To transition from classical to quantum code design, let's first think about how to write a classical code in quantum language:

Parity-check matrix:

$$H = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix} \longrightarrow$$

quantum Hamiltonian

$$H_0 = -Z_1Z_2 - Z_2Z_3$$

Codewords generated by:

$$G = (1 \quad 1 \quad 1) \longrightarrow$$

has symmetry

$$X_L = X_1X_2X_3$$

and $[H_0, X_L] = 0$.

↳ codespace = $\ker(H) = \mathrm{im}(G^T)$ ⟶ ground states of H_0.

This is just the bit flip repetition code we've already seen.

But more generically:

parity-check matrix H ⟶ $H_0 = -\sum_{\text{check } c} \prod_{\substack{\text{bit } b \in c \\ (H_{cb}=1)}} Z_b$

codeword generator matrix G ⟶ Symmetry $X_{L,\alpha}$: $[H_0, X_{L,\alpha}] = 0$

and $X_{L,\alpha} = \prod_{\substack{b \in \partial\alpha \\ (G_{\alpha b}=1)}} X_b$

Punchline: look for <u>Pauli stabilizer code</u> by using

$$H_0 \to -\sum S_i$$

↳ each S_i is Pauli string (e.g. $X_1Y_2I_3Z_4Z_5\cdots$)

and $[S_i, S_j] = 0$.

To build up to this point we need to go through a bit of a formal detour w/ lots of definitions...

But remember that we already saw how the Shor code fits into this picture! The main goal today is to build a framework that cleanly allows us to generalize.

Def: Group is a set G w/ multiplication: $g_1 \cdot g_2 \in G$
identity $1 \in G$: $1 \cdot g = g \cdot 1 = g \quad \forall g$
inverse: $g^{-1} \in G$: $g \cdot g^{-1} = g^{-1} \cdot g = 1$
associative: $g_1(g_2 g_3) = (g_1 g_2) g_3$

Def: group G is Abelian $\Leftrightarrow$ $g_1 \cdot g_2 = g_2 \cdot g_1 \quad \forall g_1, g_2$.

Def: Pauli group is the group generated by Pauli strings:

$$P_n = \{ I_1 X_2 Y_3 \cdots, -iZ_1 I_2 I_3, \cdots \}$$

↖ can have prefactors $\{\pm 1 \text{ or } \pm i\}$

This forms a non-Abelian group since (on a single site)

$X^2 = Y^2 = Z^2 = I, \quad IX = XI, \quad XZ = -iY \quad (iZ)(iX) = -iY$, etc.

Def: Pauli stabilizer code is a quantum code whose codespace

$$\mathcal{C} = \{ |\psi\rangle \in (\mathbb{C}^2)^{\otimes n} : S|\psi\rangle = |\psi\rangle \quad \forall S \in \mathcal{S} \}$$

where $\mathcal{S} \leq P_n$ is the stabilizer group of the code

↖ is a subgroup

A key question we'll be turning to, then, is how to make good choices for $\mathcal{S}$.

Prop: $\mathcal{S}$ is Abelian group

Proof: $S_1 S_2 |\psi\rangle = S_1 |\psi\rangle = |\psi\rangle = S_2 |\psi\rangle = S_2 (S_1 |\psi\rangle)$.

↑ shows $S_1, S_2 \in \mathcal{S}$

But for any two $S_1, S_2 \in P_n$, $S_1 S_2 = \pm S_2 S_1$

on each site Paulis commute or anticommute....
$XX = XX, \quad YI = IY, \quad XZ = -ZX$ etc.

$\Rightarrow S_1 S_2 = S_2 S_1$ as operators if $S_1, S_2 \in \mathcal{S}$.

Prop: if $S \in \mathcal{S}$, then $-S, \pm iS \notin \mathcal{S}$.

Notice that the classical codes from the start of the lecture fit neatly into this framework!

Now let's talk about why stabilizer codes are so useful!

