

Towards universal quantum computation with QLDPCs

Bruno Coutinho

Structure of the talk

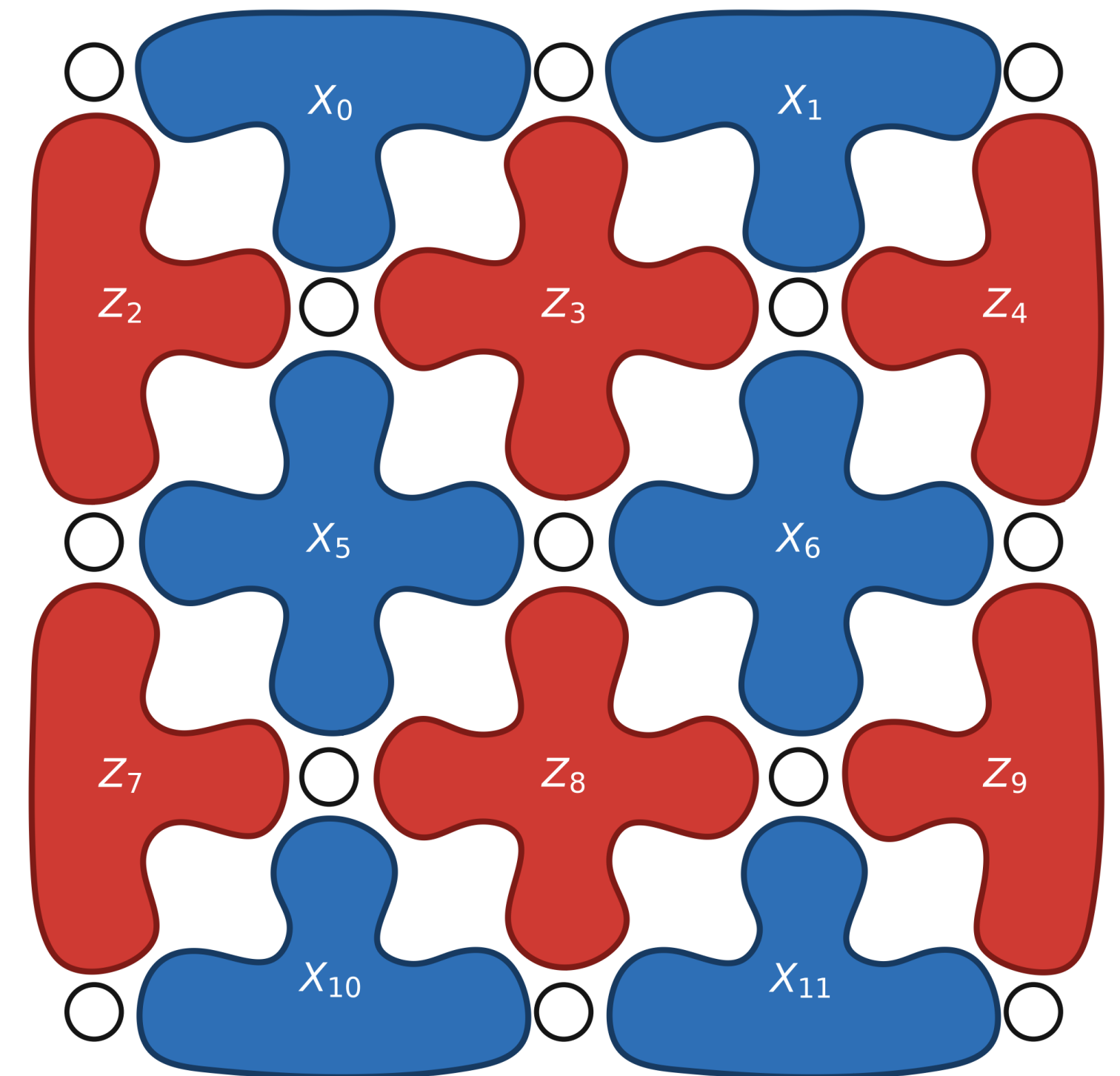
- First I'm going to do a **brief** introduction to **quantum error correction**
 - What it is and how it works
- Then **three differences** between **quantum** and **classical error correction**
- Then quantum **LDPCs**
 - Why they are so popular right now
- Finally, mention recent advances in **logical operations**

Quantum error correction

- Quantum error correction is very **similar** to classical error correction
- We start with a logical state $|\psi\rangle_L$ of k qubits
- And **encode** those k in a **larger physical state** with $n > k$
- The code is then defined by three terms $[n, k, d]$
 - d is the code distance

$$|\psi\rangle_L = a|0\rangle_L + b|1\rangle_L$$

$$a, b \in \mathbb{C}, |a|^2 + |b|^2 = 1$$



Quantum error correction

- But there are **three** big **differences** between **quantum** and **classical** error correction
 - Quantum **errors** are **different** from classical errors
 - **Quantum states** can't be **measured directly** without collapsing the state
 - **Encoded states** are used for **computation** not for communication

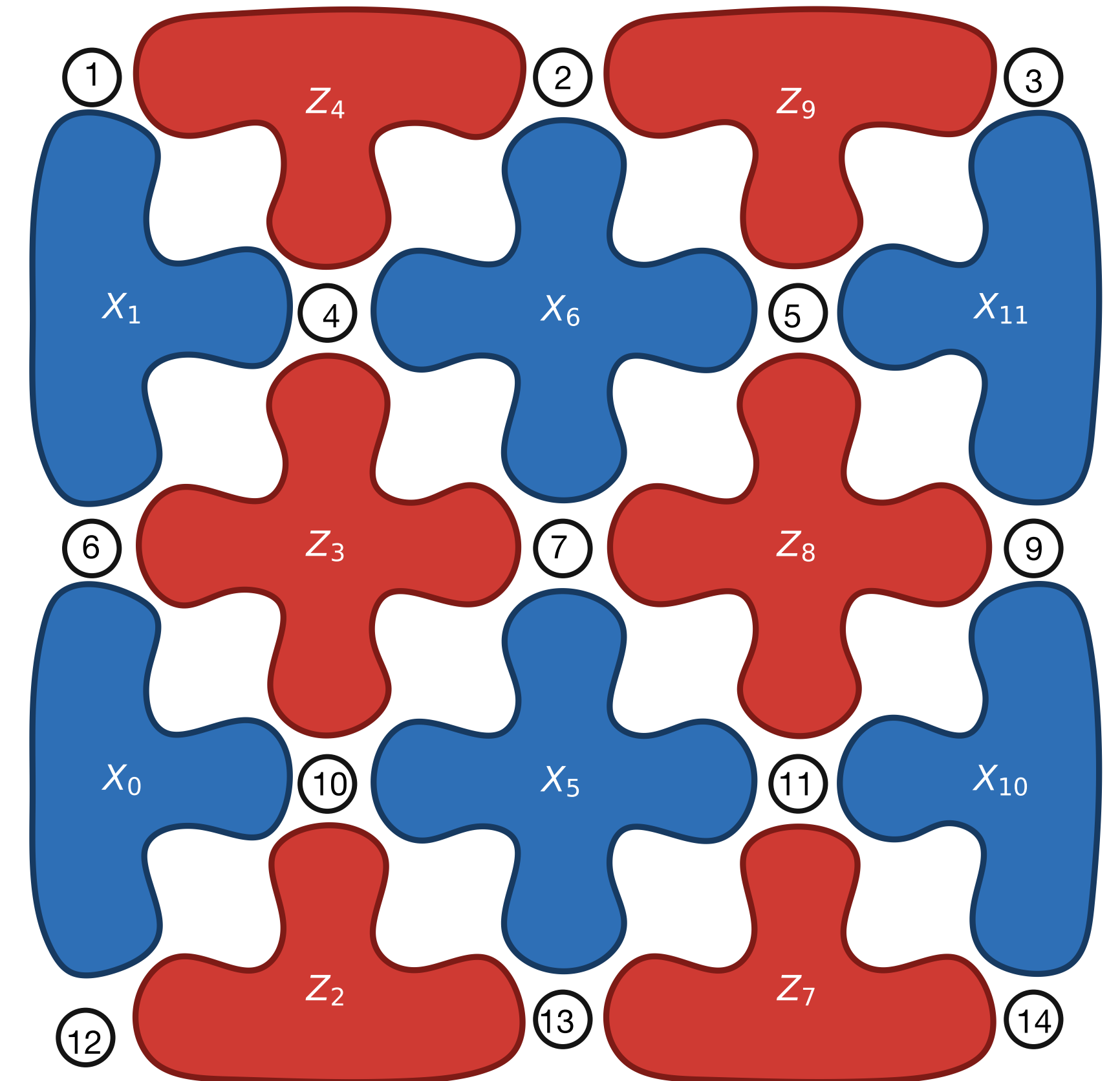
Quantum error correction

1st difference with the classical error correction

- There are two types of quantum error
- Bit-flip X errors
 - $|0\rangle \rightarrow |1\rangle, |1\rangle \rightarrow |0\rangle$
- Phase-flip Z errors
 - $|0\rangle \rightarrow |1\rangle, |1\rangle \rightarrow -|1\rangle$
- Quantum codes need to detect both

