Local noise

Noisy quantum circuit realized as a unitary U_{SB} between the system S and bath B.

$U_{SB} = \Pi_{j=1}^{L} U_{SB}^{j}$

where U_{SB}^{j} = sys-bath evolution in the j-th location.

$U_{SB}^{j} = (\text{alpha}_j \, I I + \beta_j F^j) \, [(U_S^j)_{ideal} \otimes V_B^j]$

where F^j = Pauli decomposition with nontrivial Pauli on S.

U_{SB} has an expansion with 2^L terms (each called a fault path). (One of them is ideal).

For any particular set of r locations I_r, let $\Gamma(I_r)$ be the subsum in the expansion of U_{SB} with faults at all r locations in I_r (unrestricted elsewhere).

Then, noise is local with strength ε if $\forall I_r$, $|| \, \Gamma(I_r) \, ||_\infty \leq \varepsilon^r$ (where $|| \, . \, ||_\infty$ is the largest singular value).

NB Local noise can be nonMarkovian.

Lemma:

If noise is local with strength ε, circuit size L, and Γ_r denote subsum over fault paths with r or more faults, then

$$|| \Gamma_r ||_\infty \leq \text{L-choose-r } \varepsilon^r e^{(L-r)\varepsilon}$$

Proof: elaborate counting involving inclusion-exclusion principle.

$\Gamma(I_r)$ & Γ_r are both subsums of fault paths.

Former: r specific locations all faulty
Latter: any $s \geq r$ faults

Special case: stochastic noise (no interference between fault paths), and a fault path has prob = amplitude-squared.

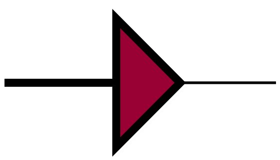
Even more special: each location has independent noise wp p.

Last time:

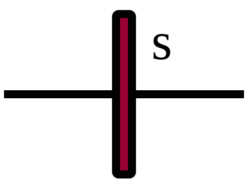
level 0 circuit: consists of 0-Rec in each location (including I)

level 1 circuit: replace each 0-Rec in level-0 circuit by corresponding 1-Rec.

level k circuit: obtained from a level k-1 circuit by replacing 0-Recs with 1-Recs.



Ideal decoder — Corrects errors & decodes state producing an unencoded qubit.



s-Error filter — Projects on states that are within s Pauli errors of a valid codeword

(Confusion last class: p13-17, add r+s $\leq$ t etc in FT conditions.)

Lemma 1: If Ex-Rec good ($\leq$ t faults), then Rec is correct

Corollary of lemma 1: if all 1-Ex-Recs are good, the level-1 simulation realized by the 1-Recs inside can be replaced by a level-0 simulation with each 1-Rec $\rightarrow$ ideal 0-Rec.

Corollary of above: it requires at least t+1 faults in ONE particular ex-Rec for the 1-simulation to fail.

1-Ex-Rec with size C fails with noisy strength ε_1=(κ C-choose-t) ε_0^t given local noise with strength ε_0.

Want to run this recursively : starting from a k-simulation of a circuit, replace the 1-Recs by 0-Recs (with stength $\leq \varepsilon_1$) that form the corresponding (k-1)-simulation.

Then, we apply the argument to the (k-1)-simulation etc.

2 missing pieces:

Good 1-ex-Recs implies 1-Rec simulates correct 0-Rec.
What about bad 1-ex-Recs? Does it simulate a fault with properties simulation to the 0-level?

ex-Recs overlap. 1 fault in the shared EC can make
2 (multiple) ex-Recs to be bad.

Does 1-Rec in a bad 1-ex-Recs simulate a 0-fault?

No .... the fault incurred depends on the syndrome.

e.g in the 3-bit repetition X error code, having 2 X errors on $\alpha\,|000\rangle + \beta\,|111\rangle$ takes it to $\alpha\,|011\rangle + \beta\,|100\rangle$ and decoding gives an error syndrome corr to X erro on qubit-1, and output $\alpha\,|111\rangle + \beta\,|000\rangle$ with encoded X error. Similarly for 3-bit Z error code.

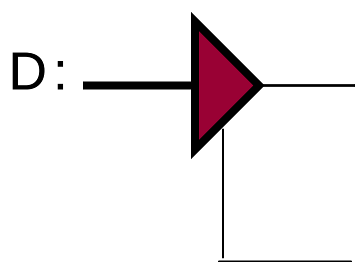
Now, take the 9-bit code, and a supposition of multiple X & Z errors (enough to cause a logical X or Z error when decoded). The syndrome collapses the error into either many X's or many Z's and the subsequent logical X or logical Z error depends on whether we get the erroneous X syndromes or Z syndromes.

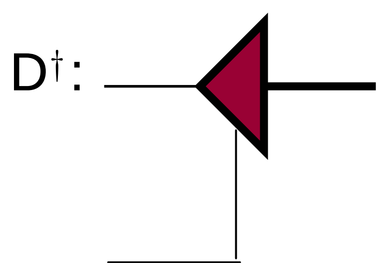
We don't have a problem if #errors small, since in each case, the correct state comes out. But trouble comes for > t errors.

In general, with $\leq$ t errors, the decoded state and the simulated 0-Rec is independent of syndrome, and with > t errors, it is dependent on syndrome.

Rough idea: redefine IDEAL decoder to (1) retain the syndrome, and (2) preserve the coherence -- a measurement first unitarily interacts system with ancilla then measures in X/Z basis. Skip the last step.

