3 - The Shor Code

↳ combination of both 3-qubit bit-flip and phase-flip QEC codes.

⇒ 9-qubit QEC code

→ can in fact be used to correct any arbitrary error on one qubit.

Again, consider we want to protect a general qubit state $|\psi\rangle = a|0\rangle + b|1\rangle$.

* Encoding: First encode the qubit using the phase-flip encoding:

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

Then, encode each of these 3 qubits with the bit-flip encoding:

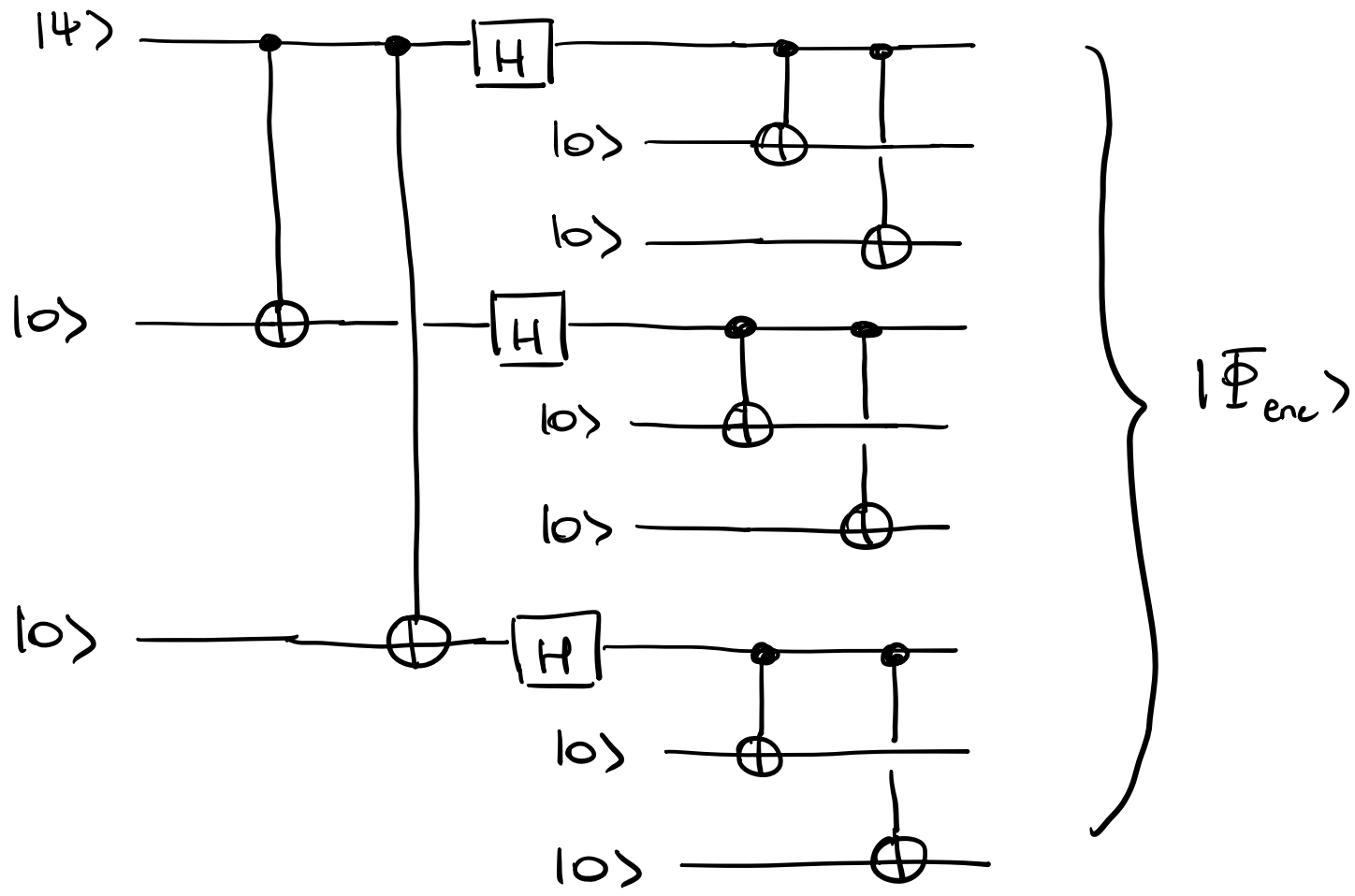
i.e. each $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \rightarrow \frac{1}{\sqrt{2}}(|000\rangle + |111\rangle)$

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

The result is a 9-qubit code which takes

$$\begin{cases} |0\rangle \rightarrow |0_L\rangle \equiv \frac{1}{2^{3/2}} \left(|000\rangle + |111\rangle\right)^{\otimes 3} \\ |1\rangle \rightarrow |1_L\rangle \equiv \frac{1}{2^{3/2}} \left(|000\rangle - |111\rangle\right)^{\otimes 3} \end{cases}$$

circuit to encode arbitrary $|\psi\rangle$:



phase-flip encoding

bit-flip encoding (for each of the three qubits)

→ method of encoding a hierarchy a levels is called "concatenation".

$|\Phi\rangle_{enc}$ is a 9-qubit entangled state:

$$|\Phi_{enc}\rangle = a\,|0_L\rangle + b\,|1_L\rangle$$

We can easily see that the Shor code is able to protect $|\psi\rangle$ against phase-flip and bit-flip errors –

* For ex suppose a bit-flip occurs on the left qubit of $|\Phi_{enc}\rangle$ ⟶ resulting state:
  of left register

$$\frac{a}{2^{3/2}}\left(|100\rangle + |011\rangle\right) \otimes \left(|000\rangle + |111\rangle\right)^{\otimes 2}$$

$$+\frac{b}{2^{3/2}}\left(|100\rangle - |011\rangle\right) \otimes \left(|000\rangle - |111\rangle\right)^{\otimes 2}$$

This can be detected (and recovered from) by parity-check measurement as in the 3-qubit bit-flip code:

Measurement of $\hat{Z}_8\,\hat{Z}_7$ ⇒ will conclude that they are ≠

- $\hat{Z}_8$ acts on left qubit
- $\hat{Z}_7$ acts on second to left qubit

——— of $\hat{Z}_7\,\hat{Z}_6$ ⇒ they are the same.

