Exploring edge cases of QEC using graphical languages

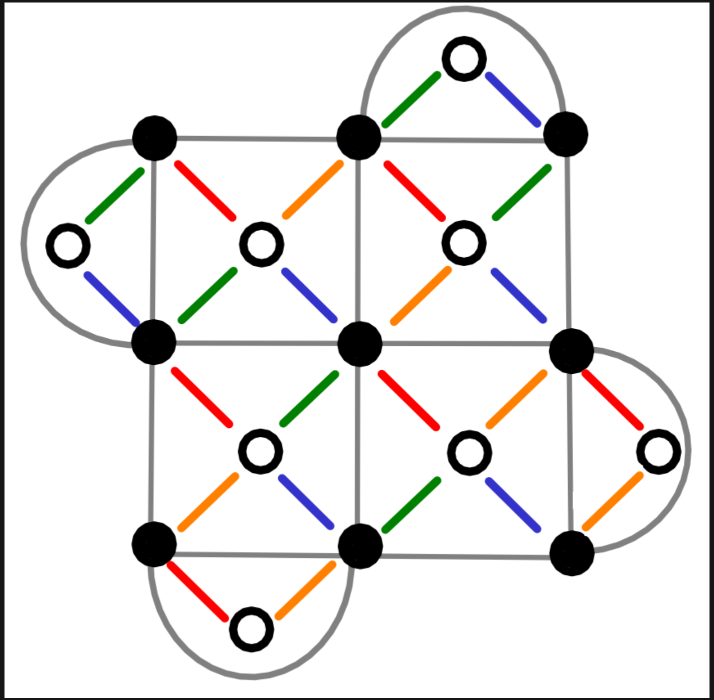
James Wootton

IBM Quantum, IBM Research - Zurich

IBM **Quantum**

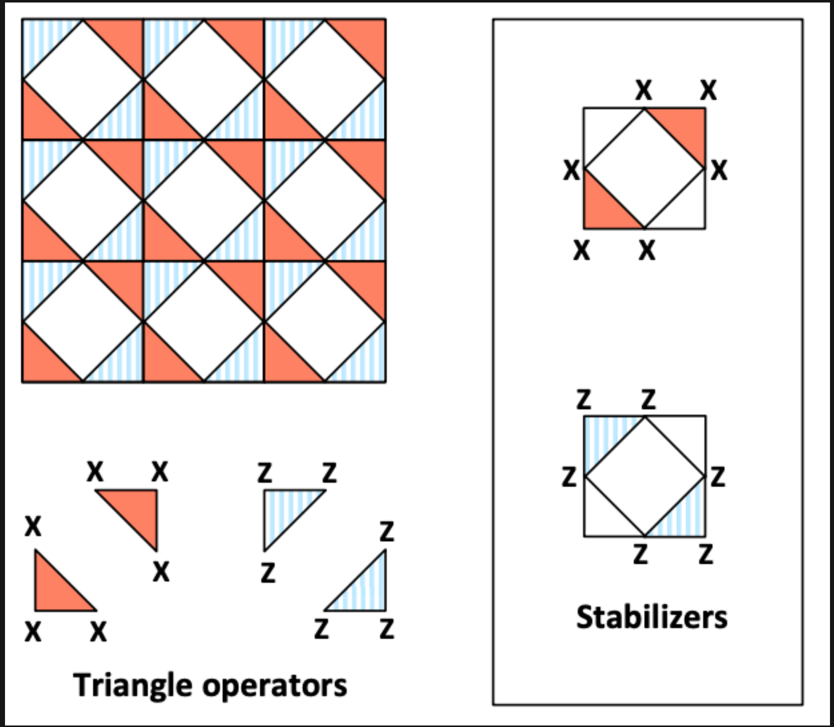
Stabilizer codes

- Define a set of mutually commuting stabilizers
- They must also commute with logical operators
- Measure them repeatedly
- Compare successive measurements of the same stabilizer to check for errors



Subsystem codes

- Define a set of gauge operators
- Find the mutually commuting stabilizers they create
- And the commuting logical operators
- Measure the gauge operators repeatedly
- Infer stabilizer measurements
- Compare successive measurements of the same stabilizer to check for errors



Bravyi et al., 2013

The rules of QEC

IBM **Quantum**

- The things you measure commute with the stabilizers
- The things you measure commute with the logical operators

- Now let’s break the rules!

