Quantum Error Correction and Fault Tolerance

Motivation

The biggest barrier to building a quantum computer capable of running the algorithms we learned about is **noise**.

Motivation

The biggest barrier to building a quantum computer capable of running the algorithms we learned about is **noise**.

Noise comes from many sources:

1. Interactions between the device and its environment.
2. Imperfections in gate operations (over-rotation, timing errors, ...).
3. Unwanted interactions between qubits (cross-talk).

Motivation

The biggest barrier to building a quantum computer capable of running the algorithms we learned about is **noise**.

Noise comes from many sources:

1. Interactions between the device and its environment.
2. Imperfections in gate operations (over-rotation, timing errors, ...).
3. Unwanted interactions between qubits (cross-talk).

The circuits we have drawn so far assume that gates work perfectly.

Solution: redundancy

Classically, use a **repetition code**:

$$0 \longmapsto 000, \qquad 1 \longmapsto 111.$$

To decode, take the majority. This corrects one bit flip.

Solution: redundancy

Classically, use a **repetition code**:

$$0 \longmapsto 000, \qquad 1 \longmapsto 111.$$

To decode, take the majority. This corrects one bit flip.

What changes for quantum information?

1. We cannot copy an unknown quantum state.
2. Measuring the state can destroy the superposition.
3. There are infinitely many possible single-qubit errors.

Solution: redundancy

Classically, use a **repetition code**:

$$0 \longmapsto 000, \qquad 1 \longmapsto 111.$$

To decode, take the majority. This corrects one bit flip.

What changes for quantum information?

1. We cannot copy an unknown quantum state.
2. Measuring the state can destroy the superposition.
3. There are infinitely many possible single-qubit errors.

Can we still protect an unknown qubit?

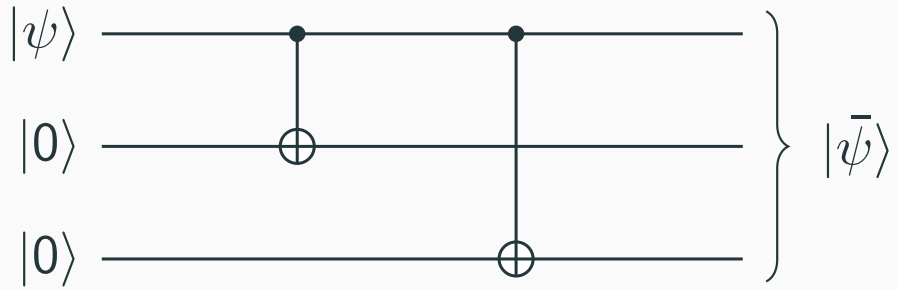
Correcting bit flip errors

Use the repetition code:

$$|0\rangle \longmapsto |000\rangle = |\bar{0}\rangle\,, \qquad |1\rangle \longmapsto |111\rangle = |\bar{1}\rangle\,.$$

By linearity,

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle \longmapsto \alpha\,|000\rangle + \beta\,|111\rangle = |\bar{\psi}\rangle\,.$$



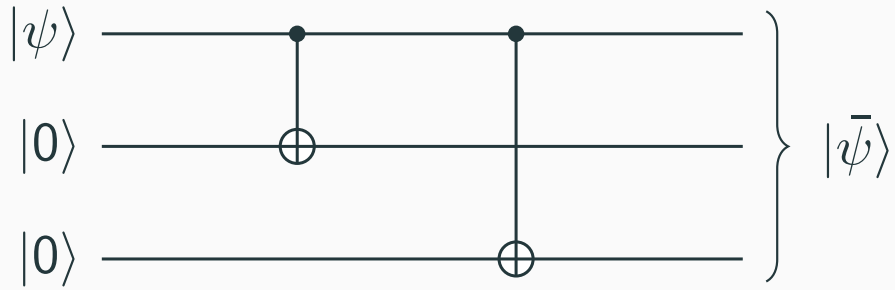
Correcting bit flip errors

Use the repetition code:

$$|0\rangle \longmapsto |000\rangle = |\bar{0}\rangle\,, \qquad |1\rangle \longmapsto |111\rangle = |\bar{1}\rangle\,.$$

By linearity,

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle \longmapsto \alpha\,|000\rangle + \beta\,|111\rangle = |\bar{\psi}\rangle\,.$$



This is *not* $|\psi\rangle \otimes |\psi\rangle \otimes |\psi\rangle$. We have encoded one **logical qubit** using three physical qubits.

Detecting and correcting bit flip errors

Suppose the second qubit flips:

$$\alpha\,|000\rangle + \beta\,|111\rangle \xrightarrow{X_2} \alpha\,|010\rangle + \beta\,|101\rangle\,.$$

Detecting and correcting bit flip errors

Suppose the second qubit flips:

$$\alpha\,|000\rangle + \beta\,|111\rangle \xrightarrow{X_2} \alpha\,|010\rangle + \beta\,|101\rangle\,.$$

Directly measuring the encoded qubits would destroy the superposition:

$$|010\rangle \text{ with probability } |\alpha|^2, \qquad |101\rangle \text{ with probability } |\beta|^2.$$

We would no longer have $\alpha\,|0\rangle + \beta\,|1\rangle$.

Detecting and correcting bit flip errors

Suppose the second qubit flips:

$$\alpha\,|000\rangle + \beta\,|111\rangle \xrightarrow{X_2} \alpha\,|010\rangle + \beta\,|101\rangle\,.$$

Directly measuring the encoded qubits would destroy the superposition:

$$|010\rangle \text{ with probability } |\alpha|^2, \qquad |101\rangle \text{ with probability } |\beta|^2.$$

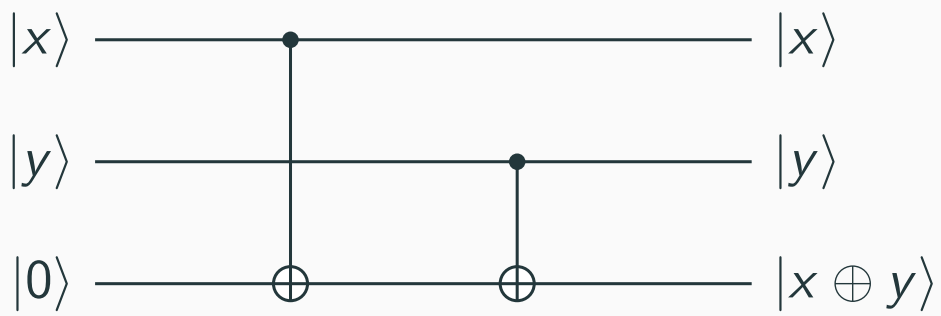
We would no longer have $\alpha\,|0\rangle + \beta\,|1\rangle$.

Instead, compute **error syndromes** and measure those.

We want to learn *which qubit flipped*, not the logical state.

Checking whether two bits agree

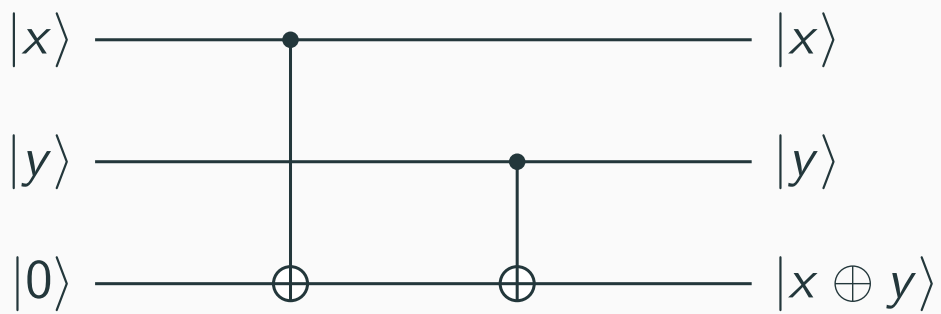
Compute the parity into an extra qubit:



$$x \oplus y = \begin{cases} 0 & \text{if } x = y, \\ 1 & \text{if } x \neq y. \end{cases}$$

Checking whether two bits agree

Compute the parity into an extra qubit:



