Lecture 11: November 15

Lecturer: Umesh Vazirani *Scribe: Michael Manapat*

In today's lecture, we are going to discuss both the motivation behind and the construction of quantum error correcting codes.

11.1 Motivation

11.1.1 The Need for Error Correction

We will begin by showing why error correcting codes are needed in the first place. Suppose we have a qubit $|\phi\rangle$. We want to use $|\phi\rangle$ in a computation, but there are some practical (physical) obstacles in the way:

1. **Unitary errors**. The gates used in quantum circuits are not perfect. For instance, if U is meant to perform a $\pi/4$ rotation, it may instead perform a $\pi/4 + \epsilon$ rotation for some $\epsilon > 0$. Recall that if each gate is correct to within ϵ/T, then the final output of a T-step computation will be correct to within ϵ. This means that, as our computation increases in length, the accuracy of our gates will also need to increase, but getting quantum gates to be arbitrarily accurate is difficult.

2. **Decoherence**. Another source of errors is decoherence—the accidental measurement of the quantum system by the environment. The power of quantum computation is a product of a system's ability to be in the superposition of several states; the effect of decoherence is to project a superposition onto a particular basis state, which (irreversibly, without error correction) corrupts the computation.

   Suppose, for example, that we have the state

   $$|\phi\rangle = \frac{1}{2}|00\rangle + \frac{i}{2}|10\rangle - \frac{1}{\sqrt{2}}|11\rangle,$$

   the first qubit of which is then measured by the environment. If the outcome is 0, then the resulting state is $|00\rangle$, and if the outcome is 1, then the resulting state is $\frac{i}{\sqrt{3}}|10\rangle - \sqrt{\frac{2}{3}}|11\rangle$. In both of these cases, at least one term in the superposition $|\phi\rangle$ has been lost.

11.1.2 The Possibility of Error Correction

An examination of the example above gives a naive argument as to why quantum error correction might not be possible. Intuitively, the linearity of quantum mechanics means that each basis state in a superposition is "unaware" both of the other terms in the superposition and of its own amplitude. A unitary transformation—which can represent a computation or an error—thus acts separately on each term in the superposition. To anthropomorphize a bit more, after an error has occurred, each basis state that remains "does not know that anything happened."

Another argument against quantum error correction is a consequence of the no-cloning theorem. Classical error correcting codes all rely on redundancy to protect data. The most extreme example is a repetition code, in which

$$0 \mapsto 00\cdots 0 = 0^n$$
$$1 \mapsto 11\cdots 1 = 1^n.$$

When a message is received, the recipient can count the number of 0's and 1's in the encoding of each bit and take the majority as the original value of the bit. This sort of encoding is not possible in the quantum realm because quantum states cannot be cloned: there is no transformation U that takes $|\phi\rangle|0\rangle$ to $|\phi\rangle|\phi\rangle$. Hence, if quantum codes exist, they will not be direct generalizations of classical repetition codes.

Despite these arguments, quantum error correction is possible, and classical error correcting codes will play a significant role in their construction. Before discussing that construction explicitly, we will first define exactly what an error is.

11.2 Errors

Suppose we have a single qubit $|\phi\rangle$. The most general sort of error that can occur is a unitary transformation U on the product of the system and environment states, $|\phi\rangle \otimes |e\rangle$. This formulation of "error" clearly captures the possibility of *unitary errors*, but it might not be apparent why decoherence, i.e. measurements by the environment, can also be represented by such transformations.

To understand this intuitively, recall that quantum mechanics has two sets of "rules": one set dictates how systems evolve (by unitary transformations) and the other specifies how measurements are made (by projection). When the environment E measures $|\phi\rangle$ according to the rules for measurement, the combination of E and $|\phi\rangle$ is just a quantum system evolving according to the rules for evolution! This is why we can model the act of measurement by a unitary operator U acting on $|\phi\rangle \otimes |e\rangle$.

Now recall that the identity $I = \left(\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right)$, the bit flip $X = \left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right)$, the phase flip $Z = \left(\begin{smallmatrix} 1 & 0 \\ 0 & -1 \end{smallmatrix}\right)$, and the bit and phase flip $XZ = \left(\begin{smallmatrix} 0 & 1 \\ -1 & 0 \end{smallmatrix}\right)$ form a basis for the linear transformations on a single qubit state. We claim that if we can correct I, X, Z, and XZ errors, then we can correct the most general one-qubit error.