$$|\psi\rangle_L = a|0\rangle_L + b|1\rangle_L$$

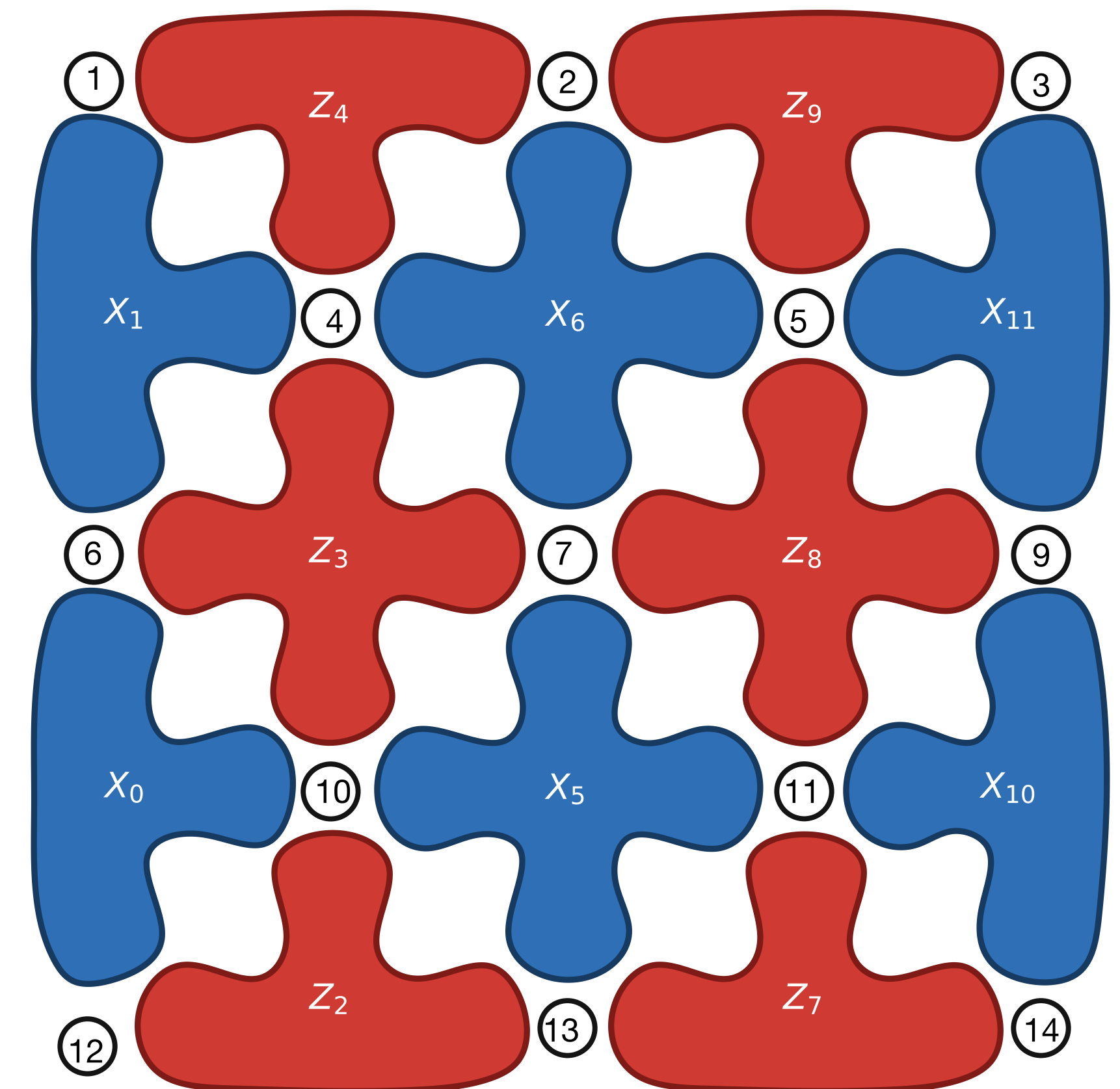
$$a, b \in C, |a|^2 + |b|^2 = 1$$



Quantum error correction

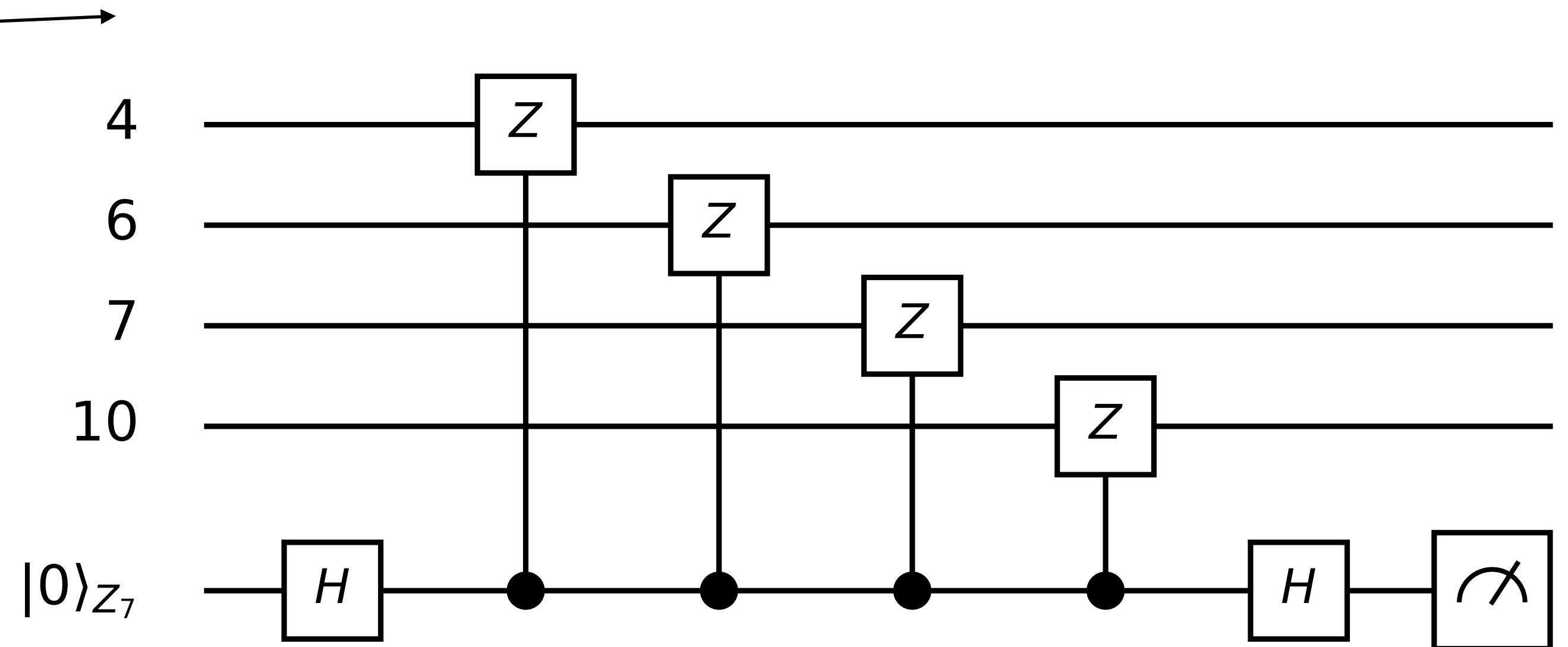
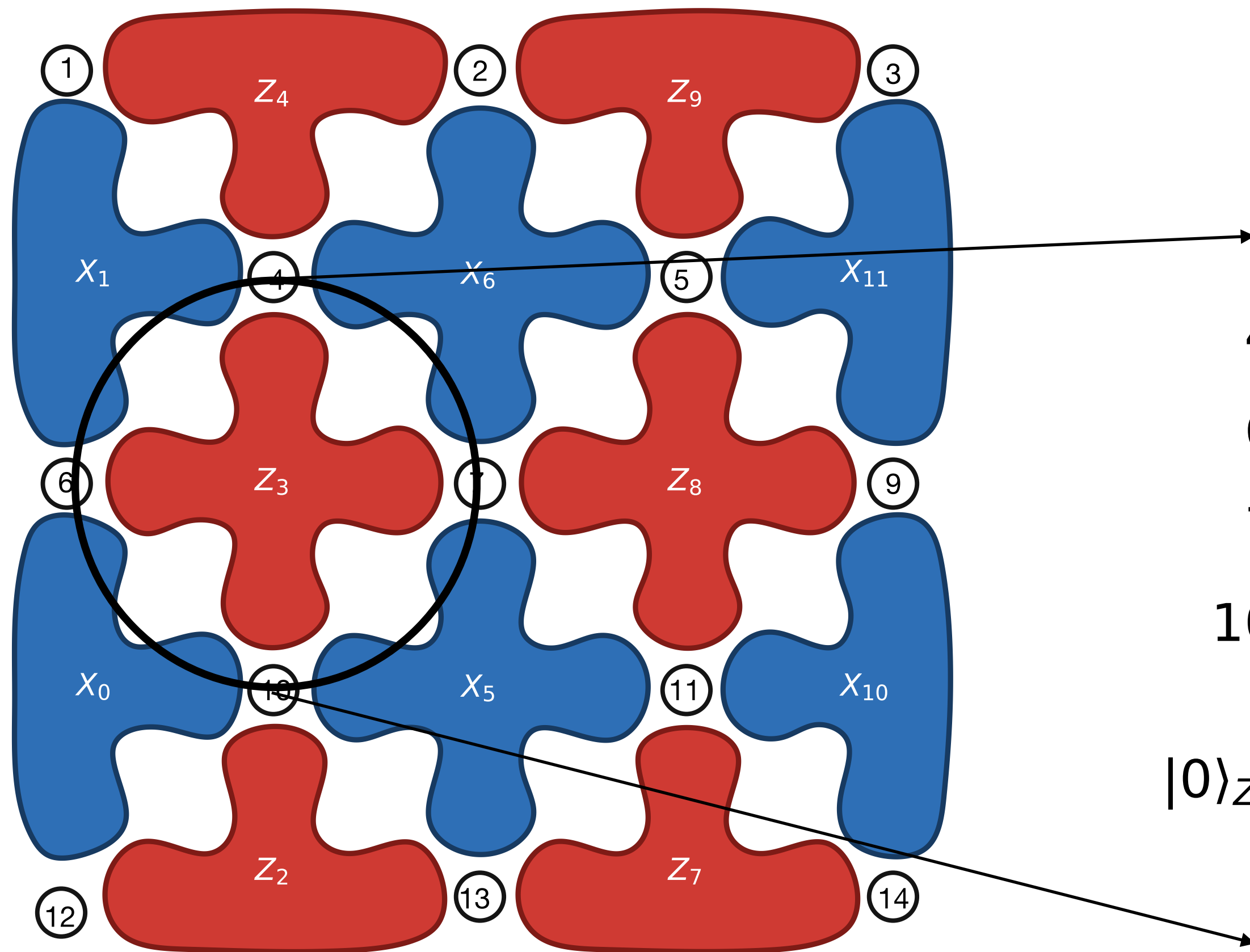
2nd difference with the classical error correction

- We can't measure the state directly without destroying it
- But we can use the stabilizer formalism to find errors
- Since X and Z anti-commute $XZ = -ZX$,
 - and we can actually use a circuit to detect this phase
 - stabilizer can be used to capture errors



Quantum error correction

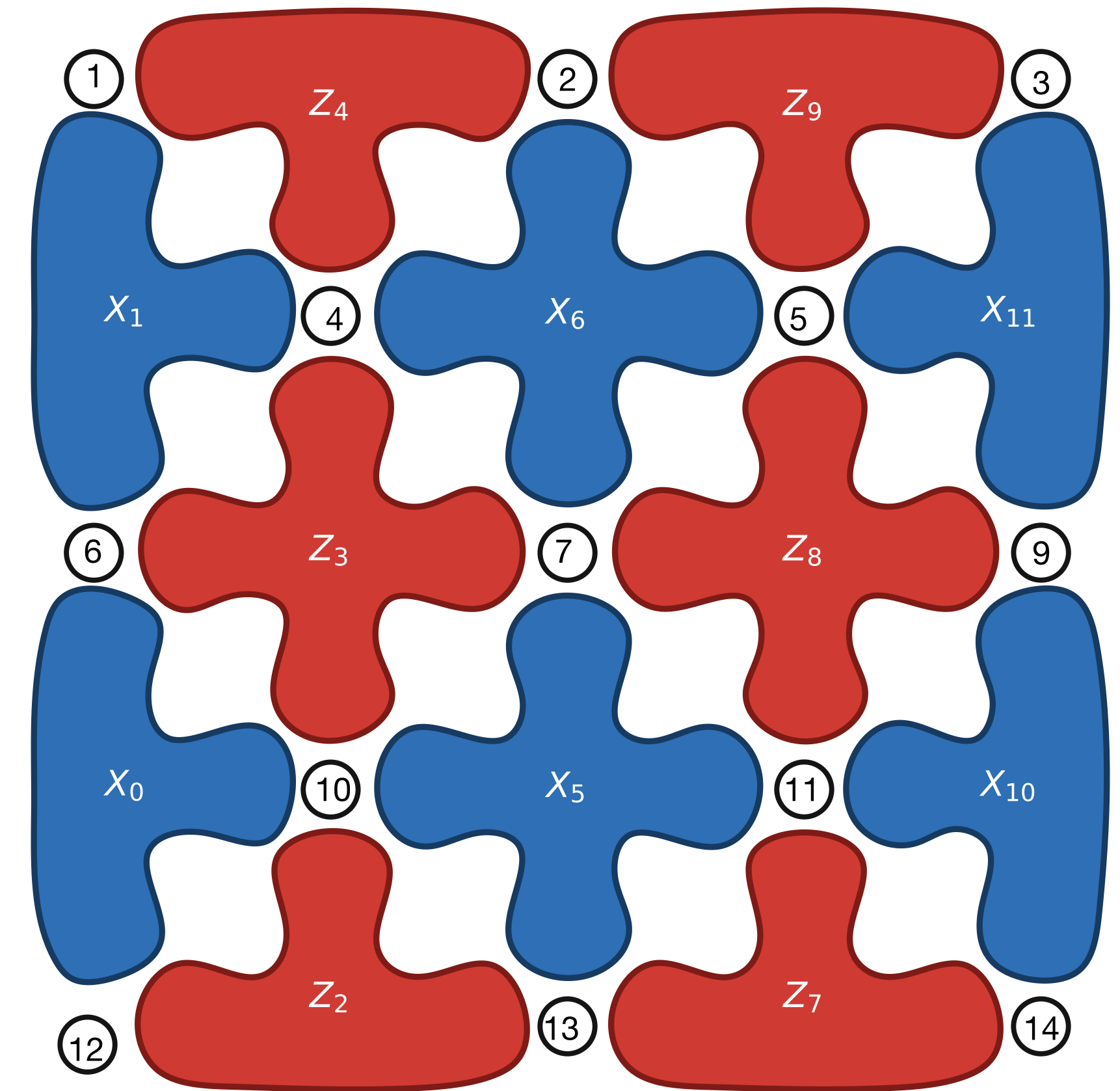
2nd difference with the classical error correction



