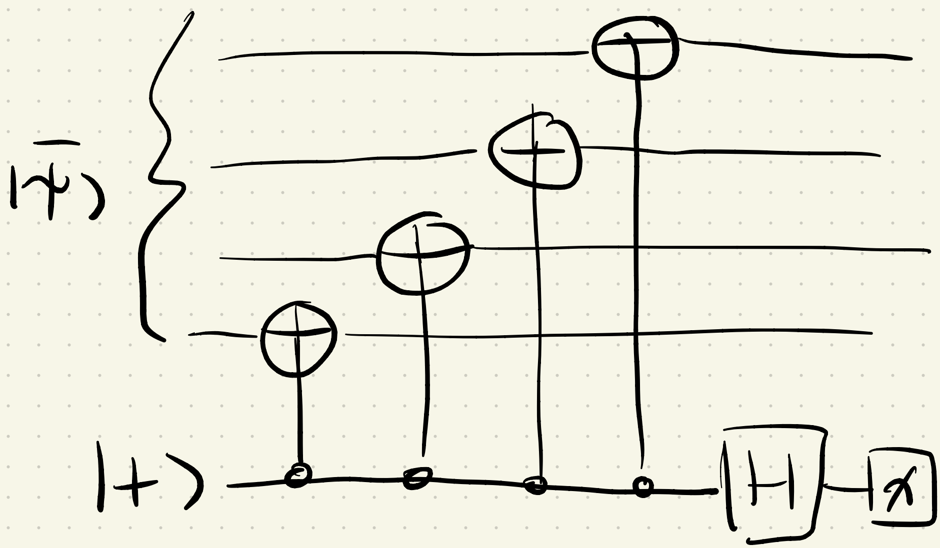
Lecture $\overline{\underline{\text{II}}}$: FT OPERATIONS PART I

FT error correction

Consider naïve circuit to measure stabilizer $X^{\otimes 4}$ & suppose code has $d=3$.



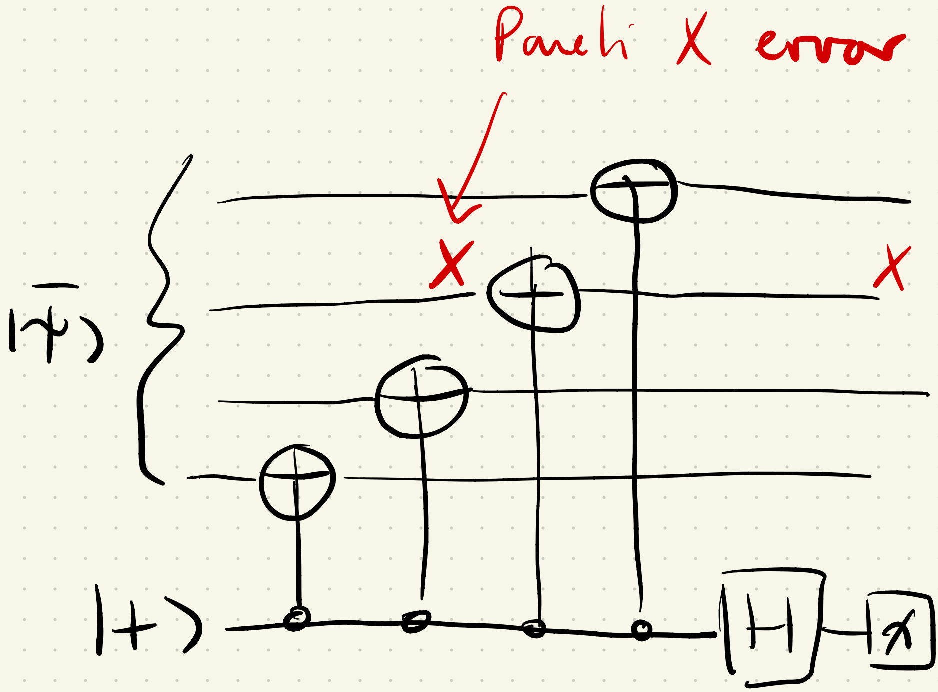
$$t = \left\lfloor \frac{d-1}{2} \right\rfloor = 1$$

For perfect input & 1 fault during circuit, we need ideal decoding of output = ideal decoding of input.