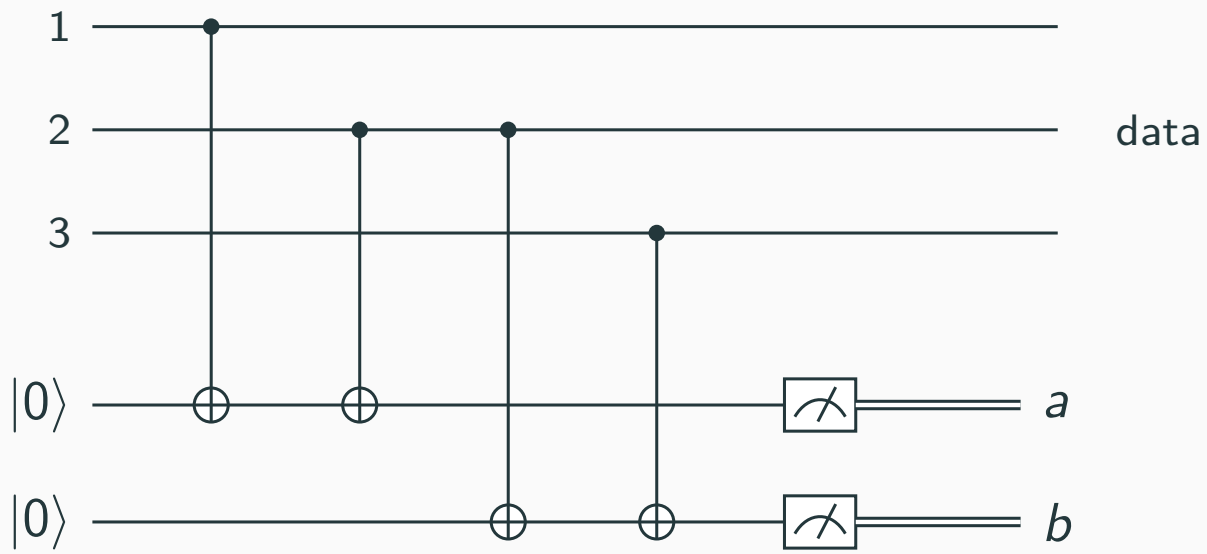
$$x \oplus y = \begin{cases} 0 & \text{if } x = y, \\ 1 & \text{if } x \neq y. \end{cases}$$

The outcome tells us whether the bits agree, not whether they are 0 or 1.

Both $|00\rangle$ and $|11\rangle$ give parity 0.

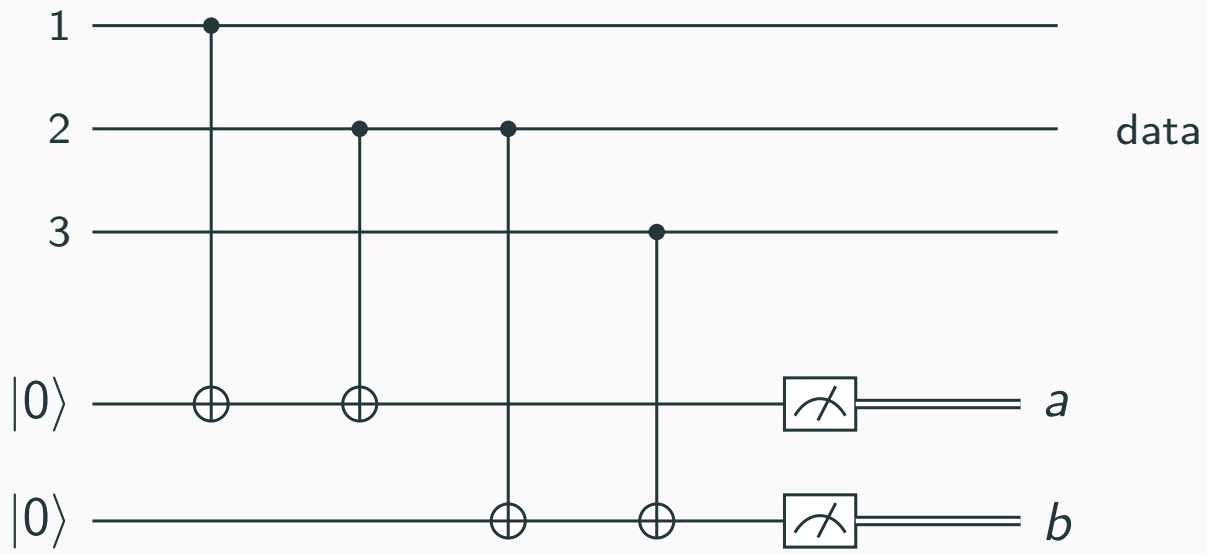
Measuring the syndrome

Compute $a = x_1 \oplus x_2$ and $b = x_2 \oplus x_3$.



Measuring the syndrome

Compute $a = x_1 \oplus x_2$ and $b = x_2 \oplus x_3$.

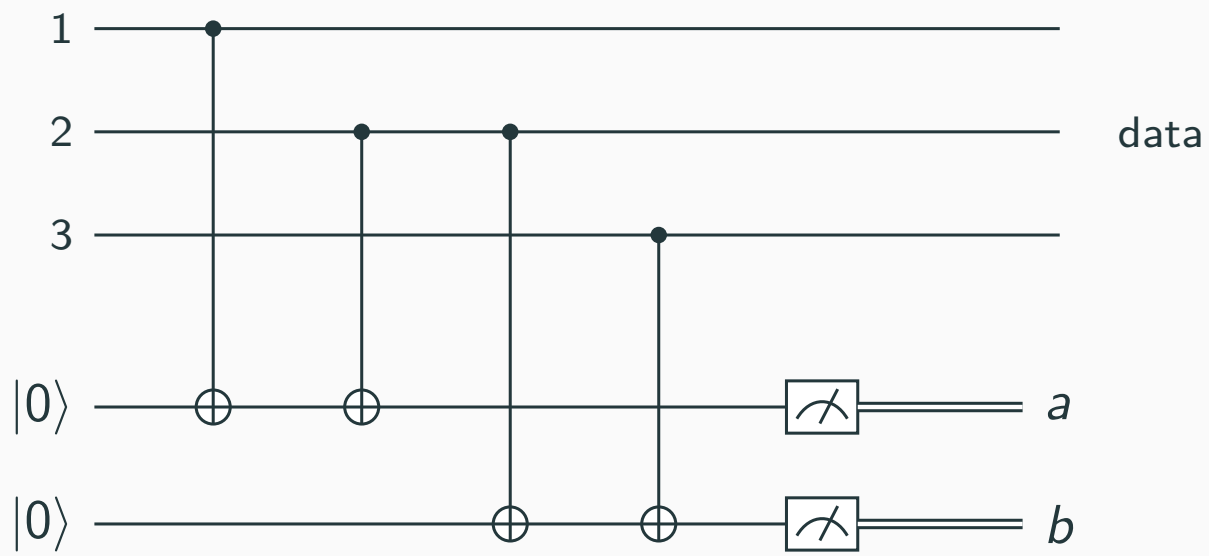


For the second-qubit error, before measuring the ancillas:

$$(\alpha\,|010\rangle + \beta\,|101\rangle)\,|00\rangle \longmapsto (\alpha\,|010\rangle + \beta\,|101\rangle)\,|11\rangle\,.$$

Measuring the syndrome

Compute $a = x_1 \oplus x_2$ and $b = x_2 \oplus x_3$.



For the second-qubit error, before measuring the ancillas:

$$(\alpha\,|010\rangle + \beta\,|101\rangle)\,|00\rangle \longmapsto (\alpha\,|010\rangle + \beta\,|101\rangle)\,|11\rangle\,.$$

The ancillas are unentangled with the data. Measuring them does not change $\alpha\,|010\rangle + \beta\,|101\rangle$.

Which qubit flipped?

Assume that **at most one** qubit has flipped.

$$a = x_1 \oplus x_2, \qquad b = x_2 \oplus x_3.$$

Error	a	b	Correction

Which qubit flipped?

Assume that **at most one** qubit has flipped.

$$a = x_1 \oplus x_2, \qquad b = x_2 \oplus x_3.$$

Error	a	b	Correction
None	0	0	Do nothing

Which qubit flipped?

Assume that **at most one** qubit has flipped.

$$a = x_1 \oplus x_2, \qquad b = x_2 \oplus x_3.$$

Error	a	b	Correction
None	0	0	Do nothing
First qubit	1	0	Apply X to qubit 1

Which qubit flipped?

