CSE 599d - Quantum Computing
Stabilizer Quantum Error Correcting Codes

Dave Bacon
Department of Computer Science & Engineering, University of Washington

In the last lecture we learned of the quantum error correcting criteria and we discussed how it was possible for us to digitize quantum errors. But we didn't talk about any concrete codes, the talk was in many ways very existential on these things called quantum error correcting codes. Of course we saw in Shor's code that such codes could exist for single qubit errors. In this lecture we will introduce a very handy set of tools for describing a class of quantum error correcting codes. This formalism is called the stabilizer formalism. Further with stabilizer codes we can also begin to discuss an important topic, which is not just doing quantum error correction, but also we can begin to discuss quantum gates on these codes and this will eventually lead us to the issues of fault-tolerant quantum computation.

I. ANTICOMMUTING

Suppose that we have a set of states $|\psi_i\rangle$ which are $+1$ eigenstate of a hermitian operator S, $S|\psi_i\rangle = |\psi_i\rangle$. Further suppose that T is an operator which anticommutes with S, $ST = -TS$ (T is not zero.) Then it is easy to see that $S(T|\psi_i\rangle) = -TS|\psi_i\rangle = -(T|\psi_i\rangle)$. Thus the states $T|\psi\rangle$ are -1 eigenstates of S. Since the main idea of quantum error correction is to detect when an error has occurred on a code space, such pairs of operators S and T can be used in such a manner: if we are in the $+1$ eigenvalue subspace of S then an error of T on these subspaces vectors will move to a -1 eigenvalue subspace of S: we can detect that this error has occurred.

In fact, we have already seen an example of this in the bit flip code. Recall that we noted that the code subspace for this code was spanned by $|000\rangle$ and $|111\rangle$ and that these two operators are $+1$ eigenstates of both $S_1 = Z \otimes Z \otimes I$ and $S_2 = Z \otimes I \otimes Z$. Further note that $(X \otimes I \otimes I)S_1 = -S_1(X \otimes I \otimes I)$ and $(X \otimes I \otimes I)S_1 = -S_2(X \otimes I \otimes I)$. Thus if we start out in the $+1$ eigenvalue subspace of both S_1 and S_2 (like the bit flip code), then if a single bit flip occurs on the first qubit, we will now have a state which is in the -1 eigenvalue subspace of both S_1 and S_2. This at least fulfills our requirement that our errors should take us to orthogonal subspaces.

More generally, considering the following situation. Suppose that we have a set of operators S_i such that our code space is defined by $S_i|\psi\rangle = |\psi\rangle$ for $|\psi\rangle$ in the code subspace. Now suppose that we have errors E_i such that the products $E_k^\dagger E_l$ always anti-commutes with at least one S_i. Recall the quantum error correcting criteria was

$$\langle\phi_i|E_k^\dagger E_l|\phi_j\rangle = C_{kl}\delta_{ij} \tag{1}$$

Since the codewords are $+1$ eigenvalue eigenstates of S_i, we find that

$$\langle\phi_i|E_k^\dagger E_l|\phi_j\rangle = \langle\phi_i|E_k^\dagger E_l S_i|\phi_j\rangle \tag{2}$$

Suppose that S_i is one of the particular S_is that anticommute with $E_k^\dagger E_l$. Then this is equal to

$$\langle\phi_i|E_k^\dagger E_l|\phi_j\rangle = \langle\phi_i|E_k^\dagger E_l S_i|\phi_j\rangle = -\langle\phi_i|S_i E_k^\dagger E_l|\phi_j\rangle \tag{3}$$

But, since S_i acts as $+1$ on the codespace, this is just

$$\langle\phi_i|E_k^\dagger E_l|\phi_j\rangle = \langle\phi_i|E_k^\dagger E_l S_i|\phi_j\rangle = -\langle\phi_i|S_i E_k^\dagger E_l|\phi_j\rangle = -\langle\phi_i E_k^\dagger E_l|\phi_j\rangle \tag{4}$$

This implies that

$$\langle\phi_i|E_k^\dagger E_l|\phi_j\rangle = 0 \tag{5}$$

Thus we have shown that, given the S_i and $E_k^\dagger E_l$ which properly anticommute with these S_i, the set of errors $\{E_k\}$ satisfies the quantum error correcting criteria and therefore the code space is a valid quantum error correcting code for these errors.

This trick, of defining the states as being the $+1$ eigenstates of some operators and then noting that if the product of error terms anticommute then this is a valid quantum error correcting code is at the heart of the reason we use the stabilizer formalism. But what should we use for the S_is? Well, you might already be guessing what to use, because you recall that the Pauli group has nice commuting, anticommuting properties and eigenvalues that are ± 1 (or $\pm i$.) Indeed this is what we will use.