Let $|\phi\rangle = \alpha|0\rangle + \beta|1\rangle$ and let the environment state be $|e\rangle$. Suppose that the error E entangles a quantum state with its environment according to the rules

$$|0\rangle|e\rangle \mapsto |0\rangle|e_0^0\rangle + |1\rangle|e_1^0\rangle$$
$$|1\rangle|e\rangle \mapsto |1\rangle|e_0^1\rangle + |1\rangle|e_1^1\rangle.$$

Then

$$\begin{aligned} E|\phi\rangle &= E(\alpha|0\rangle + \beta|1\rangle) \\ &= \alpha(|0\rangle|e_0^0\rangle + |1\rangle|e_1^0\rangle) + \beta(|1\rangle|e_0^1\rangle + |1\rangle|e_1^1\rangle) \\ &= |0\rangle(\alpha|e_0^0\rangle + \beta|e_0^1\rangle) + |1\rangle(\alpha|e_1^0\rangle + \beta|e_1^1\rangle). \end{aligned}$$

We can write this transformation as a matrix whose entries are environment states:

$$\begin{pmatrix} |e_0^0\rangle & |e_0^1\rangle \\ |e_1^0\rangle & |e_1^1\rangle \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} \alpha|e_0^0\rangle + \beta|e_0^1\rangle \\ \alpha|e_1^0\rangle + \beta|e_1^1\rangle \end{pmatrix}.$$

Since the Pauli operators are a basis for linear transformations on a single qubit, this operator is just a linear combination of Pauli operators with coefficients being environment states:

$$\begin{pmatrix} |e_0^0\rangle & |e_0^1\rangle \\ |e_1^0\rangle & |e_1^1\rangle \end{pmatrix} = |e_I\rangle I + |e_X\rangle X + |e_Z\rangle Z + |e_{XZ}\rangle XZ,$$

so the corrupted version of the state $|\phi\rangle$ is

$$E|\phi\rangle = |e_I\rangle I|\phi\rangle + |e_X\rangle X|\phi\rangle + |e_Z\rangle Z|\phi\rangle + |e_{XZ}\rangle XZ|\phi\rangle.$$

From this last expression, it is clear that if we can correct against I, X, Z, and XZ, then we can restore the state $U|\phi\rangle$ to just $|\phi\rangle$, where $U \in \{I, X, Z, XZ\}$. This puts $E|\phi\rangle$ into a direct product state of the environment and $|\phi\rangle$, which is precisely what we want.

While we've been focusing on errors on a single qubit, all of this generalizes easily to the case in which k errors can occur among n qubits. An error in this general setting is then a linear combination of the form

$$E = \sum_{i=1}^{m} \alpha_i U_{i1} \otimes U_{i2} \otimes \cdots \otimes U_{in},$$

where each U_{ij} is one of I, X, Z, or XZ and for any fixed i at most k of the U_{ij} are not the identity. If $|\phi\rangle$ were a 5-qubit state, for instance, and up to three errors were allowed, then one possible error would be

$$\frac{1}{\sqrt{2}} I \otimes X \otimes Z \otimes I \otimes XZ + \frac{1}{\sqrt{2}} I \otimes Z \otimes I \otimes XZ \otimes I.$$

11.3 Shor's 9-Qubit Code

We will now examine a specific example of an error correcting code—Shor's 9-qubit code. This particular code encodes a one-qubit state as nine qubits and can recover from an arbitrary X error and an arbitrary Z error. Since the X and Z error can act on the same qubit, it follows that the code protects against XZ errors as well and thus against the most general error that can occur.

Suppose first that the only bit flips (X errors) can occur. Then we will encode states according to the rules

$$|0\rangle \mapsto |000\rangle \qquad \text{and} \qquad |1\rangle \mapsto |111\rangle,$$

so

$$|\phi\rangle = \alpha|0\rangle + \beta|1\rangle \mapsto \alpha|000\rangle + \beta|111\rangle.$$

If a single bit-flip occurs—say the second qubit is flipped—then the corrupted state is $\alpha|010\rangle + \beta|101\rangle$. We can then decode by doing a majority count as in the classical repetition code, producing the state $(\alpha|010\rangle + \beta|101\rangle) \otimes |2\rangle$[1]. Given this last state, the second register instructs us to flip the second bit, which restores the original state[2].

[1] Recall that for any efficiently computable function f, there is an efficient quantum circuit U_f that computes $\sum_x \alpha_x|x\rangle|0\rangle \mapsto \sum_x \alpha_x|x\rangle|f(x)\rangle$. In this case, the function f is "which of the qubits differs from the others." The use of U_f might seem circular, for U_f itself might be subject to unitary errors. For now, we'll just assume that U_f acts perfectly. We'll show how to get around this later when we discuss fault tolerant quantum computation.

