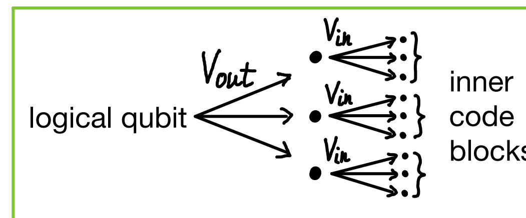
Error threshold for quanum memory

In the previous lecture, we assumed that the faults are sparse, i.e. that two faults cannot happen in adjacent gadgets. If the the physical fault rate is p (using the stochastic error model), then the probability of such an unfortunate event is $O(p^2)$ per gadget. In this lecture, we will achieve an arbitrary small logical error rate, assuming that $p<\varepsilon$ for a certain threshold value ε .

Achieving logical error rate $O(p^2)$	Achieving arbitrary small logical error rate if $p<\varepsilon$
Automatic using LDPC codes	Automatic using large-size surface codes and <u>optimal decoding</u>
Fault-tolerant error-correction circuit using a distance 3 stabilizer code	Hierarchical construction based on concatenation of even self-dual CSS codes

Concatenated codes

$$V = V_{in}^{\otimes n} V_{out} \qquad \left(\text{e.g. } V_{Shor} = V_{rep}^{\otimes 3} V_{drep}\right)$$



Errors from which they protect

Let $\mathcal{E}_{in}$ be arbitrary, $\mathcal{E}_{out} = \mathcal{E}(n, r)$, $\quad 2r < d_{out}$

$\mathcal{E}$ is the linear span of $E_1 \otimes \cdots \otimes E_n$:

inner errors

r inner errors are arbitrary, the other belong to $\mathcal{E}_{in}$

Multiple levels of concatenation: $V_k = V^{\otimes n^{k-1}} \cdots V^{\otimes n} V,$ where V protects from r errors

We assume that V is a self-dual CSS code of type *[[n,1,3]]*, e.g. Steane's 7-qubit code. Thus, r=*1.*

Two-tier classification of errors (which determines the corresponding subsystem codes)

$$\underbrace{\mathcal{E}_k^{(good)}}_{\text{"good" errors at level } k} \subseteq \underbrace{\mathcal{E}_k^{(OK)}}_{\text{OK errors at level } k}$$

$$\mathcal{M}_k = \text{Image}(V_k) \subseteq \mathcal{M}_k^{(good)} = \mathcal{E}_k^{(good)}\mathcal{M} \subseteq \mathcal{M}_k^{(OK)} = \mathcal{E}_k^{(OK)}\mathcal{M}$$

This is a technical trick to be used in the proof of the threshold theorem. A fault-tolerant circuit usually cleans up most errors so that the residual error pattern is "good". But some rare combinations of faults may result in more errors remaining. Such errors patterns are considered OK if they are likely to be corrected in the future.

OK errors are defined recursively:

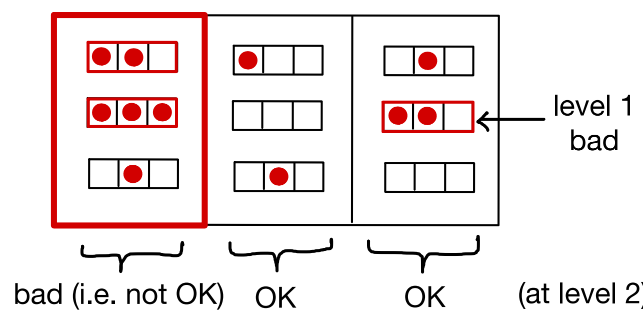
OK errors at level k = linear span of $E_1^{(k-1)} \otimes \cdots \otimes E_n^{(k-1)}$

level k-1 errors

all of the level k-1 errors are OK, except maybe one

OK error at level 0 = $\{cI : c \in \mathbb{C}\}$

Example: 3 levels, *n=3*



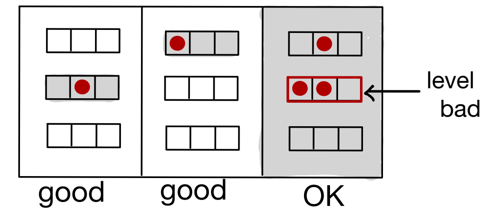
Good errors are defined recursively using the definition of OK errors:

Good errors at level k = linear span of $E_1^{(k-1)} \otimes \cdots \otimes E_n^{(k-1)}$:

all of $E_1^{(k-1)}, \dots, E_n^{(k-1)}$ are good, except maybe one, which must still be OK

Good error at level 0 = OK error at level 0

Example (grey = OK but not good):



Bounds on the probabilities of bad events if errors hit the physical qubits independently with probability p

$$p_k := \Pr_{E^{(k)}}[E^{(k)} \text{ is not OK}] = \Pr_{E_1^{(k-1)}, \dots, E_2^{(k-1)}}[\text{at least two of } E_1^{(k-1)}, \dots, E_2^{(k-1)} \text{ are not OK}] \leq \underbrace{\binom{n}{2}}_{a^{-1}} p_{k-1}^2$$

$$\frac{p_k}{a} \leq \left(\frac{p_{k-1}}{a}\right)^2 \leq \cdots \leq \left(\frac{p_0}{a}\right)^{2^k}, \qquad p_0 = p \quad \Rightarrow \quad \boxed{p_k \leq a\left(\frac{p}{a}\right)^{2^k} \quad \text{Threshold property: } \lim_{k\to\infty} p_k = 0 \text{ if } p < a}$$