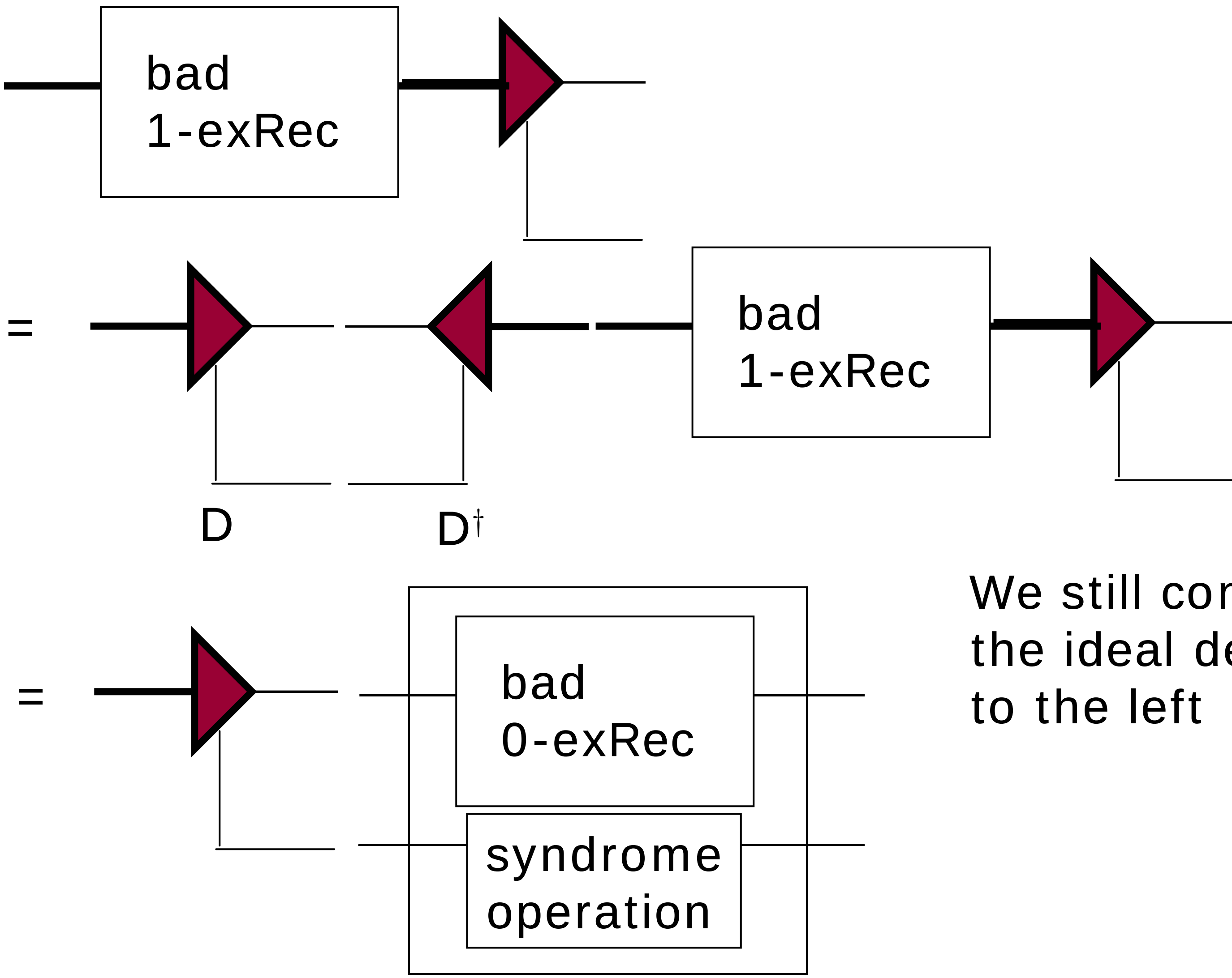
Ideal decoder is now isometric. Extend to a unitary D. Then, $D^\dagger$ is an encoder with 2 inputs: the codespace and syndrome space.





(a) New definition preserves [Good $\rightarrow$ Correct lemma], since syndrome factors out for 1-exRecs.

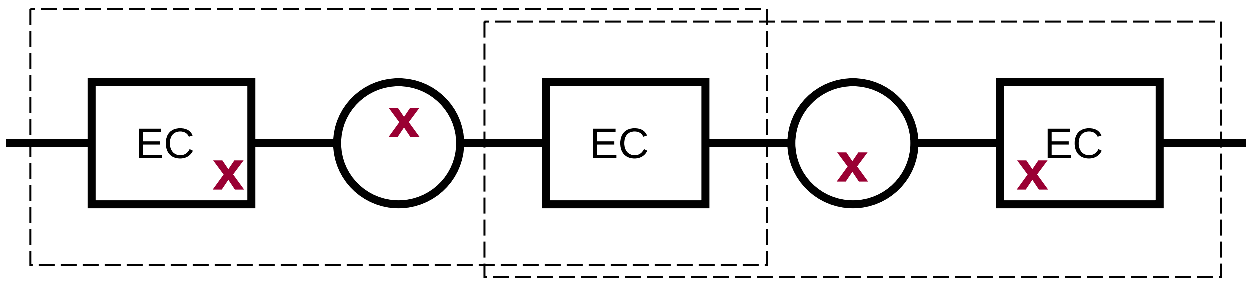
(b) For bad ex-Recs, cannot use lemma, but can insert identity:



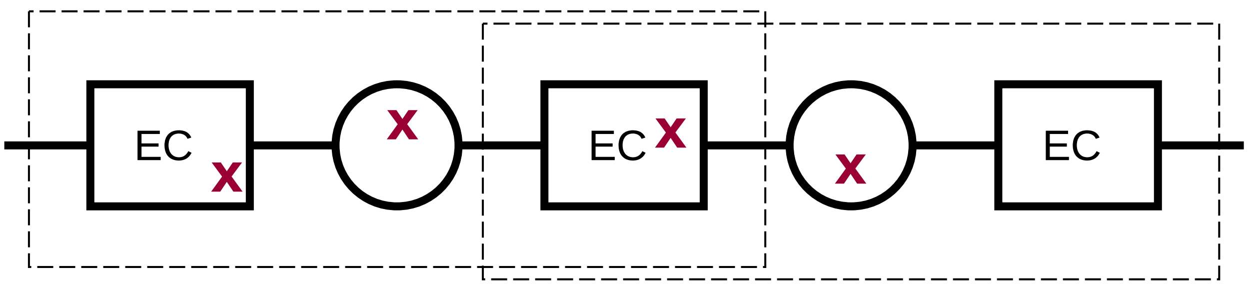
We still commute the ideal decoder to the left

The syndrome for different bad exRecs on the same logical qubit interact with one another. But all we want is 0-noise remain local … (OK to be correlated).

