Fault Tolerance and Thresholds

Bryan Eastin

Department of Physics and Astronomy
University of New Mexico

May 2006

What do we mean by "quantum computer"?

Quantum computer properties (in theory)

1. General purpose - Not limited to a single class of problems. Universal.
2. Scalable - Operations on an arbitrary number of qubits can be performed.
3. Crash proof - Programs of arbitrary length can be run.
4. Efficient - Resource requirements do not grow exponentially in the the above three parameters.

Our goal is to achieve these properties in an imperfect device.

Knowing your error

Codes with non-zero rate have logical operators.

Logical operators can be considered uncorrectable errors.

Every code has a weakness, else it couldn't store data.

Independence

$$p(i,j) = p(i)p(j)$$

where $p(i)$ is the probability of some error on qubit i.

All error correction depends on knowing something about your error channel.

We will assume that errors occur independently.

Preventing correlated errors

Fault tolerance - An approach to quantum circuit design that seeks to insure that r component failures do not result in s errors in a logical state where $s > r$.

Methods of fault tolerance

- Transversal gates - No operations which couple different qubits within the same block are applied.
- Qubit expenditure - Low error ancillae are constructed.
- Repetition - Error diagnoses are verified.

We will discuss each of these while constructing a fault tolerant procedure for a subset CSS codes.

Constructing a doubly even $[[n, 1, d]]$ CSS code (1)

1. Find a $[2l, l, d+1]$ doubly even self-dual classical code, $\mathcal{C}_D$, s.t. $l = \frac{n+1}{2}$ where $n = 8g - 1$, $g \in \mathbb{N}$.
2. Choose D, the generator, s.t.
$$\mathrm{D}^{i1} = \delta_{il}.$$
3. Consider a new code, $\mathcal{C}$, generated by
$$\mathrm{G}^{ij} = \mathrm{D}^{i,j+1}$$
$$i \in [1, l], \; j \in [1, n].$$

$$\vec{v} \in \mathcal{C} \text{ iff } \vec{v}' = \begin{bmatrix} \text{a} \\ \vec{v} \end{bmatrix} \in \mathcal{C}_D$$

Example (Steane code)

$$\mathrm{D} = \begin{bmatrix} 01010101 \\ 00110011 \\ 00001111 \\ 11110000 \end{bmatrix}$$

$$\mathrm{G} = \begin{bmatrix} 1010101 \\ 0110011 \\ 0001111 \\ 1110000 \end{bmatrix}$$

Constructing a doubly even $[[n, 1, d]]$ CSS code (2)

$\mathcal{C}$ has at least distance d since

$$\vec{v} \in \mathcal{C} \quad \text{iff} \quad \vec{v}' = \begin{bmatrix} a \\ \vec{v} \end{bmatrix} \in \mathcal{C}_D$$

From this expression we can also derive the parity check matrix of $\mathcal{C}$. If $\vec{v} \in \mathcal{C}$

$$0 = \mathrm{D} \cdot \vec{v}' = \delta_{il} a + \sum_{j=1}^{n} \mathrm{D}^{i,j+1} v^j \qquad \forall\, i \in [1, l]$$

$$\rightarrow 0 = \sum_{j=1}^{n} \mathrm{D}^{i,j+1} v^j = \sum_{j=1}^{n} \mathrm{H}^{i,j} v^j = \mathrm{H}^i \cdot \vec{v} \qquad \forall\, i \in [1, m = l - 1]$$

$$\rightarrow \mathrm{H} \cdot \vec{v} = 0$$

Example (Steane code)

$$\mathrm{H} = \begin{bmatrix} 1010101 \\ 0110011 \\ 0001111 \end{bmatrix}$$

Constructing a doubly even $[[n, 1, d]]$ CSS code (3)

Finally, completing our construction

4. Define a quantum code with (note $n \in$ odd)

$$S^{\mathcal{G}} = \left[\begin{array}{c|c} \mathrm{H} & 0 \\ 0 & \mathrm{H} \end{array}\right] \qquad \textit{and} \qquad \begin{array}{l} \bar{X} = X^{\otimes n} \\ \bar{Z} = Z^{\otimes n} \end{array}$$

for stabilizer generators and logical operators.

The matching logical states are (note $1^{,n} \notin \mathrm{H}$ but $1^{,n} \in \mathrm{G}$)

$$|\bar{0}\rangle = 2^{\frac{1-n}{4}} \sum_{v \in \mathcal{C}^{\perp}} |v\rangle \qquad |\bar{1}\rangle = 2^{\frac{1-n}{4}} \sum_{v \in \mathcal{C} - \mathcal{C}^{\perp}} |v\rangle = \bar{X}\,|\bar{0}\rangle$$

Example (Steane code)

$$|\bar{0}\rangle = \frac{1}{\sqrt{8}}\big(|0000000\rangle + |1010101\rangle + |0110011\rangle + |0001111\rangle + |1100110\rangle + |1011010\rangle + |0111100\rangle + |1101001\rangle\big)$$

Apropos: Why transversal gates?

Transversal operations do not couple different qubits in a block.

Figure: (Steane code) Non-transversal encoded T

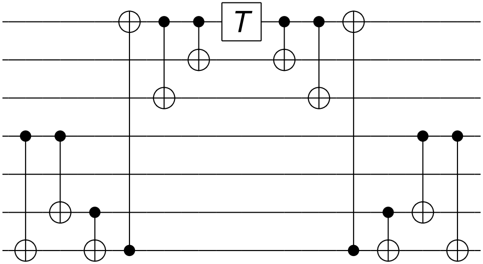
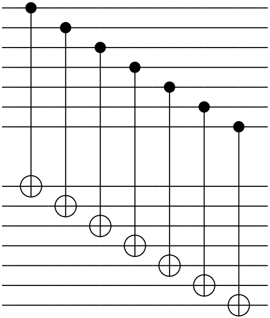


Figure: (Steane Code) Transversal encoded ${}^{C}X$



Apropos: Why transversal gates?

Transversal operations do not couple different qubits in a block.

