Quantum Error Correction

Louis Golowich Wenjie Gong Ari Hatzimemos
Dylan Li Dylan Zhou

Physics 160
Harvard University

Final Project Presentation, 13 May 2020

Table of Contents

1. Introduction and Review of Quantum Error Correction
2. The 3-Qubit Codes
3. The Shor 9-Qubit Code
4. The 7-Qubit Code

Introduction

"To be an Error and to be Cast out is part of God's Design."

William Blake

- Noise as a longstanding problem in information processing systems
  - *e.g.*, classical computers, modems, CD players, etc.
  - Noise is still a problem in quantum information
- Key idea: to protect a message against noise, *encode* the message by adding redundant information; even if some information is corrupted, redundancy allows us to *decode* and recover the original message

Project Framework

- Goals:
  - to implement various quantum error-correcting codes
    - we chose the 3-qubit, 9-qubit, 7-qubit codes
  - to analyze and compare their performances
    - *when are they effective?*
    - *when should we use error-correcting codes?*
- Tools:
  - Python's Qiskit package
  - IBM's quantum machines

3-Qubit Codes: Classical Inspiration

Classical Error Correction

- Encoding by *repetition codes*:

$$0 \to 000$$
$$1 \to 111.$$

- Decoding by *majority voting*:

$$\textit{Ex.:}\ \ 001 \to 0.$$

- Analysis: Let p be the probability that a bit is flipped. This method fails when 2 or more bits are flipped, which occurs with probability $3p^2(1-p)+p^3$, so the probability of error is $p_e = 3p^2 - 2p^3$. Then this method is preferred when $p_e < p$, or $p < 1/2$.

Noisy Channels: The Bit Flip Channel

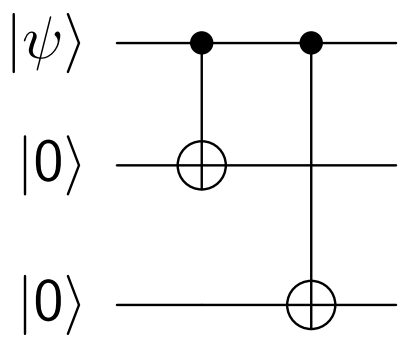
- One model for noise is the *bit flip channel* (analogous to classical channel).
- The bit flip channel flips qubits with probability p and leaves them untouched with probability $1 - p$.
- Equivalent to applying X gate with probability p.
- We protect qubits from this channel with the *bit flip code*.

3-Qubit Bit Flip Code: Encoding Logical Bits

- The goal is to correct bit flip errors.
- Encoding:

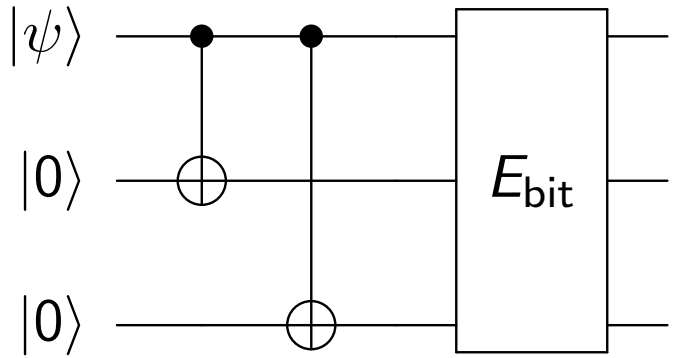
$$|0\rangle \rightarrow |0_L\rangle \equiv |000\rangle$$
$$|1\rangle \rightarrow |1_L\rangle \equiv |111\rangle\,.$$

- Encoding circuit for 3-qubit bit flip code:



3-Qubit Bit Flip Code: Detecting Errors

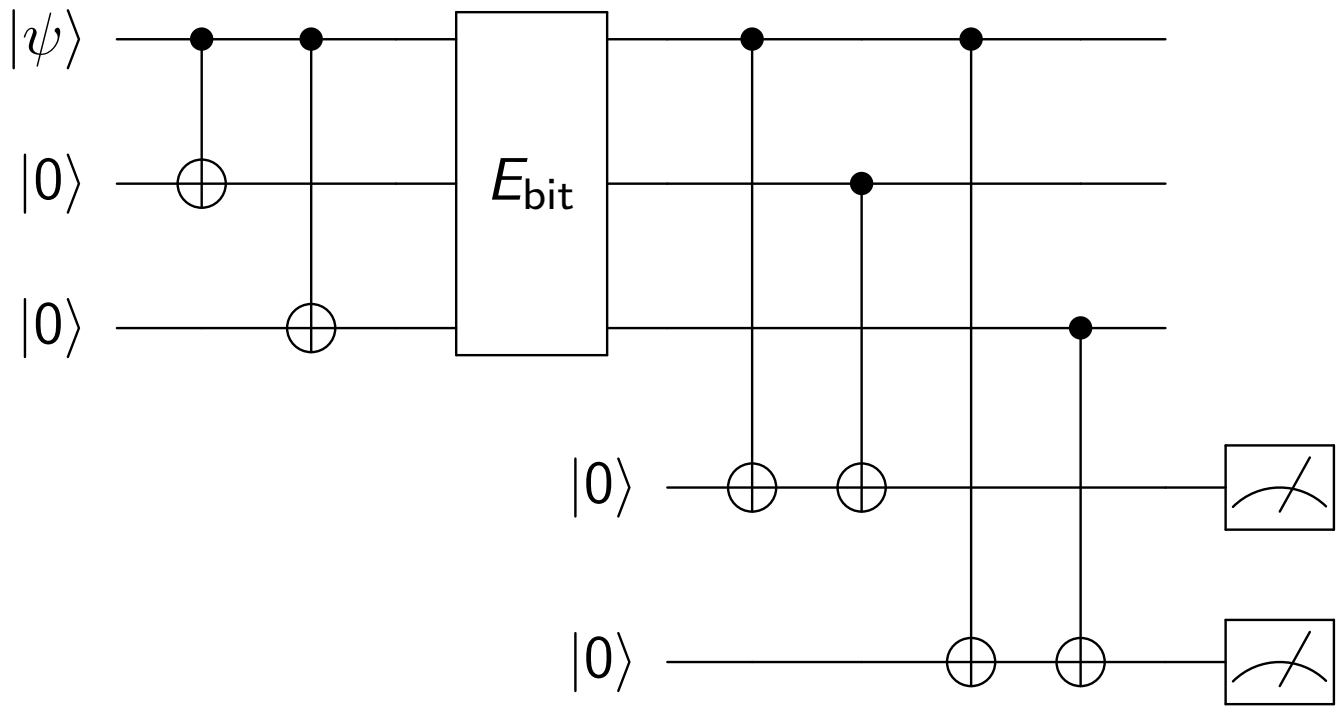
- Suppose there is a bit flip error after encoding:



- Error Detection (or *syndrome diagnosis*):
  - we would like to determine which, if any, of the qubits have been corrupted
  - four error syndromes: no error, bit flip on qubit one, bit flip on qubit two, bit flip on qubit three

3-Qubit Bit Flip Code: Detecting Errors

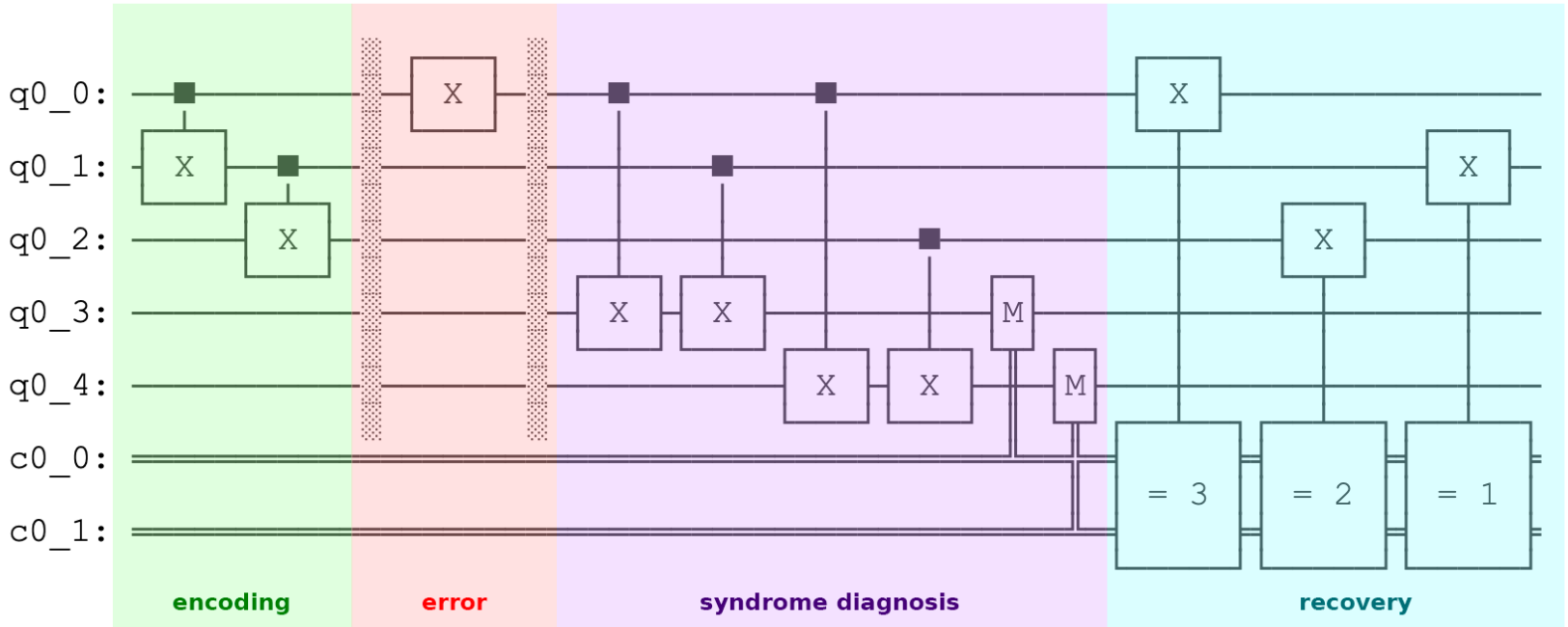
- We can diagnose the syndrome using two ancillary qubits:



- Based on measurement results, we know where the error occured.

