Understanding Quantum Information and Computation

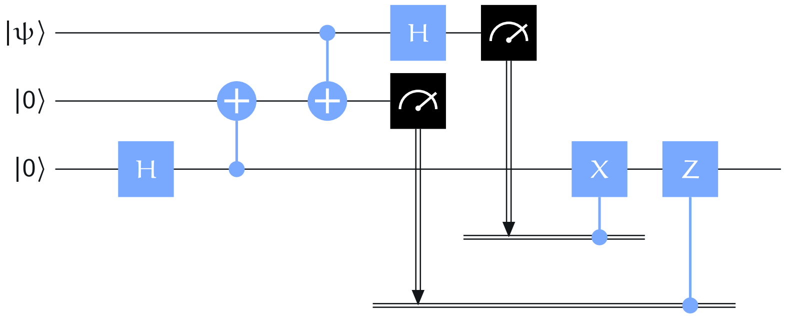
Lesson 16

Fault-Tolerant Quantum Computation

John Watrous

Model for fault tolerance

Consider a quantum circuit that we might hope to implement.



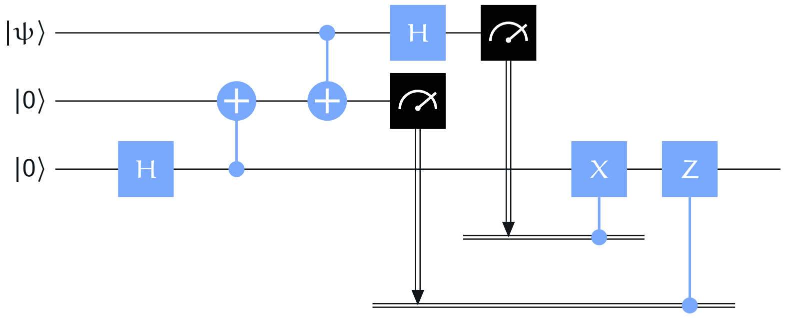
What could possibly go wrong?

- State initializations
- Unitary gates
- Measurements
- Qubit storage

We typically assume classical computations are perfect — but anything that involves quantum information could be faulty.

Model for fault tolerance

Consider a quantum circuit that we might hope to implement.



What could possibly go wrong?

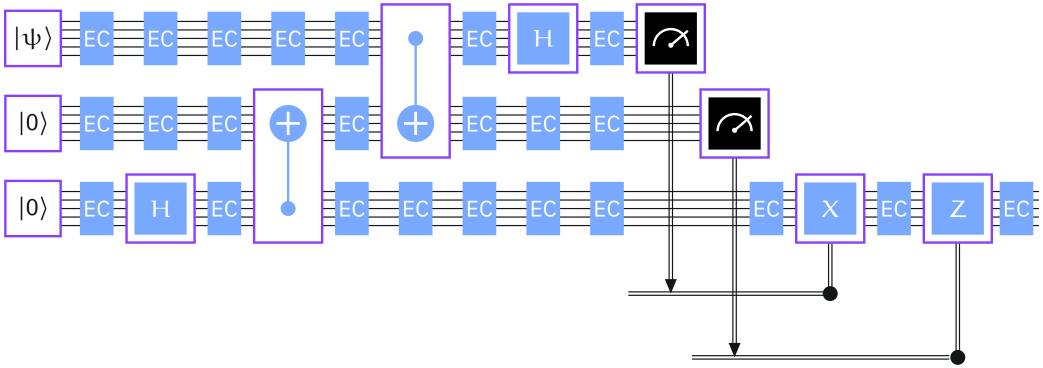
- State initializations
- Unitary gates
- Measurements
- Qubit storage

Independent stochastic noise model

Faults are assumed to be *uncorrelated* — occurring independently at eash possible location with a given probability.

Fault-tolerant implementations

A given *logical* quantum circuit might be implemented fault tolerantly as follows.



