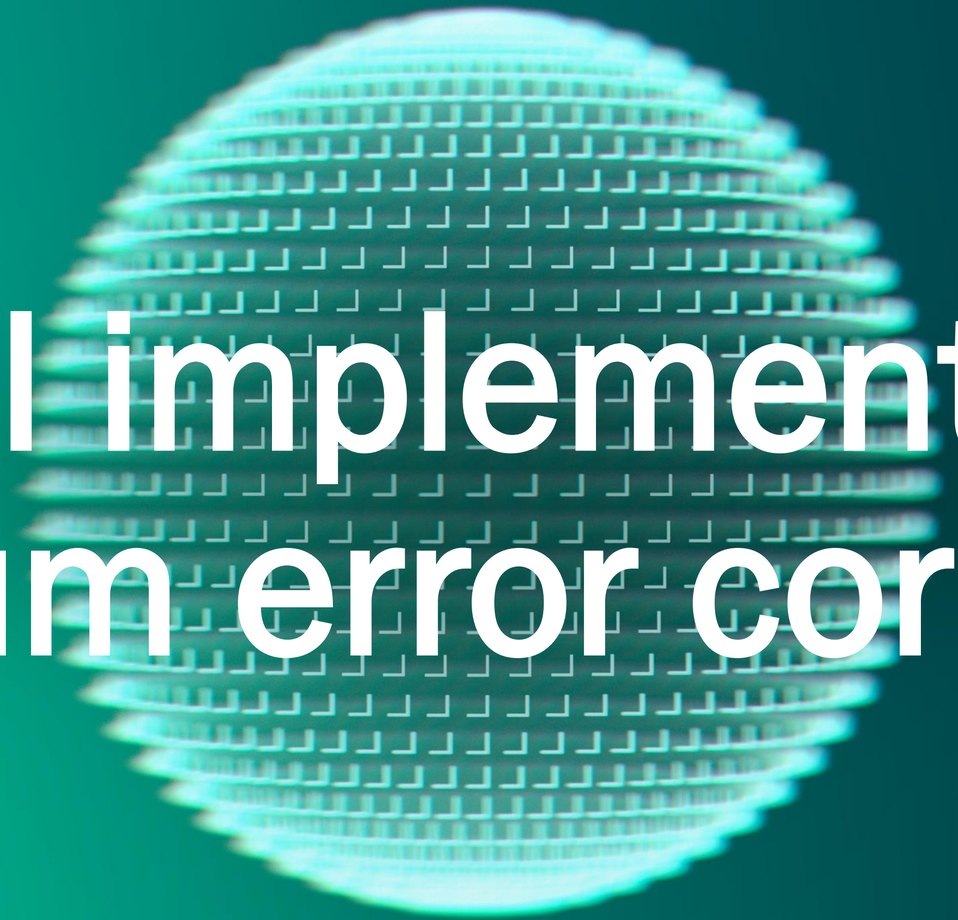
riverlane

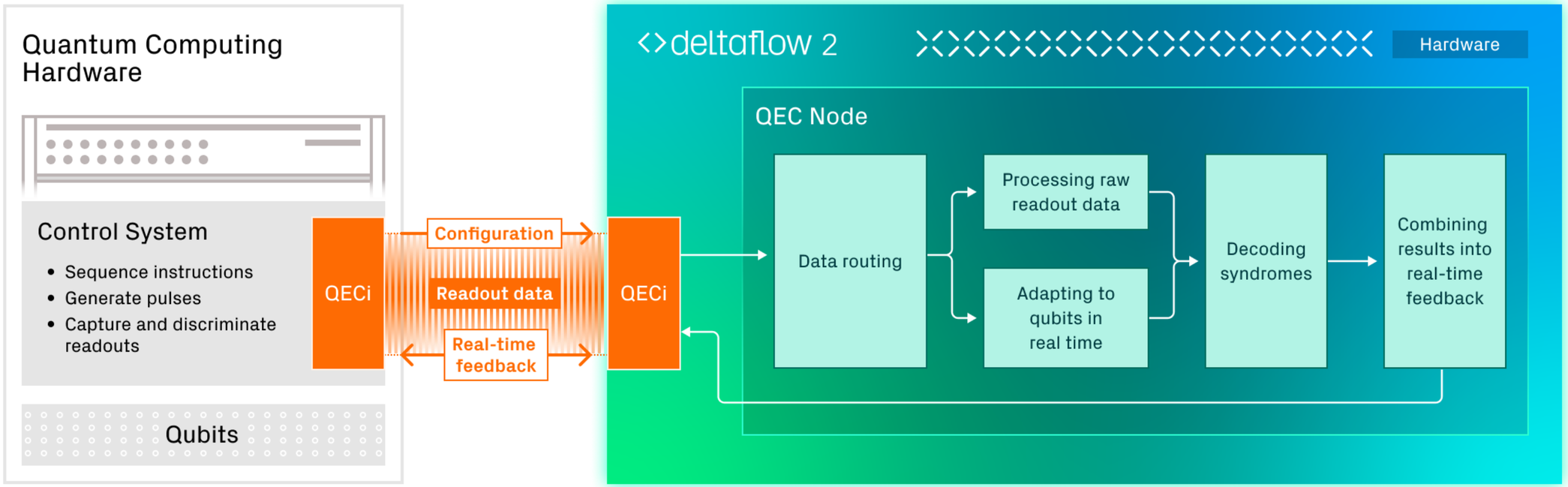


# Practical implementation of quantum error correction

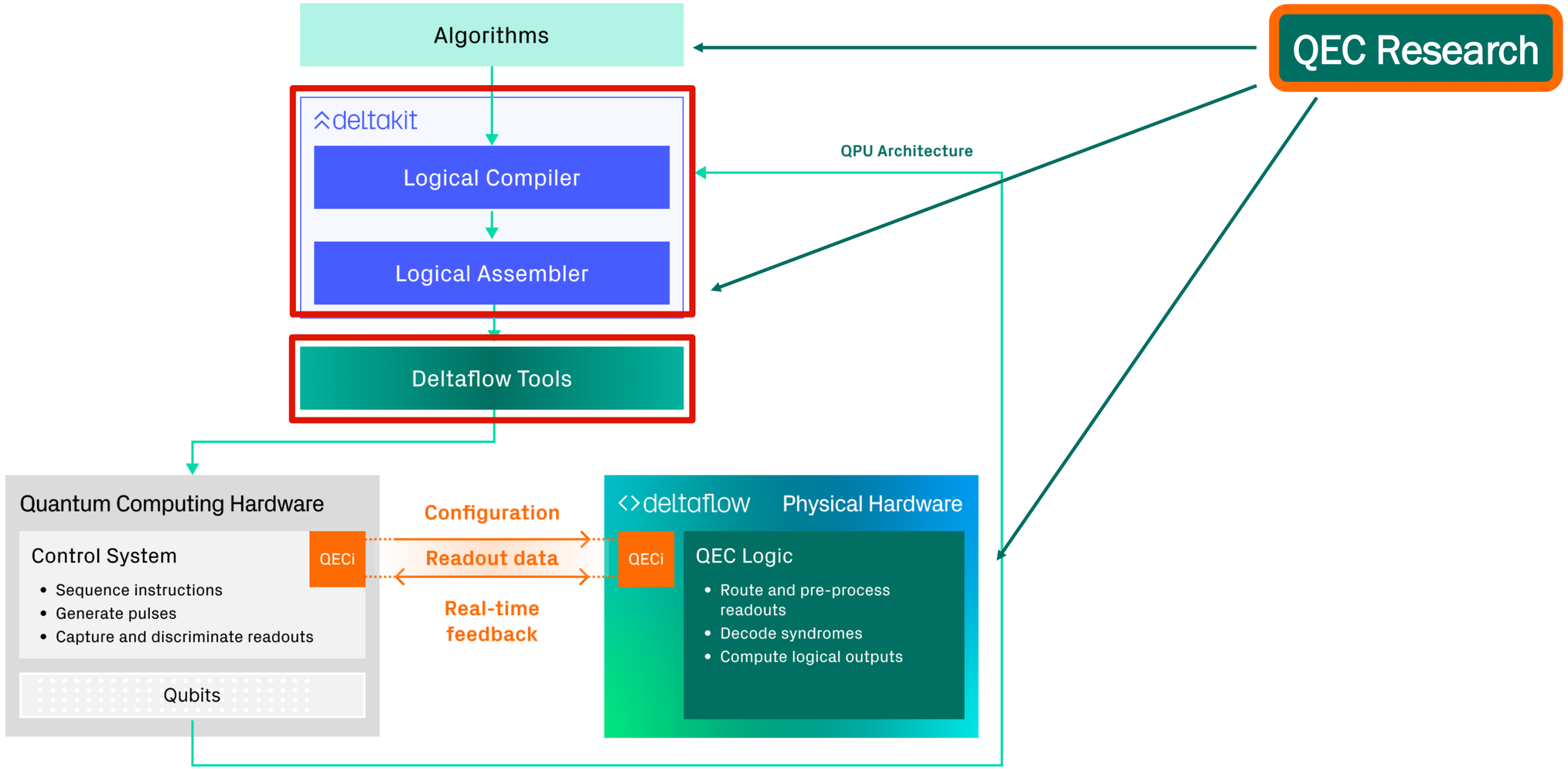
Annie Ray, Adrien Suau, Adnaan Walayat

# Our mission is to make quantum computers useful sooner

- One of the biggest challenges is dealing with noise in quantum systems.
- We are tackling the hard problem of implementing QEC in real-time.