Another way of saying this:

We want to construct a circuit that fails with probability $\mathcal{O}(p^2)$ ie it can deal with all single qubit errors (assuming iid noise).

'Good error'



Very useful:

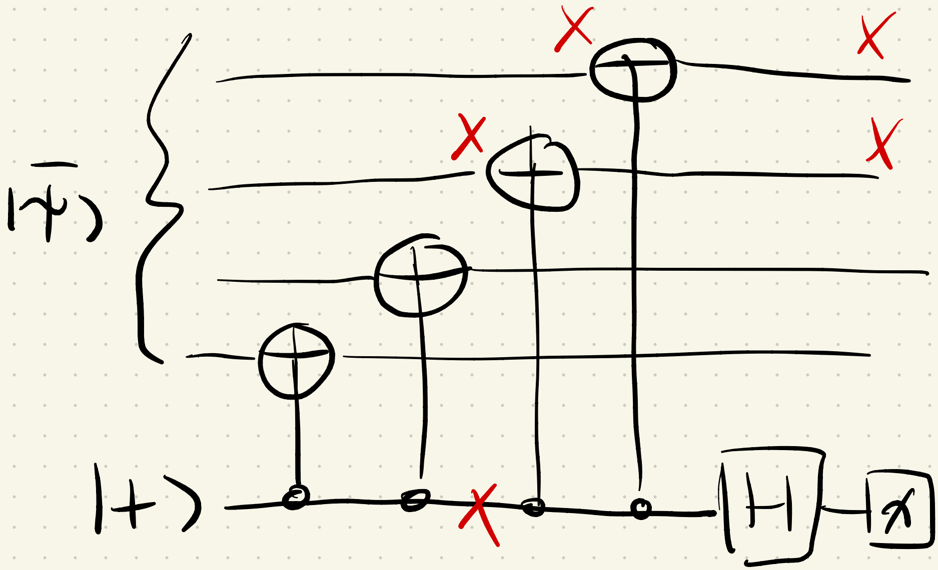
$$X \otimes 1 \xrightarrow{\text{CNOT}} X \otimes X$$

$$1 \otimes X \xrightarrow{\text{CNOT}} 1 \otimes X$$

$$Z \otimes 1 \xrightarrow{\text{CNOT}} Z \otimes 1$$

$$1 \otimes Z \xrightarrow{\text{CNOT}} Z \otimes Z$$

But 'Bad error'



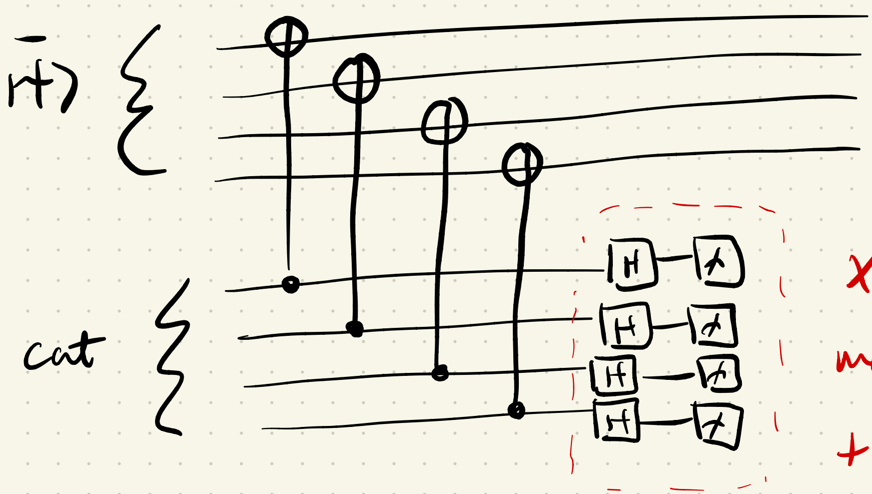
1 ancilla error led to 2 errors on output (an uncorrectable error).

