Quantum error correction - A practical introduction

Introduction

- Quantum error correction is **advancing extremely fast** right now
- First quantum computers emerged in 2016-2019
- Good **QLDPCs** were discovered in **2022**
- A “good” General **logical operations** framework for QLDPCs was discovered in **2025/6**

Introduction

- My **name** is Bruno
  Contact **me anytime**: bruno.coelhocoutinho@dlr.de
  all will be here the **entire week**

- I'm a **scientist** at the DLR (German Aerospace agency)

- **Mostly** work on **quantum error correction**

  - Some work on quantum **communications** and quantum **computation**

- Part of my work involves **PhD** and **master student** supervision

  - I'm looking for one more good master student this semester

Class 1

- The problem of quantum error correction
- Historical codes
- Stabilizer formalism

- Bibliography
  - Nielsen & Chuang, Quantum Computation and Quantum Information (Cambridge, 2010)
  - Devitt, Munro & Nemoto, "Quantum error correction for beginners," Rep. Prog. Phys. 76, 076001 (2013)

Error correction

- Quantum technologies are **particularly** susceptible to **noise**
  $|0\rangle \otimes |0\rangle \otimes |0\rangle \rightarrow_e |1\rangle \otimes |0\rangle \otimes |0\rangle$
- Error correction allows us to **detect** and **correct errors** in a state
- But, unlike in the classical case
  - **We can't measure** the **state**
  - Needs to correct **two types** of **errors**: bit flips and phase flips
    $\sigma_x |0\rangle \rightarrow |1\rangle, \sigma_x |1\rangle \rightarrow |0\rangle$
    - $\sigma_z |0\rangle \rightarrow |0\rangle, \sigma_z |1\rangle \rightarrow -|1\rangle$

$$\sigma_0 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad \sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

Classical codes - Three bit flip code

- We make **quantum** states **resilient against noise** the same way we make **classical** states resilient to noise
  - We **add redundancy**
- The **three-bit flip code** is the first quantum error correction code proposed by Asher Peres in 1985

PHYSICAL REVIEW A VOLUME 32, NUMBER 6 DECEMBER 1985

Reversible logic and quantum computers

Asher Peres
Department of Physics, Technion—Israel Institute of Technology, 32000 Haifa, Israel
(Received 27 March 1985)

This article is concerned with the construction of a quantum-mechanical Hamiltonian describing a computer. This Hamiltonian generates a dynamical evolution which mimics a sequence of elementary logical steps. This can be achieved if each logical step is *locally* reversible (*global* reversibility is insufficient). Computational errors due to noise can be corrected by means of redundancy. In particular, reversible error-correcting codes can be embedded in the Hamiltonian itself. An estimate is given for the minimum amount of entropy which must be dissipated at a given noise level and tolerated error rate.

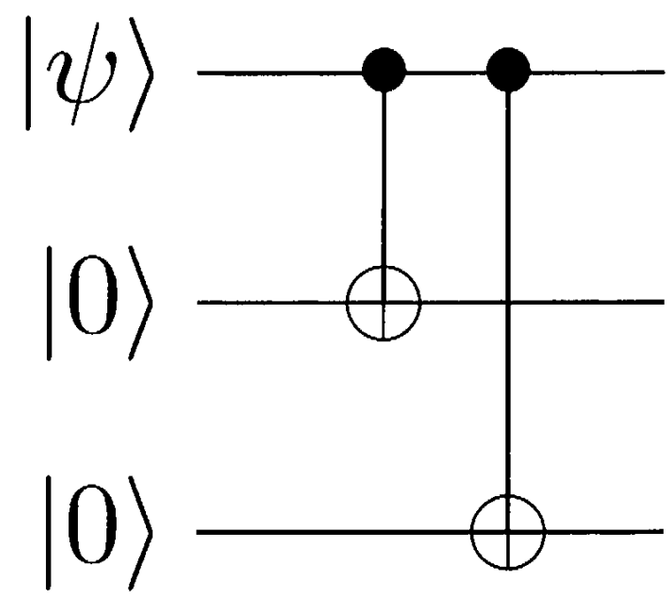
- It only protects against **bit-flips, X error**
  - But shows that quantum error correction is possible

Three-qubit bit-flip code

- First we need to encode a logical state $|\psi_L\rangle = \alpha|0_L\rangle + \beta|1_L\rangle$ into multiple physical states

$$|0_L\rangle := |000\rangle$$

$$|1_L\rangle := |111\rangle$$



- Modern quantum error correction does not use this definition of logical states, or encode states like this

Three-qubit bit-flip code

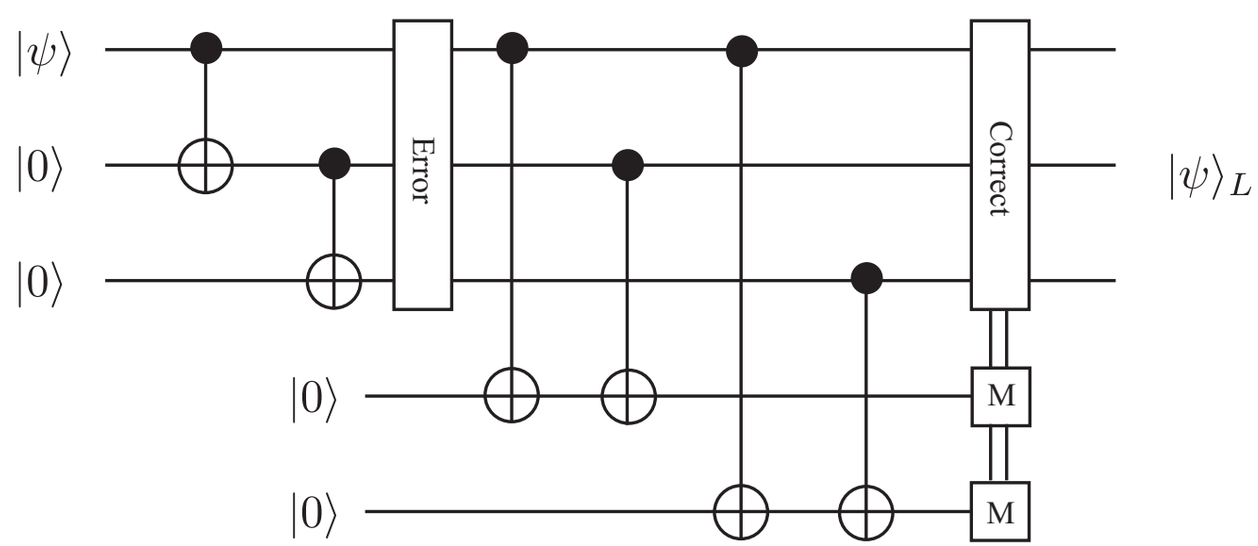
$$|\psi_L\rangle = \alpha|000\rangle + \beta|111\rangle$$

- Let us look at the effect of different bit-flip
- $\sigma_x^1|\psi\rangle = \alpha|100\rangle + \beta|011\rangle$
- $\sigma_x^2|\psi\rangle = \alpha|010\rangle + \beta|101\rangle$
- $\sigma_x^3|\psi\rangle = \alpha|001\rangle + \beta|110\rangle$

Three-qubit bit-flip code

$$|\psi\rangle_L = \alpha|000\rangle + \beta|111\rangle$$

$$P_0 \equiv |000\rangle\langle 000| + |111\rangle\langle 111| \text{ no error}$$
$$P_1 \equiv |100\rangle\langle 100| + |011\rangle\langle 011| \text{ bit flip on qubit one}$$
$$P_2 \equiv |010\rangle\langle 010| + |101\rangle\langle 101| \text{ bit flip on qubit two}$$
$$P_3 \equiv |001\rangle\langle 001| + |110\rangle\langle 110| \text{ bit flip on qubit three.}$$



Three-qubit bit code

- Now let us assume we have two bit-flip errors on different qubits? What happens?

$P_0 \equiv |000\rangle\langle 000| + |111\rangle\langle 111|$ no error

$P_1 \equiv |100\rangle\langle 100| + |011\rangle\langle 011|$ bit flip on qubit one

$P_2 \equiv |010\rangle\langle 010| + |101\rangle\langle 101|$ bit flip on qubit two

$P_3 \equiv |001\rangle\langle 001| + |110\rangle\langle 110|$ bit flip on qubit three.

Error Location	Final State, $\lvert\text{data}\rangle\,\lvert\text{ancilla}\rangle$
No Error	$\alpha\,\lvert 000\rangle\,\lvert 00\rangle + \beta\,\lvert 111\rangle\,\lvert 00\rangle$
Qubit 1	$\alpha\,\lvert 100\rangle\,\lvert 11\rangle + \beta\,\lvert 011\rangle\,\lvert 11\rangle$
Qubit 2	$\alpha\,\lvert 010\rangle\,\lvert 10\rangle + \beta\,\lvert 101\rangle\,\lvert 10\rangle$
Qubit 3	$\alpha\,\lvert 001\rangle\,\lvert 01\rangle + \beta\,\lvert 110\rangle\,\lvert 01\rangle$

Three-qubit phase-flip code

- How do we do the same to protect against phase-flips?
- $\sigma_z |0\rangle = |0\rangle$
- $\sigma_z |1\rangle = -|1\rangle$
- Anyone?

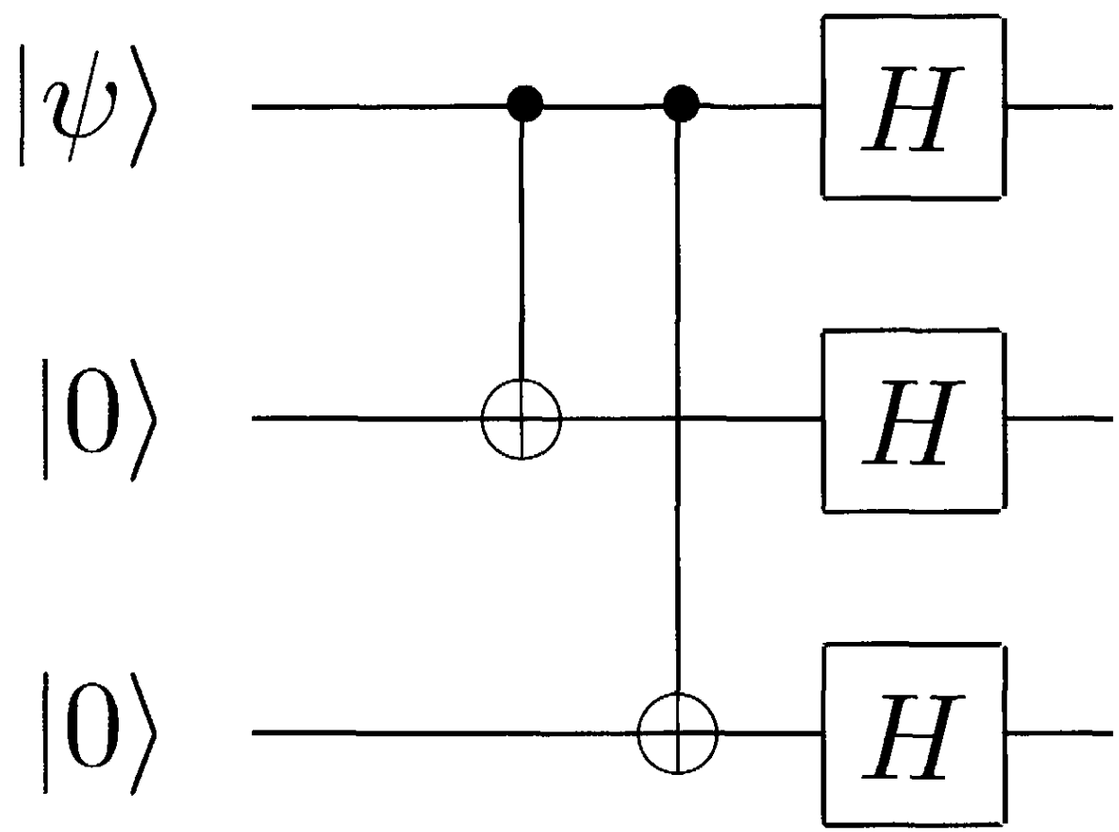
Three-qubit phase-flip code

- Let use consider the states $|+\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}, |-\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}}$
- Notice that
- $\sigma_z |+\rangle = |-\rangle$
- $\sigma_z |-\rangle = |+\rangle$
- So we can encode our state as

$$|\psi\rangle_L = \alpha |+++\rangle + \beta |---\rangle$$

Three-qubit phase-flip code

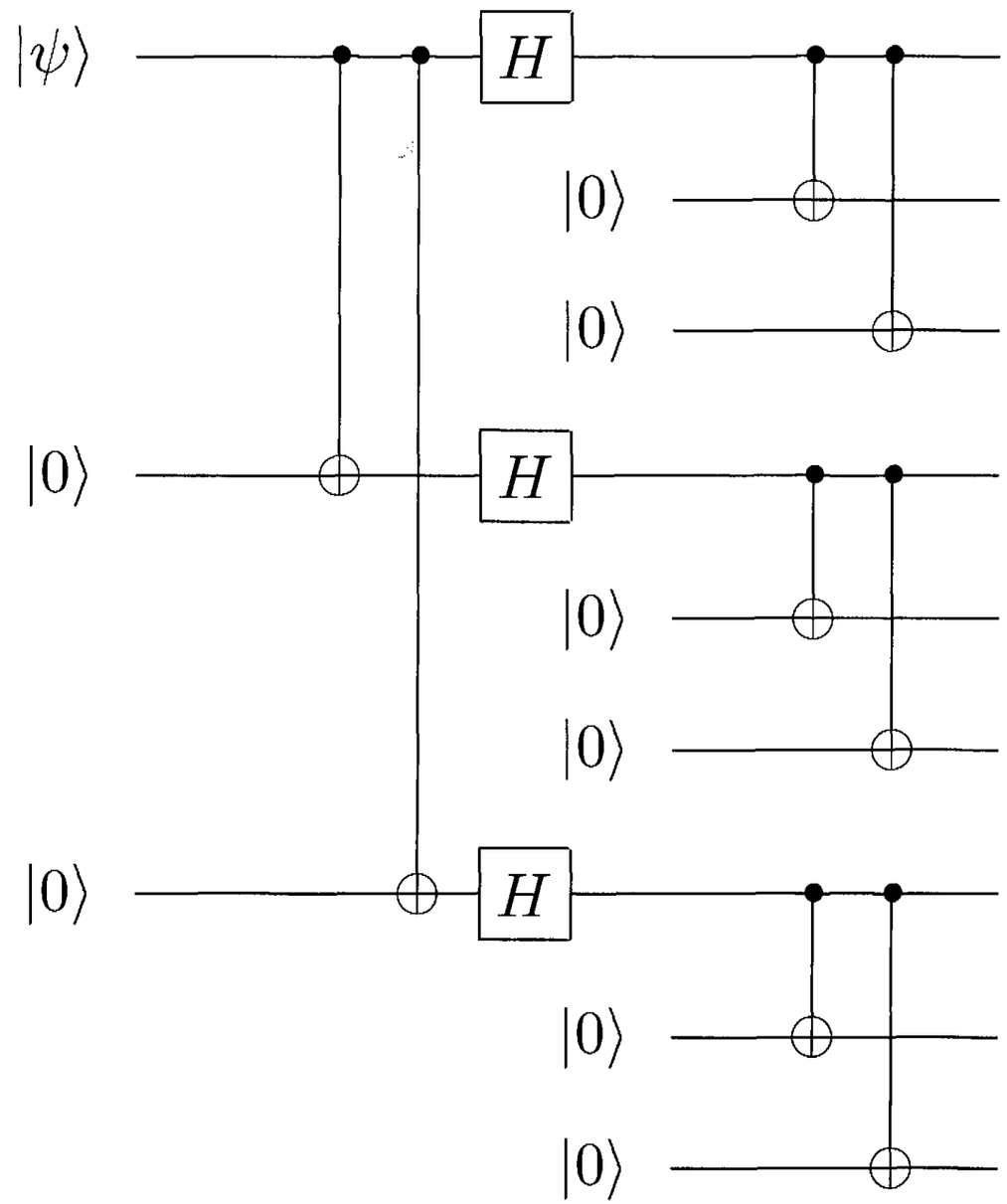
$$|\psi\rangle_L = \alpha|+++\rangle + \beta|---\rangle$$