The rebirth of QEC: Part 1

IBM **Quantum**

Floquet codes

- Introduced by Hastings and Haah in 2021
- Gauge operators commute with stabilizers
- But don’t commute with logical operators!
  - Logical operators are defined dynamically

Dynamically Generated Logical Qubits

Matthew B. Hastings[1,2] and Jeongwan Haah[2]

[1]Station Q, Microsoft Quantum, Santa Barbara, CA 93106-6105, USA

[2]Microsoft Quantum and Microsoft Research, Redmond, WA 98052, USA

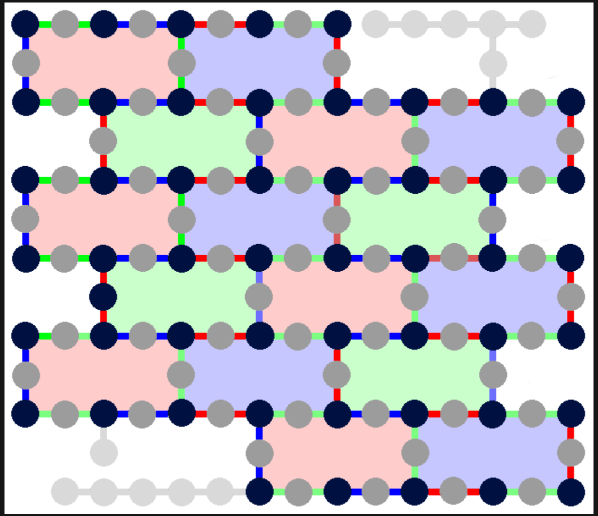
We present a quantum error correcting code with *dynamically generated logical qubits*. When viewed as a subsystem code, the code has no logical qubits. Nevertheless, our measurement patterns generate logical qubits, allowing the code to act as a fault-tolerant quantum memory. Our particular code gives a model very similar to the two-dimensional toric code, but each measurement is a *two*-qubit Pauli measurement.

The rebirth of QEC: Part 2

IBM **Quantum**

Matching codes with 2-body measurements

- Introduced by Wootton in 2021
- Gauge operators commute with *some* stabilizers
- But not all the stabilizers
  - Checks defined dynamically
- And don't commute with logical operators!
  - Also defined dynamically



Hexagonal matching codes with 2-body measurements

James R. Wootton[1]

[1] *IBM Quantum – IBM Research Zurich*

(Dated: December 7, 2021)

Matching codes are stabilizer codes based on Kitaev's honeycomb lattice model. The hexagonal form of these codes are particularly well-suited to the heavy-hexagon device layouts currently pursued in the hardware of IBM Quantum. Here we show how the stabilizers of the code can be measured solely through the 2-body measurements that are native to the architecture. The process is then run on 27 and 65 qubit devices, to compare results with simulations for a standard error model. It is found that the results correspond well to simulations where the noise strength is similar to that found in the benchmarking of the devices. The best devices show results consistent with a noise model with an error probability of around 1.5% – 2%.

The rebirth of QEC: Part 3

Floquet Color code

- By Davydova et al and Kesselring et al in 2022
- Gauge operators don't commute with the stabilizers
- Or the logical operators!

Floquet codes without parent subsystem codes

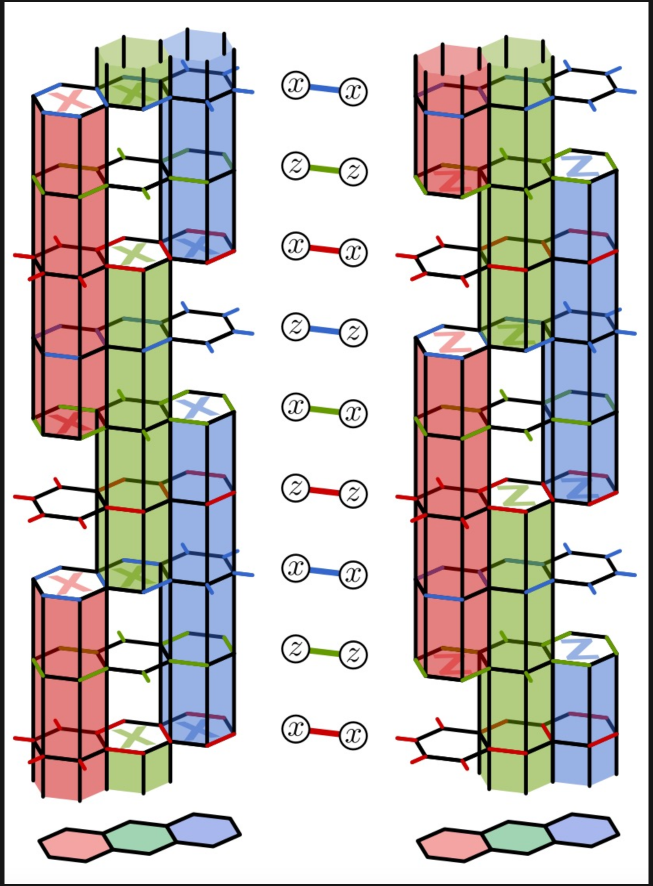
Margarita Davydova,[1,2] Nathanan Tantivasadakarn,[3,4] and Shankar Balasubramanian[5]