[2] In the decoding procedure, the second register must be initialized with a clean ancilla state $|0\rangle$, which eventually is modified to store the position of the flipped bit. A moment's thought shows that such an ancilla is in fact required—there is no way to go directly from the corrupted state back to the original state. The reason for this is simple: unitary transformations are reversible, so we can't erase any information in the decoding procedure. If we didn't store the location of the error in some supplementary register, we would be erasing the information introduced by the error, but quantum mechanics forbids such erasure.

Of course, an error may be a linear combination of bit flips acting on different qubits in the encoding, for instance

$$\frac{1}{\sqrt{2}}X_1 + \frac{1}{\sqrt{2}}X_2,$$

where X_1 flips the first qubit and X_2 flips the second. In this case, the majority count yields the state

$$\frac{1}{\sqrt{2}}(\alpha|100\rangle + \beta|011\rangle) \otimes |1\rangle) + \frac{1}{\sqrt{2}}(\alpha|010\rangle + \beta|101\rangle) \otimes |2\rangle). \qquad (*)$$

When the second register is measured to determine the position at which the error occurred, $(*)$ is projected onto one of the two states in the superposition—assume the first, so we get

$$(\alpha|100\rangle + \beta|011\rangle) \otimes |1\rangle,$$

and at this point the correction can proceed exactly as above.

Now suppose that only phase errors occur. We first write $|\phi\rangle$ in the Hadamard basis, $|\phi\rangle = \alpha'|+\rangle + \beta'|-\rangle$, where $|+\rangle = \frac{|0\rangle+|1\rangle}{\sqrt{2}}$ and $|-\rangle = \frac{|0\rangle-|1\rangle}{\sqrt{2}}$. Then $Z|+\rangle = |-\rangle$ and $Z|-\rangle = |+\rangle$, so a phase flip is just a bit flip in the new basis.

The encoding for phase flips, then, will be just like the encoding for bit flips except that we'll be working in the Hadamard basis: $\alpha'|+\rangle + \beta'|-\rangle$ is encoded as $\alpha'|+++\rangle + \beta'|---\rangle$. Since we've reduced phase flips to bit flips, the decoding is exactly as before (again in the new basis though).

We can now protect against both bit and phase flips by concatenating these two codes, first encoding to protect against phase flips and then encoding that intermediate result to protect against bit flips:

$$\begin{aligned}
|0\rangle \mapsto |+++\rangle = \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right)^{\otimes 3} &\mapsto \frac{1}{\sqrt{2}^3}(|000\rangle + |111\rangle)^{\otimes 3} \\
|1\rangle \mapsto |+++\rangle = \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right)^{\otimes 3} &\mapsto \frac{1}{\sqrt{2}^3}(|000\rangle - |111\rangle)^{\otimes 3}.
\end{aligned}$$

The encoding of the general state $|\phi\rangle = \alpha|0\rangle + \beta|1\rangle$ is then $\frac{\alpha}{\sqrt{2}^3}(|000\rangle + |111\rangle)^{\otimes 3} + \frac{\beta}{\sqrt{2}^3}(|000\rangle - |111\rangle)^{\otimes 3}$ (we will normally omit the normalization constant $\frac{1}{\sqrt{2}^3}$ for clarity). It is easy to see that this encoding protects against at most one bit error in each of the three block of three qubits.

To decode, we correct bit errors, undo the bit encoding, and the correct for phase errors. Here is a concrete example:

We start with the encoding of the state $|\phi\rangle$ above. Suppose that the second qubit is subjected to a bit flip X and the fourth to a phase flip Z, so the corrupted state is then

$$\alpha(|010\rangle + |101\rangle)(|000\rangle - |111\rangle)(|000\rangle + |111\rangle) + \beta(|010\rangle - |101\rangle)(|000\rangle + |111\rangle)(|000\rangle - |111\rangle).$$

The bit flip decoding procedure takes this state as input in the first register and a clean ancilla qubit $|0\rangle$ as input in the second register. By running the "majority" computation, we find that the bit flip occurred in the second bit, and we store the location of the error in the second register, giving

$$[\alpha(|010\rangle + |101\rangle)(|000\rangle - |111\rangle)(|000\rangle + |111\rangle) + \beta(|010\rangle - |101\rangle)(|000\rangle + |111\rangle)(|000\rangle - |111\rangle)]|2\rangle.$$

As before, we then flip the second bit, which corrects the bit flip error and yields

$$\alpha(|000\rangle + |111\rangle)(|000\rangle - |111\rangle)(|000\rangle + |111\rangle) + \beta(|000\rangle - |111\rangle)(|000\rangle + |111\rangle)(|000\rangle - |111\rangle)$$

(we won't write the state of the second register unless it's important).