Let $S_1, \dots S_m$ $(m < n)$ be independent generators for $\mathcal{S}$ w/ ± 1 prefactors

$\Rightarrow$ every $S \in \mathcal{S}$ is a product of S_j's

$\Rightarrow$ no $S_j = S_{i_1} \cdots S_{i_\ell}$ identity exists

$S_j S_j = I$

Prop: $\dim(\mathcal{C}) = 2^{n-m} = 2^k$, where $k = \#$ of logical qubits

Proof: projector onto codespace is

$$P_{\mathcal{C}} = \prod_{j=1}^{m} \frac{I + S_j}{2}$$

since any non-identity Pauli has vanishing trace

$$\dim(\mathcal{C}) = \mathrm{tr}(P_{\mathcal{C}}) = 2^{-m} \sum_{S \in \mathcal{S}} \mathrm{tr}(S) = 2^{-m} \cdot \mathrm{tr}(I) = 2^{n-m}$$

∎

Replacing projector above w/ projector onto any other syndrome sector we have:

Cor: 2^k states w/ every syndrome $(\eta_1, \dots, \eta_n) \in \{\pm 1\}^n$.

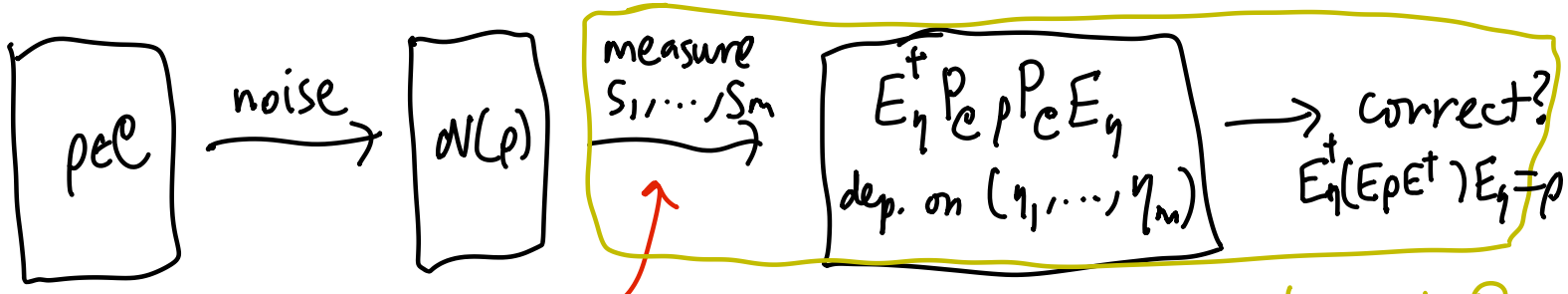
Similar to what we did w/ Shor code, now observe that:

Prop: if $E \in P_n$ and $|\psi\rangle \in \mathcal{C}$, then $E|\psi\rangle$ is an eigenvector of all S_j, w/

$$S_j(E|\psi\rangle) = \eta_j (E|\psi\rangle) \quad \text{if} \quad S_j E = \eta_j E S_j$$

η_j: $+1$ commute, -1 anticommute

As insinuated by KL conditions, we can then measure all the η_j and use outcomes to determine and fix error...



order doesn't matter since they commute

recovery channel $\mathcal{R}$.

error E is <u>detectable</u> if $\eta_j = -1$ for at least one j ($\{E, S_j\} = 0$)

<u>undetectable</u> if all $\eta_j = +1$ but $EP_C \neq P_C$ (KL!)

$[E, S_j] = 0$ for all j.

Our goal is now to better understand how to actually do QEC on a stabilizer code. Let's first start by classifying all the Pauli operators by how they act on the code...

<u>Prop:</u> $\left|\{E \in P_n : S_j E S_j = \eta_j E \ \forall j\}\right| = 4 \cdot 2^{n+k}$

Proof: $|P_n| = 4^{\text{phase}} \cdot 4^n$ ($\{I, X, Y, Z\}$ on each qubit.)

