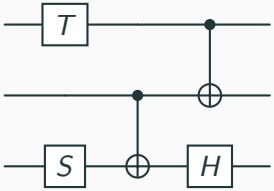
Lattice surgery and the ZX-calculus

Robert I. Booth

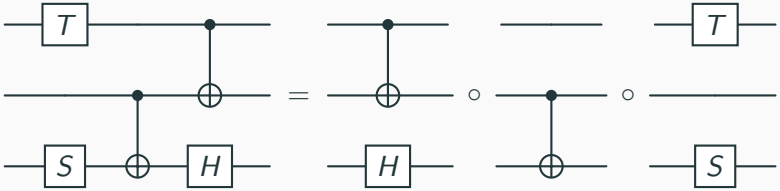
QEC Journal Club
23 October 2024

Diagrammatics

Quantum circuits

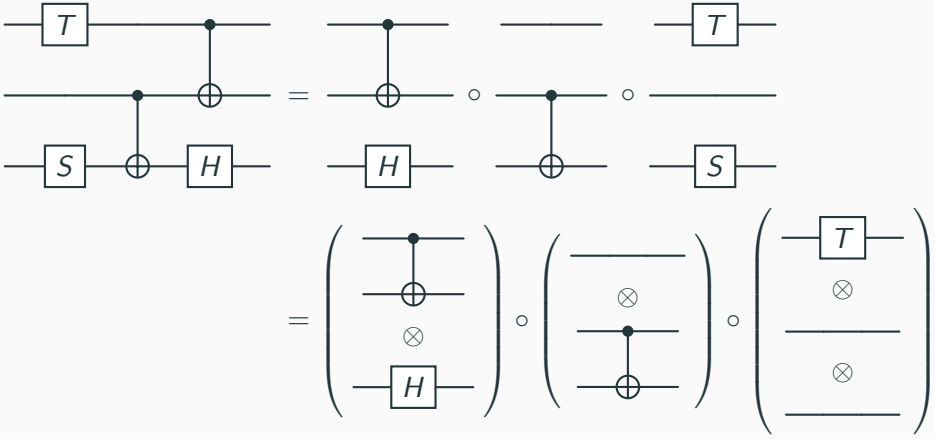


Quantum circuits



"The (de)compositional approach to quantum computing"

Quantum circuits



"The (de)compositional approach to quantum computing"

Quantum circuits: generators and interpretation

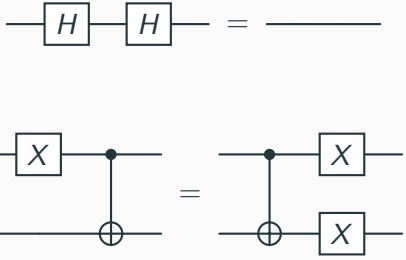
$$\left[\!\left[-\boxed{X}- \right]\!\right] = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \qquad \left[\!\left[-\boxed{S}- \right]\!\right] = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$$

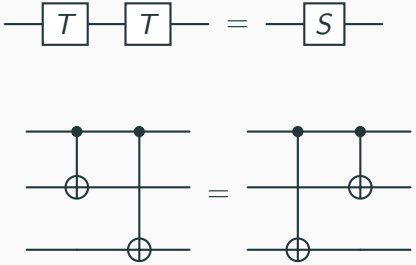
$$\left[\!\left[-\boxed{T}- \right]\!\right] = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\frac{2\pi}{8}} \end{pmatrix} \qquad \left[\!\left[-\boxed{H}- \right]\!\right] = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

$$\left[\!\left[\quad \right]\!\right] = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Clifford+T circuits are approximately universal for unitary matrices.