Now we know that there are no bit flip errors, so there is no longer any use in keeping the redundancy that we introduced to guard against bit flips. We can "throw away" this unneeded information by applying a controlled-NOT to the $(k+1)$st and $(k+2)$nd qubits for $k = 1, 4, 7$, where the kth qubit is the control. The result of this operation is

$$\alpha(|000\rangle + |100\rangle)(|000\rangle - |100\rangle)(|000\rangle + |100\rangle) + \beta(|000\rangle - |100\rangle)(|000\rangle + |100\rangle)(|000\rangle - |100\rangle),$$

which is in a direct product state. We can factor out the $|00\rangle$ to get

$$\alpha|+\rangle|-\rangle|+\rangle + \beta|-\rangle|+\rangle|-\rangle,$$

and now we correct for a bit flip error in the Hadamard basis.

11.4 CSS (Calderbank-Shor-Steane) Codes

Shor's 9-qubit code is a novel adaptation of the classical repetition codes (that gets around the no-cloning theorem). Since general quantum codes are themselves—at least in some sense—adaptations of general classical codes, we will review the basics of classical error correction.

11.4.1 Classical Error Correcting Codes

In the following, we will examine linear error correcting codes over the group $\mathbb{Z}_2^n$. In a classical $[n, k, d]$ code, a message $m \in \mathbb{Z}_2^n$ is encoded as an element $c \in \mathbb{Z}_2^n$. The code is able to detect $d-1$ bit errors and correct $\frac{d-1}{2}$ errors.

Generally, the first k bits of the encoding consist of the original message, and the last $n-k$ bits are "parity check bits," i.e., if the message is $x_1 x_2 \cdots x_k$, a typical parity check bit might be $x_1 \oplus x_2 \oplus x_k$. The parity check bits are chosen so that if $y_1 y_2 \cdots y_k$ is another message not equal to $x_1 x_2 \cdots x_k$, then the encodings of x and y differ in at least d positions.

Such an encoding allows us to detect $d-1$ errors since no other codeword is within a Hamming distance of $d-1$ from the encoding of the message. Similarly, we can correct $\frac{d-1}{2}$ errors since the balls of radius $\frac{d-1}{2}$ centered at each codeword are disjoint (for if two such balls had a nonempty intersection, the corresponding centers would have a Hamming distance of less than d, which is impossible by construction–see figure 11.1).

There are two matrices that come up in the use of these codes. The first is a $k \times n$ generator matrix G. The rows of G span the space of the codewords, and since each codeword must be a associated with a unique message, the rows are therefore linearly independent (so G has rank k). Given a message m (written as a row vector), its encoding is mG.

The second matrix is the $n \times (n-k)$ parity check matrix P. P is chosen so that, for an n-bit string c, $cP = 0$ if and only if c is a valid codeword. This is equivalent to the statement that the kernel of P is the space generated by the codewords of the code. The column space of P is then $C^\perp$, the dual group of C[3]. If c is a codeword of the code and e is an error vector, then a corrupted codeword is the sum of these, $c' = c + e$. Then $c'P = eP$, and eP has the following nice properties:

- If $0 < |e| \leq d-1$, then $eP \neq 0$.
- If $0 < |e| \leq \frac{d-1}{2}$, then eP uniquely determines e.

Given a message c', $c'P$ is called the **syndrome** since it indicates if errors have occurred and, if sufficiently few have, tells precisely what the errors were.

[3]Recall that the dual $C^\perp$ of a group C is defined to be $\{d : cd = 0, c \in C\}$.

