9-qubit Shor code:

The Shor code can correct arbitrary errors since it corrects Z and X, separately, which corrects $Y = iXZ$.

This is accomplished by combining the bit flip encoding with the phase flip encoding.

We start by encoding:

$$|0\rangle \rightarrow |{+}{+}{+}\rangle$$
$$|1\rangle \rightarrow |{-}{-}{-}\rangle$$

Then for each bit we use the 3 qubit bit flip encoding:

$$|+\rangle \rightarrow \frac{1}{\sqrt{2}}(|000\rangle + |111\rangle)$$
$$|-\rangle \rightarrow \frac{1}{\sqrt{2}}(|000\rangle - |111\rangle)$$

The resulting 9-qubit code-words are:

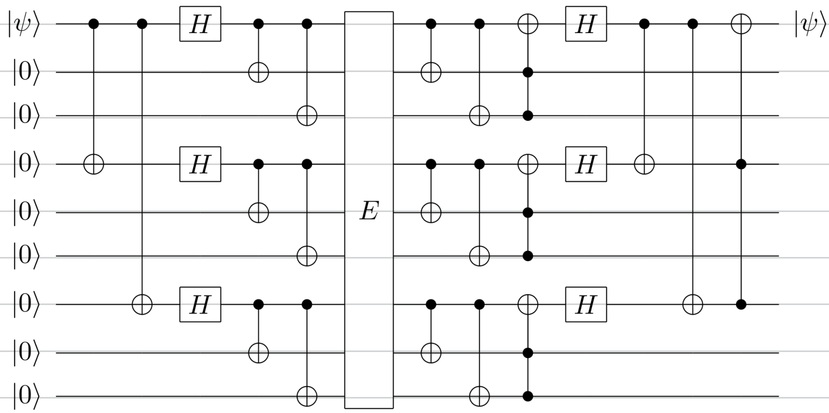
$$|0\rangle \rightarrow |0\rangle_L = \frac{1}{2\sqrt{2}}(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)$$
$$|1\rangle \rightarrow |1\rangle_L = \frac{1}{2\sqrt{2}}(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)$$

(see slide)

$$|\psi\rangle = \alpha_0|0\rangle_L + \alpha_1|1\rangle_L$$

- This is a degenerate code, in that different errors will result in the same state. For example, Z_1, Z_2 and Z_3 give the same state. However, an X_i on any qubit gives an orthogonal state to any other X_j, and a Z_i is orthogonal to any X_i.
- For a classical code that would be a problem, but in the quantum case it doesn't matter, we can still correct it. Most quantum codes are degenerate.
- A combined $XZ = -iY$ also transforms the code word to an orthogonal state that can be corrected.
- This was the first example of a quantum error correction code capable of correcting an arbitrary single qubit error. (Shor, 1995)

Circuit for Shor code:



- As with repitition code, should actually use ancilla qubits to do the parity checks in order to preserve $|\psi_{enc}\rangle$.

- What about the issue of "analog errors"? Well, as we saw, any single qubit error can be discretized into some superposition of X, Y, and Z. The syndrome measurement projects this superposition into one of the discrete errors! In this case, the collapse of the wavefunction by measurement actually allows us to preserve an unknown state, which is a bit counterintuitive.

Other codes:

The shor code is a [[9, 1, 3]] code.

There are smaller codes that can correct an arbitrary single qubit error.

Steane code	7-qubits + ancillas	(1996)	[[7, 1, 3]]
Laflamme, et al.	5-qubits	(1996)	[[5, 1, 3]]

What is the minimum codesize?

Say the code words are n bits long.
Due to a single qubit error many different states can appear.
The number of states is:

$$\underbrace{\overset{X,Y,Z}{3n}+\overset{\text{No error}}{1}}_{|0\rangle\text{ encoding}} + \underbrace{3n+1}_{|1\rangle\text{ encoding}} = 2(3n+1)$$

With n qubits we span a 2^n dimensional space. For a non-degenerate code, the spanned space must be larger than or equal to the number of possible states after an error, or

$$2^n \geq 2(3n+1) \Rightarrow 2^{n-1} \geq 3n+1$$

$\Rightarrow$ Need $\boxed{n \geq 5}$. This is the limit for a non-degenerate code. There is no known degenerate code w/ $n<5$, and this can also be proven.