[1] *Department of Physics, Massachusetts Institute of Technology, Cambridge, MA 02139, USA*
[2] *Kavli Institute for Theoretical Physics, University of California, Santa Barbara, California 93106, USA*
[3] *Walter Burke Institute for Theoretical Physics and Department of Physics, California Institute of Technology, Pasadena, CA 91125, USA*
[4] *Department of Physics, Harvard University, Cambridge, MA 02138, USA*
[5] *Center for Theoretical Physics, Massachusetts Institute of Technology, Cambridge, MA 02139, USA*

Anyon condensation and the color code

Markus S. Kesselring[1], Julio C. Magdalena de la Fuente[1], Felix Thomsen[2], Jens Eisert[1,3], Stephen D. Bartlett[2], and Benjamin J. Brown[2]

[1]Dahlem Center for Complex Quantum Systems, Freie Universität Berlin, 14195 Berlin, Germany
[2]Centre for Engineered Quantum Systems, School of Physics, University of Sydney, Sydney, New South Wales 2006, Australia
[3]Helmholtz-Zentrum Berlin für Materialien und Energie, 14109 Berlin, Germany



The rebirth of QEC: Part 4

Dynamic surface codes

- Introduced by McEwen et al in 2023
- A normal stabilizer code
- Implemented in a completely abnormal way

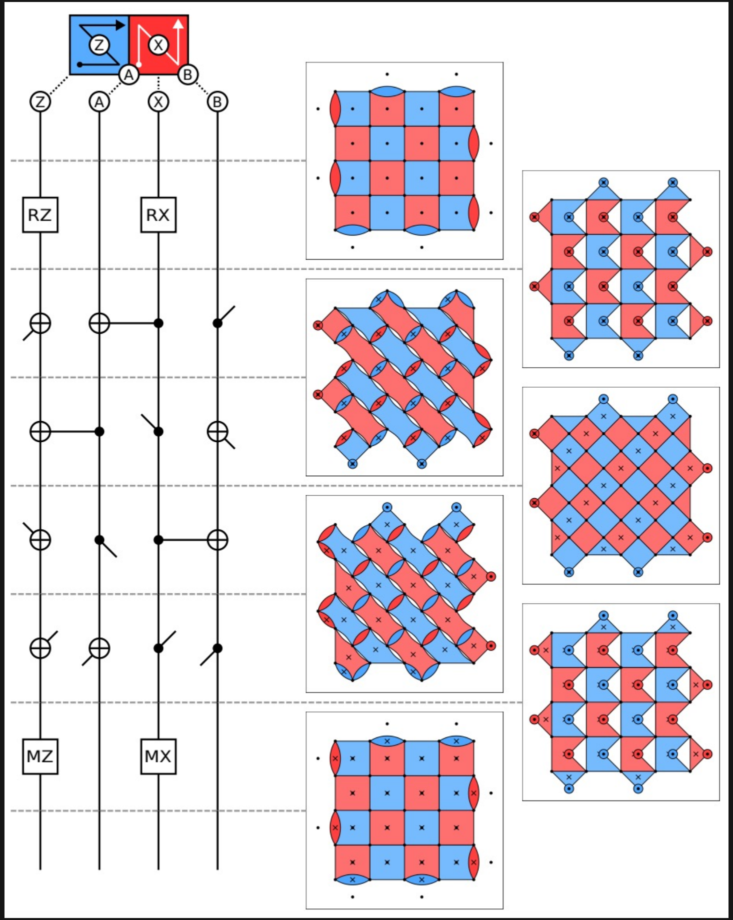
Relaxing Hardware Requirements for Surface Code Circuits using Time-dynamics