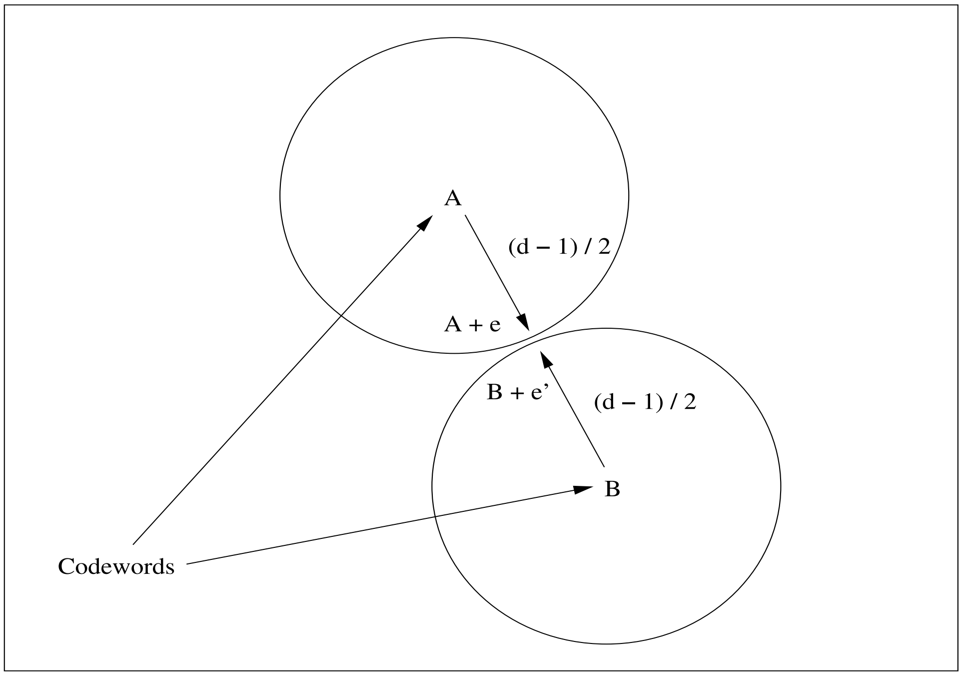

Figure 11.1: A and B are codewords of the code, and e and e' are error vectors. The balls of radius $\frac{d-1}{2}$ centered at A and B are disjoint, so errors that affect no more than $\frac{d-1}{2}$ bits can be corrected.

11.4.2 CSS Codes

The key idea in Shor's 9-qubit code is that phase errors in the standard basis are bit errors in the Hadamard basis, so the ability to correct bit errors implies the ability to correct phase errors.

For the general case, in which we want to correct more than one error, the principle is the same: we first correct bit errors in the standard basis, and then we apply a Fourier transform and correct bit errors in the Hadamard basis.

Before continuing, let us recall some properties of the Fourier transform. Let G be a subgroup of $\mathbb{Z}_2^n$ and fix a vector $x \in \mathbb{Z}_2^n$. Let

$$|x+G\rangle = \frac{1}{\sqrt{|G|}} \sum_{y \in G} |x+y\rangle.$$

Let H denote the Hadamard transform. Then

$$\begin{aligned} H|x+G\rangle &= \frac{1}{\sqrt{|G|}} \sum_{y \in G} H|x+y\rangle \\ &= \frac{1}{\sqrt{|G|}} \sum_{y \in G} \left(\frac{1}{\sqrt{2^n}} \sum_{z \in \mathbb{Z}_2^n} (-1)^{(x+y)\cdot z} |z\rangle \right) \\ &= \frac{1}{\sqrt{|G|2^n}} \sum_{z \in \mathbb{Z}_2^n} (-1)^{x \cdot z} \left(\sum_{y \in G} (-1)^{y \cdot z} \right) |z\rangle. \qquad (*) \end{aligned}$$

Suppose that $z \in G^\perp$, so $y \cdot z = 0$ for all $y \in G$. Then we have

$$\sum_{y \in G} (-1)^{y \cdot z} = \sum_{y \in G} 1 = |G|,$$

so we can rewrite $(*)$ as

$$\frac{|G|}{\sqrt{|G|2^n}} \sum_{w \in G^\perp} (-1)^{x \cdot w} |w\rangle + \frac{1}{\sqrt{|G|2^n}} \sum_{z \notin G^\perp} (-1)^{x \cdot z} \left(\sum_{y \in G} (-1)^{y \cdot z} \right) |z\rangle. \qquad (**)$$

The first term then simplifies to

$$\frac{1}{\sqrt{2^n/|G|}} \sum_{w \in G^\perp} (-1)^{x \cdot w} |w\rangle = \frac{1}{\sqrt{|G^\perp|}} \sum_{w \in G^\perp} (-1)^{x \cdot w} |w\rangle,$$

but this superposition already has an amplitude of 1, so we conclude that

$$\frac{1}{\sqrt{|G|2^n}} \sum_{z \notin G^\perp} (-1)^{x \cdot z} \left(\sum_{y \in G} (-1)^{y \cdot z} \right) |z\rangle = 0$$

in $(**)$ and therefore that

$$H|x + G\rangle = \frac{1}{\sqrt{|G^\perp|}} \sum_{w \in G^\perp} (-1)^{x \cdot w} |w\rangle.$$

What this means is that the effect of the Hadamard transform is to transfer the coset information to the phase. We will often write

$$\frac{1}{\sqrt{|G^\perp|}} \sum_{w \in G^\perp} (-1)^{x \cdot w} |w\rangle = \frac{1}{\sqrt{|G^\perp|}} \sum_{w \in G^\perp} \chi_x(w) |w\rangle,$$

where $\chi_x(w) = (-1)^{x \cdot w}$ is the "character."