II. THE PAULI AND STABILIZER GROUPS

A. Pauli Group

First recall the definition of a group. A group is a set of objects $\mathcal{G}$ along with a binary operation of multiplication which satisfies (0) [closure] $g_1 g_2 \in \mathcal{G}$ for all $g_1, g_2 \in \mathcal{G}$, (1) [associativity] $g_1(g_2 g_3) = (g_1 g_2) g_3$ for all $g_1, g_2, g_3 \in \mathcal{G}$, (2) [inverse] There exists an element $e \in \mathcal{G}$ such that for all $g_1 \in \mathcal{G}$, $g_1 e = g_1$, (3)[inverse] for every $g_1 \in \mathcal{G}$ there exists an element $g_2 \in \mathcal{G}$ such that $g_1 g_2 = e$, which we call the inverse of g_1, written $g_2 = g_1^{-1}$. The Pauli group is a particular group which satisfies these group axioms. Actually people in quantum computing are very sloppy and when they refer to the Pauli group usually refer to a particular representation of the Pauli group by unitary matrices. We will slip into this nomenclature soon enough, having learned a bad habit it is hard to go back.

What is this representation of the Pauli group, $\mathcal{P}_n$? Recall that the Pauli operators on a single qubit are $\{I, X, Y, Z\}$. The representation of the Pauli group we will deal with is the group formed by elements of the form $i^k P_1 \otimes P_2 \otimes \cdots \otimes P_n$ where each P_i is an element of $\{I, X, Y, Z\}$. From this representation, we see that the Pauli group is a nonabelian group, i.e. the elements of the group do not all commute with each other. Some important properties of elements of the Pauli group:

1. Elements of the Pauli group square to $\pm I$: $P^2 = \pm I$.
2. Elements of a the Pauli group either commute $PQ = QP$ or anticommute $PQ = -QP$.
3. Elements of the Pauli group are unitary $PP^\dagger = I$

B. Stabilizer Group

Define a stabilizer group $\mathcal{S}$ is a subgroup of $\mathcal{P}_n$ which has elements which all commute with each other and which does not contain the element $-I$. An example of a stabilizer group on three qubits is the group with elements $\mathcal{S} = \{III, ZZI, ZIZ, IZZ\}$. Notice that here we have dropped the tensor product between the elements, i.e. $ZZI = Z \otimes Z \otimes I$. We usually don't specify all of the elements of the stabilizer group. Instead we specify a minimal set of generators. A set of generators of a group is a set of elements of the group such that multiplication of these generators leads to the full group. A minimal set of such generators is a set of generators of minimal size which has this property (there may be multiple such sets.) In the example of $\mathcal{S} = \{III, ZZI, ZIZ, IZZ\}$, this group is generated by ZZI and ZIZ: $(ZZI)(ZIZ) = IZZ$ and $(ZZI)^2 = III$. We write this fact as $\mathcal{S} = \langle ZZI, ZIZ \rangle$. For a stabilizer $\mathcal{S}$ we write a set of minimal generators as $S_1, S_2, \dots, S_r$.

Now since $-I$ is not in a stabilizer, all elements of our stabilizer must square to $+I$. Operators which square to $+I$ must have eigenvalues $+1$ or -1. Since $\mathcal{S}$ is abelian, we can write a generic element of the stabilizer as $S_1^{a_1} S_2^{a_2} \cdots S_k^{a_r}$. Since $S_i^2 = I$, $a_i \in \{0, 1\}$. Further for each $a \in \mathbb{Z}_2^r$ the elements of the stabilizer so specified is unique. Suppose that this was not true, that $S_1^{a_1} S_2^{a_2} \cdots S_k^{a_r} = S_1^{b_1} S_2^{b_2} \cdots S_k^{b_r}$ for $a \neq b$. Then $S_1^{a_1+b_1} S_2^{a_2+b_2} \cdots S_k^{a_r+b_r} = I$. But this is only true if $a_i = b_i$.

C. Stabilizer Subspace and Error Correction

Now given a stabilizer group $\mathcal{S}$, we can define a subspace on our n qubits. In particular we define this subspace as all states $|\psi\rangle$ which satisfy $S|\psi\rangle = |\psi\rangle$ for all stabilizer elements $S \in \mathcal{S}$. Actually we don't need all of these equations to define the code space. All we need to do is the equations for the generators of the stabilizer: $S_i|\psi\rangle = |\psi\rangle$. Lets call the subspace defined by these equations $\mathcal{H}_S$. Such stabilizer subspace are very nice. One reason the are nice is that instead of specifying the states of the subspace we can just specify the generators of the stabilizer group. This is oftentimes much easier. Further, as we shall see, it is easy to figure out the error correcting properties of stabilizer subspaces.

In particular, suppose we have a stabilizer subspace for a code generated by $S_1, S_2, \dots, S_r$. Then suppose the Pauli operator P anticommutes with one of these generators S_i. Then, as above, for $|\psi_i\rangle$ a basis for $\mathcal{H}_S$,

$$\langle \psi_i | P | \psi_j \rangle = \langle \psi_i | P S_i | \psi_j \rangle = -\langle \psi_i | S_i P | \psi_j \rangle \tag{6}$$

so

$$\langle \psi_i | P | \psi_j \rangle = 0 \tag{7}$$

Thus, as above, if $\{E_a\}$ is a set of Pauli group errors, if we consider the products $E_a^\dagger E_b$ and these anti-commute with at least one of the generators of $\mathcal{S}$, then we have satisfied the error correcting criteria for these errors:

$$\langle\psi_i|E_b^\dagger E_a|\psi_j\rangle = 0 \tag{8}$$

If these elements are themselves elements of the stabilizer: $E_a^\dagger E_b \in \mathcal{S}$, then

$$\langle\psi_i|E_b^\dagger E_a|\psi_j\rangle = \langle\psi_i|S|\psi_j\rangle = \delta_{ij} \tag{9}$$

If for an error set $\{E_a\}$ if all of the products $E_a^\dagger E_b$ either anticommute with generators of the stabilizer $S_1, S_2, \dots, S_r$ or are elements of the stabilizer, then we see the E_a satisfy the quantum error correcting criteria.