Quantum error correction

2nd difference with the classical error correction

- Let me give an example $Z_3 := Z_4^p Z_6^p Z_7^p Z_{10}^p$
 - 1 X error in qubit 4 doesn't commute with Z_3 and will be detected
 - 2 X errors (in qubits 4 and 6) will commute with Z_3 and not be detected
- Important X and Z stabilizers must always **commute**
- In this case they do since they share two qubits



Quantum error correction

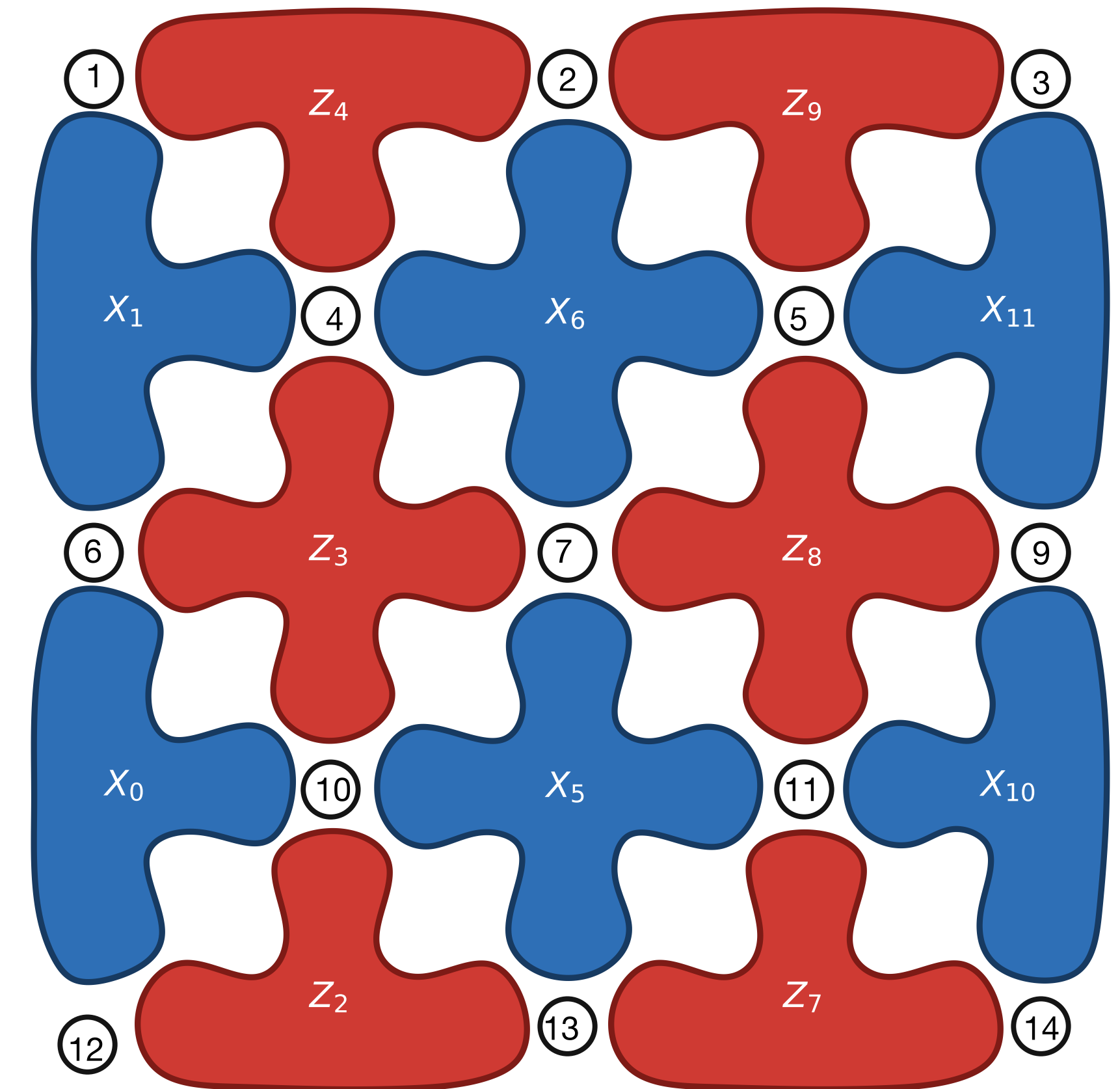
2nd difference with the classical error correction

- Important X and Z stabilizers must always **commute**

- $H_x H_z^T = 0$

$$H_Z = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$

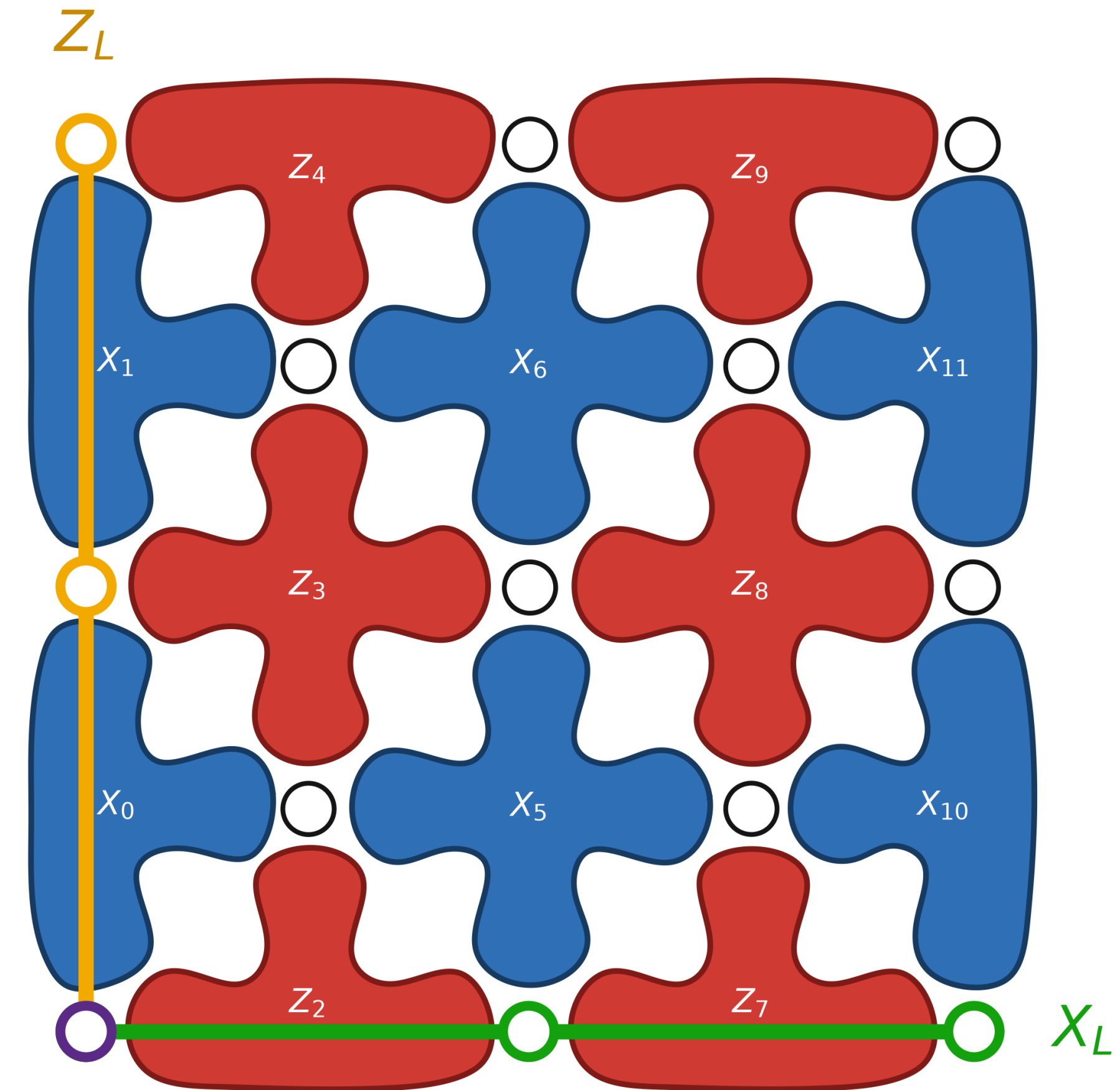
$$H_X = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}$$



Quantum error correction

2nd difference with the classical error correction

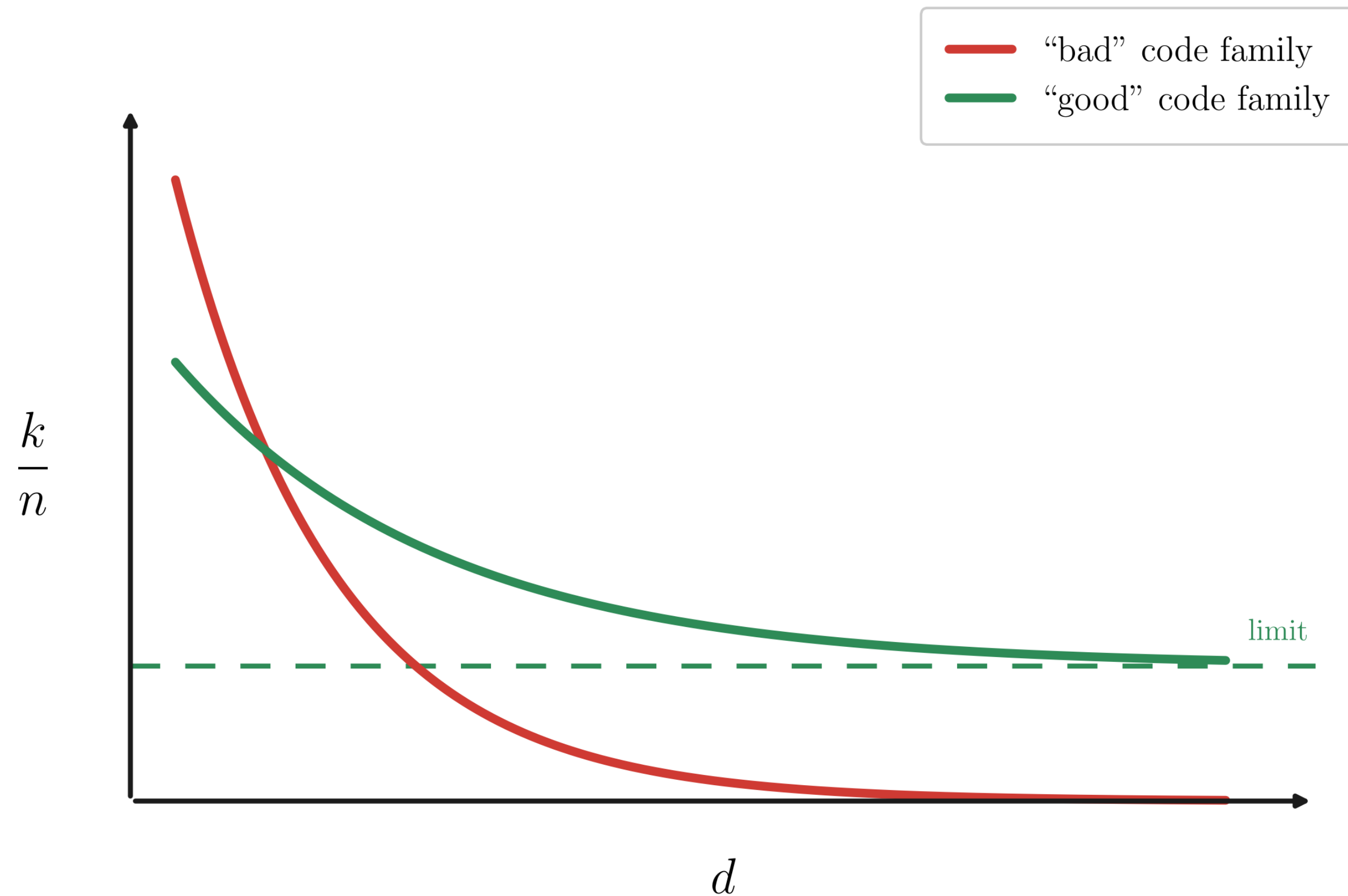
- OK, we have a way to find errors, how do we measure the logical information?
- There are two logical measurements for a single qubit
- Once again both of these need to commute with the stabilizers



Quantum LDPCs and good codes

- A “**good**” code in information theory is part of a code construction, such that
- For **any distance** d , there is a code in the family with a **finite rate** k/n
- **Classical LDPCs** (from low-density parity check) can be “**good**” codes
- **Question:** Given the commutation constraint, $H_x H_z^T = 0$, are there good quantum LDPC codes?