Since the Hadamard transform puts coset information in the phase, we might guess that it puts phase information in the coset. In fact this is exactly what the transformation does, as we will show with a computation similar to the one above. First observe that

$$\begin{aligned} H\chi_p|G\rangle &= \frac{1}{\sqrt{|G|}} \sum_{y \in G} (-1)^{p \cdot y} H|y\rangle \\ &= \frac{1}{\sqrt{|G|}} \sum_{y \in G} (-1)^{p \cdot y} \left(\frac{1}{\sqrt{2^n}} \sum_{z \in \mathbb{Z}_2^n} (-1)^{y \cdot z} |z\rangle \right) \\ &= \frac{1}{\sqrt{|G|2^n}} \sum_{z \in \mathbb{Z}_2^n} \left(\sum_{y \in G} (-1)^{(p+z) \cdot y} \right) |z\rangle. \qquad (*) \end{aligned}$$

Let $p + z = w$. If $w \in G^\perp$, then

$$\sum_{y \in G} (-1)^{w \cdot y} = \sum_{y \in G} 1 = |G|,$$

so $(*)$ becomes

$$\frac{|G|}{\sqrt{|G|2^n}} \sum_{w \in G^\perp} |p + w\rangle + \frac{1}{\sqrt{|G|2^n}} \sum_{z \notin G^\perp} \left(\sum_{y \in G} (-1)^{(p+z) \cdot y} \right) |z\rangle.$$

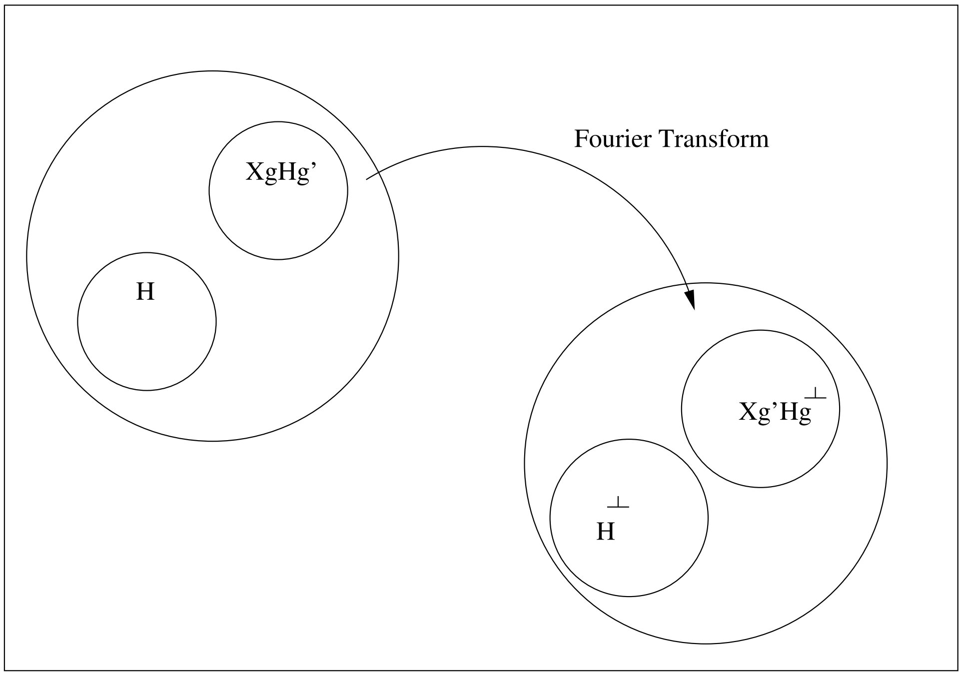

Figure 11.2: The effect of the Fourier transform is to encode phase information in the coset and coset information in the phase.