Matt McEwen[1], Dave Bacon[2], and Craig Gidney[1]

[1]Google Quantum AI, Santa Barbara, California 93117, USA
[2]Google Quantum AI, Seattle, Washington 98103, USA

February 7, 2023

The typical time-independent view of quantum error correction (QEC) codes hides significant freedom in the decomposition into circuits that are executable on hardware. Using the concept of detecting regions, we design time-dynamic QEC circuits directly instead of designing static QEC codes to decompose into circuits. In particular, we improve on the standard circuit constructions for the surface code, presenting new circuits that can embed on a hexagonal grid instead of a square grid, that can use ISWAP gates instead of CNOT or CZ gates, that can exchange qubit data and measure roles, and that move logical patches around the physical qubit grid while executing. All these constructions use no additional entangling gate layers and display essentially the same logical performance, having teraquop footprints within 25% of the standard surface code circuit. We expect these circuits to be of great interest to quantum hardware engineers, because they achieve essentially the same logical performance as standard surface code circuits while relaxing demands on hardware.

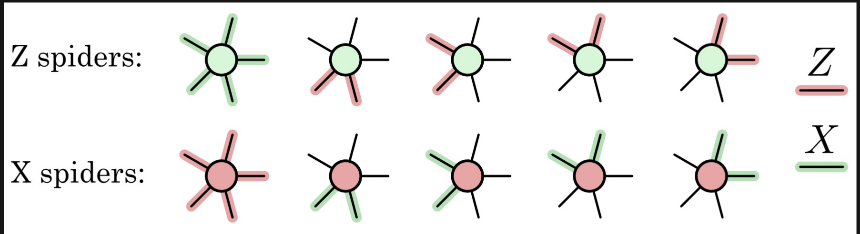


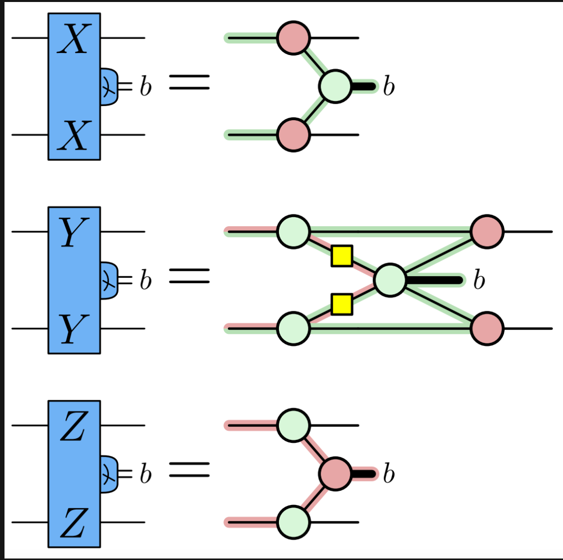
A new framework for QEC

IBM **Quantum**

Pauli webs

- Introduced by Bombin et al in 2023
- Analyzing circuits using ZX
- Checks and logicals defined by Pauli webs





Unifying flavors of fault tolerance with the ZX calculus

Hector Bombin, Daniel Litinski, Naomi Nickerson, Fernando Pastawski, and Sam Roberts
PsiQuantum, Palo Alto

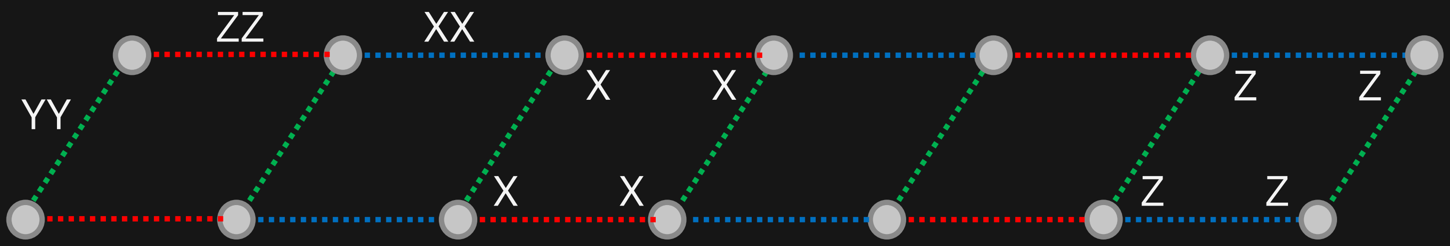
There are several models of quantum computation which exhibit shared fundamental fault-tolerance properties. This article makes commonalities explicit by presenting these different models in a unifying framework based on the ZX calculus. We focus on models of topological fault tolerance – specifically surface codes – including circuit-based, measurement-based and fusion-based quantum computation, as well as the recently introduced model of Floquet codes. We find that all of these models can be viewed as different flavors of the same underlying stabilizer fault-tolerance structure, and sustain this through a set of local equivalence transformations which allow mapping between flavors. We anticipate that this unifying perspective will pave the way to transferring progress among the different views of stabilizer fault-tolerance and help researchers familiar with one model easily understand others.