3-Qubit Bit Flip Code: Correcting Errors

- Complete circuit for error correction (or *recovery*):

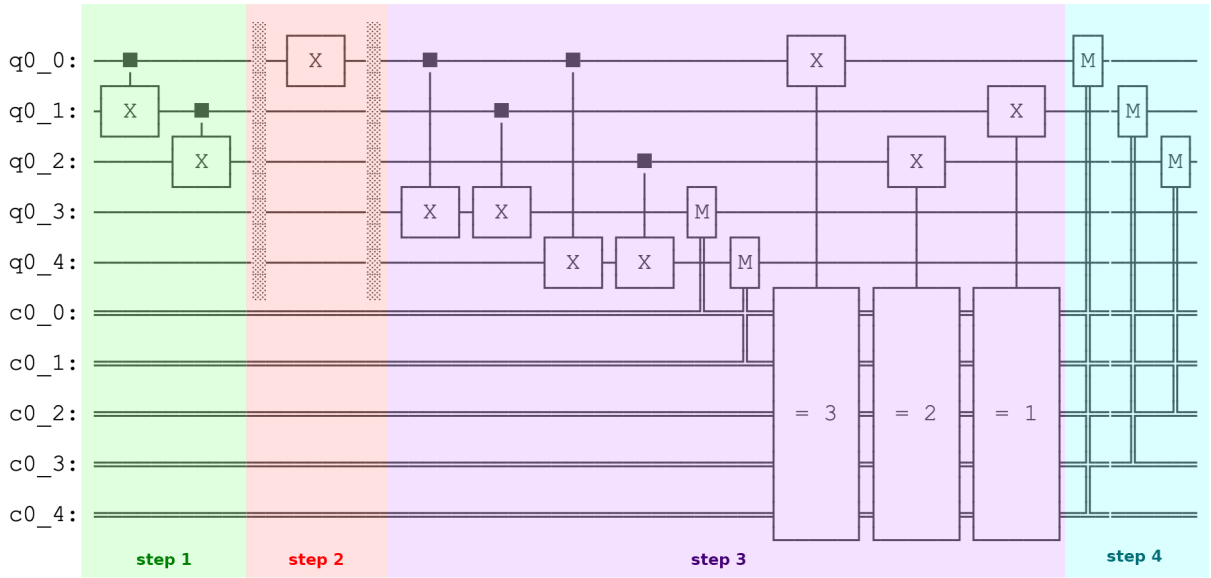


Analyzing the Bit Flip Code: Simulation

- Let's look at the performance of the 3-qubit bit flip code against bit flip channels of varying error probabilities p.
- Setup:
  1. encode a single qubit in state $|0\rangle$ into a logical state $|0_L\rangle = |000\rangle$
  2. create a bit flip channel which adds X gates with probability p
  3. run error correcting code
  4. measure final state

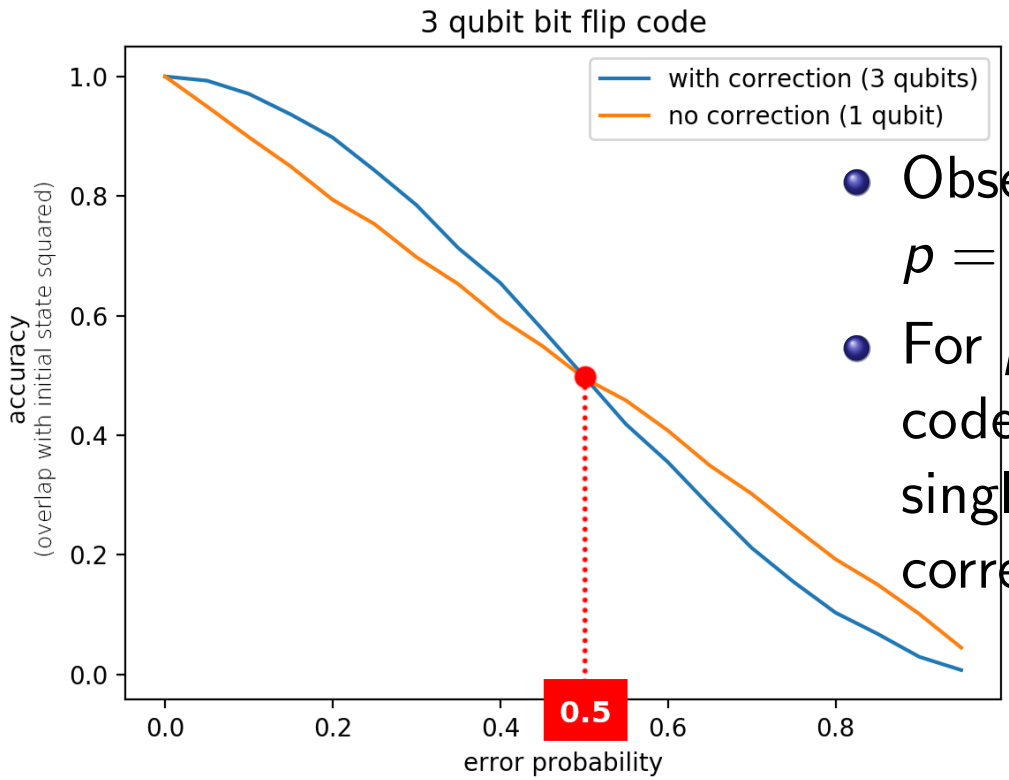
Analyzing the Bit Flip Code: Simulation

- Let's look at the performance of the 3-qubit bit flip code against bit flip channels of varying error probabilities p.
- Setup:
  1. encode a single qubit in state $|0\rangle$ into a logical state $|0_L\rangle = |000\rangle$
  2. create a bit flip channel which adds X gates with probability p
  3. run error correcting code
  4. measure final state



Analyzing the Bit Flip Code: Simulation

- Ran tests on Qiskit's simulator
- Probability p ranging from 0 to 1; 10000 trials for each p



- Observe crossover point at $p = 0.5$.
- For $p < 0.5$, error correcting code performs better than a single qubit with no correction.

copy.png

Noisy Channels: Phase Flip Channel

- Another quantum channel is the *phase flip* error model.
- With probability p the relative phase of states $|0\rangle$ and $|1\rangle$ is flipped, with probability $1 - p$ it is left alone.
- Equivalent to applying Z operator with probability p.
- We fight this channel with the *phase flip code*.

3-Qubit Phase Flip Code

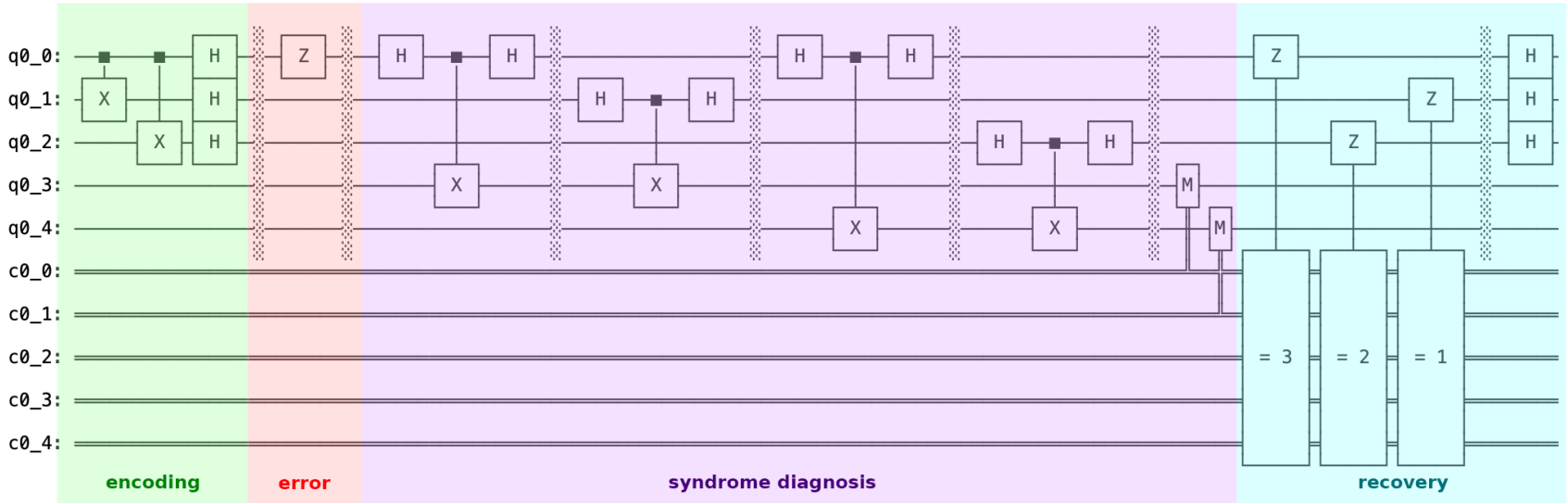
- No classical analog, but it is easy to turn the phase flip channel into a bit flip channel.
- Use x-basis for encoding:

$$|0\rangle \rightarrow |0_L\rangle \equiv |+++\rangle$$
$$|1\rangle \rightarrow |1_L\rangle \equiv |---\rangle\,.$$

- Phase flip Z acts as bit flip for this encoding!