# Our mission is to make quantum computers useful sooner.

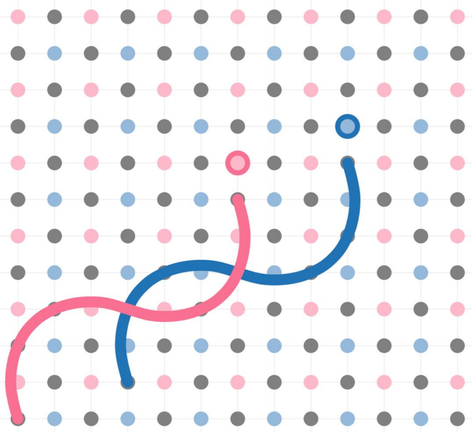


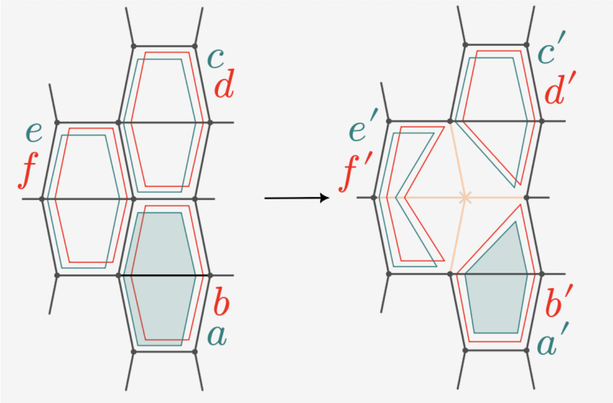
# QEC Research at Riverlane

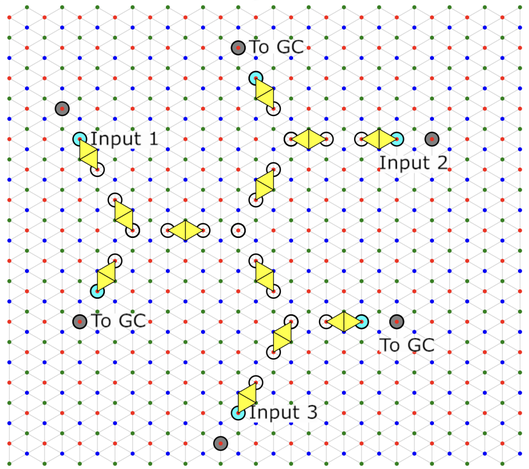
## Hardware-aware Research

*"Developing QEC strategies that address and exploit the idiosyncrasies of quantum hardware"*

1. Directional codes
2. ACID
3. Colour Code Decoding







1. Gyorgy Geher, et. al, "*Directional Codes: a new family of quantum LDPC codes on hexagonal- and square-grid connectivity hardware*", arXiv:2507.19430.
2. Stasiu Wolanski, "*Automated Compilation Including Dropouts: Tolerating Defective Components in Stabiliser Codes*", arXiv:2512.01943.
3. Mark Walters and Mark Turner, "*Minimum Weight Decoding in the Colour code is NP-hard*", arXiv:2603.04234.

# Hardware-aware Research at Riverlane



## 1. Directional Codes

- QLDPC codes:
  - Good QLDPC codes have $k \propto n$, $d \propto n$.
  - Encode multiple logical qubits per code block
  - Typically higher weight stabilisers than surface code.
  - Require long-range connectivity. Eg: BB codes.

- But engineering long-range couplers/interactions is hard for superconducting
- <u>Motivation:</u>
  - Can we exploit the native gate set of SC hardware to measure high weight stabilisers of qLDPC codes?
  - …without requiring long range couplers?