Assume that **at most one** qubit has flipped.

$$a = x_1 \oplus x_2, \qquad b = x_2 \oplus x_3.$$

Error	a	b	Correction
None	0	0	Do nothing
First qubit	1	0	Apply X to qubit 1
Second qubit	1	1	Apply X to qubit 2

Which qubit flipped?

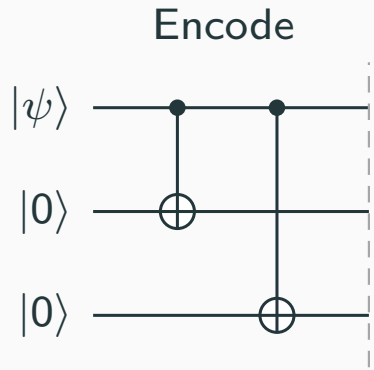
Assume that **at most one** qubit has flipped.

$$a = x_1 \oplus x_2, \qquad b = x_2 \oplus x_3.$$

Error	a	b	Correction
None	0	0	Do nothing
First qubit	1	0	Apply X to qubit 1
Second qubit	1	1	Apply X to qubit 2
Third qubit	0	1	Apply X to qubit 3

Correction and decoding

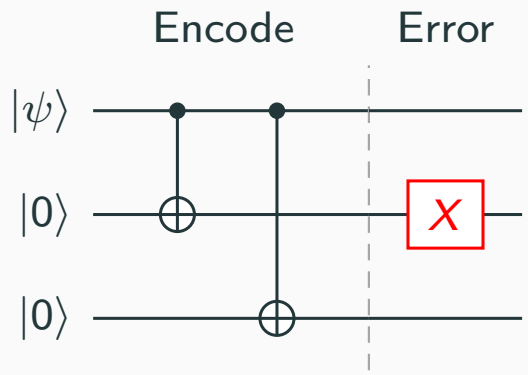
1. **Encode** $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$.



$$\alpha\,|000\rangle + \beta\,|111\rangle\,.$$

Correction and decoding

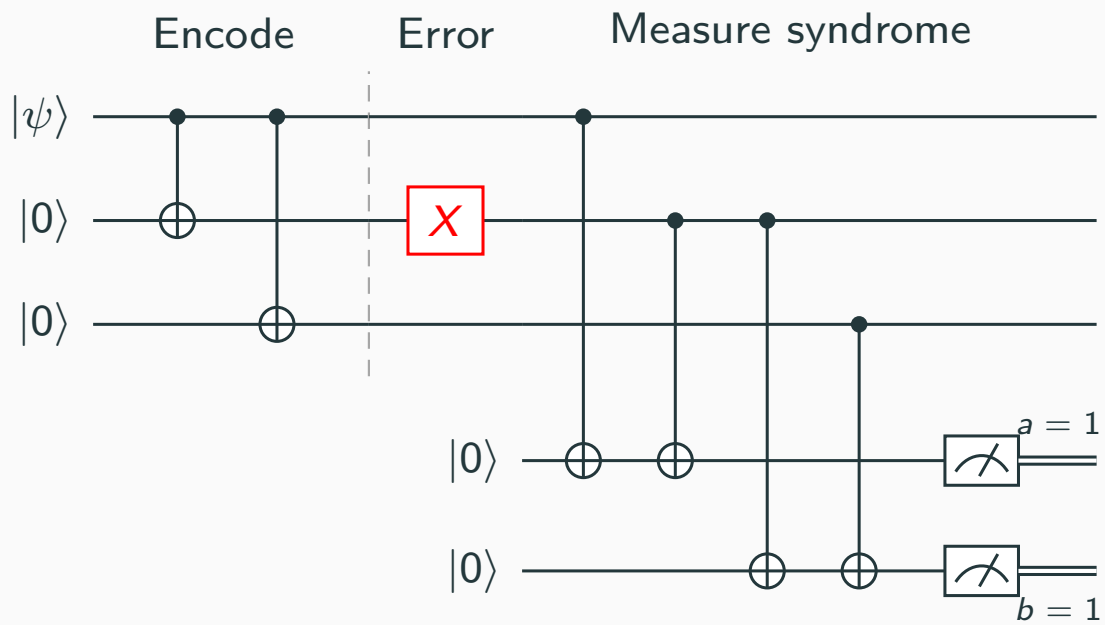
2. **An error occurs.** Suppose the second qubit flips.



$$\alpha\,|010\rangle + \beta\,|101\rangle\,.$$

Correction and decoding

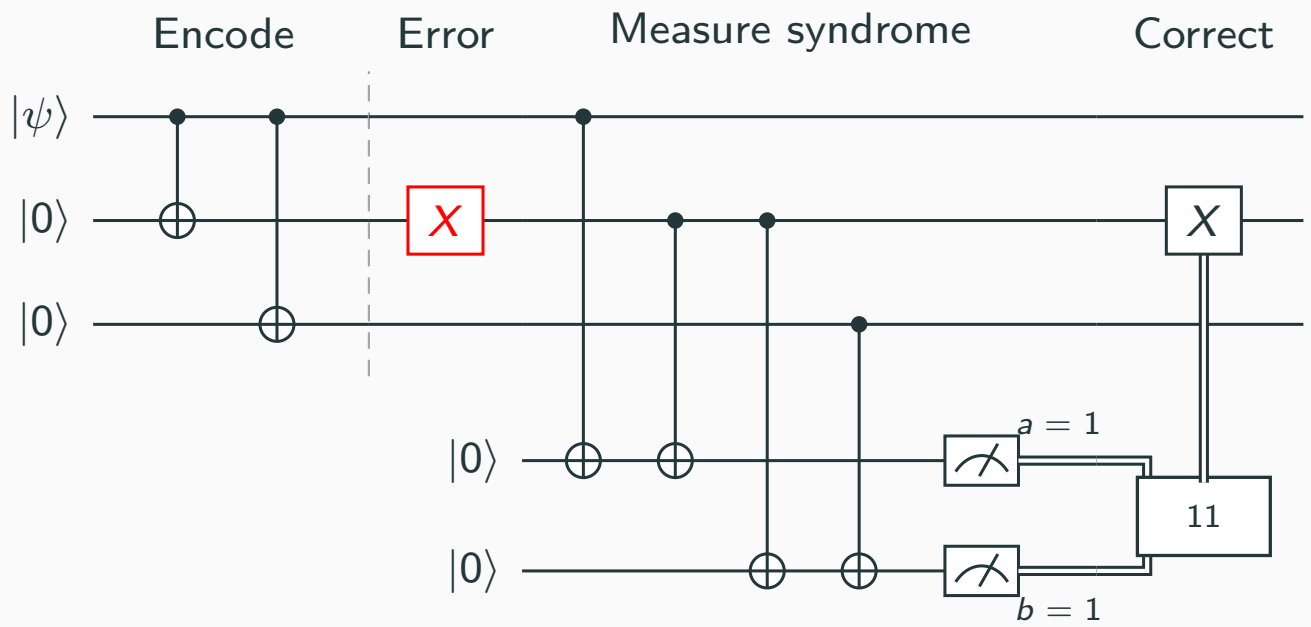
3. Measure the syndrome. Here $a = b = 1$.