5-Qubit code:

$$
\begin{aligned}
|0_L\rangle = {} & |00000\rangle + |10010\rangle + |01001\rangle + |10100\rangle + |01010\rangle - |11011\rangle \\
& - |00110\rangle - |11000\rangle - |11101\rangle - |00011\rangle - |11110\rangle - |01111\rangle \\
& - |10001\rangle - |01100\rangle - |10111\rangle + |00101\rangle
\end{aligned}
$$

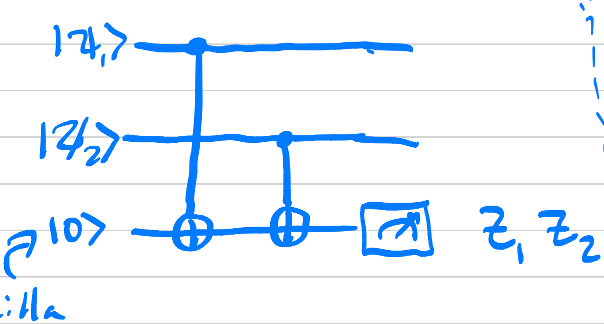
$$
\begin{aligned}
|1_L\rangle = {} & |11111\rangle + |01101\rangle + |10110\rangle + |01011\rangle + |10101\rangle - |00100\rangle \\
& - |11001\rangle - |00111\rangle - |00010\rangle - |11100\rangle - |00001\rangle - |10000\rangle \\
& - |01110\rangle - |10011\rangle - |01000\rangle - |11010\rangle
\end{aligned}
$$

- This is a mess! Not obvious how it works, where it came from, how to do syndrome measurement + recovery...

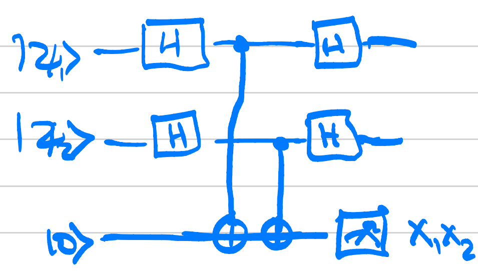
- Motivates a different formalism: "Stabilizers"

- Stabilizers consist of Pauli parity checks:

Z-parity check:



X-parity check



Stabilizer codes:

- As we have seen, any single qubit error can be discretized into tensor products of Pauli operators. As a result, it is possible to define a large class of error correcting codes as the subspace spanned by the eigenstates with eigenvalue +1 of a set of ~~of~~ operators that are also composed of tensor products of the Pauli operators, and that perform parity checks on pairs of qubits. These operators generate a group, called the "stabilizer." Measuring the generators of the stabilizer gives the error syndromes.

- Conveniently, you can learn the error syndrome ~~by~~ for a given error and stabilizer code by learning how the generators commute or anticommute with each error operator, w/out worrying about how the errors affect the codewords themselves. This is easier than looking at the encoding, error, and decoding procedures as we have been doing.

- Let's start w/ the simple example of the 3 qubit bit flip code.

The stabilizer generators are:

$$\begin{array}{ccc} & & (\text{Qubits: } 1 \quad 2 \quad 3) \\ Z_1 Z_2 I_3 & & Z \otimes Z \otimes I \quad (S_1) \\ & = & \\ \text{and } Z_1 I_2 Z_3 & & Z \otimes I \otimes Z \quad (S_2) \end{array}$$

Not unique!

The codewords are all states of the form $\alpha_0|000\rangle + \alpha_1|111\rangle$, which correspond to the subspace of eigenvectors with eigenvalues +1.