In our example where $\mathcal{S} = \langle ZZI, ZIZ\rangle$, we can consider the set of errors $\{III, XII, IXI, IIX\}$. Then the set of products of these errors is $\{III, XII, IXI, IIX, XXI, XIX, IXX\}$. Of these, the first of these III is in the stabilizer. All of the others, however, anticommute with either ZZI or ZIZ. For example $(XXI)(ZIZ) = -(ZIZ)(XXI)$. How do we check whether two Pauli group elements commute or anticommute? Suppose these group elements are $P_1 \otimes P_2 \otimes \cdots \otimes P_n$ and $Q_1 \otimes Q_2 \otimes \cdots \otimes Q_n$. Then we count the locations where P_i and Q_i differ and neither P_i or Q_i is I. If this number is even, then these two Pauli group elements commute and if it is odd then they anticommute.

If a stabilizer group has a minimal number of generators which is $S_1, S_2, \dots, S_r$, what is dimension of the stabilizer subspace? Well take the first stabilizer generator. This stabilizer generator squares to identity, so has ±1 eigenvalues. Further this stabilizer generator has trace zero. Why? Well all of the Pauli operators are trace 0 except I which has trace 2. Since the stabilizer generator can't be identity (unless the stabilizer consists solely of the identity, a case we will disregard) it is tensor product of terms, at least one of which must be a Pauli element not equal to I. Then since $\mathrm{Tr}[A \otimes B] = \mathrm{Tr}[A]\mathrm{Tr}[B]$, this implies that $\mathrm{Tr}[P] = 0$, for all Pauli group elements. Thus if we take S_1 it must have 2^{n-1} eigenvalues $+1$ and 2^{n-1} eigenvalues -1. So $S_1|\psi\rangle = |\psi\rangle$ splits the Hilbert space of our n qubits in half. What happens when we impose $S_2|\psi\rangle = |\psi\rangle$? Well now, define $\frac{1}{2}(I + S_1)$. This is the projector onto the $+1$ eigenvalue eigenspace of S_1. We can thus use $\frac{1}{2}(I + S_1)S_2$ to understand how much of this subspace has eigenvalues of S_2 which are $+1$. Note that $\mathrm{Tr}[\frac{1}{2}(I + S_1)S_2] = 0$. Thus we see that for the 2^n dimensional subspace that satisfies $S_1|\psi\rangle = |\psi\rangle$, a subspace of dimension 2^{n-2} satisfies $S_2|\psi\rangle = |\psi\rangle$. Continuing inductively, we see that each $S_i|\psi\rangle = |\psi\rangle$ cuts the space of the previous $S_1|\psi\rangle = |\psi\rangle, \dots, S_{i-1}|\psi\rangle = |\psi\rangle$ in half.

What does this mean? This means that the dimension of a stabilizer subspace for a stabilizer with r generators is 2^{n-r}. It is also useful to notice that the dimension of the subspace with a fixed set of ±1 eigenvalues of the S_i operators are also of dimension 2^{n-r}. For our example of $S_1 = ZZI$ and $S_2 = ZIZ$, this implies that the stabilizer subspace is 2 dimensional, which is correct.

III. LOGICAL OPERATORS FOR STABILIZER CODES

So, given a stabilizer group $\mathcal{S}$ generated by r elements, we now know that this specifies a subspace of dimension 2^{n-r}. This subspace can be used to encode $k = n - r$ qubits. Is there a nice way to talk about this subspaces as k qubits? Indeed there is.

The centralizer of the stabilizer group $\mathcal{S}$ in $\mathcal{P}_n$ is the set of operators $P \in \mathcal{P}_n$ which satisfy $PS = SP$ for all $S \in \mathcal{S}$. Since our stabilizer group does not contain $-I$, it turns out that the centralizer is equal to the normalizer. The normalizer $\mathcal{N}$ of a $\mathcal{S}$ in $\mathcal{P}_n$ is the set of operators $P \in \mathcal{P}_n$ such that $PSP^\dagger \in S$ for all $S \in \mathcal{S}$. Notice that the stabilizer is automatically in the normalizer, since all of the elements of the stabilizer commute with each other and are all unitary. An important set of operators are those which are in the normalizer $\mathcal{N}$ but not in the stabilizer, $\mathcal{N} - \mathcal{S}$. Why is these elements important? Because they represent *logical Pauli* operators on the k encoded qubits of our subsystem code.