- State preparations, unitary gates, and measurement are performed by *gadgets.*
- Qubits are *encoded* using a quantum error correcting code.
- Encoded qubits are repeatedly *error corrected* throughout the computation.

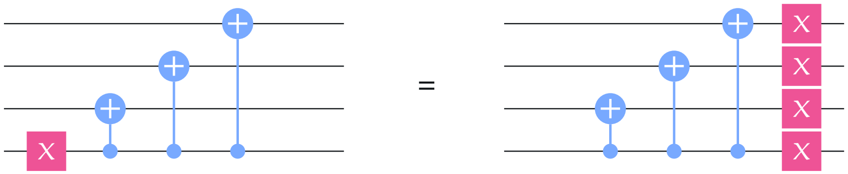
For a given noise model and choice of gadgets and code, we can ask a fundamental question: Are we making things better or worse?

Error propagation

Two-qubit gates can propagate errors (even when they're perfect).

This can create correlated errors on two qubits. (Two-qubit gates can also be faulty, causing correlated errors on multiple qubits.)

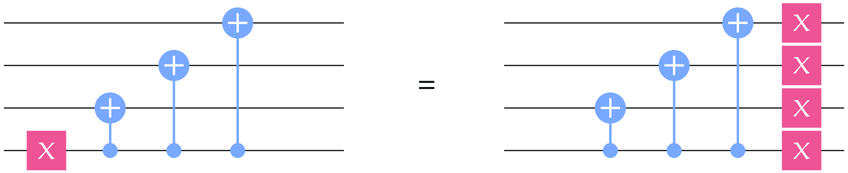
These errors can propagate further as we add additional two-qubit gates.



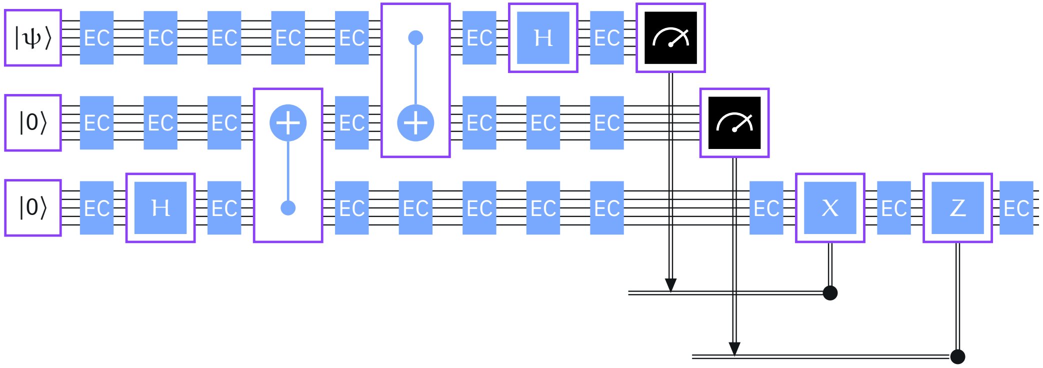
Error propagation

Two-qubit gates can propagate errors (even when they're perfect).

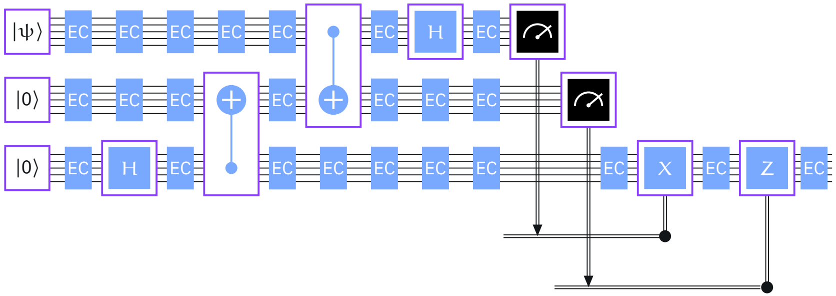
These errors can propagate further as we add additional two-qubit gates.