Figure: (Steane code) Non-transversal encoded T

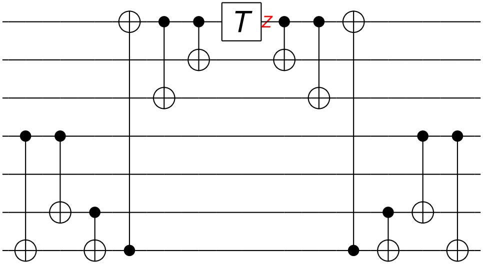
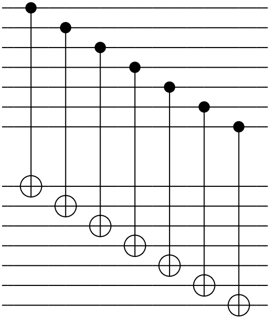


Figure: (Steane Code) Transversal encoded ${}^{C}X$



Apropos: Why transversal gates?

Transversal operations do not couple different qubits in a block.

Figure: (Steane code) Non-transversal encoded T

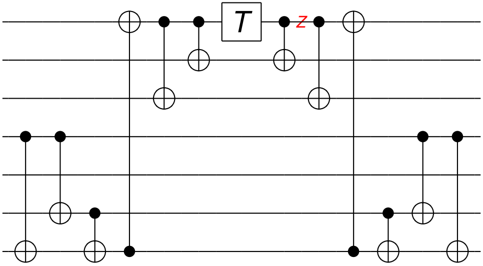
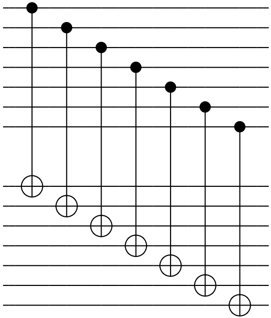


Figure: (Steane Code) Transversal encoded ${}^{C}X$



Apropos: Why transversal gates?

Transversal operations do not couple different qubits in a block.

Figure: (Steane code) Non-transversal encoded T

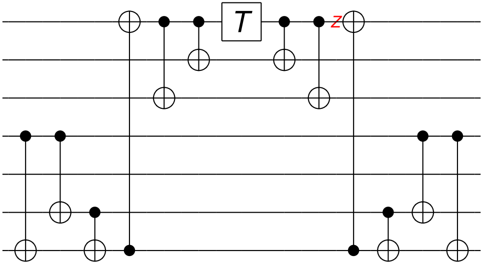
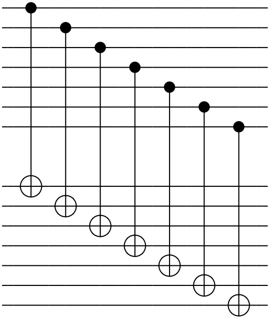


Figure: (Steane Code) Transversal encoded ${}^{C}X$



Apropos: Why transversal gates?

Transversal operations do not couple different qubits in a block.

Figure: (Steane code) Non-transversal encoded T

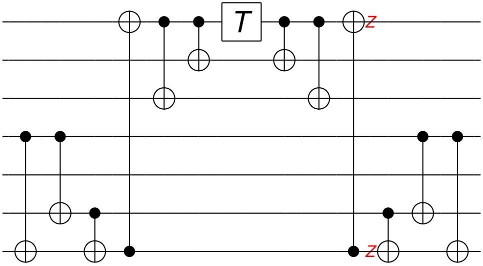
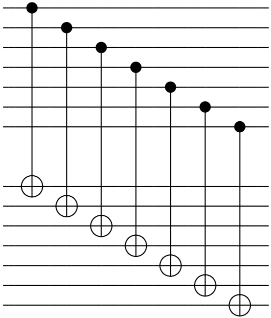


Figure: (Steane Code) Transversal encoded ${}^{C}X$



Apropos: Why transversal gates?

Transversal operations do not couple different qubits in a block.

Figure: (Steane code) Non-transversal encoded T

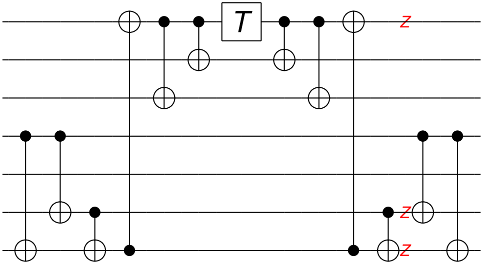
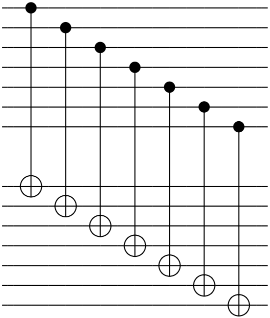


Figure: (Steane Code) Transversal encoded ${}^{C}X$



Apropos: Why transversal gates?

Transversal operations do not couple different qubits in a block.

Figure: (Steane code) Non-transversal encoded T

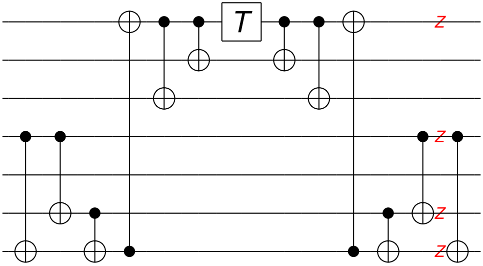
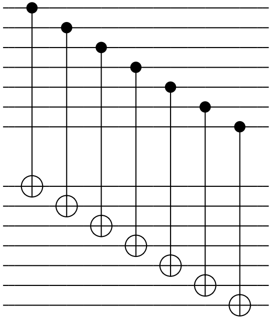


Figure: (Steane Code) Transversal encoded ${}^{C}X$



Apropos: Why transversal gates?

Transversal operations do not couple different qubits in a block.

Figure: (Steane code) Non-transversal encoded T

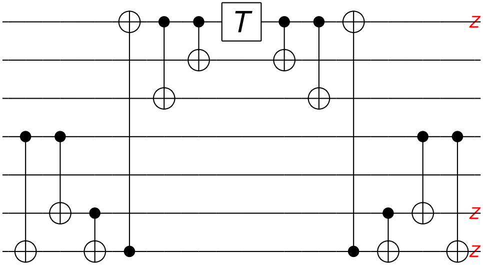
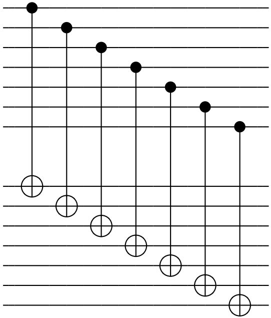


Figure: (Steane Code) Transversal encoded ${}^{C}X$



Apropos: Why transversal gates?