But, as above,

$$\frac{|G|}{\sqrt{|G|2^n}} \sum_{w \in G^\perp} |p+w\rangle = \frac{1}{\sqrt{|G^\perp|}} \sum_{w \in G^\perp} |p+w\rangle,$$

and this superposition has amplitude 1, so we deduce that

$$H\chi_p|G\rangle = \frac{1}{\sqrt{|G^\perp|}} \sum_{w \in G^\perp} |p+w\rangle = |p+G^\perp\rangle.$$

Finally, putting it all together, we can say that the Hadamard transform switches the phase and coset information (an overall global phase is introduced, but we know that such phases are irrelevant). One last, laborious computation confirms that the Hadamard transform has this critical property:

$$\begin{aligned}
H\chi_p|x+G\rangle &= \frac{1}{\sqrt{|G|}} \sum_{y \in G} (-1)^{(x+y)\cdot p} H|x+y\rangle \\
&= \frac{1}{\sqrt{|G|}} \sum_{y \in G} (-1)^{(x+y)\cdot p} \left(\frac{1}{\sqrt{2^n}} \sum_{z \in \mathbb{Z}_2^n} (-1)^{(x+y)\cdot z} |z\rangle \right) \\
&= \frac{1}{\sqrt{|G|2^n}} \sum_{z \in \mathbb{Z}_2^n} (-1)^{x\cdot(p+z)} \left(\sum_{y \in G} (-1)^{y\cdot(p+z)} \right) |z\rangle \\
&= \frac{1}{\sqrt{|G|2^n}} (-1)^{x\cdot p} \sum_{z \in \mathbb{Z}_2^n} (-1)^{x\cdot z} \left(\sum_{y \in G} (-1)^{y\cdot(p+z)} \right) |z\rangle.
\end{aligned}$$

Now, as before, we let $p + z = w \in G^\perp$, and after applying exactly the same reasoning as we did above, we find that

$$\begin{aligned} H\chi_p|x+G\rangle &= (-1)^{x\cdot p}\frac{1}{\sqrt{|G^\perp|}}\sum_{w\in G^\perp}(-1)^{x\cdot(p+w)}|p+w\rangle \\ &= (-1)^{x\cdot p}\chi_x|p+G^\perp\rangle, \end{aligned}$$

which is what we wanted to show. In what follows, we will omit the global phase $(-1)^{x\cdot p}$ for clarity.

We can now construct an error correcting code as follows. We start by taking two classical linear error correcting codes over $\mathbb{Z}_2^n$, C_1 and C_2, such that $C_1 \subset C_2$ and $C_1^\perp$ and C_2 are both codes capable of detecting up to d errors ($C_1^\perp$ is an $[n, k_1, d]$ and C_2 is an $[n, k_2, d]$ code). We will get an $[[n, k_2 - k_1, d]]$ quantum error correcting code which encodes $k_2 - k_1$ qubits as n qubits, detects $d - 1$ bit and phase errors, and corrects $\frac{d-1}{2}$ errors of each type. The codewords of the code are uniform superpositions over all elements in a coset of C_1 in C_2,

$$\frac{1}{\sqrt{|C_1|}}\sum_{y\in C_1}|x+y\rangle = |x+C_1\rangle,$$

where x is some element in C_2.

We can represent both a bit flip error e_b and a phase flip error e_p as vectors in $\mathbb{Z}_2^n$; the entries of e_b (respectively e_p) with 1's index which qubits were subjected to bit flips (respectively phase flips). A the bit flip error yields the superposition

$$\frac{1}{\sqrt{|C_1|}}\sum_{y\in C_1}|x+e_b+y\rangle = |x+e_b+C_1\rangle,$$

which moves us from one coset of C_1 to another, and a phase flip e_p results in the phase $(-1)^{y\cdot e_p}$ for the term $|x + y\rangle$ in the superposition. The result of both of these errors is the state

$$\begin{aligned} \frac{1}{\sqrt{|C_1|}}\sum_{y\in C_1}(-1)^{(x+y)\cdot e_p}|x+e_b+y\rangle &= \frac{1}{\sqrt{|C_1|}}\sum_{y\in C_1}\chi_{e_p}(x+y)|x+e_b+y\rangle \\ &= \chi_{e_p}|x+e_b+C_1\rangle. \end{aligned}$$

Since C_2 is an error correcting code, we can correct the bit flip error e_b classically to get the state $\chi_{e_p}|x + C_1\rangle$ (see figure 11.3). We then apply the Fourier transform, which gives $\chi_x|e_p + C_1^\perp\rangle$, and we then correct the *bit* flip error e_p classically, which we can do since $C_1^\perp$ is an error correcting code (see figure 11.4). This gives $\chi_x|C_1^\perp\rangle$, and if we perform the Fourier transform once more, the original encoded state $|x + C_1\rangle$ is restored.