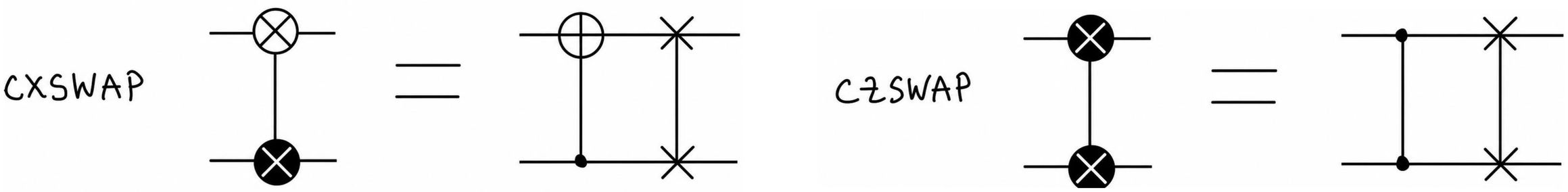
- Gyorgy P. Geher, David Byfield, Archibald Ruban, arXiv:2507.19430.

# Hardware-aware Research at Riverlane

- Inspired by the use of ISWAP gates for surface codes[1].
- ISWAP gate ~ Clifford equivalent to a CXSWAP or CZSWAP gate (henceforth CPSWAP).
- ISWAP is a native gate for superconducting qubits.

**Qubits interact and then their states get swapped,**

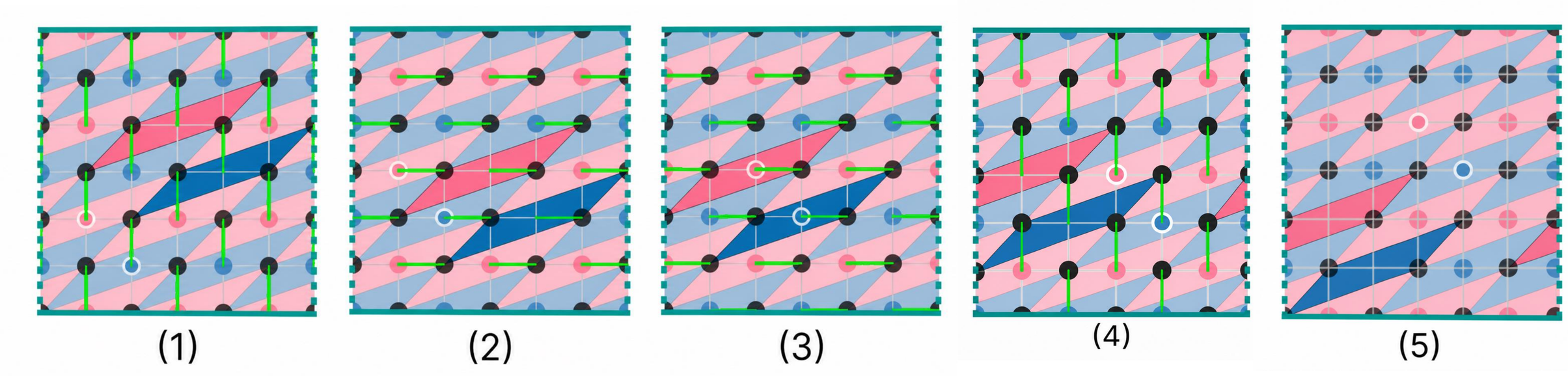
**effectively allowing interactions with a new set of qubits after each gate.**




1. M. McEwen, et. al., "*Relaxing Hardware Requirements for Surface Code Circuits using Time-dynamics*", Quantum 7, 1172 (2023).

# Hardware-aware Research at Riverlane

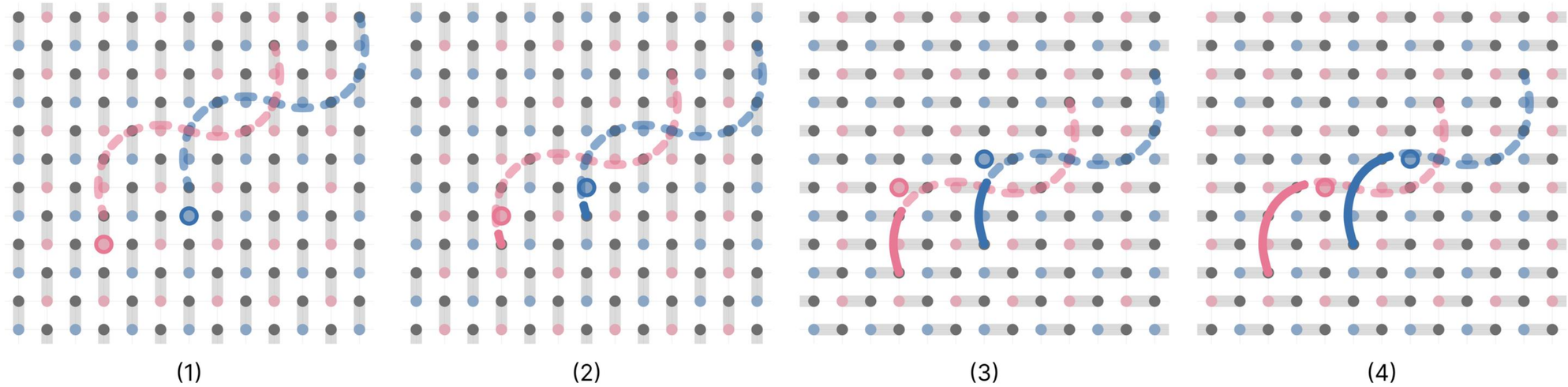
## Using CPSWAP to enable long-range interactions on square grid connectivity