Let's see this for an example and then move on to understanding the logical operators in general. Our example is the stabilizer group generated by $S_1 = ZZI$ and $S_2 = ZIZ$. Then elements of $\mathcal{N} - \mathcal{S}$ are $i^k \times \{XXX, YYX, YXY, XYY, ZII, IZI, IIZ, ZZZ, YYY, XXY, XYX, YXX\}$. Notice that if we take the group generated by the two operators XXX and ZII, each of these elements is equal to such a group member times an element of the stabilizer group. What does XXX do to our code space? Well recall that the stabilizer subspace in this example is spanned by $|0_L\rangle = |000\rangle$ and $|1\rangle_L = |111\rangle$. Thus we see that $XXX|0_L\rangle = |1_L\rangle$, $XXX|1_L\rangle = |0_L\rangle$ and $ZII|0_L\rangle = |0_L\rangle$, $ZII|1_L\rangle = -|1_L\rangle$. In other words XXX acts like an encoded X operation on the subspace and ZII acts like an encoded Z operation on the subspace. Similarly one can calculate that YXX acts as Y on the code subspace. Also notice that these operators preserve the code subspace. Now since all the other elements of the normalizer are either stabilizer elements or products of stabilizer elements, they will also preserve the code subspace.

Following this example, we are motivated to define the group $\mathcal{N}/\mathcal{S}$. This is the normalizer group quotient the stabilizer. We can write every element of $\mathcal{N}$ as RS where S is a stabilizer element and R is not. Then multiplication of these elements is like $R_1S_1R_2S_2 = (R_1R_2)(S_1S_2)$. This defines a multiplication rule for the R_is. This group is the group $\mathcal{N}/\mathcal{S}$. It is possible to show that this group is equal to the Pauli group of size $k = n - r$. This means that it is possible to generate this group by a set of Pauli operators $\bar{X}_1, \bar{Z}_1, \dots, \bar{X}_k, \bar{Z}_k$ (along with a phase $i^k I$). These operators are the encoded Pauli operators on the subspace. (One thing to note is that this choice of division into k different qubits is not unique. However we will rarely deal with the case where there is more than one encoded qubit, so this won't matter much for us.)

Elements of $\mathcal{N}$ represent non-trivial operations on the encoded subspace. They are, then, exactly the type of operators whose action we cannot detect on our quantum error correcting code. Using this fact, it is possible to give a very nice characterization of the quantum error correcting criteria. Suppose that we have a stabilizer $\mathcal{S}$ with normalizer $\mathcal{N}$. Let E_a denote a set of Pauli errors on the qubits. Then if $E_a^\dagger E_b \notin \mathcal{N} - \mathcal{S}$ for all possible error pairs, then E_a is a set of correctable errors. Why is this so? Well there are two cases. One is that $E_a^\dagger E_b$ is in $\mathcal{S}$. Then $\langle\phi_i|E_a^\dagger E_b|\phi_j\rangle = \langle\phi_i|\phi_j\rangle = \delta_{ij}$ since $E_a^\dagger E_b$ acts trivially on the stabilizer subspace. The other case is that $E_a^\dagger E_b$ is not in stabilizer and not in the normalizer. This implies that there must exists an element of the stabilizer S, such that $E_a^\dagger E_b S \neq S E_a^\dagger E_b$ (recall the centralizer and normalizer are the same for stabilizers.) But all elements of the Pauli group either commute or anticommute. This implies that $E_a^\dagger E_b S = -S E_a^\dagger E_b$. By our previous argument this implies that $\langle\phi_i|E_a^\dagger E_b|\phi_j\rangle = 0$.

IV. EXAMPLES OF STABILIZER CODES

Here we present some examples of stabilizer codes.

A. Three Qubit Bit Flip and Phase Flip Codes

The example we have been dealing with, which is generated by ZZI and ZIZ is able to correct a single bit flip. This code is called the three qubit bit flip code. It's stabilizer and logical operators are

Element	Operator
S_1	ZZI
S_2	ZIZ
$\bar{X}$	XXX
$\bar{Z}$	ZII

Similarly we can construct the code which is designed to correct single phase flip errors. Recall that this code was related to the bit flip code by a Hadamard change of basis. This implies that it's stabilizer and logical operators are

Element	Operator
S_1	XXI
S_2	XIX
$\bar{X}$	XII
$\bar{Z}$	ZZZ

Now let's get on to codes than can correct single qubit errors. If a code can correct t errors and encodes k qubits into n bare qubits, we call it a $[n, k, 2t+1]$ code. We've already seen an example of such a single qubit error correcting code, the Shor code. This is an example of a $[9, 1, 3]$ quantum code. It turns out, since this code is really a concatenation of the bit flip and phase flip codes, that the Shor code is also a stabilizer code. In fact we see that in the Shor code, we use three three qubit bit flip codes, plus a single phase flip code on the encoded information. From this it is easy to deduce what the stabilizer of the Shor code is.

Element	Operator
S_1	$ZZIIIIIII$
S_2	$ZIZIIIIII$
S_3	$IIIZZIIII$
S_4	$IIIZIZIII$
S_5	$IIIIIIZZI$
S_6	$IIIIIIZIZ$
S_7	$XXXXXXIII$
S_8	$XXXIIIXXX$
$\bar{X}$	$XXXXXXXXX$
$\bar{Z}$	$ZZZZZZZZZ$

Notice that this is a degenerate code for single qubit errors: operators like $ZZIIIIIII$ act as identity on the code space even though they are products of two single qubit errors.