⇒ the first qubit must have flipped

⇒ recover by flipping again with X_8.

Can do the same to correct for bit flip on any of the 9 qubits.

* Similarly we can detect and correct for phase flips:

Suppose for instance that a phase-flip has occured on the left qubit of $|\Phi_{enc}\rangle$ -

New state:

$$= \frac{a}{2^{3/2}}\left(|000\rangle \ominus |111\rangle\right)\otimes\left(|000\rangle + |111\rangle\right)\otimes\left(|000\rangle + |111\rangle\right)$$

$$+ \frac{b}{2^{3/2}}\left(|000\rangle \oplus |111\rangle\right)\otimes\left(|000\rangle - |111\rangle\right)\otimes\left(|000\rangle - |111\rangle\right)$$

⇒ the relative sign in the first block of qubit is changed.

we will have the same state if any of the 3 qubits in the 1st block

undergoes phase flip.

=> The goal is to determine which of the 3 blocks has undergone a change of relative sign.

This can be done by comparing the relative signs of the first 2 blocks with:

$$X_8 X_7 X_6 \quad X_5 X_4 X_3$$

This op has eigenvalues ± 1 with

$+1$ corresponding to the case when the 2 blocks have same sgn

-1 corresponding to the case when the 2 blocks have opposite sgn.

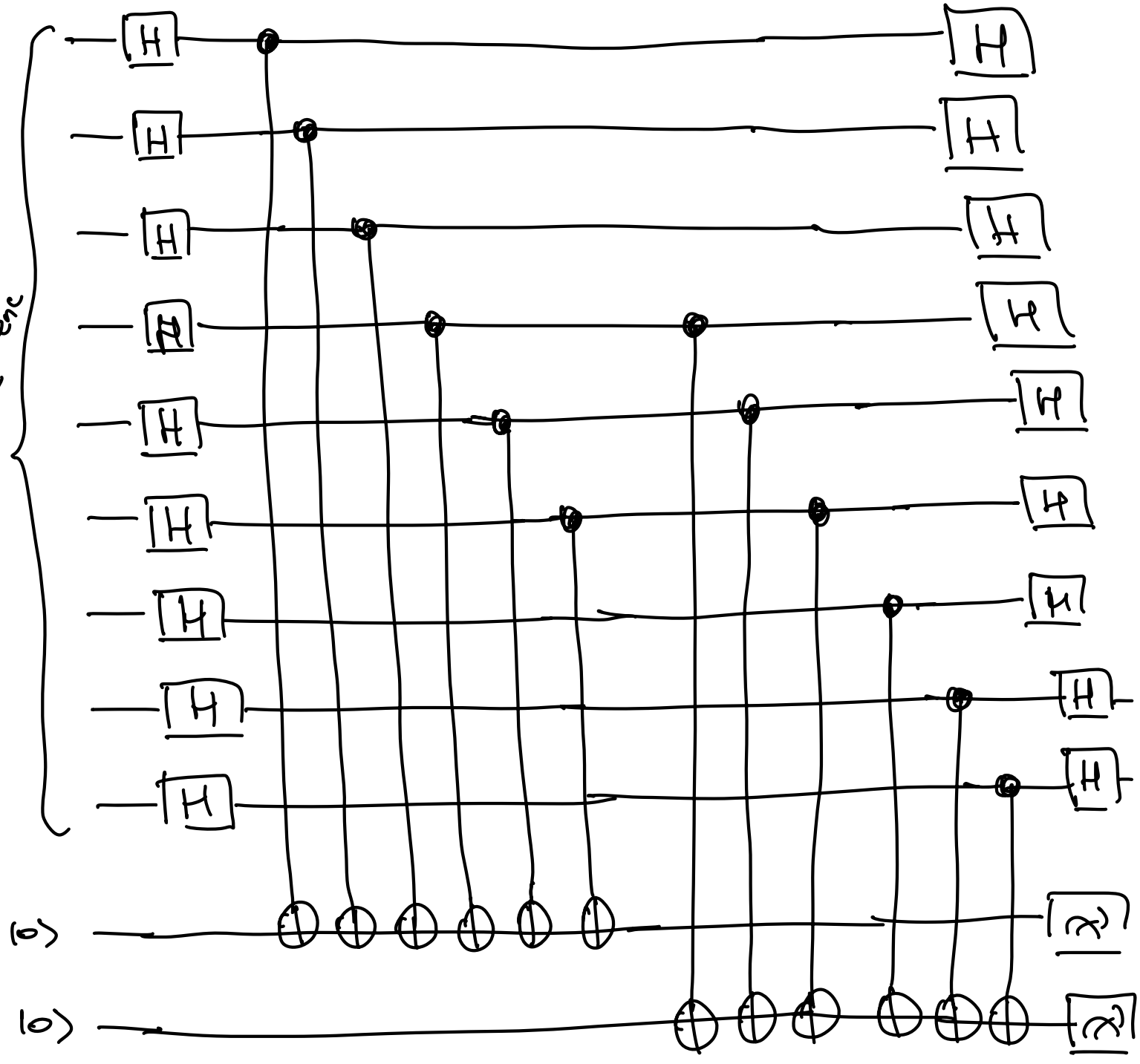
Then compare the relative sign of the two right groups of qubits.

→ will find they are the same

=> phase flip must have occurred on one of qubit in the left group.

=> recover with $Z_8 Z_7 Z_6$ to change the relative sgn in the left group.