One solution

Shor EC [Shor '94]

Instead of using a bare ancilla use a cat state.

$|0\rangle^{\otimes n} + |1\rangle^{\otimes n}$ (omit normalization)



$X \otimes X \otimes X \otimes X$ measurement

+1 for $|0000\rangle + |1111\rangle$

−1 for $|0000\rangle - |1111\rangle$

Now any single fault during the circuit can only lead to at most one fault on the output.

But how do we prepare the cat states fault-tolerantly?

Cat state is a stabilizer state

(stabilizer code w/ $k=0$)

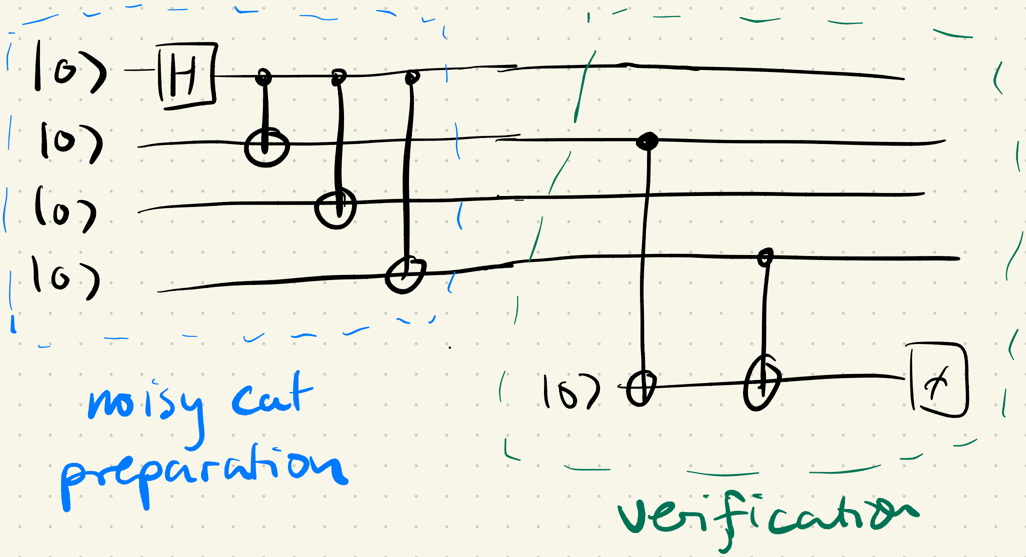
$|0000\rangle + |1111\rangle$

is stabilized by $Z_1 Z_2$, $Z_2 Z_3$, $Z_3 Z_4$,

$X_1 X_2 X_3 X_4$

Verification circuit

Check cat state stabilizer eigenvalues



Accept if measurement result is +1, reject otherwise.

Remember as our code has $d=3$ we we only worried about single faults.

The verification circuit catches all X errors on the input cat states.

It is possible for single qubit X errors to be introduced into the cat state during the verification part, but have the same effect as single qubit X errors on the cat state during the measurement of the stabilizer.

We don't measure the $X_1 X_2 X_3 X_4$ stabilizer.

So single qubit Z errors can occur meaning that we will have $|0000\rangle - |1111\rangle$ instead of $|0000\rangle + |1111\rangle$.

This could cause us to apply the wrong correction as the measurement outcome would be flipped.

To deal with this we repeat the whole procedure 3 times and take the majority vote for the stabilizer measurement outcome.

As we are only considering single faults during the entire procedure, we will get an accurate result.

This whole procedure is rather complicated and required $w+1$ ancillas where w is the weight of the stabilizer we want to measure.