Now the nine qubit code that Shor came up with is a rather large code (although it has a certain beauty to it in its simplicity.) What is the smallest code which can correct a single qubit error? This code is the five qubit quantum error correcting code. It is specified by

Element	Operator
S_1	$XZZXI$
S_2	$IXZZX$
S_3	$XIXZZ$
S_4	$ZXIXZ$
$\bar{X}$	$XXXXX$
$\bar{Z}$	$ZZZZZ$

Notice that this code has stabilizer generators which are related to each other by a cyclic shift of the qubits. This makes it rather easy to remember its stabilizer. The five qubit code is notable in that it is not an example of a CSS code (Calderbank, Shor, Steane): it is not possible to write its generators as operators which consist of all X or all Z (and I) operators. This complicates things for this code when we look at the issue of fault-tolerance, so while this is a nice code, it isn't as nice as we would like. The five qubit code is a $[5, 1, 3]$ code. It is useful to test your ability to spot when elements of the Pauli group commute or anticommute on the five qubit code, since it has nontrivial stabilizer elements.

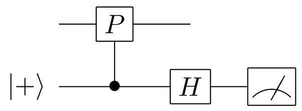
Probably the most studied code and one of the easiest to work with is the Steane code. This is a $[7, 1, 3]$ code. It has the stabilizer

Element	Operator
S_1	$IIIXXXX$
S_2	$IXXIIXX$
S_3	$XIXIXIX$
S_4	$IIIZZZZ$
S_5	$IZZIIZZ$
S_6	$ZIZIZIZ$
$\bar{X}$	$XXXXXXX$
$\bar{Z}$	$ZZZZZZZ$

The Steane code is nice since it is a CSS code. We will use it in a lot of our examples since it has some very nice properties and is easy to discuss but not so big as to disallow us writing down statements about it in explicit form.

V. MEASUREMENT OF PAULI GROUP PROJECTORS

Now that we have defined stabilizer codes, it is useful to discuss how to use them in practice. On essential task we will need to perform is to be able to make a projective measurement onto the $+1$ and -1 eigenvalue eigenspaces of a general Pauli operator P which squares to identity $P^2 = I$. How do we do this? Well we've already basically see a trick for doing this. Consider the following circuit:



(10)

What is the effect of this circuit? Well we can calculate that if our measurement obtains the result $|0\rangle$, then the measurement operator we are measuring on the first qubit is

$$M_0 = \langle 0|_B U|+\rangle_B = \frac{1}{2}(I + P) \tag{11}$$

and if the measurement outcome is $|1\rangle$ then the measurement operator is

$$M_1 = \langle 1|_B U|+\rangle_B = \frac{1}{2}(I - P) \tag{12}$$

Thus we see that this circuit can be used to make a projective measurement onto the $+1$ and -1 eigenvalue eigenspaces of P. This is a very useful primitive for stabilizer codes. Notice, that unlike in our introduction to error correction codes, here in order to make a measurement we need to attach an ancilla qubit and this ancilla qubit will have the result of the measurement in this ancilla qubit. Further note that the measurement is not destructive on the original qubit, in the sense that while the measurement projects onto the subspaces, it leaves the resulting state in the appropriate subspace.

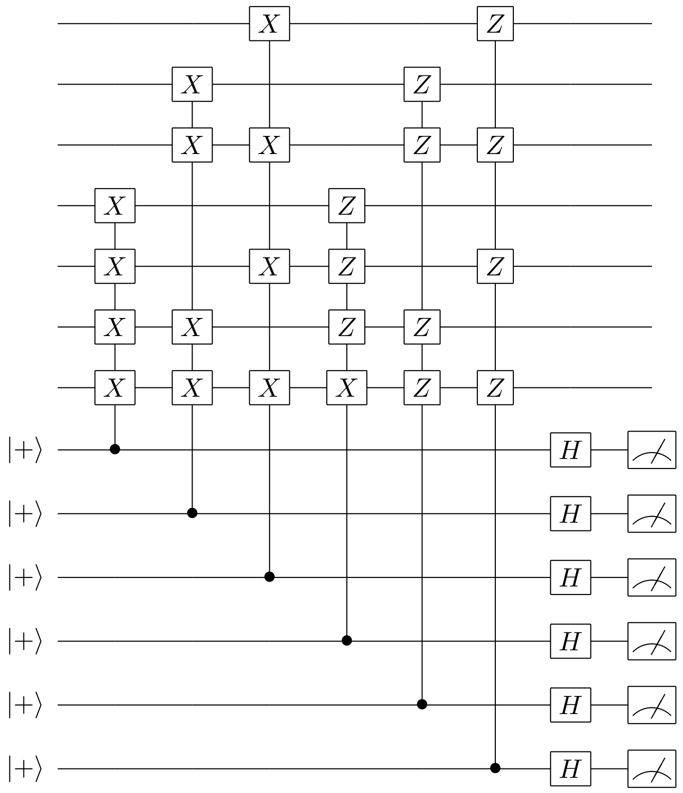
VI. ERROR RECOVERY ROUTINE FOR STABILIZER CODES

Given a stabilizer code, how do perform a procedure which performs quantum error correction?