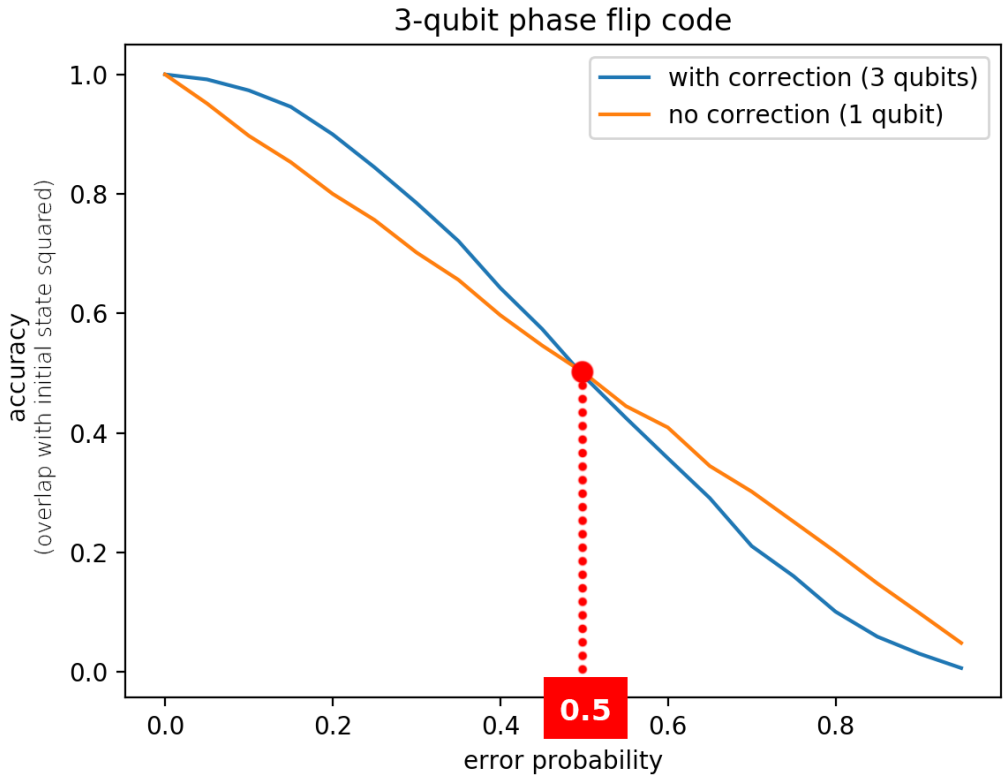
3-Qubit Phase Flip Code

Complete 3-qubit phase flip code:



Analyzing the Phase Flip Code: Simulation

- Ran tests on Qiskit's simulator
- Probability p ranging from 0 to 1; 10000 trials for each p



- Observe crossover point at $p = 0.5$.
- For $p < 0.5$, error correcting code performs better than a single qubit with no correction.
- Nearly identical result to the bit flip code.

3-Qubit Codes on IBM's Machines

Adjusting the circuit for IBM's machines (which do not have feed-forward capability)

The Shor 9-Qubit Code

- Can we protect against *arbitrary* errors?

The Shor 9-Qubit Code

- Can we protect against *arbitrary* errors?
- Yes! $\longrightarrow$ The *Shor code*

The Shor Code: Encoding

- By combining the 3-qubit phase flip and bit flip codes, the Shor code protects against arbitrary errors.

The Shor Code: Encoding

- By combining the 3-qubit phase flip and bit flip codes, the Shor code protects against arbitrary errors.
- First encode the qubit using the phase flip code:

$$|0\rangle \to |+++\rangle\,, \quad |1\rangle \to |---\rangle\,.$$

The Shor Code: Encoding

- By combining the 3-qubit phase flip and bit flip codes, the Shor code protects against arbitrary errors.
- First encode the qubit using the phase flip code:

$$|0\rangle \to |+++\rangle\,, \quad |1\rangle \to |---\rangle\,.$$

- Then encode each of those qubits with the bit flip code:

$$|+\rangle \to (|000\rangle + |111\rangle)/\sqrt{2}, \quad |-\rangle \to (|000\rangle - |111\rangle)/\sqrt{2}.$$

The Shor Code: Encoding

- By combining the 3-qubit phase flip and bit flip codes, the Shor code protects against arbitrary errors.
- First encode the qubit using the phase flip code:

$$|0\rangle \to |+++\rangle\,, \quad |1\rangle \to |---\rangle\,.$$

- Then encode each of those qubits with the bit flip code:

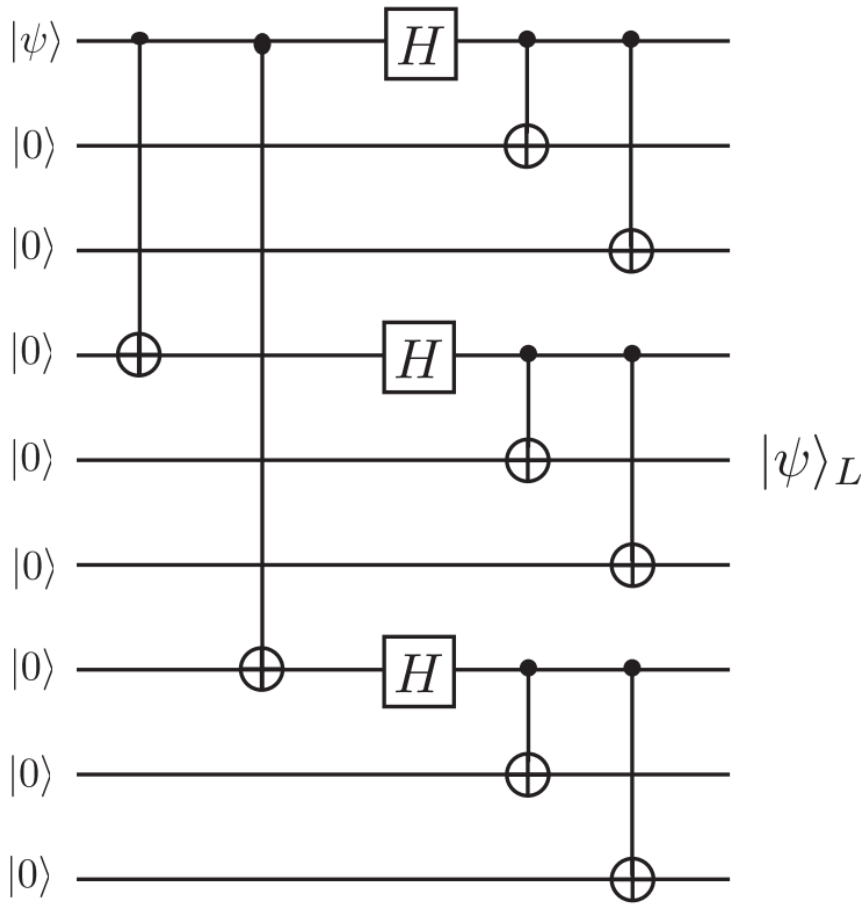
$$|+\rangle \to (|000\rangle + |111\rangle)/\sqrt{2}, \quad |-\rangle \to (|000\rangle - |111\rangle)/\sqrt{2}.$$

- The result is a 9-qubit code with codewords

$$|0\rangle \to |0_L\rangle \equiv \frac{(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)}{2\sqrt{2}}$$
$$|1\rangle \to |1_L\rangle \equiv \frac{(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)}{2\sqrt{2}}.$$

The Shor 9-Qubit Code: Encoding

Encoding circuit for 9-qubit code:



The Shor 9-Qubit Code: Correcting Errors

Bit Flip Error Correction

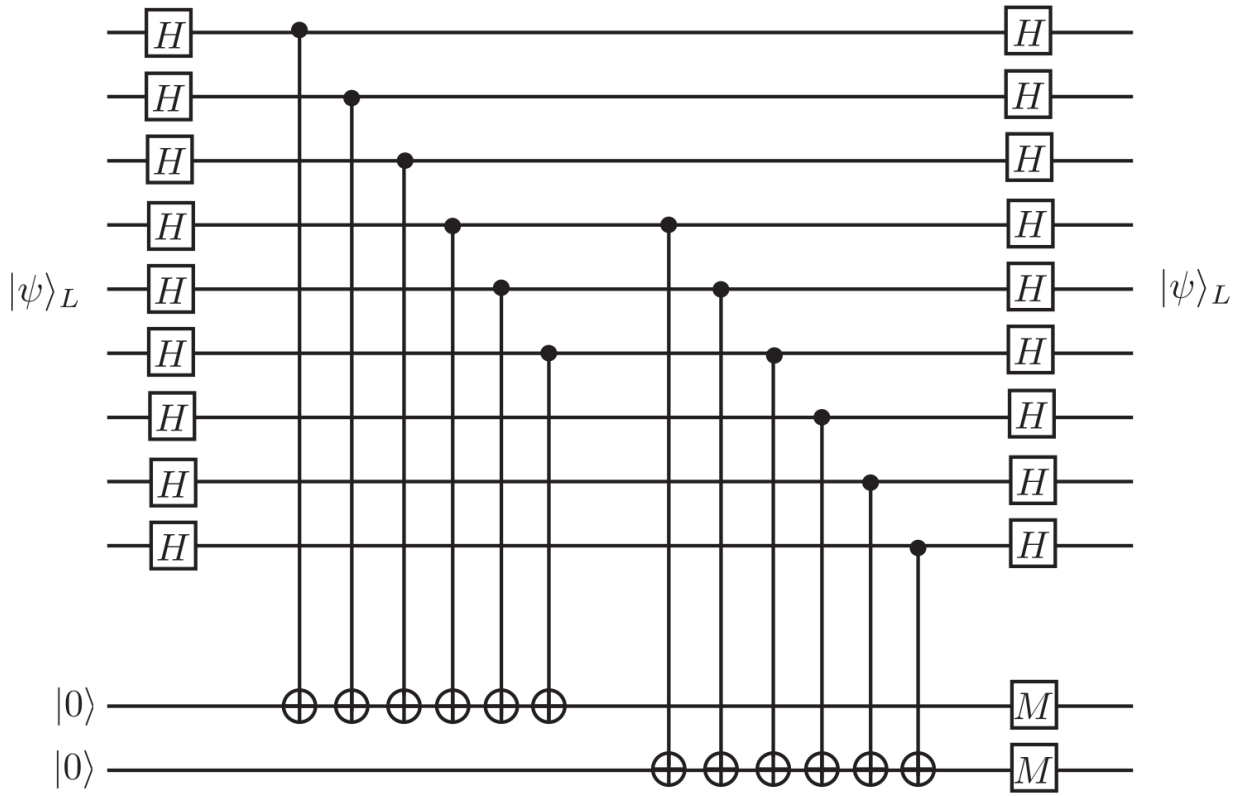
- On each block of three (i.e. qubits 0-2, 3-5, and 6-8), the 3-qubit circuit is utilized to correct for bit flips.