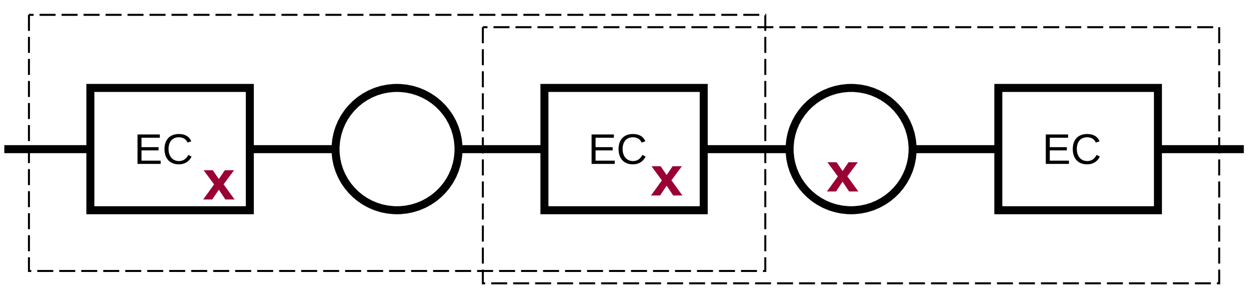
2nd issue: ex-Recs overlap. e.g.:



two independent bad ex-Recs

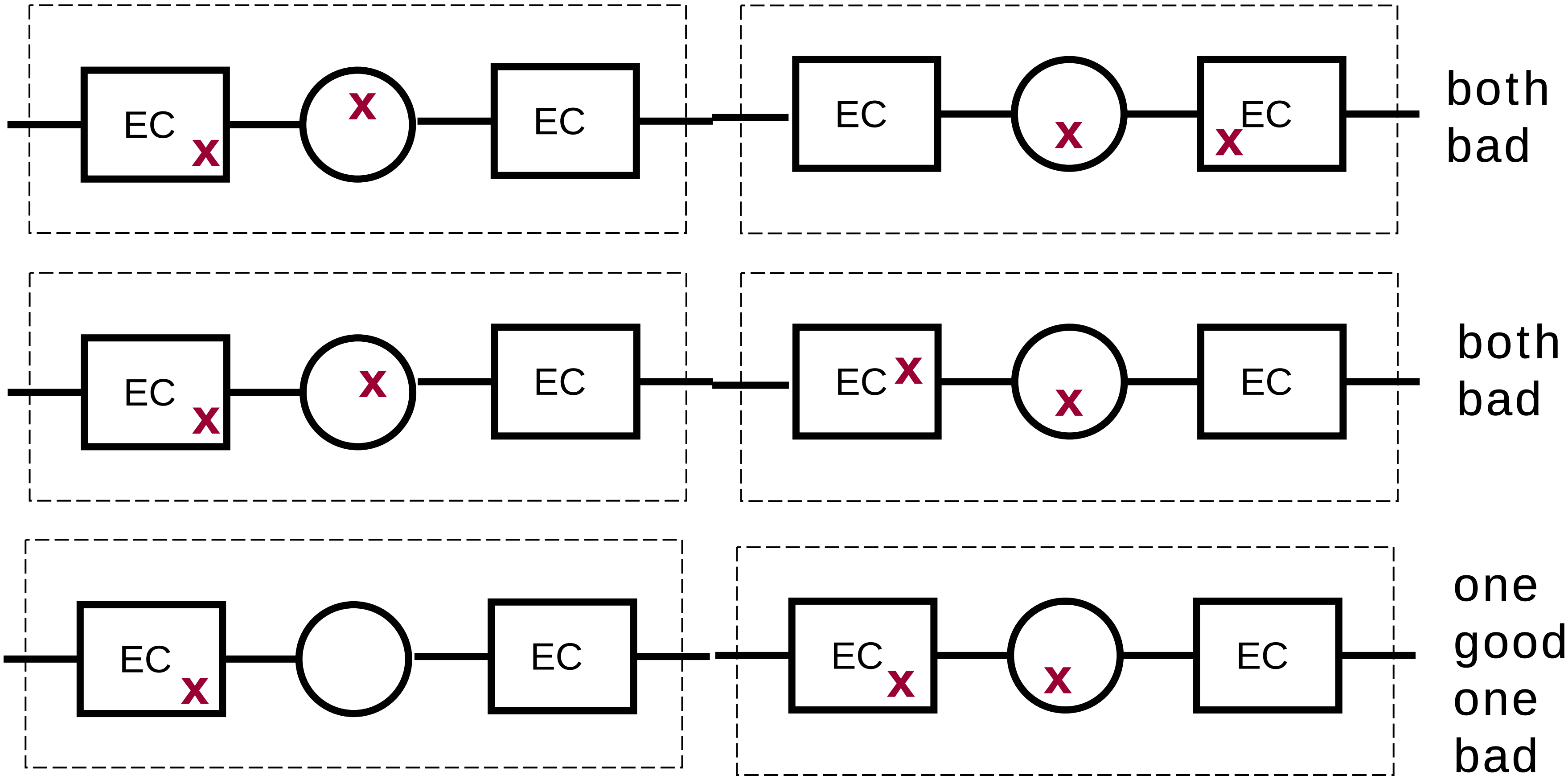


two independent bad ex-Recs



dependent bad ex-Recs

Idea: when 2 consecutive ex-Recs are bad, truncate the trailing EC from the first ex-Rec, reinsert ideal EC.
Decide good/bad as defined before



Then $(t+1)^r$ independent faults are needed to create r bad ex-Recs. (Proof omitted.)

The 2 issues are now solved to make 1-Rec simulate 0-Rec, preserving local noise structure, and modifying the strength from ε_0 to ε_1.

Big picture: start with k-sim of a circuit.
Consider the meas 1-exRecs at the end.
If bad, replace by an ideal decoder followed by a bad measurement (that can depend on the syndrome).
If good, move the ideal decoder to the left.
Repeat for all other 1-exRecs until the ideal decoders hit the far-left -- the beginning of the k-sim (state preparation).

Finally, run this recursively.

from k-sim to (k-1)-sim:

$\varepsilon_0 \to \varepsilon_1 = (\text{C-choose-t})\ \varepsilon_0^{t+1}$ where C=size of largest 1-ex-Rec

$\varepsilon_1 \leq \varepsilon_0$ iff $r_0 = \varepsilon_0 / \varepsilon_{crit} \leq 1$ where $\varepsilon_{crit} = (\kappa\ \text{C-choose-t})^{-1/t}$

1 if stochastic
const $\geq$ 1 else

equivalently:

$\varepsilon_0 / \varepsilon_{crit} \to \varepsilon_1 / \varepsilon_{crit} = (\kappa\ \text{C-choose-t})\ \varepsilon_0^{t+1} / \varepsilon_{crit} = \varepsilon_{crit}^{-t-1}\ \varepsilon_0^{t+1} = r_0^{t+1}$

from k-sim to 0 sim:

$\varepsilon_k / \varepsilon_{crit} = (((r_0^{t+1})^{t+1}) \ldots)^{t+1} = \ldots (r_0)^{(t+1)^k}$

Note, for a size L circuit, to achieve overall prob of error $\leq \delta$, each k-Rec should has error $\leq \delta/L$, and k = O(log log L/δ). Resources scale by a multiplicative factor of $O(n^k)$ where n is the block size of the underlying code, and $O(n^k) \sim \log L/\delta$

Level reduction holds for many other noise models:

Leakage errors

Qubits drift out of the 2-dim space

Measurement based QC models

Threshold theorem is proved.