$$(\alpha\,|010\rangle + \beta\,|101\rangle)\,|11\rangle\,.$$

Correction and decoding

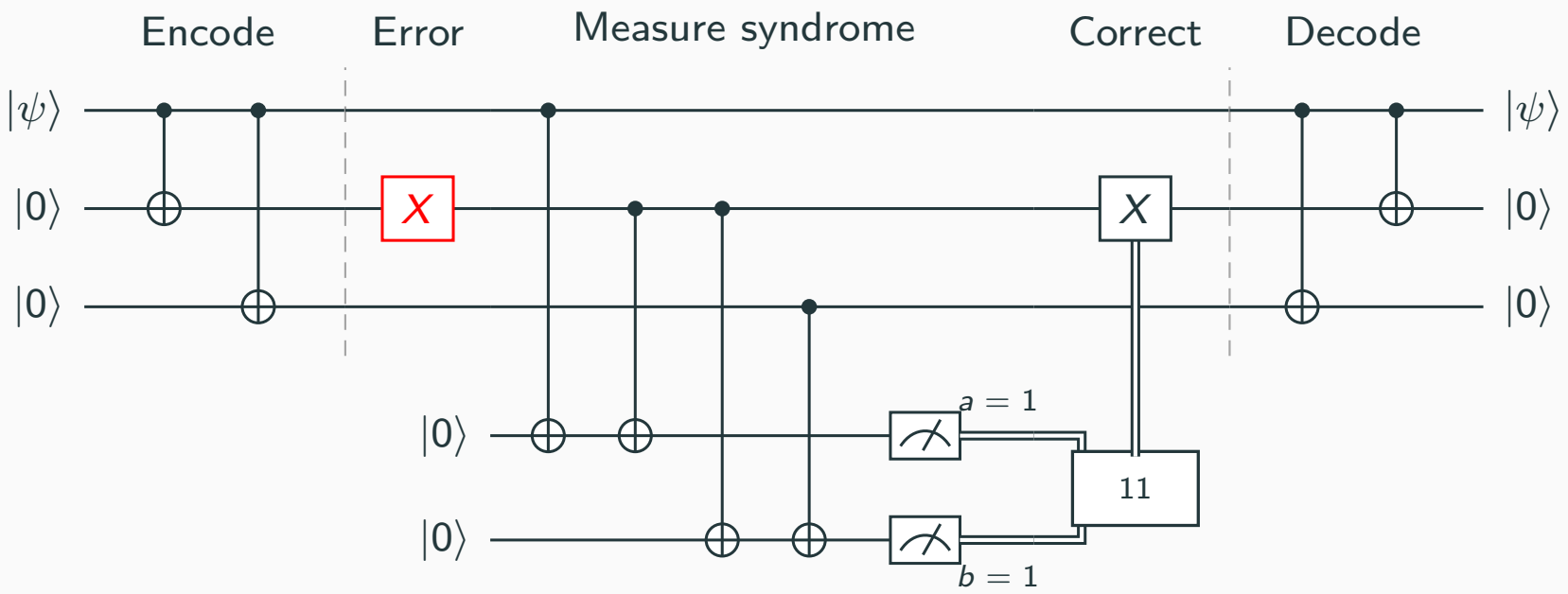
4. Correct. Syndrome 11: apply X to qubit 2.



$$\alpha\,|000\rangle + \beta\,|111\rangle\,.$$

Correction and decoding

5. Decode. Run the two encoding gates in reverse.



$$(\alpha\,|0\rangle + \beta\,|1\rangle)\,|00\rangle = |\psi\rangle\,|00\rangle\,.$$

Correction restores the encoded state. Decoding recovers $|\psi\rangle$.

What if two qubits flip?

Suppose the first two qubits flip:

$$\alpha\,|000\rangle + \beta\,|111\rangle \xrightarrow{X_1X_2} \alpha\,|110\rangle + \beta\,|001\rangle\,.$$

What if two qubits flip?

Suppose the first two qubits flip:

$$\alpha\,|000\rangle + \beta\,|111\rangle \xrightarrow{X_1X_2} \alpha\,|110\rangle + \beta\,|001\rangle\,.$$

The syndrome is 01, just as if only the third qubit had flipped. Our correction procedure applies X_3:

$$\alpha\,|110\rangle + \beta\,|001\rangle \xrightarrow{X_3} \alpha\,|111\rangle + \beta\,|000\rangle\,.$$

What if two qubits flip?

Suppose the first two qubits flip:

$$\alpha\,|000\rangle + \beta\,|111\rangle \xrightarrow{X_1X_2} \alpha\,|110\rangle + \beta\,|001\rangle\,.$$

The syndrome is 01, just as if only the third qubit had flipped. Our correction procedure applies X_3:

$$\alpha\,|110\rangle + \beta\,|001\rangle \xrightarrow{X_3} \alpha\,|111\rangle + \beta\,|000\rangle\,.$$

After decoding, we get

$$\beta\,|0\rangle + \alpha\,|1\rangle = X\,|\psi\rangle\,,$$

not the original state. We have introduced a **logical error**.

What if two qubits flip?

Suppose the first two qubits flip:

$$\alpha\,|000\rangle + \beta\,|111\rangle \xrightarrow{X_1X_2} \alpha\,|110\rangle + \beta\,|001\rangle\,.$$

The syndrome is 01, just as if only the third qubit had flipped. Our correction procedure applies X_3:

$$\alpha\,|110\rangle + \beta\,|001\rangle \xrightarrow{X_3} \alpha\,|111\rangle + \beta\,|000\rangle\,.$$

After decoding, we get

$$\beta\,|0\rangle + \alpha\,|1\rangle = X\,|\psi\rangle\,,$$

not the original state. We have introduced a **logical error**.

Three flips give syndrome 00 and the same logical error.

What about phase flips?

We can correct a single bit flip.

What about phase flips?

We can correct a single bit flip.

What happens if a qubit experiences a phase flip instead?

$$\alpha\,|000\rangle + \beta\,|111\rangle \xrightarrow{Z_2} \alpha\,|000\rangle - \beta\,|111\rangle\,.$$

What about phase flips?

We can correct a single bit flip.

What happens if a qubit experiences a phase flip instead?

$$\alpha\,|000\rangle + \beta\,|111\rangle \xrightarrow{Z_2} \alpha\,|000\rangle - \beta\,|111\rangle\,.$$

Bit flip repetition code no longer works! Do you see why?

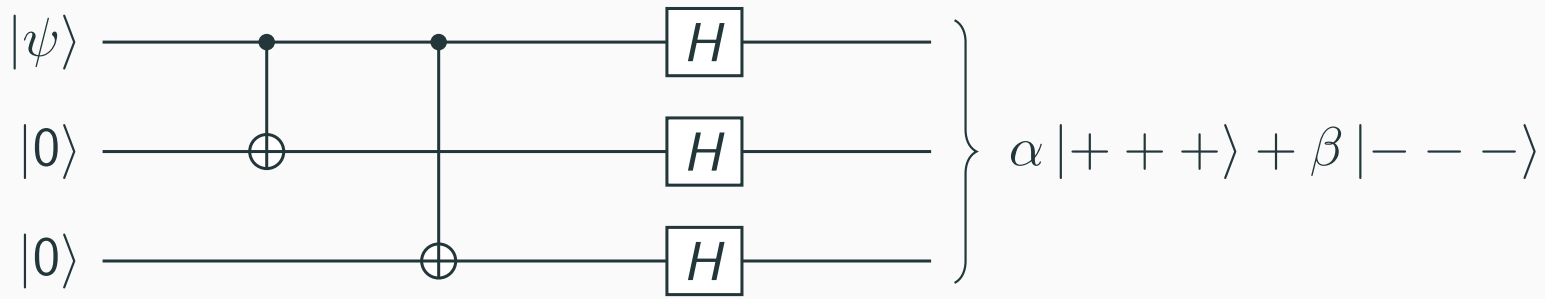
Correcting phase flip errors

Use the repetition code in the **diagonal (Hadamard) basis**:

$$|0\rangle \longmapsto |+++\rangle\,, \qquad |1\rangle \longmapsto |---\rangle\,.$$

By linearity,

$$\alpha\,|0\rangle + \beta\,|1\rangle \longmapsto \alpha\,|+++\rangle + \beta\,|---\rangle\,.$$



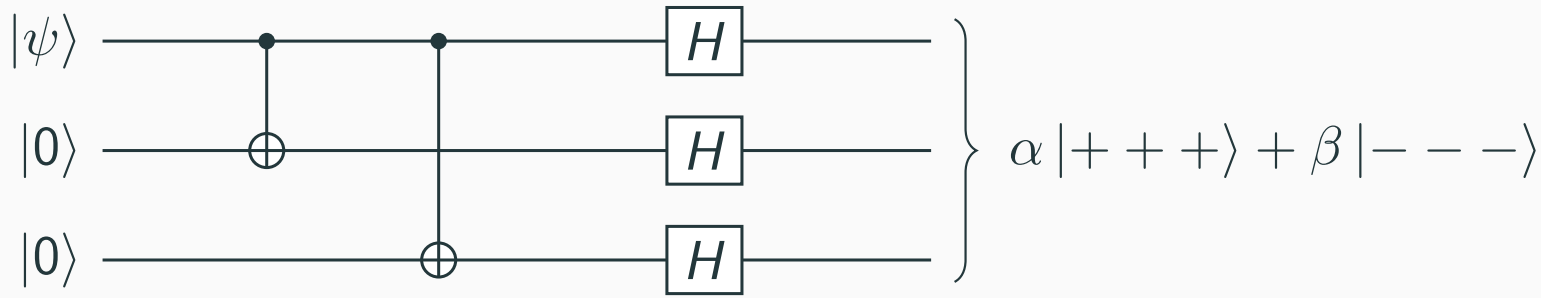
Correcting phase flip errors

Use the repetition code in the **diagonal (Hadamard) basis**:

$$|0\rangle \longmapsto |+++\rangle\,, \qquad |1\rangle \longmapsto |---\rangle\,.$$

By linearity,

$$\alpha\,|0\rangle + \beta\,|1\rangle \longmapsto \alpha\,|+++\rangle + \beta\,|---\rangle\,.$$



$$Z\,|+\rangle = |-\rangle\,, \qquad Z\,|-\rangle = |+\rangle\,.$$

A phase flip is just a bit flip in the diagonal basis!