The Shor code

- The Shor code was designed by Shor in **1995** and marks the **first** code that can correct any **bit-flip** and **phase-flip** affecting only one qubit

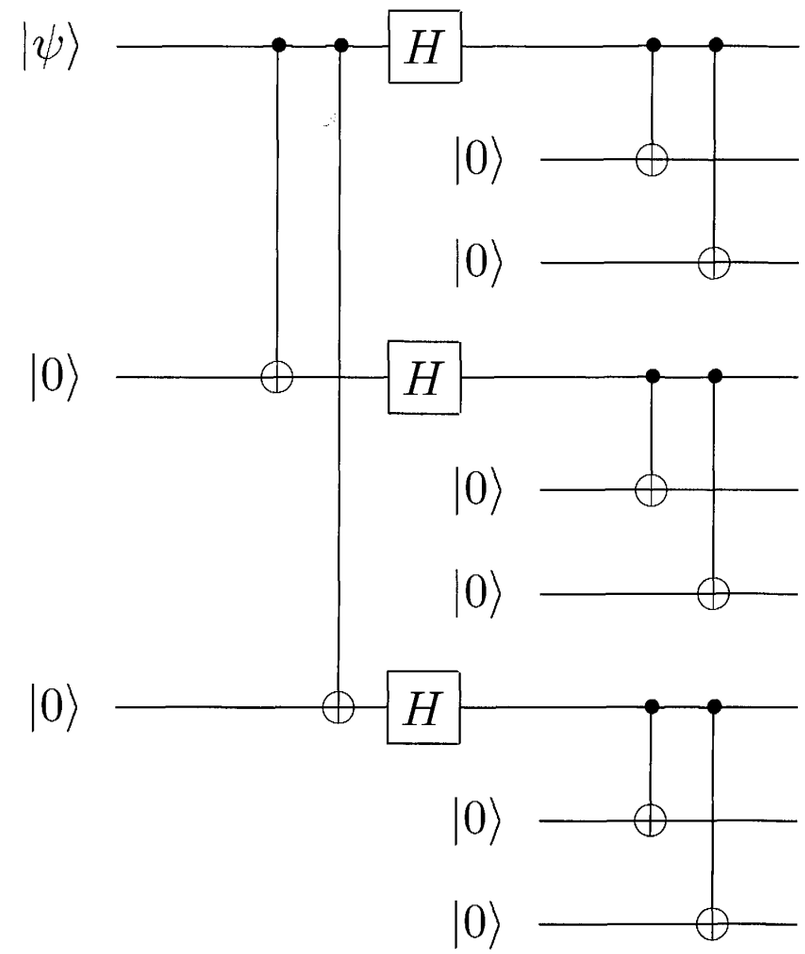
- **9** qubits encoding 1 **single logical** state



The Shor code

- The Shor code was designed by Shor in **1995** and marks the **first** code that can correct any **bit-flip** and **phase-flip** affecting only one qubit

- **9** qubits encoding 1 **single logical** state



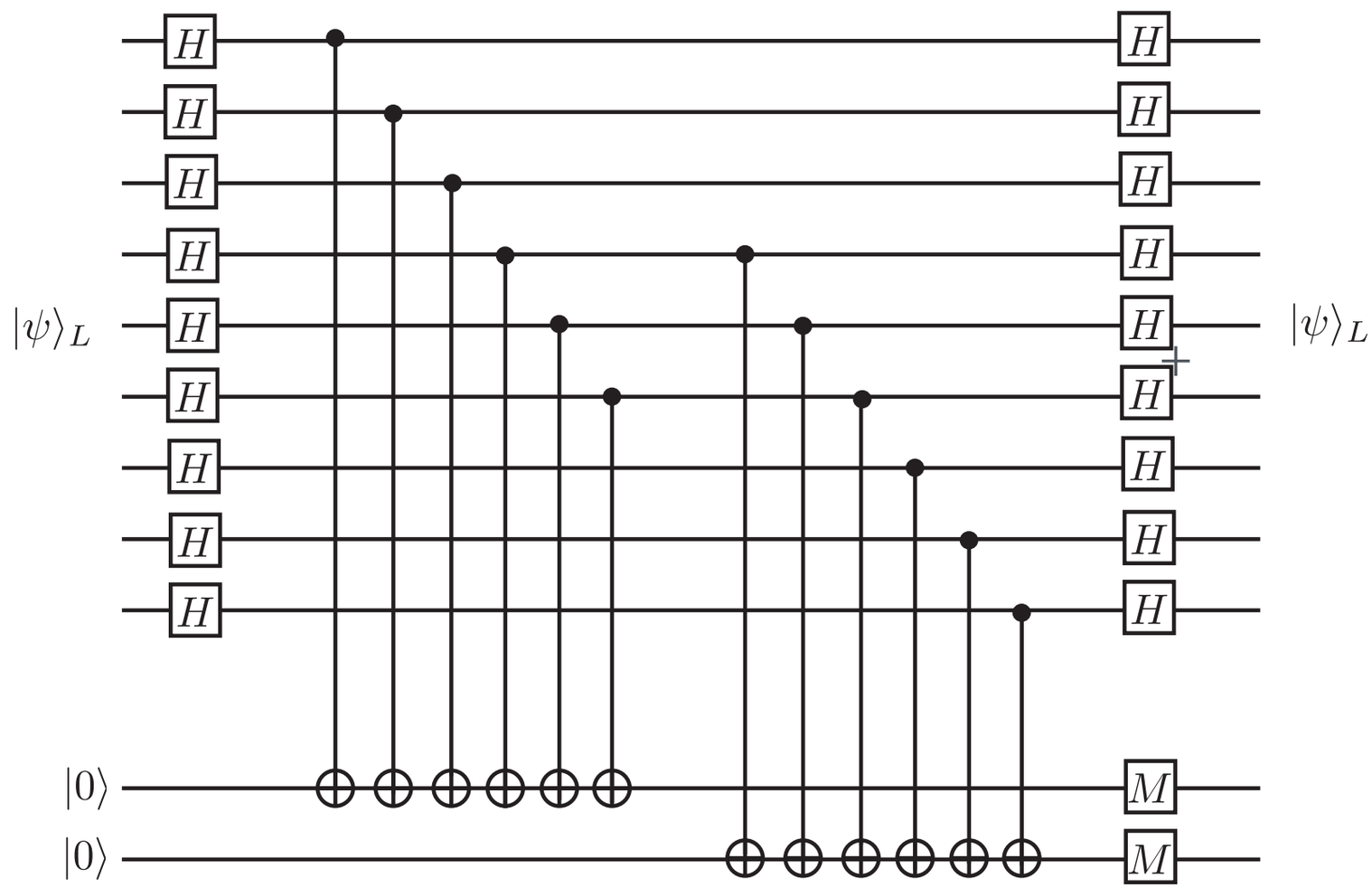
$$|0\rangle \rightarrow |0_L\rangle \equiv \frac{(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)}{2\sqrt{2}}$$

$$|1\rangle \rightarrow |1_L\rangle \equiv \frac{(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)}{2\sqrt{2}}$$

The Shor code

$$|0\rangle \rightarrow |0_L\rangle \equiv \frac{(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)}{2\sqrt{2}}$$

$$|1\rangle \rightarrow |1_L\rangle \equiv \frac{(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)}{2\sqrt{2}}$$



Circuit required to perform Z-error correction for the 9-qubit code.

Error correction

- What we need to do for error correction
  - Encode a lower-dimensional state in a larger one (we need redundancy)
  - We need a membership test
    - **Know if the code word is valid without measuring the state**
  - When a word is not valid, we need to be able to guess the likely error that occurred

Valid codewords

$|000\rangle$

$|111\rangle$

Non-valid codeword

$|100\rangle$

The stabilizer formalism

- Stabilizer formalism is an important tool for error correction
- It is a general formalism used to test if a state is of a certain form without measuring it
- With it we can create and understand error correction codes easily

A state $|\psi\rangle$ is defined to be stabilized by some operator K,
if it is a eigenstate of , i.e.

$$K|\psi\rangle = |\psi\rangle$$

The stabilizer formalism

- Examples:
  - The qubit state $|0\rangle$ is stabilized by $K = \sigma_{z'}\, \sigma_z |0\rangle = |0\rangle$
  - The qubit state $|1\rangle$ is stabilized by $K = -\sigma_{z'}\, -\sigma_z |1\rangle = |1\rangle$
  - The qubit state $|+\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right)$ is stabilized by $K = \sigma_x$
  - The qubit state $|-\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle - |1\rangle\right)$ is stabilized by $K = -\sigma_x$

$$\sigma_0 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad \sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

The stabilizer formalism

- Let us suppose we have the state, $|\psi\rangle = |0\rangle$ and the σ_z

$$\sigma_z|0\rangle = |0\rangle$$

- And there is an error $\sigma_{x'}$ $\quad \sigma_x|\psi\rangle = |1\rangle$

- Now

$$\sigma_z|\psi\rangle = -|\psi\rangle$$

The stabilizer formalism

- Let us suppose we have the state, $|\psi\rangle = |+\rangle$ and the stabilizer σ_x

$$\sigma_x |+\rangle = |+\rangle$$

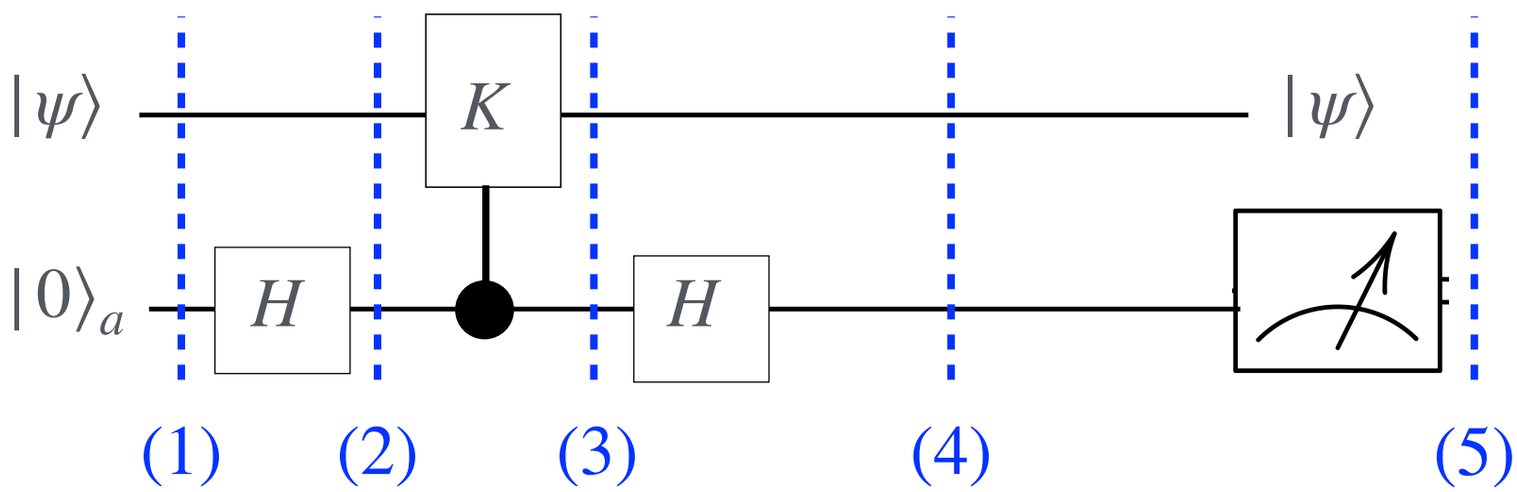
- And there is an error $\sigma_{z'}$ $\quad \sigma_z |\psi\rangle = |1\rangle$

- Now

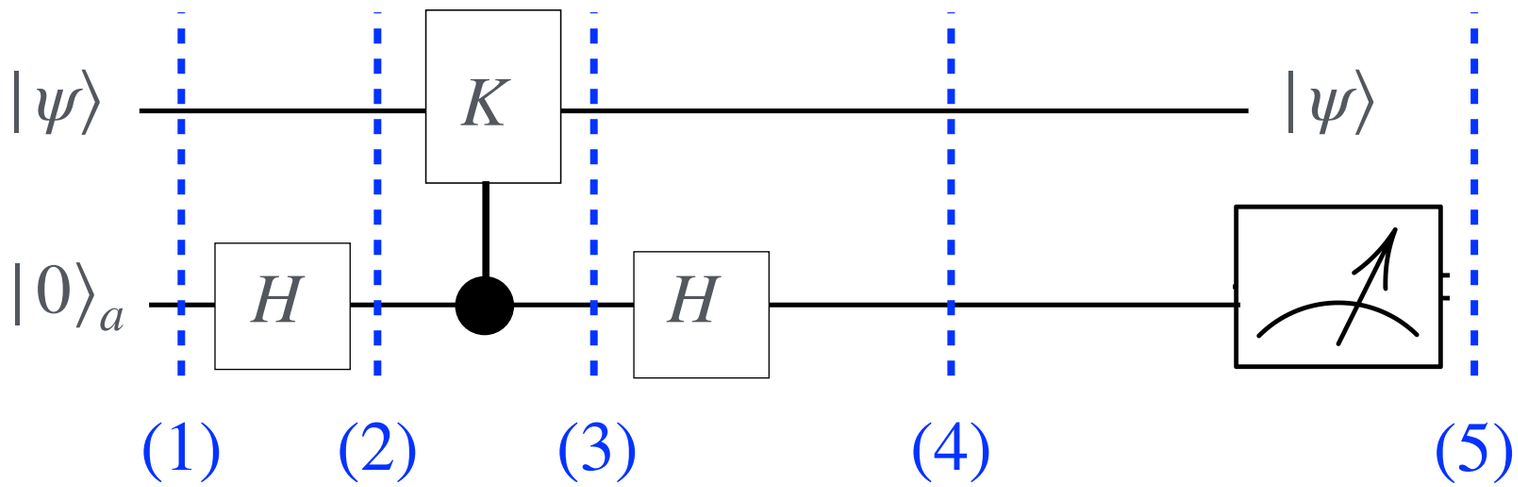
$$\sigma_x |\psi\rangle = -|\psi\rangle$$

Test

- We can test the phase shift without directly measuring the state



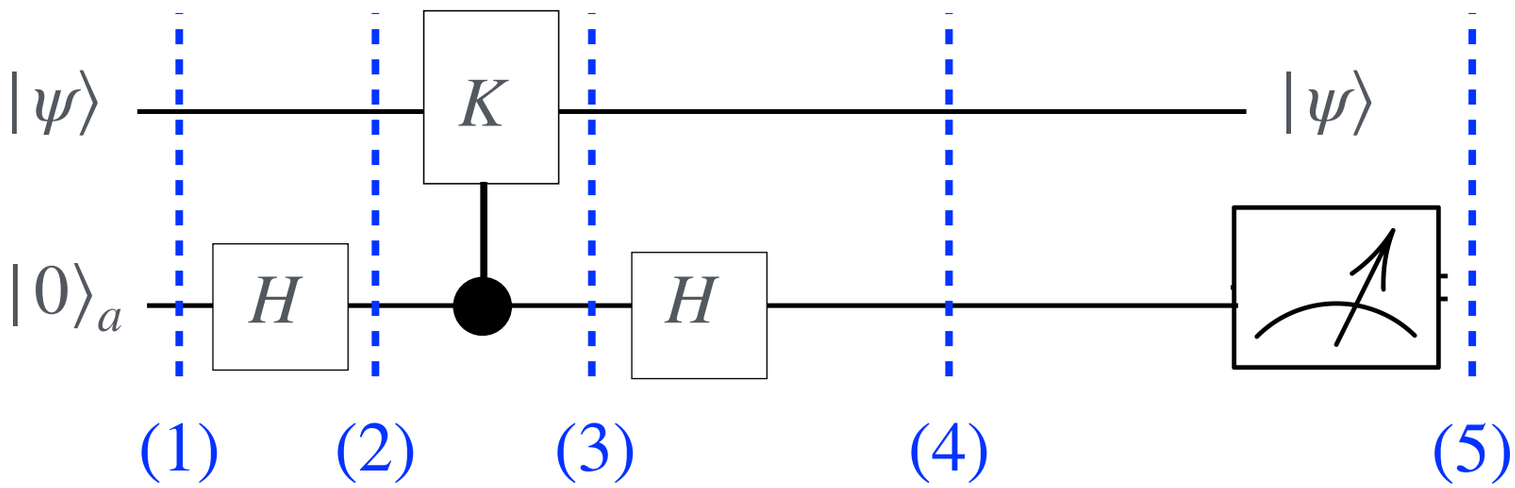
Test



(1) $|\psi\rangle|0\rangle_a$

(2) $\frac{1}{\sqrt{2}}|\psi\rangle(|0\rangle_a + |1\rangle_a) = \frac{1}{\sqrt{2}}\left(|\psi\rangle|0\rangle_a + |\psi\rangle|1\rangle_b\right)$