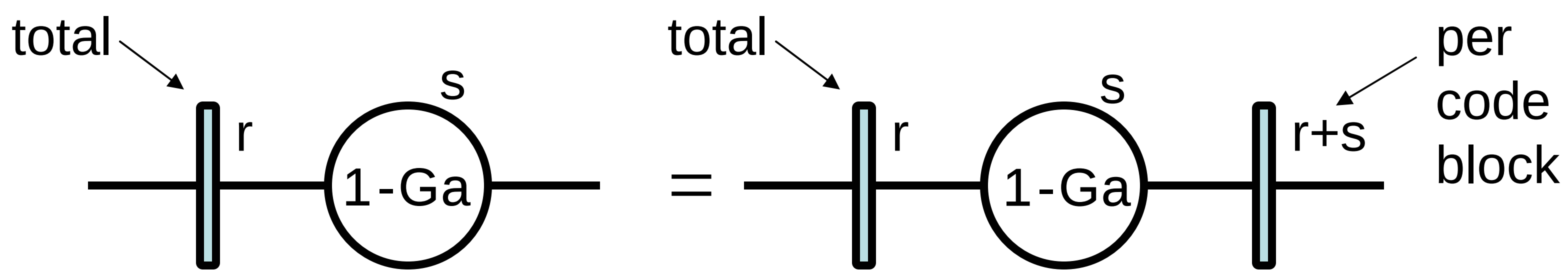
If the gadgets satisfying the FT properties can be obtained, we believe that QC can be performed despite some noise in the physical implementation.

Now, next task -- to obtain those gadgets.

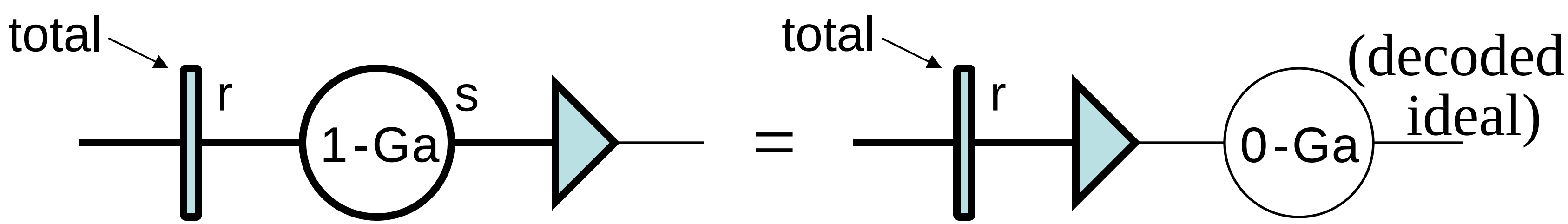
Will not do everything needed in this class, but to show some of the ideas etc. Punge line -- they're all possible, with the best provable threshold $\approx$ 0.1% stochastic local noise prob.

Building the FT gadgets …

Ga FT-1: Faults propagate benignly if $r+s \leq t$



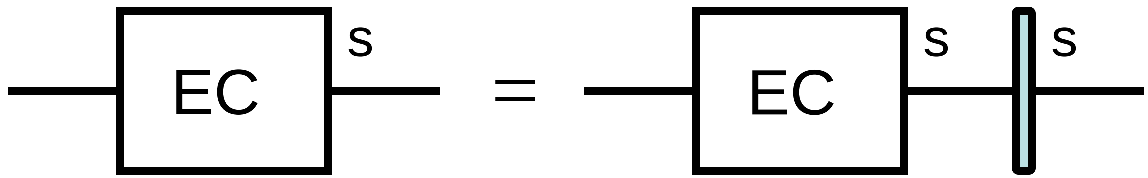
Ga FT-2: performs the encoded gate ideally if $r+s \leq t$



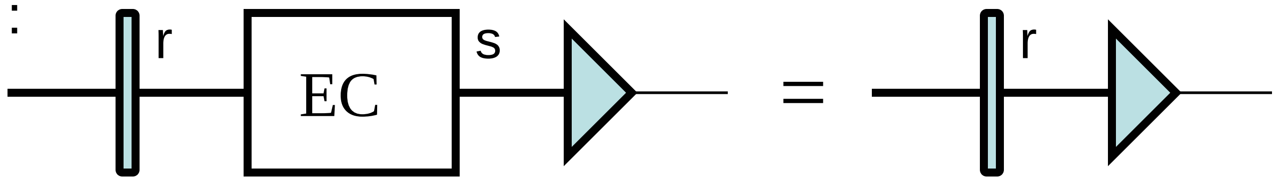
The bitwise X, Z, P, H, CNOT for the 7-bit code works :)
HW: prove that for H and CNOT.

How to build EC satisfying EC FT 1 and 2?

for $s \leq 1$:



for $s+r \leq 1$:



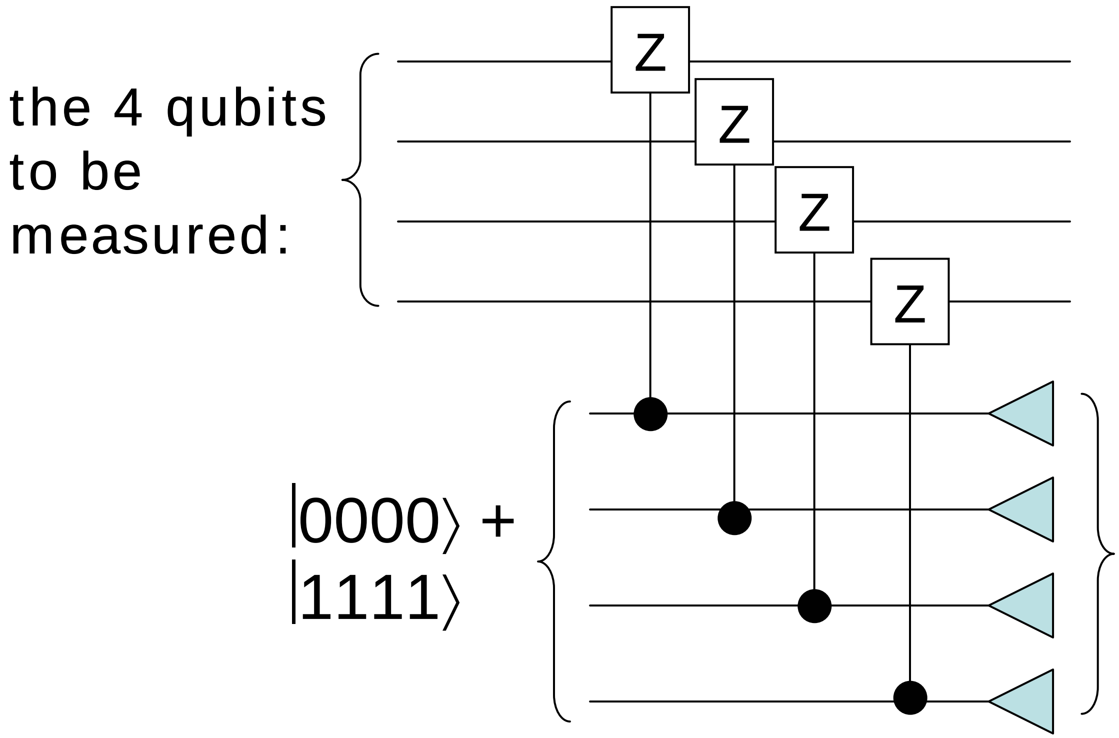
Back to t=1, 7-bit code. s=0 easy. So, consider s=1.