Quantum circuits: (some) equations

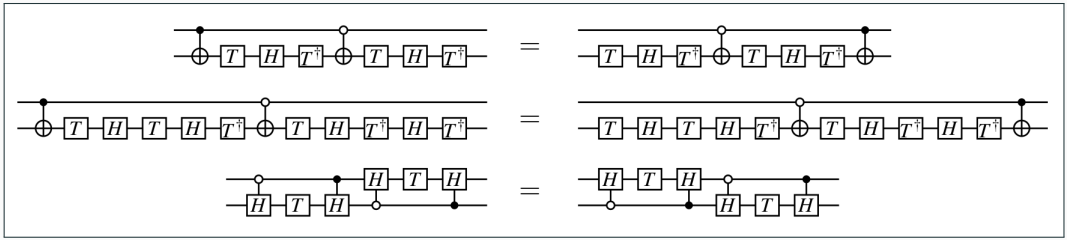




Question

How strong are these identities of quantum circuits?

Quantum circuits: nasty equations



Xiaoning Bian and Peter Selinger. **"Generators and Relations for 2-Qubit Clifford+T Operators".** In: *Proceedings 19th International Conference on Quantum Physics and Logic*. Quantum Physics and Logic 2022. Vol. 394. version: 3. Oxford, United Kingdom: EPCTS, July 2022, pp. 13–28.

Quantum circuits: universal generators

$$\left[\!\left[-\boxed{X}- \right]\!\right] = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \qquad \left[\!\left[-\boxed{S}- \right]\!\right] = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$$

$$\left[\!\left[-\boxed{T}- \right]\!\right] = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\frac{2\pi}{8}} \end{pmatrix} \qquad \left[\!\left[-\boxed{H}- \right]\!\right] = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

$$\left[\!\left[\right]\!\right] = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

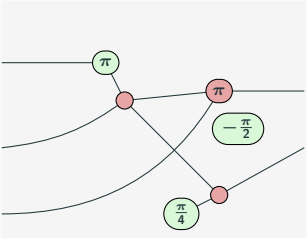
Quantum circuits: universal generators

$$\left[\!\left[\text{—}\boxed{P(\alpha)}\text{—} \right]\!\right] = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\alpha} \end{pmatrix} \qquad \left[\!\left[\text{—}\boxed{H}\text{—} \right]\!\right] = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

$$\left[\!\left[\quad \right]\!\right] = \begin{pmatrix} 1&0&0&0 \\ 0&1&0&0 \\ 0&0&0&1 \\ 0&0&1&0 \end{pmatrix}$$

CNOT+Phase+Hadamard circuits are universal for unitary matrices.

The ZX-calculus



The ZX-calculus: generators and interpretation

$$\left[\!\left[\;\right]\!\right] = |0\rangle\langle 00| + |1\rangle\langle 11|$$

$$\left[\!\left[\;\right]\!\right] = |00\rangle\langle 0| + |11\rangle\langle 1|$$

$$\left[\!\left[\;\right]\!\right] = |0\rangle + |1\rangle$$

$$\left[\!\left[\;\right]\!\right] = \langle 0| + \langle 1|$$

$$\left[\!\left[\;\right]\!\right] = |0\rangle\langle 0| + e^{i\alpha}\,|1\rangle\langle 1|$$

$$\left[\!\left[\;\right]\!\right] = |+\rangle\langle ++| + |-\rangle\langle --|$$

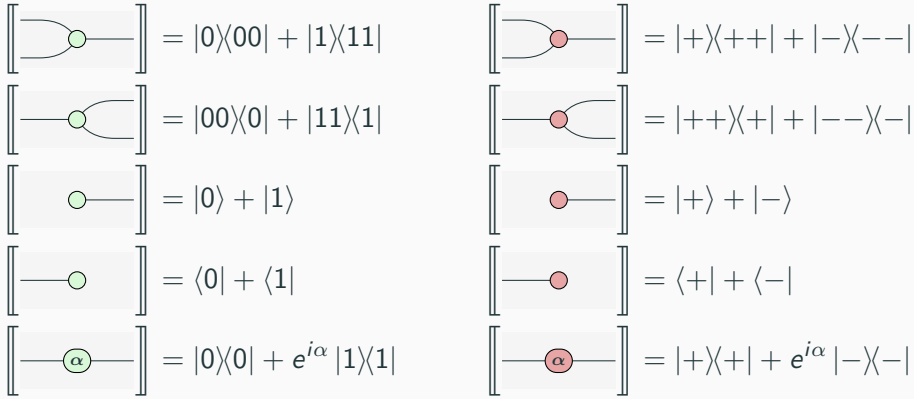
$$\left[\!\left[\;\right]\!\right] = |++\rangle\langle +| + |--\rangle\langle -|$$

$$\left[\!\left[\;\right]\!\right] = |+\rangle + |-\rangle$$

$$\left[\!\left[\;\right]\!\right] = \langle +| + \langle -|$$

$$\left[\!\left[\;\right]\!\right] = |+\rangle\langle +| + e^{i\alpha}\,|-\rangle\langle -|$$