⇒ possible circuit for phase flip detection

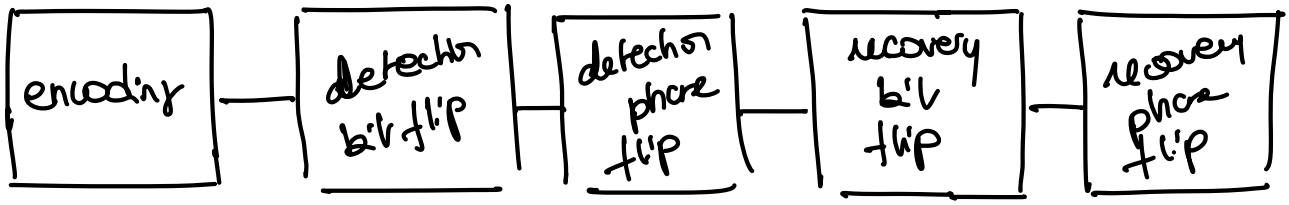


Since again $X = HZH$

Then, depending on the outcome

⇒ apply ZZZ to the correct block.

The full circuit would be something like



Have 6 ZZ operators for bit flip detection:

$Z_8 Z_7$, $Z_7 Z_6$, $Z_5 Z_4$, $Z_4 Z_3$, $Z_2 Z_1$, $Z_1 Z_0$

($Z_8 Z_7$, $Z_7 Z_6$: block 2; $Z_5 Z_4$, $Z_4 Z_3$: block 1; $Z_2 Z_1$, $Z_1 Z_0$: block 0)

+ 2 XXXXXX for phase flip detection.

Each need one ancilla qubit $\Rightarrow$ 8

+ 8 for the encoding

$\Rightarrow$ Shor code requires apriori 16 ancilla qubits.

But the circuit can be further optimized.

We can see that if both bit flip and phase flip errors occur on the same qubit, the Shor code will correct for it, since both X_8 and Z_8 are applied.

=> The Shor code can correct for bit-flip, phase-flip and combined bit-phase-flip errors on a single qubit.

Consider for ex the depolarizing qubit channel:

proba $\begin{cases} p/3 \text{ of bit-flip } (X), \\ p/3 \text{ of phase-flip } (Z) \\ p/3 \text{ of both } (Y \propto XZ) \end{cases}$

it can be shown that the Shor code suppresses the proba of error p to $O(p^2)$.

In fact it can be shown that the Shor code can protect against any arbitrary error on a qubit.

Correcting arbitrary single-qubit error with the Shor code:

Suppose the left qubit of left register encounters an error which sends:

$$|0\rangle \mapsto \alpha|0\rangle + \beta|1\rangle$$
$$|1\rangle \mapsto \gamma|0\rangle + \delta|1\rangle$$

The encoded state

$$|\Phi_{enc}\rangle = \frac{a}{2^{3/2}}\left(|000\rangle + |111\rangle\right)^{\otimes 3} + \frac{b}{2^{3/2}}\left(|000\rangle - |111\rangle\right)^{\otimes 3}$$

becomes

$$|\tilde{\Phi}_{enc}\rangle =$$

$$\frac{a}{2^{3/2}}\left[(\alpha|0\rangle+\beta|1\rangle)|00\rangle + (\gamma|0\rangle+\delta|1\rangle)|11\rangle\right]\otimes\left[|000\rangle+|111\rangle\right]^{\otimes 2}$$

$$+\frac{b}{2^{3/2}}\left[(\alpha|0\rangle+\beta|1\rangle)|00\rangle - (\gamma|0\rangle+\delta|1\rangle)|11\rangle\right]\otimes\left[|000\rangle-|111\rangle\right]^{\otimes 2}$$

New denote

$$\begin{cases} k+m = \alpha \\ k-m = \delta \\ l+n = \beta \\ l-n = \gamma. \end{cases}$$

we obtain:

$$|\tilde{\Phi}\rangle_{enc} =$$

$$\frac{k}{2^{312}}\Big\{ a\left(|000\rangle + |111\rangle\right) \otimes \left[|000\rangle+|111\rangle\right]^{\otimes 2}$$

$$+ b\left(|000\rangle - |111\rangle\right) \otimes \left[|000\rangle-|111\rangle\right]^{\otimes 2}\Big\}$$

$\otimes |100\rangle$ (after cnot's)

$$+\frac{l}{2^{312}}\Big\{ a\left(|100\rangle + |011\rangle\right) \otimes [\ +\ ]^{\otimes 2}$$

$$+ b\left(|100\rangle - |011\rangle\right) \otimes [\ -\ ]^{\otimes 2}\Big\}$$

$\otimes |10\rangle$ (after)

$$+\frac{m}{2^{312}}\Big\{ a\left(|000\rangle - |111\rangle\right) \otimes [\ +\ ]^{\otimes 2}$$

$$+ b\left(|000\rangle + |111\rangle\right) \otimes [\ -\ ]^{\otimes 2}\Big\}$$

$\otimes |00\rangle$

$$+\frac{n}{2^{312}}\Big\{ a\left(|100\rangle - |011\rangle\right) \otimes [\ +\ ]^{\otimes 2}$$

$$+ b\left(|100\rangle + |011\rangle\right) \otimes [\ -\ ]^{\otimes 2}\Big\}$$

$\otimes |10\rangle$

The bit-flip detection is done with (Z_8Z_7, Z_7Z_6)
2 ancillas
and CNOT gates
$\Rightarrow$ resulting ancilla states are above (green)

After measuring ancillae:

* the bit-flip detection collapses the state - for instance if outcome is 10:

$\Rightarrow$ 9-q state collapses to (unnormalized)

$$\frac{1}{2^{312}} \Big\{ a\,(|100\rangle + |011\rangle) \otimes [\quad + \quad]^{\otimes 2} + b\,(|100\rangle - |011\rangle) \otimes [\quad - \quad]^{\otimes 2} \Big\}$$

$$+ \frac{1}{2^{312}} \Big\{ a\,(|100\rangle - |011\rangle) \otimes [\quad + \quad]^{\otimes 2} + b\,(|100\rangle + |011\rangle) \otimes [\quad - \quad]^{\otimes 2} \Big\}$$

(if outcome was 00 $\rightarrow$ collapses to the other combination.)

