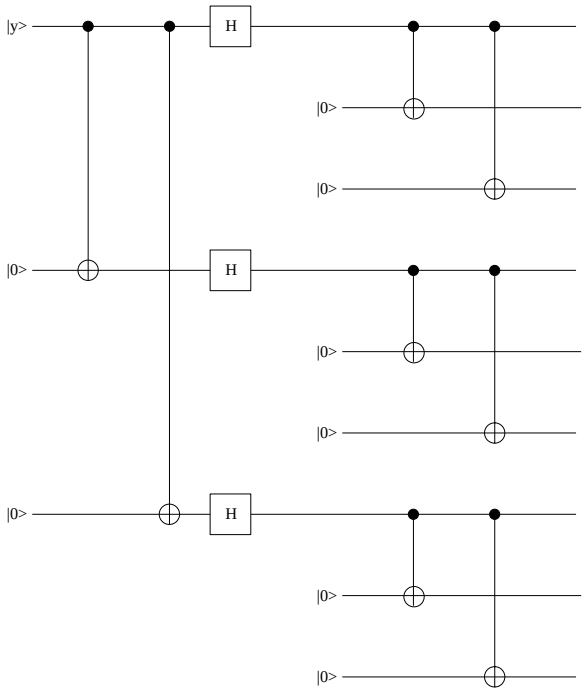

Figure 22: Encoder that takes an arbitrary state $|\psi\rangle$ and encodes it in the 9 qubit Shor code.

(challenge: what are the logical implementations of these?), but logical H, CNOT, and T cannot. A vast amount of quantum coding research is out there, but one of the easiest to transform codes is the 7 qubit Steane code.

To introduce the 7 qubit Steane code we'll take a brief detour into stabilizer codes. Stabilizer codes and their associated stabilizers is a neat compact theory useful for describing quantum codes.

Here is the idea with stabilizer codes. Suppose:

$$|\psi\rangle = \frac{|00\rangle+|11\rangle}{\frac{1}{\sqrt{2}}}$$

Then $X_1X_2|\psi\rangle = |\psi\rangle$ and $Z_1Z_2|\psi\rangle = |\psi\rangle$. Thus we say that $|\psi\rangle$ is stabilized by X_1X_2 and Z_1Z_2 (here X_1 means the X gate applied to the first qubit). Note that X_1X_2 is only a stabilizer for this particular $|\psi\rangle$, and does not generalize to arbitrary $a|00\rangle + b|11\rangle$.

Thus, the 3 qubit bit-flip Shor code is stabilized by Z_1Z_2 and Z_2Z_3. The 3 qubit phase-flip code is stabilized by X_1X_2 and X_2X_3. What is nifty about the stabilizer formalism is that they indicate precisely how to measure the error. To measure the error in a logical quantum bit one measures the stabilizers.

The stabilizers for the 7 qubit Steane code are:

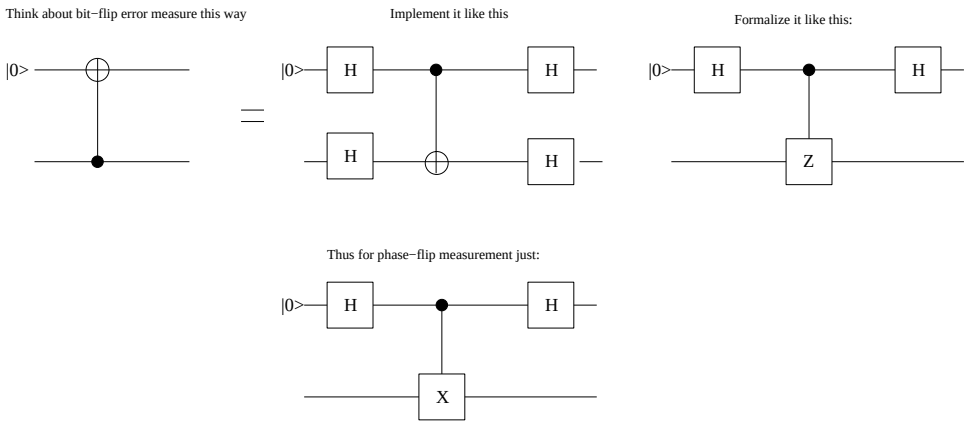

Figure 23: Recall that to measure the amplitude error we performed two CNOT operations on an ancilla qubit initially in the $|0\rangle$ state. To implement this, however, we surrounded both the ancilla and the code bits with H gates and inverted the direction of the CNOT gate. This was so that an phase error on the ancilla did not pollute the qubits in the code word. To formalize this thinking we thought of this as a controlled Z operation with the ancilla as the control. To measure the phase error we just directly implement the formalism, which is to perform a controlled X (i.e. CNOT) operation.

g_1	I_1	I_2	I_3	X_4	X_5	X_6	X_7
g_2	I_1	X_2	X_3	I_4	I_5	X_6	X_7
g_3	X_1	I_2	X_3	I_4	X_5	I_6	X_7
g_4	I_1	I_2	I_3	Z_4	Z_5	Z_6	Z_7
g_5	I_1	Z_2	Z_3	I_4	I_5	Z_6	Z_7
g_6	Z_1	I_2	Z_3	I_4	Z_5	I_6	Z_7

The Steane code is particularly nice because once we measure these stabilizers if any stabilizer $\{g_1, g_2, g_3\}$ is non-zero then a phase error has occurred on the physical qubit that is part of the logical qubit at position $g_1 2^2 + g_2 2^1 + g_3 2^0 - 1$ and if any stabilizer $\{g_4, g_5, g_6\}$ is non-zero then a bit-flip error has occurred on the physical qubit $g_4 2^2 + g_5 2^1 + g_6 2^0 - 1$.

Figure 24 depicts how to (non-fault tolerantly) encode a physical qubit into the Steane code. This figure is from John Preskill at Caltech and is not in your book.

The important part about the 7 qubit Steane code is that the H, X, Z, and CNOT gates can be applied in logical form transversally. That is, bit-wise across the qubits of the code. Unfortunately, T cannot. See pages 485-491 of the book for details on how to implement T.

11.4 Recursive error correction and the threshold theorem

How fault tolerant is the Steane code? For this lets perform a calculation. Assume the following:

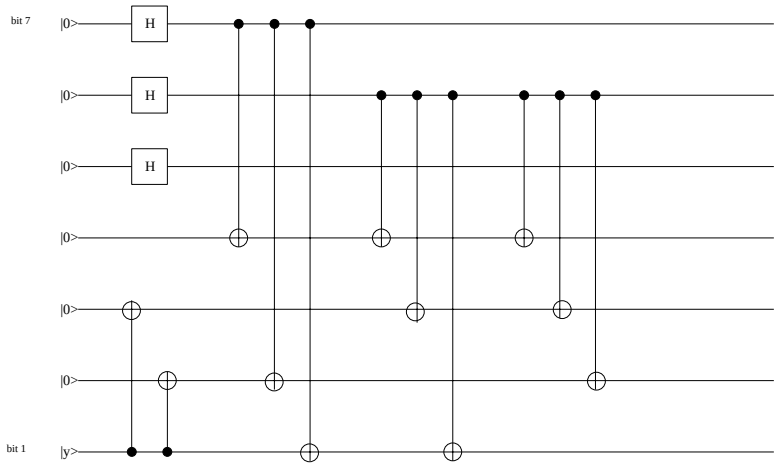

Figure 24: Non-fault tolerant encoder for the 7 Qubit Steane code. This figure is from John Preskill at Caltech.

- On average, no fixup is required (the likely case) after each logical operation and error correction step.
- Failure occurs when two or more errors show up in a code word.
- In general, error correction takes an error rate of $p = 1 - e^{-\lambda}$ to cp^2. The constant c is the number of double-points of failure in the logical operation and error correction step (i.e. the number of places that can both fail to produce two errors in a code word). It is about 21,942 for a straightforward implementation of stabilizer measurement for the 7 qubit Steane code.