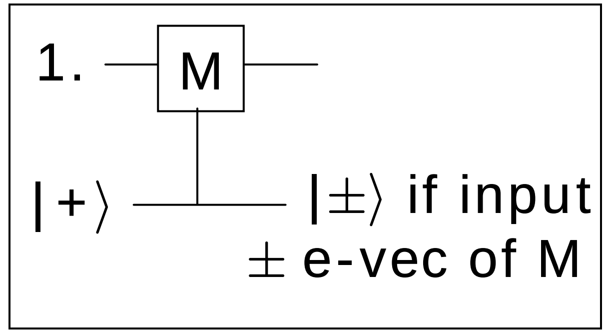
Consider the 2nd property. s=1 so r=0. Incoming state is error free. EC contains syndrome measurement boxes, perfect classical computation, and possibly a 1-qubit Pauli correction. If syndrome meas has no faults, 1 fault in the correction puts 1 error in the output state. Remains to see what happens in syndrome measurements up to 1 fault.

Fault tolerant measurements:

Only worry about s = 1

e.g. measure ZZZZ on 4 specific qubits in the codeblock:





2. $|0000\rangle \pm |1111\rangle$ = $\pm$ 1 eig-vec of XXXX. Measure all 4 X's and multiply to get the ans

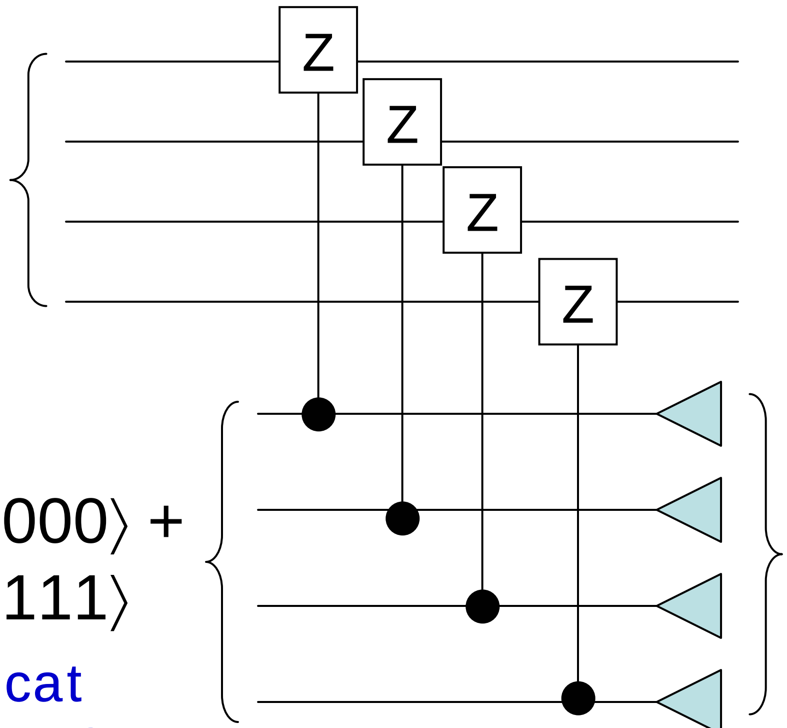
3. If 4-cat state is perfect, one faulty c-Z affects one data qubit. After c-Z, 1 storage/meas error in ancilla at worst flips y_1, but data block unharmed.

So, repeat 3 times, take majority of y_1, y_2, y_3

Fault tolerant measurements:

e.g. measure ZZZZ on 4 specific qubits in the codeblock:

the 4 qubits to be measured:

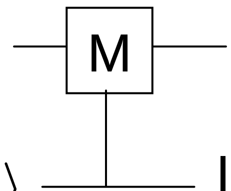


$|0000\rangle$ + $|1111\rangle$

take product $\rightarrow y_1$

X meas

1.