This must be kept in mind as we consider our gadgets and error correction procedure.



Transversal gates

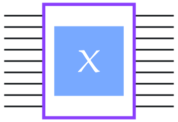


Some codes allow for *transversal* implementations of certain gates — meaning a tensor product of operations acting on a single qubit position within each code block.

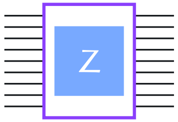
Example: transversal Pauli gates

Every stabilizer code allows for a transversal implementation of Pauli gates.

For the 3×3 surface code, for instance, we can implement X and Z as follows:

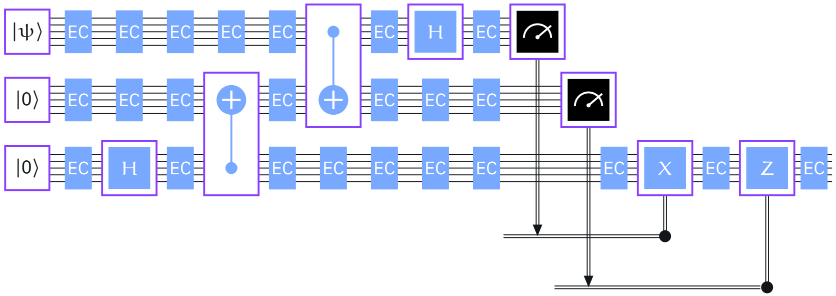


$$= \begin{matrix} X & \mathbb{1} & \mathbb{1} \\ X & \mathbb{1} & \mathbb{1} \\ X & \mathbb{1} & \mathbb{1} \end{matrix}$$



$$= \begin{matrix} Z & Z & Z \\ \mathbb{1} & \mathbb{1} & \mathbb{1} \\ \mathbb{1} & \mathbb{1} & \mathbb{1} \end{matrix}$$

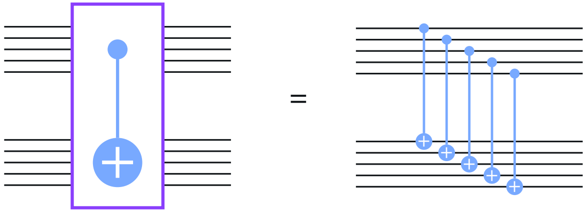
Transversal gates



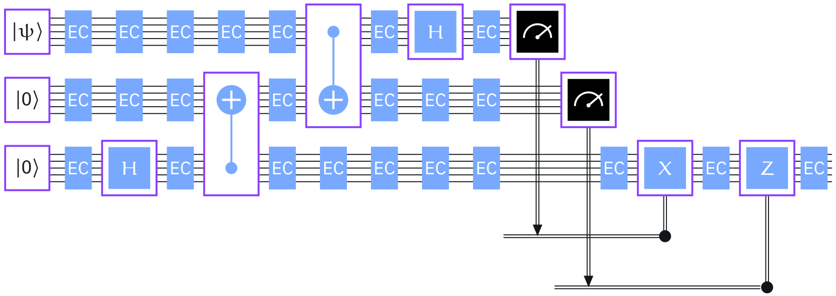
Some codes allow for *transversal* implementations of certain gates — meaning a tensor product of operations acting on a single qubit position within each code block.

Example: transversal CNOT

Every CSS code allows for a transversal implementation of CNOT gates.