The ZX-calculus: generators and interpretation



The ZX-calculus is universal for complex matrices.

The ZX-calculus: generators and interpretation

$$\left[\!\!\left[\; m \;\; \alpha \;\; n \; \right]\!\!\right] = \left|0 \cdots^{m} 0\right\rangle\!\left\langle 0 \cdots^{n} 0\right| + e^{i\alpha} \left|1 \cdots^{m} 1\right\rangle\!\left\langle 1 \cdots^{n} 1\right|$$

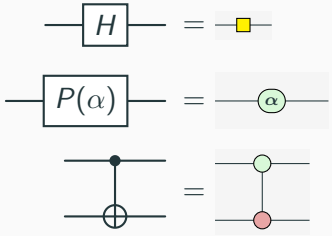
$$\left[\!\!\left[\; m \;\; \alpha \;\; n \; \right]\!\!\right] = \left|+ \cdots^{m} +\right\rangle\!\left\langle + \cdots^{n} +\right| + e^{i\alpha} \left|- \cdots^{m} -\right\rangle\!\left\langle - \cdots^{n} -\right|$$

$$[\![\; \text{—□—} \;]\!] = |+\rangle\langle 0| + |-\rangle\langle 1|$$

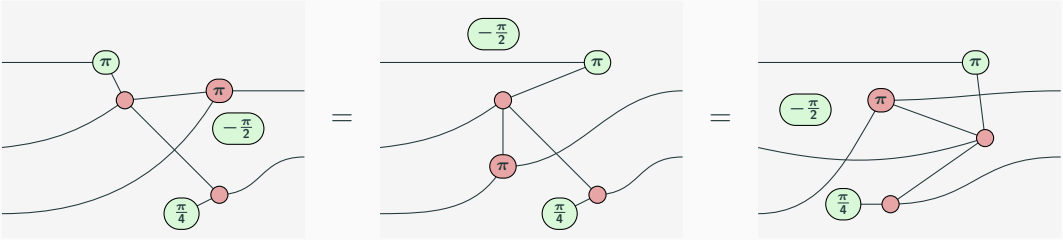
The ZX-calculus is universal for complex matrices.

The ZX-calculus: quantum circuits

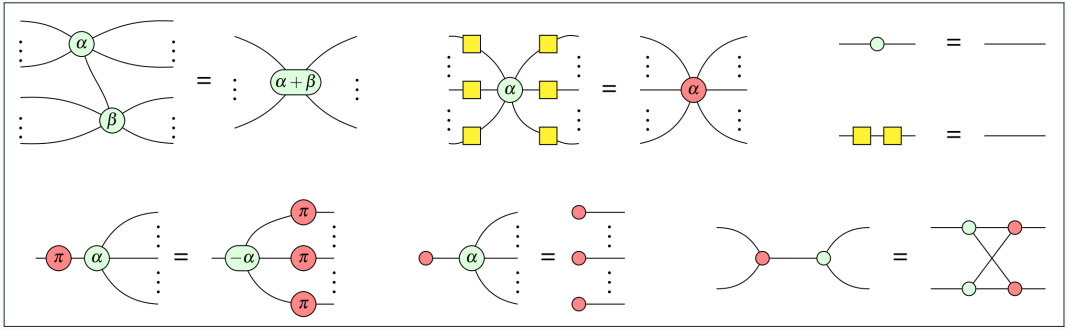
We can represent quantum circuits as ZX-diagrams:



The ZX-calculus: Only Connectivity Matters

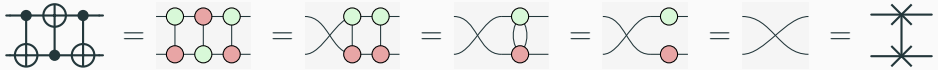


The ZX-calculus: equations



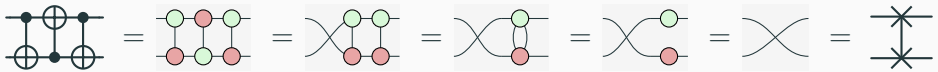
Thomas McElvanney and Miriam Backens. **"Complete flow-preserving rewrite rules for MBQC patterns with Pauli measurements".** In: *Proceedings 19th International Conference on Quantum Physics and Logic*. Quantum Physics and Logic (QPL 2022). June 1, 2022.

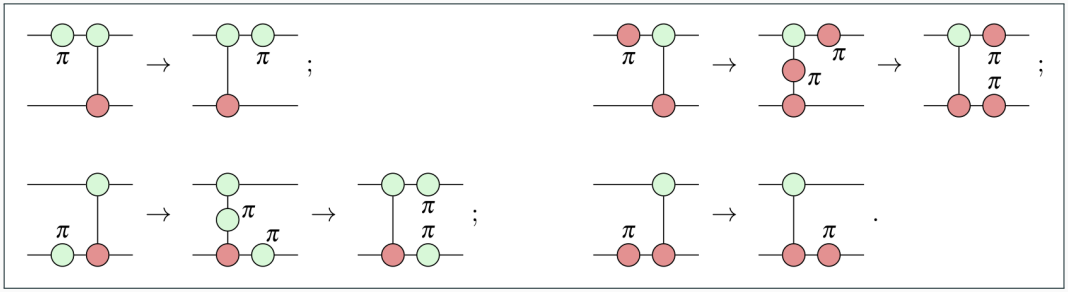
The ZX-calculus: examples



Niel de Beaudrap and Dominic Horsman. **"The ZX calculus is a language for surface code lattice surgery".** In: *Quantum* 4 (Jan. 9, 2020). Publisher: Verein zur Förderung des Open Access Publizierens in den Quantenwissenschaften, p. 218.

The ZX-calculus: examples





Niel de Beaudrap and Dominic Horsman. **"The ZX calculus is a language for surface code lattice surgery".** In: *Quantum* 4 (Jan. 9, 2020). Publisher: Verein zur Förderung des Open Access Publizierens in den Quantenwissenschaften, p. 218.

Lattice surgery, logically

The paper

The ZX calculus is a language for surface code lattice surgery

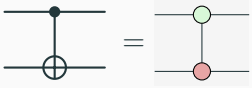
Niel de Beaudrap[1] and Dominic Horsman[2]

[1]Department of Computer Science, University of Oxford, Parks Road, Oxford, OX1 3QD
[2]Department of Physics, Durham University, South Road, Durham, DH1 1LE
Department of Computer Science, University of Oxford, Parks Road, Oxford, OX1 3QD

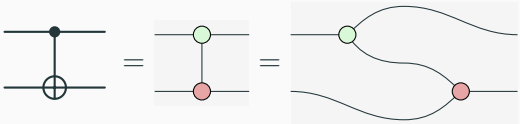
2 September, 2019

A leading choice of error correction for scalable quantum computing is the surface code with lattice surgery. The basic lattice surgery operations, the merging and splitting of logical qubits, act non-unitarily on the logical states and are not easily captured by standard circuit notation. This raises the question of how best to design, verify, and optimise protocols that use lattice surgery, in particular in architectures with complex resource management issues. In this paper we demonstrate that the operations of the ZX calculus — a form of quantum diagrammatic reasoning based on bialgebras — match exactly the operations of lattice surgery. Red and green "spider" nodes match rough and smooth merges and splits, and follow the axioms of a dagger special associative Frobenius algebra. Some lattice surgery operations require non-trivial correction operations, which are captured natively in the use of the ZX calculus in the form of ensembles of diagrams. We give a first taste of the power of the calculus as a language for lattice surgery by considering two operations (T gates and producing a CNOT) and show how ZX diagram re-write rules give lattice surgery procedures for these operations that are novel, efficient, and highly configurable.