Fault tolerant operation transforms:

$$1 - \varepsilon < e^{\lambda p(n)}$$

into:

$$1 - \varepsilon < \left[1 - cp^2\right]^{p(n)}$$

$$\approx 1 - \varepsilon < 1 - cp^2 \cdot p(n)$$

$$\frac{\varepsilon}{p(n)} > cp^2$$

Assuming we want a 95% chance of obtaining the correct answer and the same error rate as before ($\lambda = 10^{-6}$) we have that:

$$\frac{0.05}{65n^3n} > 21,942\left[1 - e^{-10^{-6}}\right]^2$$

$$n < \left[35,057\right]^{1/4}$$

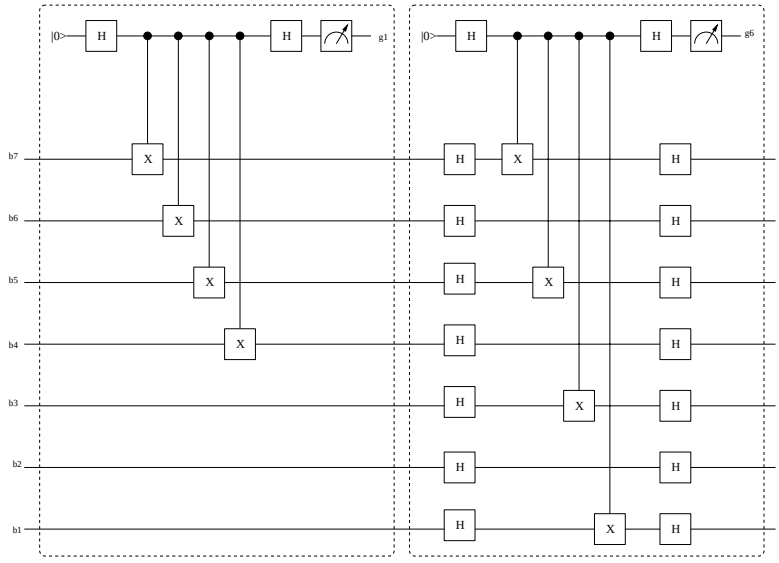

Figure 25: Almost fault-tolerant measurement of the stabilizers g_1 and g_6. To make this circuit fault tolerant, the ancilla must be a checked CAT state, and the measurement must be repeated at least two times.

$n \approx 13$ qubits

Clearly we have to do better, and for this we turn to recursive error correction. Similar to the concatenation we used to combine the bit and phase flip codes we can concatenate a quantum code with another quantum code. For our example, we are going to concatenate the 7 qubit Steane code with itself. The idea is to take a qubit and encode it in the 7 qubit Steane code. Then take each physical qubit that is part of that single logical qubit and re-encode it again in the 7 qubit Steane code. Repeat this procedure until we have a sufficiently strong error correcting code.

Note that this repetition would seem to imply an exponential blowup in resources – and it does, but it also creates an exponentially strong error correcting code. Thus the overhead is polynomial to perform computation. As we will see, however, just being polynomial doesn't make the overhead insignificant.

Key:
Concatenating codes recursively works so long as the underlying decoherence rate $p = 1 - e^{-\lambda}$ is less than $1/c$, the unit of complexity of the error correction step. This is the result of the threshold theorem which underpins fault tolerant computation. The concept is straightforward: we have to be able to correct fast enough such that any decoherence that occurs while we are correcting can be corrected away. If error correction takes so long that it would do more harm than good, then arbitrarily long quantum computation cannot be sustained.