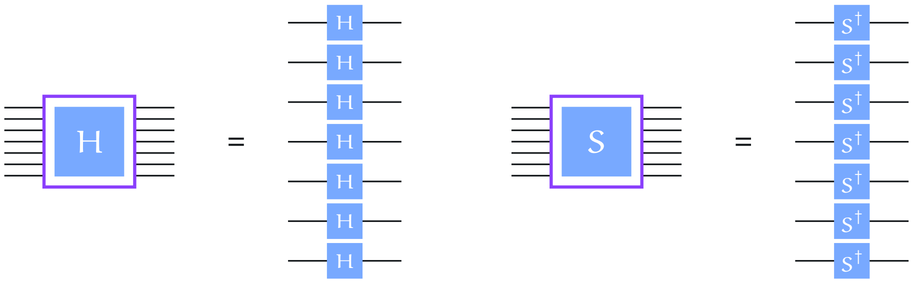
Transversal gates



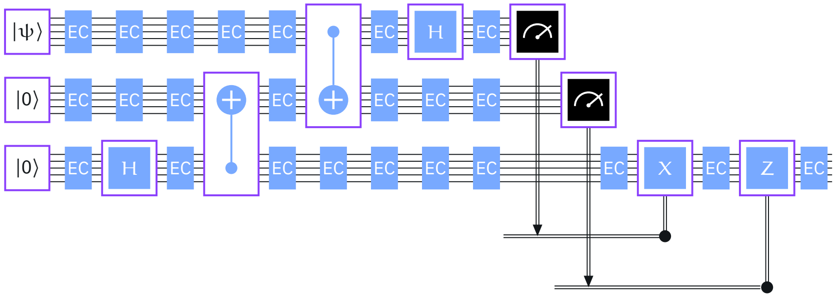
Some codes allow for *transversal* implementations of certain gates — meaning a tensor product of operations acting on a single qubit position within each code block.

Example: transversal Clifford gates for Steane code

The Steane 7-qubit code allows *all Clifford gates* to be implemented transversally.



Transversal gates



Some codes allow for *transversal* implementations of certain gates — meaning a tensor product of operations acting on a single qubit position within each code block.

Transversal gate gadgets are *inherently fault-tolerant* — they never propagate errors within a code block. (Subsequent error correction steps can correct induced errors.)

Eastin–Knill theorem

For any quantum error correcting code with distance at least 2, the set of logical gates that can be implemented transversally generates a discrete set of operations (and is therefore not universal).

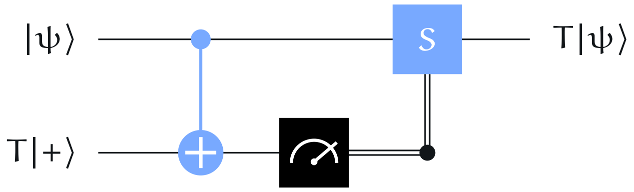
Magic states

$$S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/2} \end{pmatrix} \qquad T = \begin{pmatrix} 1 & 0 \\ 0 & \frac{1+i}{\sqrt{2}} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}$$

- S is a Clifford operation
- T is not a Clifford operation — $\{H, T, \text{CNOT}\}$ is universal for quantum computation

We cannot implement T gates using Clifford operations and standard basis measurements alone — but we can if we also have a copy of this *magic state:*

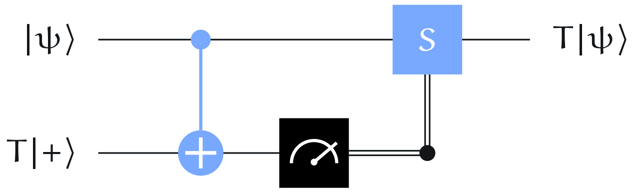
$$T|+\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + e^{i\pi/4}|1\rangle\right)$$



Magic states

We cannot implement T gates using Clifford operations and standard basis measurements alone — but we can if we also have a copy of this *magic state:*

$$T|+\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + e^{i\pi/4}|1\rangle\right)$$



$$T|+\rangle \otimes |\psi\rangle \xmapsto{\text{CNOT}} \frac{1}{\sqrt{2}}|0\rangle \otimes T|\psi\rangle + \frac{1+i}{2}|1\rangle \otimes T^{\dagger}|\psi\rangle$$