$\Rightarrow$ collapse to the state which carries the bit-flip error (or no error if there was none)

* After that we do a phase flip detection:

$\rightarrow$ $X_1 X_7 X_6$ $X_5 X_4 X_3$ to compare the phases of the left & middle blocks (one ancilla, CNOT's + H's)

$\Rightarrow$ after the check:

$$\frac{l}{2^{312}}\left\{ a\left(|100\rangle + |011\rangle\right) \otimes \left[\ \overline{\overline{+}}\ \right]^{\otimes 2} \right.$$

$$\left. + b\left(|100\rangle - |011\rangle\right) \otimes \left[\ \overline{\overline{-}}\ \right]^{\otimes 2} \right\} \otimes |0\rangle$$

$$+ \frac{n}{2^{312}}\left\{ a\left(|100\rangle - |011\rangle\right) \otimes \left[\ \overline{\overline{+}}\ \right]^{\otimes 2} \right.$$

$$\left. + b\left(|100\rangle + |011\rangle\right) \otimes \left[\ \overline{\overline{-}}\ \right]^{\otimes 2} \right\} \otimes |1\rangle$$

if we get 0 when measure the ancilla

$\Rightarrow$ 9-q state collapses to

$$\propto \frac{l}{2^{312}}\left\{ a\left(|100\rangle + |011\rangle\right) \otimes \left[\ +\ \right]^{\otimes 2} \right.$$

$$\left. + b\left(|100\rangle - |011\rangle\right) \otimes \left[\ -\ \right]^{\otimes 2} \right\}$$

(state with no phase flip)

$\Rightarrow$ 6 can correct for the bit flip

if we get 1 $\Rightarrow$ state collapses to

$$\propto \frac{n}{2^{3/2}} \Big\{ a\left(|100\rangle - |011\rangle\right) \otimes \Big[\quad + \quad \Big]^{\otimes 2}$$

$$+ b\left(|100\rangle + |011\rangle\right) \otimes \Big[\quad - \quad \Big]^{\otimes 2} \Big\}$$

(state which carries the phase flip)

$\Rightarrow$ correct for bit & phase flip.

$\Rightarrow$ if an error occurs, the detection step always makes the system collapse to the encoded state carrying the error which we can then correct.

This property allows to correct for a continuum of errors.

Consider an arbitrary noise channel $\mathcal{E}$ acting on single-qubit state $|\psi\rangle$.

$$\mathcal{E}(|\psi\rangle\langle\psi|) = \sum_{\mu} M_{\mu} |\psi\rangle\langle\psi| M_{\mu}^{\dagger}$$

Since M_μ acts on one qubit, it can be expanded on $\{1, X, Y, Z\}$ basis:

$$M_\mu = \alpha_\mu 1 + \beta_\mu X + \gamma_\mu \underbrace{Y}_{iXZ} + \delta_\mu Z$$

$$\Rightarrow M_\mu |\psi\rangle = \alpha_\mu |\psi\rangle + \beta_\mu X|\psi\rangle + i\gamma_\mu XZ|\psi\rangle + \delta_\mu Z|\psi\rangle$$

Error detection (measurement) will collapse $M_\mu|\psi\rangle$ into one of the 4 states above ($|\psi\rangle$, $X|\psi\rangle$, $Z|\psi\rangle$, $XZ|\psi\rangle$)

Recovery can then be performed by applying the appropriate inverse operation, resulting in $|\psi\rangle$.

The same is true for all M_μ's.

$\Rightarrow$ Thus, by correcting only a discrete (finite) set of errors (bit-flip, phase-flip and combined bit-phase flip), a QEC code is able to automatically correct a continuum of errors.

So far, we have considered that errors only occur on one qubit.

There are ≠ ways to deal with the fact that noise can affect more than one:

- in many situation, it is a good approx to assume that noise acts independently on qubits. Then if the noise is "small", one can expand the total noise effect as sum: no qubit errors, + one-qubit errors + 2-qubit errors .... where the fist two terms dominate.

- if the noise doesn't act independently on the qubits, one has to develop QEC codes that can correct errors on more than one qubit.

They are based on similar ideas as the Shor code:

→ encoding

→ error detection (syndrome diagnosis)

→ recovery

conditions for QEC codes include:

subspaces spanned by the projectors applied in the error detection step must:

* be orthogonal (otherwise cannot reliably distinguish errors).

* correspond to "undeformed" versions of the original code space: the error must take orthogonal codewords (eg $|0_L\rangle, |1_L\rangle$) to orthogonal states (to be able to recover from the error)

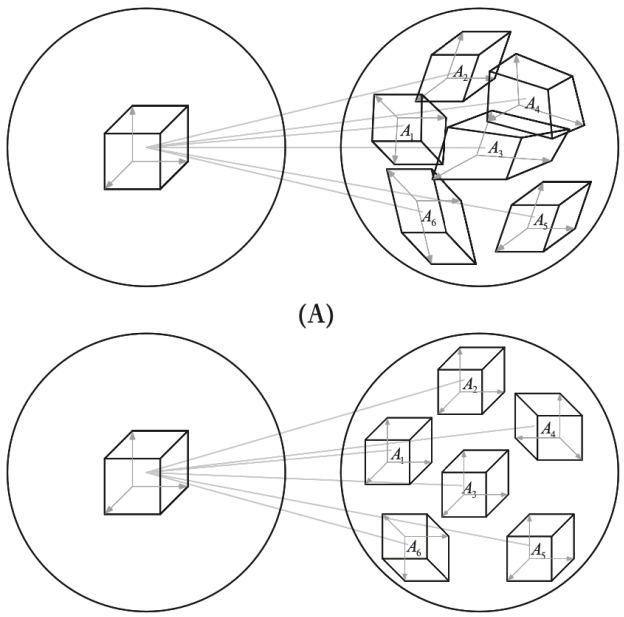


Figure 10.5. The packing of Hilbert spaces in quantum coding: (A) bad code, with non-orthogonal, deformed resultant spaces, and (B) good code, with orthogonal (distinguishable), undeformed spaces.