Pauli webs in action

We'll look at two simple examples

- The 1D version of Hastings-Haah's Floquet code
- The 1D version of my code

Though the 2D versions are distinct, the 1D versions look very similar

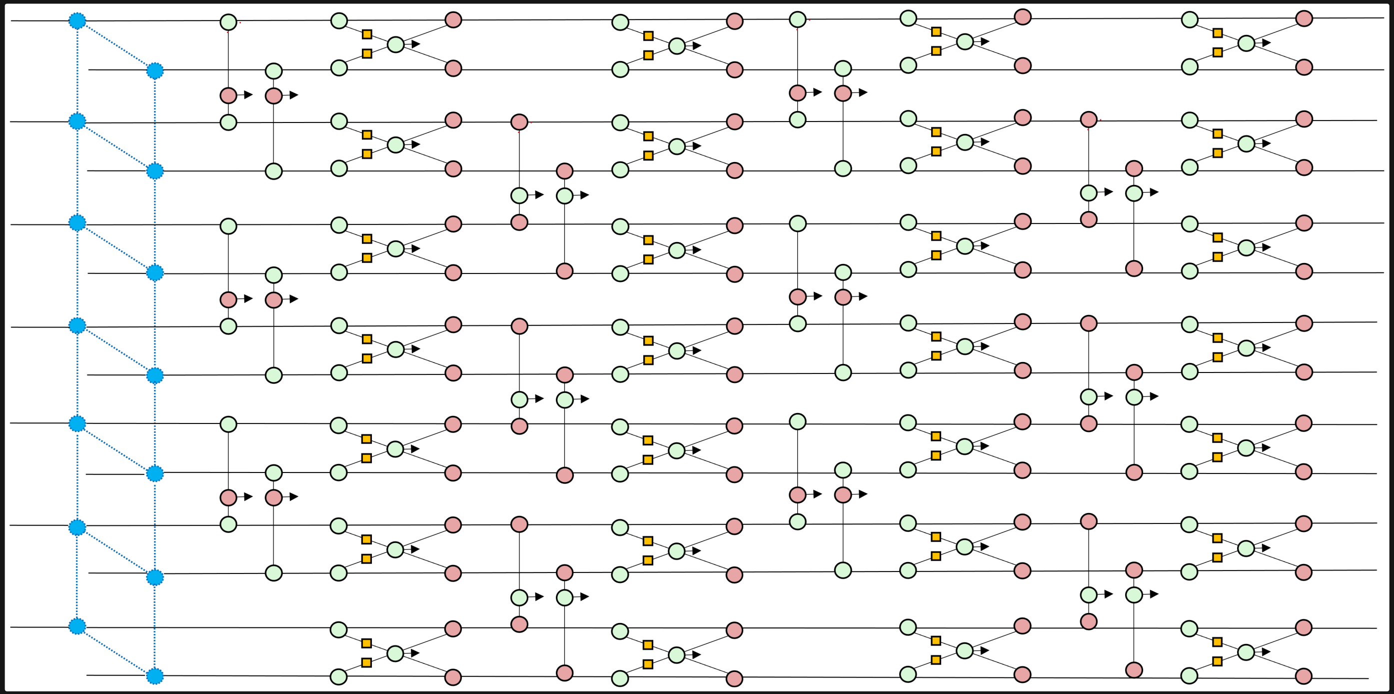


Measure ZZ → YY → XX → YY …

Pauli webs in action

IBM **Quantum**

Two rounds of measurements, expressed in ZX