Each $E \in P_n$ belongs to one set.

Let $N(S) = \{E : S_j E S_j = E\}$. If $S_j F S_j = \eta_j F$

"normalizer" (here, equivalent to centralizer...) $\Rightarrow$ $S_j(FE)S_j = \eta_j(FE)$.

So each $|\{E : \cdots\}| = |N(S)|$ is identical.

By corollary, there are 2^{n-k} syndromes reachable $\Rightarrow |N(S)| = \frac{4 \cdot 4^n}{2^{n-k}}$ ∎.

Key interpretation: $4 \cdot 2^{n+k} = 4^{\text{phase } \{\pm 1, \pm i\}} \cdot 2^{n-k} \cdot 2^{2k}$

$2^{n-k} \in S$; 2^{2k} ← logical operator

Given any $Q \in N(S)$, $|\psi\rangle \in C$, indeed

$$QS|\psi\rangle = Q|\psi\rangle \quad \forall\, S \in S$$

stabilizer equivalent operators

$|\psi\rangle$ equivalent to $e^{i\phi}|\psi\rangle$ $\left[\rho \to e^{i\phi}\rho e^{-i\phi} = \rho\right]$

What's left $\Rightarrow$ logical operators! $(2^k)^2$ such Hermitian op.

More generally this discussion implies that:

Prop: error $E_1, E_2 \in P_n$ equivalent $\Leftrightarrow$ $E_1 = e^{i\phi} E_2 S$, $S \in S$.

Proof: KL condition! $P_C E_2^\dagger E_1 P_C \neq 0 \Rightarrow E_2^\dagger E_1 \in N(S)$

Now to rule out a logical operation:

Group theory: $EP_C \propto P_C \Leftrightarrow E \in S$ (up to phase)

$$\hookrightarrow = \frac{1}{2^{n+k}} \sum_{S \in S} ES = \frac{1}{2^{n+k}} \sum_{S \in S} S$$

Paulis match on both sides, $ES \in S$ $\forall S \Rightarrow E \in S$. ▨

Problem: given syndrome $\{\eta_j\}$, what $E \in P_n$ to apply?

OK: $E \to ES$ for $S \in S$

bad: $E \to EL$ for $L \in N(S) \setminus (S \times \{\pm 1, \pm i\})$ ← "set minus"

Def: code distance $d :=$ smallest weight logical Pauli

↳ # of non-I sites.

Prop: correct every error w/ weight $\leq \lfloor \frac{d-1}{2} \rfloor$.

Proof: two such errors E_1, E_2 have weight $(E_2^\dagger E_1) \leq 2\lfloor \frac{d-1}{2} \rfloor < d$.

So we can't have a logical. This is identical to the analogous proof for a classical code!

So if $P_C E_1^\dagger E_2 P_C \propto P_C$, $E_1^\dagger E_2 \in \mathcal{S}$ (up to phase).

or: errors have different syndromes η_j and distinguishable ∎

Let's briefly revisit the Shor code to see how all of this fits together. (May not have time in class, but suggest students look @ notes!)

Shor code: stabilizers $X_1X_2X_3X_4X_5X_6$, $X_4X_5X_6X_7X_8X_9$, Z_1Z_2, Z_2Z_3, Z_4Z_5, Z_5Z_6, Z_7Z_8, Z_8Z_9 } generate $\mathcal{S}$

Codespace has $k=1$ since 8 independent stabilizers

(we already saw they commute before!)

$N(\mathcal{S})$ generated by (up to phase) $\mathcal{S}$ and logicals $X_1X_2X_3$, $Z_1Z_4Z_7$

Logical representatives not unique! Can take

$X_1X_2X_3\,(X_1X_2X_3X_4X_5X_6) \to X_4X_5X_6$ etc.

But one can show that $\underline{d=3}$ for this code ⇒ correct 1 error.

Recall that not all single errors are indistinguishable:

Z_1 has same syndrome as $Z_1(Z_1Z_2) = Z_2$

⇒ same correction scheme for both!