[From book by Nielsen & Chuang].

(2) Fault-tolerant Quantum Computing

In the previous section we have discussed the general principles of quantum-error correcting codes which aim at correcting the effect of noise due to the int° of the system with the envt.

Such codes have been designed to deal with a setting where we can encode our quantum information perfectly, then subject the qubits in the code to noise, and finally apply the QEC procedure perfectly.

→ thus we only assumed that errors occur during the transmiss°n of info

This is clearly not adequate if we are talking about a quantum computat°n which also requires to dynamically act on the quantum states with gates.

In fact we can have problems at every step of the computat°n:

1) state preparation : we want to initialize the quantum state to some fixed state (say $|0\rangle$), but such preparation may already fail.

2) when acting on qubits with gates :

These gates are imperfect (faulty), meaning that when we try to implement some operation U, we actually implement another operation $\tilde{U}$ which only approx U up to a certain accuracy.

Each gate then yields a small error on the output state, and these errors can accumulate and result in the failure of the computation.

This also has consequences on the QEC procedure itself → which itself can be imperfect.

Thus it is also necessary to protect the quantum info as it dynamically undergoes computation.

3) measurements :

Measurements can also be affected by errors and yield wrong results.

4) during the simple transmission of info along the quantum wires :
as we have discussed previously

=> How can we deal with all these sources of errors?

What we can do is to apply QEC procedures periodically (say after each gate, each measurement, state prep ...) to prevent the error from accumulating, but this is not so straight forward for 2 main reasons:

a) QEC uses encoded states → how do we apply the gates?

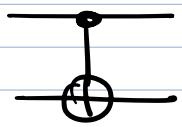
we cannot decode, apply the gates on physical qubits, and re-encode because the info would not be protected btw

decoding and re-encoding -
Thus if an error occured it would not be detected/corrected.

=> need to be able to compute directly on encoded states such that decoding is never required during the computation.

b) most importantly: the gates are not only able to cause error, they also propagate the error.

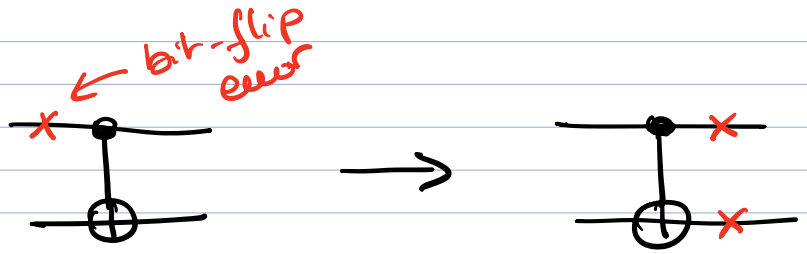
Consider for example the CNOT gate.



This gate acts on computational basis states as

$$\text{CNOT}\ |00\rangle = |00\rangle$$
$$\text{CNOT}\ |01\rangle = |11\rangle$$
$$\text{CNOT}\ |10\rangle = |10\rangle$$
$$\text{CNOT}\ |11\rangle = |01\rangle$$

If a bit-flip error was to occur on the control qubit before the gate acts the error will propagate to the target qubit.



for ex instead of CNOT $|00\rangle = |00\rangle$ we will do CNOT $|01\rangle = |11\rangle$

Then the quantum-error correction procedure, which typically assumes that the error only occurs on one qubit, will fail.

=> The goal is to <u>prevent such error propagation</u> - This means: we want to ensure that a single error on the input state (before the gate) results in a single error on the output state.

Such procedures (which avoid error propagation) are called **fault-tolerant procedures**.

In general, for fault-tolerant QC:

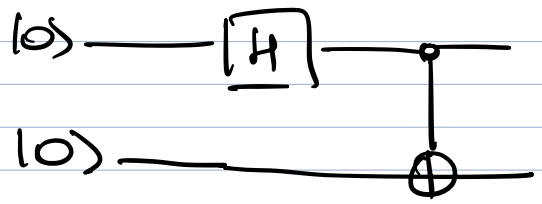
* we use encoded qubits, rather than physical qubits (using e.g. Steane code) and compute directly on these encoded qubits.

* we use fault-tolerant (FT) procedures: FT quantum gates, FT measurements, FT state preparations.