Applying error correction recursively we obtain the following equation:

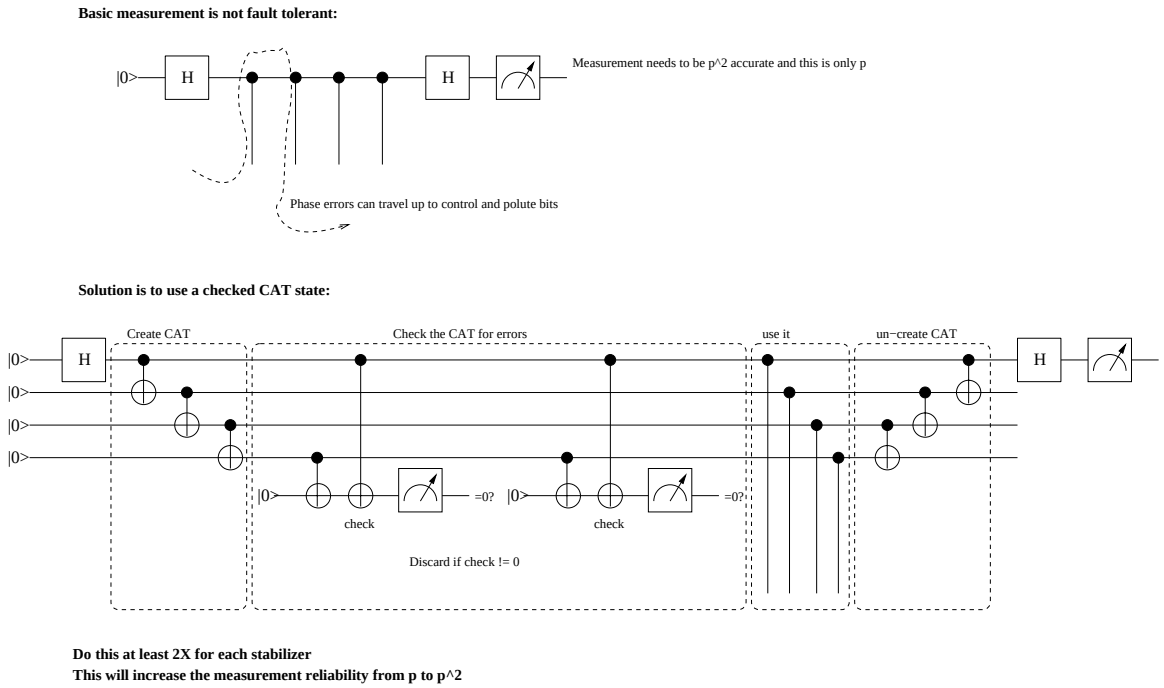

Figure 26: Making measurement of stabilizers fault-tolerant is an arduous task. First, a CAT state must be used in order to prevent errors in one qubit of the code word from polluting other qubits via the ancilla qubit. This CAT state must be checked to make sure it is reliable. Furthermore, the whole process must be repeated at least twice for each stabilizer since the measurement must be p^2 reliable.

$$\frac{(cp)^{2^k}}{c} < \frac{\varepsilon}{p(n)}$$

Here, $k \geq 1$ is the number of recursive levels of error correction. Note the 2^k in the equation, this is where the exponential improvement in tolerance to errors comes from.

Lets examine what happens when we apply 5 layers of error correction:

$k = 5$, $c = 21{,}942$, $p = 1 - e^{-10^{-6}}$, $\varepsilon = 0.05$, $p(n) = 65n^3n$

$$\frac{\left[21{,}942\left[1-e^{-10^{-6}}\right]\right]^{2^5}}{21{,}942} \leq \frac{0.05}{65n^3n}$$

$n < 3.7 \times 10^{13}$ qubits.

Now we can factor large numbers. However, there is a down side. The overhead for each qubit is $7^5 = 16{,}807$ physical qubits. Worse still 1 logical gate requires about $153^5 = 83{,}841{,}135{,}993$ physical gates. At 1Mhz operation 1 logical gate takes 23 hours to do (serially), so at $k = 5$, factoring a 1024 bit number takes 200 million years!