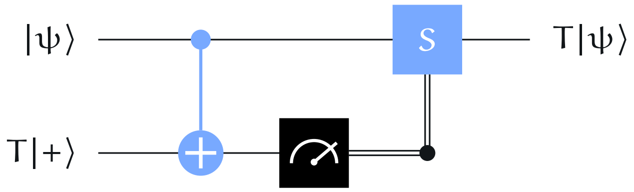
Measure 0: output $T|\psi\rangle$

Measure 1: output $ST^{\dagger}|\psi\rangle = T|\psi\rangle$

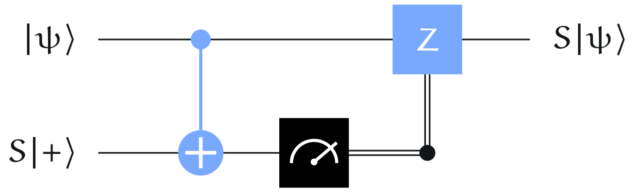
Magic states

We cannot implement T gates using Clifford operations and standard basis measurements alone — but we can if we also have a copy of this *magic state:*

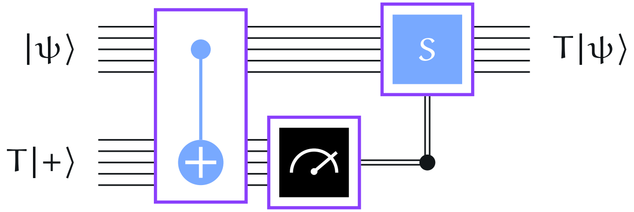
$$T|+\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + e^{i\pi/4}|1\rangle\right)$$



An S gate can be implemented similarly using a copy of the state $S|+\rangle = |+i\rangle$.



Magic states



This method allows for a fault-tolerant implementation of a T gate:

- The circuit is performed on *encoded qubits,* using fault-tolerant gadgets for the required gates.
- Requires an *encoded magic state.*

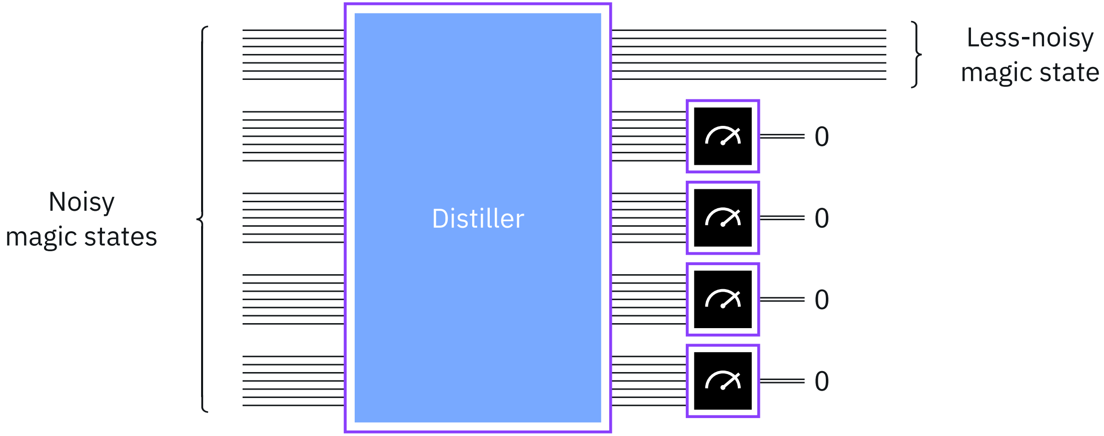
Key idea: magic state distillation

Encoded magic states can be prepared separately with a *probabilistic process* that need not succeed every time. (If it fails we simply try again.)

Magic states

Key idea: magic state distillation

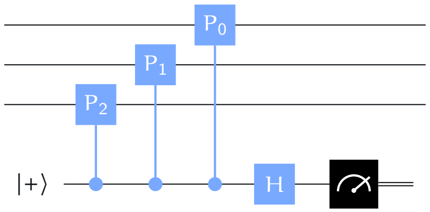
Encoded magic states can be prepared separately with a *probabilistic process* that need not succeed every time. (If it fails we simply try again.)