Test



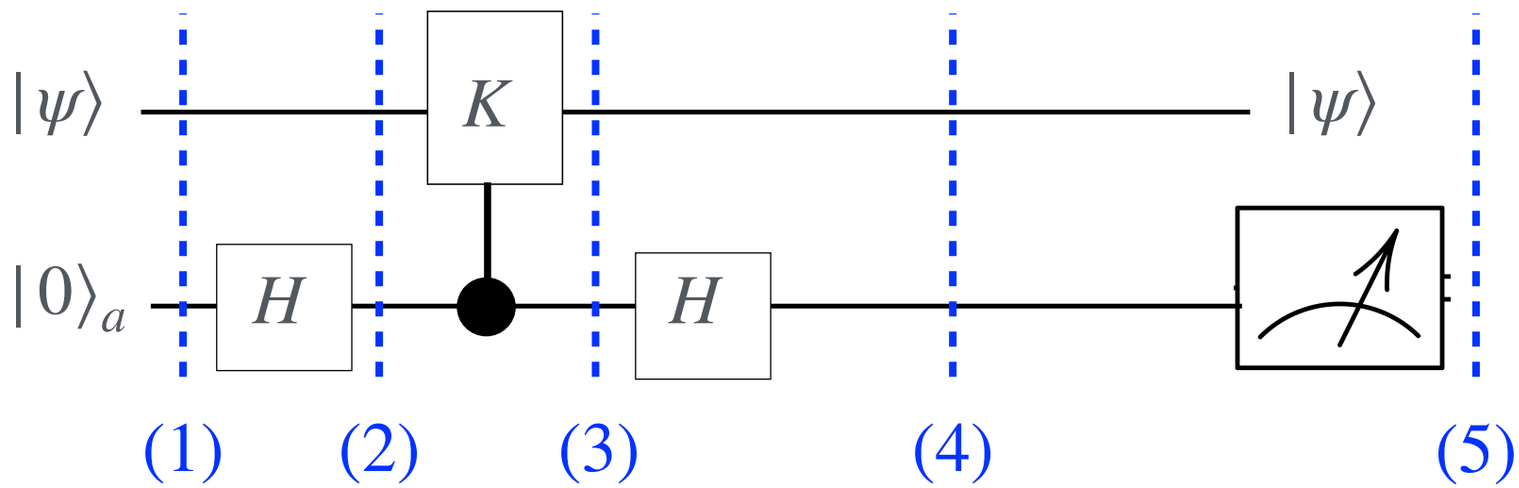
(2) $$\frac{1}{\sqrt{2}}|\psi\rangle(|0\rangle_a + |1\rangle_a) = \frac{1}{\sqrt{2}}\left(|\psi\rangle|0\rangle_a + \psi\rangle|1\rangle_b\right)$$

(3) $$\frac{1}{\sqrt{2}}\left(|\psi\rangle|0\rangle_a + K|\psi\rangle|1\rangle_b\right) = \left(|\psi\rangle|0\rangle_a \pm |\psi\rangle|1\rangle_b\right) = |\psi\rangle|\pm\rangle$$

Test

- But if $K|\psi\rangle$ is not either -1 or 1?
  - Because we are measuring the ancilla, the state always collapses to one of the two subspaces associated with -1 and 1
  - Let us consider
  - the state $|\psi\rangle = a|0\rangle + b|1\rangle$
  - And the stabilizer σ_z

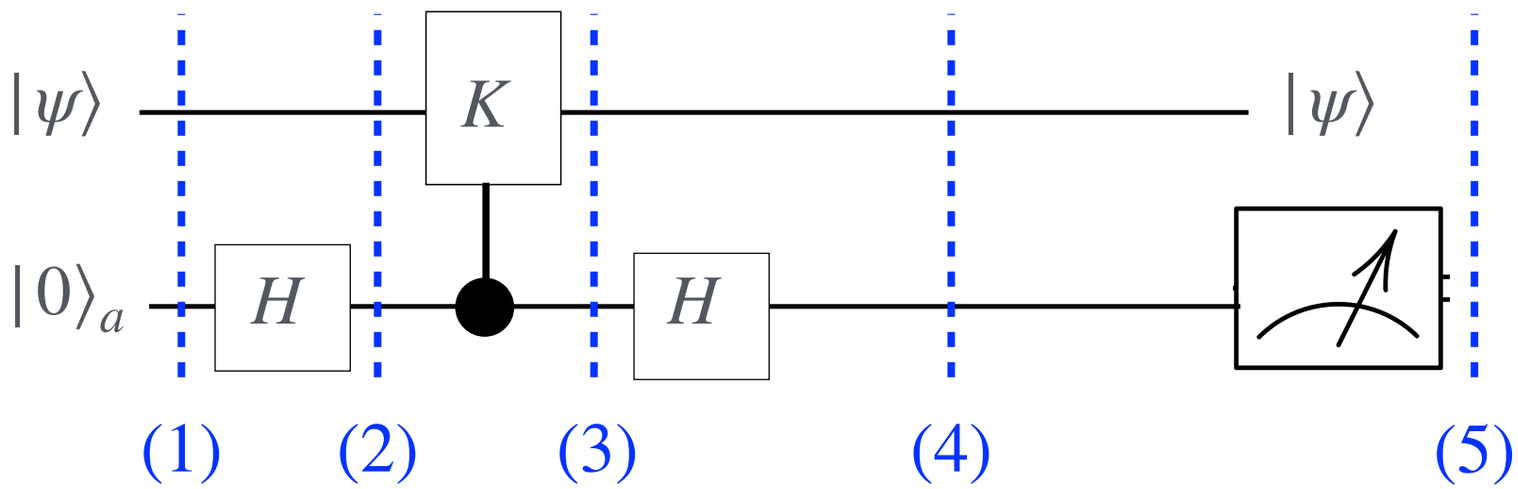
Test



(1) $\big(a|0\rangle + b|1\rangle\big)|0\rangle_a$

(2) $\dfrac{a}{\sqrt{2}}|0\rangle\big(|0\rangle_a + |1\rangle_a\big) + \dfrac{b}{\sqrt{2}}|1\rangle\big(|0\rangle_a + |1\rangle_a\big)$

Test

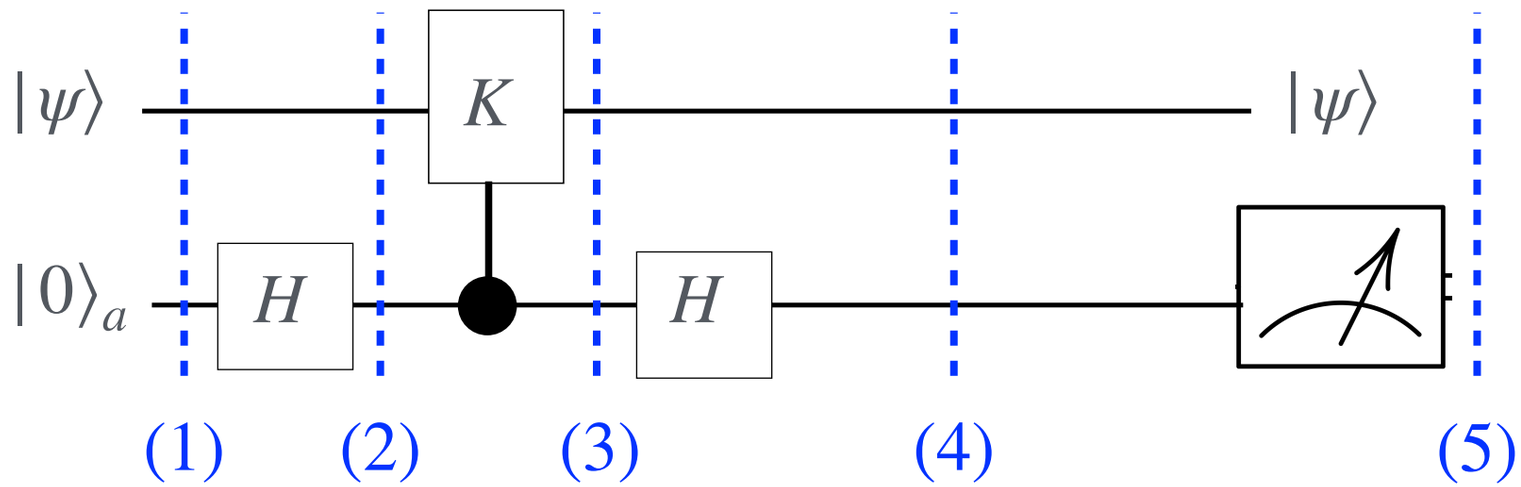


(2) $$\frac{a}{\sqrt{2}}|0\rangle\left(|0\rangle_a+|1\rangle_a\right)+\frac{b}{\sqrt{2}}|1\rangle\left(|0\rangle_a+|1\rangle_a\right)$$

$$=\frac{a}{\sqrt{2}}\left(|0\rangle|0\rangle_a+|0\rangle|1\rangle_a\right)+\frac{b}{\sqrt{2}}\left(|1\rangle|0\rangle_a+|1\rangle|1\rangle_a\right)$$

(3) $$\frac{a}{\sqrt{2}}\left(|0\rangle|0\rangle_a+\sigma_z|0\rangle|1\rangle_a\right)+\frac{b}{\sqrt{2}}\left(|1\rangle|0\rangle_a+\sigma_z|1\rangle|1\rangle_a\right)$$

Test



(3) $$\frac{a}{\sqrt{2}}|0\rangle|+\rangle_a + \frac{b}{\sqrt{2}}|1\rangle|-\rangle_a$$

(4) $$\frac{a}{\sqrt{2}}|0\rangle|0\rangle_a + \frac{b}{\sqrt{2}}|1\rangle|1\rangle_a$$

Test

- In this case, the **superposition is lost because** we know exactly the state associated with each measurement of the ancilla qubit
  - **Not very useful**
- But we can use the same trick to **reduce** the **dimensionality** of a higher-dimensional state to a lower-dimensional one

Stabilizers examples

- Let us consider a two-qubit system. The following states form a complete basis

$$|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \quad |\Phi^-\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle)$$
$$|\Psi^+\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle) \quad |\Psi^-\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle)$$

- That are stabilized by

$$\Phi^+ \equiv \begin{pmatrix} K^1 = XX \\ K^2 = ZZ \end{pmatrix} \quad \Phi^- \equiv \begin{pmatrix} K^1 = -XX \\ K^2 = ZZ \end{pmatrix}$$
$$\Psi^+ \equiv \begin{pmatrix} K^1 = XX \\ K^2 = -ZZ \end{pmatrix} \quad \Psi^- \equiv \begin{pmatrix} K^1 = -XX \\ K^2 = -ZZ \end{pmatrix}$$

Stabilizers examples

$$|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \quad |\Phi^-\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle)$$

$$|\Psi^+\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle) \quad |\Psi^-\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle)$$

- For the stabilizer $K = \sigma_z \otimes \sigma_z \rightarrow ZZ$
- Only the $|\Phi^+\rangle$ and $|\phi^+\rangle$ states are stabilized by K

Stabilizers examples

- All states that pass a test with stabilizer KK are a linear combination of

$$|0\rangle_L = |\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$

$$|1\rangle_L = |\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle)$$

- In other words, all the states such that $XX|\psi\rangle = |\psi\rangle$, are of the form
  - $|\psi\rangle = a|0\rangle_L + b|1\rangle_L$
- In also another take, we measure the parity between states using stabilizers without measuring the state

Stabilizers examples

$$|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \quad |\Phi^-\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle)$$

$$|\Psi^+\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle) \quad |\Psi^-\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle)$$

- For the stabilizer $K_2 = \sigma_x \otimes \sigma_x \rightarrow XX$
- Only the $|\Phi^+\rangle$ and $|\psi^+\rangle$ states are stabilized by K

Stabilizers examples

- All states that pass a test with stabilizer K_2 are a linear combination of

$$|0\rangle_L = |\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$

$$|1\rangle_L = |\Psi^+\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle)$$

- In other words, all the states such that $XX|\psi\rangle = |\psi\rangle$, are of the form
  - $|\psi\rangle = a|0\rangle_L + b|1\rangle_L$
- We measure the parity of the relative phase

Stabilizers examples - building the Shor code

$$|0\rangle \rightarrow |0_L\rangle \equiv \frac{(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)}{2\sqrt{2}}$$

$$|1\rangle \rightarrow |1_L\rangle \equiv \frac{(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)}{2\sqrt{2}}$$

- Exercise
- What are the parities here?
- What are the stabilizers?

Stabilizers examples - building the Shor code

$$|0\rangle \rightarrow |0_L\rangle \equiv \frac{(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)}{2\sqrt{2}}$$

$$|1\rangle \rightarrow |1_L\rangle \equiv \frac{(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)}{2\sqrt{2}}$$

- Exercise
- What are the parities here?
- What are the stabilizers?