Phase Flip Error Correction

- The phase of the first two blocks of three (qubits 0-2 and 3-5) and the second two blocks of three (qubits 3-5 and 6-8) are compared to correct for phase flips.
- The phase correction necessitates two ancillary qubits. Thus, we need 8 ancilla: 6 for bit flip correction, and 2 for phase flip correction.

The Shor 9-Qubit Code: Correcting Phase Errors

- The phase correction circuit, shown below, converts the qubits from the x-basis to the z-basis and checks parity of each block of two.



The Shor 9-Qubit Code: Correcting Phase Errors

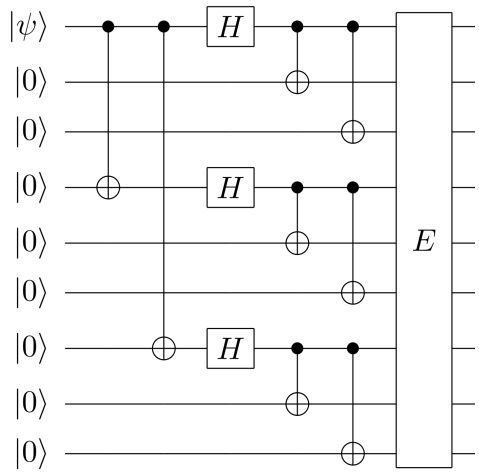
- The following corrections are performed depending on the measured ancilla for phase flip correction:

$$10 \rightarrow \sigma_z \text{ on block 1}$$
$$01 \rightarrow \sigma_z \text{ on block 3}$$
$$11 \rightarrow \sigma_z \text{ on block 2.}$$

- As two bit-flip errors occur with probability $3p^2$ per block for 3 blocks, and two phase-flip errors occur on separate blocks with probability $3p^2$, the total error $\sim 12p^2$. Thus, this error code is efficient for $12p^2 < p \implies p < 0.083$.

The Shor 9-Qubit Code: Error Correction Methodology

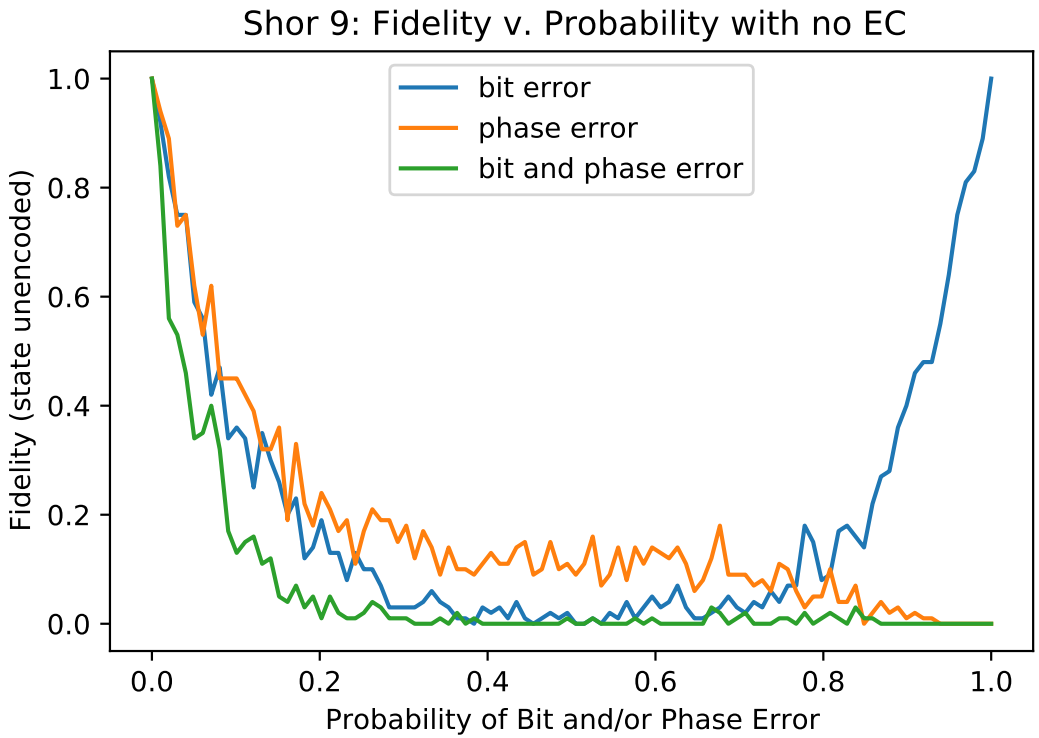
- We only consider error that occurs between the encoding step and the correcting step, thus simulating a memory error.



- Specifically, we consider a complete phase flip and/or bit flip (i.e. X or Z) that occurs independently on each of the 9 physical qubits with probability p.
- After the error, we measure the ancilla and apply the appropriate error correcting steps. Finally, we run the encoding circuit in reverse and measure the output to determine fidelity.

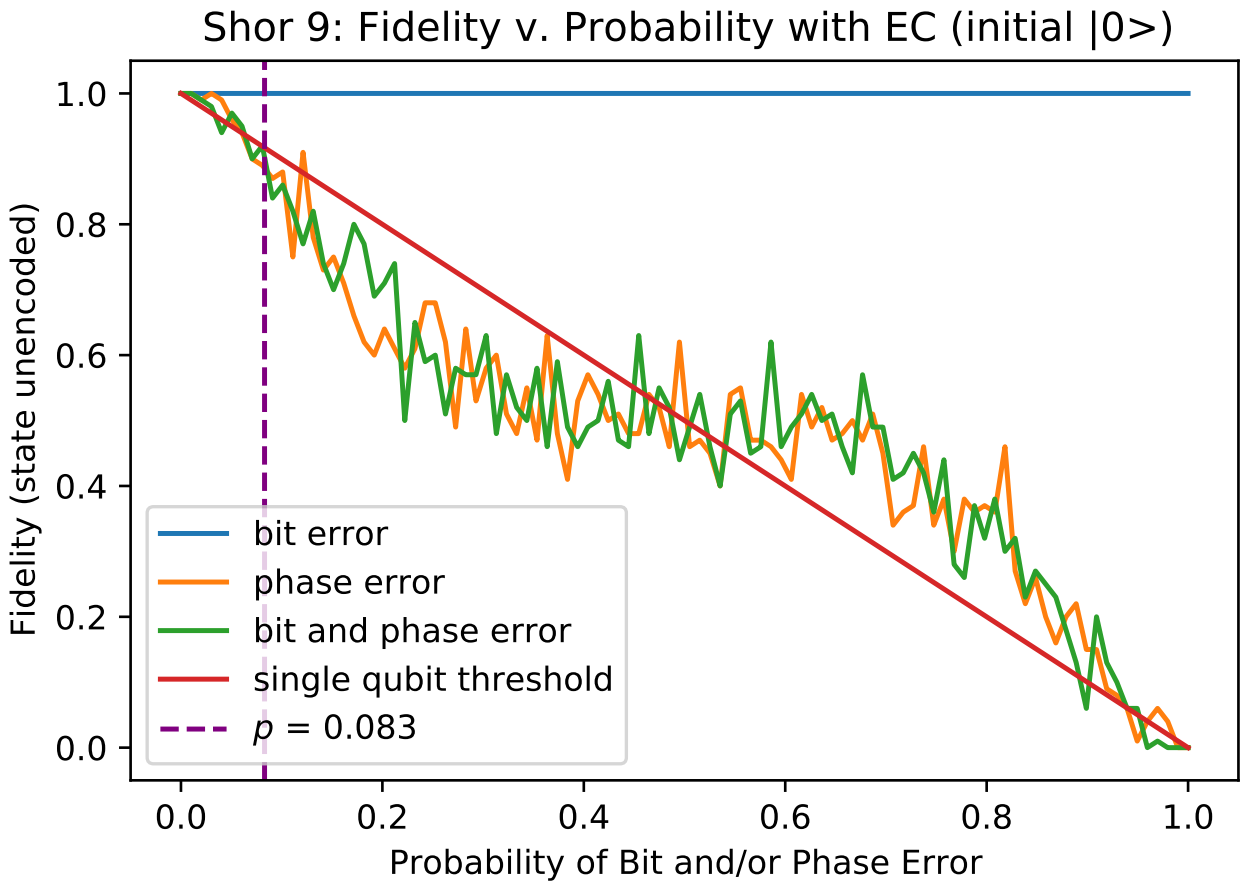
The Shor 9-Qubit Code: Simulation Performance with No Error Correction

- Initial state:
  $|0\rangle \rightarrow |0_L\rangle \equiv \frac{1}{\sqrt{8}}(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)$.
- Fidelity of un-encoded state measured against $|000000000\rangle$.