Quantum LDPCs and good codes



- **Question:** Given the commutation constraint, $H_x H_z^T = 0$, are there good quantum LDPC codes?

Quantum LDPCs and good codes

- **Question:** Are there “good” quantum LDPC codes?
- Yes, proven in 2022

Asymptotically Good Quantum and Locally Testable Classical LDPC Codes

Pavel Panteleev and Gleb Kalachev*

January 24, 2022

Abstract

We study classical and quantum LDPC codes of constant rate obtained by the lifted product construction over non-abelian groups. We show that the obtained families of quantum LDPC codes are asymptotically good, which proves the qLDPC conjecture. Moreover, we show that the produced classical LDPC codes are also asymptotically good and locally testable with constant query and soundness parameters, which proves a well-known conjecture in the field of locally testable codes.

Quantum LDPCs and practical good codes

- IBM's BB codes were published around one year later

Article

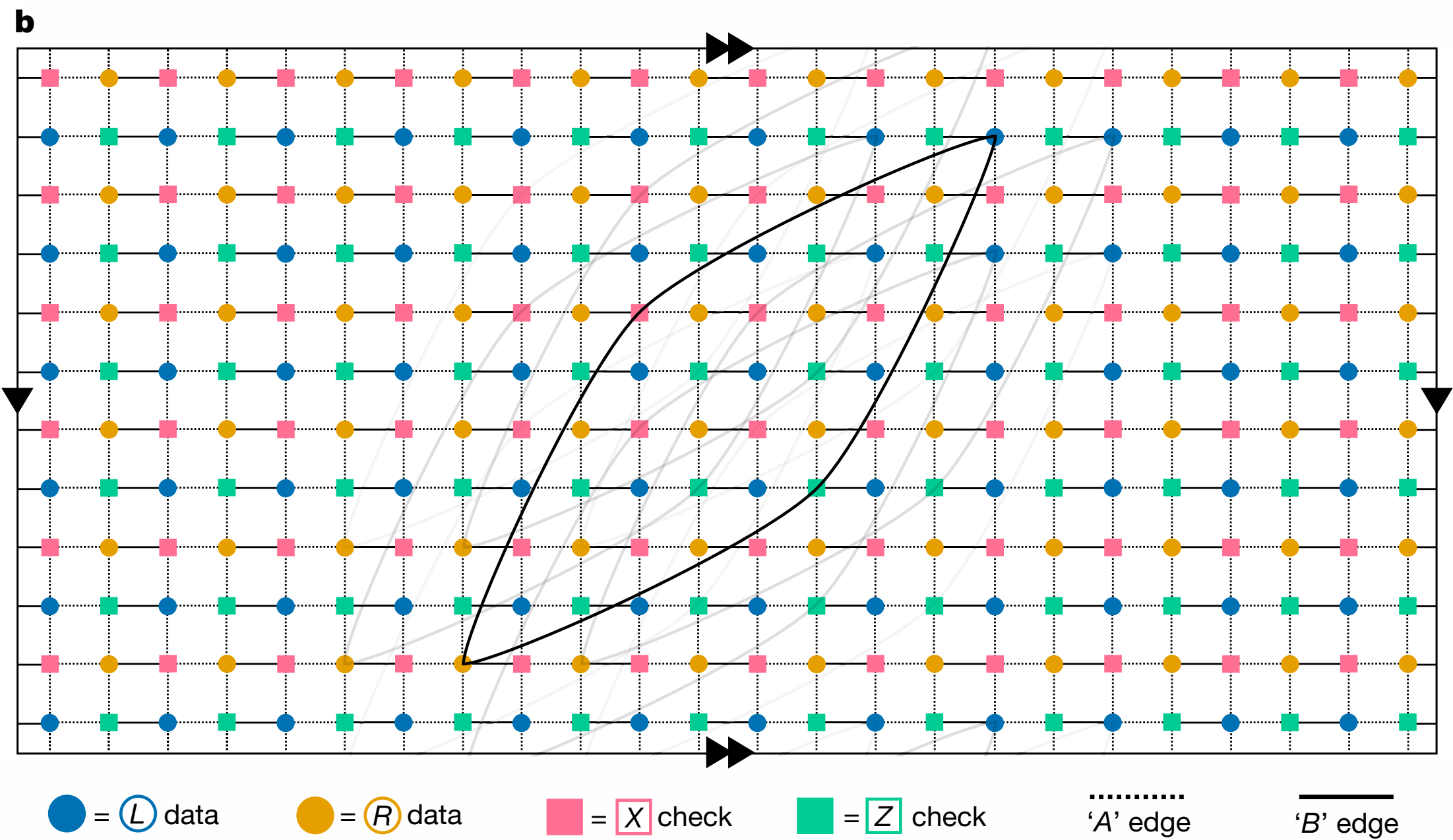
High-threshold and low-overhead fault-tolerant quantum memory

<https://doi.org/10.1038/s41586-024-07107-7>

Received: 25 August 2023

Accepted: 23 January 2024

Sergey Bravyi¹, Andrew W. Cross¹, Jay M. Gambetta¹, Dmitri Maslov¹✉, Patrick Rall² & Theodore J. Yoder¹



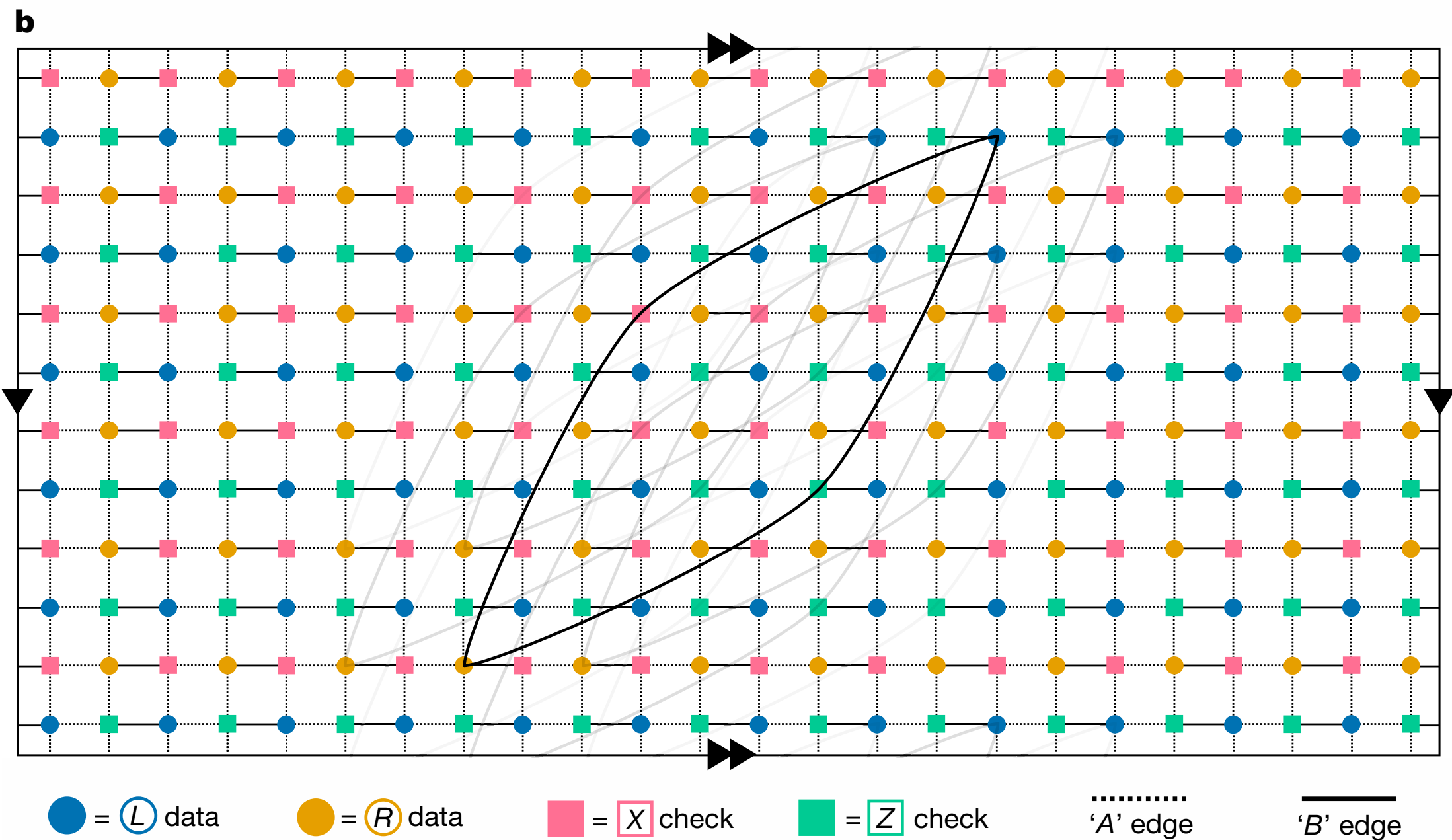
Quantum LDPCs and practical good codes

- IBM's BB codes were published around one year later

Article

Table 1 | Performance of BB codes

$[[n,k,d]]$	Net encoding rate, r	Circuit-level distance, d_{circ}	Pseudo-threshold, p_0	$p_L(10^{-3})$	$p_L(10^{-4})$
$[[72,12,6]]$	1/12	≤ 6	0.0048	7×10^{-5}	7×10^{-8}
$[[90,8,10]]$	1/23	≤ 8	0.0053	5×10^{-6}	4×10^{-10}
$[[108,8,10]]$	1/27	≤ 8	0.0058	3×10^{-6}	1×10^{-10}
$[[144,12,12]]$	1/24	≤ 10	0.0065	2×10^{-7}	8×10^{-13}
$[[288,12,18]]$	1/48	≤ 18	0.0069	2×10^{-12}	1×10^{-22}