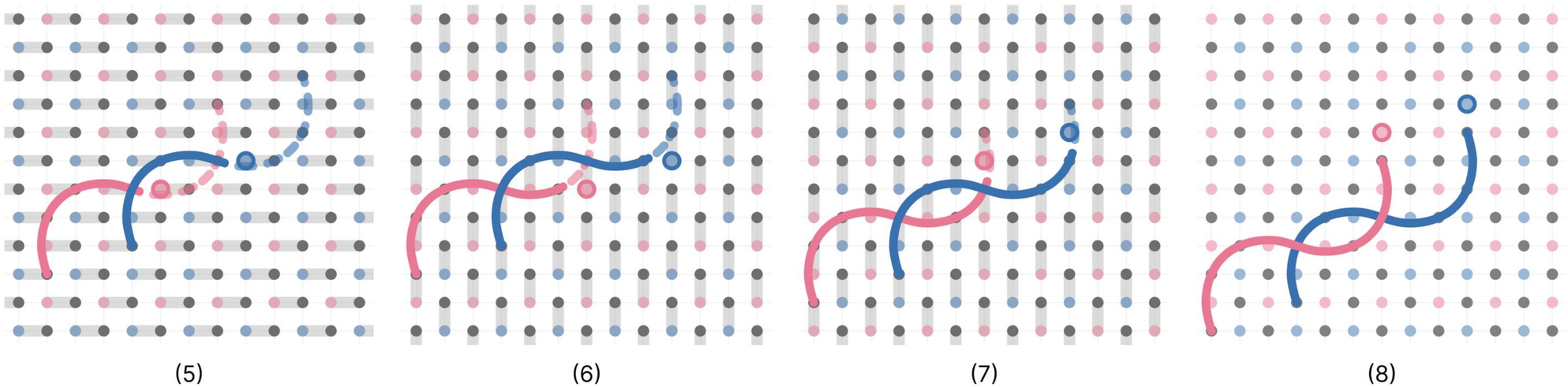



• Data qubits, • X-ancilla qubits, • Z-ancilla qubits, — CPSWAP gate

# Hardware-aware Research at Riverlane

## Discovery of new families of QLDPC codes : $N^2E^3N^2$ code

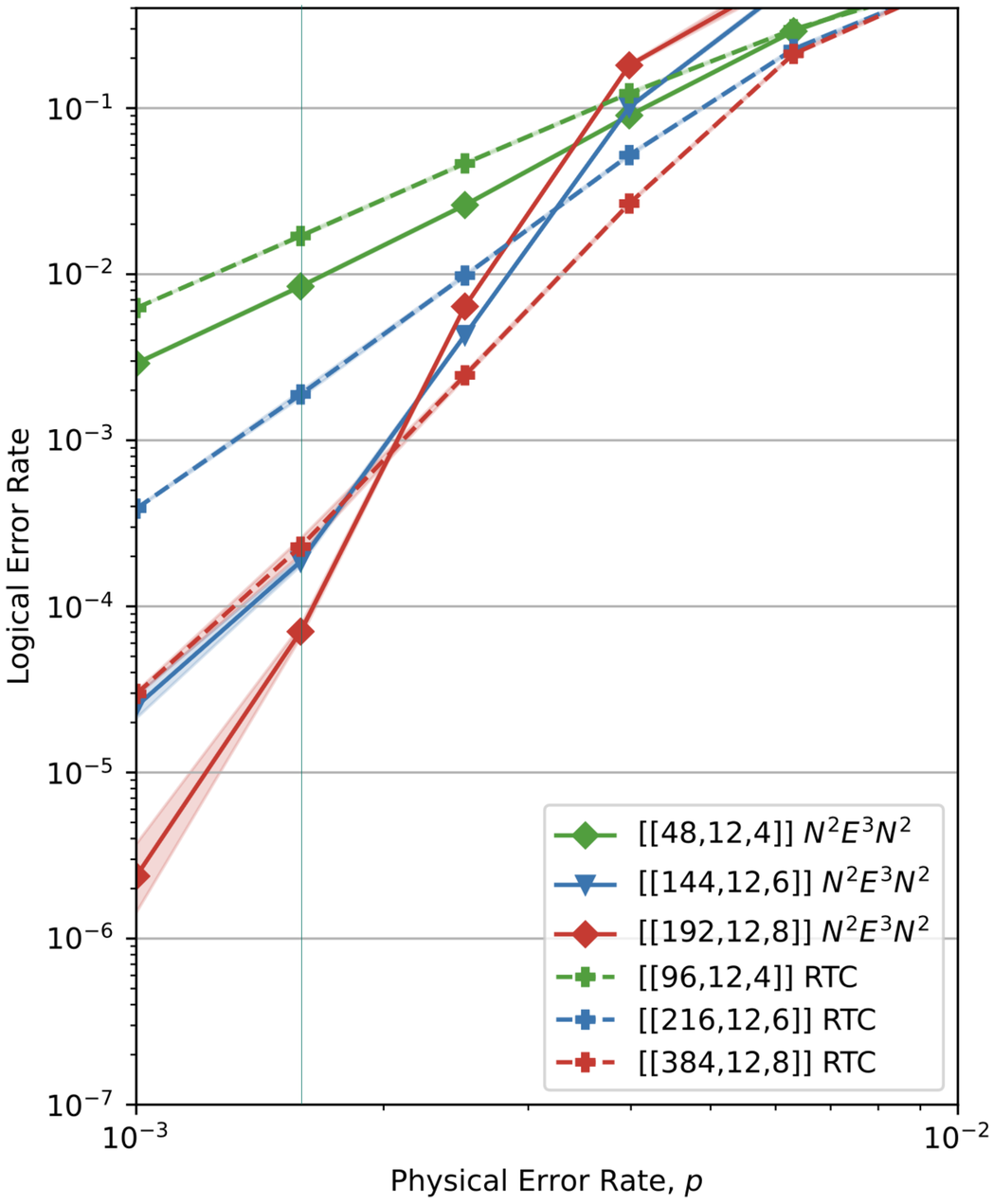