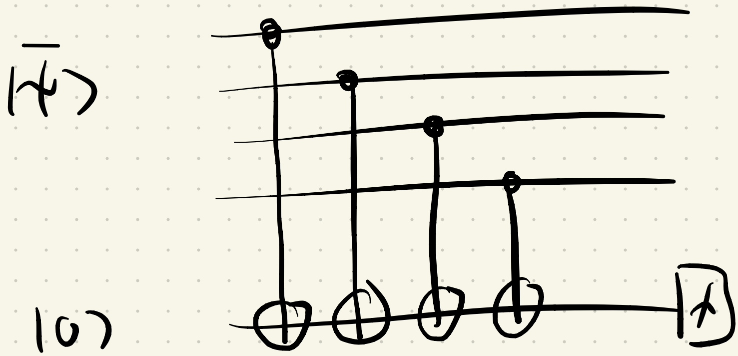
Can we do better?

Yes, using 'flag qubits'.

Flag error correction

[Chao, Reichardt 2018]

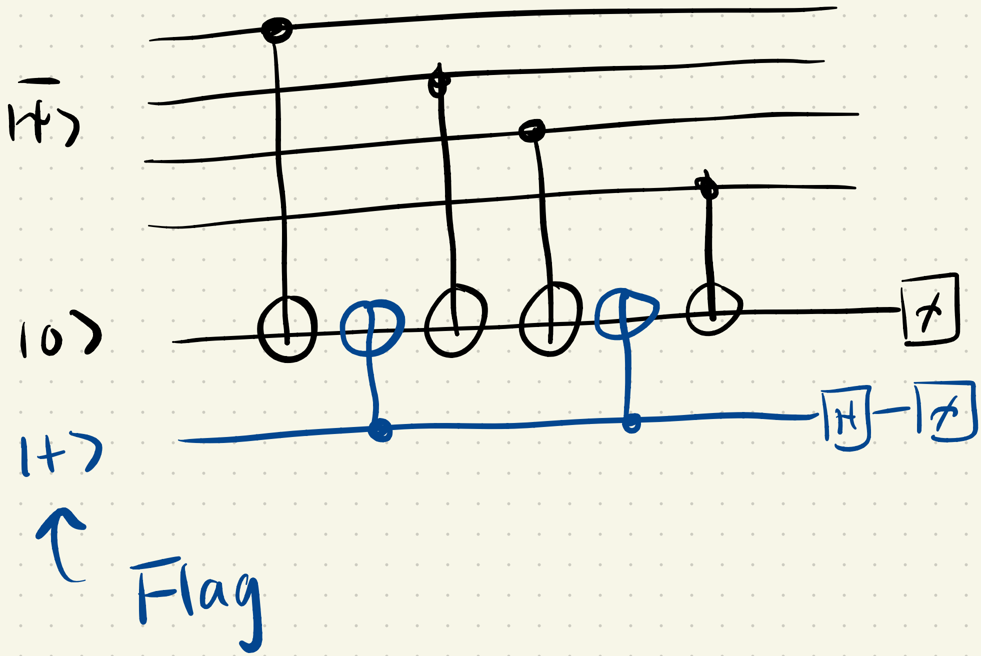
First, another way to measure $Z^{\otimes 4}$



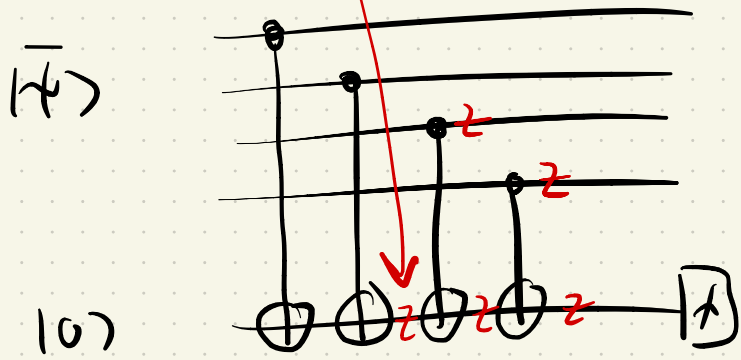
Can be derived from our previous circuit by inserting

