Lecture 4: Transversal gates

- T injection circuit and universality.

1. Bring a T state = $\ket 0 + e^{i \pi /4} \ket 1$ on an ancilla.
2. Measure ZZ on the pair of the ancilla and a data, target qubit.
3. Measure X on the ancilla.
4. Apply correction based on the measurement outcomes.

Exercise

Write down the post-measurement states for all four possible measurement outcomes {00,01,10,11}, and determine necessary correction operators. Make sure to ignore global phase factors!

- Clifford twirling -- simplification in error model

The biggest reason that Clifford + T gate set is dominating fault-tolerant quantum computing is that every component has a relatively straightforward fault-tolerant protocols. Pauli stabilizer codes admit gadgets for Clifford operations. Having a logical qubit that encodes a high-fidelity T state is often a separate problem.

The approach that Knill, Bravyi, and Kitaev pioneered is to consider very good Clifford operations first, so that we may assume perfect Clifford operations when it comes to T states.

The next observation is that the T state is an eigenstate of a Clifford operator.

Definition: Clifford hierarchy.

The level 1 set is the set of all Pauli operators. Modulo phase factors, this set is an abelian group. Moreover, a vector space over the binary field.

Inductively, the level k set is defined by the set of all unitaries U such that $UPU^\dag$ is a member of level k-1 set for every Pauli P. By definition, higher level sets are supersets of a lower level set.

The level 1 is exceptional. The level 2 is the Clifford group; it is group because it is an automorphism group. Starting from level 3, the set cease to be a group.

So, $T = diag(1,e^{i\pi/4})$ is in the level 3, and the T state $\ket T = T \ket +$ is stabilized by $TXT^\dag$, which is a Clifford operator. The orthogonal state is $T \ket - = Z \ket T$.

Now consider the problem of distilling a T state. A general state's density matrix is 2-by-2. If we write it in the basis of $\ket T$ and $Z\ket T$, then it should be $\diag(1-\eps, \eps)$ plus some offdiagonal terms. What happens if we apply the stabilizer $TXT^\dag$? Tautologically, nothing happens to $\ket T$ the eigencomponents. But the offdiagonal entries are affected. What if we apply the stabilizer according to a random coin but discard the coin afterwards? A density matrix captures any probability distribution, so the post-channel state is precisely the original density matrix with the offdiagonals removed. If you are confused how our lack of knowledge (no access to the coin) affects the density matrix, think operationally. Any observer that has no access to the coin will not be able to distinguish our diagonal density matrix from the result of our toss-apply channel, so our diagonal density matrix is the correct state.

This technique to chop off offdiagonal elements is called Clifford twirling. After this twirling, our error model is now a stochastic Z error.

Exercise:
If the T injection circuit is provided with a Z-faulty T state, then what is the action on the data qubit?

- Distillation protocols and triorthogonality.

Since we assume perfect Cliffords by which we twirl any input T state/gate, the protocols role is to catch any Z errors that could happen at T gates. If the underlying code

1. admits transversal T as a logical operation and implements logical T gates on the logical qubits, and
2. Distance against Z errors is high

Then, we would have a higher fidelity T states.

There is a class of CSS codes that empowers distillation protocols, called triorthogonal codes whose X stabilizer group and X logical operators form a triorthogonal matrix.

Definition: A binary matrix that is (k+s)-by-n is triorthogonal if
1. First k rows have odd weight, and last s rows have even weight.
2. The bitwise AND of any pair of rows has even weight. (orthogonal)
3. The bitwise AND of any triple of rows has even weight. (triorthogonal)

The stabilizer group of a triorthogonal code is determined by a triorthogonal matrix. The remaining ambiguity is in the choices of Z-logical operators.

The calculation with triorthogonal codes is not always easy in the sense that it is rather difficult to see how the stabilizer group evolves under transversal T gates. Somewhat surprisingly it becomes easier to examine the code state vectors that are in the eigenstate of logical Z with eigenvalues +1. We have a basis for the logical space thanks to the X logical operators.

For simplicity, let us consider level-3 divisible subspace.

Definition: A binary vector subspace V on m qubits with a coefficient vector t \in \ZZ_8^m with all odd entries is called divisible at level 3, if for any vector v of V the sum of the components of t where v is 1 is 0 mod 8. That is, you think of the entries of v as integers 0 and 1 and take a dot product with t, and calculate the result mod 8, then you must get zero.

With t being the all-one vector, the condition is more often called triply even. We allow odd coefficients that may have some interesting signed components.

Exercise: Show that the generating matrix for a level-3 divisible subspace is triorthogonal where every row has even weight.

