Guest Lecture: Fault-Tolerant Quantum Computing*

Michael Beverland, IBM

Note: The material in these notes is loosely based on lectures given by Michael Vasmer at the University of Waterloo in a joined lecture course with Debbie Leung.

https://www.math.uwaterloo.ca/~wcleung/qic890-w2024.html

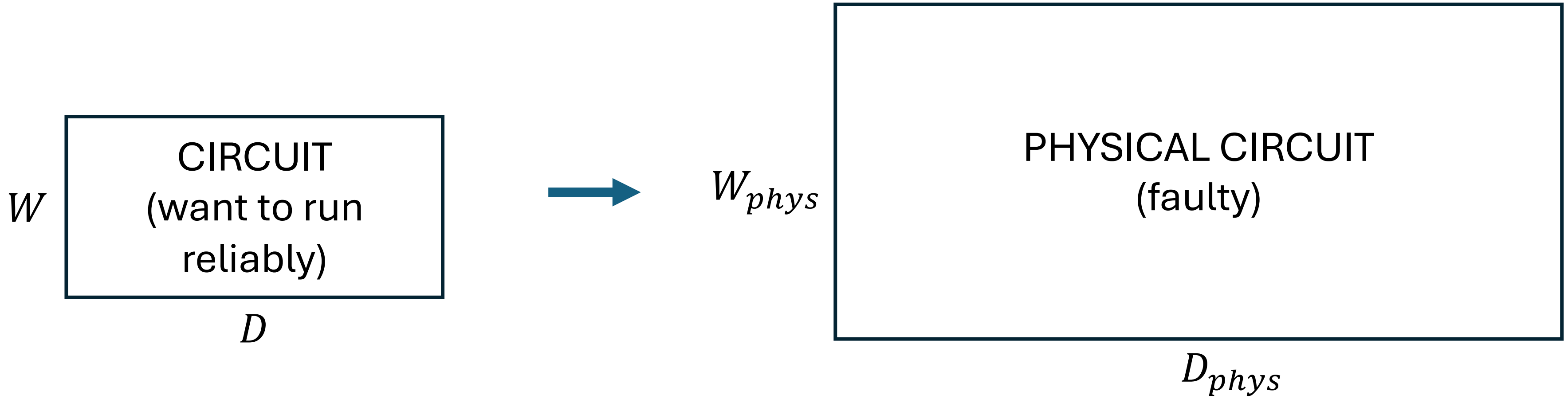
Summary

- Goal of fault-tolerant (FT) quantum computing
- FT quantum error correction (FTQEC): challenges
- FTQEC: overcoming challenges (Shor 94)
- FTQEC: general conditions
- FT logical gates
- Threshold theorem

Goal of fault-tolerant (FT) quantum computing

Goal of fault-tolerant (FT) quantum computing

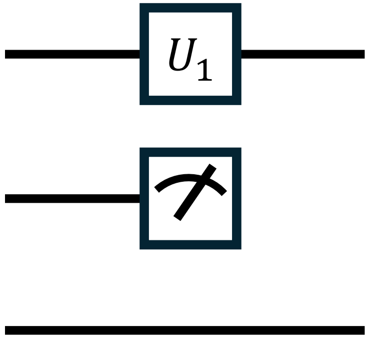
- We want to implement a large quantum circuit (e.g. Shor's algo to factor a large number) reliably.
- However, hardware has errors - current worst-operation error rates $> 10^{-3}$ (all platforms). Optimistic future: 10^{-4}?
- But more than 10^{15} operations in a cryptographically-relevant run of Shor's algorithm. ***Errors are a certainty.***



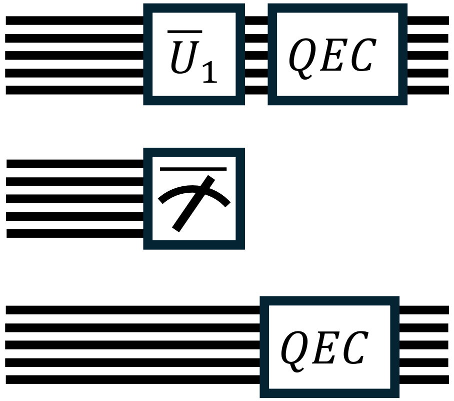
Basic approach for FT quantum computing

The basic approach is to:

- Encode qubits of circuit in quantum error-correcting (QEC) code.
- Replace physical operations with corresponding logical operation.
- Ensure logical operations do not spread errors excessively.
- Periodically apply error correction (to prevent error build-up).



circuit we want to implement



fault tolerant implementation

(bar denotes logical operation)

Circuit noise model

Useful to have a concrete mathematical model for noise in the physical circuit. Here we describe the simplest noise model used in fault tolerance analysis.