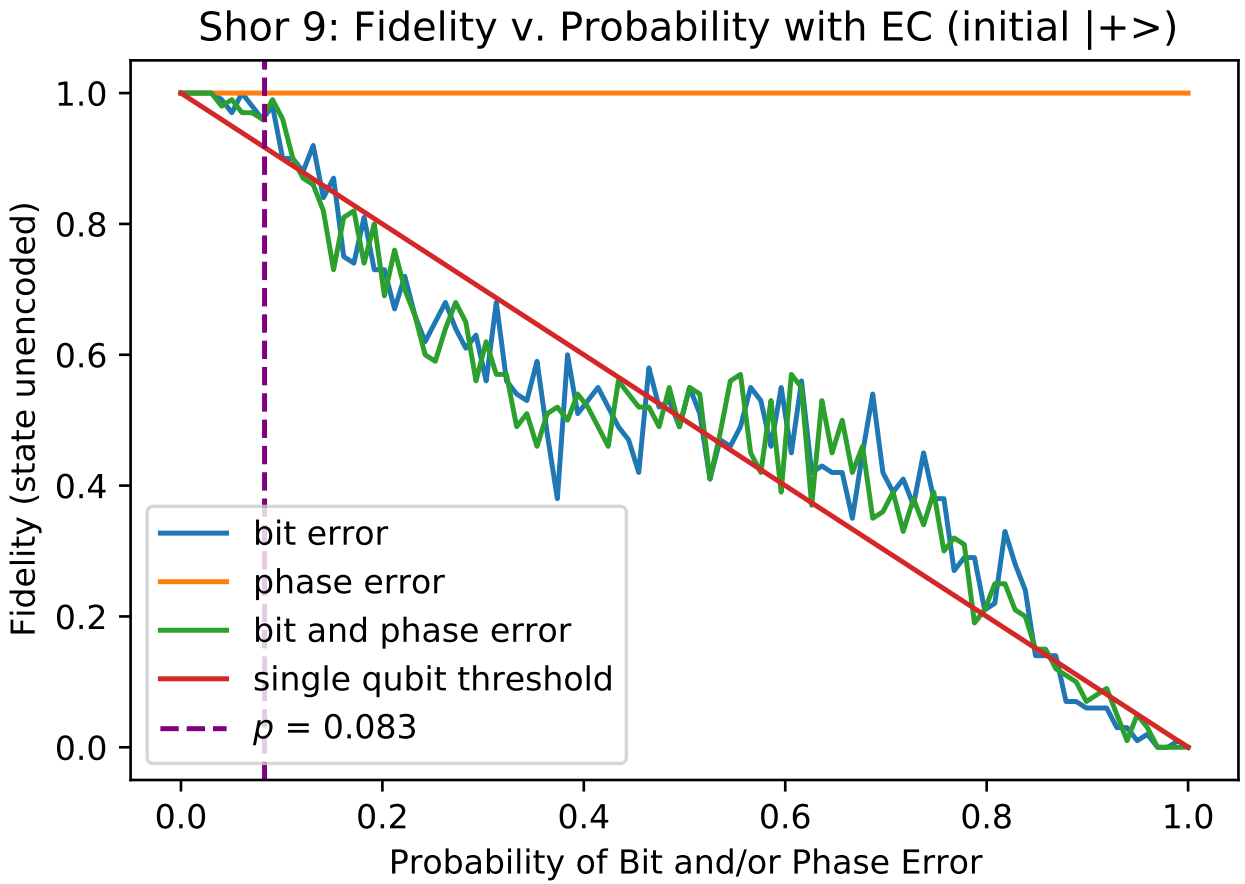
The Shor 9-Qubit Code: Simulation Performance with Error Correction

- Initial state: $|0\rangle \rightarrow |0_L\rangle$
- Fidelity of un-encoded state measured against $|000000000\rangle$.



The Shor 9-Qubit Code: Simulation Performance with Error Correction

- Initial state: $|+\rangle \rightarrow \frac{1}{\sqrt{2}}(|0_L\rangle + |1_L\rangle)$
- Fidelity of un-encoded state measured against $|000000000\rangle$.



7-Qubit Code

Encodes 1 logical qubit using 7 physical qubits:

$$|0_L\rangle = \frac{|0000000\rangle + |1010101\rangle + |0110011\rangle + |1100110\rangle + |0001111\rangle + |1011010\rangle + |0111100\rangle + |1101001\rangle}{\sqrt{8}}$$

$$|1_L\rangle = \frac{|1111111\rangle + |0101010\rangle + |1001100\rangle + |0011001\rangle + |1110000\rangle + |0100101\rangle + |1000011\rangle + |0010110\rangle}{\sqrt{8}}$$

$$H^{\otimes 7}|0_L\rangle = \frac{|0_L\rangle + |1_L\rangle}{\sqrt{2}}$$

$$H^{\otimes 7}|1_L\rangle = \frac{|0_L\rangle - |1_L\rangle}{\sqrt{2}}$$

7-Qubit Code

$$|0_L\rangle = \frac{|0000000\rangle + |1010101\rangle + |0110011\rangle + |1100110\rangle + |0001111\rangle + |1011010\rangle + |0111100\rangle + |1101001\rangle}{\sqrt{8}}$$
$$|1_L\rangle = \frac{|1111111\rangle + |0101010\rangle + |1001100\rangle + |0011001\rangle + |1110000\rangle + |0100101\rangle + |1000011\rangle + |0010110\rangle}{\sqrt{8}}$$

$$H^{\otimes 7}|0_L\rangle = \frac{|0_L\rangle + |1_L\rangle}{\sqrt{2}}$$
$$H^{\otimes 7}|1_L\rangle = \frac{|0_L\rangle - |1_L\rangle}{\sqrt{2}}$$

- Of the 16 bit strings above, any two differ by ≥ 3 bits
- Intuition: therefore a single bit flip can be recovered
  - X error flips bit in $|0_L\rangle, |1_L\rangle$
  - Z error flips bit in $H^{\otimes 7}|0_L\rangle, H^{\otimes 7}|1_L\rangle$

Intuition: Stabilizers

7-qubit code is stabilized by the operators:

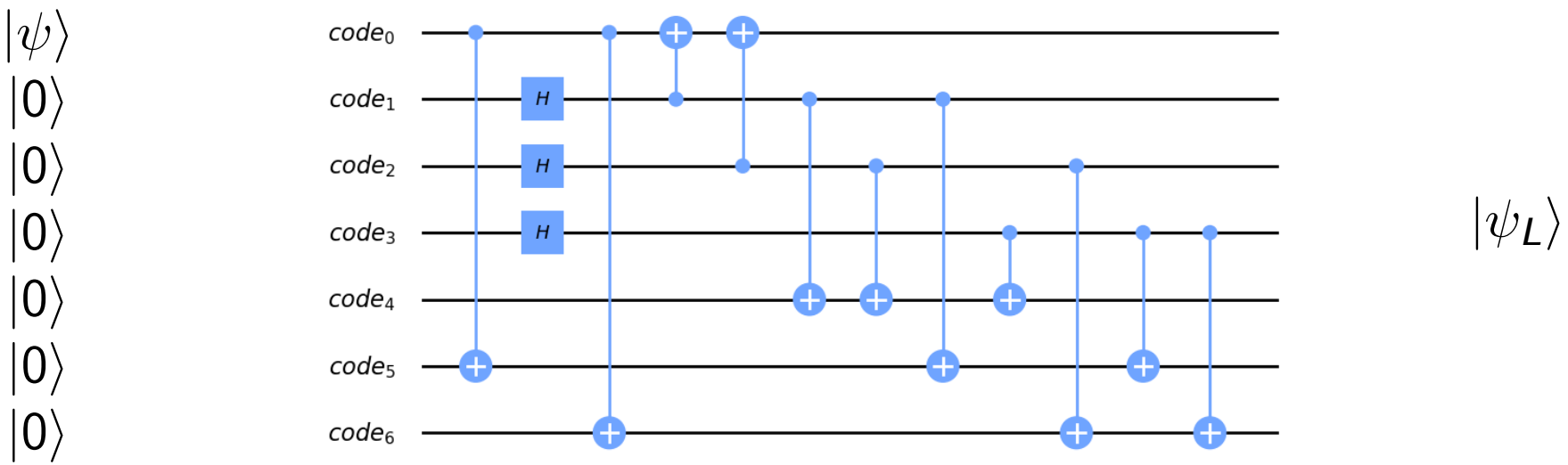
$$IIIXXXX$$
$$IXXIIXX$$
$$XIXIXIX$$
$$IIIZZZZ$$
$$IZZIIZZ$$
$$ZIZIZIZ$$

Bit flip or phase flip takes $|0_L\rangle, |1_L\rangle$ outside of subspace stabilized by above operators, which can then be detected with ancilla qubits.