Well suppose that we have a set of Pauli errors E_a which are correctible via the error correcting routine. Now we know that there must be a recovery routine for these errors. What is this recovery routine?

The first possibility for an error E_a is that it is in the stabilizer. In this case there is no need to perform quantum error correction, since the effect of a stabilizer operator on our encoded quantum information is identity $E_a|\psi\rangle = |\psi\rangle$. So that case was easy! A second case is that E_a is not in the stabilizer. Now we know that E_a, since it satisfies the error correcting criteria, must take us to an orthogonal subspace. What is this orthogonal subspace? Well suppose that the stabilizer generators are $S_1, \dots, S_r$. The if E_a is a correctable error not in the stabilizer, it will anticommute with some of these generators S_i. For these generators, the state $E_a|\psi\rangle$ will no longer be in the $+1$ eigenvalue subspace, but will be in the -1 eigenvalue eigenspace. To see this simply note that that $S_i E_a|\psi\rangle = -E_a S_i|\psi\rangle = -E_a|\psi\rangle$ for $|\psi\rangle \in \mathcal{H}_S$. Further if E_a commutes with S_i, then it does not change the eigenvalue of S_i. Thus we see that the subspace the error sends us to is labelled by the ± 1 eigenvalues of the stabilizer operators. Thus to perform quantum error correction on a stabilizer code, it is enough to make measurements which project onto the ± 1 eigenvalue eigenspaces of the stabilizers. From this information, one can deduce what the error was and then apply an appropriate recovery operation. We call the values of the measured stabilizer generators the syndrome of the error. Now having diagnosed what the error is by measuring the syndrome, one then applies the appropriate Pauli operator to reverse the error. One interesting fact is that, for degenerate codes, there are often multiple errors corresponding to a syndrome. In this case, however one just reverses one of these errors and, one is guaranteed that the net effect of this procedure is to either apply I or a stabilizer element, which is the same as applying I to the code space.

Now how do we perform a measurement onto the ± 1 eigenvalue eigenspace of the S_i operators? Well we can measure them using the circuit described above for measuring Pauli operators. Another important fact to note is that, since the stabilizer generators all commute with each other, measurements which project onto their eigenstates can be simultaneously performed. Here, for example, is the syndrome measurement circuit



(13)

The top seven qubits are the encoded qubits. The bottom six qubits are the ancilla registers which will hold the syndrome operators.

VII. PREPARATION AND MEASUREMENT OF STABILIZER STATES

How do we prepare a stabilizer state? Well this is rather easy. For example, suppose we just prepare the $|0^n\rangle$ state for our bare, unencoded qubits. Now if we measure the syndrome, and apply the error correcting recovery procedure, then we are guaranteed to be in the code subspace. Now if, say we have one encoded qubit, we can measure in the encoded $|0\rangle$, $|1\rangle$ basis by measuring the eigenvalues of the $\bar{Z}$ operator (using the tricks we learned above.) If this outcome is $+1$ we have prepared the encoded $|0\rangle$ state. If this outcome is -1 then we have prepared the encoded $|1\rangle$ state and application of the $\bar{X}$ operator will then turn this into the $|0\rangle$ state. For multiple encoded qubits a similar procedure can be enacted. Thus we have seen how to prepare encoded stabilizer states starting in, say, the encoded $+1$ eigenstates of the $\bar{Z}$ operator. Similar procedures apply for preparation of eigenstates of the other encoded Pauli operators. Thus we have seen how to prepare these simple states.

What about more general states? Well those will be harder. We will discuss some particular examples of these preparation procedures when we talk about fault-tolerant quantum computation.

And a further note. We could also ask how to take a generic unencoded qubit and then design a circuit for encoding into a stabilizer code. But this primitive: encoding into a code, is not a very useful one for our purposes (building a quantum computer.) Why? Well because we really don't ever want to work with unencoded quantum computation. We always want to work with encoded quantum information. Every once in a while you will come across an argument against quantum computing which discusses a decoherence mechanism which always leads to err on a single qubit. Such arguments miss the point, simply because we don't ever work with quantum information in a single qubit: we always work with encoded quantum information.

Finally we can ask how do we perform measurements on our stabilizer states. In particular suppose we have k qubits and we want to measure in the encoded computational basis. Then we can do this by performing our Pauli measuring circuit for the encoded operators $\bar{Z}_i$. It is thus simple to see how to measure encoded Pauli operators on our qubits.

So we have seen how to prepare computational basis states, and measure computational basis states for our encoded information. The next obvious question is how do we perform quantum computation, i.e. unitary gates, on our encoded quantum information. We will take this up in the next section.

VIII. GATES ON STABILIZER CODES

How do we perform unitary transforms on quantum information encoded into a stabilizer code? Well in some sense this question is trivial: just define unitary operations which carry out the desired unitary on the encoded subspace. On the other hand, this observation is not very useful for reasons that will become apparent when we talk about fault-tolerant quantum computation. This here we will discuss a particularly beautiful and useful set of gates which we can implement on our stabilizer code.