~~Suppose~~ The range of possible errors this code can correct are:

	$Z \otimes Z \otimes I$	$Z \otimes I \otimes Z$	corrective action:
$I \otimes I \otimes I$	+1	+1	Nothing
$X \otimes I \otimes I$	−1	−1	X_1
$I \otimes X \otimes I$	−1	+1	X_2
$I \otimes I \otimes X$	+1	−1	X_3

commutation relations

Logical gates: $X_L = X_1 X_2 X_3$, $Z_L = Z_1 Z_2 Z_3$

- It is immediately obvious that this code cannot correct for phase errors, as a Z rotation on any qubit will commute with both generators.

- The stabilizer generators for the 9-qubit Shor code are:

$$\left.\begin{array}{l} S_1: Z_1 Z_2 \\ S_2: Z_1 Z_3 \\ S_3: Z_4 Z_5 \\ S_4: Z_4 Z_6 \\ S_5: Z_7 Z_8 \\ S_6: Z_7 Z_9 \end{array}\right\} \text{Checks for a bit flip in any of the 9 qubits.}$$

$$\left.\begin{array}{l} S_7: X_1 X_2 X_3 X_4 X_5 X_6 \\ S_8: X_4 X_5 X_6 X_7 X_8 X_9 \end{array}\right\}$$

checks for a phase flip in first 6 and ~~second~~ last six qubits, determines which triplet (if any) had a phase flip.

- For an ~~code~~ $[[n,k,d]]$ stabilizer code, the minimal number of independent generators is $n-k$.

- The code words $|\Psi\rangle$ can be quite messy, but the stabilizer formalism is fairly simple.

- What about logical gates on the logical qubits?

$$Z_L = X_1 X_2 X_3 X_4 X_5 X_6 X_7 X_8 X_9$$

$$X_L = Z_1 Z_2 Z_3 Z_4 Z_5 Z_6 Z_7 Z_8 Z_9$$

- How do I measure which codeword I have at the end?

- Do the parity check corresponding to Z_L:

$$X_1 X_2 X_3 X_4 X_5 X_6 X_7 X_8 X_9$$

Let's return to the 5-Qubit code, but consider it with the stabilizer formalism:

$n=5$, $k=1$, so need 4 stabilizers:

S_1	X_1	Z_2	Z_3	X_4	I_5
S_2	I_1	X_2	Z_3	Z_4	X_5
S_3	X_1	I_2	X_3	Z_4	Z_5
S_4	Z_1	X_2	I_3	X_4	Z_5
Z_L	Z_1	Z_2	Z_3	Z_4	Z_5
X_L	X_1	X_2	X_3	X_4	X_5

- 4 stabilizers, each can return +1 or −1, so $2^4 = 16$ outcomes.

- How many single qubit errors are there?

$$\underbrace{1}_{\text{No error, } I} + \underbrace{3}_{X,Z,XZ} \times \underbrace{5}_{5\text{-qubits}} = 16.$$

(To figure out which syndrome corresponds to which error, take the commutator of each error with each stabilizer to make a truth table.)

- How do I measure the logical qubit?

Take 5 qubits in <u>any</u> state.

Measure the stabilizers S_1, S_2, S_3, S_4, and perform the corresponding corrective actions.

Now the 5 qubits are either in $|0\rangle_L$ or $|1\rangle_L$.

Measure the parity operator corresponding to Z_L, $Z_1 Z_2 Z_3 Z_4 Z_5$.

Knill-Laflamme Condition:

Alot of what we just discussed can be condensed into a single condition, called the Knill-Laflamme condition (1997):

Given a code with codewords $\{|\psi_i\rangle\}$ and a set of errors $\{E_a\}$, errors can be corrected iff:

$$\langle \psi_j | E_a^{\dagger} E_b | \psi_i \rangle = C_{ab} \delta_{ij}$$

with C_{ab} as a number that can depend on a and b but not i or j.

Basically, this condition enforces two things:

1) If $a=b$ and $E_a = E_b$, then this means no error in the set $\{E_a\}$ can take you from one codeword to the other.

2) If $E_a \neq E_b$, this enforces that the environment does not "learn" any information about the code word.