Detecting and correcting phase flip errors

For example, a phase flip on the second qubit gives

$$\alpha \left|+ - +\right\rangle + \beta \left|- + -\right\rangle .$$

Detecting and correcting phase flip errors

For example, a phase flip on the second qubit gives

$$\alpha\,|+-+\rangle + \beta\,|-++\rangle\,.$$

Apply H to each qubit:

$$\alpha\,|+-+\rangle + \beta\,|-++\rangle \longmapsto \alpha\,|010\rangle + \beta\,|101\rangle\,.$$

We already know how to correct this bit flip.

Detecting and correcting phase flip errors

For example, a phase flip on the second qubit gives

$$\alpha\,|+-+\rangle + \beta\,|-+-\rangle\,.$$

Apply H to each qubit:

$$\alpha\,|+-+\rangle + \beta\,|-+-\rangle \longmapsto \alpha\,|010\rangle + \beta\,|101\rangle\,.$$

We already know how to correct this bit flip.



Apply H to each qubit again to return to

$$\alpha\,|+++\rangle + \beta\,|---\rangle\,.$$

Shor's 9 qubit code

Concatenate the bitflip and phaseflip codes:

1. Encode the logical qubit using the 3-qubit phaseflip code.
2. Encode each of those 3 qubits using the bitflip code.

Shor's 9 qubit code

Concatenate the bitflip and phaseflip codes:

1. Encode the logical qubit using the 3-qubit phaseflip code.
2. Encode each of those 3 qubits using the bitflip code.

$$|0\rangle \longmapsto |+\rangle^{\otimes 3} \longmapsto \left(\frac{|000\rangle + |111\rangle}{\sqrt{2}}\right)^{\otimes 3},$$

$$|1\rangle \longmapsto |-\rangle^{\otimes 3} \longmapsto \left(\frac{|000\rangle - |111\rangle}{\sqrt{2}}\right)^{\otimes 3}.$$

Shor's 9 qubit code

Concatenate the bitflip and phaseflip codes:

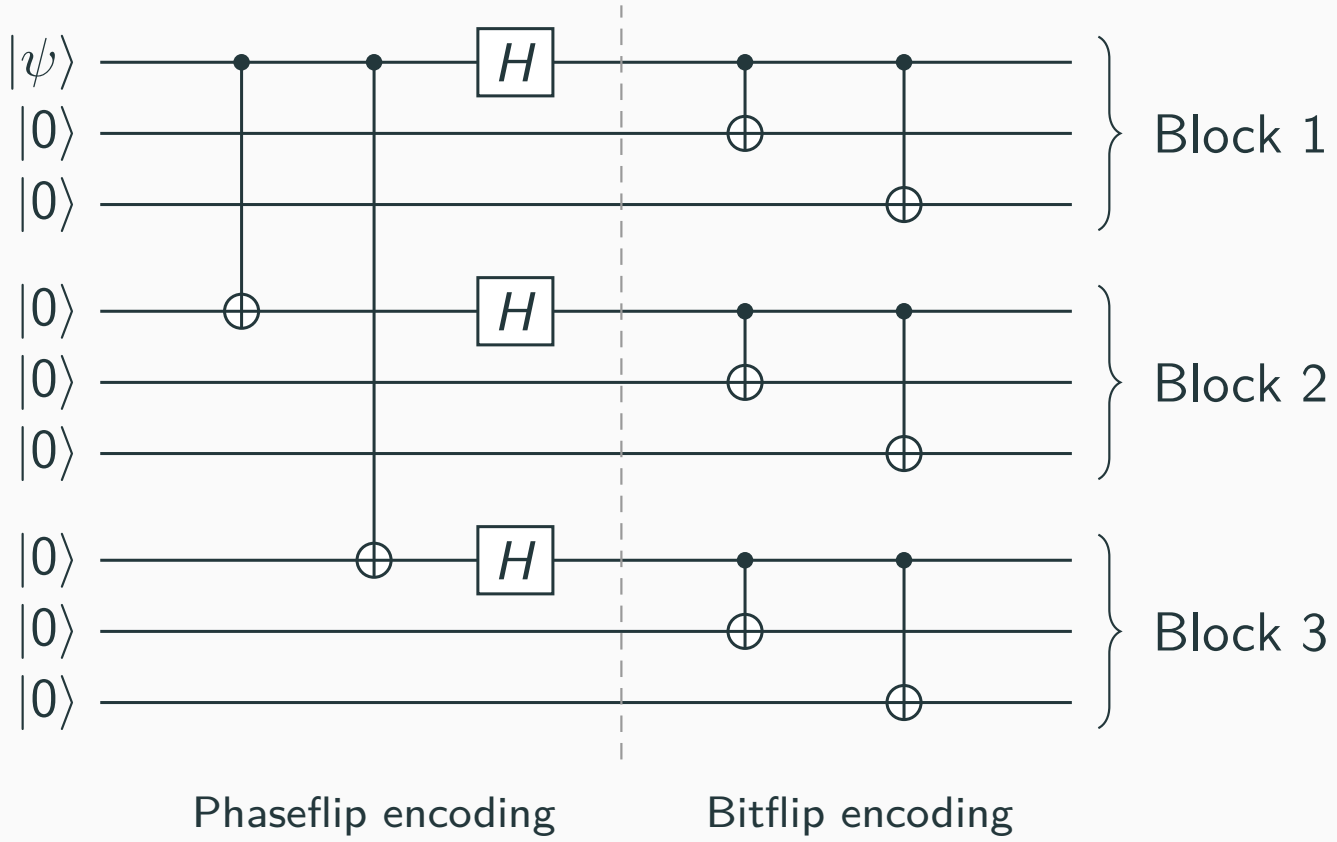
1. Encode the logical qubit using the 3-qubit phaseflip code.
2. Encode each of those 3 qubits using the bitflip code.

$$|0\rangle \longmapsto |+\rangle^{\otimes 3} \longmapsto \left(\frac{|000\rangle + |111\rangle}{\sqrt{2}}\right)^{\otimes 3},$$

$$|1\rangle \longmapsto |-\rangle^{\otimes 3} \longmapsto \left(\frac{|000\rangle - |111\rangle}{\sqrt{2}}\right)^{\otimes 3}.$$

Claim: This code corrects bitflip and phaseflip errors, as long as they occur on a single qubit.