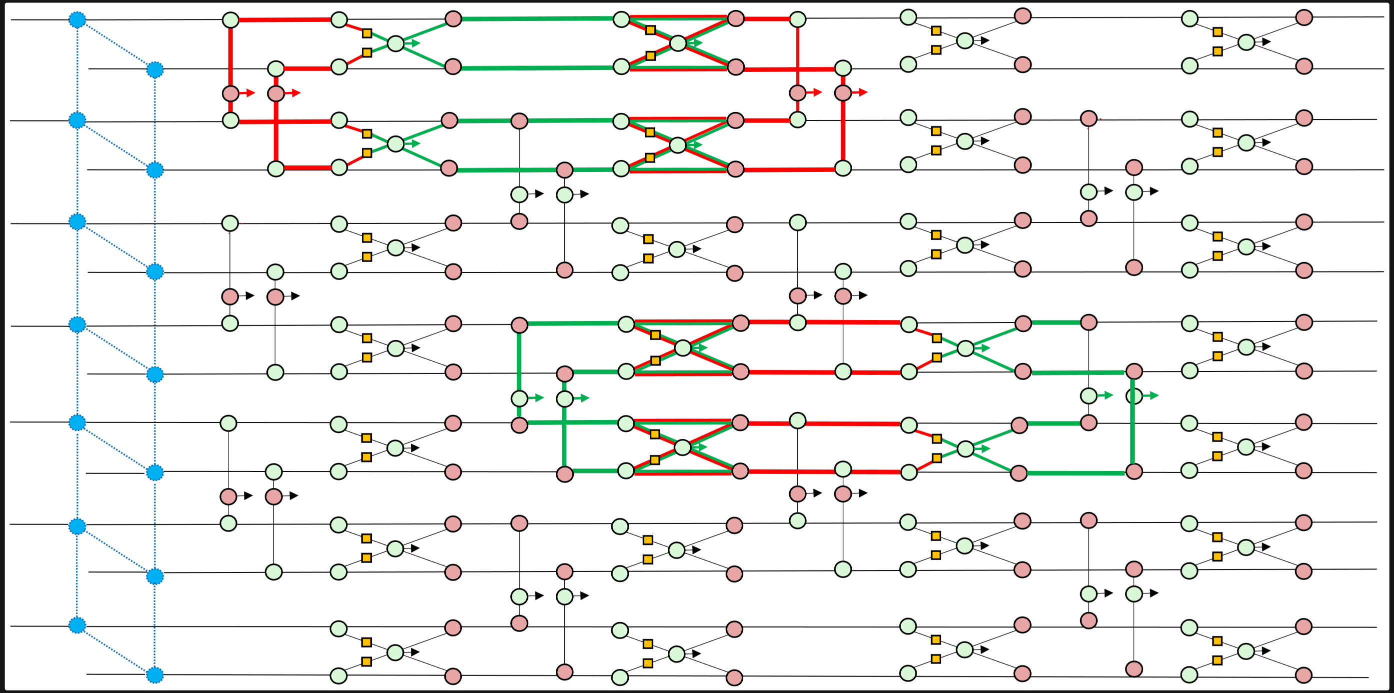


Pauli webs in action

IBM **Quantum**

Pauli webs identify the plaquette checks

- Compare plaquette values for successive rounds

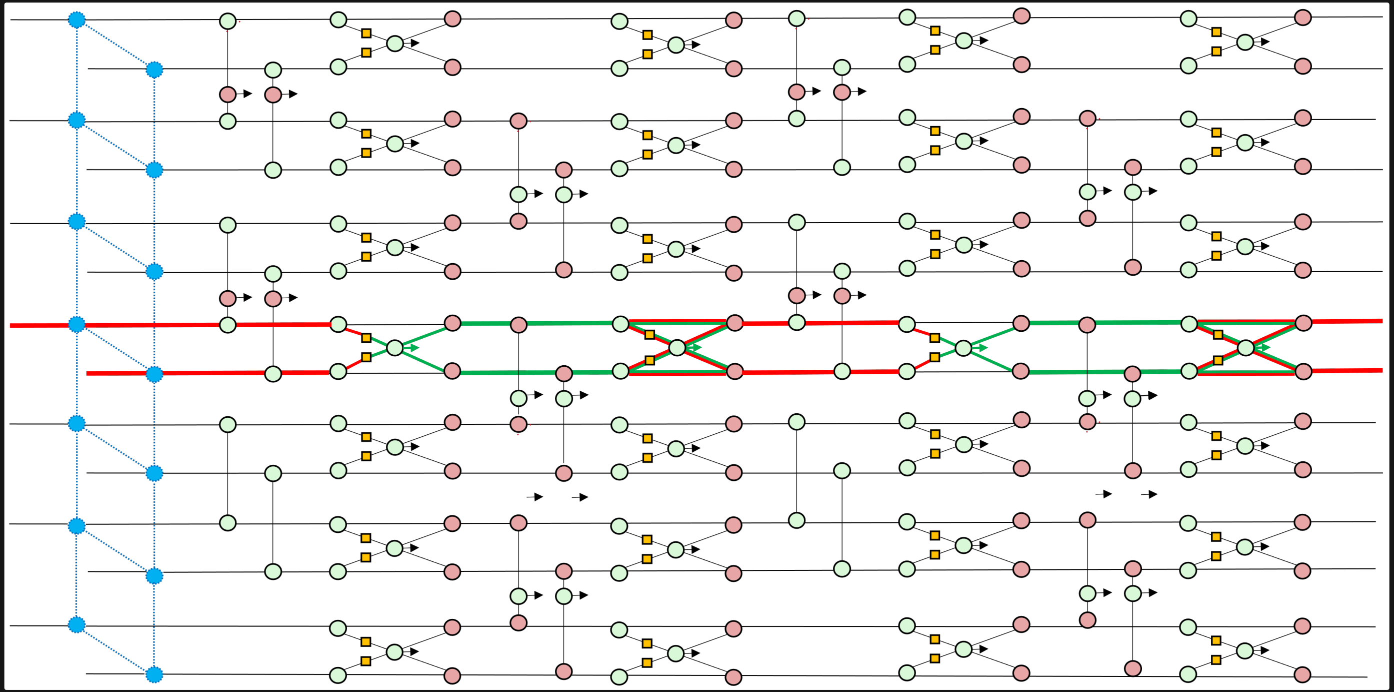


Pauli webs in action

IBM **Quantum**

Also find the Hastings-Haah logical operator

- Dynamically defined: flips between XX and ZZ

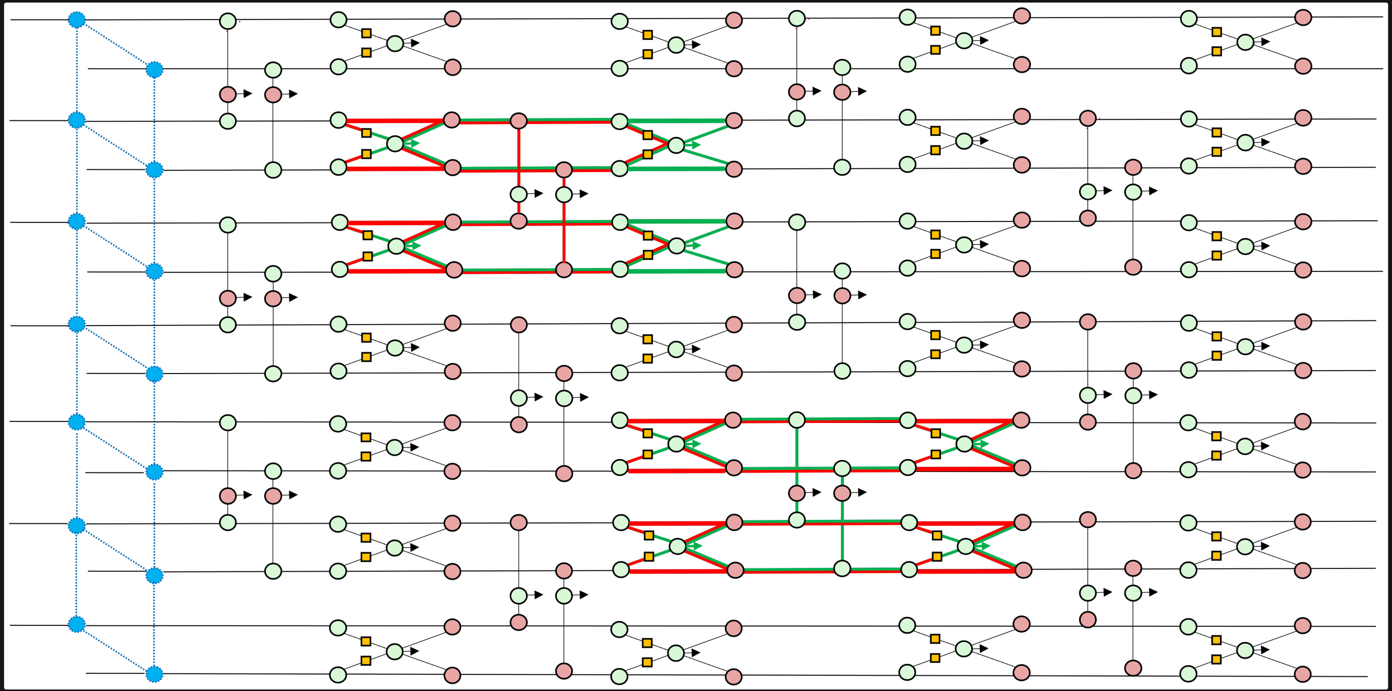