Quantum LDPCs and practical good codes

3rd difference with the classical error correction

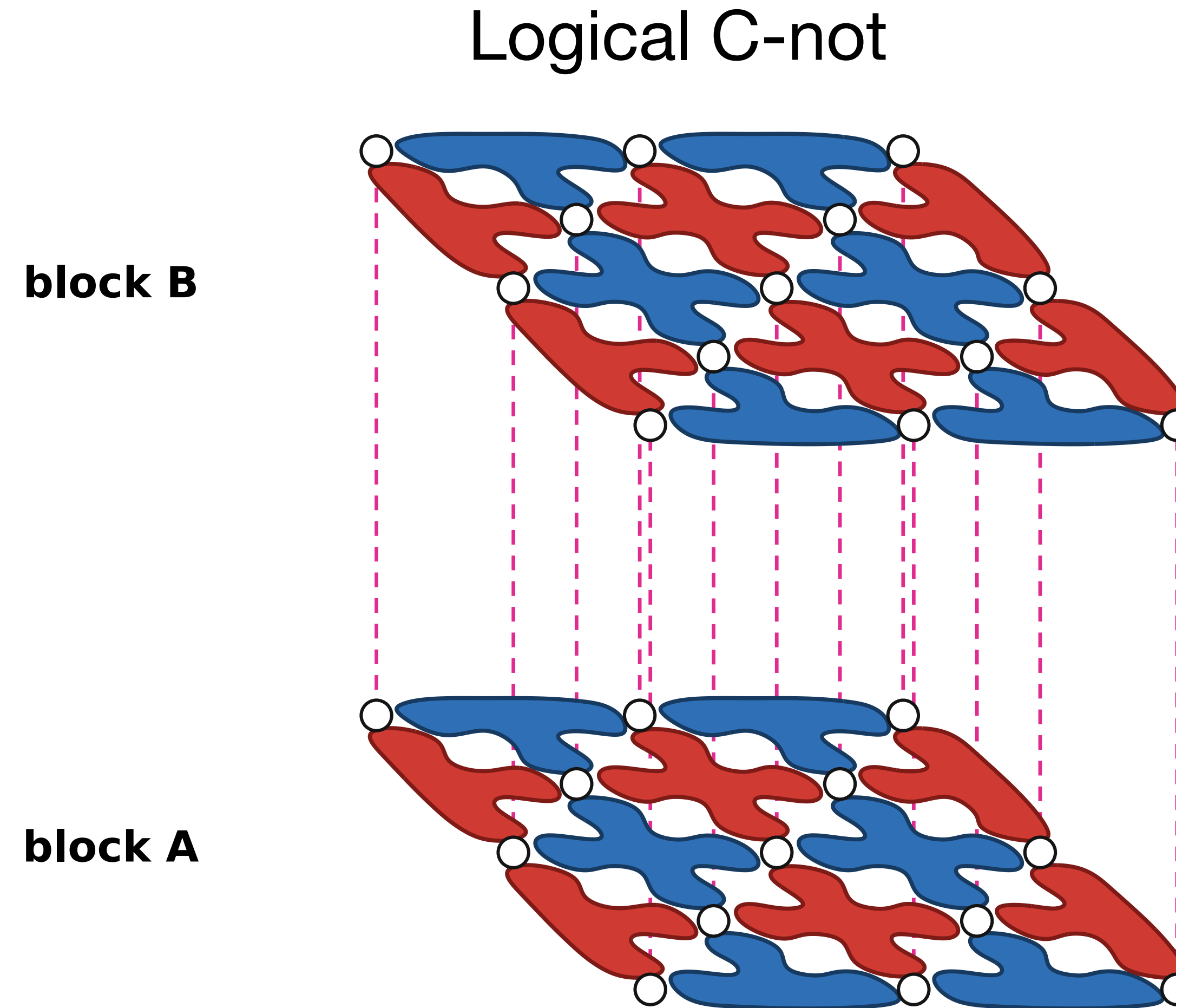
- But **another question remained** until very recently
- Can we **use** these codes for **computation**?
- Doing quantum logical **operations** on **encoded** quantum states **is not trivial**
 - Operation adds errors that need to be corrected
 - Whatever we do to a logical state needs to preserve the superposition
 - And we need to be able to do an **universal set of gate**

Universal set of quantum gates

Example	Clifford gates	T-gates
Single qubit gates	$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ $S = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/2} \end{pmatrix}$	$T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}$
Two qubit gates	$\text{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$	

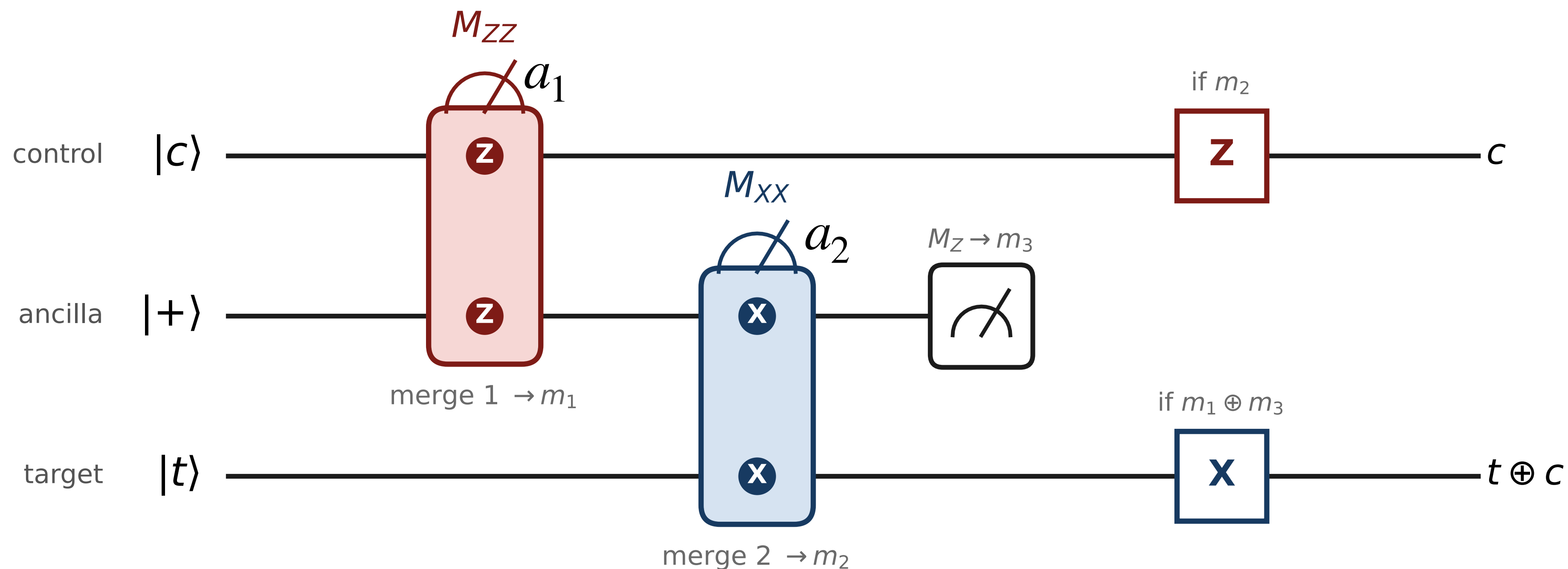
Solution 1 - Transversal gates

- A gate is **transversal** if it factorizes as a tensor product of **simple qubit** gates
- **No** quantum error-correcting code can have a **transversal gate set** that is both **universal** and **fault-tolerant** for a fixed code space¹



¹ B. Eastin and E. Knill, "Restrictions on Transversal Encoded Quantum Gate Sets," Phys. Rev. Lett. 102, 110502 (2009).

Solution 2 - Lattice surgery (for surface codes)

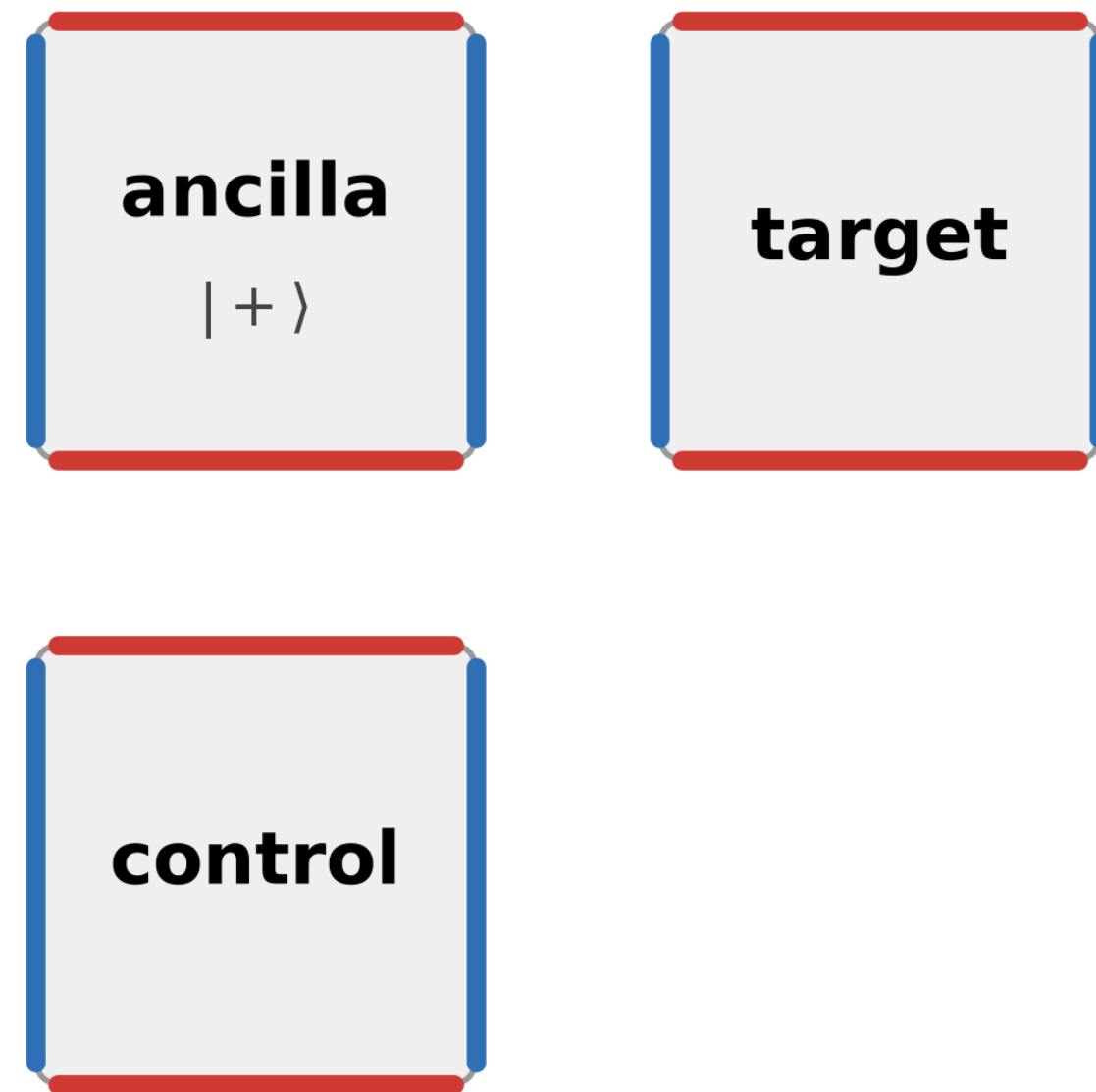


$$M_{ZZ} \rightarrow \begin{cases} |00\rangle\langle 00| + |11\rangle\langle 11| & \text{for } a_1 = 0 \\ |01\rangle\langle 01| + |10\rangle\langle 10| & \text{for } a_1 = 1 \end{cases} \quad M_{XX} \rightarrow \begin{cases} |++\rangle\langle ++| + |--\rangle\langle --| & \text{for } a_2 = 0 \\ |-+\rangle\langle -+| + +- \rangle\langle +-| & \text{for } a_2 = 1 \end{cases}$$