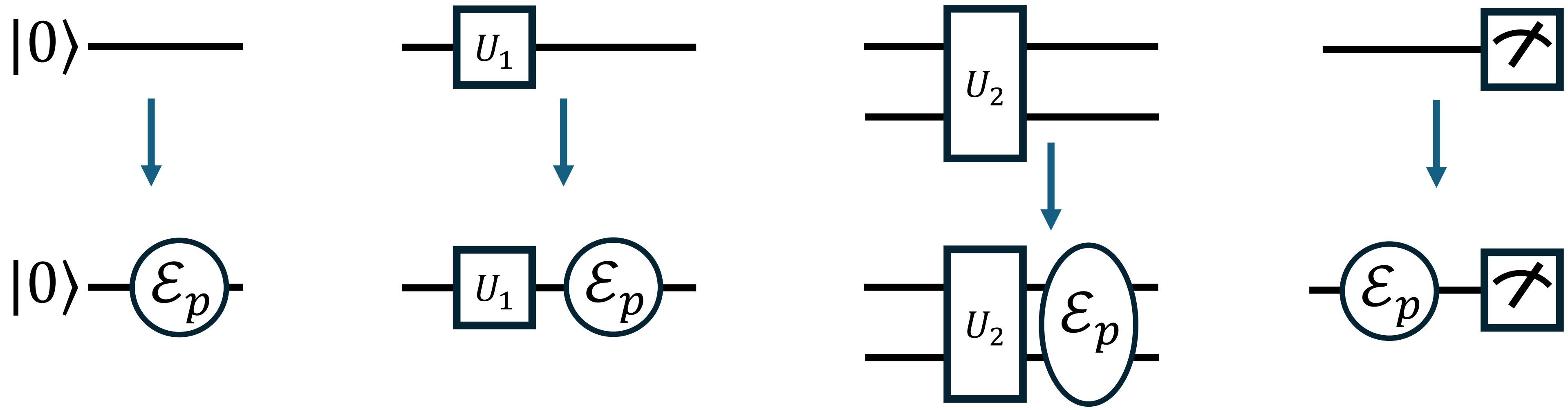
Break physical circuit into locations, where a location is:

- A gate (1-qubit, 2-qubit),
- A measurement,
- A state preparation (generally |0⟩),
- A storage/wait location.

Noise:

- with independent probability $(1 - p)$, each location functions as intended.
- With probability p, a random Pauli operator is applied to support.

Circuit noise model



The random Paulis are equivalent to the "depolarizing channel".

Other more nuanced noise models can be considered.

We will focus most of lecture on how to do QEC fault tolerantly.