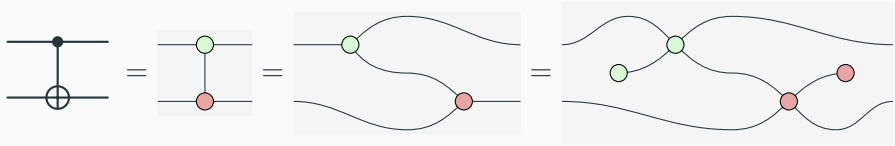
Splitting the atom



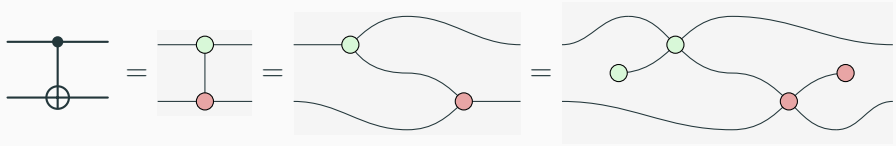
Splitting the atom

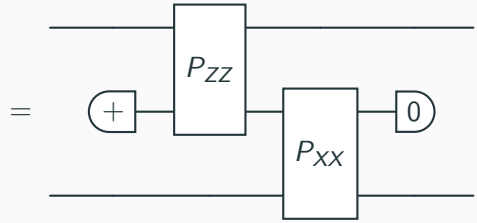


Splitting the atom

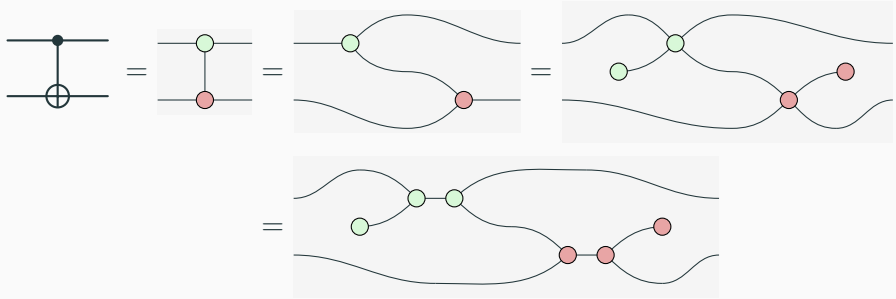


Splitting the atom





Splitting the atom



Merges and splits

- Z-**split**:

- Z-**merge**:

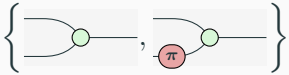
- X-**split**:

- X-**merge**:

Merges and splits

- Z-**split**:

- Z-**merge**:



- X-**split**:

- X-**merge**:

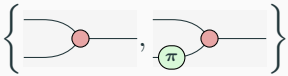
Merges and splits

- Z-**split**:

- Z-**merge**:

- X-**split**:

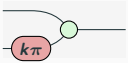
- X-**merge**:



Merges and splits

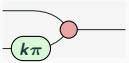
- Z-**split**:

- Z-**merge**:

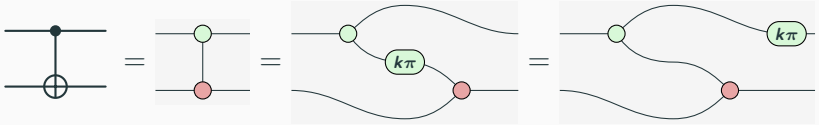


- X-**split**:

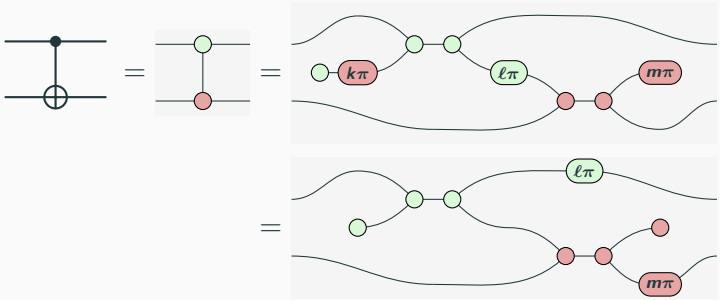
- X-**merge**:



Lattice surgery, in all its glory



Better surgery, in all its glory



Lattice surgery, physically

Another paper

Phase-free ZX diagrams are CSS codes
(...or how to graphically grok the surface code)

Aleks Kissinger

May 2, 2022

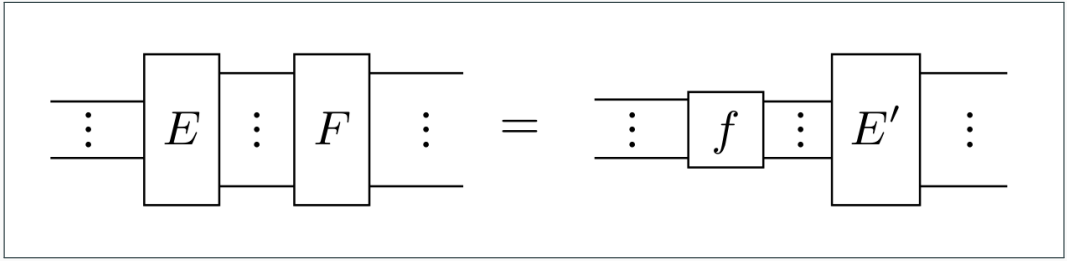
Abstract

In this paper, we demonstrate a direct correspondence between phase-free ZX diagrams, a graphical notation for representing and manipulating a certain class of linear maps on qubits, and Calderbank-Shor-Steane (CSS) codes, a large family of quantum error correcting codes constructed from classical codes, including for example the Steane code, surface codes, and colour codes. The stabilisers of a CSS code have an especially nice structure arising from a pair of orthogonal $\mathbb{F}_2$-linear subspaces, or in the case of maximal CSS codes, a single subspace and its orthocomplement. On the other hand, phase-free ZX diagrams can always be efficiently reduced to a normal form given by the basis elements of an $\mathbb{F}_2$-linear subspace. Here, we will show that these two ways of describing a quantum state by an $\mathbb{F}_2$-linear subspace S are in fact the same. Namely, the maximal CSS code generated by S fixes the quantum state whose ZX normal form is also given by S.

This insight gives us an immediate translation from stabilisers of a maximal CSS code into a ZX diagram describing its associated state. We show that we can extend this translation to stabilisers and logical operators of any (possibly non-maximal) CSS code by "bending wires". To demonstrate the utility of this translation, we give a simple picture of the surface code and a fully graphical derivation of the action of physical lattice surgery operations on the space of logical qubits, completing the ZX presentation of lattice surgery initiated by de Beudrap and Horsman.

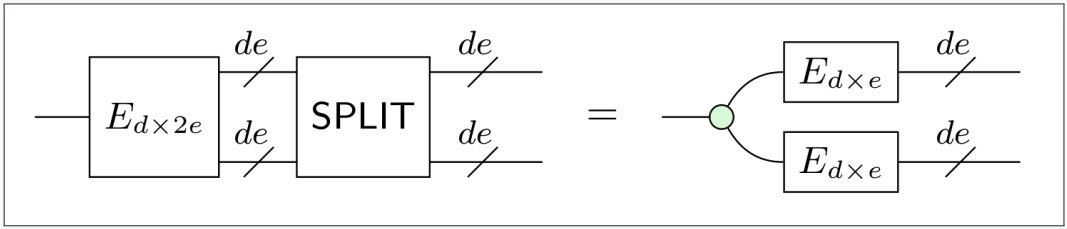
Logical and physical operations, diagrammatically

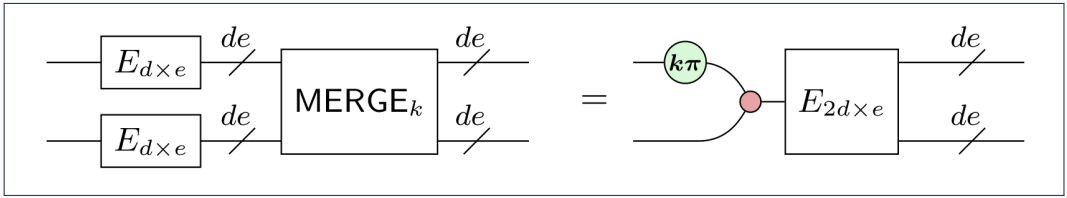
Implementing a logical operation f for an encoder E, means finding an F such that:



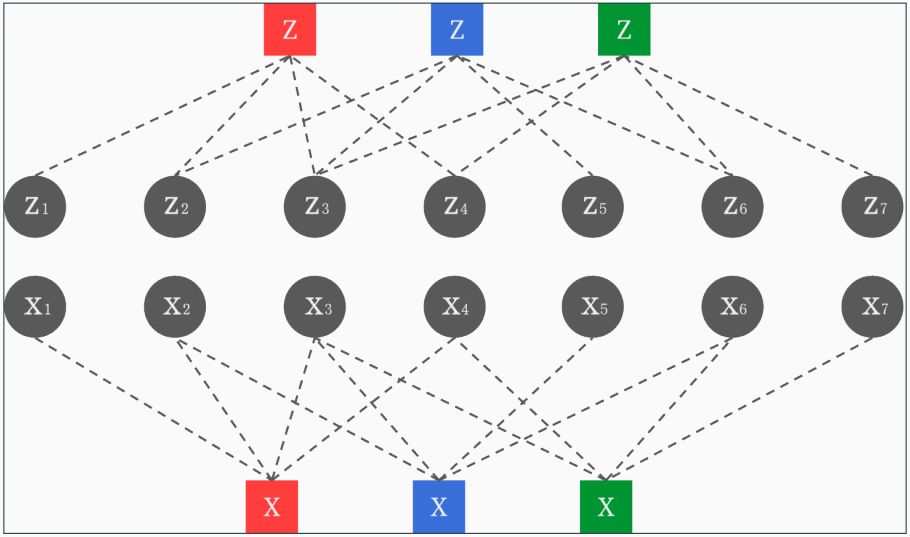
(often, $E = E'$, but this isn't compulsory)

Lattice surgery, logically and physically





CSS codes are Tanner graphs



Arthur Pesah, https://arthurpesah.me/blog/2023-03-28-stabilizer-formalism-3

CSS states are phase-free ZX-diagrams

A maximal CSS code is a stabiliser state called a CSS state. In ZX, these have two extremal representations:

X-representation

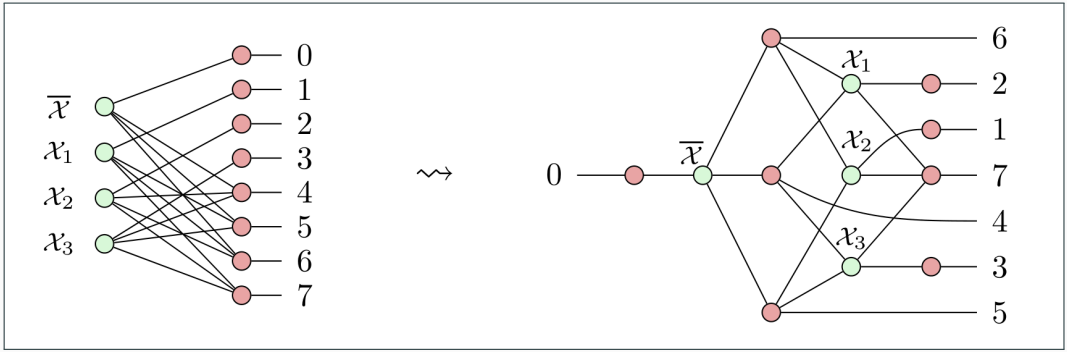
$\{X\otimes X\otimes X\}$ $\rightsquigarrow$ 