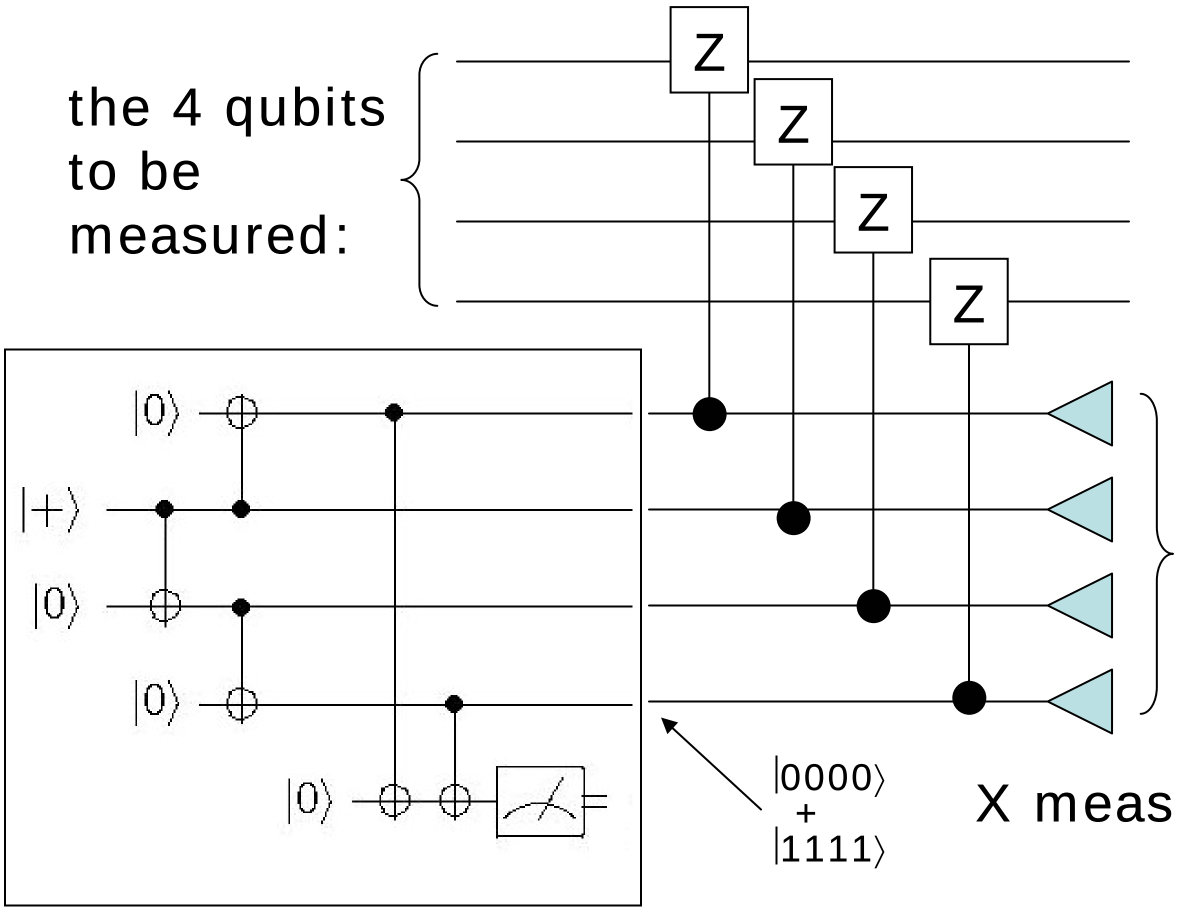
$|+\rangle$ $|\pm\rangle$ if input $\pm$ e-vec of M

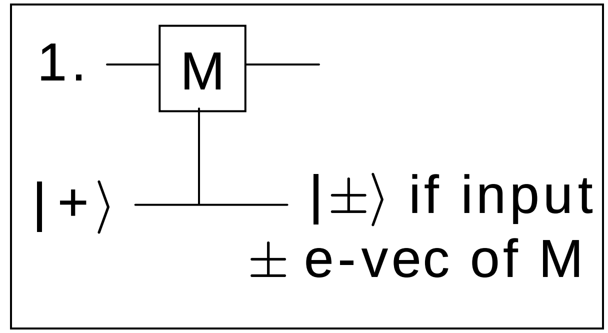
2. $|0000\rangle \pm |1111\rangle$ = $\pm$ 1 eig-vec of XXXX. Measure all 4 X's and multiply to get the ans

4. Errors in 4-cat state occurs BEFORE c-Z. 1 Z error negates the ± 1 eig-value of XXXX and flips the outcome y_1, but data block is unharmed. 1 X error propagates to 1 Z error to the data block but does not affect measurement. OK if there's only 1 X or Z error in all 3 ancillas.

Fault tolerant measurements:

e.g. measure ZZZZ on 4 specific qubits in the codeblock:





2. $|0000\rangle \pm |1111\rangle$ = $\pm$ 1 eig-vec of XXXX. Measure all 4 X's and multiply give the ans

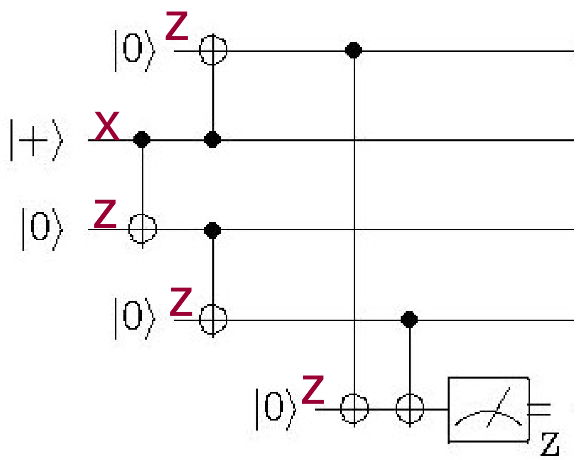
accept only if outcome =1

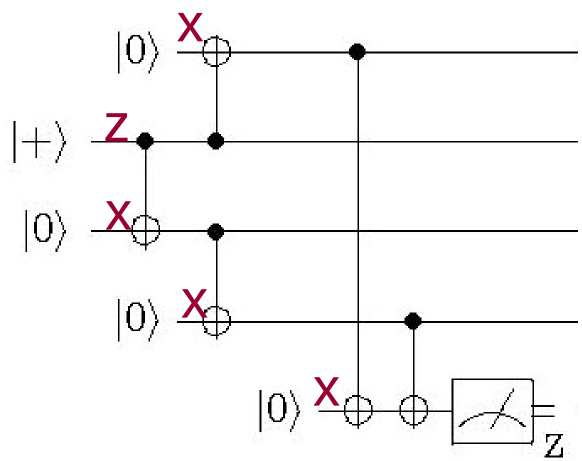
claim: at most 1 X or 1 Z error.

1. If noiseless, this prepares $|0000\rangle + |1111\rangle$
2. Output cannot have more than 2 X errors or 1 Z error.
3. 1 fault cannot result in 2 X errors:

USE s=1, that if there is a fault we analyze, no other faults are around.

1 fault at time step 1:





Each of above has no effect.

Y $\sim$ an X and a Z, same as an error in the RHS

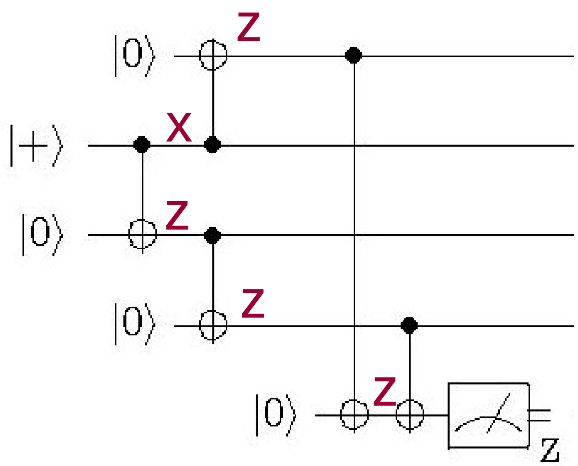
Each of above commutes pass CNOT. So, equivialent to 1 fault in 2nd step.