The exercise above should be easy, but an interesting question is the converse:

Lemma: A subspace spanned by the rows of a binary matrix M is level-3 divisible with respect to a coefficient vector t if

0. The vector t has odd entries,
1. The dot product between t and any row is 0 mod 8,
2. The dot product between t and bitwise AND of any pair of rows is 0 mod 4,
3. The dot product between t and bitwise AND of any triple of rows is 0 mod 2.

Proof)

An arbitrary vector in the subspace has components that is mod 2 sum of components of the basis vectors. We need to express this mod 2 sum by arithmetic mod 8. How do you write the function of w that is 0 if w is an even integer and 1 if w is odd? One choice is $(1 - (-1)^w)/2$. We can write it as $(1-(1-2)^w)/2$. By binomial theorem,

$$\sum_{p=1}^w \binom{w}{p} (-2)^{p-1}.$$

Set w = the number of 1s in a list of bits $(y_1, y_2,\ldots)$, then $\binom{w}{p}$ may be equated with $\sum_{\text{all unordered p-tuples of the list}}$ (Bitwise AND of the tuple) $= \sum_{a_1 < a_2 <\ldots< a_p } y_{a_1} \ldots y_{a_p}$.

This means that
$\sum_j y_j \bmod 2 = \sum_{p=1}^w (-2)^{p-1} \sum_{\text{all unordered p-tuples}}$ (product of the tuple)

That is, we derived a formula that expresses the mod 2 sum of bits as a polynomial in the components with integer arithmetic. Taking mod 8, we are left with a cubic formula

$$\sum_j y_j \bmod 2 = \sum_j y_j - 2 \sum_{a < b} y_a y_b + 4 \sum_{a<b<c} y_a y_b y_c \bmod 8.$$

Weighting this formula with t and summing over components, we complete the proof.

- The triorthogonal code defined by a level-3 divisible subspace

The resulting CSS code from a level-3 divisible subspace encodes no logical qubit. Nonetheless let us verify that the code space, the unique code state, remains invariant under a transversal T gate. What is the expansion of the code state? Starting with product state $\ket{0^{m}}$, all Z-stabilizers are automatically satisfied. Applying X stabilizers, we make superpositions and that is the unique code state.

Apply T^{t_k} on qubit k where the qubits are in the unique code state. This is a product of single-qubit unitaries. In the expansion, each product state summand acquires some phase factor by the layer of T's, but by the divisibility all these phases cancel! Hence, the unique code state remains invariant and so is the code space projector.

- Induced logical T

Now suppose the generating matrix for a level-3 divisible space has in the upper-left corner a block of k-by-k identity matrix and zero block beneath the identity block. Think of the k qubits corresponding to the first k columns as ancillas, and all the rest as a code qubits. The ancillas don't have any X stabilizers, and we choose to put no Z stabilizers there either. Choose Z logical operators in the code side by the same bit string as the first k rows of the matrix in the code side. Then, the overall system, the code and the ancillas are in the state where each ancilla qubit is in the Bell state with a logical qubit of the code. So, abstracting away all the code structure, we have k Bell pairs.

We know that certain transversal T induces identity action. But on the ancillas, the action of T state is some nonClifford operation, so the underlying Bell state must be cancelling the action. How is that possible? It is a special property of the Bell state that a unitary on one party has the same effect as its transpose acting on the other party. So, the transpose of T is acting on the code's logical qubit, preserving the code space.

In conclusion, if a triorthogonal matrix can be complemented by an identity matrix such that the extended matrix defines a level-3 divisible subspace, then the triorthogonal code admits a nontrivial nonClifford logical operation by some transversal T gate.

- 3d color code and transversal T

An infinite family of examples is provided by so-called 3d color code. It is a CSS code defined on a system of qubits. The geometry of qubits is as follows. Suppose there is a 4-colorable cellulation such that no neighboring 3-cells have the same color. A qubit is place on each vertex. An X-stabilizer is defined for each 3-cell. A Z-stabilizer is defined for each 2-cell.

Let us show that the X-stabilizers define a level-3 divisible subspace, using the lemma.

Observe that the vertices are bipartite; the 1-skeleton, the collection of vertices and edges is a bipartite graph. Why? An equivalent condition for a bipartite graph is that every cycle has even length. A cycle is a homological sum of plaquettes, so it suffices to check that every plaquette has an even number of boundary edges. A plaquette is shared between exactly two 3-cells. A boundary edge of a plaquette is shared among exactly three 3-cells. So, as we go along the boundary of a plaquette the color of the 3-cell that identifies an edge must alternate between the two that are not equal to the colors of the 3-cells that define the plaquette. This shows that the plaquette has an even number of edges.