FT quantum error correction (FTQEC): challenges

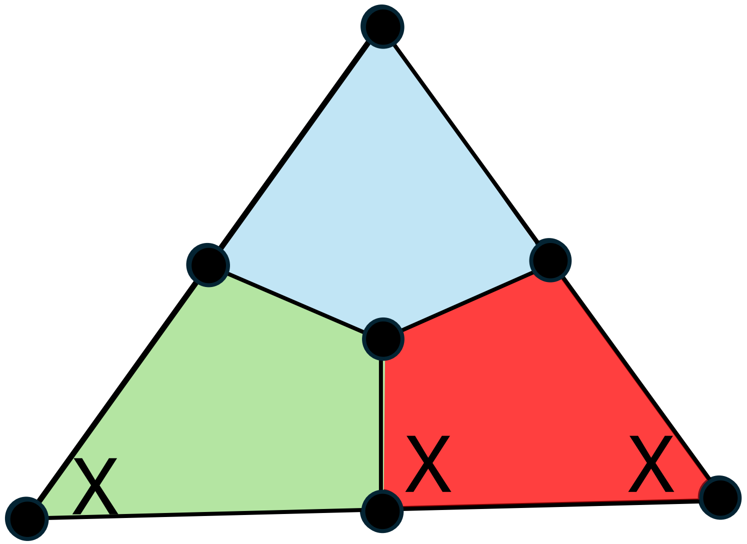
FTQEC: challenges

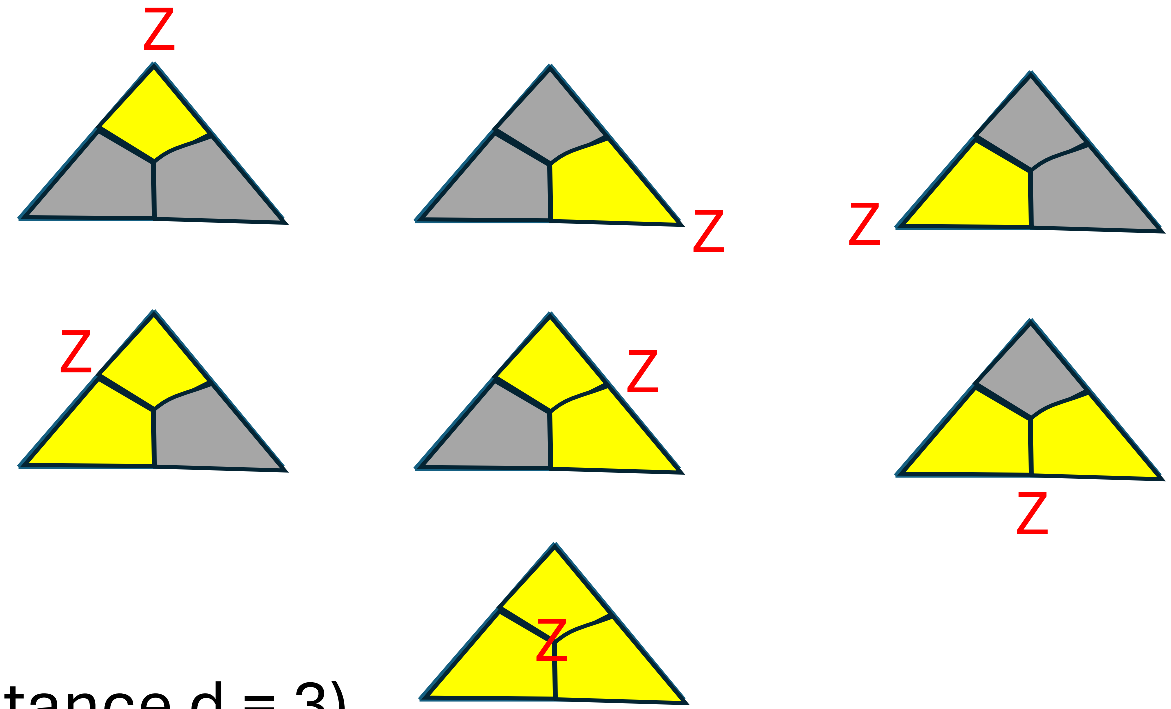
In your previous lectures, you considered error models where data qubits are affected by errors.

But what if the error correction circuits (e.g for measuring stabilizers) are themselves also noisy? This is the setting of FTQEC.

FTQEC is a bit like putting out a fire with a fire extinguisher that is also on fire!

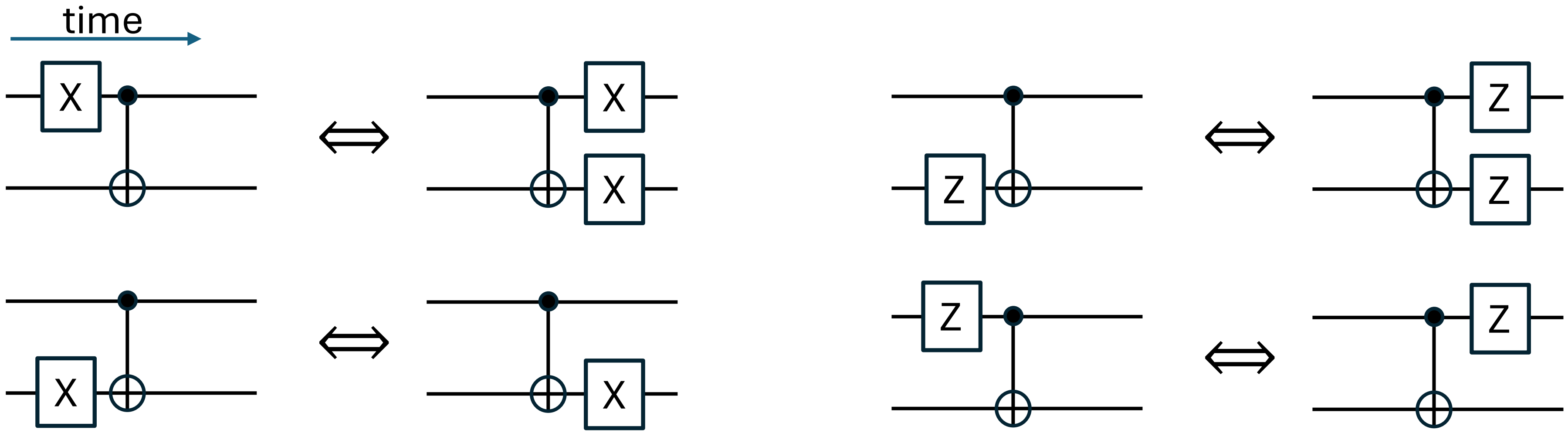
Reminder of regular (non-FT) QEC





- Steane code (n = 7 qubits, distance d = 3).
- Four qubits in each colored face form an X stabilizer.
- Z stabilizers have the same support.
- A logical X operator is on 3 outer-edge qubits.
- Error patterns can be identified visually as shown on the right.

