Fault-tolerant quantum gates
Ph/CS 219
2 February 2011

Note Title 2/2/2011

Last time we considered the requirements for fault-tolerant quantum gates that act nontrivially on the codespace of a quantum error-correcting code. In the special case of a code that corrects t=1 error, the requirements are:

-- if the gate gadget is ideal (has no faults) and its input is a codeword, then the gadget realizes the encoded operation U acting on the code space.
-- if the gate gadget is ideal and its input has at most one error (is one-deviated from the codespace), then the output has at most one error in each output block.
-- if the gate has one fault and its input has no errors, then the output has at most one error in each block (the errors are correctable).

We considered the Clifford group, the finite subgroup of the m-qubit unitary group generated by the Hadamard gate H, the phase gate P (rotation by Pi/2 about the z-axis) and the CNOT gate. For a special class of codes, the generators of the Clifford group can be executed *transversally* (i.e., bitwise). The logical U can be done by applying a product of n U (or inverse of U) gates in parallel (where n is the code's length). If we suppose that the number of encoded qubits is k=1, then:

-- the CNOT gate is transversal for any CSS code.
-- the H gate is transversal for a CSS code that uses the same classical code to correct X errors and Z errors.
-- the P gate is transversal if the stabilizer generators have weight a multiple of 4, and the logical Pauli operators X and Z have weight = 1 (mod 4), or have weight = 3 (mod 4). (In the latter case we do the transversal inverse of P to execute the logical P.)

In particular, Steane's [[7,1,3]] code has all of these properties, so we can do Clifford group computation transversally for that code. Transversal operations are fault tolerant: they don't propagate errors from one qubit in a block to another qubit in the same block, and a single faulty gate damages at most one qubit in each block.

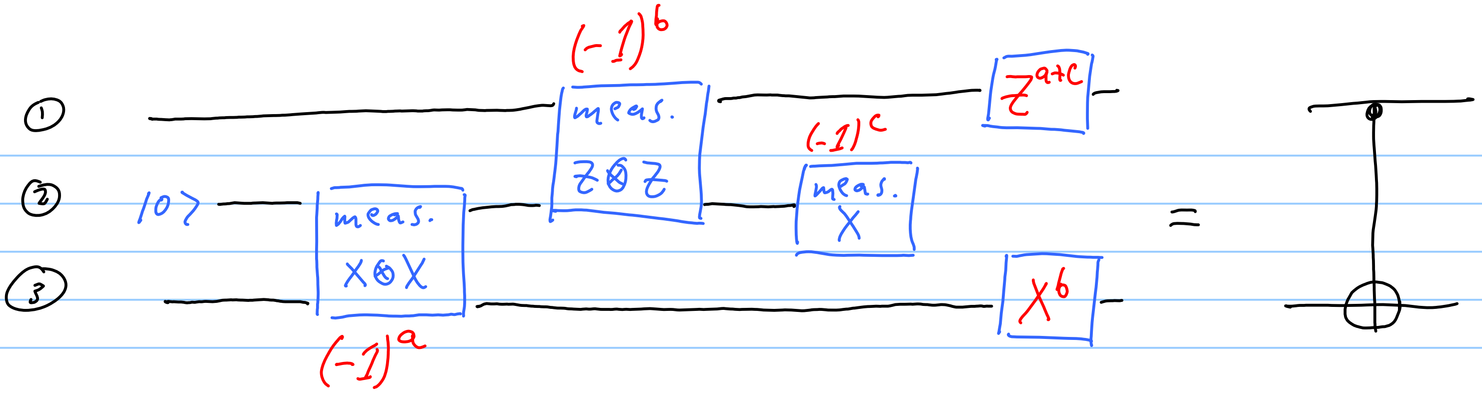
Of course, the Clifford group is discrete so that the Clifford generators are not a universal gate set for quantum computing; in fact Clifford group computation can be simulated efficiently on a classical computer. So we need to consider how to augment our fault-tolerant Clifford gates with another gate that completes a universal set. But before we do that, let's generalize the observation that we can do Clifford group computation fault tolerantly. We will show this is possible for any stabilizer code.