The Shor code - revisited

- 9 qubits encode one single qubit
- Corrects all single-qubit errors

K^1	Z	Z	I	I	I	I	I	I	I
K^2	Z	I	Z	I	I	I	I	I	I
K^3	I	I	I	Z	Z	I	I	I	I
K^4	I	I	I	Z	I	Z	I	I	I
K^5	I	I	I	I	I	I	Z	Z	I
K^6	I	I	I	I	I	I	Z	I	Z
K^7	X	X	X	X	X	X	I	I	I
K^8	X	X	X	I	I	I	X	X	X

How to correct errors?

$$|0\rangle_L = |\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$

$$|1\rangle_L = |\Psi^+\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle)$$

$$|\Phi^-\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle) \quad ZI|\Phi^-\rangle = |0\rangle_L$$

$$|\Psi^-\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle) \quad ZI|\Psi^-\rangle = |1\rangle_L$$

- Remember that $\sigma_i^2 = I$
- So if we have an z error on qubit 1,
  - that error can be corrected by applying σ_z on qubit 1

Steane code

- The Steane code was Introduced in 1

- It is one of the smallest codes that can connect any single arbitrary error in a quantum system

- Seven physical qubits, 1 logical qubit, distance 3. [7,1,3]

Steane code

K^1	I	I	I	X	X	X	X
K^2	X	I	X	I	X	I	X
K^3	I	X	X	I	I	X	X
K^4	I	I	I	Z	Z	Z	Z
K^5	Z	I	Z	I	Z	I	Z
K^6	I	Z	Z	I	Z	Z	I

The Steane code

- 7 qubits encode one single qubit
- Corrects all single-qubit errors

K^1	I	I	I	X	X	X	X
K^2	X	I	X	I	X	I	X
K^3	I	X	X	I	I	X	X
K^4	I	I	I	Z	Z	Z	Z
K^5	Z	I	Z	I	Z	I	Z
K^6	I	Z	Z	I	Z	Z	I

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

The Shor code

	σ_x				σ_y				
K^1	Z	Z	I	I	I	I	I	I	I
K^2	Z	I	Z	I	I	I	I	I	I
K^3	I	I	I	Z	Z	I	I	I	I
K^4	I	I	I	Z	I	Z	I	I	I
K^5	I	I	I	I	I	I	Z	Z	I
K^6	I	I	I	I	I	I	Z	I	Z
K^7	X	X	X	X	X	X	I	I	I
K^8	X	X	X	I	I	I	X	X	X

The Steane code

$i = S_3, S_2, S_1$

Detects

K^1	X	I	X	I	X	I	X	
K^2	I	X	X	I	I	X	X	**Z**
K^3	I	I	I	X	X	X	X	
K^4	Z	I	Z	I	Z	I	Z	
K^5	I	Z	Z	I	Z	Z	I	**X**
K^6	I	I	I	Z	Z	Z	Z	

$j = S_6, S_5, S_4$

The Steane code

No error

$i = 000$

$$
\begin{array}{ccccccc}
K^1 & X & I & X & I & X & I & X \\
K^2 & I & X & X & I & I & X & X \\
K^3 & I & I & I & X & X & X & X \\
\hdashline
K^4 & Z & I & Z & I & Z & I & Z \\
K^5 & I & Z & Z & I & Z & Z & I \\
K^6 & I & I & I & Z & Z & Z & Z
\end{array}
$$

$j = 000$

The Steane code

$i = 001$

Error	σ_z						
K^1	X	I	X	I	X	I	X
K^2	I	X	X	I	I	X	X
K^3	I	I	I	X	X	X	X
K^4	Z	I	Z	I	Z	I	Z
K^5	I	Z	Z	I	Z	Z	I
K^6	I	I	I	Z	Z	Z	Z

$j = 000$

Can be corrected with $\boldsymbol{\sigma}_z$ on qubit 1

The Steane code

$i = 010$

Error		σ_z				
K^1	X	I	X	I	X	I	X
K^2	I	X	X	I	I	X	X
K^3	I	I	I	X	X	X	X
K^4	Z	I	Z	I	Z	I	Z
K^5	I	Z	Z	I	Z	Z	I
K^6	I	I	I	Z	Z	Z	Z

$j = 000$

Can be corrected with $\boldsymbol{\sigma}_z$ on qubit 2

The Steane code

$i = 011$

Error σ_z

K^1	X	I	X	I	X	I	X
K^2	I	X	X	I	I	X	X
K^3	I	I	I	X	X	X	X
K^4	Z	I	Z	I	Z	I	Z
K^5	I	Z	Z	I	Z	Z	I
K^6	I	I	I	Z	Z	Z	Z

$j = 000$

Can be corrected with $\boldsymbol{\sigma}_z$ on qubit 3

The Steane code

$i = 100$

Error σ_z

$$
\begin{array}{cccccccc}
K^1 & X & I & X & I & X & I & X \\
K^2 & I & X & X & I & I & X & X \\
K^3 & I & I & I & X & X & X & X \\
\hline
K^4 & Z & I & Z & I & Z & I & Z \\
K^5 & I & Z & Z & I & Z & Z & I \\
K^6 & I & I & I & Z & Z & Z & Z
\end{array}
$$

$j = 000$

Can be corrected with $\boldsymbol{\sigma}_z$ on qubit 4

The Steane code

$i = 101$

Error

$$
\begin{array}{c|ccccccc}
 & & & & & \sigma_z & & \\
K^1 & X & I & X & I & X & I & X \\
K^2 & I & X & X & I & I & X & X \\
K^3 & I & I & I & X & X & X & X \\
\hline
K^4 & Z & I & Z & I & Z & I & Z \\
K^5 & I & Z & Z & I & Z & Z & I \\
K^6 & I & I & I & Z & Z & Z & Z
\end{array}
$$

$j = 000$

Can be corrected with $\boldsymbol{\sigma}_z$ one qubit 5

The Steane code

$i = 110$

Error σ_z

K^1	X	I	X	I	X	I	X
K^2	I	X	X	I	I	X	X
K^3	I	I	I	X	X	X	X
K^4	Z	I	Z	I	Z	I	Z
K^5	I	Z	Z	I	Z	Z	I
K^6	I	I	I	Z	Z	Z	Z

$j = 000$

Can be corrected with $\boldsymbol{\sigma}_z$ one qubit 6

The Steane code

$i = 111$

Error

							σ_z
K^1	X	I	X	I	X	I	X
K^2	I	X	X	I	I	X	X
K^3	I	I	I	X	X	X	X
K^4	Z	I	Z	I	Z	I	Z
K^5	I	Z	Z	I	Z	Z	I
K^6	I	I	I	Z	Z	Z	Z

$j = 000$

Can be corrected with $\boldsymbol{\sigma}_z$ one qubit 7

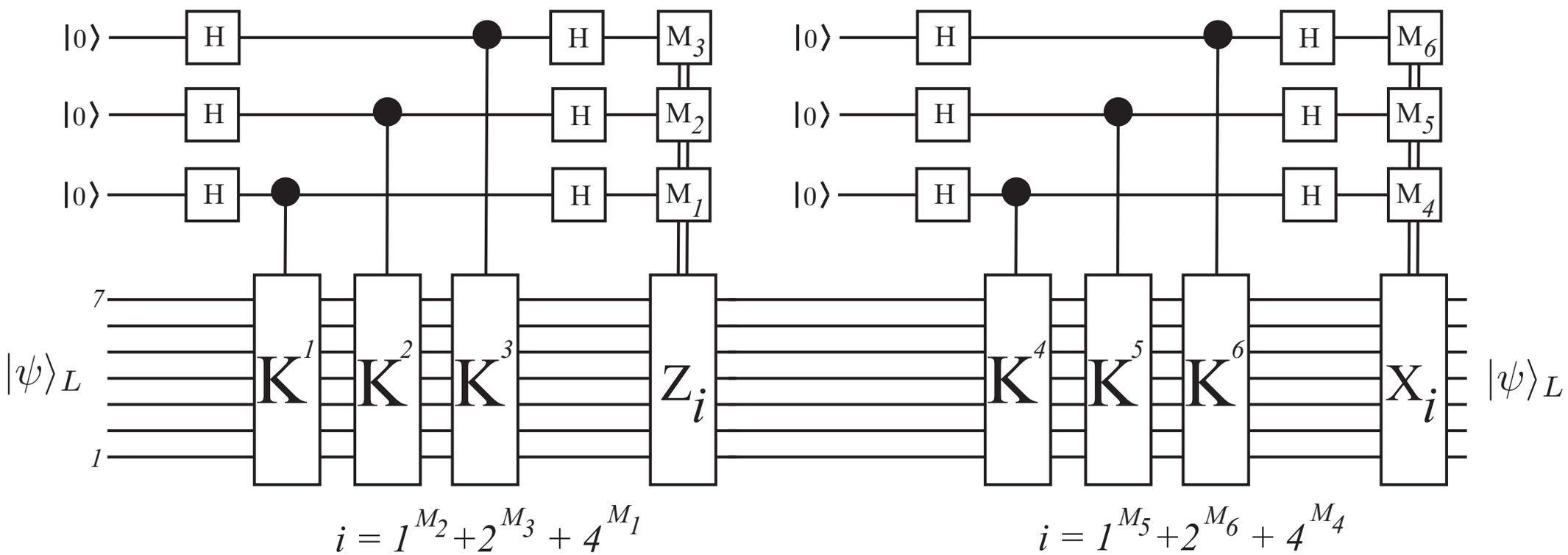
The Steane code

$i = 000$

Error	σ_x						
K^1	X	I	X	I	X	I	X
K^2	I	X	X	I	I	X	X
K^3	I	I	I	X	X	X	X
K^4	Z	I	Z	I	Z	I	Z
K^5	I	Z	Z	I	Z	Z	I
K^6	I	I	I	Z	Z	Z	Z

$j = 001$ Can be corrected with $\boldsymbol{\sigma}_x$ one qubit 1

Steane code


|0⟩
H
M3
M2
M1
M6
M5
M4
7
1
|ψ⟩L
K1
K2
K3
Zi
K4
K5
K6
Xi
i = 1^M2 + 2^M3 + 4^M1
i = 1^M5 + 2^M6 + 4^M4

Steane code

- Is this the smallest code capable of correcting all single-qubit errors?

Class 2

- CSS-codes and non-CSS codes
- The parity check matrix representation
- Surface codes
- Degeneracy
- The decoding problem
- Bibliography
  - Nielsen & Chuang, Quantum Computation and Quantum Information (Cambridge, 2010)
  - Devitt, Munro & Nemoto, "Quantum error correction for beginners," Rep. Prog. Phys. 76, 076001 (2013)

Steane code

- Is this the smallest code capable of correcting all single-qubit errors?

K^1	I	I	I	X	X	X	X
K^2	X	I	X	I	X	I	X
K^3	I	X	X	I	I	X	X
K^4	I	I	I	Z	Z	Z	Z
K^5	Z	I	Z	I	Z	I	Z
K^6	I	Z	Z	I	Z	Z	I

$$2^{\#_s} \geq \#_E$$

Steane code

- Is this the smallest code capable of correcting all single-qubit errors?

$$
\begin{array}{lccccccl}
K^1 & I & I & I & X & X & X & X \\
K^2 & X & I & X & I & X & I & X \\
K^3 & I & X & X & I & I & X & X \\
\hdashline
K^4 & I & I & I & Z & Z & Z & Z \\
K^5 & Z & I & Z & I & Z & I & Z \\
K^6 & I & Z & Z & I & Z & Z & I
\end{array}
\qquad
\begin{array}{l}
\\ \mathbf{Z} \\ \\ \\ \mathbf{X} \\ \\
\end{array}
$$

Steane code

- Is this the smallest code capable of correcting all single-qubit errors?

$$2^{\#_{sx}} \geq \#_{Ex} \qquad \#_{sx} = \#_{sz} = \#_s/2 = (N-1)/2$$

$$2^{\#_{sz}} \geq \#_{Ez} \qquad \#_{Ez} = \#_{Ex} = N+1$$

$$2^{\frac{N-1}{2}} \geq N+1$$

N must be at least 7

Short codes

- Is this the smallest code capable of correcting all single-qubit errors?

$$2^{\#_s} \geq \#_E$$

$$\#_s = N - 1$$

$$\#_E = \#_{Ex} + \#_{Ey} + \#_{Ez} + 1 = 3N + 1 \qquad \sigma_y = -i\sigma_x\sigma_z$$

$$2^{N-1} \geq 3N + 1 \longrightarrow N = 5$$

The 5-qubit code

K^1	X	Z	Z	X	I
K^2	I	X	Z	Z	X
K^3	X	I	X	Z	Z
K^4	Z	X	I	X	Z

You can check that for every X, Y, and Z affecting only one qubit, there is one and only one set of syndromes associated

The 5-qubit code

K^1	X	Z	Z	X	I
K^2	I	X	Z	Z	X
K^3	X	I	X	Z	Z
K^4	Z	X	I	X	Z

There is a constraint do in code creation with mixed or not mixed stabilizers

All stabilizers must commute [$[K_i, K_j] = 0$ for all i, j

CSS codes

- The Steane code is the smallest CSS code that correct one error
  - CSS -> Calderbank, Shor, and Steane
- This class of codes correct X and Z errors independently

$$
\begin{array}{ccccccc}
K^1 & I & I & I & X & X & X & X \\
K^2 & X & I & X & I & X & I & X \\
K^3 & I & X & X & I & I & X & X \\
\hdashline
K^4 & I & I & I & Z & Z & Z & Z \\
K^5 & Z & I & Z & I & Z & I & Z \\
K^6 & I & Z & Z & I & Z & Z & I
\end{array}
$$

Parity check matrices

$$
H_x = \begin{matrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \end{matrix}
$$

$$
H_z = \begin{matrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \end{matrix}
$$

CSS codes

- Most codes today are CSS
  - For reasons that I am not entirely convinced by
- Let me use this moment to introduce the parity check matrices
  - We can consider an error vector $\vec{e} = \{\vec{e}_z, \vec{e}_x\}$
  - And a parity check matrix $\mathbf{H}$ that measures the parities defined by the stabilizers
- The syndrome measurement of an error is given by

$$\vec{S} = \mathbf{H}\vec{\mathbf{e}}$$

CSS codes

$$H = \begin{pmatrix} H_X & 0 \\ 0 & H_Z \end{pmatrix}$$

K^1	I	I	I	X	X	X	X
K^2	X	I	X	I	X	I	X
K^3	I	X	X	I	I	X	X
K^4	I	I	I	Z	Z	Z	Z
K^5	Z	I	Z	I	Z	I	Z
K^6	I	Z	Z	I	Z	Z	I

Parity check matrices

$$H_x = \begin{matrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \end{matrix}$$

$$H_z = \begin{matrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \end{matrix}$$

CSS codes

- The stabilizer commutation condition for CSS codes is just

$$H_x H_z^T = 0 \mod 2$$

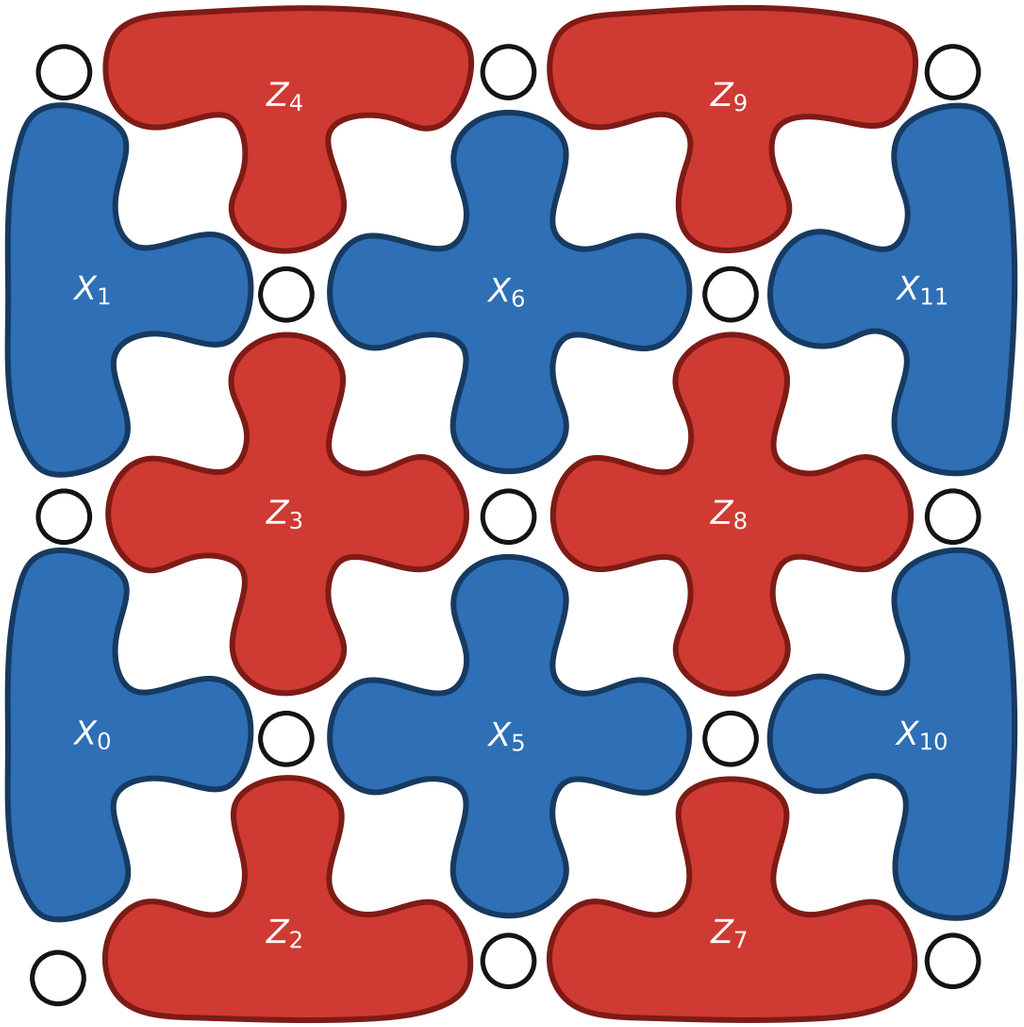
$$H_x = \begin{matrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \end{matrix} \qquad H_z = \begin{matrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \end{matrix}$$