Pauli webs in action

IBM **Quantum**

Dynamically defined checks are also found on the rungs

- Required for Wootton code, but missed by Hastings-Haah

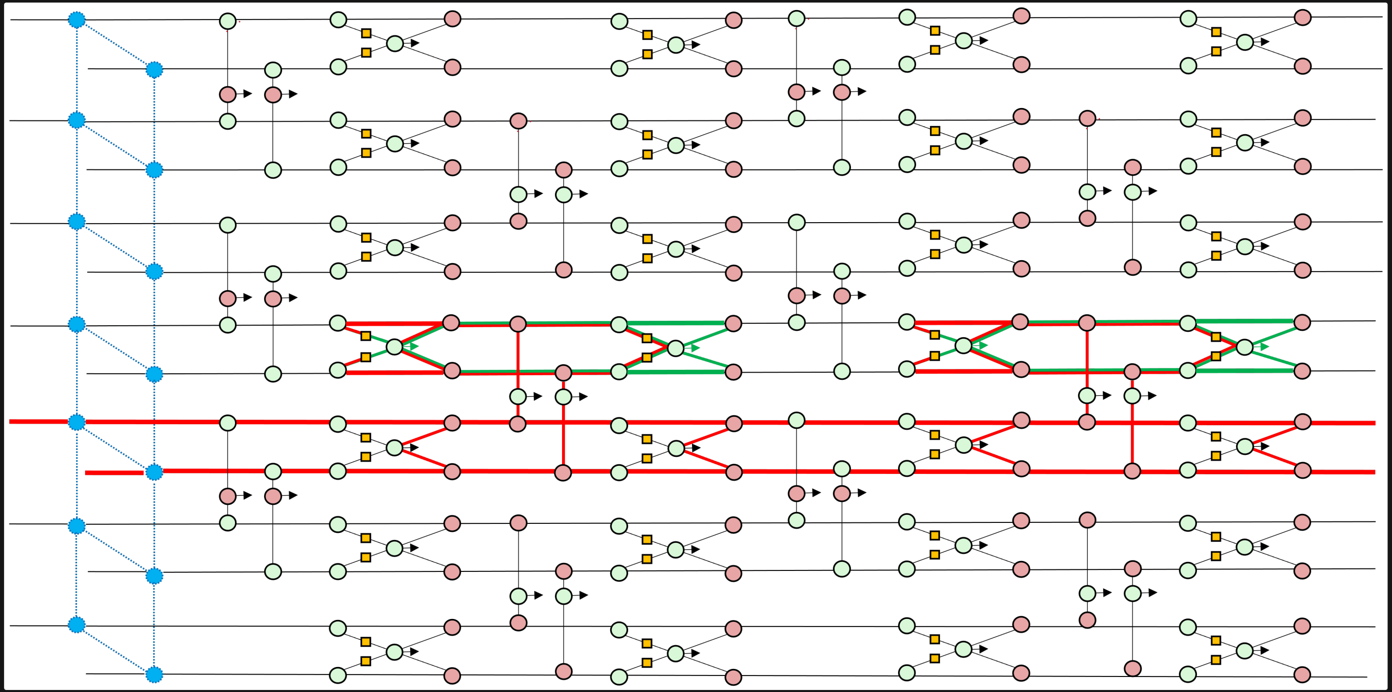


Pauli webs in action

IBM **Quantum**

Pauli webs also find the Wootton logical operator

- Can be viewed as static with corrections, or dynamically defined



Conclusions

IBM **Quantum**

- Fault-tolerance is in the midst of a revolution!
- Novel weird codes will lead the way

- ZX + Pauli webs seems a promising tool to map new lands
- But there are still things to figure out
  - How to automate it?
  - How to use it to inform decoding?
  - How to find new approaches with it?

Thanks for your attention!

Qiskit-QEC github.com/qiskit/qiskit-qec