CNOT circuit identities

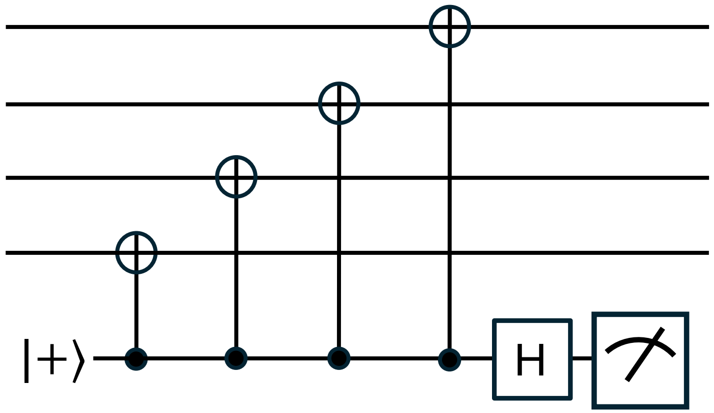


These circuit equivalence rules will help us later to understand how errors can affect a circuit.

Measuring a stabilizer with a noisy circuit

Cannot directly measure a stabilizer, which are weight-4 Paulis. (Assuming hardware just has 1- and 2-qubit gates.)

Consider this circuit to measure a stabilizer using an extra qubit:



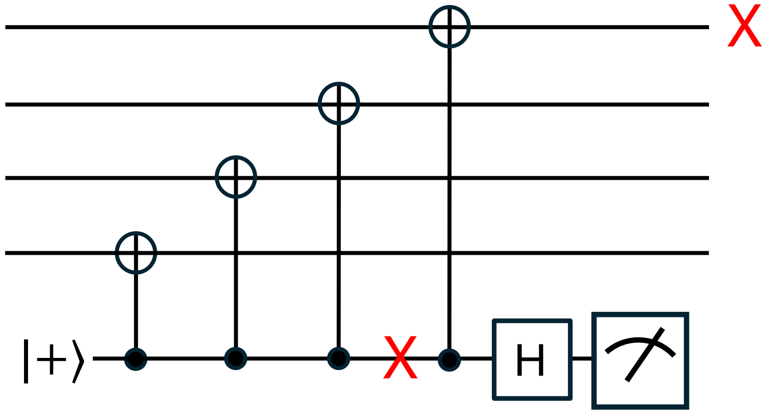
To verify, could write out matrices and verify the action.

For intuition, consider:

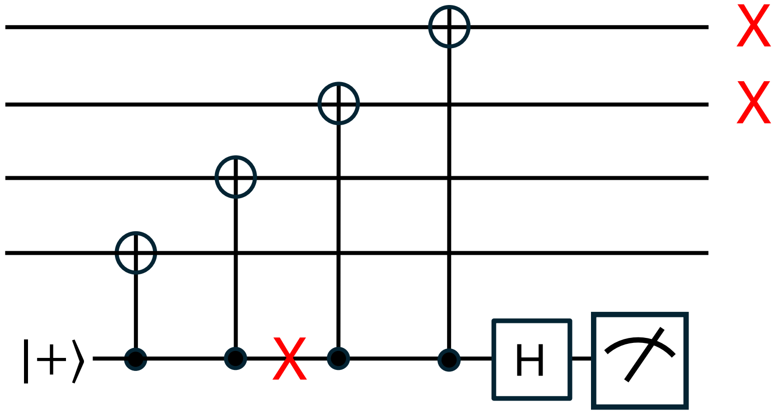
- what happens when X on data qubits?
- what happens when Z on data qubits?
- what about two Zs?

Challenge 1: error spread

- The code has d = 3, which means that $t = \lfloor (d-1)/2 \rfloor = 1$, i.e. it can correct a single error.
- We want to construct a circuit that can deal with all single faults.



Single fault propagates to single error (okay)

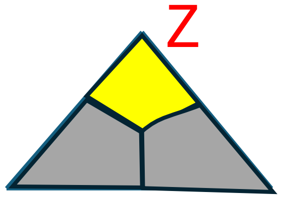


Single fault propagates to two errors (bad!)

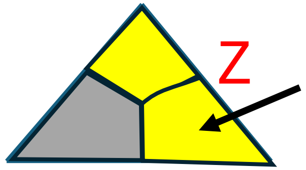
Challenge 2: unreliable stabilizer outcome

- The code has $d = 3$, which means that $t = \lfloor (d - 1)/2 \rfloor = 1$, i.e. it can correct a single error.
- We want to construct a circuit that can deal with all single faults.

real error

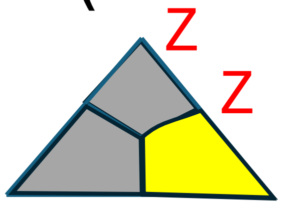


wrong correction



wrong outcome

net effect (uncorrectable)