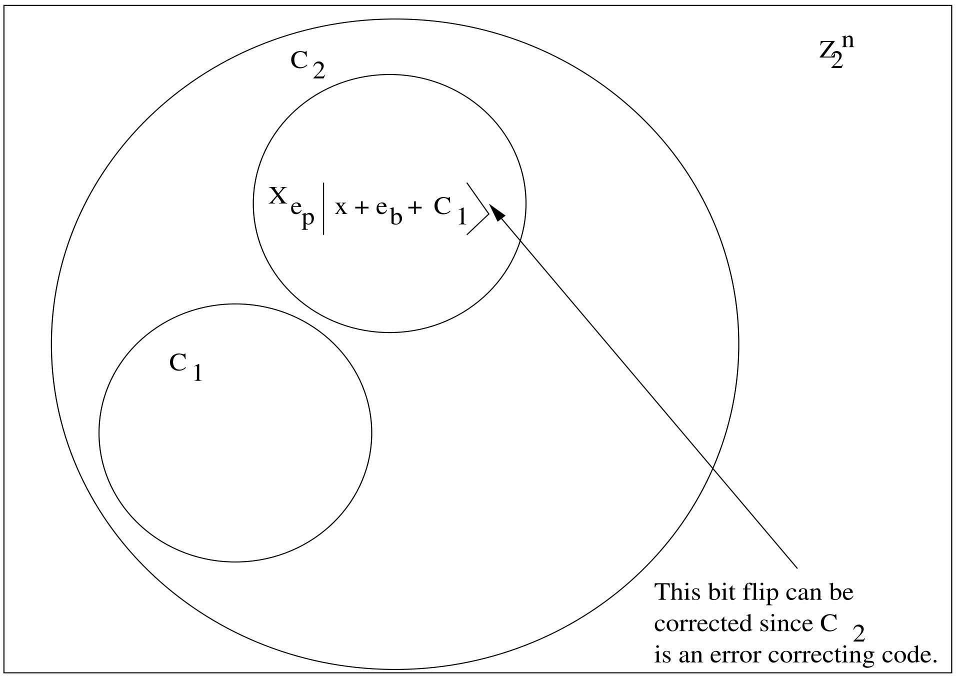

Figure 11.3: Correction of bit flip errors is done classically, using the code C_2.

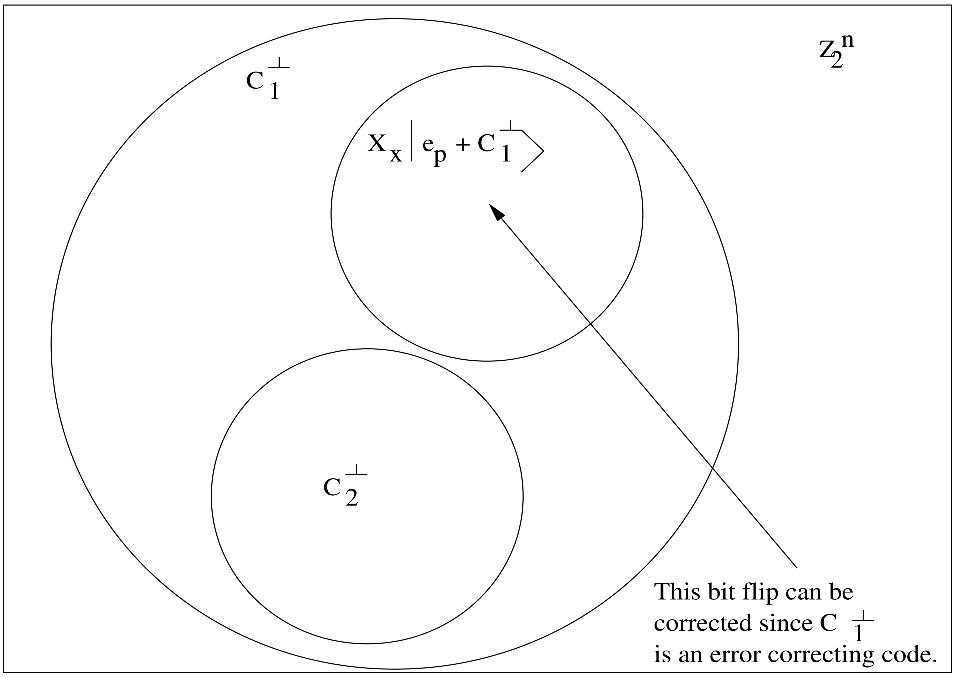

Figure 11.4: Phase flip errors are first transformed to bit flip errors by the Fourier transform. Since $C_1^\perp$ is an error correcting code, we can then correct those bit flips classically.