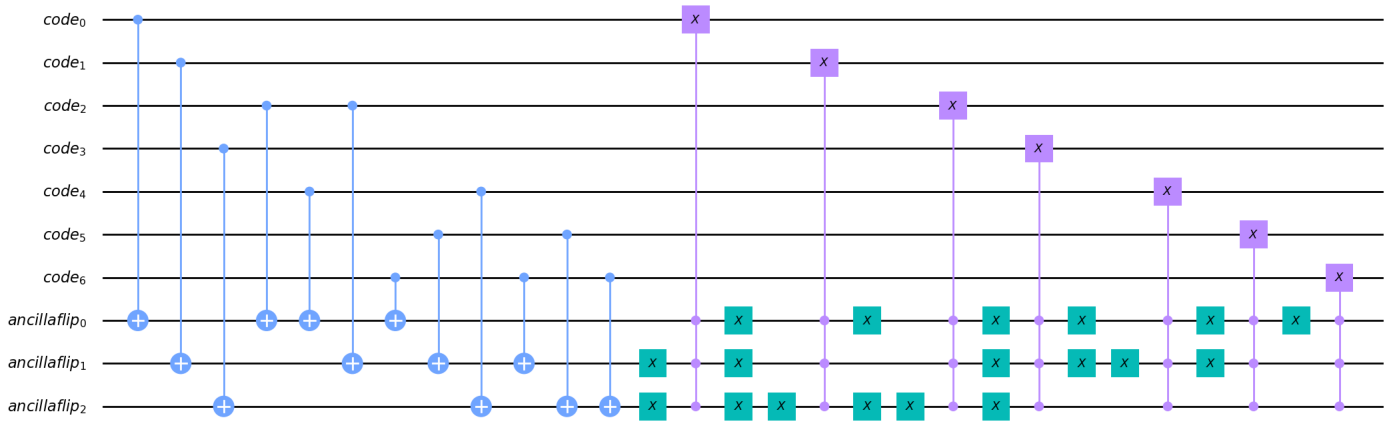
7-qubit code: Encoding

$$|0_L\rangle = \frac{|0000000\rangle + |1010101\rangle + |0110011\rangle + |1100110\rangle + |0001111\rangle + |1011010\rangle + |0111100\rangle + |1101001\rangle}{\sqrt{8}}$$

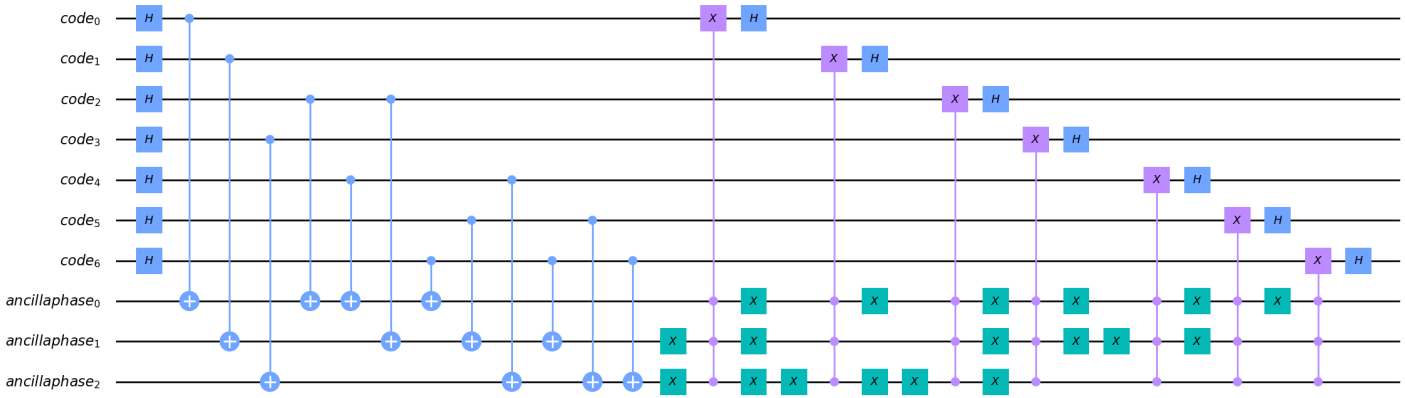
$$|1_L\rangle = \frac{|1111111\rangle + |0101010\rangle + |1001100\rangle + |0011001\rangle + |1110000\rangle + |0100101\rangle + |1000011\rangle + |0010110\rangle}{\sqrt{8}}$$



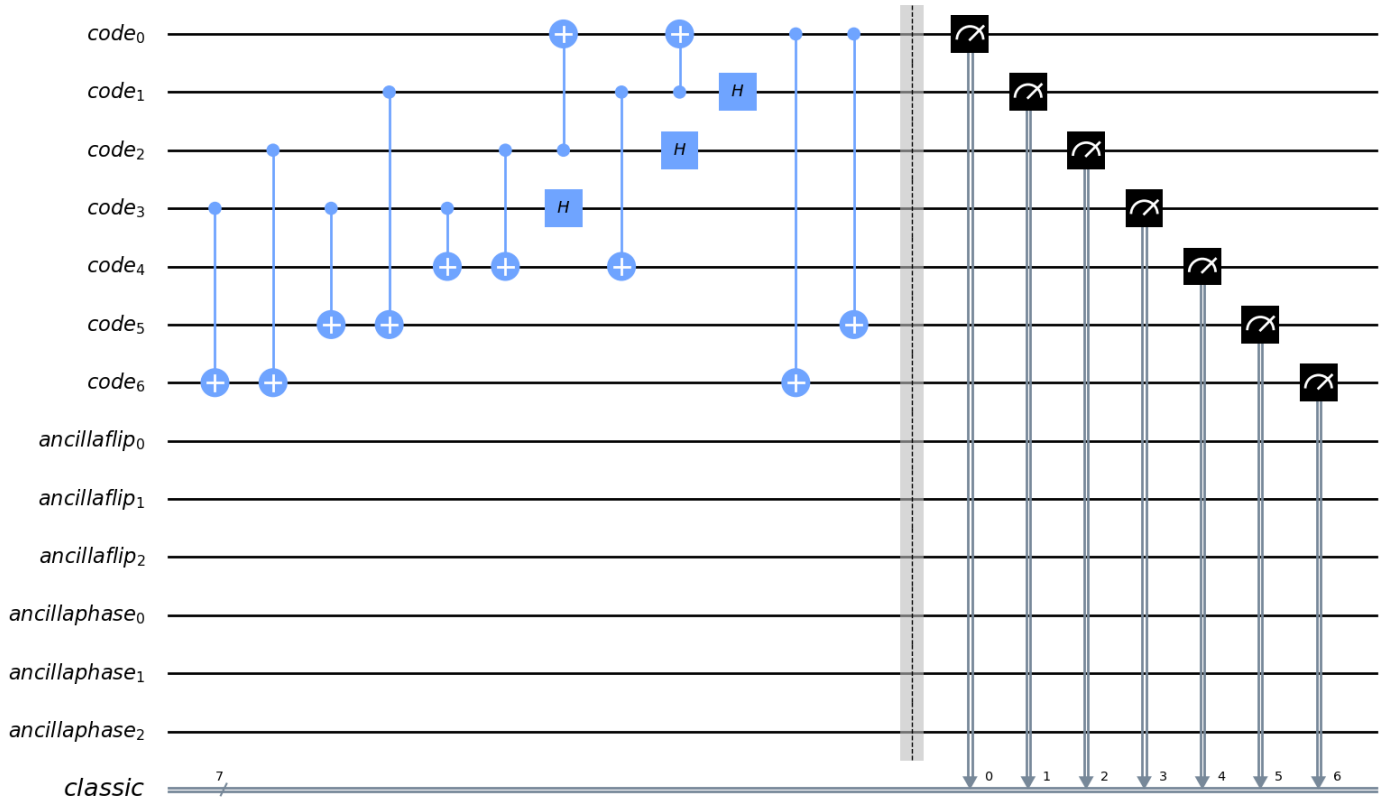
7-qubit code: Flip correction



7-qubit code: Phase correction



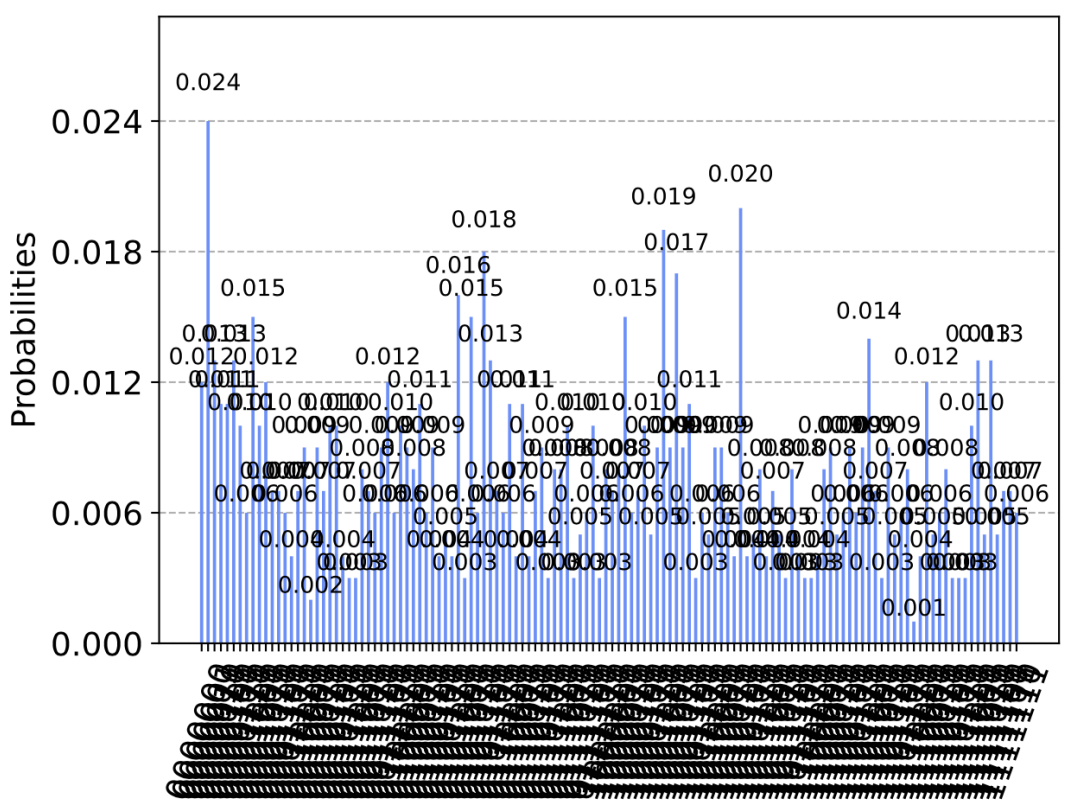
7-qubit code: Decoding and measurement



7-qubit code: Simulation vs running on quantum computers?