Specifically, we will show (an observation that has useful applications even beyond the study of fault-tolerance):
The following operations suffice for realizing the Clifford group:
-- preparing the Z eigenstate |0>.
-- applying the Pauli operators X, Z to any qubit.
-- measuring weight-1 Pauli operators X, Y and measuring weight-two Pauli operators XX, ZZ, ZX.

This observation is useful in the study of fault tolerance because, for any stabilizer code, any logical Pauli operator can be realized as a Pauli operator (a tensor product of Pauli matrices) which is fault tolerant (*r* faults cause at most *r* errors in the block). Furthermore, we have seen that Pauli operators can be measured fault tolerantly, e.g. by using the cat-state method, repeating measurements, and doing majority voting on the observed outcomes. This is true even for the measurement of the tensor product of two logical Pauli operators in the same code block, since the weight-two logical Pauli operator is also just a tensor product of Pauli operators acting on qubits in the block.

As we already saw last time, since Clifford gates acting by conjugation take Pauli operators to Pauli operators, it is quite convenient to describe these gates in the "Heisenberg picture" -- i.e., in terms of their action on operators rather than states.

First consider execution of the CNOT gate. It can be achieved by the circuit:



Here $(-1)^a$ denotes the outcome of meas of $X\otimes X$, etc., and $Z^{a+c}$ means apply $Z$ conditioned on parity of outcomes of $X\otimes X$ and $X$ meas.

How do we see that it works? We recall the action of CNOT (by conjugation) on Pauli ops.

$$\text{CNOT:}\quad \begin{array}{lcl} XI \to XX & \qquad & ZI \to ZI \\ IX \to IX & \qquad & IZ \to ZZ \end{array}$$

control ↗ ↖ target

We'll consider how $X_1, X_3, Z_1, Z_3$ propagate through the circuit (where $X_i$ denotes qubit # $i$). We'll also keep track of how the stabilizer of the 3-qubit state evolves as the circuit is executed.

Initially, the stabilizer is $M = IZI$, because the middle qubit is prepared in the state $Z_2 = +1$. Then we measure $IXX$, obtaining outcome $(-1)^a$. This measurement commutes with

$$X_1 = XII \qquad X_3 = IIX$$
$$Z_1 = ZII \qquad \text{But not with } Z_3 = IIZ$$

However, because $Z_2 = 1$, we are free to replace

$$Z_3 = IIZ \longrightarrow IZZ,$$

which does commute with $IXX$

The next step is to measure $ZZI$, which does not commute with

$X_1 = XII$. But after the first measurement, the stabilizer of the 3-qubit state has been transformed to $(-1)^a IXX$. Therefore we may make the replacement

$$X_1 = XII \longrightarrow (-1)^a XXX,$$

which *does* commute with $ZZI$. Now in the last step we measure $IXI$, which does not commute with $Z_3 = IZZ$. But after the 2nd meas., the stabilizer is $(-1)^b ZZI$, so we replace

$$Z_3 = IZZ \longrightarrow (-1)^b ZIZ,$$

which does commute with $IXI$. Using the final stabilizer $(-1)^c IXI$, we find the Pauli operators on qubits 1 and 3 have been transformed as:

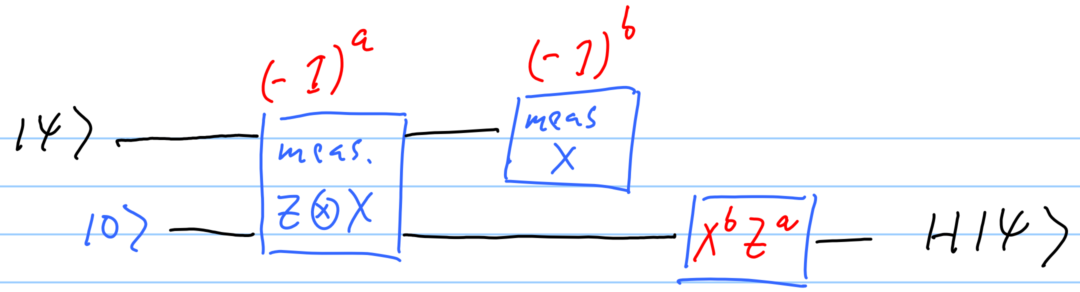
$$X_1 \to (-1)^a XXX \to (-1)^{a+c} XIX$$
$$Z_1 \to ZII$$
$$X_3 \to IIX$$
$$Z_3 \longrightarrow (-1)^b ZIZ$$

This is the action of CNOT with control 1 and target 3, except for the minus signs.