Transversal operations do not couple different qubits in a block.

Figure: (Steane code) Non-transversal encoded T

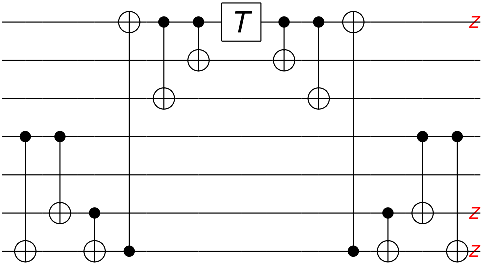
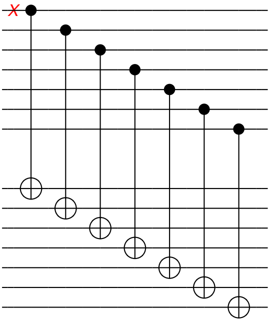


Figure: (Steane Code) Transversal encoded ${}^{C}X$



Apropos: Why transversal gates?

Transversal operations do not couple different qubits in a block.

Figure: (Steane code) Non-transversal encoded T

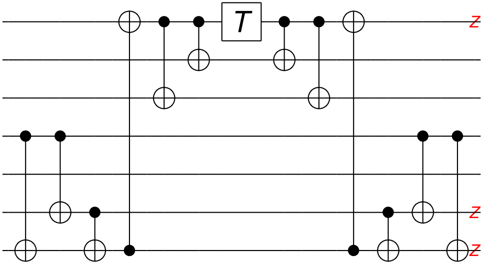
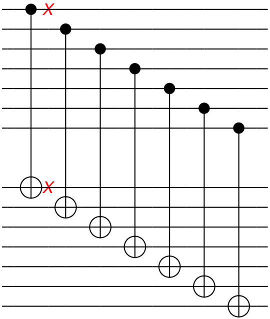


Figure: (Steane Code) Transversal encoded ${}^{C}X$



Apropos: Why transversal gates?

Transversal operations do not couple different qubits in a block.

Figure: (Steane code) Non-transversal encoded T

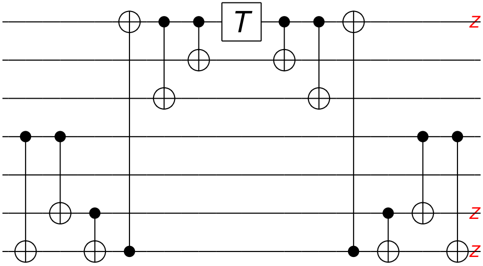
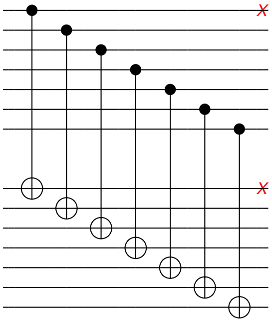


Figure: (Steane Code) Transversal encoded ${}^{C}X$



A Heisenberg approach to logical gates

We have defined $\bar{X} = X^{\otimes n}$, $\bar{Z} = Z^{\otimes n}$, and, implicitly,

$$\bar{Y} = -i\bar{Z}\bar{X} = -i^{n+1}Y^{\otimes n} = -Y^{\otimes n}$$

$$\bar{I} = S^{i} \qquad \text{or any combination thereof.}$$

For further gates we need only check that

$$\bar{U}^{\dagger}\bar{X}\bar{U} = \bar{?} \quad \sim \quad U^{\dagger}XU = ?$$

$$\bar{U}^{\dagger}\bar{Z}\bar{U} = \bar{?}' \quad \sim \quad U^{\dagger}ZU = ?'$$

$$\bar{U}^{\dagger}S^{\mathcal{G}}\bar{U} = S^{\mathcal{G}} \quad \sim \quad U^{\dagger}U = I$$

Clearly, $\bar{Y}$ satisfies this.

Example (Steane code)

$$\bar{Y} = -iX^{\otimes 7}Z^{\otimes 7} = -Y^{\otimes 7}$$

$$\bar{Y}\bar{X}\bar{Y} = -\bar{Y}$$

$$\bar{Y}\bar{Z}\bar{Y} = -\bar{Z}$$

$$\bar{Y}S^{\mathcal{G}}\bar{Y} = (-1)^{4}S^{\mathcal{G}} = S^{\mathcal{G}}$$

Transversal logical H

Let $S^{\mathcal{G}xi}$ and $S^{\mathcal{G}zi}$ be corresponding stabilizer generators with only X's and only Z's respectively.

Show $\bar{H} = H^{\otimes n}$.

Proof

1. $\bar{H}\bar{X}\bar{H} = H^{\otimes n}X^{\otimes n}H^{\otimes n} = Z^{\otimes n} = \bar{Z}$
2. $\bar{H}\bar{Z}\bar{H} = H^{\otimes n}Z^{\otimes n}H^{\otimes n} = X^{\otimes n} = \bar{X}$
3. $\bar{H}S^{\mathcal{G}xi}\bar{H} = H^{\otimes n}S^{\mathcal{G}xi}H^{\otimes n} = S^{\mathcal{G}zi}$
4. $\bar{H}S^{\mathcal{G}zi}\bar{H} = H^{\otimes n}S^{\mathcal{G}zi}H^{\otimes n} = S^{\mathcal{G}xi}$

Example (Steane code)

$$H^{\otimes 7} \cdot X_1X_3X_5X_7 \cdot H^{\otimes 7} = Z_1Z_3Z_5Z_7$$

$$H^{\otimes 7} \cdot X_2X_3X_6X_7 \cdot H^{\otimes 7} = Z_2Z_3Z_6Z_7$$

$$H^{\otimes 7} \cdot X_4X_5X_6X_7 \cdot H^{\otimes 7} = Z_4Z_5Z_6Z_7$$

Transversal logical ${}^{C}X$

Let $S_1^{\mathcal{G}xi}$ denote a stabilizer generator applied to logical qubit 1, and similarly for $S_2^{\mathcal{G}xi}$, $S_1^{\mathcal{G}zi}$, and $S_2^{\mathcal{G}zi}$.