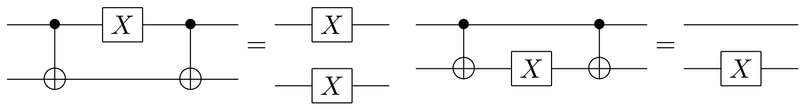
Actually we've already seen some encoded operations on our code: the Pauli operators on our code. These operations were enacted on our encoded quantum information by the application of some Pauli group element. But in this section we will go beyond these simple gates.

A. The Clifford Group

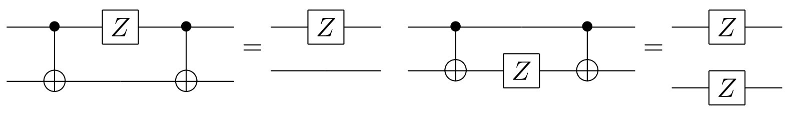
Consider our Pauli group on n qubits, $\mathcal{P}_n$. This is a subgroup of the unitary group on n qubits, $U(2^n)$. An important group, and we'll see why in a second, is the Clifford group (in an abuse of mathematical nomenclature I won't even begin to try to undo.) The Clifford group is the normalizer of the Pauli group in the unitary group on n qubits. In other words, it is the group of operators N which satisfy $NPN^\dagger \in \mathcal{P}_n$ for all $P \in \mathcal{P}_n$. What is an example of such an operation? Our good friend the Hadamard operator. In fact, we can easily show that

$$HIH^\dagger = I \quad HXH^\dagger = Z \quad HYH^\dagger = -Y \quad HZH^\dagger = X \tag{14}$$

Thus a Hadamard on a single qubit is an operator in the normalizer of the Pauli group in $U(2^n)$. Indeed, multiple Hadamards are also in this normalizer. What are other examples? Well elements of $\mathcal{P}_n$ themselves are obviously in the normalizer. Perhaps the first really interesting example beyond the Hadamard operator is the controlled-NOT operation. The effect of the controlled-NOT on Pauli operators is so useful we will write the action down here:



(15)



(16)

Note that we can deduce the action of conjugating a controlled-NOT about Y operations by simply applying the X and Z conjugations in turn. Thus we see that the controlled-NOT operation is indeed in the Clifford group.

A great question to ask is what is needed from the Clifford group on n qubits to generate the entire group. In fact, the Clifford group on n qubits can be generated by H and the controlled-NOT, C_X, and one extra gate,

$$S = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}. \tag{17}$$

Note that this gate acts as $SXS^\dagger = Y$, $SYS^\dagger = -X$, and $SZS^\dagger = Z$.

[insert proof of Clifford gate generation here]

B. Clifford Group Gates on Stabilizer Codes

Suppose that we have a stabilizer code generated by the stabilizer operators $S_1, \ldots, S_r$. Now we saw that elements of the normalizer of the stabilizer group in the Pauli group were logical operators on our stabilizer code. What if we now examine the normalizer of the stabilizer group, but now not just in the Pauli group, but now in the full unitary group $U(2^n)$. These gates will certainly be in the Clifford group, since the stabilizer group is made up of Pauli group operators. But now these operations might perform a nontrivial operation on our encoded qubits.

Let's look at an example. Consider the seven qubit Steane code with generators specified above. Next consider the operator $H^{\otimes 7}$. This operator when conjugated about stabilizer elements will produce another stabilizer element for the Steane code. How do we see this? Well recall that $HXH = Z$ and vice versa. Now notice that if we conjugate

$H^{\otimes 7}$ about one of the generators that is made up completely of X operations, then we obtain a new operator which is made up completely of Z operations. But now notice that there is a symmetry of the Steane code that guarantees that for the generators of the stabilizer made up completely of X operations, there is an equivalent generator formed by replacing the X operations with Z operation. Thus we see that on the generators of the stabilizer code, conjugating by $H^{\otimes 7}$ produces a stabilizer generator. Since all stabilizer elements are made as products of stabilizer generators, this implies that the stabilizer group is preserved under conjugation by $H^{\otimes 7}$.

So now the question arises as to what $H^{\otimes 7}$ does on our logical encoded qubit? Recall that the logical operators on the Steane code are $\bar{X} = X^{\otimes 7}$ and $\bar{Z} = Z^{\otimes 7}$. We can then see that $H^{\otimes 7}\bar{X}H^{\otimes 7} = \bar{Z}$ and $H^{\otimes 7}\bar{Z}H^{\otimes 7} = \bar{X}$. Thus we see that $H^{\otimes 7}$ acts in the same way on the encoded quantum gates as a single H does on a single Pauli. In fact this is enough to imply that $H^{\otimes 7}$ is exactly the Hadamard gate on our encoded quantum information.

More generally, then, we can consider elements of the Clifford group which, when conjugated about the stabilizer generators produce elements of the stabilizer. Such elements act to preserve the stabilizer subspace, since $NS_iN^\dagger = S_j$, and therefore $N|\psi_C\rangle = NS_i|\psi_C\rangle = NS_iN^\dagger N|\psi_C\rangle = S_jN|\psi_C\rangle$. Thus $S_j(N|\psi_C\rangle) = (N|\psi_C\rangle$, so again $N|\psi_C\rangle$ is still stabilized (further note that $NS_iN^\dagger$ preserves the group multiplication rule of the stabilizer and thus $NS_iN^\dagger$ if N is in the normalizer are again generators of the stabilizer group.) Further these gates also act on the encoded qubits as elements of the Clifford group on these encoded qubits.