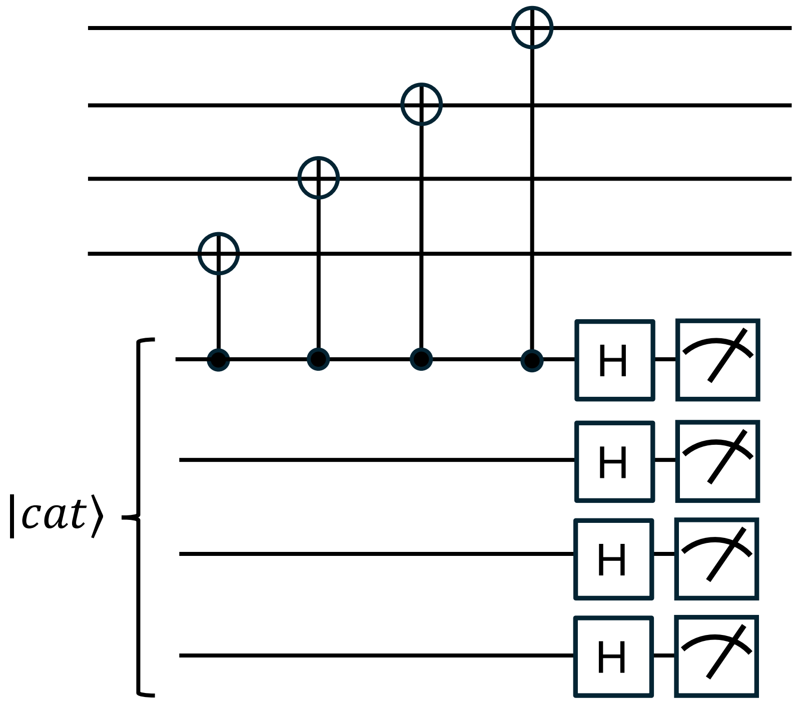


FTQEC: overcoming challenges (Shor 94)

Using cat states to avoid error spread

- Instead of using a single ancilla, use a cat state:

$$(|0000\rangle + |1111\rangle)/\sqrt{2}.$$



Output is parity of measurements at end.

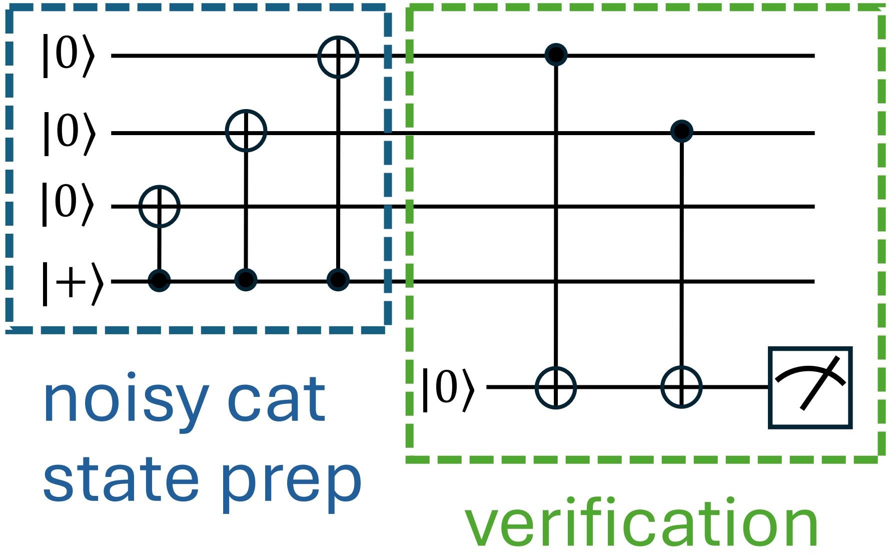
To verify, could write out matrices and verify the action.

Note, no-longer have spreading to multiple data qubits.

Making reliable cat states

To make the cat state reliably:

- Verify eigenvalues of cat state stabilizers: $X_1X_2X_3X_4$, Z_1Z_2, Z_2Z_3, Z_3Z_4.
- Accept if measurement result is +1, reject otherwise.



- The verification circuit catches all X errors on the input ancilla states.
- Single-qubit X errors can be introduced into the cat state by the verification process,
- but they have same effect as single-qubit X errors during stabilizer measurement.

Repetition to overcome unreliable outcomes

To avoid problems from wrong stabilizer outcomes, Shor’s protocol for FTQEC is essentially:

- Repeat the whole procedure (make and verify cat state, use for stabilizer measurement) 3 (or more) times for each stabilizer, and take the majority vote for the stabilizer measurement outcome.
- Considering only single faults anywhere during the entire procedure ensures an accurate result.

This is actually a slight simplification of the protocol, but gives the rough idea.

FTQEC: general conditions

FTQEC: general conditions

The previous analysis was a little bit unstructured – we pointed out some issues and found ways to overcome them.