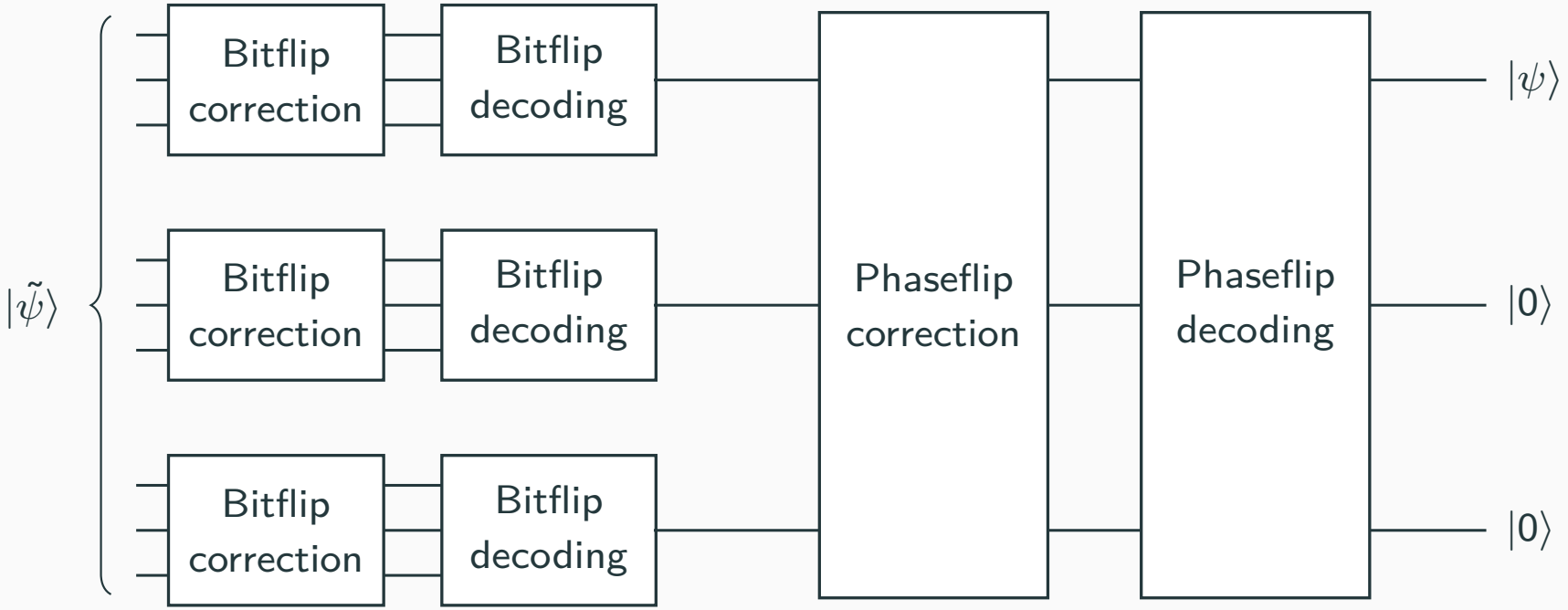
Shor's 9 qubit code: encoding



One logical qubit, encoded into nine physical qubits.

Shor's 9 qubit code: recovery

Correct and decode the bitflip blocks, then the phaseflip code.



Only the three remaining data qubits enter phaseflip correction.

The six unused $|0\rangle$ outputs of bitflip decoding are omitted. Syndrome ancillas are also omitted.

Shor code can detect and fix:

1. **Any single-qubit X error.**
   The bitflip correction in that block fixes it.

Shor code can detect and fix:

1. **Any single-qubit X error.**
   The bitflip correction in that block fixes it.
2. **Any single-qubit Z error.**
   Within a block, a phase flip changes

$$\frac{|000\rangle + |111\rangle}{\sqrt{2}} \longleftrightarrow \frac{|000\rangle - |111\rangle}{\sqrt{2}}.$$

   After bitflip decoding, this is a Z error on one of the three phase-code qubits. The phaseflip correction fixes it.

Shor code can detect and fix:

1. **Any single-qubit X error.**
   The bitflip correction in that block fixes it.
2. **Any single-qubit Z error.**
   Within a block, a phase flip changes

$$\frac{|000\rangle + |111\rangle}{\sqrt{2}} \longleftrightarrow \frac{|000\rangle - |111\rangle}{\sqrt{2}}.$$

   After bitflip decoding, this is a Z error on one of the three phase-code qubits. The phaseflip correction fixes it.
3. **Any single-qubit XZ error.**
   First correct the bit flip, then the phase flip.

Why errors can be discretized

The **Pauli matrices** are

$$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

We already know how to correct X, Z, and $Y = iXZ$.

Why errors can be discretized

The **Pauli matrices** are

$$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \; X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \; Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \; Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

We already know how to correct X, Z, and $Y = iXZ$.

Fact: Every 2×2 matrix is a linear combination of these matrices:

$$E = aI + bX + cY + dZ, \qquad a, b, c, d \in \mathbb{C}.$$

Why errors can be discretized

The **Pauli matrices** are

$$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \; X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \; Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \; Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

We already know how to correct X, Z, and $Y = iXZ$.

Fact: Every 2×2 matrix is a linear combination of these matrices:

$$E = aI + bX + cY + dZ, \qquad a, b, c, d \in \mathbb{C}.$$

This includes a small over-rotation, not just a discrete flip.

Why should the *same correction procedure* work for E?

Why errors can be discretized

Let C be the correction and decoding procedure. For each correctable Pauli error P,

$$C\left(P\left|\bar{\psi}\right\rangle\left|0\cdots0\right\rangle\right) = \left|\psi\right\rangle\left|\mathsf{anc}(P)\right\rangle.$$

The ancilla state depends on the error, but not on $|\psi\rangle$.

Why errors can be discretized

Let C be the correction and decoding procedure. For each correctable Pauli error P,

$$C\left(P\left|\bar{\psi}\right\rangle\left|0\cdots0\right\rangle\right) = \left|\psi\right\rangle\left|\mathrm{anc}(P)\right\rangle.$$

The ancilla state depends on the error, but not on $\left|\psi\right\rangle$.

Suppose $E = aI + bX + cY + dZ$ acts on the first qubit. By linearity,

$$\begin{aligned} &C\left(E_1\left|\bar{\psi}\right\rangle\left|0\cdots0\right\rangle\right) \\ &\quad = \left|\psi\right\rangle \otimes \Big(a\left|\mathrm{anc}(I)\right\rangle + b\left|\mathrm{anc}(X_1)\right\rangle \\ &\qquad\qquad + c\left|\mathrm{anc}(Y_1)\right\rangle + d\left|\mathrm{anc}(Z_1)\right\rangle\Big). \end{aligned}$$

Why errors can be discretized

Let C be the correction and decoding procedure. For each correctable Pauli error P,

$$C\left(P\left|\bar{\psi}\right\rangle\left|0\cdots0\right\rangle\right) = \left|\psi\right\rangle\left|\mathrm{anc}(P)\right\rangle.$$

The ancilla state depends on the error, but not on $|\psi\rangle$.

Suppose $E = aI + bX + cY + dZ$ acts on the first qubit. By linearity,

$$\begin{aligned} &C\left(E_1\left|\bar{\psi}\right\rangle\left|0\cdots0\right\rangle\right) \\ &\quad= \left|\psi\right\rangle \otimes \Big(a\left|\mathrm{anc}(I)\right\rangle + b\left|\mathrm{anc}(X_1)\right\rangle \\ &\qquad\qquad + c\left|\mathrm{anc}(Y_1)\right\rangle + d\left|\mathrm{anc}(Z_1)\right\rangle\Big). \end{aligned}$$

We recover $|\psi\rangle$. All the dependence on E is in the ancillas.

Fault tolerance

Fault tolerance

Our analysis assumes that encoding, correction, and decoding are all done perfectly.

But these operations will also experience errors!

Fault tolerance

Our analysis assumes that encoding, correction, and decoding are all done perfectly.

But these operations will also experience errors!

Also, we do not just want to store information. We want to **compute** with it.

Fault tolerance

Our analysis assumes that encoding, correction, and decoding are all done perfectly.

But these operations will also experience errors!

Also, we do not just want to store information. We want to **compute** with it.

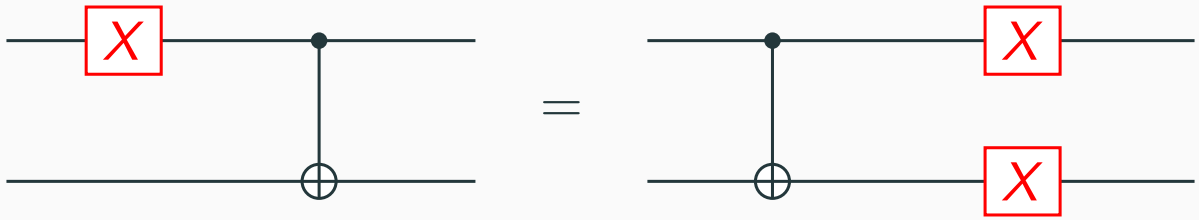
Main idea: Perform computations on encoded states, without decoding between gates.



The gates *and* the correction procedures must tolerate faults.