Reminder (${}^{C}X$ properties)

$$
\begin{aligned}
{}^{C}X_{12} X_1 {}^{C}X_{12} &= X_1 X_2 \\
{}^{C}X_{12} X_2 {}^{C}X_{12} &= X_2 \\
{}^{C}X_{12} Z_1 {}^{C}X_{12} &= Z_1 \\
{}^{C}X_{12} Z_2 {}^{C}X_{12} &= Z_1 Z_2
\end{aligned}
$$

Show $\overline{{}^{C}X}_{12} = \bigotimes_{i=1}^{n} {}^{C}X_{i,n+i}$.

Proof

1. $\overline{{}^{C}X}_{12} \bar{X}_1 \overline{{}^{C}X}_{12} = \bar{X}_1 \bar{X}_2$
2. $\overline{{}^{C}X}_{12} \bar{X}_2 \overline{{}^{C}X}_{12} = \bar{X}_2$
3. $\overline{{}^{C}X}_{12} \bar{Z}_1 \overline{{}^{C}X}_{12} = \bar{Z}_1$
4. $\overline{{}^{C}X}_{12} \bar{Z}_2 \overline{{}^{C}X}_{12} = \bar{Z}_1 \bar{Z}_2$
5. $\overline{{}^{C}X}_{12} S_1^{\mathcal{G}xi} \overline{{}^{C}X}_{12} = S_1^{\mathcal{G}xi} S_2^{\mathcal{G}xi}$
6. $\overline{{}^{C}X}_{12} S_2^{\mathcal{G}xi} \overline{{}^{C}X}_{12} = S_2^{\mathcal{G}xi}$
7. $\overline{{}^{C}X}_{12} S_1^{\mathcal{G}zi} \overline{{}^{C}X}_{12} = S_1^{\mathcal{G}zi}$
8. $\overline{{}^{C}X}_{12} S_2^{\mathcal{G}zi} \overline{{}^{C}X}_{12} = S_1^{\mathcal{G}zi} S_2^{\mathcal{G}zi}$

Transversal logical P

Show $\bar{P} = (P^\dagger)^{\otimes n}$.

Reminder

$$P = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$$

$$P^\dagger XP = -iZX$$
$$PXP^\dagger = iZX$$
$$P^\dagger ZP = Z$$

Proof

1. $\bar{P}^\dagger \bar{X} \bar{P} = P^{\otimes n} X^{\otimes n} (P^\dagger)^{\otimes n} = (iZX)^{\otimes n} = -i\bar{Z}\bar{X}$
2. $\bar{P}^\dagger \bar{Z} \bar{P} = P^{\otimes n} Z^{\otimes n} (P^\dagger)^{\otimes n} = Z^{\otimes n} = \bar{Z}$
3. $\bar{P}^\dagger S^{\mathcal{G}_x i} \bar{P} = P^{\otimes n} S^{\mathcal{G}_x i} (P^\dagger)^{\otimes n} = i^{4g} S^{\mathcal{G}_z i} S^{\mathcal{G}_x i} = S^{\mathcal{G}_z i} S^{\mathcal{G}_x i}$
4. $\bar{P}^\dagger S^{\mathcal{G}_z i} \bar{P} = P^{\otimes n} S^{\mathcal{G}_z i} (P^\dagger)^{\otimes n} = S^{\mathcal{G}_z i}$

Example (Steane code)

$$(P^\dagger)^{\otimes 7} X^{\otimes 7} P^{\otimes 7} = (iZX)^{\otimes 7} = -iZ^{\otimes 7} X^{\otimes 7}$$
$$(P^\dagger)^{\otimes 7} Z^{\otimes 7} P^{\otimes 7} = Z^{\otimes 7}$$
$$(P^\dagger)^{\otimes 7} \cdot X_1 X_3 X_5 X_7 \cdot P^{\otimes 7} = (i)^4 Z_1 Z_3 Z_5 Z_7 X_1 X_3 X_5 X_7$$

Transversal $\frac{\pi}{4}$ rotation?

For a universal logical gate set we need something like T, the $\frac{\pi}{4}$ rotation.

$$T = \begin{bmatrix} 1 & 0 \\ 0 & e^{\frac{i\pi}{4}} \end{bmatrix}$$

T is ill suited to our Heisenberg analysis because it is not a Clifford gate (which is why we want it).

The Schrodinger picture suggests a simple transversal implementation of T using quadruply even self-dual classical codes. . .

Transversal $\frac{\pi}{4}$ rotation?

For a universal logical gate set we need something like T, the $\frac{\pi}{4}$ rotation.

$$T = \begin{bmatrix} 1 & 0 \\ 0 & e^{\frac{i\pi}{4}} \end{bmatrix}$$

T is ill suited to our Heisenberg analysis because it is not a Clifford gate (which is why we want it).

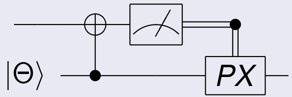
The Schrodinger picture suggests a simple transversal implementation of T using quadruply even self-dual classical codes…
…which do not exist.

Teleporting in the $\frac{\pi}{4}$ rotation

Suppose we have access to the state

$$|\Theta\rangle = \frac{1}{\sqrt{2}}(|0\rangle + e^{\frac{i\pi}{4}} |1\rangle).$$

Teleportation circuit for T



The following shows that the circuit above right applies T.

$$\begin{aligned} |\Psi\rangle |\Theta\rangle &= (\alpha |0\rangle + \beta |1\rangle) \frac{1}{\sqrt{2}}(|0\rangle + e^{\frac{i\pi}{4}} |1\rangle) \\ &\xrightarrow{c_X} \frac{1}{\sqrt{2}} \left[(\alpha |0\rangle + \beta |1\rangle) |0\rangle + e^{\frac{i\pi}{4}} (\alpha |1\rangle + \beta |0\rangle) |1\rangle \right] \\ &= \frac{1}{\sqrt{2}} \left[|0\rangle (\alpha |0\rangle + e^{\frac{i\pi}{4}} \beta |1\rangle) + |1\rangle (e^{\frac{i\pi}{4}} \alpha |1\rangle + \beta |0\rangle) \right] \end{aligned}$$

We know how to encode the gates, but whence comes $|\bar{\Theta}\rangle$?