We can flip the sign of $X_1$ (not affecting $X_3$) by applying $Z_1$, if $a+c = 1 \pmod 2$. We can flip the sign of $Z_3$ (without affecting $Z_1$) by apply $X_3$, if $b = 1$. That completes the CNOT gate.

And ... if the state preparations and measurements can be done fault tolerantly, so can the CNOT gate, if we insert an error correction state after each preparation or measurement. Furthermore, we can apply a CNOT gate from any encoded qubit in the control block to any encoded qubit in the target block, as long as we can measure the corresponding weight-two logical Pauli operators.

Next, a circuit for the Hadamard gate H:



Note that in this case the "ancilla" and data switch places.

To check that it works, recall action of $H$ by conjugation:

$$H: \quad X \leftrightarrow Z$$

Initially the stabilizer is $M = IZ$. How do $X_1 = X\,I$, $Z_1 = Z\,I$ propagate through the circuit?

First we measure $ZX$, which does not commute with $X_1$. But using the stabilizer

$$X_1 = X I \longrightarrow X Z$$ which does commute with $ZX$

Then we measure $XI$, which does not commute with $Z_1 = ZI$. But the stabilizer after the first meas. is $(-1)^a ZX$, so we may replace

$$Z_1 = Z I \longrightarrow (-1)^a I X .$$

After the 2nd meas., then, the Pauli ops acting on the input qubit have been transformed as

$$X_1 \longrightarrow XZ = (-1)^b IZ$$
$$Z_1 \longrightarrow (-1)^a IX$$

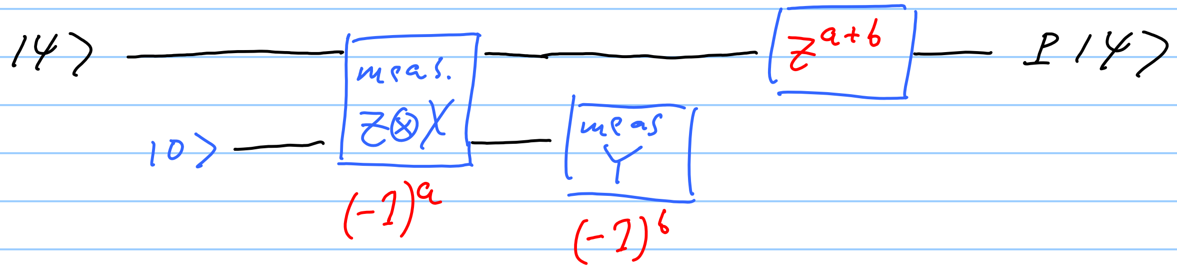
The quantum info initially carried by qubit 1 is transferred to qubit 2, but with a nontrivial operation applied.

We can get rid of the minus sign by applying $X$ if $b=1$ and applying $Z$ if $a=1$.

Finally, we need a circuit for

$P = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$ : $\quad Z \to Z$ $\quad$ (a $\pi/2$ rotation in x-y plane)

$\qquad X \to P X P^{-1} = Y$



Here the operations commute with $Z_1 = ZI$, so clearly $Z_1 \to Z_1$. But how does $X_1 = XI$ propagate through the circuit? The initial stabilizer is $M = IZ$, and first meas. is $Z \otimes X$, as for the Hadamard circuit, so that $X_1 \to XZ$ in the first step.

Since $XZ$ does not commute with $Y$ meas, we use the new stabilizer $(-1)^a ZX$ to replace

$$X_1 = XZ \to (-1)^a (ZX)(XZ) = (-1)^a YY$$

Then, when we measure $Y_2$, we have

$$X_1 = (-1)^a YY \longrightarrow (-1)^{a+b} YI$$

We can remove the minus sign by applying $Z$ (which commutes with $Z_1$) if $a+b \pmod 2 = 1$.

Thus we have the $P$ gate: $X \to Y$

Now we have seen how to apply fault-tolerant Clifford group gates for any stabilizer code (since we can do Pauli operators, Pauli operator eigenstate preparations and Pauli operator measurements fault-tolerantly).