Other methods for implementing gates fault-tolerantly include *code deformation* and *code switching.*

Fault-tolerant error correction

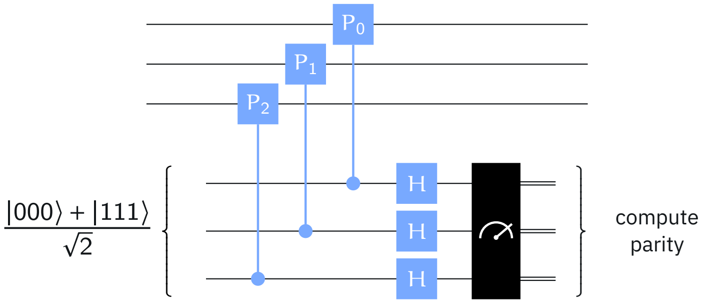
Straightforward implementations of syndome measurements are not fault-tolerant — they can cause errors to propagate within code blocks.



There are multiple known ways to address this problem.

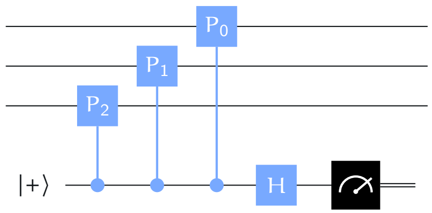
Shor error correction

Use *cat states* to measure syndomes.



Fault-tolerant error correction

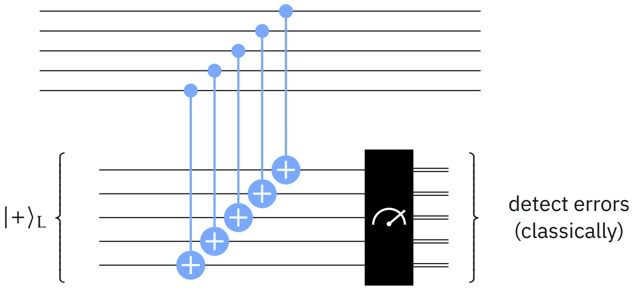
Straightforward implementations of syndome measurements are not fault-tolerant — they can cause errors to propagate within code blocks.



There are multiple known ways to address this problem.

Steane error correction (CSS codes only)

Use *encoded states* to intentionally propagate errors to workspace qubits.