It is important to note that when it comes to Clifford group elements, not all stabilizer codes were created equal. In particular implementing different Clifford group elements is often much more difficult on some codes than on other codes. This fact is one reason for choosing different stabilizer codes.

C. The Gottesman-Knill Theorem

Our discussion of the Clifford group brings out a particularly nice theorem due to Gottesman and to Knill. Suppose that we have n qubits in the $|0\rangle^{\otimes n}$ state. Now examine the operators Z_i which is the Z operator on the ith qubit tensored with I on all other qubits. Then $Z_i|0\rangle^{\otimes n} = |0\rangle^{\otimes n}$. So if we take our stabilizer to be generated by the Z_i, then this code stabilizes a single state, the $|0\rangle^{\otimes n}$ state. Notice that instead of specifying the state, we could simply specify the list of operators Z_i. Now suppose that we apply Clifford group operators to our state. We could, of course, check how this changes our state. But then we might end up with states which have exponentially many different amplitudes and therefore this might quickly become intractable. But notice that if we have a state stabilized by the n stabilizer generators $S_1, \dots, S_n$, then after the application of a Clifford group operator N, the new state will be stabilized by a new stabilizer $S_1', \dots, S_n'$ where $S_i' = NS_iN^\dagger$. Why? Well if $|\psi\rangle$ is stabilized by the $S_1, \dots, S_n$ operators, then $N|\psi\rangle = NS_i|\psi\rangle = NS_iN^\dagger N|\psi\rangle$ Or $S_i'(N|\psi\rangle) = (N|\psi\rangle)$ showing that the new state is stabilized bye the S_i' operators.

Now return to our circuits made up completely of Clifford group gates. Our circuit initially has the stabilizer $S_i = Z_i$. Each Clifford group gate then changes the state to a state stabilized by a new stabilizer. If our circuit contains small Clifford group gates, or Clifford group gates whose action we can easily update our rules on (in fact all Clifford group elements have an efficient canonical form, but this result is beyond the scope of our time here) then we just need to update the stabilizer instead of describing the new generic state. Thus we see that this is an example of what we encountered when we discussed the role of entanglement in quantum speedups: we can efficiently describe the state by given n n qubit Pauli operators. This can be done with only $2n^2 + 4n$ bits of classical information.

This is particularly nice, since, we can use Clifford group elements to generate quite nontrivially looking entangled quantum states. This is the root of the reason we said before that entanglement is necessary for a quantum speedup over classical computers, but not sufficient. But we can even go beyond just describing the state at any given point after our applications of the Clifford, we can actually describe the result of measurements in the computational basis on these states!

How do we do this? Well suppose we want to measure the first qubit, i.e. determine the eigenvalue of Z_1 and that we currently have a state stabilized by $S_1, \dots, S_n$. What we do is we can check whether Z_1 commutes or anticommutes with each of these stabilizer elements. Now if Z_1 commutes with all of the elements of the stabilizer, then Z_1 or $-Z_1$ is, in fact, in the stabilizer. Then making a measurement of the eigenvalues of Z_1 simply corresponds to outputing the appropriate sign of this stabilizer element. Further note that this measurement does not change thes state, since the state is an eigenstate of the observable we are measuring. What if Z_1 does not commute with some of the elements of the stabilizer. We can assume, then, that it anticommutes with one of the stabilizer generators and commutes with the rest, since if a it anticommutes with a two stabilizer elements, S_i and S_j, then it commutes with S_iS_j. Now in this case, where Z_1 anticommutes with one S_i what are the probabilities of the outcomes corresponding to the $+1$ and -1 eigenvalues of Z_1? Note that

$$Pr(+1) = \langle\psi|\frac{1}{2}(I + Z_1)|\psi\rangle = \langle\psi|\frac{1}{2}(I + Z_1S_i)|\psi\rangle = \langle\psi|\frac{1}{2}(I - S_IZ_1)|\psi\rangle = \langle\psi|\frac{1}{2}(I - Z_1)|\psi\rangle = Pr(-1) \quad (18)$$

Thus the probability of the two outcomes are equal to $\frac{1}{2}$. How does the state change after this this outcome? Well we can describe this new state as being stabilized by the original stabilizer elements that commuted with Z_1 plus, instead of S_i, $\pm Z_1$ corresponding to the two possible outcomes for our measurement result. Similar results can be obtained for measuring in the other computational basis elements.

Thus we've seen that we can efficiently simulate elements from the Clifford group, starting in the $|0\rangle^{\otimes n}$ state and then performing measurements in the computational basis. This is, as we noted above, fairly astounding, given that the states involved in such computations can be pretty nontrivial looking states.

Acknowledgments

The diagrams in these notes were typeset using Q-circuit a latex utility written by graduate student extraordinaires Steve Flammia and Bryan Eastin.