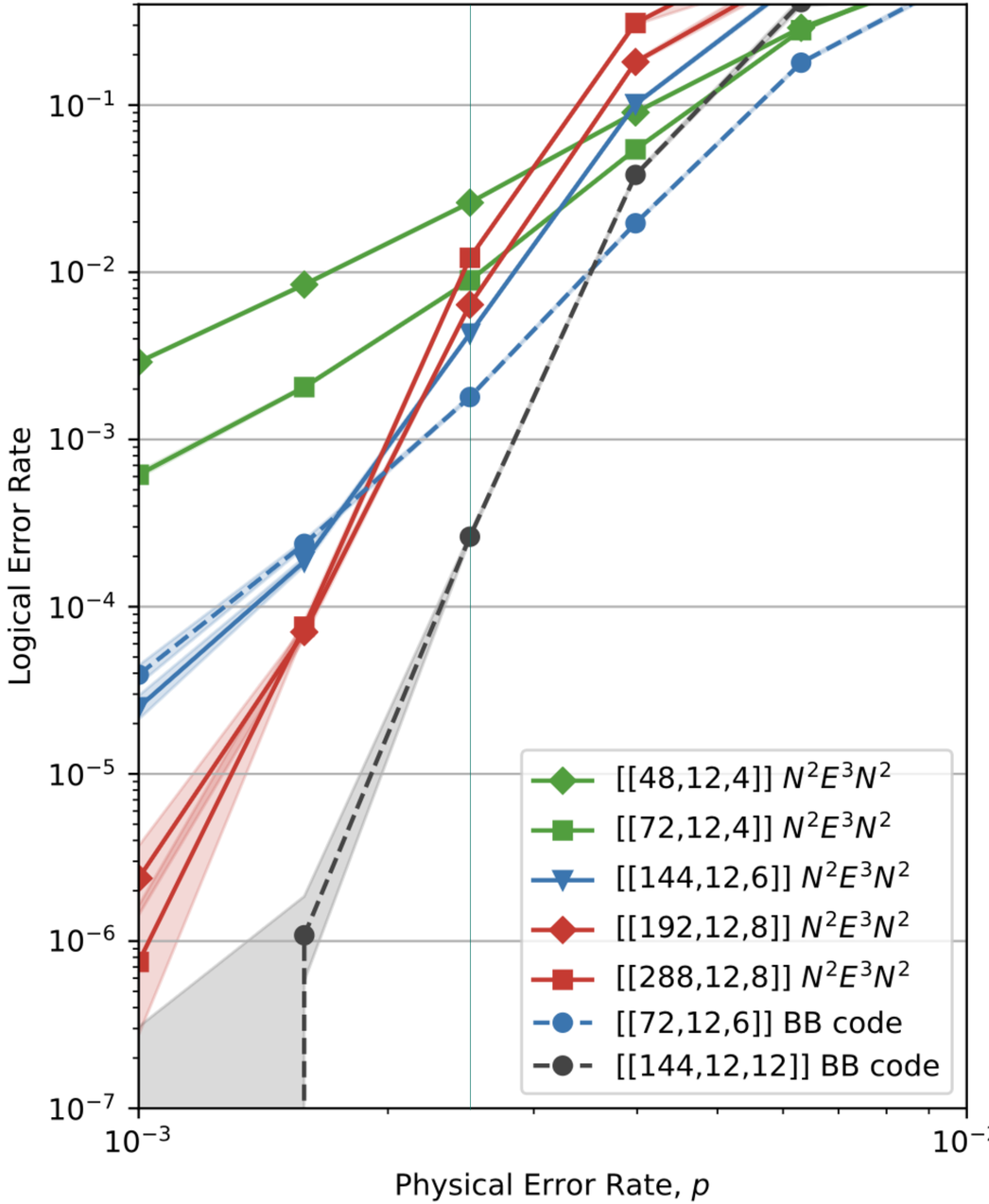




- n, n, e, e, e, n, n
- weight-7 stabilisers
- 12 logical qubits
- $[[\sim 6d^2, 12, \leq d]]$
- Periodic Boundaries

# Comparison to Toric and BB codes

(Decoder: Tesseract, short-beam)