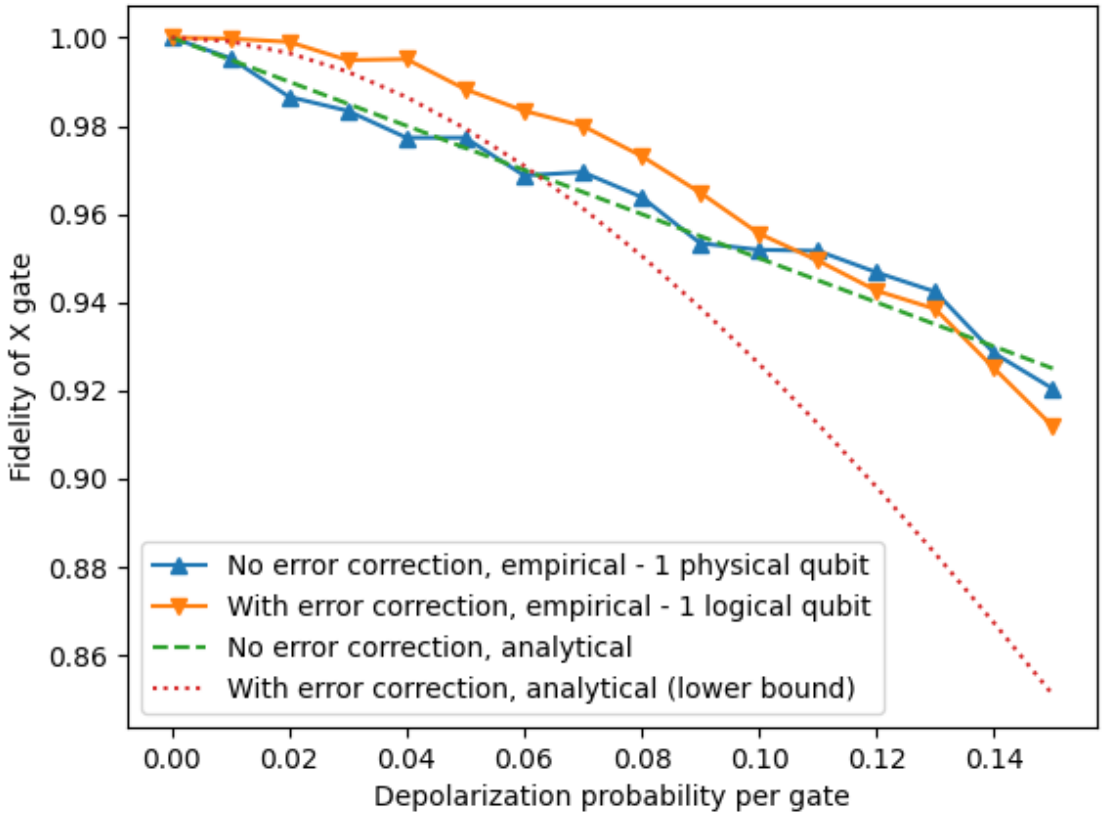
- The states should be clearly defined, but noise dominates the system

```
[ ]  # using optimization_level=3
     counts = job.result().get_counts(projection_circuit)
     plot_histogram(counts)
```



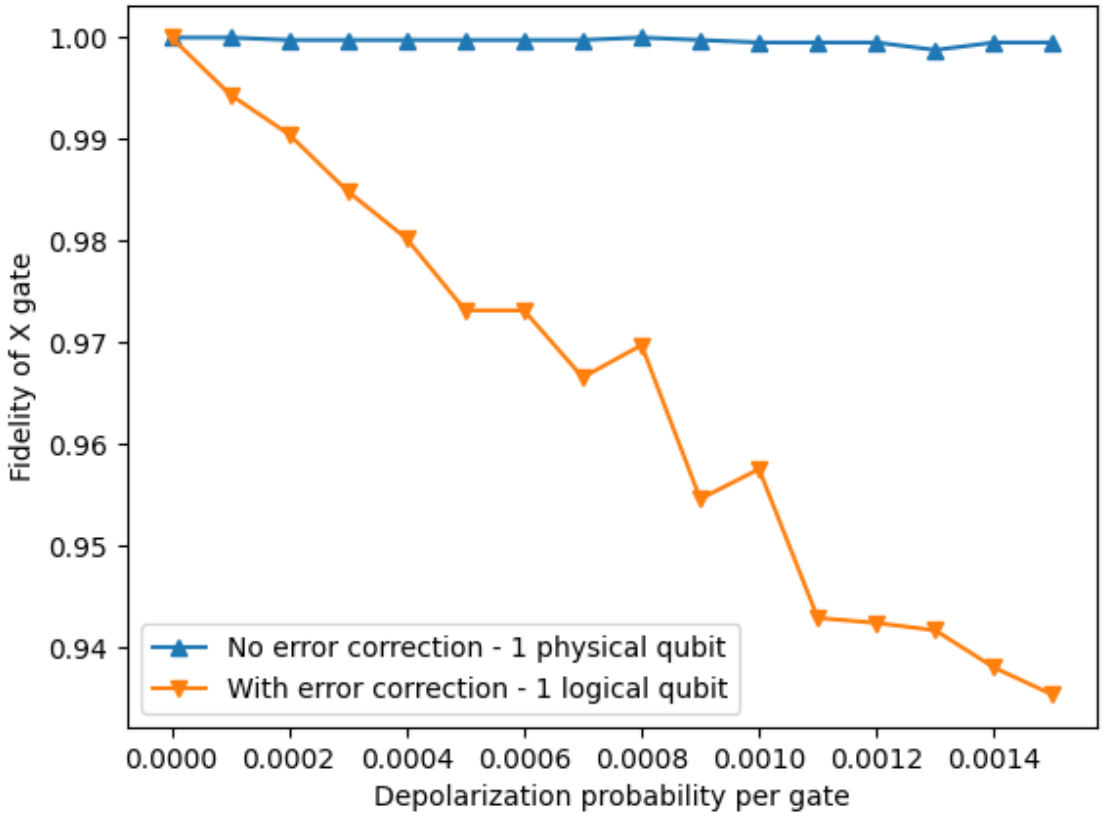
7-qubit code: Fidelity of X Gate under Depolarization

Encode $\|0_L\rangle$	X	Correct flip	Correct phase	Decode, measure



7-qubit code: Fidelity of X Gate under Depolarization

Encode $\|0_L\rangle$	X	Correct flip	Correct phase	Decode, measure



7-qubit code: Interpretation of fidelities

- With no depolarization in correction gates, error correction improves fidelity for depolarization probability $\lessapprox .12$
- With depolarization in all gates, error correction never improves fidelity

7-qubit code: Interpretation of fidelities

- With no depolarization in correction gates, error correction improves fidelity for depolarization probability $\lessapprox .12$
- With depolarization in all gates, error correction never improves fidelity
- 7-qubit code reduces single gate error rate for depolarization rates lower than detectable by our qiskit simulations ($p \lessapprox 10^{-4}$)

7-qubit code: Extending to Multiple Gates

- Here, we assumed that the $|0\rangle_L$ state was perfectly encoded, before performing up to 200 pairs of X gates on the seven qubits simultaneously

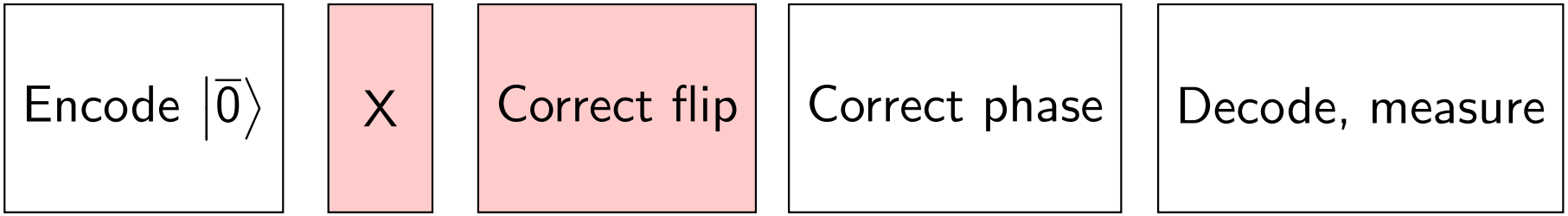
7-qubit code: Extending to Multiple Gates