Surface codes

- Is an example of a CSS code construction that respects the commutation constraint
  - I am not a big fan of surface codes
  - They are overused in research — we have much better codes
  - For quantum computers they are a bit more convenient than the alternatives, but with very low logical-to-physical encoding rates
  - They are, however, very good for explaining things

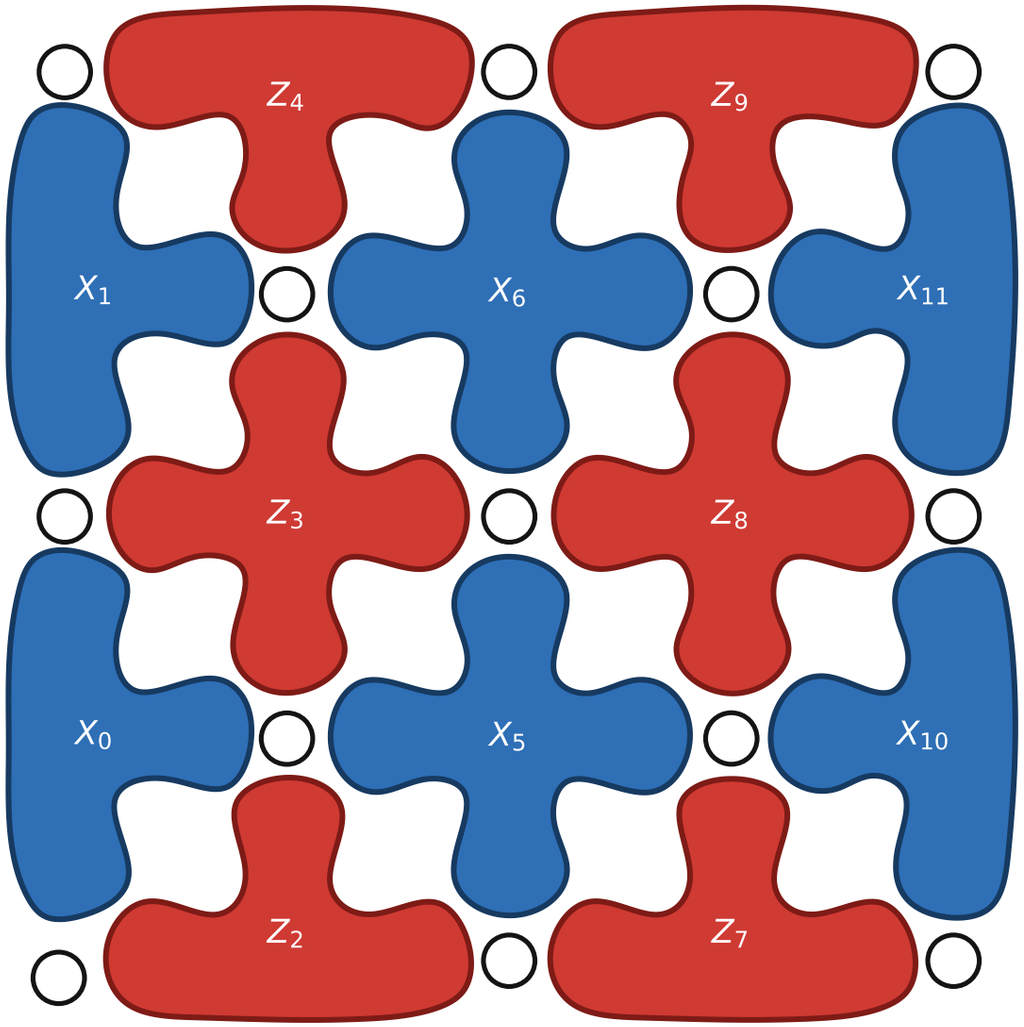
Surface codes

- What is a surface code?
- A surface code is a quantum error-correcting code laid out on a grid, with Z and X stabilizers each connected only to neighboring qubits.
- k is always 1
- n=13
- d=3 ($d \sim \sqrt{n}$)
- [13,1,3]



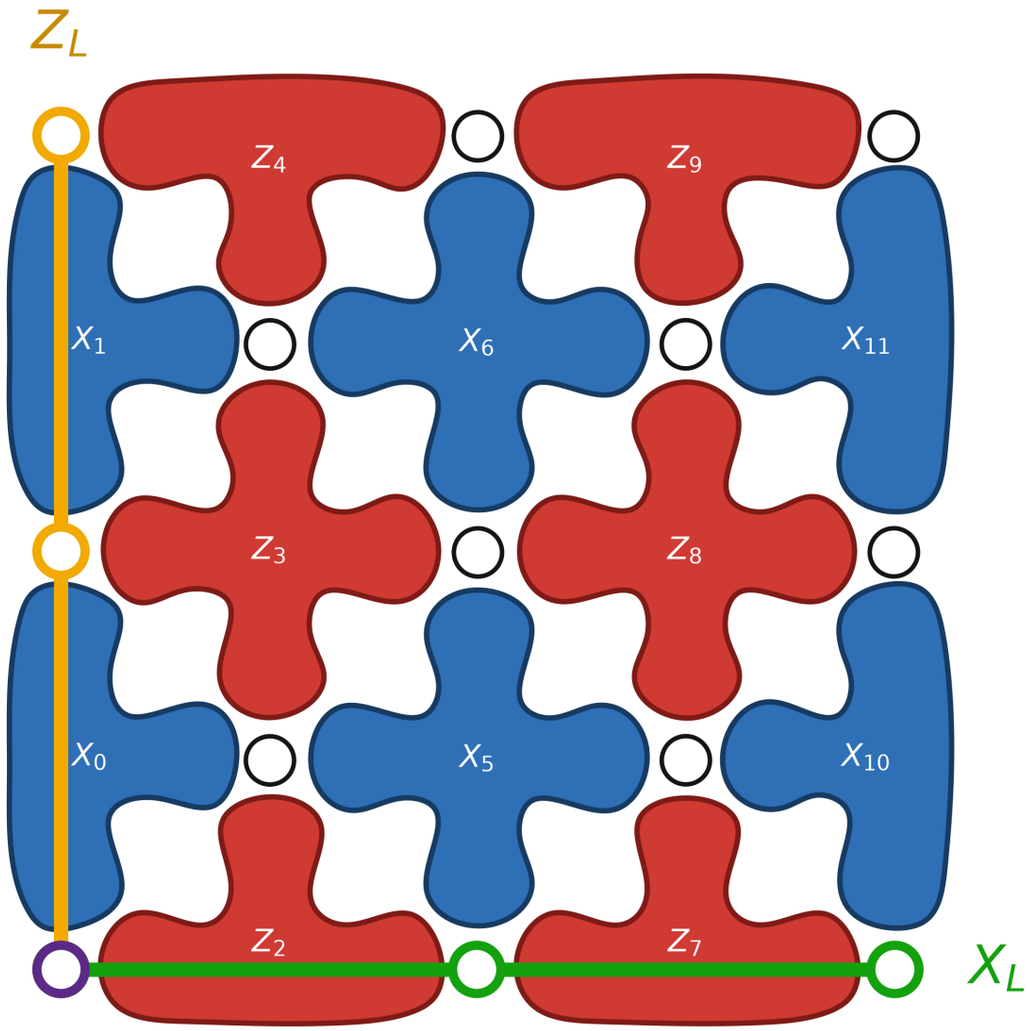
Surface codes

- Things to notice
  - All Z stabilizers and X stabilizers share two qubits
    - So $[H_x, H_z] = 0$
  - All connections are local



Surface codes

- I lied two times in this class,
  - **First lie**, I said a logical states is
  - $|\phi_L> = ..$
- Logical states are not defined like this in modern quantum error correction
- This kind of assumes we have access to a state, or at least that the state is well defined
  - This is not always true



Logical measurements as operators

- The Shor code again

$$|0\rangle \rightarrow |0_L\rangle \equiv \frac{(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)}{2\sqrt{2}}$$

$$|1\rangle \rightarrow |1_L\rangle \equiv \frac{(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)}{2\sqrt{2}}$$

- For the logical Z $X^{\mathrm{L}} = X_1 X_2 X_3,$ For the logical X $Z^{\mathrm{L}} = Z_1 Z_4 Z_7$

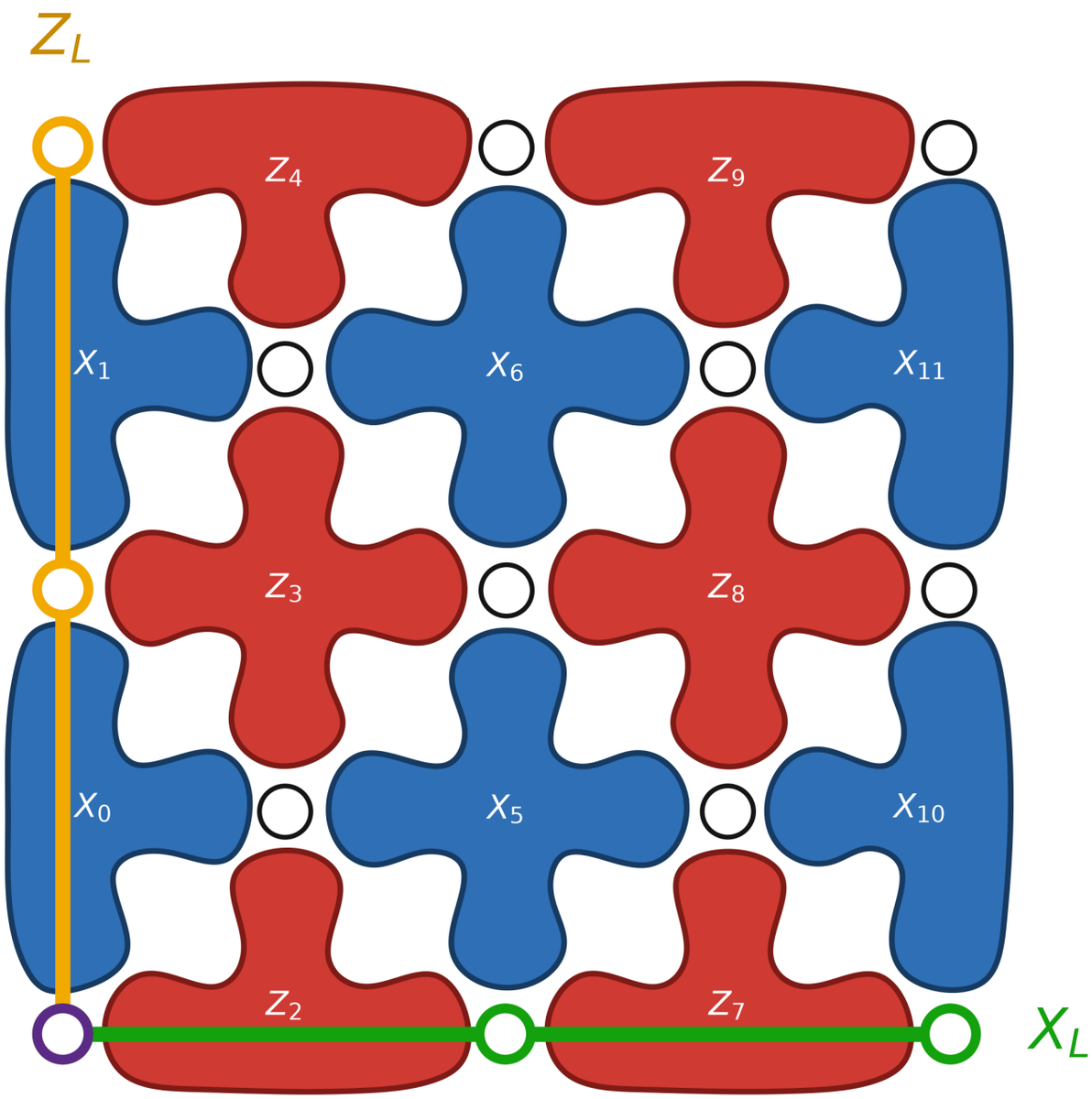
- There is a second set of states that give the same logical information and pass all the stabilizer tests

$$|0_L\rangle \propto (|000\rangle + |011\rangle + |101\rangle + |110\rangle)(|000\rangle + |011\rangle + |101\rangle + |110\rangle)(|000\rangle + |011\rangle + |101\rangle + |110\rangle)$$

$$|1_{\mathrm{L}}\rangle \propto (|001\rangle + |010\rangle + |100\rangle + |111\rangle)(|001\rangle + |010\rangle + |100\rangle + |111\rangle)(|001\rangle + |010\rangle + |100\rangle + |111\rangle)$$

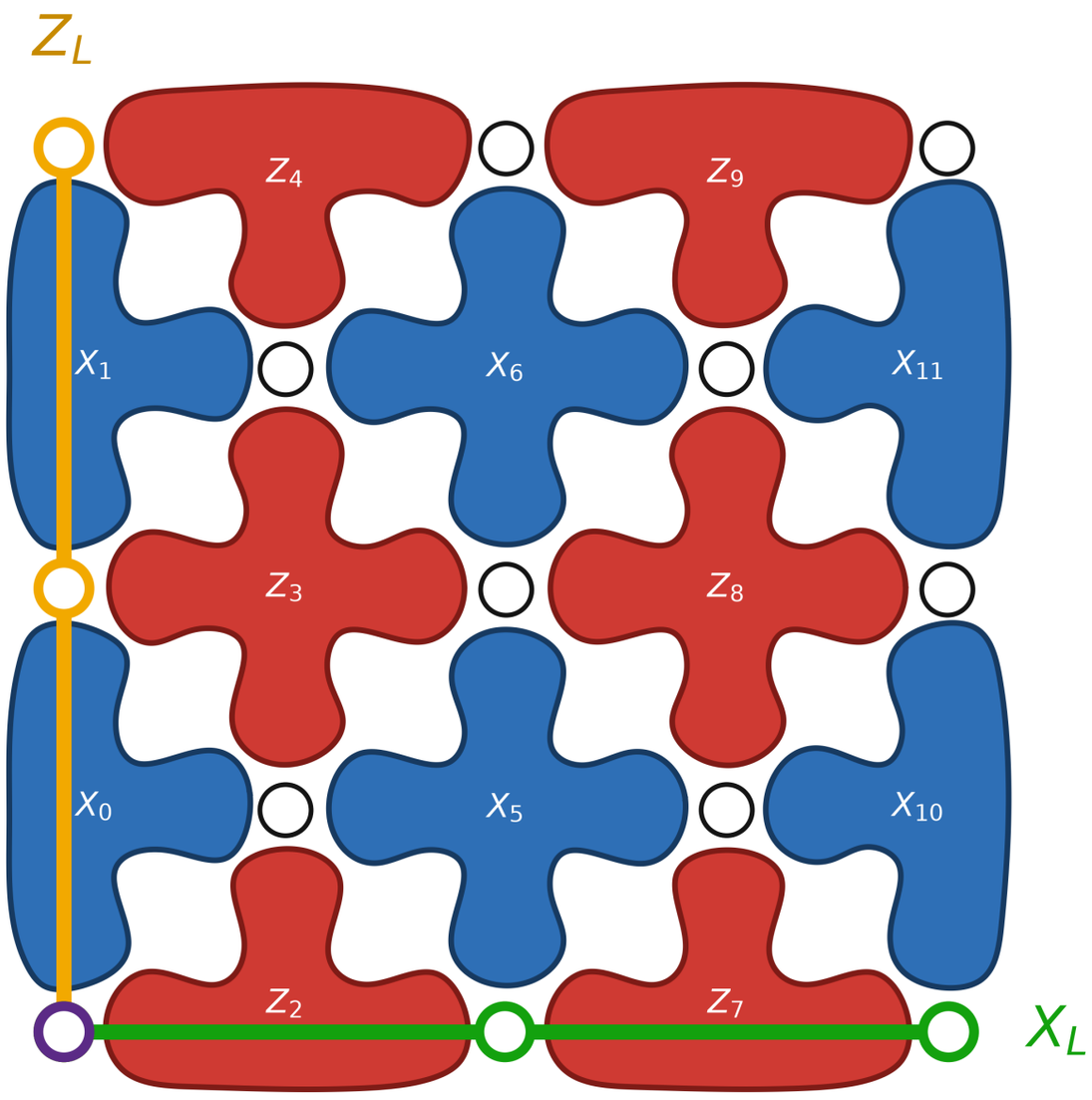
Surface codes

- Logical operators cross the entire codes
- Some errors do not change the logical state