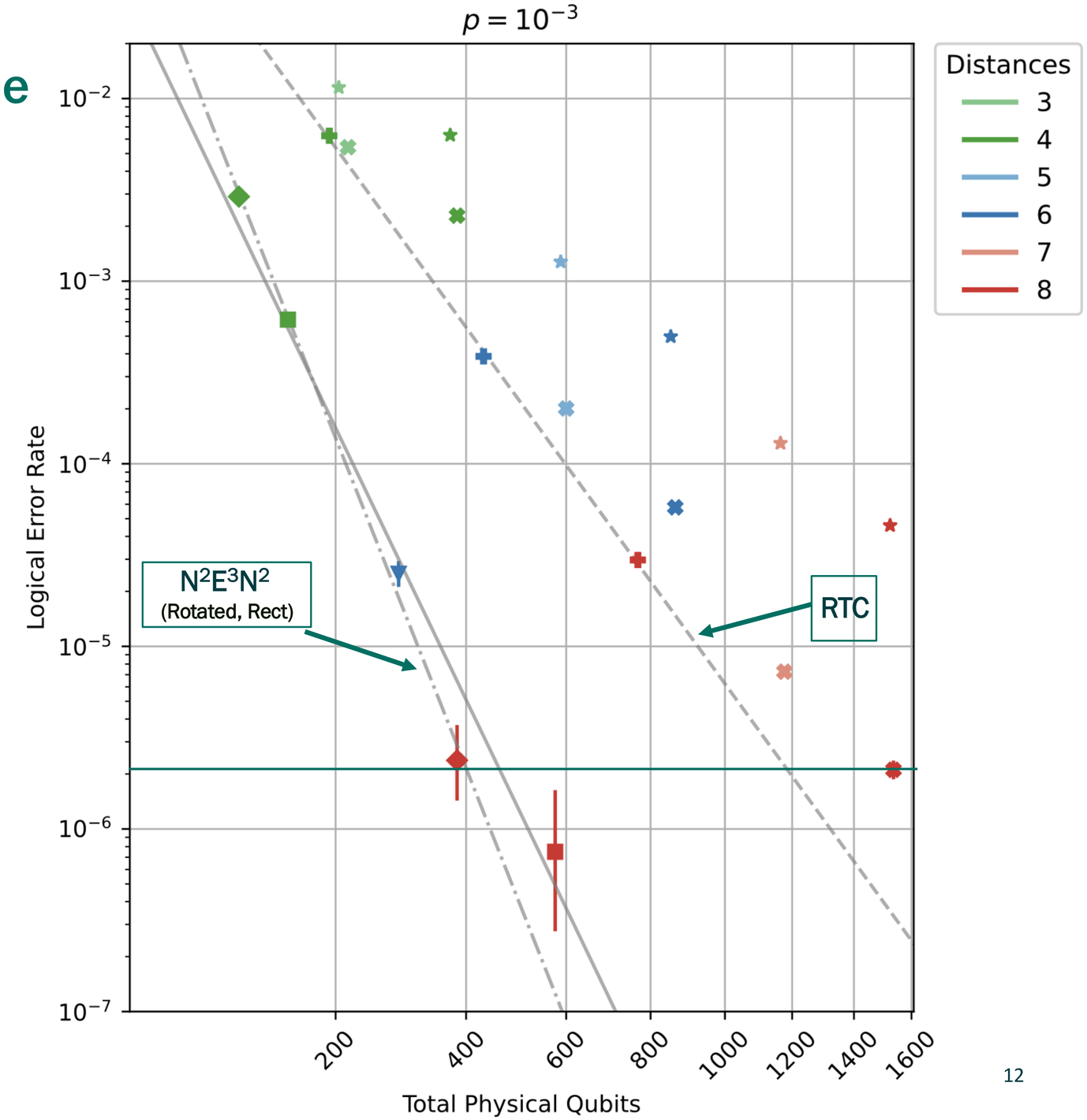


# Hardware-aware Research at Riverlane

- Reduced physical qubit overhead w.r.t. surface/toric code.
- At error rate of $10^{-3}$ (SI 1000),

  $N^2E^3N^2$ code requires 4x fewer physical qubits than the Rotated Toric code to achieve LER = $2 \times 10^{-6}$.



# Hardware-aware Research at Riverlane



## 1. Directional Codes: Summary

Gyorgy Geher, David Byfield, Archibald Ruban, arXiv:2507.19430.

- A new paradigm for designing QLDPC codes tailored to platforms with local connectivity.
- Future scope:
  - **Embedding the codes on a plane instead of a torus.**
  - **Logical architectures based on Directional Codes.**

- Checkout the poster presentation **#49** by **Archibald Ruban** on Tuesday for more details!