Error propagation

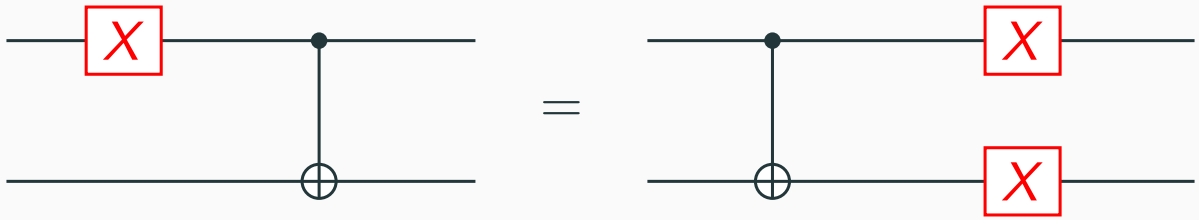
Single-qubit gates do not spread errors to other qubits. Two-qubit gates can.



$$\mathrm{CNOT}(X \otimes I) = (X \otimes X)\mathrm{CNOT}.$$

Error propagation

Single-qubit gates do not spread errors to other qubits. Two-qubit gates can.



$$\mathrm{CNOT}(X \otimes I) = (X \otimes X)\mathrm{CNOT}.$$

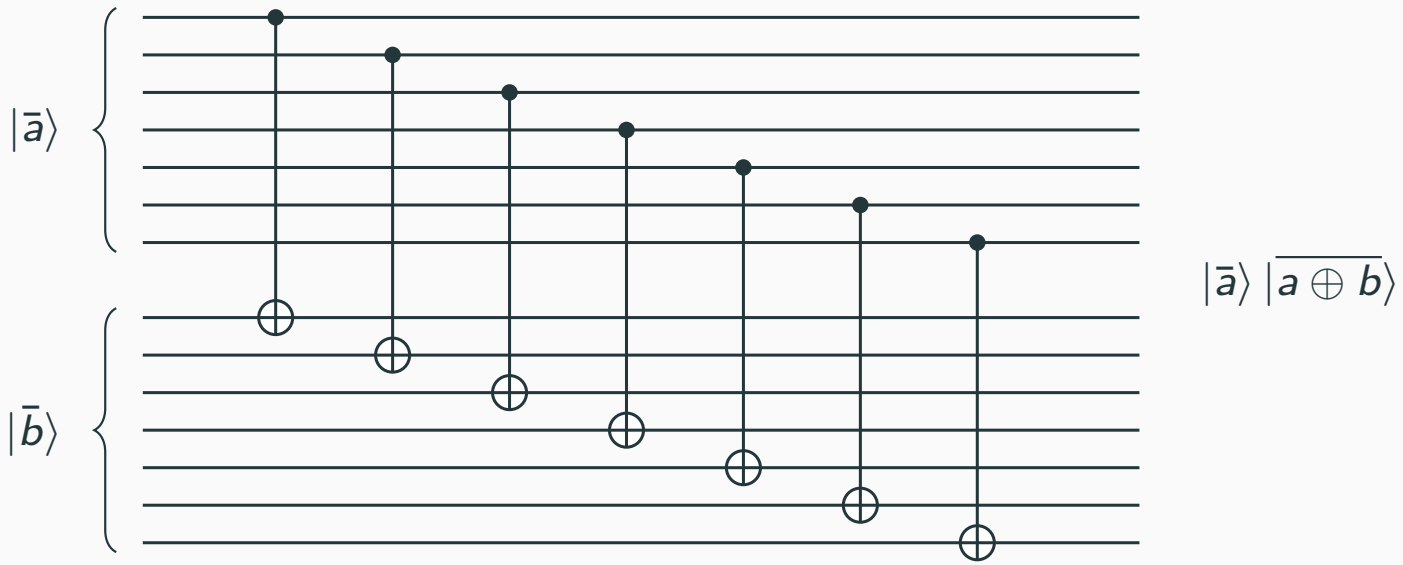
One error has become two errors.

If both qubits belong to the same code block, this may be more than the code can correct.

We must design the circuit so that a single fault does not cause an uncorrectable error.

Performing encoded gates

For the 7-qubit Steane code, pairwise CNOTs implement an encoded CNOT:



This is a **transversal** gate. One faulty CNOT affects at most one qubit in each block.

The Steane code supports transversal **Clifford gates**, including H, the phase gate P, and CNOT.

Eastin–Knill Theorem

Life would be really nice if there was a code that supported transversal gates for a universal gate set.

Eastin–Knill Theorem

Life would be really nice if there was a code that supported transversal gates for a universal gate set.

Eastin–Knill Theorem: A code that detects arbitrary single-qubit errors cannot have a universal set of transversal gates.

Eastin–Knill Theorem

Life would be really nice if there was a code that supported transversal gates for a universal gate set.

Eastin–Knill Theorem: A code that detects arbitrary single-qubit errors cannot have a universal set of transversal gates.

For the Steane code, what's missing is a transversal implementation of

$$T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}.$$

Need another way of performing encoded T gates.

Magic states are one solution. Code switching is another.

Adapted from the 2025 lecture. Eastin–Knill (2009). The theorem does not require magic-state distillation specifically.

Magic states

To apply an encoded T gate, use a special ancilla:

$$|T\rangle = T\,|+\rangle = \frac{|0\rangle + e^{i\pi/4}\,|1\rangle}{\sqrt{2}}.$$

Magic states

To apply an encoded T gate, use a special ancilla:

$$|T\rangle = T\,|+\rangle = \frac{|0\rangle + e^{i\pi/4}\,|1\rangle}{\sqrt{2}}.$$



The magic state is consumed. We need a new one for the next T gate.

Magic states

To apply an encoded T gate, use a special ancilla:

$$|T\rangle = T\,|+\rangle = \frac{|0\rangle + e^{i\pi/4}\,|1\rangle}{\sqrt{2}}.$$



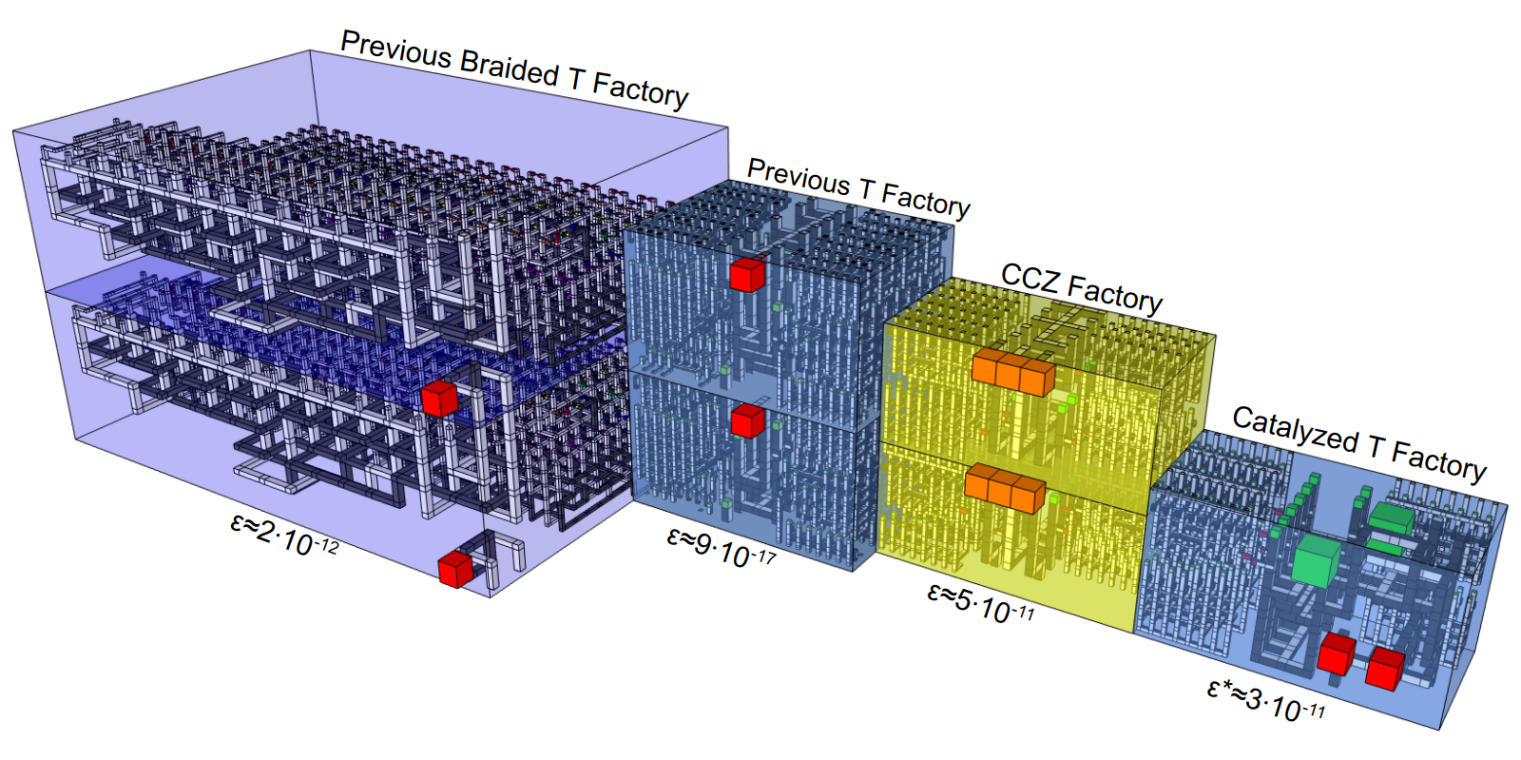
The magic state is consumed. We need a new one for the next T gate.