Surface codes

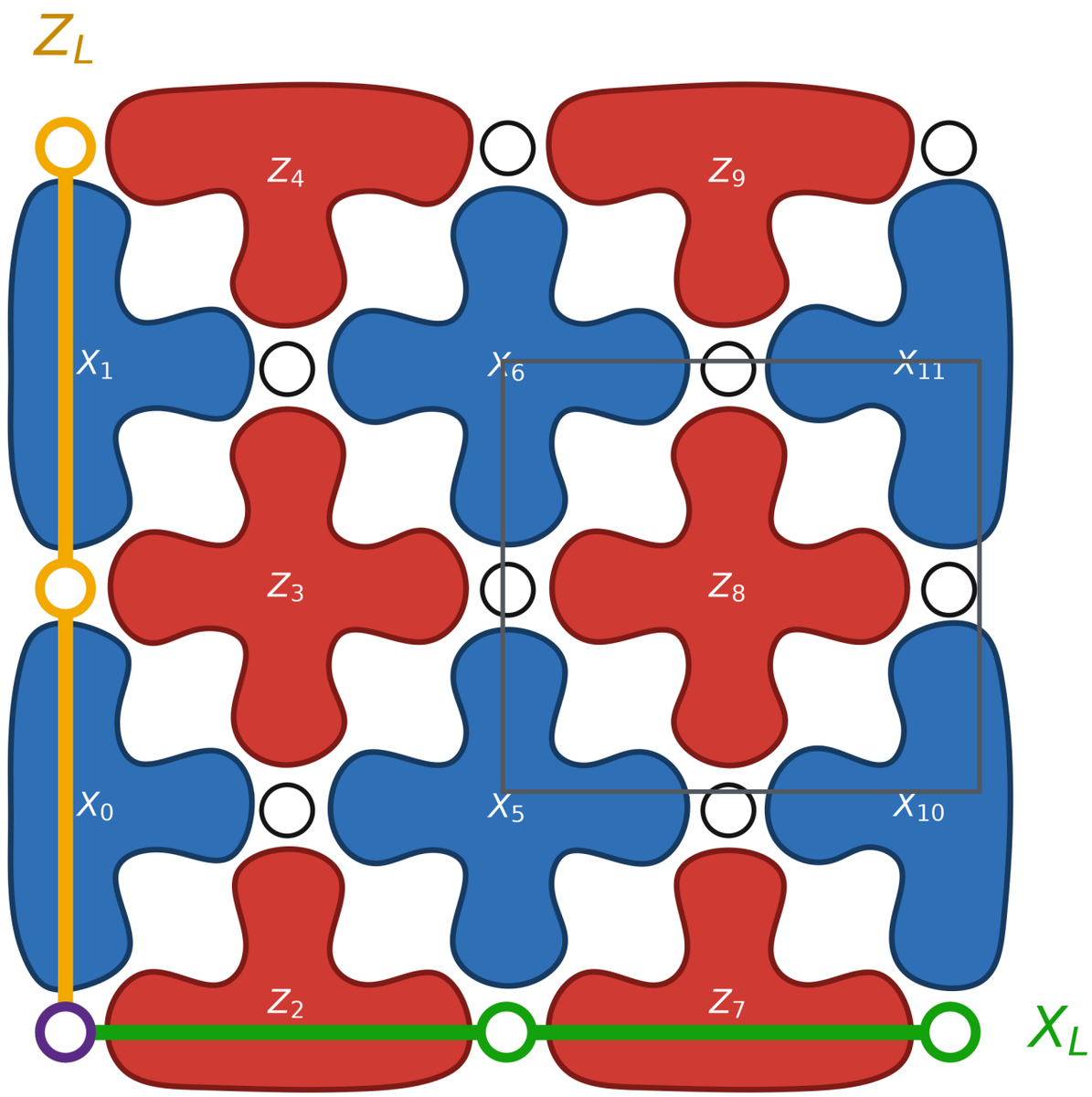
- There are tree types of error on surface codes
  - Degenerate errors
  - Correctable errors
  - And actual logical errors



Surface codes

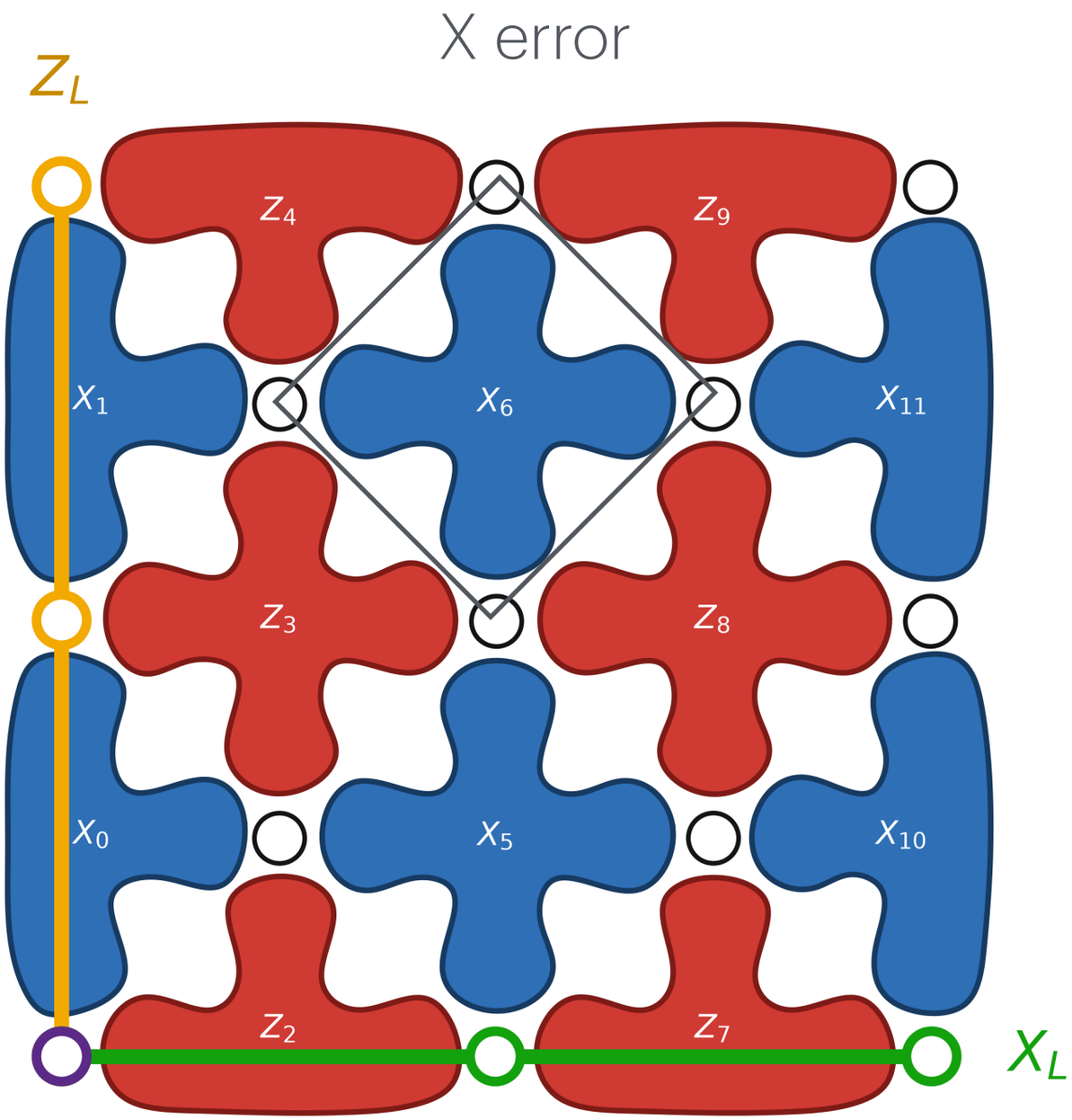
- There are tree types of error on surface codes
  - **Degenerate errors**
  - Correctable errors
  - And actual logical errors

Z error



Surface codes

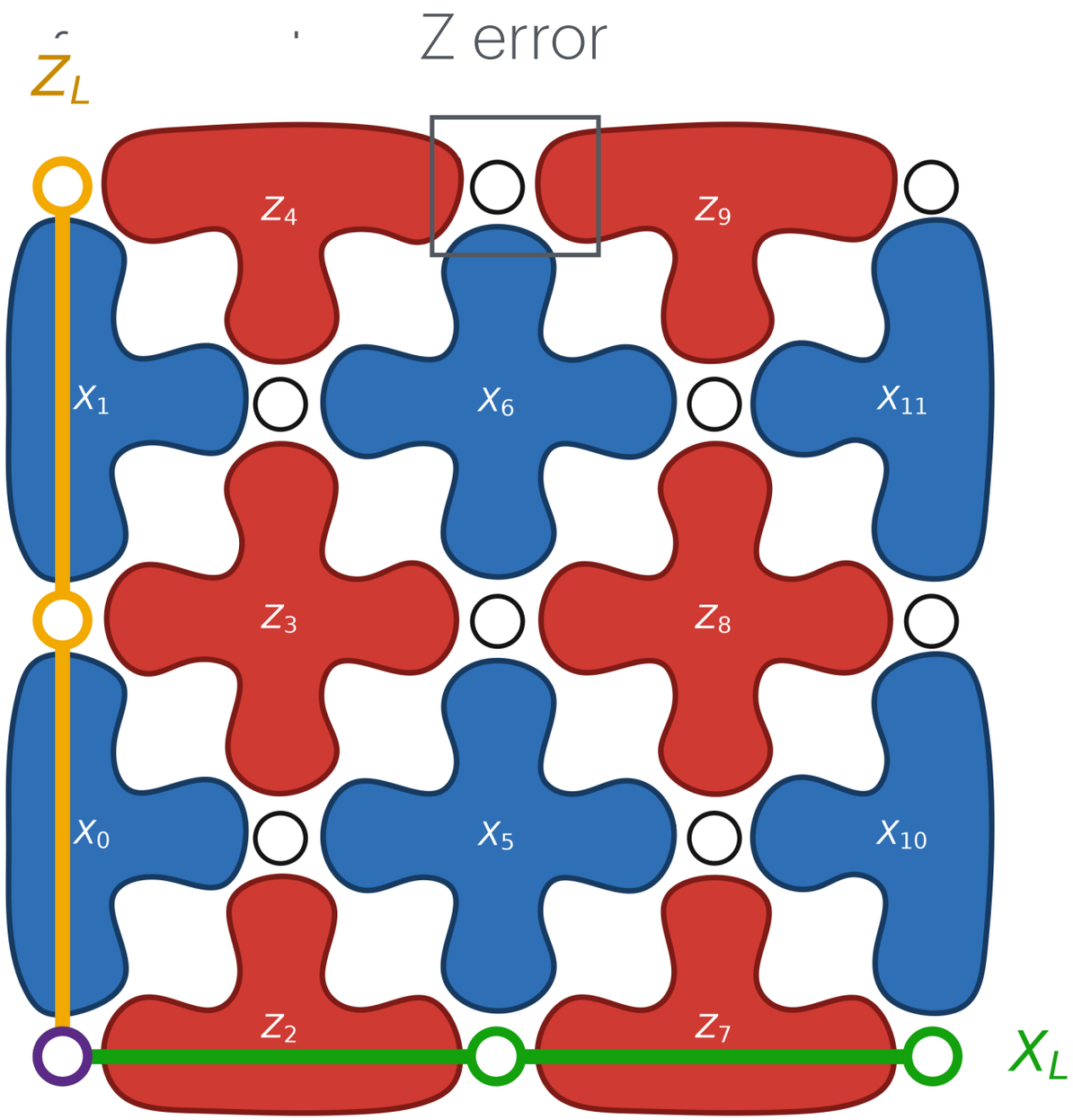
- There are tree types of error on surface codes
  - **Degenerate errors**
  - Correctable errors
  - An actual logical errors



Surface codes

- There are tree types of error on

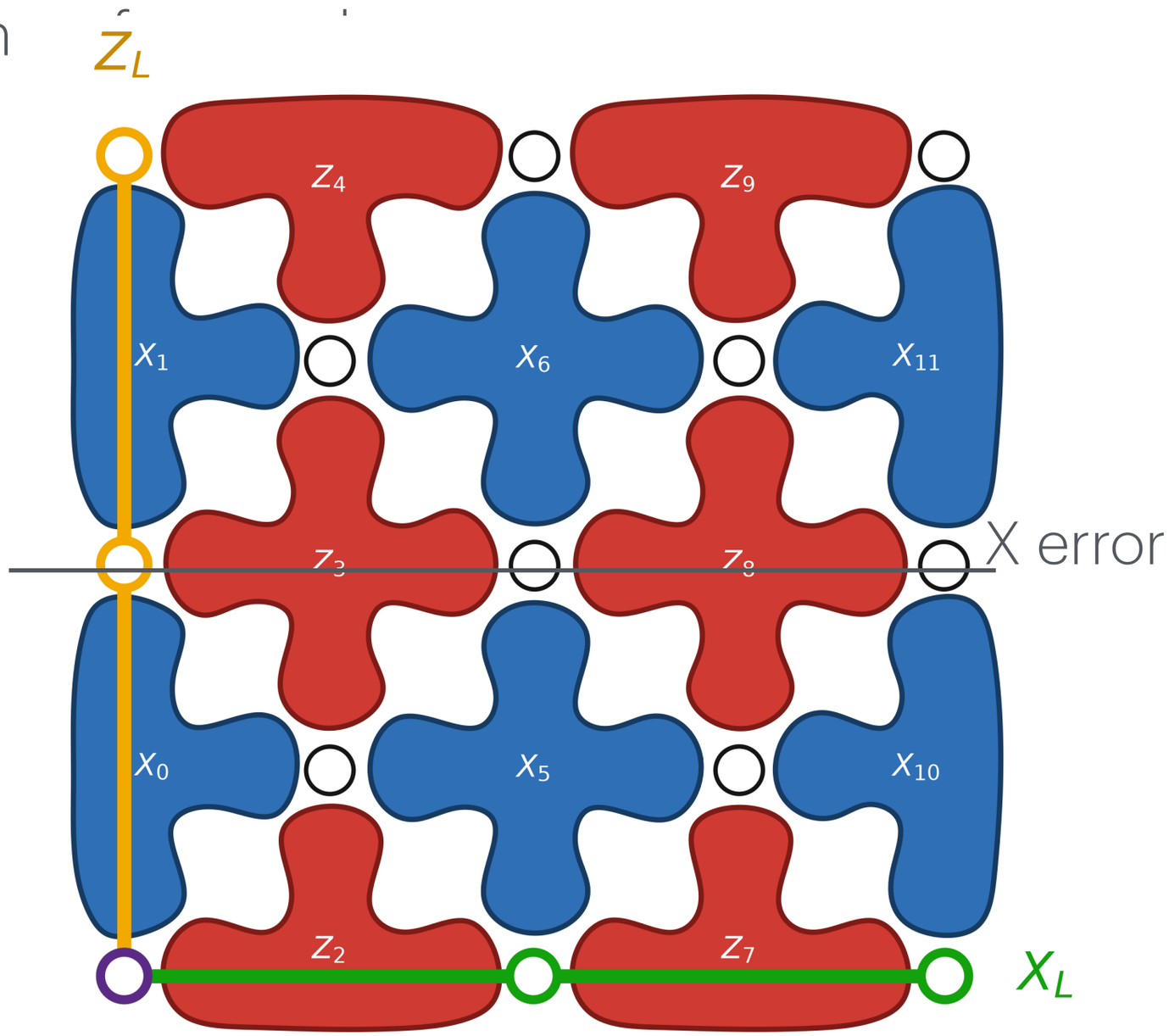
  - Degenerate errors
  - **Correctable errors**
  - An actual logical errors