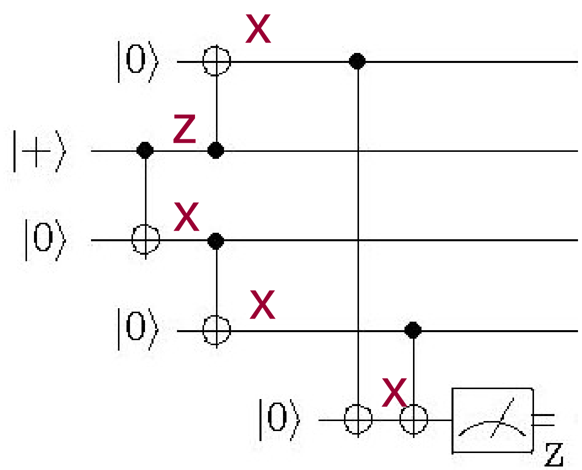
All case then covered in consideration for step 2.

1. If noiseless, this prepares $|0000\rangle + |1111\rangle$
2. Output cannot have more than 2 X errors or 1 Z error.
3. 1 fault cannot result in 2 X errors:

1 fault at time step 2:

1 fault in all subsequent steps results in 1 error. Thus [3] holds.





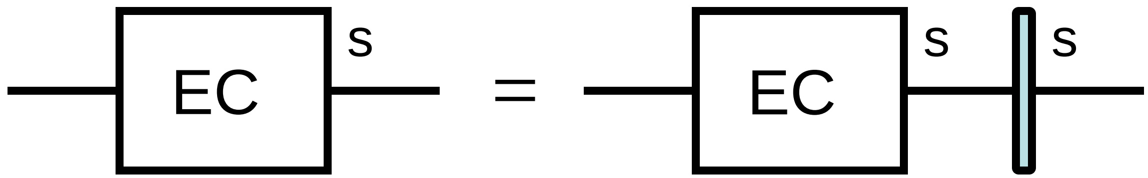
Each Z stays and affect the final state, but that's OK.

Right before the verification (last two CNOT's) $X_2 \rightarrow X_1 X_2$. $X_3 \rightarrow X_3 X_4$. Any of $X_{1,2,3,4}$ will result in meas=-1 & we reject. Same for the X in the verification step. Same for 1 Y.

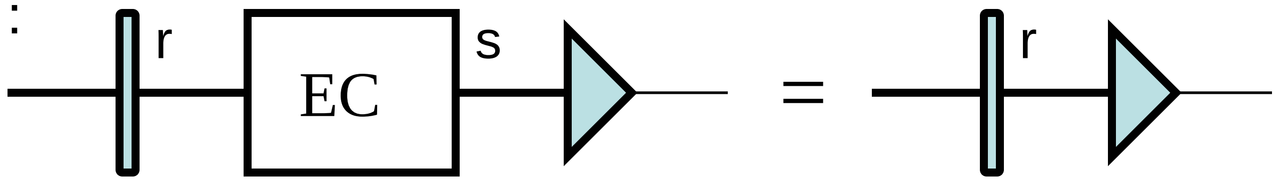
So, 2nd property proved. Omit discussion of 1st property.

How to build EC satisfying EC FT 1 and 2?

for s $\leq$ 1:



for s+r $\leq$ 1:

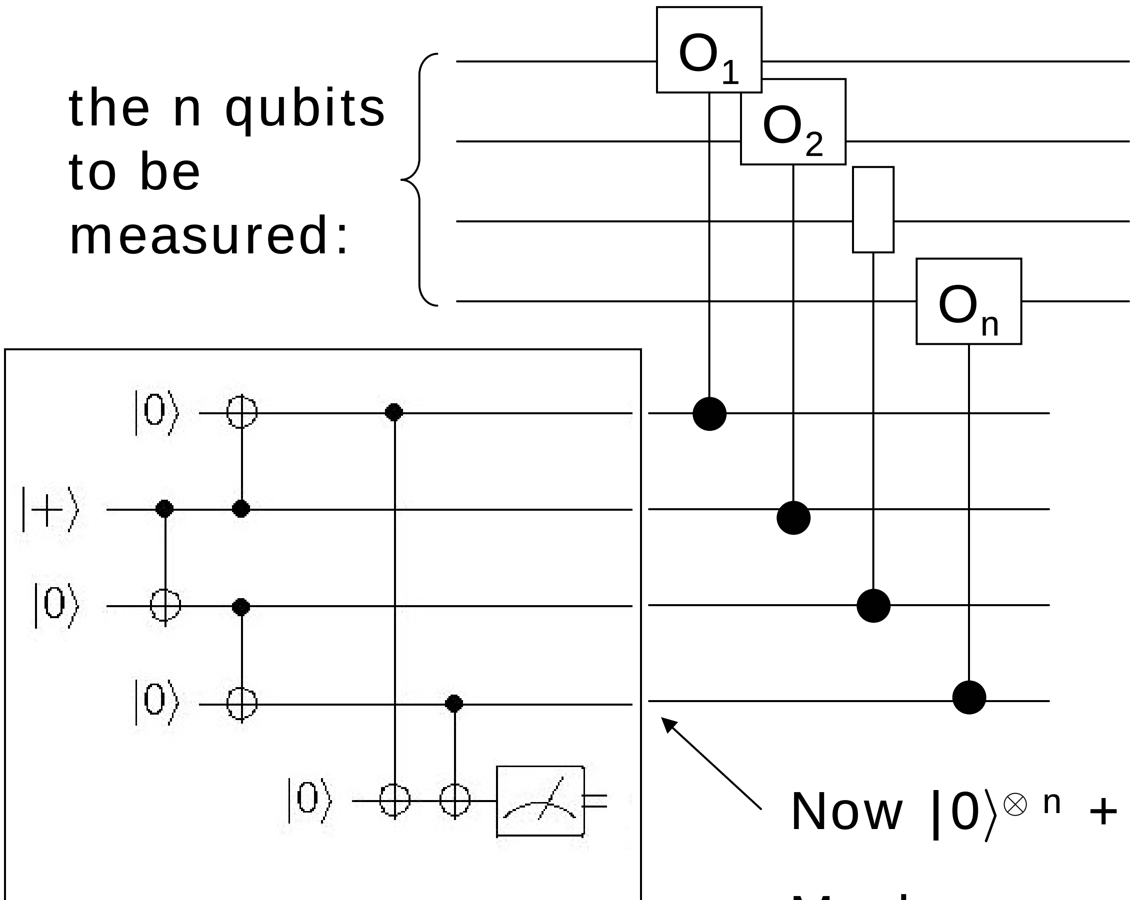


Consider s=1.