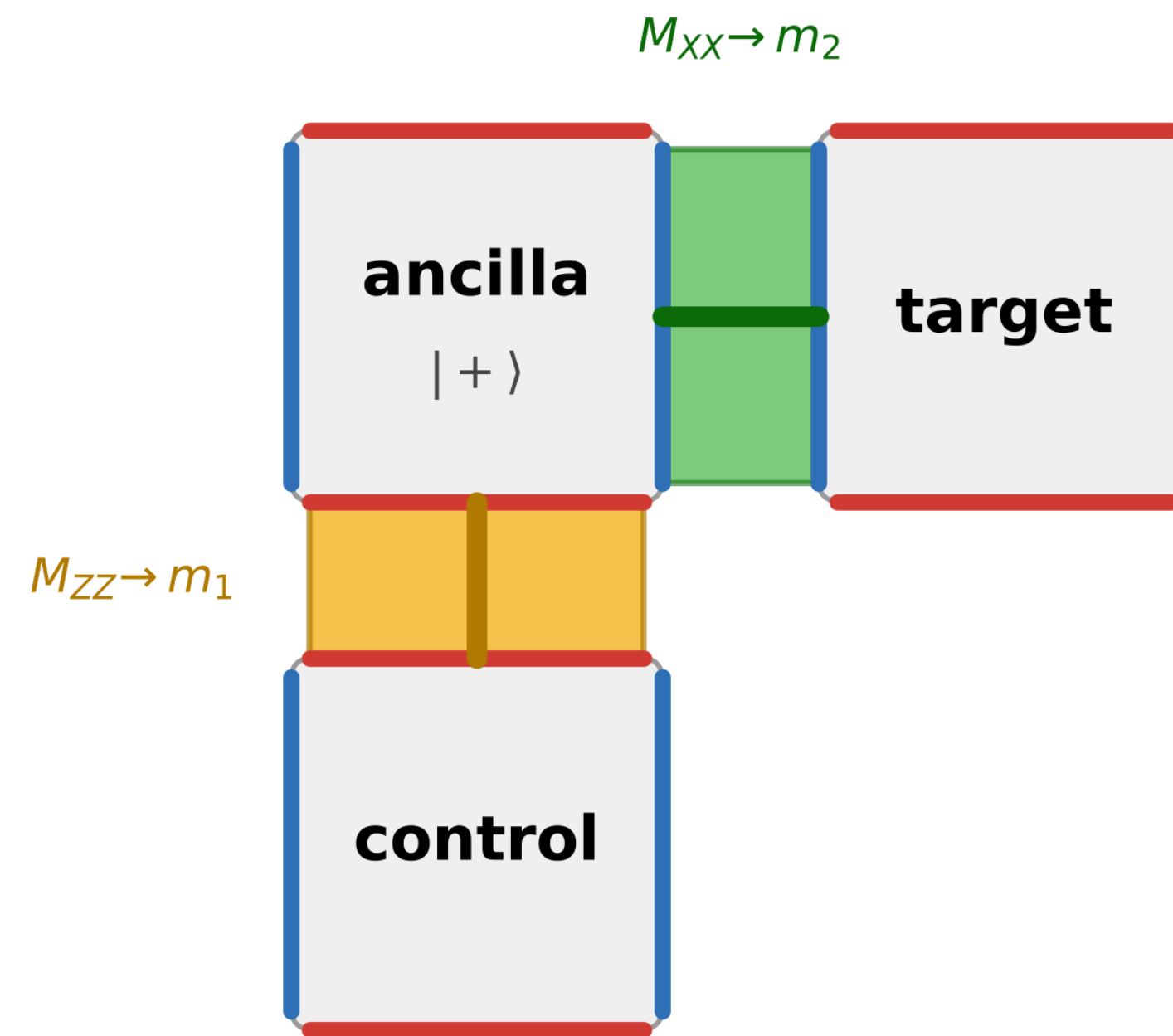
Solution 2 - Lattice surgery (for surface codes)