Surface codes

- There are tree types of error on

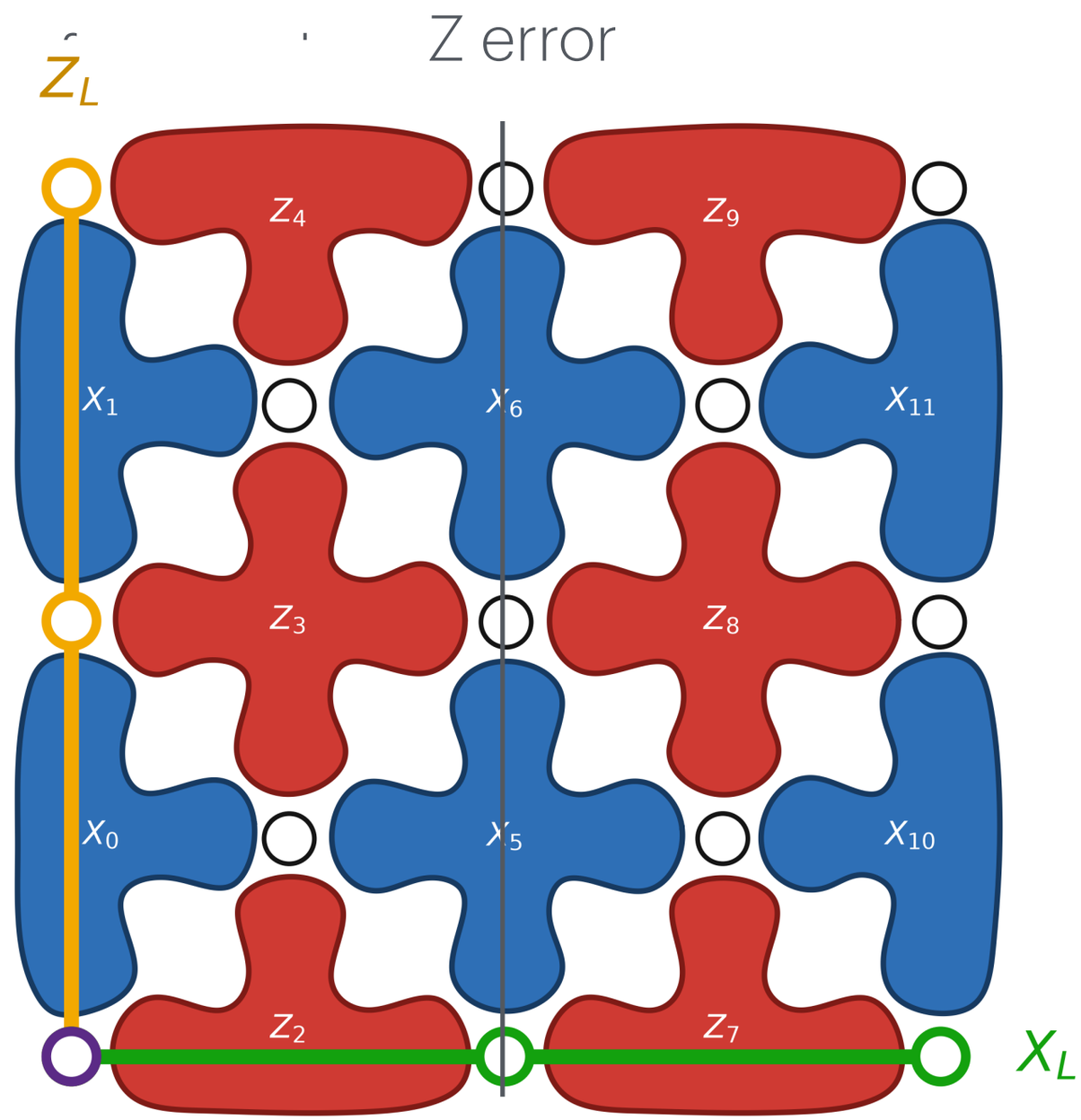
  - Degenerate errors
  - Correctable error
  - **Actual logical errors**



Surface codes

- There are tree types of error on

  - Degenerate errors
  - Correctable error
  - **And actual logical errors**



Surface codes

- Logical errors,
  - Let us suppose the real error is $e^x_{\text{real}} = [0001000000000]$
  - Our decoder, using the syndrome information predicted an error
  - $e^x_{\text{dec}} = [0010000000000]$
- The error is logical if and only if $d := e^x_{\text{real}} \ominus e^x_{\text{dec}} = e^x_{\text{real}} \oplus e^x_{\text{dec}}$
  - Is in $d \in \text{kernelH}_\text{z} \backslash \text{span} H_x$

Decoders

- I did not talk about **decoders**
- A decoder is algorithm that takes into account the **syndrome** S and predict the **most likely** (or close to it) error that occurred e_{pred}
  - **Not always** the error that **actually** that occurred e_{real}
  - **Two error might** produce the **same syndrome** but have different **logical effects,**
    - and the decoder might predict the wrong ones
- **Decoders will be left for your projects**

Decoders

- Your project will be to **choose** a **code**:
  - Classical codes, surface codes, toric code, LDPC
- Simulate an error channel with **Stim**
  - bit flips, phase flips, etc.
  - Circuit-level noise for the brave ones (not recommended)
- Test a **decoder**
- Tesseract, belief propagation, BP+OSD, relay BP, matching
- Compute the **LER curves**

Decoders

- The project will be evaluated based on:
  - Understanding of the code used, rate, distance, and limitations
  - Understanding of the error model used
  - **Most importantly, understanding of the decoder used — especially its limitations**
- Better understanding of the concepts will be valued over choosing hard codes and decoders

Class 3

- Noise models
  - Code-capacity
  - Circuit-level noise
- STIM simulation
- 3 qubit code and a decoder

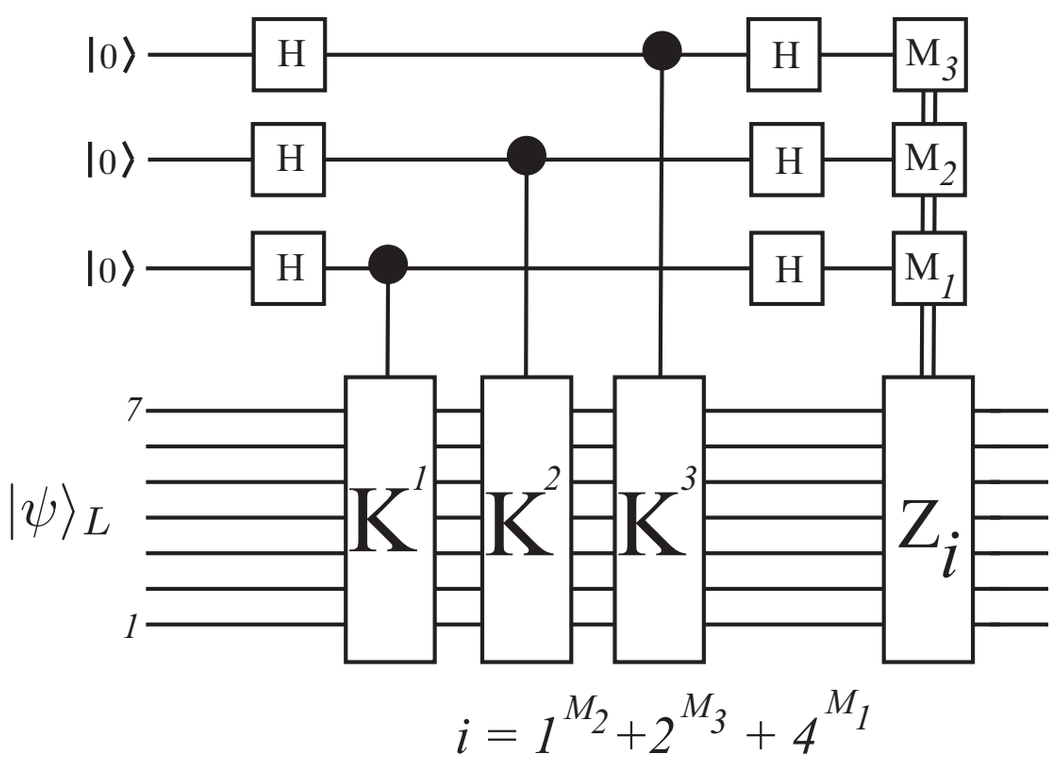
Noise models

- There are two types of errors affecting qubits
  - $\sigma_x|0\rangle \to |1\rangle, \sigma_x|1\rangle \to |0\rangle$
  - $\sigma_z|0\rangle \to |0\rangle, \sigma_z|1\rangle \to -|1\rangle$
  - and a $Y = i\sigma_x\sigma_z$
- Do they all occur with the same probability ?

$$\sigma_0 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad \sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

Bit flip

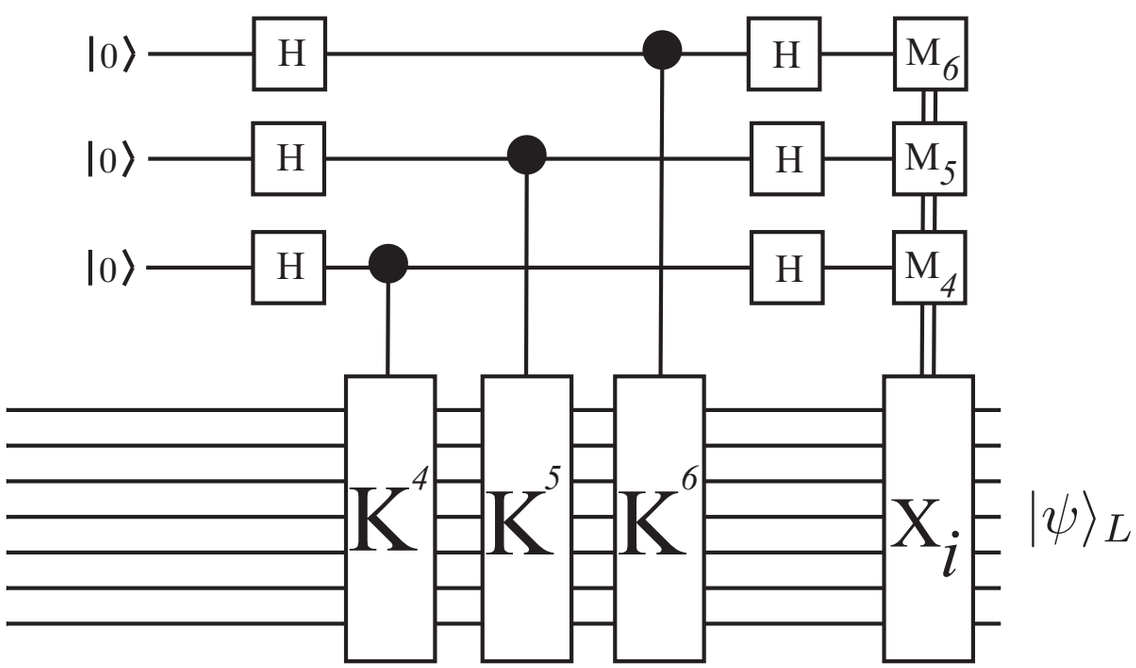
- Only X errors
- Each data qubit i has an error X_i with probability p.
- Ancillas are perfect
- The syndromes are then $S_{\mathrm{x}} = H_z e_{\mathrm{x}}$



$$H_z = \begin{matrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \end{matrix}$$

Phase flip

- Only Z errors
- Each qubit i has error Z_i with probability p.
- All errors are independent
- The syndromes are then $\vec{S}_z = H_x \vec{e}_z$



$$i = 1^{M_5} + 2^{M_6} + 4^{M_4}$$

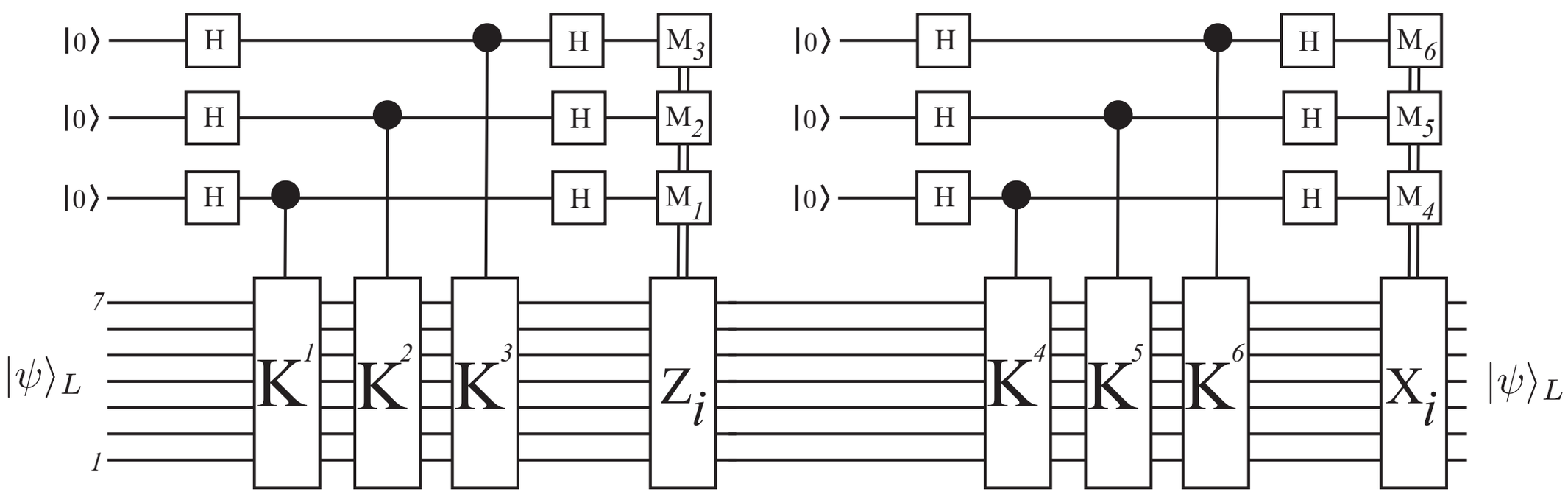
$$H_x = \begin{matrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \end{matrix}$$

Code capacity noise

- Each data qubit i has an error X_i, Y_i or Z_i with probability p/3.
  - Also known as the "Depolarizing channel"