Z-representation

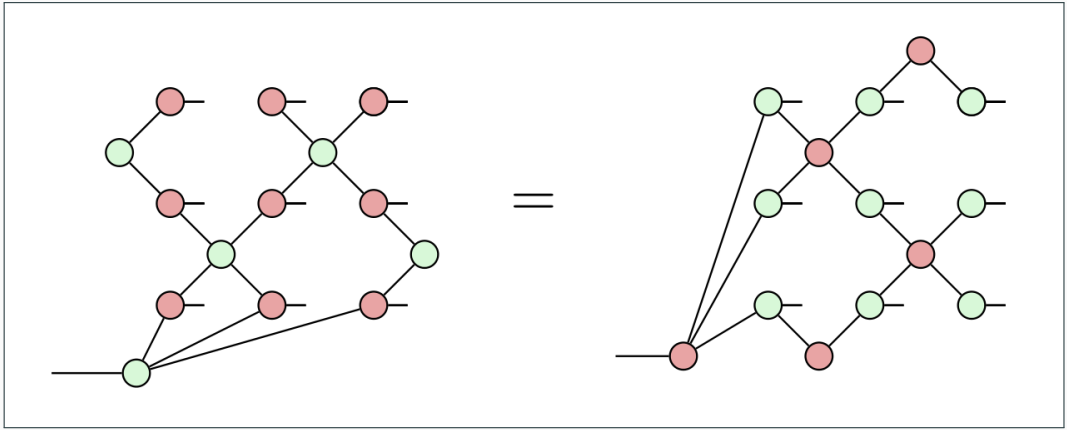
$\{Z\otimes Z\otimes I,\ I\otimes Z\otimes Z\}$ $\rightsquigarrow$ 

Tanner graphs and encoders

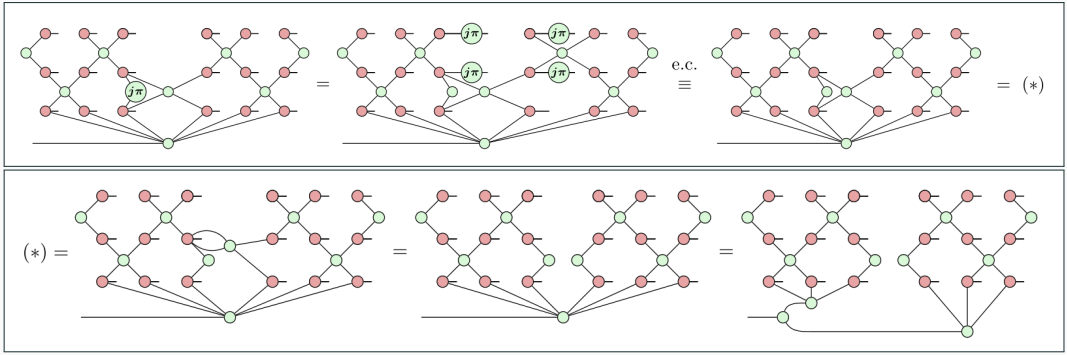
CSS states on $m + n$ qubits are in bijection with CSS encoders $m \to n$:



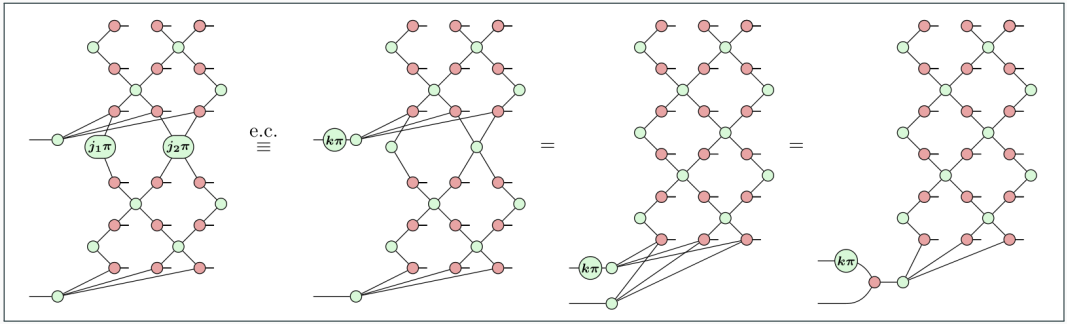
Encoders for the surface code



Splits in the surface code



Merges in the surface code



Questions?