But how do we know we are done?

And what if we come up with other approaches to FTQEC - how do we verify them?

We can define two sufficient conditions for FTQEC.

An intuitive condition and a less intuitive one

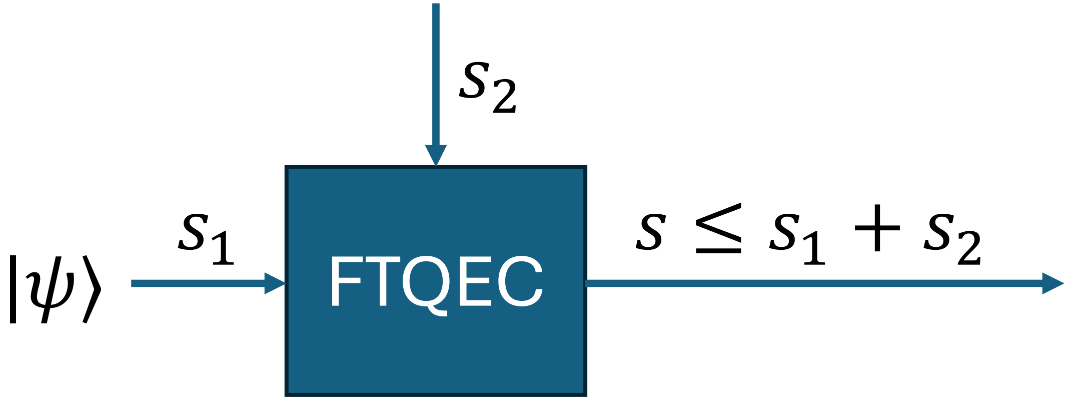
Let C be an [[n, k, d]] stabilizer code, and let t = ⌊(d − 1)/2⌋. An error correction protocol for C is fault-tolerant if:

(I) For an input codeword |ψ⟩ with error of weight s_1, if s_2 faults occur during the protocol such that $s_1 + s_2$ ≤ t, then perfectly decoding the output state gives |ψ⟩.
(II) For s ≤ t faults occurring during the protocol for an arbitrary input state, the output state differs from a codeword by an error of weight ≤ s.

Condition (I) seems fairly intuitive: it ensures that correctable errors don’t spread to uncorrectable errors during the course of the protocol.

Condition (II) may seem less intuitive. It essentially ensures that QEC can remove errors.

Why condition 1 alone is not enough



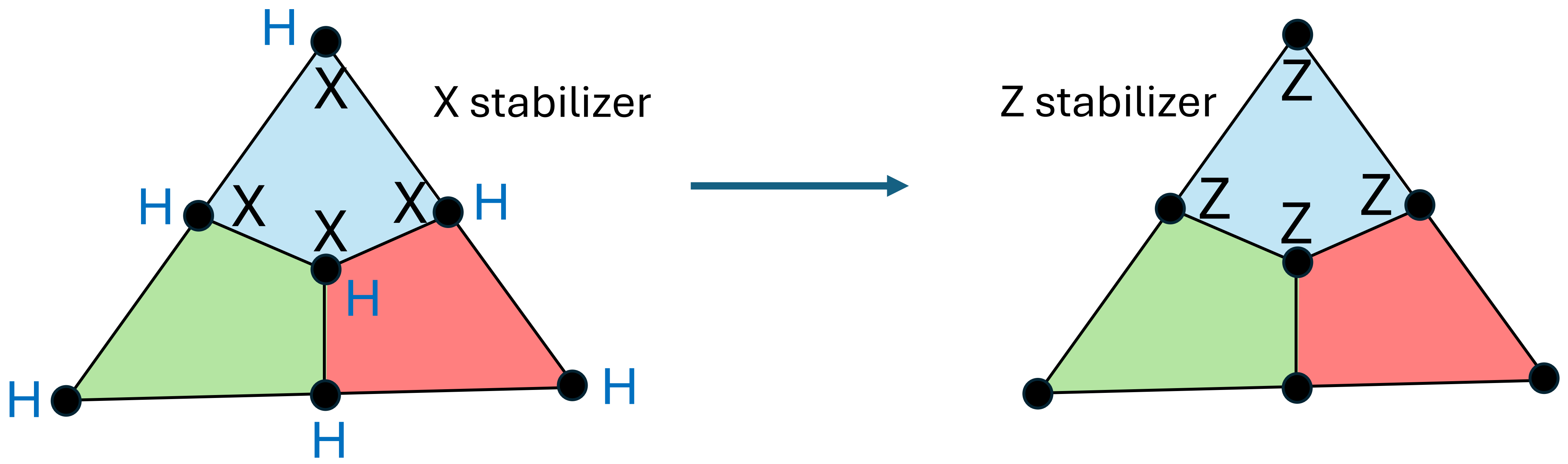
condition 1

Just condition 1 implies:
Suppose s2=1 and s1 =0, output can be s=1
Suppose s2=0 and s1 =1, output can be s=1 (NOT CORRECTING ERROR!)