Preparing $|\Theta\rangle$ via measurement

$|\Theta\rangle$ is a component of many states, including

$$|0\rangle = \frac{1}{2}(|0\rangle + e^{\frac{i\pi}{4}}|1\rangle) + \frac{1}{2}(|0\rangle - e^{\frac{i\pi}{4}}|1\rangle) = \frac{1}{\sqrt{2}}(|\Theta\rangle + Z|\Theta\rangle).$$

Non-destructive measurement leaves behind an eigenstate, effectively extracting an eigenvector, so the following

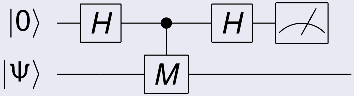
$$e^{\frac{-i\pi}{4}}PXZ^a|\Theta\rangle = e^{\frac{-i\pi}{4}}PXZ^a\frac{1}{\sqrt{2}}(|0\rangle + e^{\frac{i\pi}{4}}|1\rangle)$$
$$= e^{\frac{-i\pi}{4}}PX\frac{1}{\sqrt{2}}(|0\rangle + (-1)^a e^{\frac{i\pi}{4}}|1\rangle) = e^{\frac{-i\pi}{4}}P\frac{1}{\sqrt{2}}(|1\rangle + (-1)^a e^{\frac{i\pi}{4}}|0\rangle)$$
$$= \frac{1}{\sqrt{2}}(e^{\frac{i\pi}{4}}|1\rangle + (-1)^a|0\rangle) = (-Z)^a|\Theta\rangle$$

implies that measuring $e^{\frac{-i\pi}{4}}PX = 1$ prepares $|\Theta\rangle$.

Non-destructive measurement

If $M^2 = I$, the circuit at right measures M non-destructively on $|\Psi\rangle$, as shown below.

Measurement circuit



$$|0\rangle|\Psi\rangle \xrightarrow{H} \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)|\Psi\rangle \xrightarrow{C_M} \frac{1}{\sqrt{2}}(|0\rangle|\Psi\rangle + |1\rangle M|\Psi\rangle)$$
$$\xrightarrow{H} \frac{1}{2}((|0\rangle + |1\rangle)|\Psi\rangle + (|0\rangle - |1\rangle)M|\Psi\rangle)$$
$$= |0\rangle(|\Psi\rangle + M|\Psi\rangle) + |1\rangle(|\Psi\rangle - M|\Psi\rangle)$$

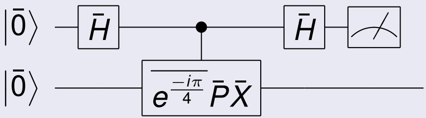
When $M = e^{\frac{-i\pi}{4}}PX$ and $|\Psi\rangle = |0\rangle$ this circuit outputs $|\Theta\rangle$ 50% of the time.

Encoding the $|\Theta\rangle$ preparation circuit

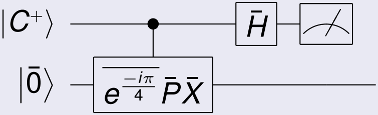
We do not know how to implement the encoded circuit at right; doing so requires $\bar{T}$.

But we can implement the one below.

Preparation circuit for $|\Theta\rangle$



Preparation circuit for $|\Theta\rangle$



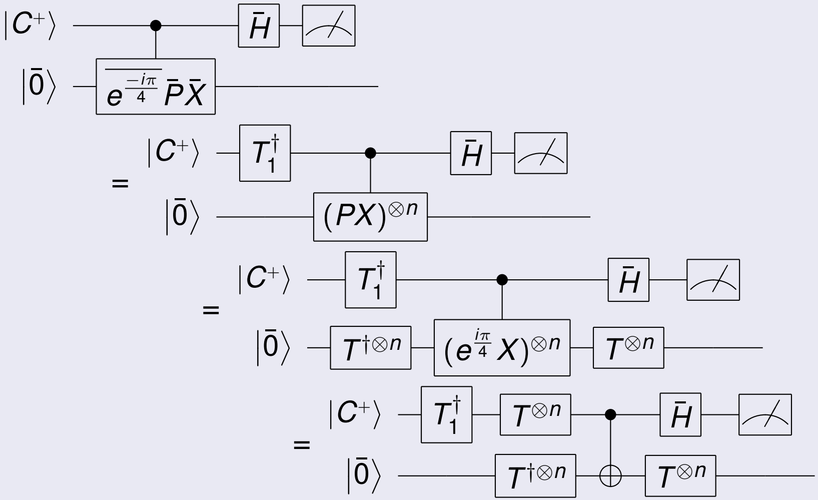
$|C^+\rangle = \frac{1}{\sqrt{2}}(|0\rangle^{\otimes n} + |1\rangle^{\otimes n})$ is akin to $|+\rangle$, but it can be coupled to transversally.

Its phase can be measured using the same gates as for $|\bar{+}\rangle$:

$$\bar{H}\,|C^{\pm}\rangle = H^{\otimes n}\frac{1}{\sqrt{2}}(|0\rangle^{\otimes n} \pm |1\rangle^{\otimes n}) = \sum_j |j\rangle \pm \sum_j (-1)^{\mathrm{wt}(j)}\,|j\rangle$$

Expanding the preparation circuit for $|\Theta\rangle$

Equivalent preparation circuits for $|\Theta\rangle$



Making $|C^+\rangle$

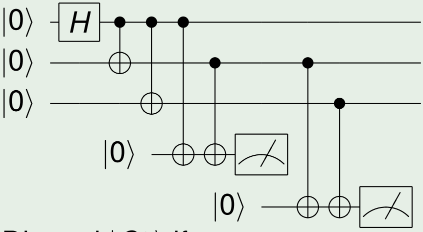
$|C^+\rangle$ recipe

1. Start with $|0\rangle^{\otimes n}$.
2. Apply H_1.
3. Apply $\bigotimes_{i=2}^{n} {}^{C}X_{1i}$.
4. Repeat the following t times
   For each pair of qubits i and $i+1$
   1. Procure a qubit $|0\rangle_a$.
   2. Apply ${}^{C}X_{ia}\,{}^{C}X_{i+1,a}$.
   3. Measure qubit a; discard $|C^+\rangle$ on 1.