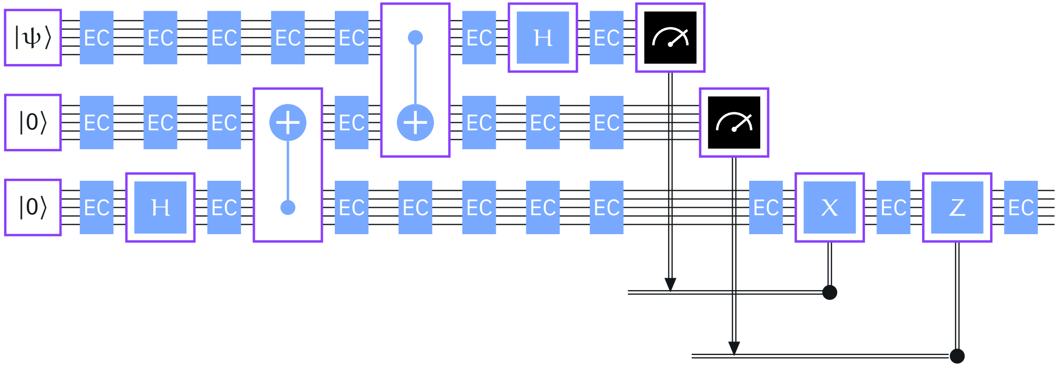


Threshold theorem

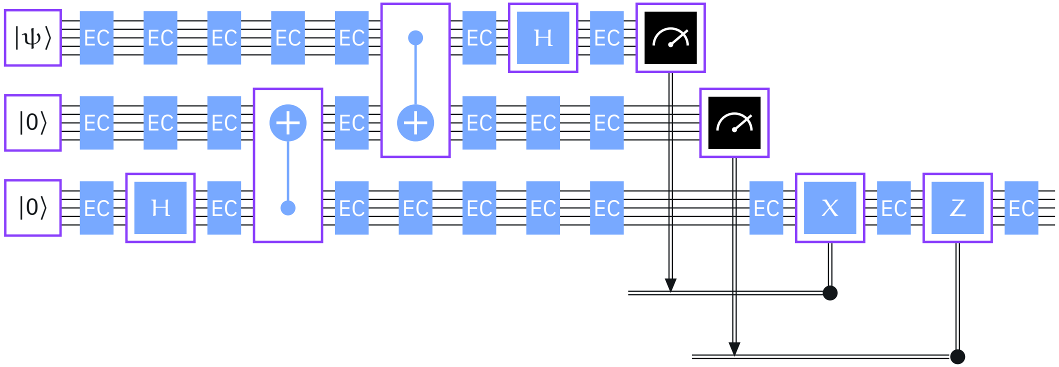
Threshold theorem (informal statement)

A quantum circuit having N gates can be implemented with high accuracy by a noisy quantum circuit, provided that the probability of error at each location in the noisy circuit is below a fixed, nonzero *threshold value* $p_{th} > 0$.

The size of the noisy circuit required scales as $O(N \log^c(N))$ for a positive constant c.



Threshold theorem



Suppose (for simplicity) that we use the 7-qubit Steane code, so our error corrections can correct for 1 error per code block.

The probability of error at each (logical) location in the original circuit is at most Cp^2 for some constant C (which depends on our gadgets).

If $p < 1/C = p_{th}$ this is a reduction in error — from p to $(Cp)p$.

Key idea

Concatenate: Think of our new (fault tolerant) circuit as a logical circuit, and implement it fault-tolerantly.

Threshold theorem

Suppose (for simplicity) that we use the 7-qubit Steane code, so our error corrections can correct for 1 error per code block.

The probability of error at each (logical) location in the original circuit is at most Cp^2 for some constant C (which depends on our gadgets).

If $p < 1/C = p_{\text{th}}$ this is a reduction in error — from p to $(Cp)p$.

Key idea

Concatenate: Think of our new (fault tolerant) circuit as a logical circuit, and implement it fault-tolerantly.

The logical error rate for the original circuit decreases rapidly with each concatenation.

$$
\begin{aligned}
p &\mapsto Cp^2 = (Cp)p \\
&\mapsto C((Cp)p)^2 = (Cp)^3 p \\
&\mapsto C((Cp)^3 p)^2 = (Cp)^7 p \\
&\mapsto \cdots \mapsto (Cp)^{2^k - 1} p
\end{aligned}
$$

Threshold theorem

Key idea

Concatenate: Think of our new (fault tolerant) circuit as a logical circuit, and implement it fault-tolerantly.

The logical error rate for the original circuit decreases rapidly with each concatenation.

$$
\begin{aligned}
p &\mapsto Cp^2 = (Cp)p \\
&\mapsto C((Cp)p)^2 = (Cp)^3 p \\
&\mapsto C((Cp)^3 p)^2 = (Cp)^7 p \\
&\mapsto \cdots \mapsto (Cp)^{2^k - 1} p
\end{aligned}
$$

Threshold theorem (informal statement)

A quantum circuit having N gates can be implemented with high accuracy by a noisy quantum circuit, provided that the probability of error at each location in the noisy circuit is below a fixed, nonzero *threshold value* $p_{\text{th}} > 0$.

The size of the noisy circuit required scales as $O(N \log^c(N))$ for a positive constant c.