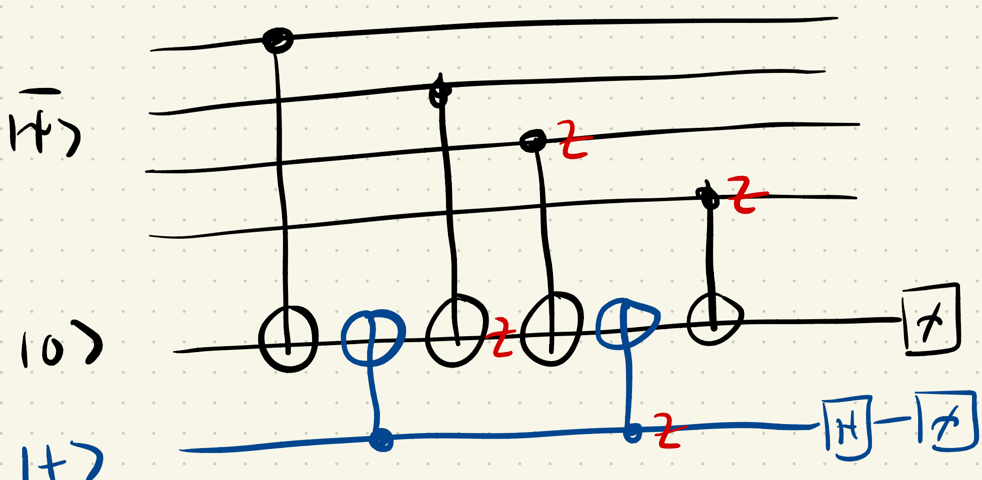
$I = HH$

Idea: add 'flag' ancilla to catch bad faults where 1 fault → 2q fault on data qubits



Bad fault





Flag

detected by flag!

In general, for flag EC we only need 2 extra flag qubits for fault tolerant stabilizer measurement.

Aside: In the (2D) surface code, these constructions are not necessary & fault tolerance can be achieved by repeating the stabilizer measurements $O(d)$ times where d is the code distance.

FT Measurement

There exists a procedure for generic stabilizer codes but it is rather cumbersome so we will consider the special case of CSS codes.

Recall that CSS codes are constructed from two classical linear codes

$\mathcal{C}_1 : [n, k_1, d_1]$ $\quad \mathcal{C}_2^{\perp} : [n, k_2, d_2]$

where $C_2^\perp \subseteq C_1$ $C_2^\perp$ dual ie $\{x \mid x \cdot z = 0 \ \forall z \in \mathcal{C}_2\}$

$CSS(\mathcal{C}_1, \mathcal{C}_2^\perp)$ has

parameters $[[n, k_1 - k_2, d]]$

$d \geq \min(d_1, d_2)$. $\mathcal{C}_1$ X stabilizes

$\mathcal{C}_2^\perp$ Z stabilizes

The important fact for us is the form of the codewords:

$$\sum_{w \in \mathcal{C}_2^\perp} |v + w\rangle \quad \text{where } v \in \mathcal{C}_1$$

Suppose we want to measure logical $\bar{Z}$ for all encoded qubits.

To do this we measure all the qubits in the Z basis. The only errors we need to worry about are bit-flips as phase-flips won't change the measurement outcomes.

Let $e \in \mathbb{F}_2^n$ represent a bit-flip error, making our state $\sum_{w \in C_2^\perp} |v+w+e\rangle$.

When we measure we will observe the outcome $v + w + e$ for random $w \in \mathcal{C}_2^\perp$.

Now, $v+w$ is a codeword of $\mathcal{C}_1$ as $v \in \mathcal{C}_1$ & $\mathcal{C}_2^\perp \subseteq \mathcal{C}_1$

Suppose $|e| \leq t = \lfloor \frac{d-1}{2} \rfloor$

We know that $d \geq \min(d_1, d_2)$.