But how do we go beyond the Clifford group and complete a universal fault-tolerant logical gate set? One thing we should recognize is that we cannot expect to be able to do all the logical gates in a universal set transversally. If the transversal operations that preserve the code space are actually universal, then these operations surely are a continuous set, so we can consider the infinitesimal operations in this set --- those that are very close to the identity operation. Because a transversal operation acting on a length-n codeblock is a product of n operations, each of which acts on only one qubit in the codeblock, an infinitesimal transversal operation V has the form

$$V = I + \varepsilon\left(A_1 + A_2 + \cdots + A_n\right)$$

where $A_i$ acts nontrivially only on qubit # $i$ in the block. Since this is a logical operation, $V$ preserves the codespace

$$V\Pi = \Pi V \Pi$$ where $\Pi$ is the projector onto the codespace

Thus $$\left(\Sigma A_i\right)\Pi = \Pi\left(\Sigma A_i\right)\Pi$$

We also know that for a code that can detect weight-one errors

$$\Pi E \Pi \propto \Pi$$ (where $E$ has weight one)

- if we successfully project onto the codespace the error $E$ is reversed. Since each $A_i$ has wt=1

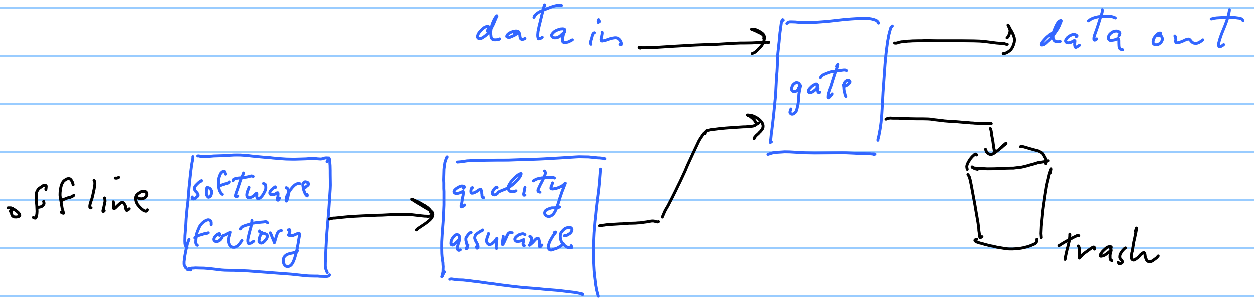
$$\Pi A_i \Pi \propto \Pi$$ for each $i$

Therefore $$\left(\Sigma A_i\right)\Pi = \Pi\left(\Sigma A_i\right)\Pi \propto \Pi$$

and the operation $V$ acting on the code space is the identity. We conclude that continuously varying logical operation cannot be done transversally — at best we are limited to a discrete set.

The problem is that if we could do a logical operation by applying infinitesimal unitaries to each individual qubit in the block, we would be unable to protect the encoded state from small unitary errors acting on qubits.

To complete the universal logical gate set, we will instead borrow the idea we have used to make error correction fault tolerant, and also to realize Clifford gates for arbitrary stabilizer codes. We prepare "offline" a special encoded ancilla state that can be verified, then use joint measurement on the data block and the ancilla to realize the desired gate. The principle behind this strategy is that it is easier to verify a known quantum state than to certify that a known unitary transformation has been properly applied to an unknown state. If the verification of the state fails, we can either repair the state or discard it and attempt to prepare again, without endangering the data.



When we think about how to complete the fault-tolerant gate set, it is useful to keep in mind a hierarchical classification of unitary gates --- the C_r classification.