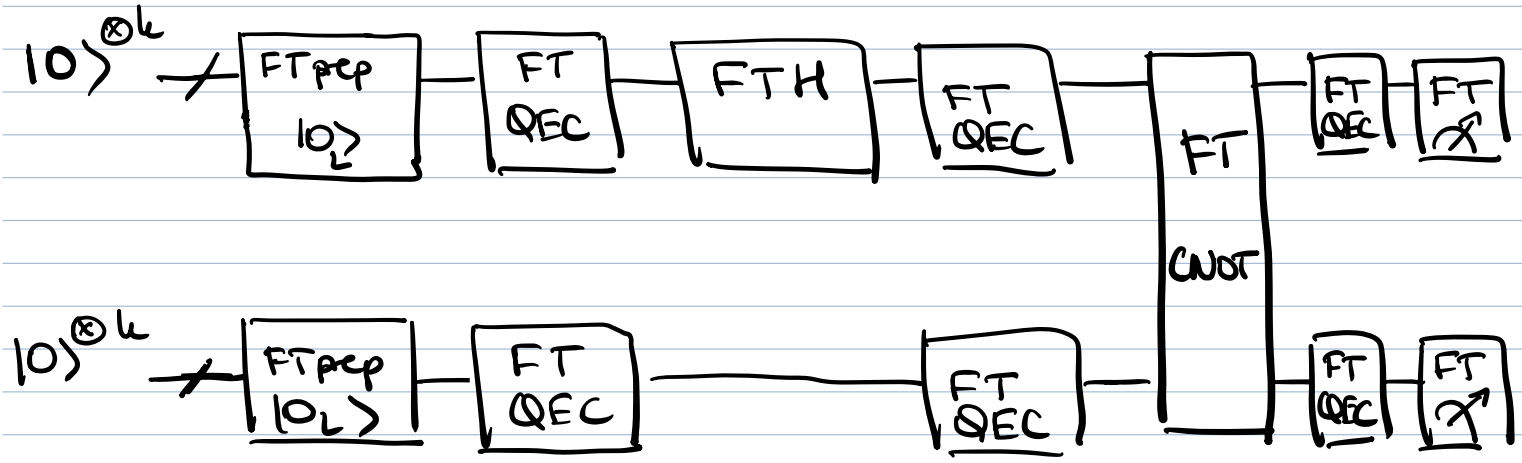
* periodically apply FT QEC to prevent accumulation of error.

Example:

Suppose we want to simulate:



The fault-tolerant version is (with a k-qubit encoding)

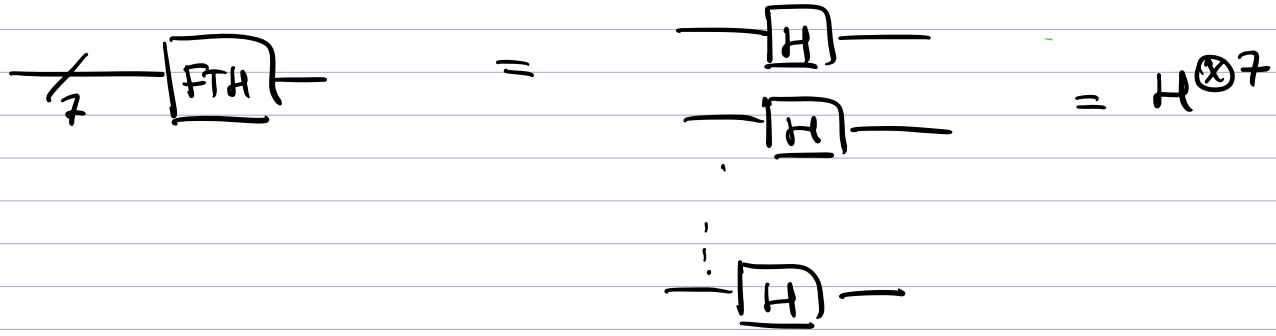


* more precise definitions :

→ FT gate : an encoded quantum gate is FT if a single error in the gate propagates to at most one error in each encoded block of qubits.

The 7-qubit encoding developped by A. Steane is often used because it allows to implement H, S and CNOT gates in a FT "transversal" way -

• Hadamard :

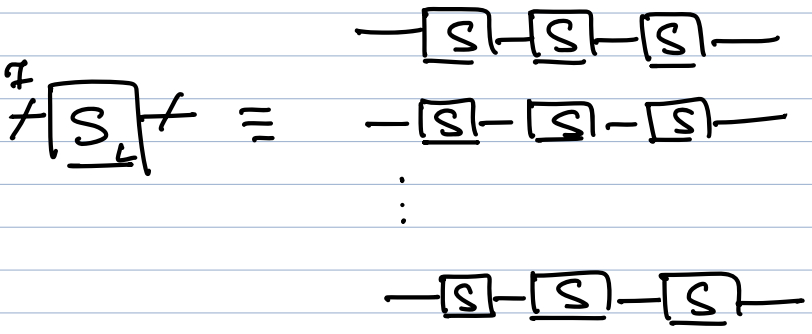


=> " transversal " implementation.

This is clearly fault-tolerant since each compo acts on one single qubit.
↳ (physical H)

If an error occurs on one qubit it does not affect the others

- S gate : apply S 3 times to each of the 7 qubits



can check that $S_L|0_L\rangle = |0_L\rangle$

$S_L|1_L\rangle = i|1_L\rangle$

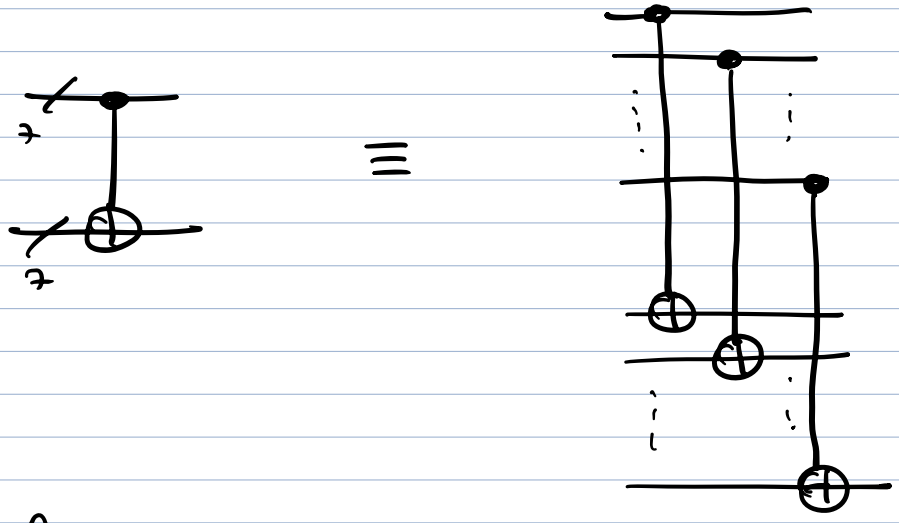
with the Steane encoding.