Using the bipartition, we define a coefficient vector with entries +1 and -1.

We have to check the three conditions of the lemma. A nonzero triple overlap occurs along an edge, which has exactly one +1 and one -1. A nonzero pair over occurs along a plaquette, which we have shown to have an even number of edges, along which there are an equal number of +1s and -1s. It remains to check that each 3-cell has an equal number of +1 and -1. This boundary 2-sphere has 2-cells that are 3-colored by the color of the 3-cell of which the 2-cell is a face. Take one color, and associate each vertex to the adjacent 2-cell of the chosen color. Since two 2-cells of the same color are separated, this is a partition of the vertices. We know that every 2-cell contains an equal number of +1 and -1 vertices, and hence so does the 3-cell.

Exercise: Define level 4 divisibility and show that the 3d color code's X stabilizer group is not level 4 divisible.

- Bravyi-Koenig bound

We have seen that 3d color code space is invariant under certain transversal T gate, a quantum circuit of depth 1. Whether this gives a nontrivial action on the code space depends on the boundary condition. It is known that some boundary condition gives a code that induces logical T.

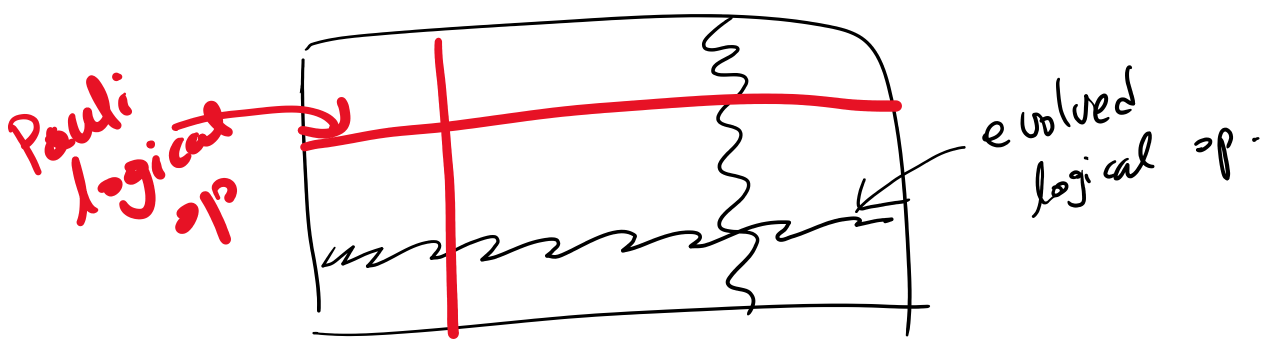
Can we do this in 2d code? We'll see that the answer is no.

Take a Pauli stabilizer code with local stabilizers in which every square region of linear size less than the linear system size is correctable.

Then, by the cleaning lemma, we know that the complement region of the correctable square supports a complete set of logical operators. In our setup, the complementary region is a union of thin strips. These thin strips can be place anywhere in the system.

Now suppose that there is some unitary circuit of constant depth induces some logical operation. In particular, it preserves the code space. As we have studied, the logical operator evolves into an operator whose support is not too different from the original's. Applying this fact to Pauli logical operator supported on the thin strip, we see that evolved logical operator, which determines the action of the circuit on the code space, is also supported on the (slightly fattened) union of strips.

To examine the property of the evolved logical operator, let us consider the group commutator with the original Pauli logical operators that are supported on a different union of strips.



All being quantum circuits of constant depth, the commutator is supported on the intersection, which is a union of two small ball-like regions. By assumption the disks that support the commutator are each correctable, and by union lemma, the union is also correctable. So, the commutator can only be a phase.

As a unitary on the code space, the evolved logical operator has group commutator with any Pauli operator that is a phase factor.

Lemma: If a unitary U commutes with every Pauli operator up to a phase, then U is itself a Pauli operator.

Proof) Expand U in the Pauli operator basis. U = \sum_j c_j P_j for some complex coefficient c_j.

Conjugate it by a Pauli. $QUQ^\dag = \sum_j c_j Q P_j Q^\dag$, and the summands differ from the original by a phase. For these phases to be all the same, there can only be one summand because for each j there exists a Pauli operator that commutes with all P_j' but P_j.

Therefore, the evolved logical operator is a logical Pauli operator. So, the induced logical action can only be Clifford.

You can generalize this argument to higher dimensions by taking group commutators over and over again. Each time you take a group commutator, the support shrinks to their intersection and eventually, it becomes a phase. Rolling back the commutator, you conclude that the induced logical action is in the D-th level in the Clifford hierarchy.