$C_1$ is the Pauli group

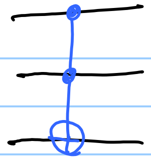
$C_2$ is the Clifford group: the unitaries whose action by conjugation maps a Pauli operator to a Pauli operator

$C_3$ gates acting by conjugation map Pauli operators to $C_2$ (Clifford) operators

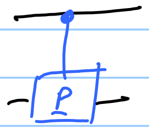
$C_r$ gates acting by conjugation map Pauli operators to $C_{r-1}$ operators

While $C_1$ and $C_2$ are groups, $C_r$ for $r \geq 3$ is not a group (the product of two $C_r$ gates need not be in $C_r$). For example, a $C_3$ gate does not necessarily map a Clifford operator to a Clifford operator. For our purposes what makes the $C_r$ classification useful is that by supplementing the Clifford group generators with a $C_3$ gate we obtain a universal gate set. Examples of $C_3$ gates are

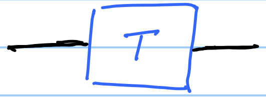
Toffoli gate



Controlled phase



"π/8 gate"



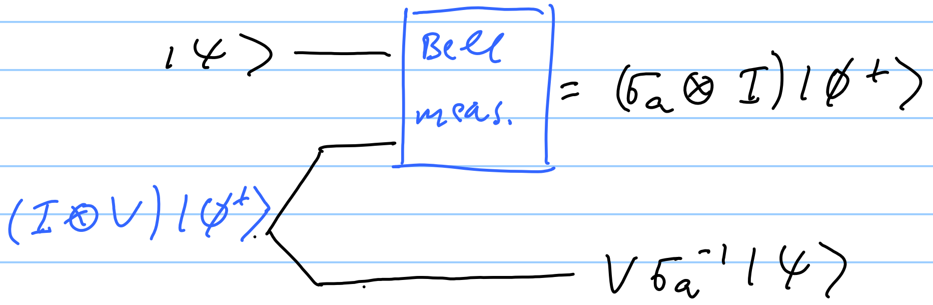
(i.e. $\sqrt{P}$ )

The π/8 gate $T=\begin{pmatrix}1&0\\0&e^{i\pi/4}\end{pmatrix}$ (despite the name) is actually a π/4 rotation about the z axis

The second reason we are interested in $C_3$ gates is that any $C_3$ gate can be realized using:

- preparation of $C_2$ operator eigenstates
- Pauli op. measurement
- $C_2$ gates

} by means of "twisted teleportation"


$|\psi\rangle$
Bell meas.
$= (\sigma_a\otimes I)|\phi^+\rangle$
$(I\otimes V)|\phi^+\rangle$
$V\sigma_a^{-1}|\psi\rangle$

If Bob applied $V^{-1}$ to his qubit this would be standard teleportation — he would apply $\sigma_a$ to recover $|\psi\rangle$

But if instead Bob applies $V\sigma_a V^{-1}$, then Bob has $V|\psi\rangle$. If $V$ is a $C_r$ gate, then $V\sigma_a V^{-1}$ is in $C_{r-1}$. So if Bob can apply $C_2$ gates, and the state $(I\otimes V)|\phi^+\rangle$ can be prepared reliably, then the $C_3$ gate $V$ can be executed.

Furthermore, the state $(I\otimes V)|\phi^+\rangle$ is an eigenstate of a $C_2$ operator — its stabilizer is

$$(I\otimes V)(X\otimes X)(I\otimes V^{-1}) = X\otimes VXV^{-1}$$
$$(I\otimes V)(Z\otimes Z)(I\otimes V^{-1}) = Z\otimes VZV^{-1}$$

So we can prepare by measuring a $C_2$ operator.

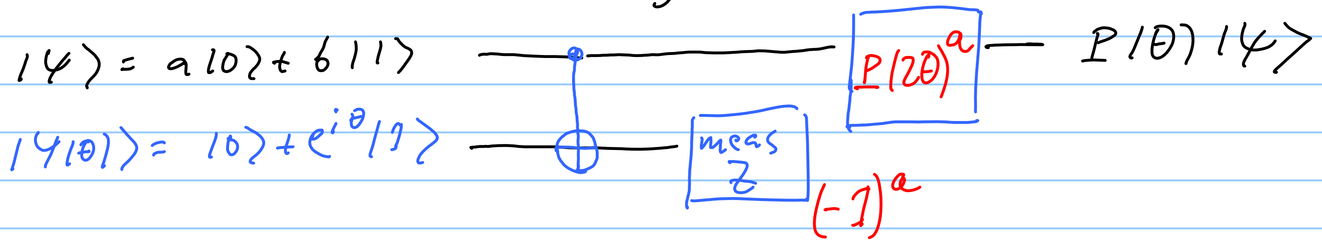
For some $C_3$ gates we can simplify this procedure. For example, to apply $P(\theta) = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{pmatrix}$ (the rotation $\exp(-i\frac{\theta}{2}Z)$, up to a phase)

it is almost enough to prepare $|\psi(\theta)\rangle = |0\rangle + e^{i\theta}|1\rangle$
$= P(\theta)(|0\rangle + |1\rangle)$