Therefore we can just run the classical decoding algorithm for $\mathcal{C}_1$ to obtain $v+w$ & hence v.

FT State prep

Problem: naively implementing the encoding circuit is not FT.

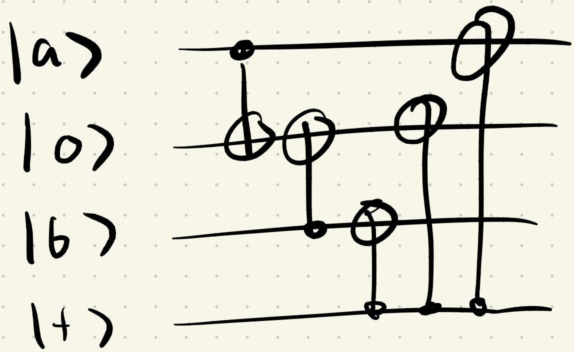
e.g. $[[4,2,2]]$ code

$S = \langle XXXX, ZZZZ \rangle$

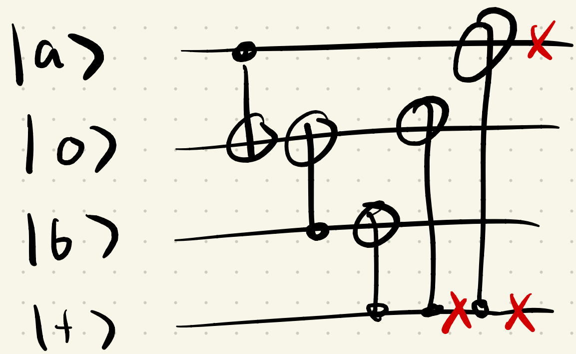
$\overline{X}_1 = XX11 \quad \overline{X}_2 = 1XX1$

$\overline{Z}_1 = 1ZZ1 \quad \overline{Z}_2 = ZZ11$

Can prepare $|\overline{ab}\rangle$ with the following circuit



Not FT! e.g.



1 fault

→ 2 faults on output

One can use flag type tricks again but for CSS codes there is a nice general method for FT state prep.

<u>Task</u> prepare $|\overline{0\ldots0}\rangle$ all zeros logical state

$$|\overline{0\cdots0}\rangle = \sum_{w\in \mathcal{C}_2^\perp} |w\rangle$$

$\mathcal{C}_2^\perp$ describes X stabs

$$= \sum_{s\in S_X} s|0\rangle^{\otimes n}$$

← X stabilizers

$$|\overline{0\ldots0}\rangle = \sum_{s\in S_X} s\,|0\rangle^{\otimes n}$$

$$= \prod_{g_i} \frac{(1+g_i)}{2}\,|0\rangle^{\otimes n}$$

where $S_X = \langle g_1, g_2, \ldots, g_m \rangle$

generating set for the X stabilizer

$\frac{(1+g_i)}{2}$ is simply a projective

measurement of the generator

g_i. We can do this

fault-tolerantly using e.g.

Shor EC.

So to prepare $|\overline{0 \ldots 0}\rangle$

we measure a generating set

of the X stabilizer group

& then apply the appropriate (Z)

recovery operator. It doesn't

matter if we apply a

logical $\bar{Z}$ as

$$\bar{Z} \prod_{g_i} \frac{(1+g_i)}{2} |0\rangle^{\otimes n}$$

$$= \prod_{g_i} \frac{(1+g_i)}{2} \underbrace{\bar{Z}|0\rangle^{\otimes n}}_{|0\rangle^{\otimes n}}$$

Will be a product of Z ops in a CSS code.

Post script

- Other FT stabilizer measurement protocols exist, notably those of Steane & Knill.
- CSS codes are by far the most popular class of codes because of their nice FT properties, and in fact any $[[n,k,d]]$ stabilizer code can be mapped onto a $[[4n,2k,2d]]$ CSS code.