This is again clearly FT because acts on phys. qubits individually.

Other possibility: apply $-Z-S-$ to each qubit.

- **CNOT** :

→ can also be applied "transversally"



if have

one in each block → OK

Note: would also need FT T gate (or other non-stabilizer gate) to make an approx. universal gate set $\{H, T, (S), CNOT\}$ -

Ideally we would like to also have a transversal implementation of T - However, in 2009, Eastin and Knill proved that this is not possible (with any encoding).

=> Non-stabil. gates like the T gates, which are required for a quantum advantage, are the most costly to implement in a FT way.

Theorem: A quantum circuit with n qubits, m stabilizer gates, and k single-q non-stabilizer gates can be classically simulated in $O(\text{poly}(n, m, 16^k))$ time.
(Aaronson, Gottesman (2004))

=> quantum advantage requires k to be at least linear in n (would then be exp. hard for class. comp).

→ FT measurements : a measurement on a set of encoded qubits is said to be FT if :

- the failure of any single compo in the procedure results in an error on at most one qubit in each encoded block at the output.

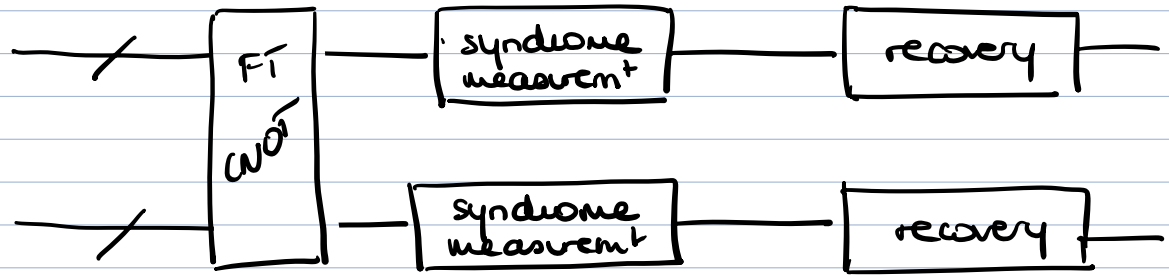
- and if one compo fails, the measurement result must have proba of error $O(p^2)$, where p = proba of failure of single compo.

→ FT state preparation A procedure for preparing a fixed encoded state is said to be FT if, given a single compo has failed during the proc, there is at most a single qubit of error in each encoded block.

Note : "component" means any "elementary operation (ie acting on physical qubits) (noisy gate, noisy measurement, noisy state prep, noisy wire)

Example :

Procedure for implementing a fault-tolerant CNOT gate, followed by a fault tolerant error-correction step.

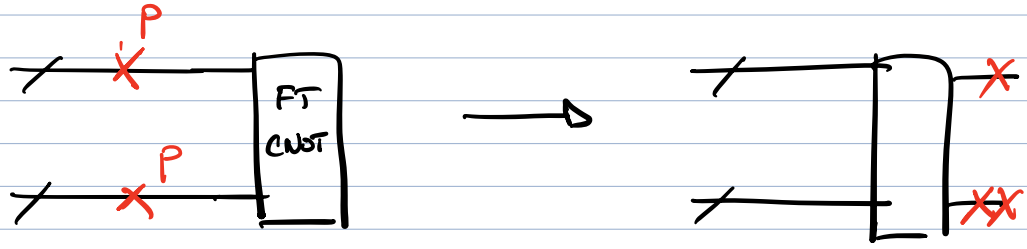


steps : ① ② ③ ④

The goal is to show that, if p is the proba of failure of each individual element, then the proba for 2 or more errors in each encoded block behaves as $O(p^2)$.

what are the ≠ ways of introducing 2 errors (or more) in one of the encoded blocks ? (say the 1st)

1) if there is 1 existing error entering the circuit at step ① in each block → proba $\propto p^2$



→ could lead to 2 errs in bottom block.

⇒ proba = $(c_0 p)^2$ where c_0 = # of ways error can occur before (in syndrome meas^t or recovery before)

2) 1 pre-existing error on one of the blocks (either top + bottom) + one failure of the CNOT itself (faulty).

⇒ proba $c_0 p \times c_1 p$.

↳ # of pairs of pts where can occur failure

$c_1 \sim 7$ for Steane

3) Two failures during the FT CNOT

4) One failure during CNOT
and one during measurement

$\Rightarrow$ every time have $p^2 \times$ proportio constant which counts the # of ways that this can happen.

$\Rightarrow$ in the end proba for 2 failures

$$= \underbrace{\left[c_0^2 + c_0 c_1 + c_2 + \dots c_6\right]}_{C} p^2$$

Typically it can be shown that $C \sim 10^4$ for typical QEC code.

If perfect decoding could be done at the end of the computation, then the proba of error would be Cp^2.

in summary: have found an implementation of CNOT with the property that indiv compo fail with proba p but the encoded procedure fails with proba cp^2.

$p < 1/c$

$\Rightarrow$ if p small enough, there is a net gain to use the encoded procedure: $cp^2 < p$.

here would need $p < 10^{-4}$

Concatenated Codes and the threshold thm.

In principle one can reduce the effective error rate achieved by the computation even further, by using concatenated codes.

The idea is to recursively apply the scheme described above for simulating a circuit using an encoded circuit.