Both conditions 1 and 2 together imply:
Suppose s2=1 and s1 =0, output can be s=1
Suppose s2=0 and s1 =1, output can be s=0 (CORRECTS ERROR)

FT logical gates

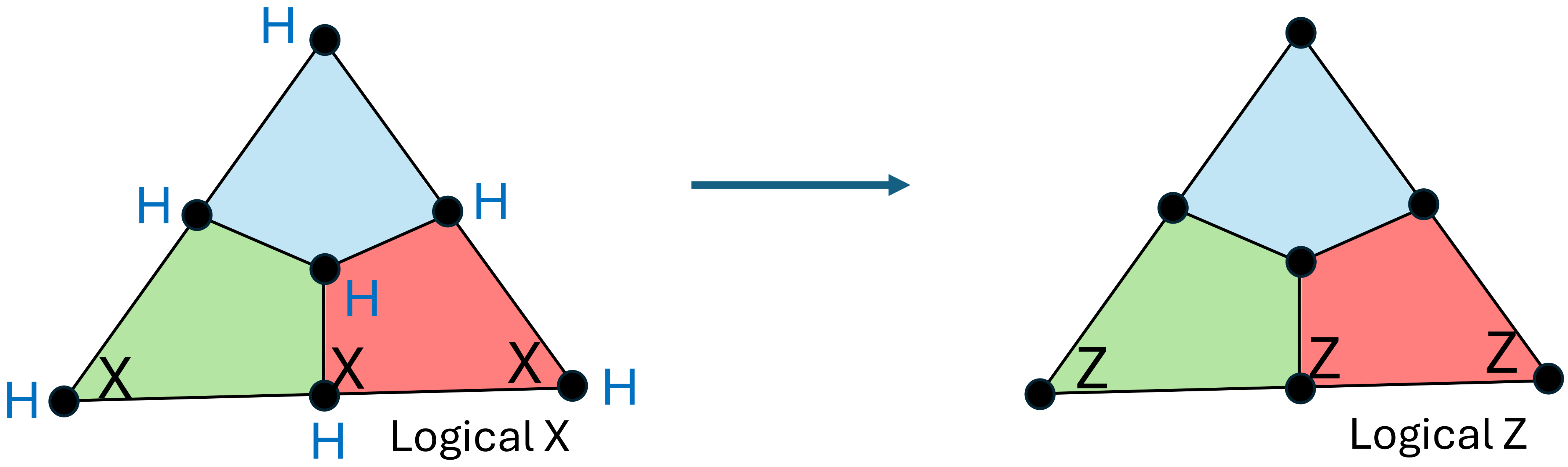
Transverse Hadamard



Applying a Hadamard to each qubit applies a logical Hadamard to logical qubit.
To see this, first note that Hadamard preserves the code space:

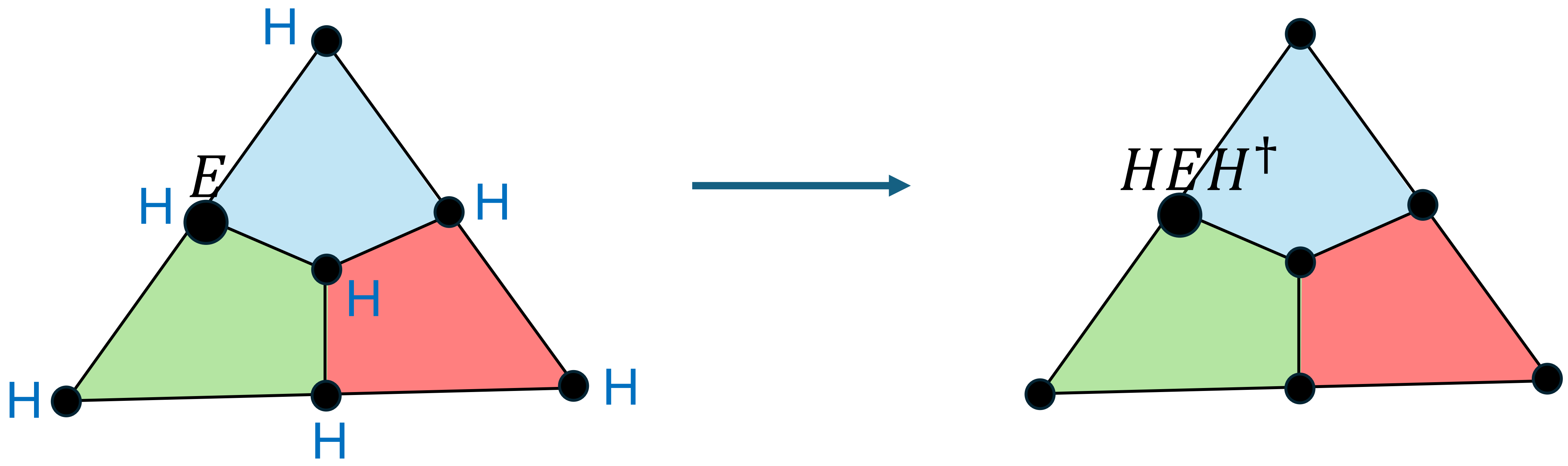
- Before the gate, a code state |ψ⟩ has S|ψ⟩ = |ψ⟩ for all stabilizers S.
- $H^{\otimes n}$|ψ⟩ is also in the code space, since:
- S($H^{\otimes n}$ |ψ⟩) = $H^{\otimes n}$($H^{\dagger\otimes n}$S $H^{\otimes n}$) |ψ⟩ = $H^{\otimes n}$ S'|ψ⟩ = ($H^{\otimes n}$ |ψ⟩).