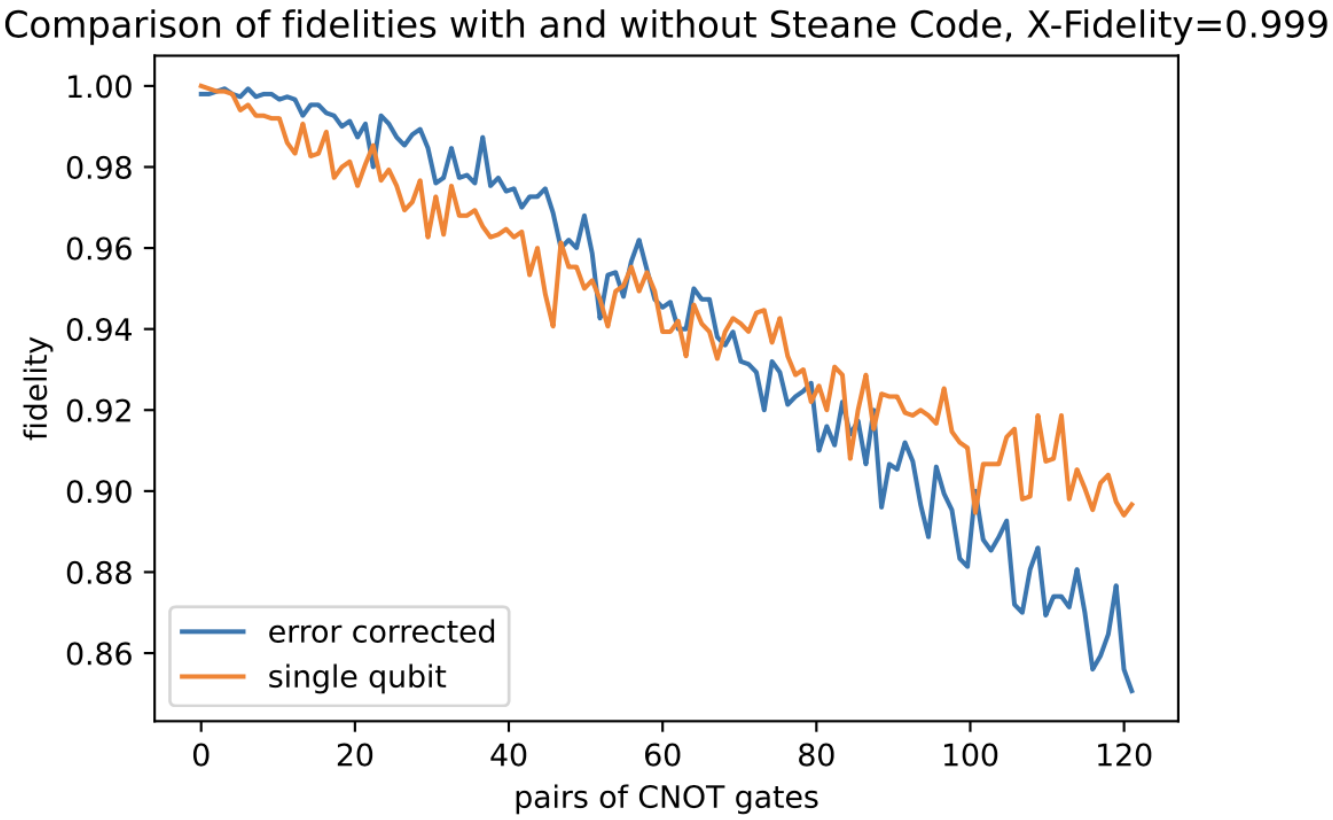
- Here, we assumed that the $|0\rangle_L$ state was perfectly encoded, before performing up to 200 pairs of X gates on the seven qubits simultaneously
- In the end, we decoded and we measured the value of the first of the seven qubits to measure its fidelity with the predicted state -¿ we should end up with $|0\rangle^{\otimes 7}$

7-qubit code: Extending to Multiple Gates

- Here, we assumed that the $|0\rangle_L$ state was perfectly encoded, before performing up to 200 pairs of X gates on the seven qubits simultaneously
- In the end, we decoded and we measured the value of the first of the seven qubits to measure its fidelity with the predicted state -¿ we should end up with $|0\rangle^{\otimes 7}$
- We compared the fidelity of the error correcting algorithm to a single qubit that goes through up to 200 X rotations

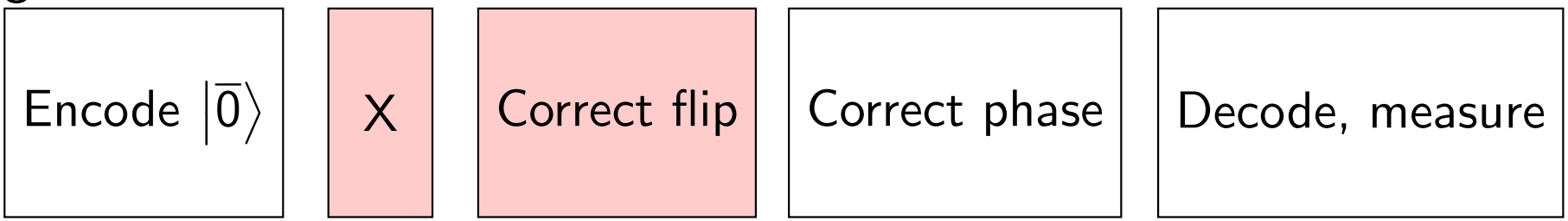
7-qubit code: Useful with lower error probabilities

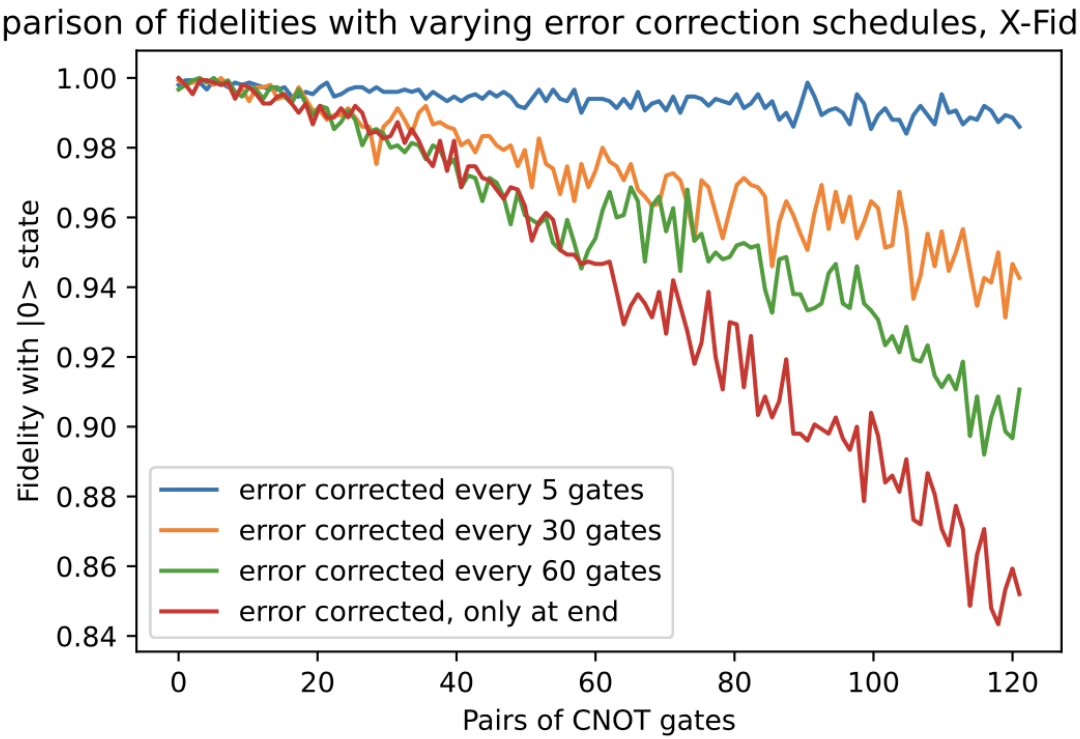




7-qubit code: Adding Error Correction at different Timesteps

- Using a noise model with (p=0.001) noise in the x and z-direction per gate

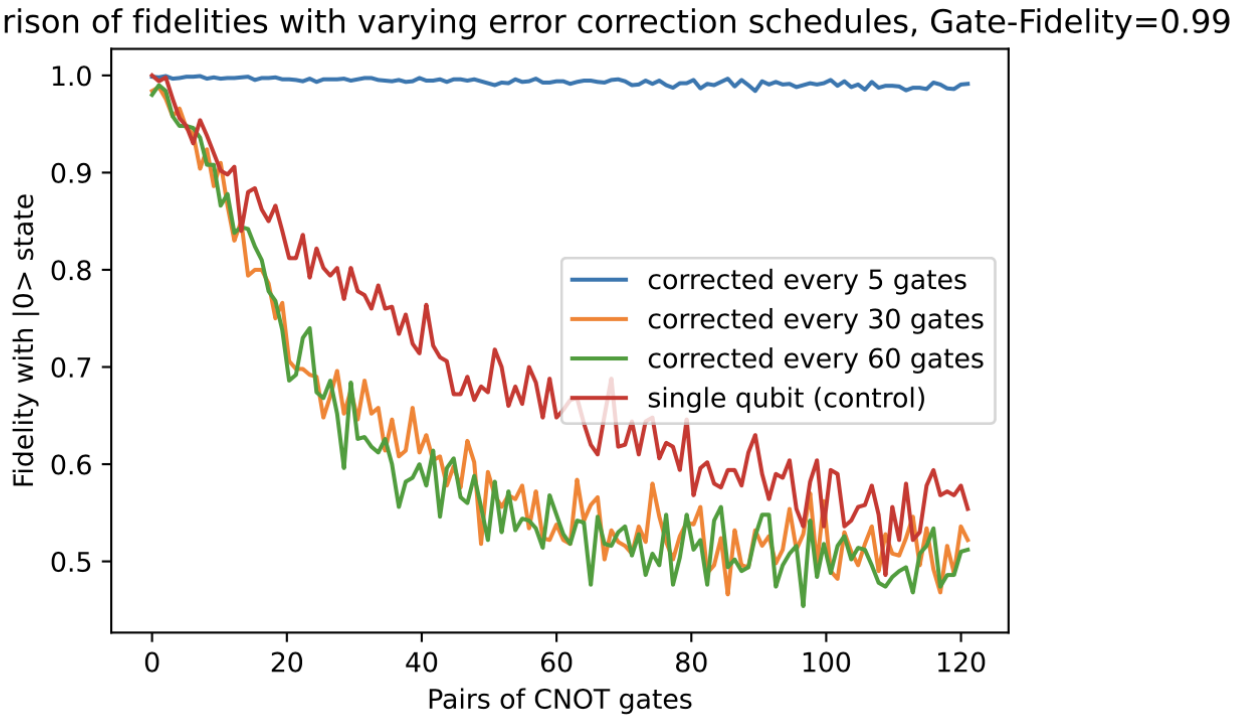




7-qubit code: Adding Error Correction at different Timesteps

Encode $\|\overline{0}\rangle$	X	Correct flip	Correct phase	Decode, measure

- Using a noise model with (p=0.01) noise in the x and z-direction per gate



7-qubit code: Generalizing to Error Correction

- Ultimately, error correction is useful, but must be intelligently deployed - here, error-correction is heavily sensitive to the noise in each gate
- It may be useful to characterize the thresholds before using error-correcting codes or consider using error-correcting codes continuously throughout a computation