Magic state distillation: use several noisy magic states to produce fewer, more accurate ones.

These routines must be fault tolerant too.

Bravyi–Kitaev (2004/2005). Input states must be sufficiently accurate for distillation to work.

T factories



A **T factory** repeatedly prepares accurate magic states.

In many fault-tolerance schemes, this is where much of the complexity and cost comes from.

Original figure from the 2025 lecture, slide 32; Gidney–Fowler (2019). Factory overhead depends on the architecture and algorithm.

The Fault-Tolerance Theorem

Also called the **Threshold Theorem**.

For suitable local-noise models, there is a constant $p_{\text{th}} > 0$ such that, if the physical error rate satisfies

$$p < p_{\text{th}},$$

any ideal size-S quantum circuit can be simulated with error at most ϵ, using

$$O(S \text{ polylog}(S/\epsilon))$$

noisy operations.

The Fault-Tolerance Theorem

Also called the **Threshold Theorem**.

For suitable local-noise models, there is a constant $p_{\text{th}} > 0$ such that, if the physical error rate satisfies

$$p < p_{\text{th}},$$

any ideal size-S quantum circuit can be simulated with error at most ϵ, using

$$O(S \text{ polylog}(S/\epsilon))$$

noisy operations.

Assumptions include parallel operations, fresh or reset qubits, and sufficiently fast, reliable classical processing.

The threshold depends on the noise model and the construction.

Aharonov–Ben-Or, quant-ph/9611025; Knill–Laflamme–Zurek (1996–98). Informal statement.

Recent results

From logical qubits to physical qubits

A **logical qubit** is a protected qubit used by the algorithm.

Physical qubits are the hardware used to store and protect it.

From logical qubits to physical qubits

A **logical qubit** is a protected qubit used by the algorithm.

Physical qubits are the hardware used to store and protect it.

To estimate the size of a computer, we need more than the number of logical qubits:

1. The circuit: how many gates, and which can run in parallel?
2. The hardware: error rates, connectivity, and operation times.
3. The code, correction circuits, and magic-state factories.

From logical qubits to physical qubits

A **logical qubit** is a protected qubit used by the algorithm.

Physical qubits are the hardware used to store and protect it.

To estimate the size of a computer, we need more than the number of logical qubits:

1. The circuit: how many gates, and which can run in parallel?
2. The hardware: error rates, connectivity, and operation times.
3. The code, correction circuits, and magic-state factories.

These are **resource estimates**, not demonstrations or predictions of a particular year.

Gidney (2025).

Google resource estimate (March 2026)

256-bit elliptic-curve discrete logarithms (secp256k1):

Logical qubits	Toffoli gates
$\leq$ 1,200	$\leq$ 90 million
$\leq$ 1,450	$\leq$ 70 million

Google resource estimate (March 2026)

256-bit elliptic-curve discrete logarithms (secp256k1):

Logical qubits	Toffoli gates
$\leq$ 1,200	$\leq$ 90 million
$\leq$ 1,450	$\leq$ 70 million

Fewer than 500,000 physical qubits; 18–23 minutes.

Assumes superconducting hardware, planar connectivity, 10^{-3} physical error rates, and a 10 μs control reaction time.

Google resource estimate (March 2026)

256-bit elliptic-curve discrete logarithms (secp256k1):

Logical qubits	Toffoli gates
$\leq$ 1,200	$\leq$ 90 million
$\leq$ 1,450	$\leq$ 70 million

Fewer than 500,000 physical qubits; 18–23 minutes.

Assumes superconducting hardware, planar connectivity, 10^{-3} physical error rates, and a 10 μs control reaction time.

Full runtime above. The 9–12 minute figure assumes the first half has already been computed.

Babbush et al., Securing Elliptic Curve Cryptocurrencies against Quantum Vulnerabilities (March 30, 2026), pp. 1, 8.

Oratomic resource estimates (March 2026)

Reconfigurable neutral atoms and high-rate error-correcting codes:

Problem	Configuration	Physical qubits	Time
ECC–256	Fewer qubits	9,739	$\sim$ 1,000 days
ECC–256	More parallelism	$\sim$ 26,000	$\sim$ 10 days
RSA–2048	More parallelism	$\sim$ 102,000	$\sim$ 97 days

Oratomic resource estimates (March 2026)

Reconfigurable neutral atoms and high-rate error-correcting codes:

Problem	Configuration	Physical qubits	Time
ECC–256	Fewer qubits	9,739	$\sim$ 1,000 days
ECC–256	More parallelism	$\sim$ 26,000	$\sim$ 10 days
RSA–2048	More parallelism	$\sim$ 102,000	$\sim$ 97 days

These numbers assume a 1 ms error-correction cycle. The faster configurations assume higher-rate codes and many logical operations running in parallel.

Oratomic resource estimates (March 2026)

Reconfigurable neutral atoms and high-rate error-correcting codes:

Problem	Configuration	Physical qubits	Time
ECC–256	Fewer qubits	9,739	$\sim$ 1,000 days
ECC–256	More parallelism	$\sim$ 26,000	$\sim$ 10 days
RSA–2048	More parallelism	$\sim$ 102,000	$\sim$ 97 days

These numbers assume a 1 ms error-correction cycle. The faster configurations assume higher-rate codes and many logical operations running in parallel.

"About 10,000 qubits" and "about 10 days" are *not* the same configuration.

Different connectivity and clock speeds from Google's estimate.

Cain et al., arXiv:2603.28627 (March 2026), Fig. 3 and Appendix F. Rounded runtimes; theoretical estimates.

An experimental result (October 2026)

Google demonstrated **magic state cultivation** on its Willow processor.

A different way to prepare accurate magic states: inject into a small code, check for errors, and grow the code. Discard failed attempts.

An experimental result (October 2026)

Google demonstrated **magic state cultivation** on its Willow processor.

A different way to prepare accurate magic states: inject into a small code, check for errors, and grow the code. Discard failed attempts.

Reported results:

- About 40× lower error than uncultivated states.
- Magic-state fidelity about 99.99%.
- 8% of attempts retained.

An experimental result (October 2026)

Google demonstrated **magic state cultivation** on its Willow processor.

A different way to prepare accurate magic states: inject into a small code, check for errors, and grow the code. Discard failed attempts.

Reported results:

- About 40× lower error than uncultivated states.
- Magic-state fidelity about 99.99%.
- 8% of attempts retained.

This is a demonstrated component, not a complete computer running the cryptographic algorithms on the previous slides.

Rosenfeld et al., Science, 10.1126/science.aee7961, published October 15, 2026.

Summary

1. Encode quantum information using redundancy, without cloning the state.

Summary

1. Encode quantum information using redundancy, without cloning the state.
2. Measure error syndromes, not the logical information.

Summary

1. Encode quantum information using redundancy, without cloning the state.
2. Measure error syndromes, not the logical information.
3. Correcting Pauli errors is enough to correct general single-qubit errors.

Summary

1. Encode quantum information using redundancy, without cloning the state.
2. Measure error syndromes, not the logical information.
3. Correcting Pauli errors is enough to correct general single-qubit errors.
4. Perform gates and error correction on encoded data. Below threshold, longer computations are possible with more encoding.

Summary

1. Encode quantum information using redundancy, without cloning the state.
2. Measure error syndromes, not the logical information.
3. Correcting Pauli errors is enough to correct general single-qubit errors.
4. Perform gates and error correction on encoded data. Below threshold, longer computations are possible with more encoding.
5. Transversal gates alone are not universal. Preparing magic states is a major part of the cost of many fault-tolerant computations.