Transverse Hadamard



Similarly, the action of $H^{\otimes n}$ on the logical qubit can be understood by considering how $H^{\otimes n}$ acts on the logical operators.

Transverse Hadamard



Transverse gates do not spread errors: a single-qubit error before the gate remains a single-qubit error after the gate.

$$H^{\otimes n}\text{: } E_j|\psi\rangle \to H^{\otimes n}E_j|\psi\rangle \;=\; (H^{\otimes n}E_jH^{\dagger\otimes n})H^{\otimes n}|\psi\rangle = (HEH^{\dagger})_j\, H^{\otimes n}\,|\psi\rangle$$

Other logical gates and operations

Not all logical gates can be implemented in this way, but in the Steane code, all ‘Clifford gates’ can.

The Clifford gates are not universal for quantum computation.

But there are also techniques for fault-tolerantly implementing non-Clifford gates, for example the T-gate.

When combined, the Clifford+T gate set is universal for quantum computation.

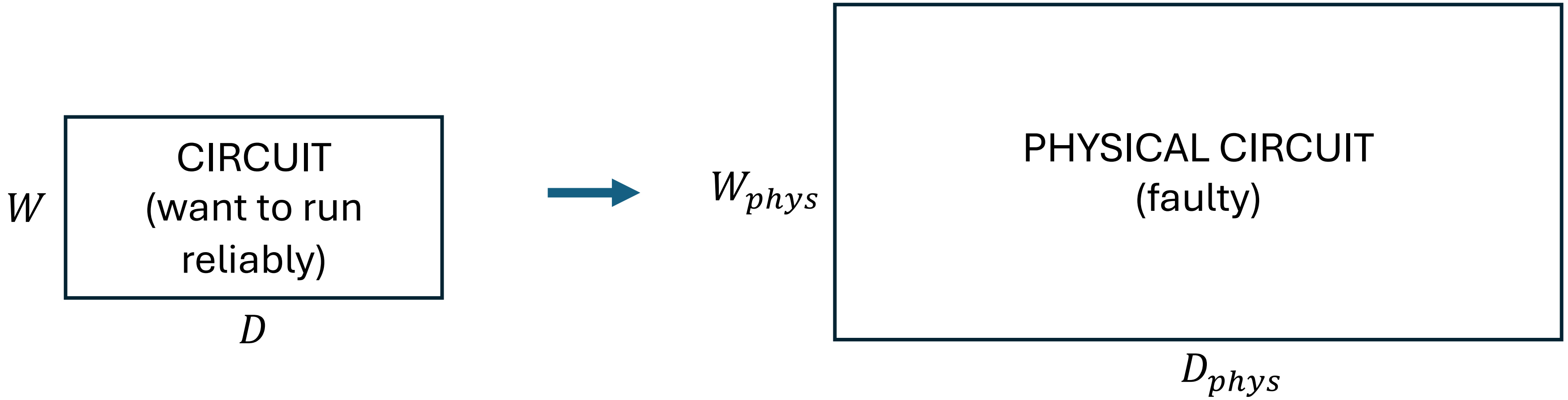
Threshold theorem

Threshold theorem (Aharanov & Ben-Or '99)

We have just seen parts of the argument, but hopefully the rough idea behind fault-tolerance is now a little clearer.

The crowning glory is the threshold theorem, which essentially states that in the noise model we have described, there is a threshold noise rate p_{th} (independent of W and D) such that:

if $\boldsymbol{p < p_{th}}$: $\boldsymbol{W_{phys}}$ and $\boldsymbol{D_{phys}}$ are only poly-log in $\boldsymbol{W}$ and $\boldsymbol{D}$.


W
CIRCUIT
(want to run reliably)
D
W_{phys}
PHYSICAL CIRCUIT
(faulty)
D_{phys}