→ construct a hierarchy of circuits

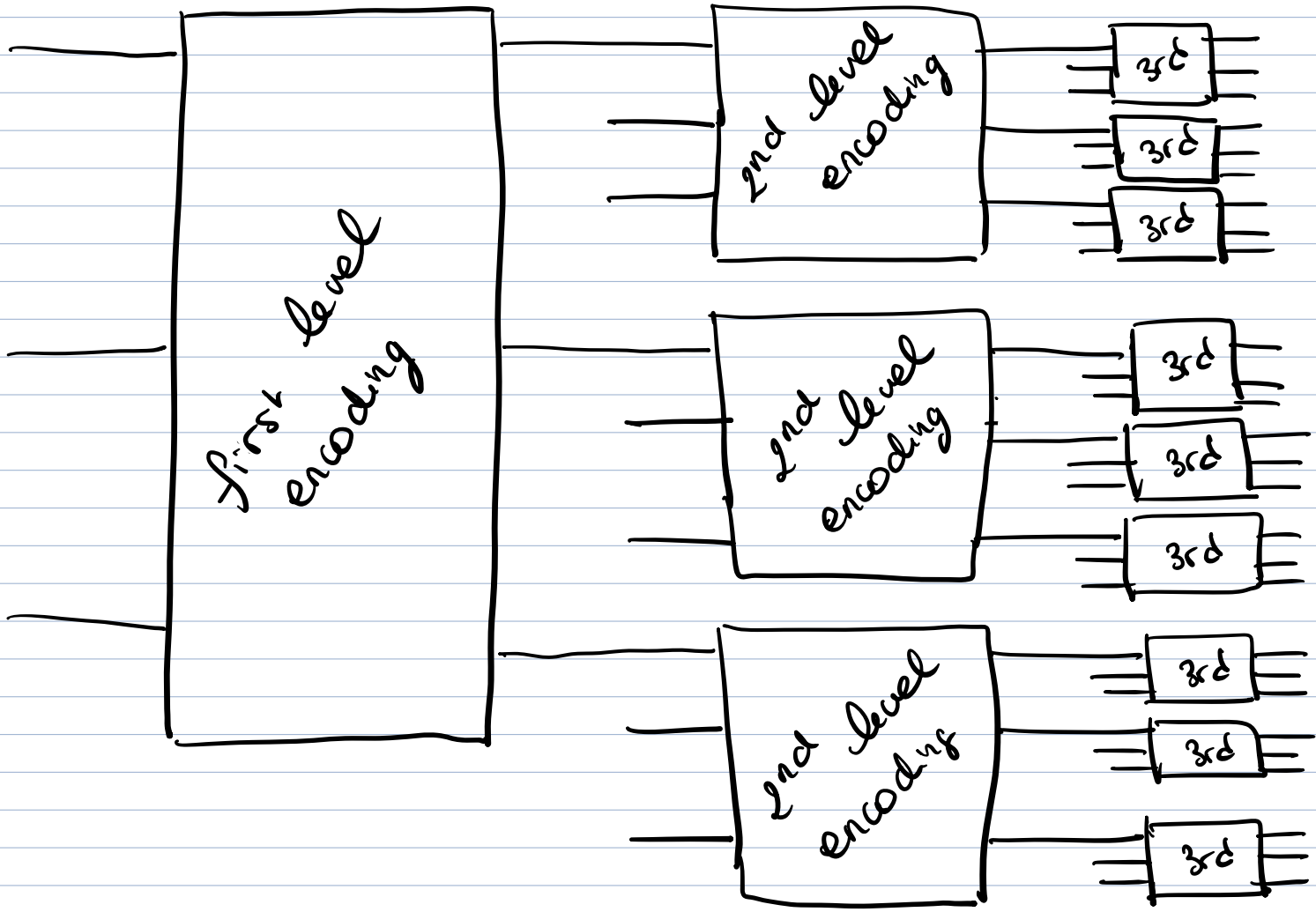
$C_0, C_1, C_2 \ldots$

↓

original circuit we wish to simulate

- In the first stage of the construction, each qubit in the original circuit is encoded in a quantum code whose qubits are themselves encoded in a quantum code and so on ...

→ Example of a 3-level concatenated code encoding one qubit into 27 qubits.
(Here use a 3-qubit code for the figure but in practice would use 9-qubit Shor code or 7-qubit Steane code).



- In the second stage, any gate in the original circuit $\mathcal{C}_0$ is replaced in $\mathcal{C}_1$ by a fault-tolerant procedure [encoded FT gate + QEC].

Each compo of $\mathcal{C}_1$ is then replaced in $\mathcal{C}_2$ by a fault-tolerant procedure implementing an encoded version of the component and QEC (at the next level).

and so on...

Suppose we have three levels of concatenat°
If the failure proba of components at the lowest level of the code (physical qubits) is p:

$\Rightarrow$ failure proba at the first level of encoding is cp^2 (at discussed above)

$\Rightarrow$ failure proba at the second level of encoding is $c(cp^2)^2 = c^3p^4$

$\Rightarrow$ third level : $c\left[c^3p^4\right]^2 = c^7p^8$

Thus if we have k levels of concatenation, the failure proba for a procedure at the highest level is $\frac{(cp)^{2^k}}{c}$

On the other hand, the size of the circuit goes as

$$d^k \times \text{size}(\mathcal{C}_0)$$

where d = max number of operations used in an FT procedure to do an encoded gate + QEC.

Now suppose we want to simulate a circuit $\mathcal{C}_0$ containing $\text{poly}(n) \equiv f(n)$ gates where n specifies the size of the pb. (e.g. circuit for factoring) -

And suppose we wish to achieve a final accuracy of ϵ.

$\Rightarrow$ our simulation of each gate must be accurate by $\frac{\epsilon}{f(n)}$

=> we must concatenate k times with k satisfying :

$$\frac{(cp)^{2^k}}{c} \leq \frac{\epsilon}{f(n)}$$

It can be shown that, provided is below some threshold : $p < p_{th} (= 1/c)$ such accuracy can be obtained with a circuit that is only polylogarithmically larger than the original circuit C_0.

=> Threshold thm for quantum computation :

A quantum circuit containing $f(n)$ gates may be simulated with proba of error ϵ using

$$O\left[\text{poly}\left(\log(f(n)/\epsilon)\right) f(n) \right]$$

gates on hardware whose components fail with proba at most p, provided that p is below some constant threshold $p < p_{th}$ (and given reasonable assumptions about the noise in the hardware)