The verification only catches X errors, but that's all we need.

Example (3 qubit cat state)

$|C^+\rangle$ construction & verification



Discard $|C^+\rangle$ if any measurements are non-zero.

Here I take $t = 1$. No single error yields two X errors on this state.

Apropos: Why repetition?

Repetition is a way of verifying crucial measurements.

Suppose that our code corrects t errors, but using the wrong measurement outcome will cause a crash.

What do we do if an incorrect measurement can result from $r < t$ errors?

Requiring g successive measurements to agree where $gr > t$ reduces the failure probability to an acceptable level.

Making $|\bar{+}\rangle$

Recall $|\bar{+}\rangle = \sum_{v \in \mathrm{G}} |v\rangle$

1. Diagonalize G to form $\check{\mathrm{G}}$
2. Initialize n qubits in $|0\rangle$
3. Apply $\bigotimes_{i=1}^{l} H_i$
4. Apply $\bigotimes_{i=1}^{l} \bigotimes_{j>l|\check{\mathrm{G}}^{i,j}=1} {}^{C}X_{i,j}$

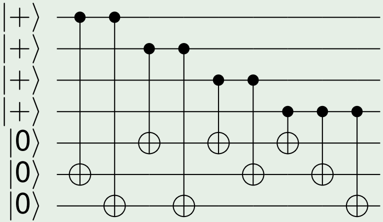
$$|0\rangle^n \xrightarrow{H} |+\rangle^l |0\rangle^m$$

$$\xrightarrow{{}^{C}X} \sum_{k=1}^{2^l} \left| \sum_{i=1}^{l} (\check{\mathrm{G}}^i)^{k_i} \right\rangle$$

Example (Steane code)

$$\check{\mathrm{G}} = \begin{bmatrix} 1000011 \\ 0100101 \\ 0010110 \\ 0001111 \end{bmatrix}$$

Circuit for making $|\bar{+}\rangle$



Verifying $|\bar{+}\rangle$ against Z errors

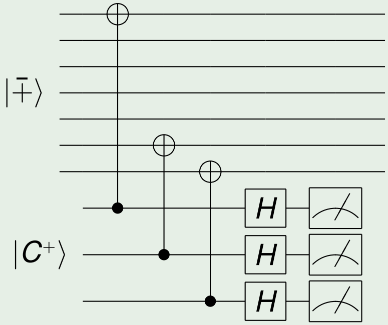
Let $w_i = \mathrm{wt}(\check{\mathrm{G}}^i)$ and k_{ij} be the index of the jth 1 in $\check{\mathrm{G}}^i$.
For each row of $\check{\mathrm{G}}$, $\check{\mathrm{G}}^i$,

1. Prepare $|C^+\rangle$ on qubits $n+1$ to $n+w_i$
2. Apply $\bigotimes_{j=1}^{w_i} {}^C X_{n+j,k_{ij}}$
3. Apply $\bigotimes_{i=n+1}^{n+w_i} H_i$ and measure
4. Discard $|\bar{+}\rangle$ when the measurement parity is odd

Now we must check for X errors.

Example (Steane code)

Circuit for checking $X_1 X_6 X_7$



Discard when measurement total is odd.

Verifying $|\mp\rangle$ against X errors

1. Diagonalize H to form $\check{\mathrm{H}}$
2. Perform verification as for Z errors but replace $\check{\mathrm{G}}$ with $\check{\mathrm{H}}$ and ${}^{C}X$ with ${}^{C}Z$

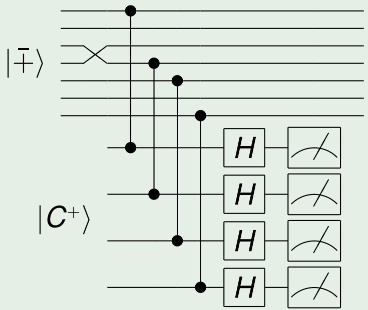
Example (Steane code)

$$\check{\mathrm{H}} = \begin{bmatrix} 1001101 \\ 0101011 \\ 0010111 \end{bmatrix}$$

Qubits 3 and 4 were switched compared to H.

Example (Steane code)

Circuit for checking $Z_1Z_4Z_5Z_7$



Discard when measurement total is odd.

Apropos: Why all the qubits?

We begin to see why qubit expenditure is a pillar of fault tolerance.

It is often the case that we need to prepare ancillae, either for the ancilla states themselves or to implement gates.

This preparation is almost never fault tolerant.

To achieve fault tolerance, we must check the state for mistakes and discard or fix it if we find any.

Summary: Implementing the $\frac{\pi}{4}$ rotation

1. $\bar{T}$ is applied to the data by teleporting it under the $\bar{T}$ in $|\bar{\Theta}\rangle = \bar{T}\,|\bar{+}\rangle$.
2. $|\bar{\Theta}\rangle$ is produced by measuring $\overline{e^{\frac{-i\pi}{4}}}\bar{P}\bar{X}$ on $|\bar{0}\rangle$ using cat states, $|C^{+}\rangle$. This measurement must be repeated $t+1$ times to insure fault tolerance.
3. $|C^{+}\rangle$ is made in the standard way. The parity of each pair of qubits is then checked in t cycles.
4. $|\bar{+}\rangle$, is produced by directly applying a superposition of each X stabilizer and nothing.
   $|\bar{0}\rangle = \bar{H}\,|\bar{+}\rangle$

It is not necessary, but probably a good idea to check $|\bar{\Theta}\rangle$ for errors before using it too.