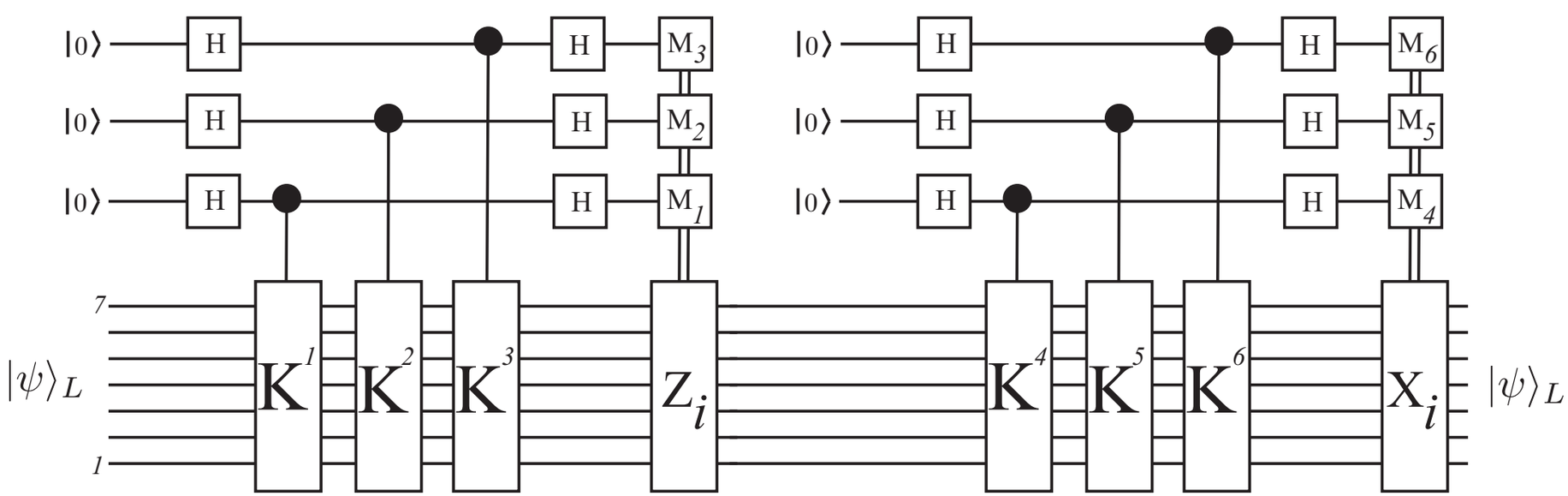
$$\sigma_0 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad \sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

- No errors with probability 1-p



Capacity noise

- All errors are independent.
- Ancilla qubits are noiseless.
- The syndromes are then $\vec{S}_{\mathrm{x}} = H_z \vec{e}_{\mathrm{x}}$ and $\vec{S}_{\mathrm{z}} = H_x \vec{e}_{\mathrm{z}}$

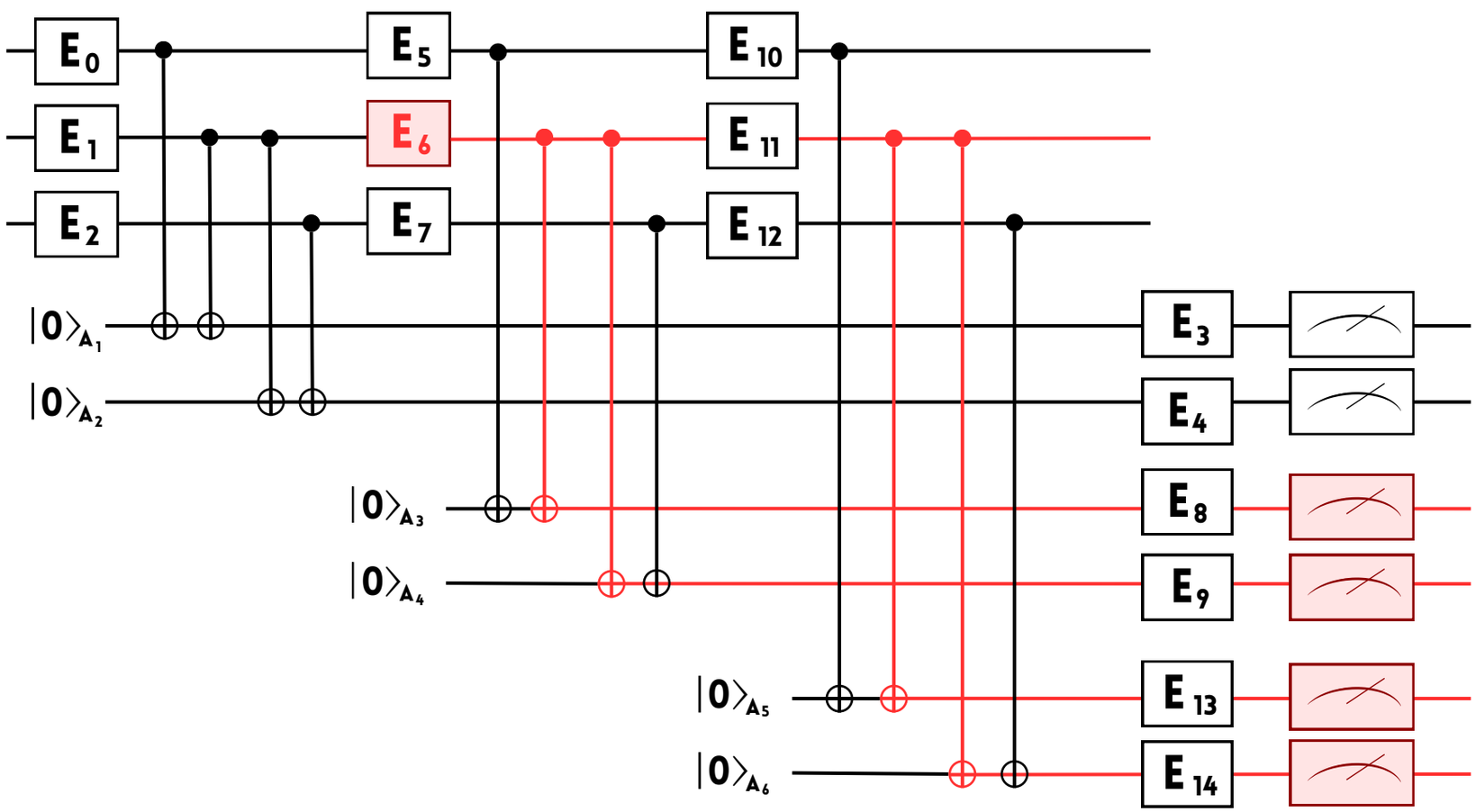


Circuit level noise

- There is no real justification for why we assume the ancilla to be noiseless or the errors to be independent.

- Circuit-level noise is a noise model built from the circuit itself

- every operation can introduce errors
  - State preparation
  - gates,
  - resets,
  - measurements
  - idle steps



Circuit level noise

- How do we simulate this ?
- Before 2023
- After 2023 Stim

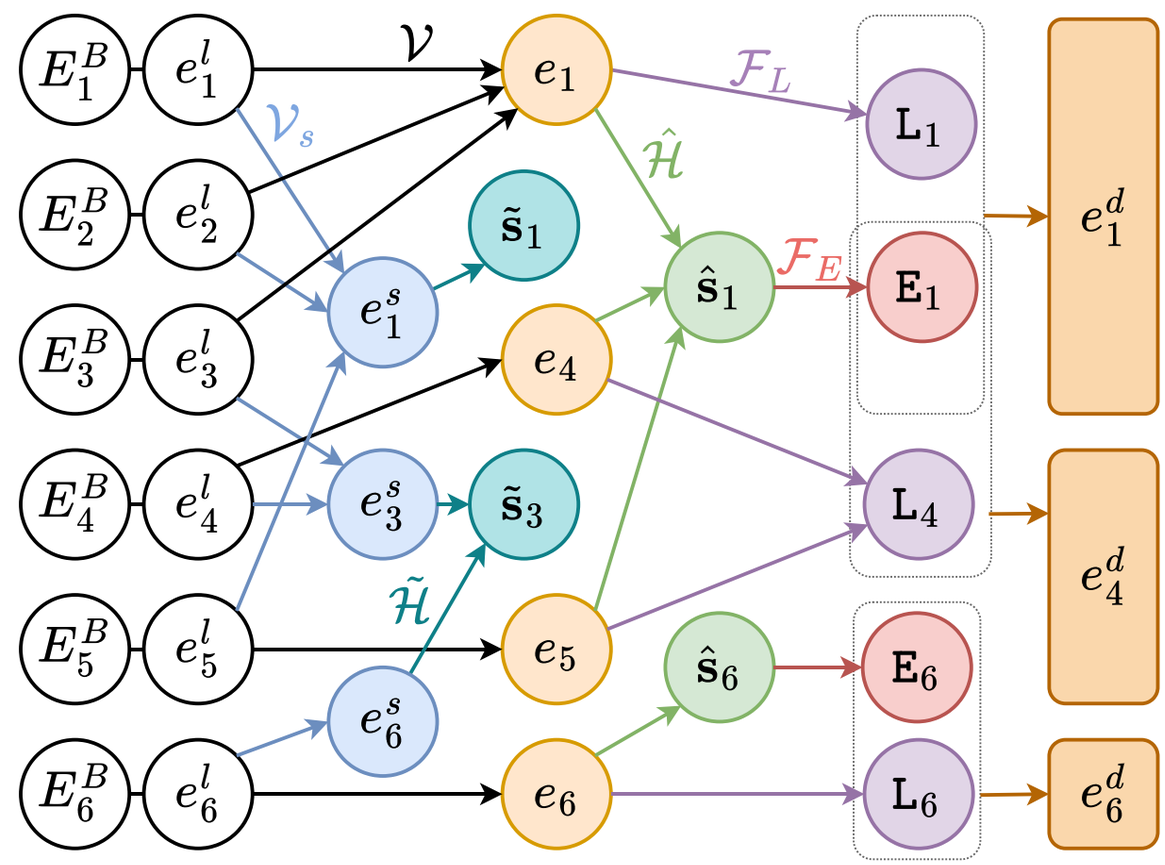


FIGURE 3. **Relations between the different quantities of interest. For simplicity, every error represented is assumed to be a base error. The notation is explained in the last paragraph of Section III.**

Circuit level noise

- Stim is a fast simulator of stabilizer circuits.
- Stim allows us to do exactly this: simulate error propagation on circuits.
- We can simulate simple error models like bit-flip, phase-flip, or code capacity noise, or circuit-level noise.
- It is available for Python and c++.

Circuit level noise

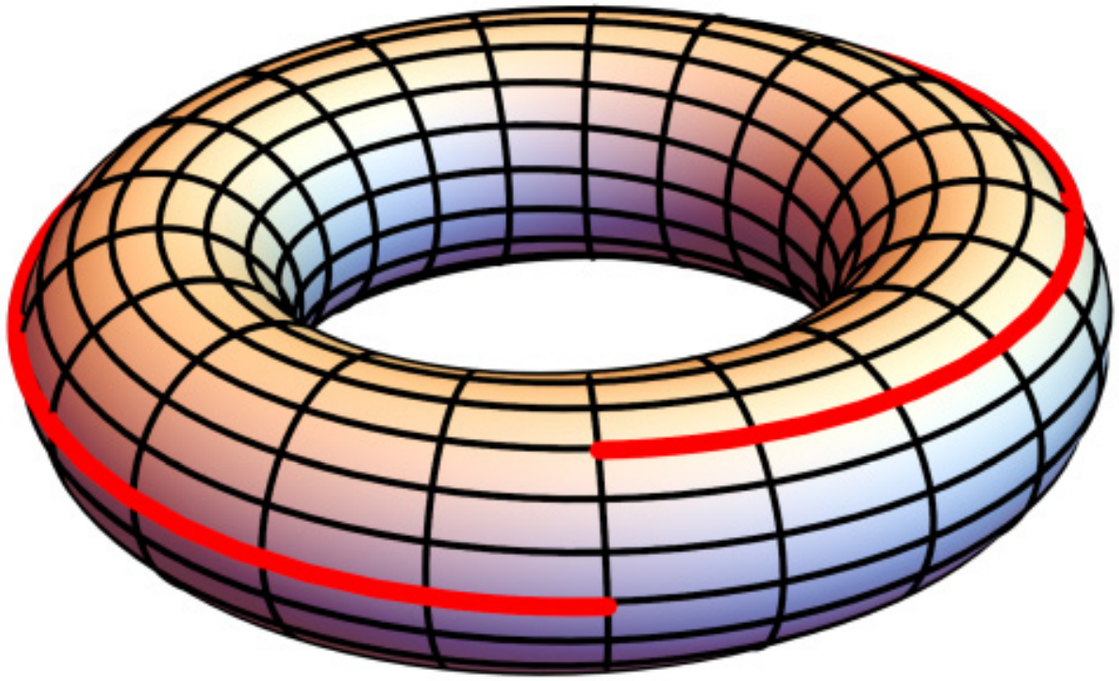
- Stim doesn't use the parity checks matrices H_x andH_z
- It uses the DEM matrix H_{DEM}
- For each error (an abstract event in spacetime), it gives the detectors it activates, and its logical effect
- For code capacity , each event is an error X Z (or both) on qubit i , thus we recover
  - $H_{\mathrm{DEM}} = \begin{pmatrix} H_X & 0 \\ 0 & H_Z \end{pmatrix}$

Circuit level noise

- Stim cannot simulate T-gates (non-Clifford circuits), or circuit with non-determinist observables
  - This restriction is what lets Stim be fast
- $T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}$
- There's a new kid on the block, April 2026: Tsim, from QuEra
  - Tsim is non-Clifford T-gate stabilizer simulator

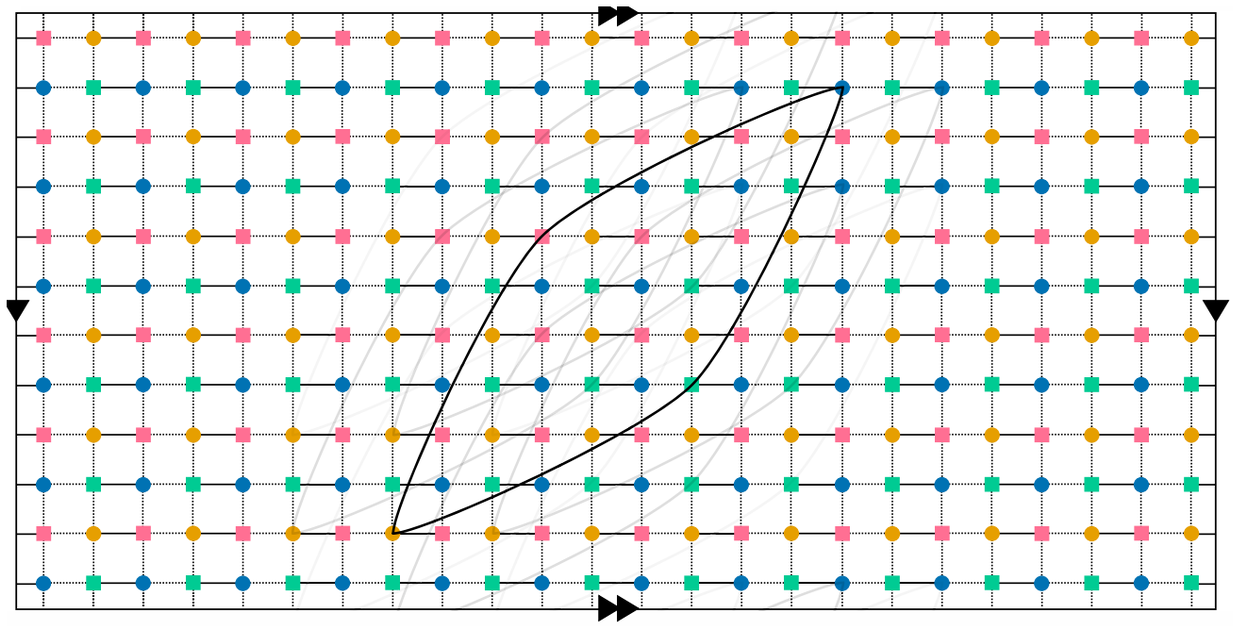
Other codes

- Rotated-surface codes



- Toric code

- Quantum LDPCs



Decoders

- Table decoders require a lot of memory.
  - $2^{\text{number of stabilizers}}$
- **Belief propagation** (BP) is a set of decoders designed for classical LDPC codes.
  - Works iteratively.
  - Does not work well on surface codes. Some variation for quantum codes
- **Search-based decoders** turn the decoding problem into a search on a graph.
  - There are a few variations designed for quantum codes.
- **Matching decoders** use the fact that surface codes are two-dimensional to define the decoding problem as a matching problem.