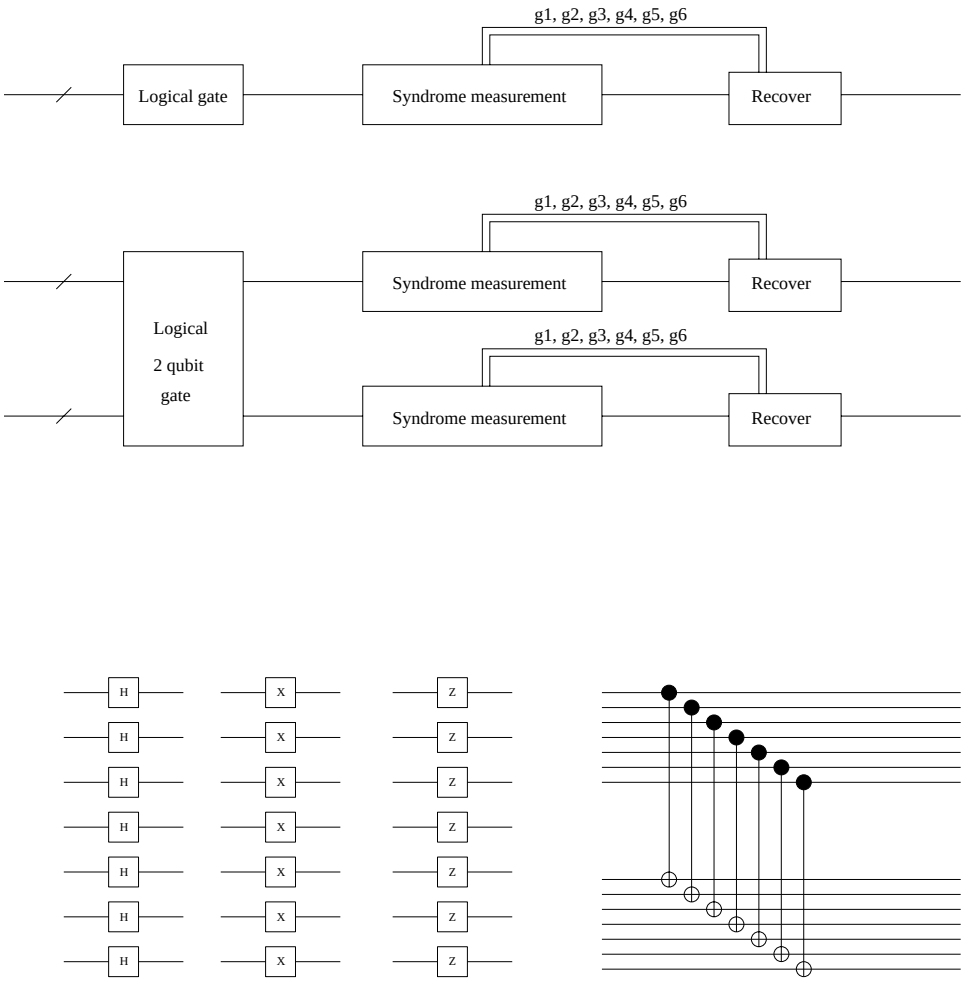

Figure 27: Fault tolerant computation proceeds by performing the coded operation, measuring the stabilizers (syndrome measurement), and then possibly recovering from an error. For the Steane code, the X, Z, H, and CNOT operations are easily applied in code-space, simply apply them transversally. Note this is not the case for T.

The solution: parallelism and quantum architecture.

- Really only need $k = 3$ for 1024 bits, so only 343 qubits of overhead, and 3,581,577 ops/logical op. This requires only 8000 years serially. Hence, still need quantum architecture.
- $k = 2$, $p = 10^{-8}$, 51 years
- $k = 2$, $p = 10^{-9}$, 1 Ghz operation (electron state), optmized error correction, clustering, 31 hours!