① initialise

three separate patches

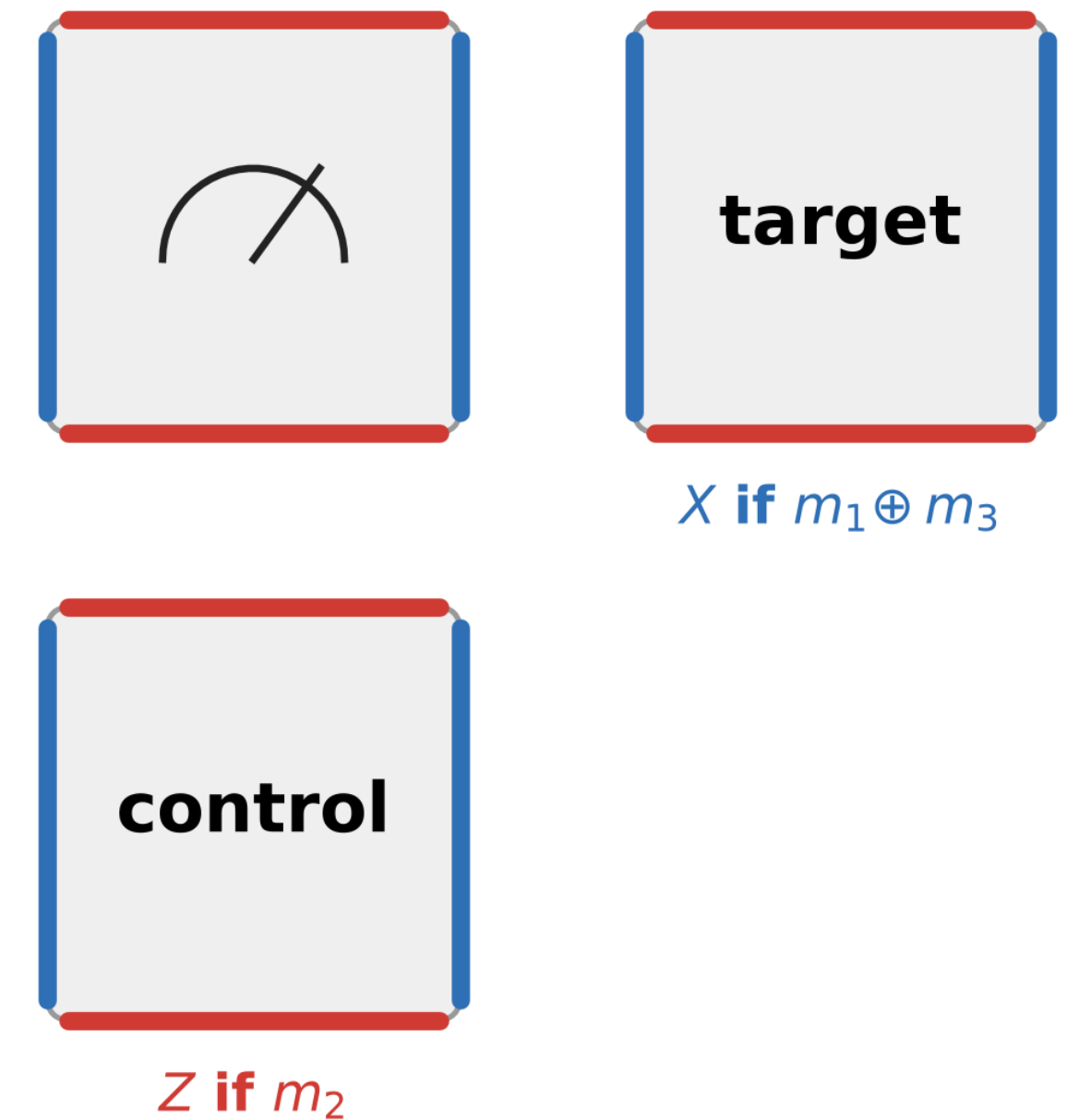


② merge



③ split & measure ancilla

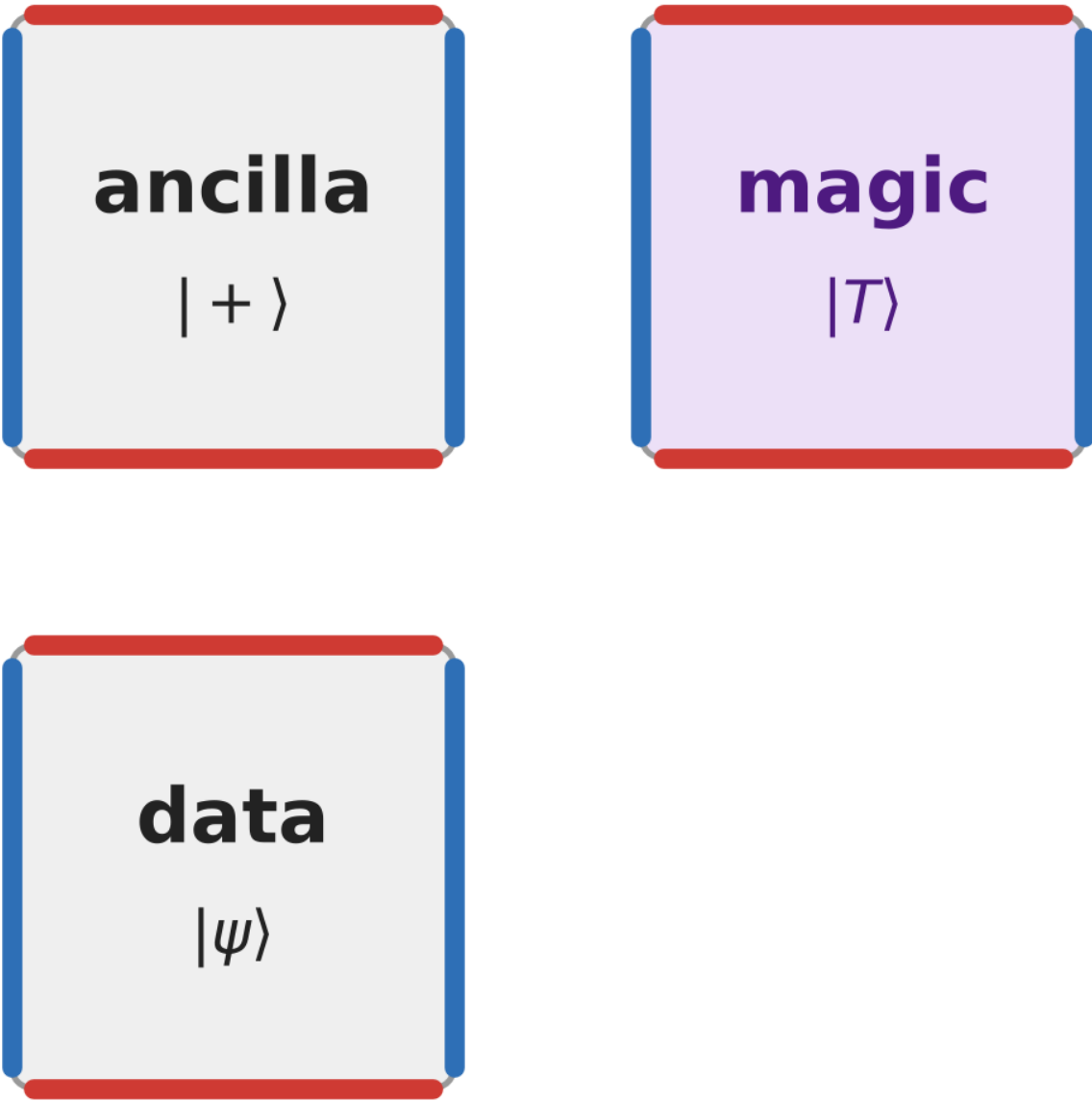
measure ancilla $\rightarrow m_3$



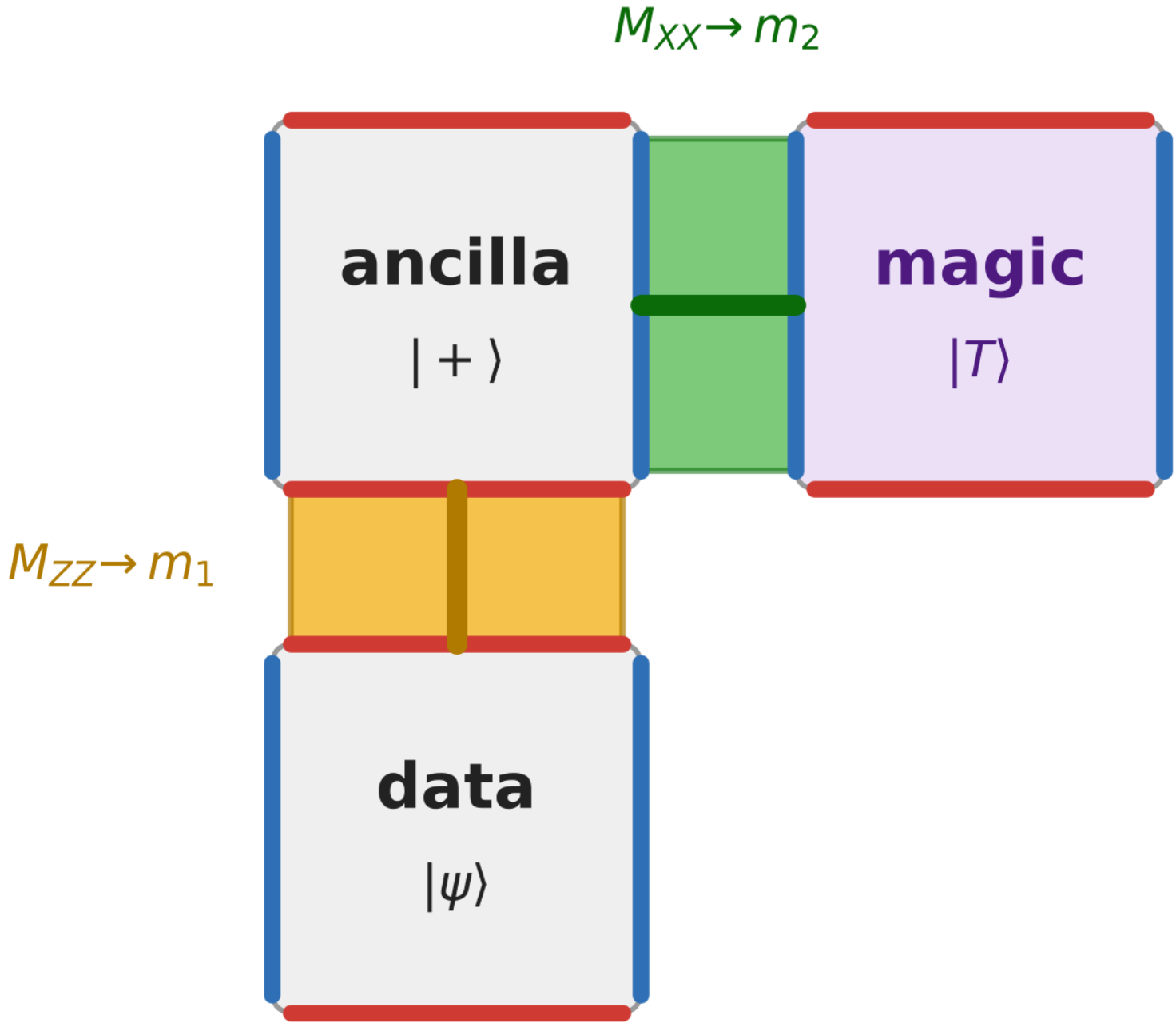
Solution 3 - T-gate injection

① initialise

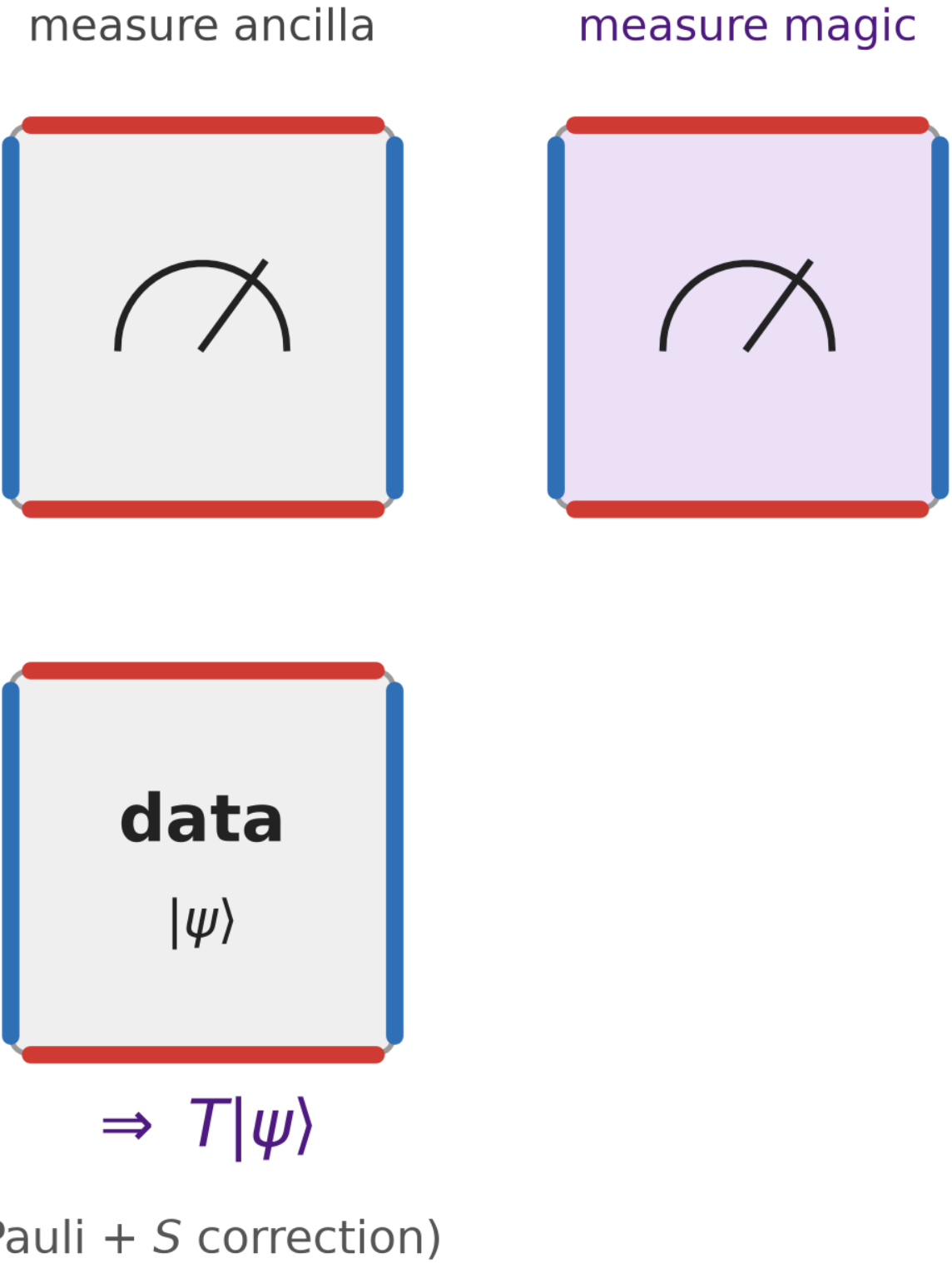
$|T\rangle = \frac{1}{\sqrt{2}} (|0\rangle + e^{i\pi/4} |1\rangle)$



② merge



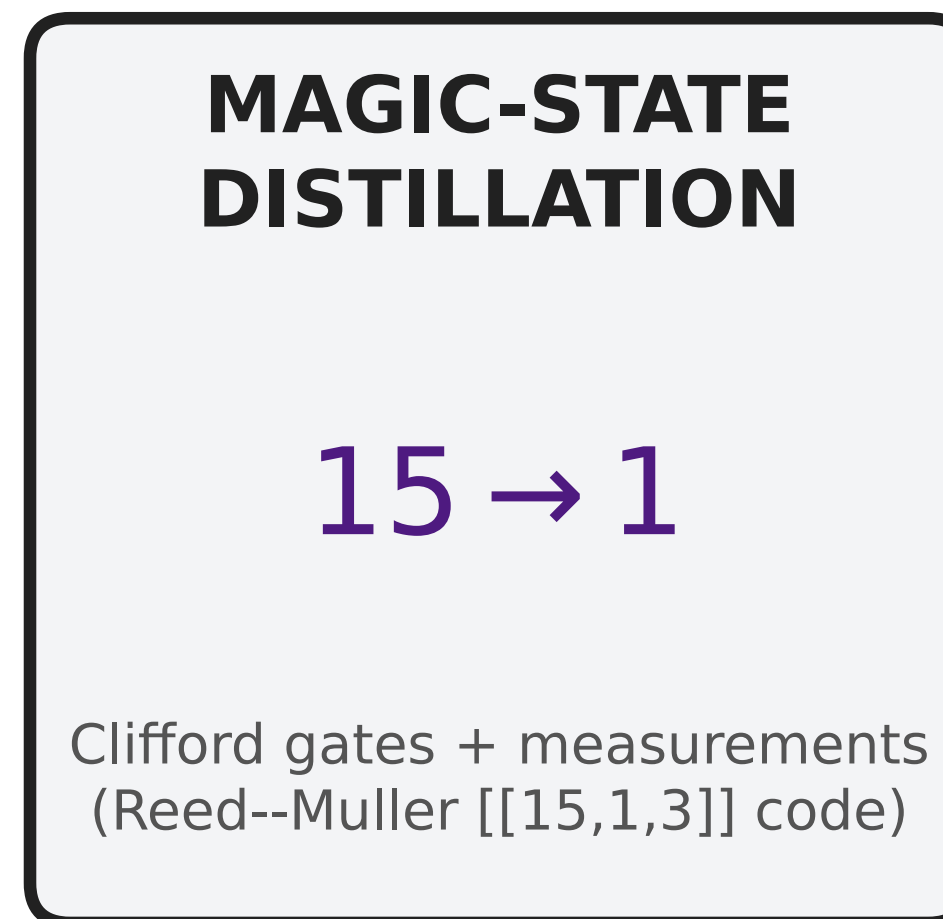
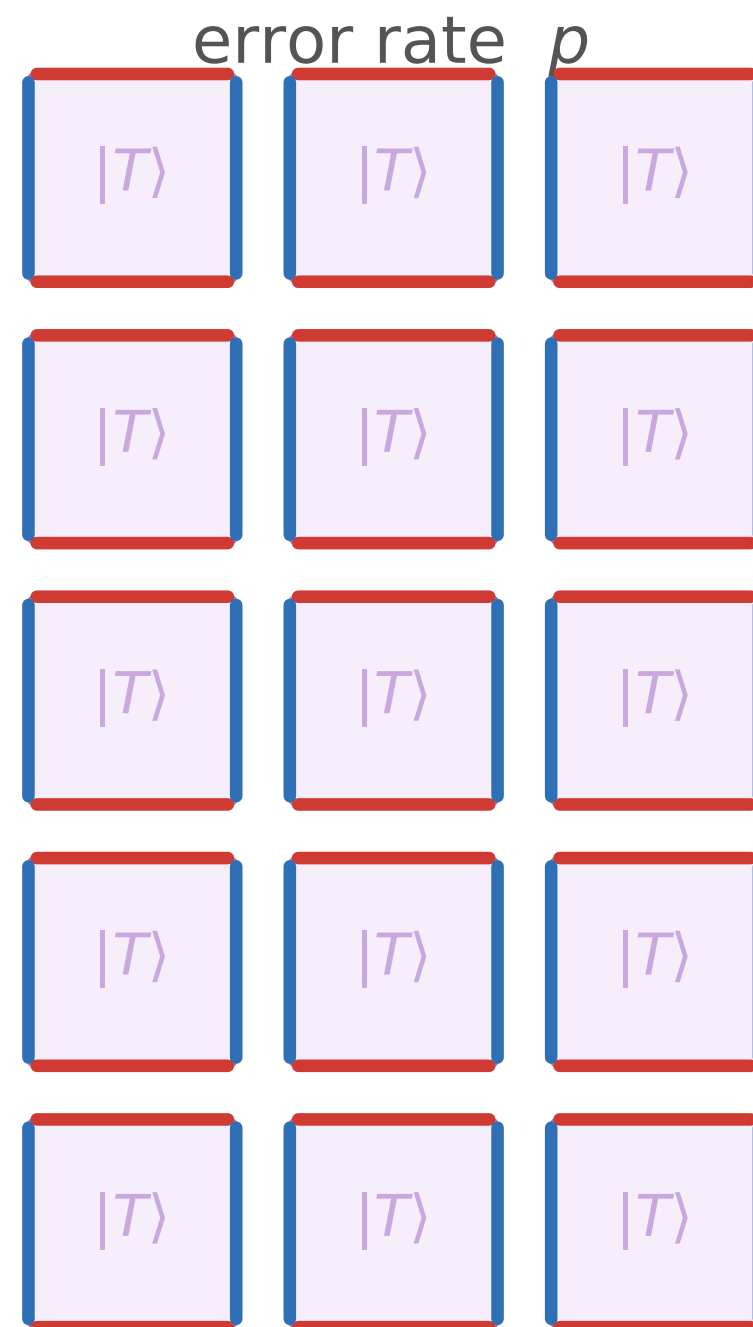
③ split & measure $\Rightarrow$ inject T