Consider the 2nd property. s=1 so r=0. Incoming state is error free. EC contains syndrome measurement boxes, perfect classical computation, and possibly a 1-qubit Pauli correction. If meas has no faults, 1 fault in the correction puts 1 error in the output state. Remains to see what happens in measurements up to 1 fault.

Same method works for all product observables $O_1 \otimes O_2 \otimes \cdots \otimes O_n$ with eigenvalues ± 1.

the n qubits to be measured:



Now $|0\rangle^{\otimes n} + |1\rangle^{\otimes n}$

Much more complicated preparation and verification. NB low wt measurements give higher threshold

Encoded non Clifford gate on the 7-bit code:

If $\theta = \pi/8$, $U = e^{iZ\pi/8}$, then $U X U^\dagger = e^{iZ\pi/4} X \in C_2$

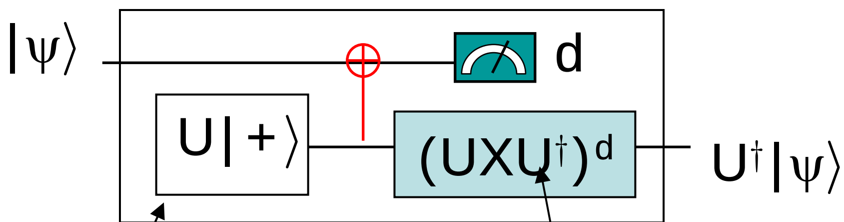
Circuit still holds if everything is in the encoded form !

THIS IS NOW DONE.

bitwise cnot

measure Z_L (wait until FT)

encoded input



Circuit performs encoded $U^\dagger$

state with stabilizer $UXU^\dagger$ (property (5)) will measure this stabilizer (wait until FT)

Clifford -- we know how to

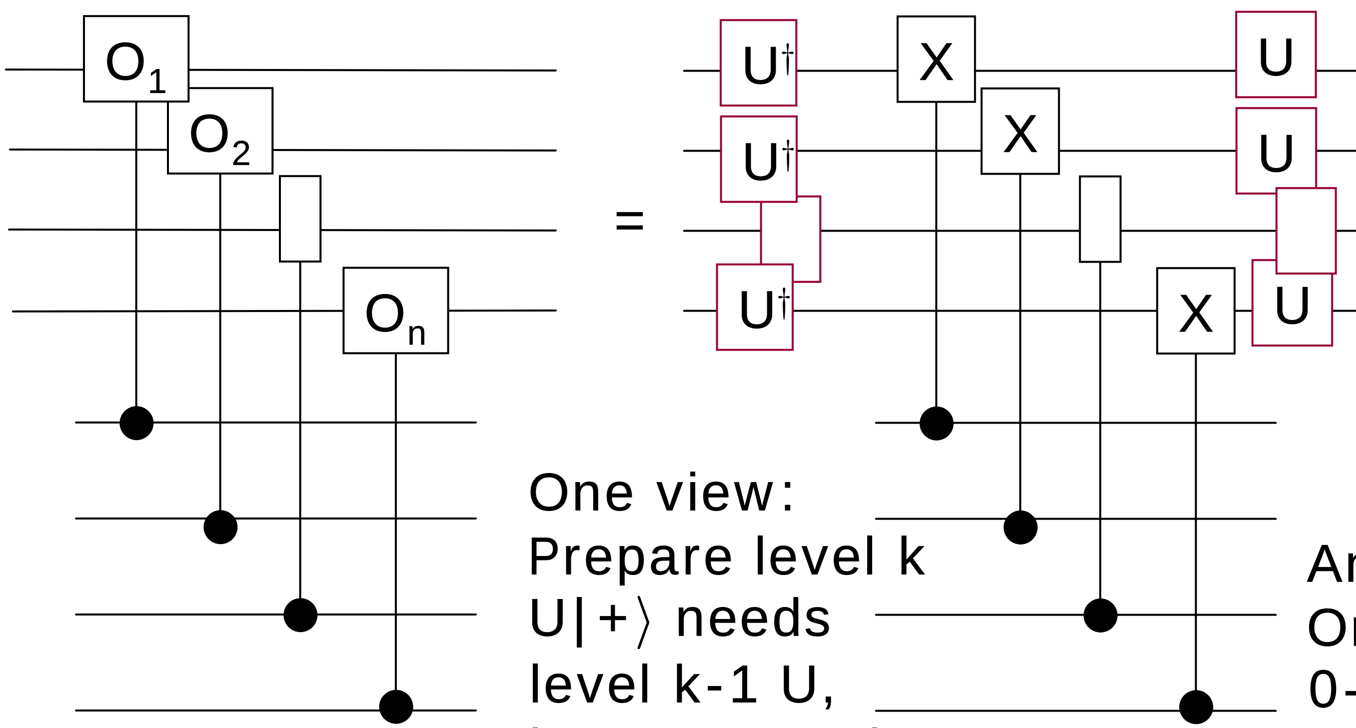
NOW THIS PART. Note $UXU^\dagger$ refers to the encoded version.

Preparing encoded $U|+\rangle$:

$\boxed{U\ X\ U^\dagger\ =\ e^{iZ\pi/4}\ X}$ unencoded

Need to measure stabilizers of the code and the [encoded $UXU^\dagger$] = [encoded $e^{iZ\pi/4}$ X] = $(e^{iZ\pi/4}\ X)^{\otimes 7}$ = $(UXU^\dagger)^{\otimes 7}$

Each O_i = $U\ X\ U^\dagger$



Still has to apply U but unencoded

Note also things are transversal

One view:
Prepare level k $U|+\rangle$ needs level k-1 U, in turns needs level k-1 $U|+\rangle$

Another view:
Only cares 0-Rec $\rightarrow$ 1-Rec
The rest recursive

Requirements for FT (or assumptions behind existence of a threshold):

1. Subthreshold error rates
2. Benign error scaling
3. Massively parallel operations
4. Method to remain in or return to codespace
5. Ability to bring in fresh ancillas DURING computation
6. Ability to measurement DURING computation

Nice to have, but surprisingly not necesary:

1. Non-nearest-neighnor coupling
2. Same code at each level
3. Fast measurement
4. Reliable and fast classical computation