Non fault tolerant CSS code error location

Let $w_i = \mathrm{wt}(\mathrm{H}^i)$ and k_{ij} be the index of the jth 1 in H^i.
To measure the X stabilizer for H^i:

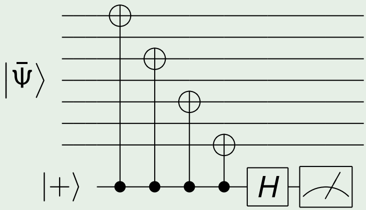
1. Procure a qubit $|+\rangle_a$.
2. Apply $H_a \bigotimes_{j=1}^{w_i} {}^{C}X_{a,k_{ij}}$.
3. Measure qubit a; $1 \rightarrow$ error.

$$|\bar{\Psi}\rangle_{1\ldots n} |0\rangle_a \xrightarrow{{}^{C}X} |\bar{\Psi}\rangle_{1\ldots n} \left| S^{\mathcal{G}_X{}^i} \right\rangle_a .$$

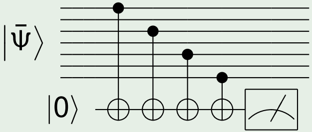
Z stabilizers are measured by starting with the state $|0\rangle_a$, reversing the direction of the ${}^{C}X$s, and omitting the H_a.

Example (Steane code)

Measuring $X_1 X_3 X_5 X_7$



Measuring $Z_1 Z_3 Z_5 Z_7$



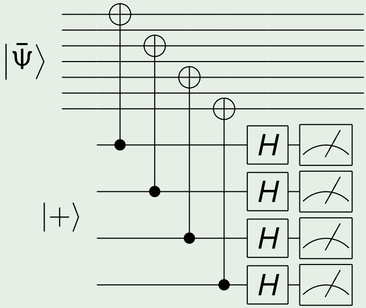
Shor's method of CSS code error location

Let $w_i = \mathrm{wt}(\mathrm{H}^i)$ and k_{ij} be the index of the jth 1 in H^i.
To measure the X stabilizer for H^i:

1. Prepare $|C^+\rangle$ on qubits $n+1$ to $n+w_i$
2. Apply $\bigotimes_{j=1}^{w_i} {}^{C}X_{n+j,k_{ij}}$
3. Apply $\bigotimes_{j=n+1}^{n+w_i} H_j$
4. Measure $|C\rangle$; odd parity indicates an error

Example (Steane code)

Measuring $X_1 X_3 X_5 X_7$



To measure Z stabilizers replace ${}^{C}X$ with ${}^{C}Z$.

Steane's method of CSS code error location

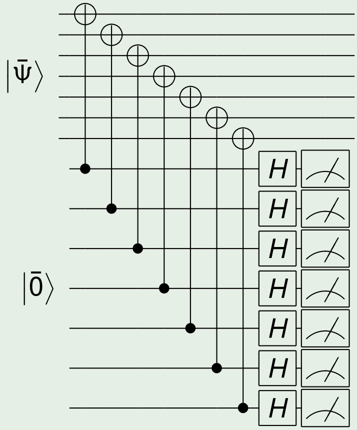
Z error location:

1. Prepare $|\bar{0}\rangle_{n+1\ldots 2n}$
2. Apply $\bigotimes_{i=1}^{n} {}^{C}X_{n+i,i}$
3. Apply $\bigotimes_{i=n+1}^{2n} H_i$
4. Treat the output as a classical code string

Repeat till $t+1$ consecutive extractions give the same syndrome.
To locate X errors replace $|\bar{0}\rangle$ with $|\bar{\mp}\rangle$ and ${}^{C}X$ with ${}^{C}Z$.

Example (Steane code)

Extract all X syndromes (do $2\times$)



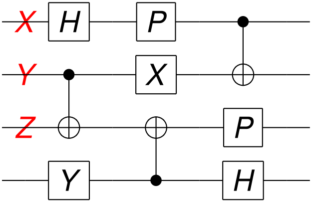
Recovery from errors

Knill has greatly simplified the process of error recovery by pointing out that it is unnecessary.
(This might also be attributed to Raussendorf & Briegel)

This is because we only apply Clifford gates to our data, and Clifford gates preserve the set of Pauli operators.

Thus, it is easy to determine the effects of those errors on our measurements.

Figure: Error propagation



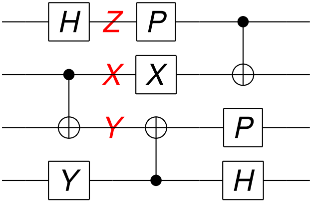
Recovery from errors

Knill has greatly simplified the process of error recovery by pointing out that it is unnecessary.
(This might also be attributed to Raussendorf & Briegel)

This is because we only apply Clifford gates to our data, and Clifford gates preserve the set of Pauli operators.

Thus, it is easy to determine the effects of those errors on our measurements.

Figure: Error propagation



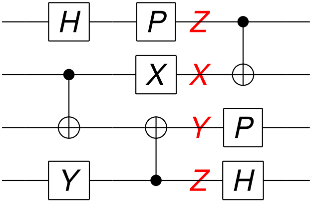
Recovery from errors

Knill has greatly simplified the process of error recovery by pointing out that it is unnecessary.
(This might also be attributed to Raussendorf & Briegel)

This is because we only apply Clifford gates to our data, and Clifford gates preserve the set of Pauli operators.

Thus, it is easy to determine the effects of those errors on our measurements.

Figure: Error propagation



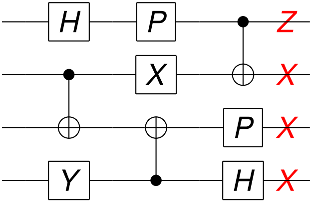
Recovery from errors

Knill has greatly simplified the process of error recovery by pointing out that it is unnecessary.
(This might also be attributed to Raussendorf & Briegel)

This is because we only apply Clifford gates to our data, and Clifford gates preserve the set of Pauli operators.

Thus, it is easy to determine the effects of those errors on our measurements.

Figure: Error propagation



Mission accomplished?

We have described a fully fault tolerant gate set. This means that

> *Errors affecting r components where $r < t$ will not result in s errors on a single encoded block where $s > r$.*

We have achieved this by following the principles below

1. Transversality - Two different parts of a data block are never coupled.
2. Clean ancillae - Ancillae are designed such that their (transmissible) failure probability is of, at least, order $t + 1$.
3. Repetition - Measurements on the data are repeated until their error probability is of, at least, order $t + 1$.

Congratulations! We're online!

Mission accomplished?

We have described a fully fault tolerant gate set. This means that

> *Errors affecting r components where $r < t$ will not result in s errors on a single encoded block where $s > r$.*

We have achieved this by following the principles below

1. Transversality - Two different parts of a data block are never coupled.
2. Clean ancillae - Ancillae are designed such that their (transmissible) failure probability is of, at least, order $t + 1$.
3. Repetition - Measurements on the data are repeated until their error probability is of, at least, order $t + 1$.