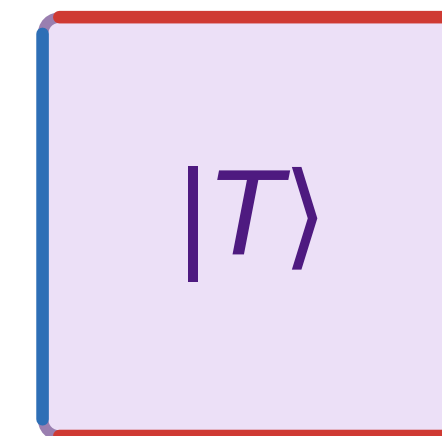
Solution 3 - T-gate distillation

Magic-state distillation factory

15 noisy magic states



1 purified
magic state



error rate $\approx 35p^3$

Limitations

- This procedure only works for **surface codes**
- The **rate** is extremely **low**
- Can we do this with **qLDPCs**?
 - Quantum LDPCs **have more** than 1 **logical** qubit, which could save a lot of **overhead**
 - Ideally we would want to do it all in **one cell**

Quantum operations using quantum LDPC codes

- **Question:** Can we do generalized lattice surgery and T-gate distillation using qLDPC codes?
- Yes for IBM codes
- Yes in general but the construction is bit complicated

2026

Tour de gross: A modular quantum computer based on bivariate bicycle codes

Theodore J. Yoder¹, Eddie Schoute¹, Patrick Rall¹, Emily Pritchett¹, Jay M. Gambetta¹, Andrew W. Cross¹, Malcolm Carroll¹, and Michael E. Beverland^{*1}

¹IBM Quantum

2025

Abstract

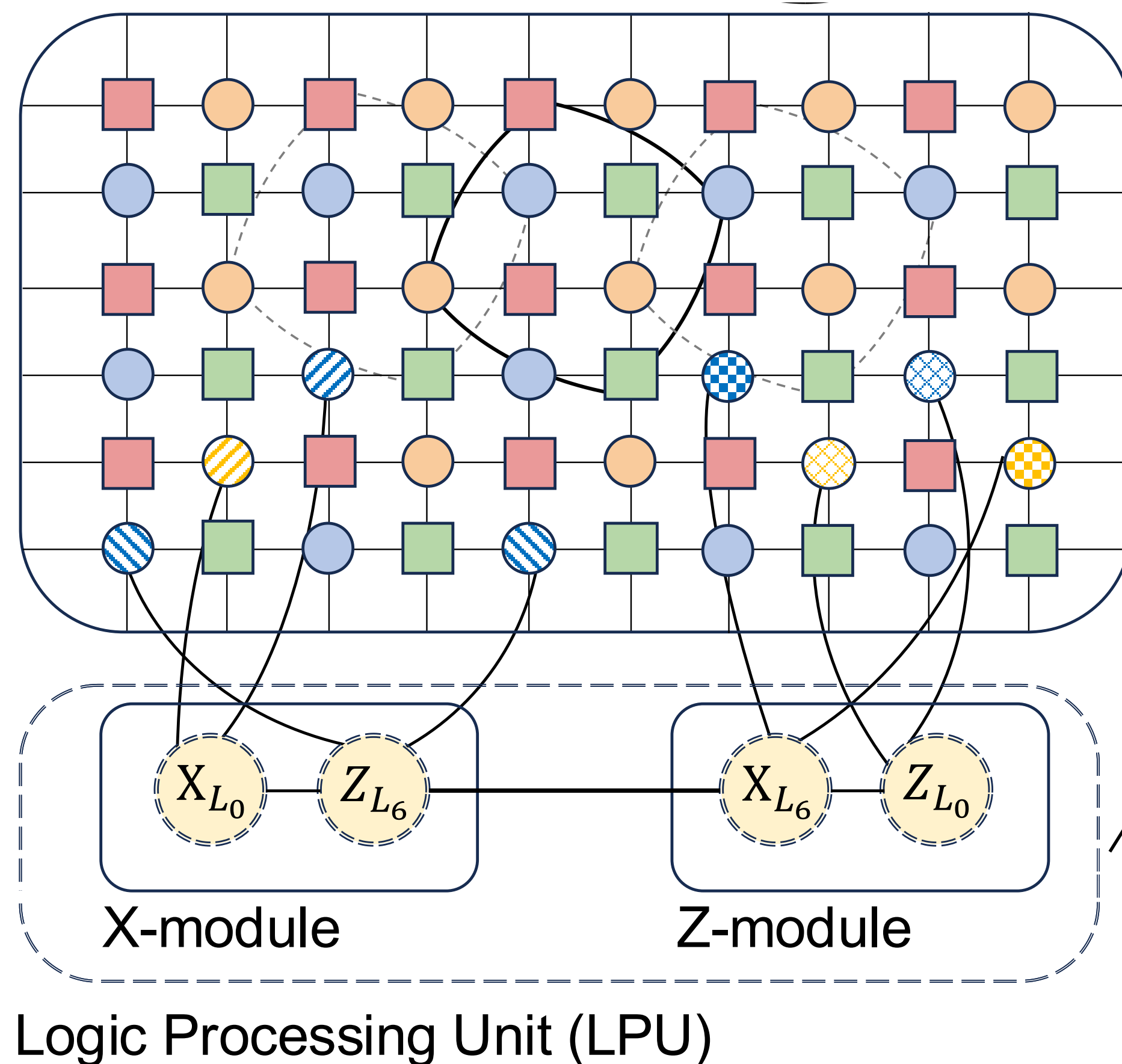
We present the *bicycle architecture*, a modular quantum computing framework based on high-rate, low-overhead quantum LDPC codes identified in prior work. For two specific bivariate bicycle codes with distances 12 and 18, we construct explicit fault-tolerant logical instruction sets and estimate the logical error rate of each instruction under circuit noise. We develop a compilation strategy adapted to the constraints of the bicycle architecture, enabling large-scale universal quantum circuit execution. Integrating these components, we perform end-to-end resource estimates demonstrating that an order of magnitude larger logical circuits can be implemented with a given number of physical qubits on a bicycle architecture than on surface code architectures. We anticipate further improvements through advances in code constructions, circuit designs, and compilation techniques.

Extractors: QLDPC Architectures for Efficient Pauli-Based Computation

Zhiyang He (Sunny)^{1,†}, Alexander Cowtan², Dominic J. Williamson³, and Theodore J. Yoder⁴
(Dated: October 28, 2025)

In pursuit of large-scale fault-tolerant quantum computation, quantum low-density parity-check (LDPC) codes have been established as promising candidates for low-overhead memory when compared to conventional approaches based on surface codes. Performing fault-tolerant logical computation on QLDPC memory, however, has been a long standing challenge in theory and in practice. In this work, we propose a new primitive, which we call an *extractor system*, that can augment any QLDPC memory into a computational block well-suited for Pauli-based computation. In particular, any logical Pauli operator supported on the memory can be fault-tolerantly measured in one logical cycle, consisting of $O(d)$ physical syndrome measurement cycles, without rearranging qubit connectivity. We further propose a fixed-connectivity, LDPC architecture built by connecting many extractor-augmented computational (EAC) blocks with bridge systems. When combined with any user-defined source of high fidelity $|T\rangle$ states, our architecture can implement universal quantum circuits via parallel logical measurements, such that all single-block Clifford gates are compiled away. The size of an extractor on an n qubit code is $\tilde{O}(n)$, where the precise overhead has immense room for practical optimizations.

Quantum operations using quantum LDPC codes



- It is a new code that performs logical operations on the LDPC code
- It attaches to the "logical qubits"
- It needs to be designed so that all its stabilizers commute with the stabilizers of the memory code.

Quantum operations using quantum LDPC codes

Distilling Magic States in the Bicycle Architecture

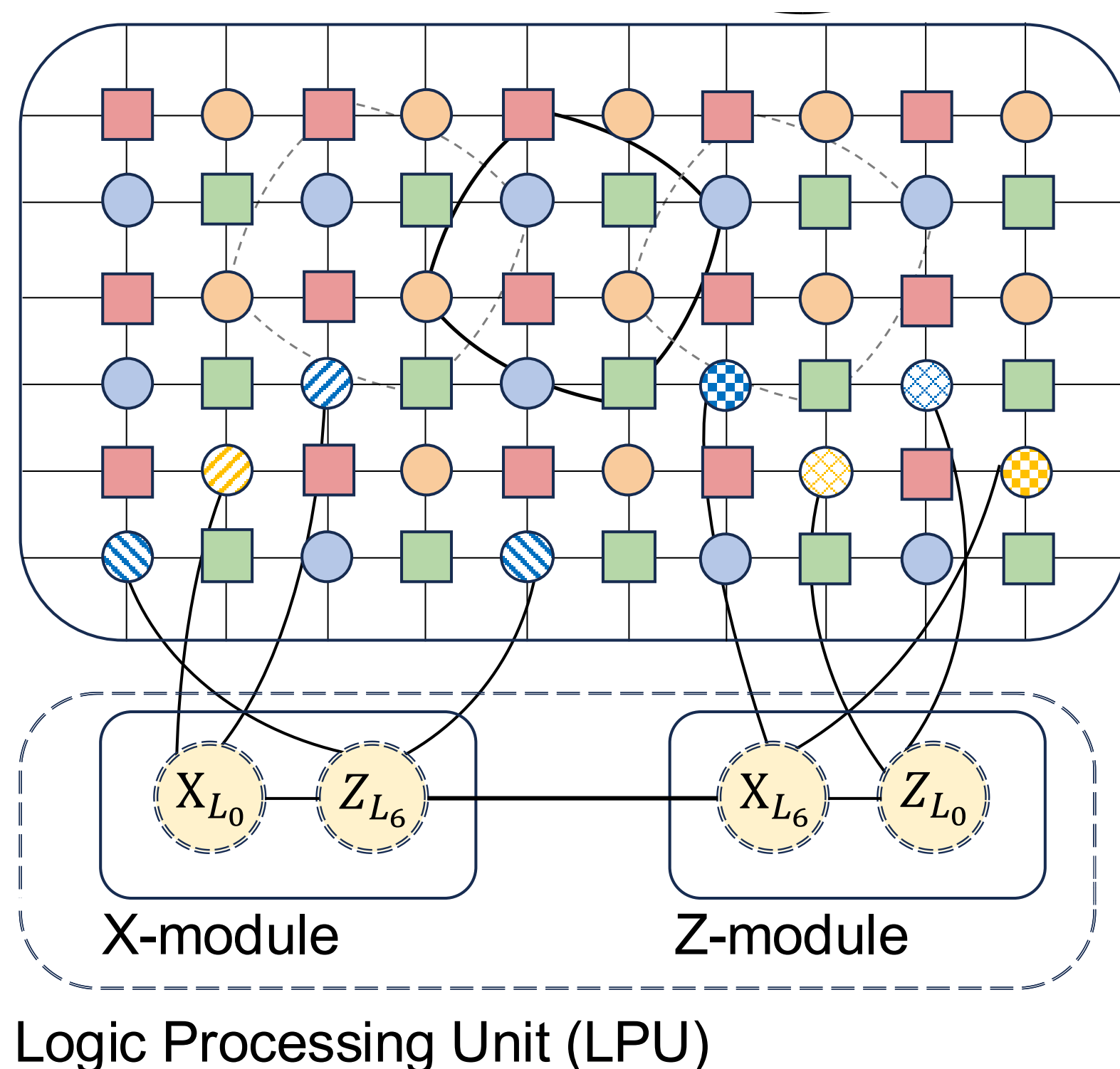
Shifan Xu*, Kun Liu†, Patrick Rall‡, Zhiyang He§, Yongshan Ding*†

*Department of Applied Physics, Yale University, New Haven, CT 06511, USA

†Department of Computer Science, Yale University, New Haven, CT 06511, USA

‡IBM Quantum, IBM Research, Cambridge, MA 02142, USA

§Department of Mathematics, Massachusetts Institute of Technology, Cambridge, MA 02139, USA



15-to-1 magic state distillation

- Two-gross code
 - 784 qubits
- Surface codes
 - ~2070 qubits

State of the art - challenges

- **General QLDPC extractors** are still very **new**
 - We are working on a more accessible version
- **Decoding** errors during a logical operation with an extractor is **complex**, and **not** a lot is **known** about **optimal decoders** for it
 - 300K error **mechanisms** over 18 rounds of syndrome **measurements**
- Implementation of extractors has only been tried for a small number of **qLDPC codes**
 - IBM codes, Product codes