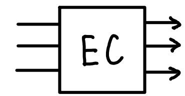
$$q_k := \Pr_{E^{(k)}}[E^{(k)} \text{ is not good}] \leq \underbrace{\binom{n}{2} q_{k-1}^2}_{\text{two of } E_1^{(k-1)}, \dots, E_n^{(k-1)} \text{ are not good}} + \underbrace{n\, p_{k-1}}_{\text{one of } E_1^{(k-1)}, \dots, E_n^{(k-1)} \text{ is not OK}} \Bigg\} \Rightarrow q_k \sim \left(\frac{p}{b}\right)^{2^{k-1}} \text{ for some } b$$

Concatenated circuits (gadgets)

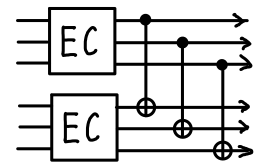
We use the basic gadgets operating on the original *n*-qubit code to construct level *k* gadgets

Some of the basic gadgets

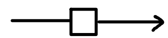
Fault-tolerant error correction circuit, which is build of Clifford gates and measurements in the $|0\rangle, |1\rangle$ basis:



Transversal realization of CNOT:

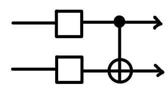


Level 0 gadgets: $I_0 =$



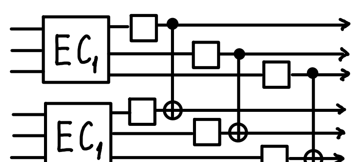
(Leakage cleanup: both needed in practice and provides useful analogy with higher levels)

$CNOT_0 =$



Level *k* gadget (for *k*>0): Replace all gates in the corresponding basic gadget with their level k-1 realization

For example, $CNOT_1 =$



not resolved in this picture: EC_1 contains a lot of gates

Requirements for the basic EC circuit

If there are no faults:

-- The output is always in the code subspace $\mathcal{M}$

-- If the input represents a codevector with at most one error, the error is corrected,
i.e. the circuit realizes the identity operator from encoding $\tilde{V}$ to encoding V .

In general (assuming up to 1 fault):

-- The output is in the subsystem code $\tilde{\mathcal{M}}$ that allows up to 1 error

-- If the input is a codevector, then the output is the same codevector with 1 possible error,
i.e. the circuit realizes the identity operator from encoding V to encoding $\tilde{V}$.

Similar properties hold for other basic gadgets:
instead of the identity operator, one should consider the gate the gadget realizes.

Generalizing properties of errors and faults from the basic code to concatenated codes

No error (code $\mathcal{M}$) $\longrightarrow$ Good error

At most one error (subsystem code $\tilde{\mathcal{M}}$) $\longrightarrow$ OK error

No faults $\longrightarrow$ Good fault pattern

At most one fault $\longrightarrow$ OK fault pattern

Defined by analogy with good an OK errors

Properties of concatenated circuits

If the faults pattern is good:

-- The output always belongs to the subsystem code $\mathcal{M}_k^{(good)}$

-- The circuit implements the desired logical gate from encoding $V_k^{(OK)}$ to encoding $V_k^{(good)}$

If the faults pattern is OK:

-- The output always belongs to the subsystem code $\mathcal{M}_k^{(OK)}$

-- The circuit implements the desired logical gate from encoding $V_k^{(good)}$ to encoding $V_k^{(OK)}$

Probability estimates (neglecting constants like *n=7*)

$\Pr[\text{fault pattern is not OK}] = (O(p))^{2^k}$ $\qquad$ $\Pr[\text{fault pattern is not good}] = (O(p))^{2^{k-1}}$

We may self-consistently assume (and later verify) that $\Pr[\text{error pattern is not good}] = (O(p))^{2^{k-1}}$

A **logical error** occurs when:

the fault pattern in the current gadget is not OK $\left(\text{Prob} = (O(p))^{2^k}\right)$, or

the input is not good and the fault patterns is not good $\left(\text{Prob} = (O(p))^{2^{k-1}} \cdot (O(p))^{2^{k-1}}\right)$

$$\boxed{\Pr[\text{logical error}] = (O(p))^{2^k} \sim \left(\frac{p}{\varepsilon}\right)^{2^k}}$$

Cost of fault tolerance

Let p_* be the maximum error probability per logical gate we can tolerate

$$p_* \sim p_k \sim \left(\frac{p}{\varepsilon}\right)^{2^k} \Rightarrow k \approx \log_2 \frac{\log p_*}{\log(p/\varepsilon)}$$

$m_k =$ # of gates in a level k gadget $\sim C^k \sim \left(\frac{\log p_*}{\log(p/\varepsilon)}\right)^{\alpha}$, where $\alpha = \log_2 C$

$$\boxed{m_k \backsim \left(\log \frac{1}{p_*}\right)^{\alpha}}$$