This is the $+1$ eigenstate (ignoring normalization)
of

$$P(\theta) X P(\theta)^{-1} = \begin{pmatrix} 0 & e^{-i\theta} \\ e^{i\theta} & 0 \end{pmatrix}$$

In this case the following circuit works:


$|\psi\rangle = a|0\rangle + b|1\rangle$
$|\psi(\theta)\rangle = |0\rangle + e^{i\theta}|1\rangle$
meas Z
$(-1)^a$
$P(2\theta)^a$
$P(\theta)|\psi\rangle$

For the measurement outcome $a=0$, the postmeasurement state is $a|0\rangle + e^{i\theta} b|1\rangle = P(\theta)|\psi\rangle$

For outcome $a=1$, it is $e^{i\theta} a|0\rangle + b|1\rangle = e^{i\theta} P(-\theta)|\psi\rangle$

Thus for the $a=0$ outcome the state after the meas. is what we want. For the $a=1$ outcome we can "fix" the state by applying $P(2\theta)$.

In the case of the $T$ gate $(\theta = \pi/4)$, $T^2 = P$ is a Clifford gate. We can execute $T$ by preparing $|\psi_{\pi/4}\rangle = |0\rangle + e^{i\pi/4}|1\rangle$, measuring, and applying $P$ if necessary. The state to be prepared is the eigenstate of the Clifford operator

$$T X T^{-1} = \begin{pmatrix} 0 & e^{-i\pi/4} \\ e^{i\pi/4} & 0 \end{pmatrix} = \frac{1}{\sqrt{2}}(X+Y).$$

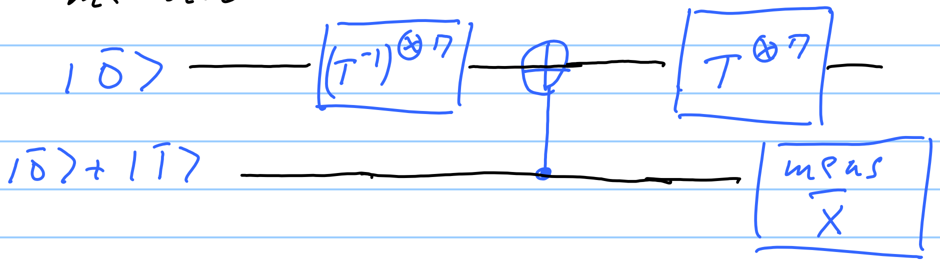
Eigenstates are $T|\pm\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ \pm e^{i\pi/4} \end{pmatrix}$

Consider the case of the [[7, 1, 3]] code. The Clifford group is transversal, except that $P^{\otimes 7} = (\bar{P})^{-1}$ (the inverse of logical $P$). So

$$\bar{T}\bar{X}\bar{T}^{-1} = (T^{-1}XT)^{\otimes 7}$$

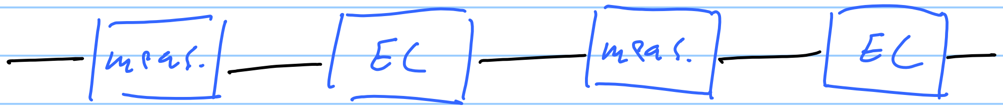
($H, P$ generate single-qubit Clifford group where $H^{\otimes 7} = \bar{H}$ and $P^{\otimes 7} = \bar{P}^*$, so for any single-qubit Clifford transformation $U^{\otimes 7} = \bar{U}^*$.)

We can measure fault tolerantly by the cat state method:



This prepares eigenstate of $\bar{T}\bar{X}\bar{T}^{-1}$ with eigenvalue $\pm 1$ determined by $\bar{X}$ meas of cat state.

To make the meas. fault tolerant, measure twice (followed each time by EC) and accept only if same outcome twice.



After preparing the $+1$ eigenstate of $TXT^{-1}$, we use it to realize the $T$ gate.