But wasn't there something in our definition of a quantum computer about it running forever?

Finite codes with uncorrelated errors

An $[[n, k, d]]$ code can correct any error on $t = \left\lfloor \frac{d-1}{2} \right\rfloor$ qubits.

If errors occur on each qubit independently then $t + 1$ errors are a possibility.

Let p be the probability a single qubit is in error.

Computations requiring more than $\frac{1}{p^{t+1}}$ steps will almost certainly fail.

Example (Steane code)

$$[[n, k, d]] = [[7, 1, 3]]$$

$$t = \left\lfloor \frac{d-1}{2} \right\rfloor = 1$$

Approximate failure probability

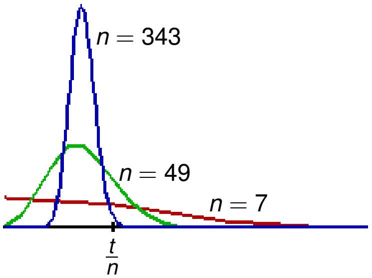
$$2 \times \binom{7}{2} p^2 = 42p^2$$

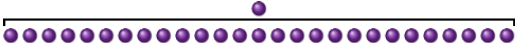
Achieving arbitrarily low error probabilities: Blocks

Block coding

- Failures result from fluctuations.
- Fluctuations reduce with code size.
- Families of codes with good fractional distance exist.

Figure: Probability vs. error fraction





Problem

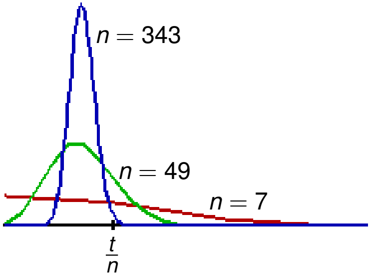
No efficient way of directly making block code ancillae is known.

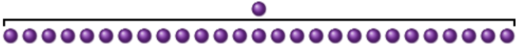
Achieving arbitrarily low error probabilities: Blocks

Block coding

- Failures result from fluctuations.
- Fluctuations reduce with code size.
- Families of codes with good fractional distance exist.

Figure: Probability vs. error fraction





Problem

No efficient way of directly making block code ancillae is known.

Achieving arbitrarily low error probabilities: Recursion

Concatenated coding

- A finite encoding reduces the failure probability.
- The encoded system resembles an unencoded system with different error probabilities.
- Recursively encoding can reduce the error probability arbitrarily.

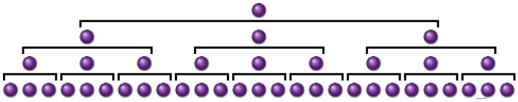
Example (Steane Code)

Assume qubits have X errors with probability p.

Unrecoverable error probability

$$p' \approx \binom{7}{2} p^2 = 21 p^2$$

What if $p > \frac{1}{21}$?



Estimating the threshold: When coding works

Sometimes unencoded errors are so likely that encoded computations are more prone to fail than unencoded ones.

We call the probability below which it is possible, using some family of encodings, to compute indefinitely the threshold.

We would like to say that a concatenated code is below threshold when

$$\left\{\begin{matrix}\text{Encoded}\\ \text{error rate}\end{matrix}\right\} < \left\{\begin{matrix}\text{Unencoded}\\ \text{error rate}\end{matrix}\right\},$$

which can be approximated for a particular code as

$$\binom{g}{t+1} p^{t+1} < p$$

where g is the number of unencoded gates in the most complex encoded gate.

Counting the unencoded gates in the Steane code $\bar{T}$

Step	Expected iterations	Gates per iteration	Total gates
Part 1: The gate			
Teleportation	1	3×7	21
Making $\lvert\Theta\rangle$	1	$12 \times 7 + 2$	86
Making $\lvert C^{+}\rangle$ on 7 qubits	2	$1 + 6 \times 7$	86
Making $\lvert\bar{0}\rangle$	1	$9 + 3 \times 3 + 3 \times 4$	30
Making $\lvert C^{+}\rangle$ on 3 qubits	3	$1 + 2 \times 7$	45
Making $\lvert C^{+}\rangle$ on 4 qubits	4	$1 + 4 \times 7$	116
Part 2: The correction			
Making $\lvert\bar{+}\rangle$	2	$9 + 3 \times 3 + 4$	44
Making $\lvert\bar{0}\rangle$	2	$9 + 3 \times 3 + 4 + 7$	58
Making $\lvert C^{+}\rangle$ on 3 qubits	6	$1 + 2 \times 7$	90
Making $\lvert C^{+}\rangle$ on 4 qubits	2	$1 + 4 \times 7$	58
Grand total			634

An approximate lower bound on the threshold

Using the gate totals from the previous slide, we can estimate the tolerable gate error rate for this implementation based on the Steane code.

The result must be a lower bound on the threshold.

$$\binom{g}{t+1} p_0^{t+1} = \binom{634}{2} p_0^2 = 200661 p_0^2 = p_0 \quad \rightarrow \quad p_0 \approx 5 \times 10^{-6}$$

In spite of neglecting memory errors (which force us to consider gate times and spacing), our bound is quite low. The first estimates were on this order.

Recent work has brought the estimated lower bound up to $p_0 \approx 10^{-2}$. Rigorous lower bounds are up to $p_0 \approx 10^{-4}$.

What makes them threshold estimates?

The threshold we calculated was only an estimate of a bound for several reasons

- We neglected higher order errors, as well as the fact that many $t + 1$ order errors are non-fatal.
- We considered only a single error probability. In practice there will be many kinds of errors and thus many probabilities.
- There is not a direct mapping between unencoded and encoded qubits. “Good” encoded qubits are not as good as “good” unencoded ones.

Suggested topics for future study

- Decoherence free subspaces/subsystems (Poulin)
- General fault tolerant constructions (Gottesman)
- Telecorrection (Knill)
- Rigorous threshold bounds (Aharonov, Reichardt, Aliferis)
- Low density parity check codes (McKay)
- Toric codes (Kitaev, Preskill)