# Hardware-aware Research at Riverlane

## 2. Automated Compilation Including Dropouts

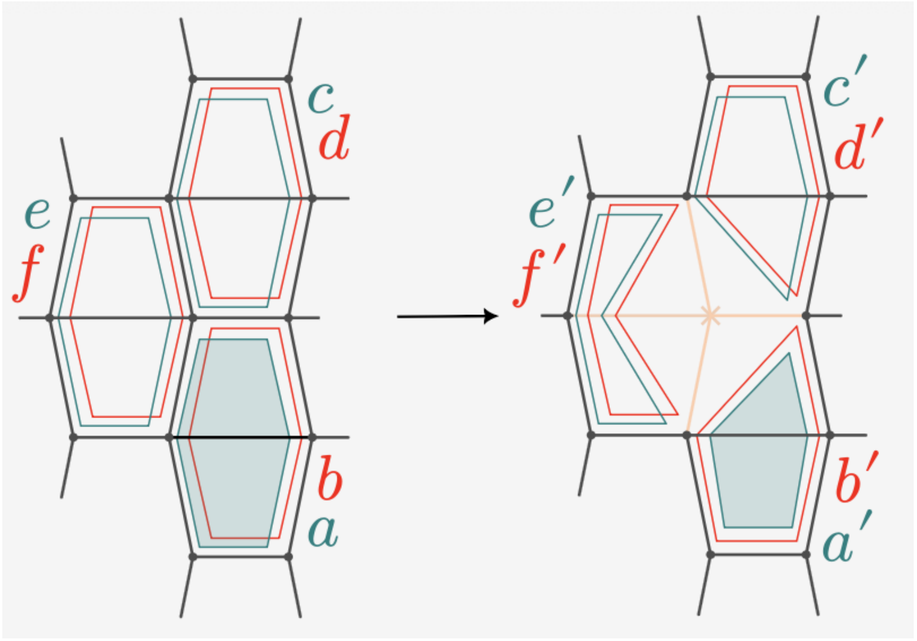
- Motivation:
  - Deal with fabrication defects (dropouts) in fixed-qubit hardware, eg: superconducting qubits.
  - Design syndrome extraction circuits for QEC codes in the presence of defective qubits or couplers.
  - Inspired by LUCI[1] for the surface code with dropouts, generalised to arbitrary CSS codes.
- Stasiu Wolanski, arXiv:2512.01943.
- Automates the design of syndrome extraction circuits for general CSS codes given some qubit connectivity and dropout configuration.
- First study of the colour code and the BB codes[2] subject to defects.

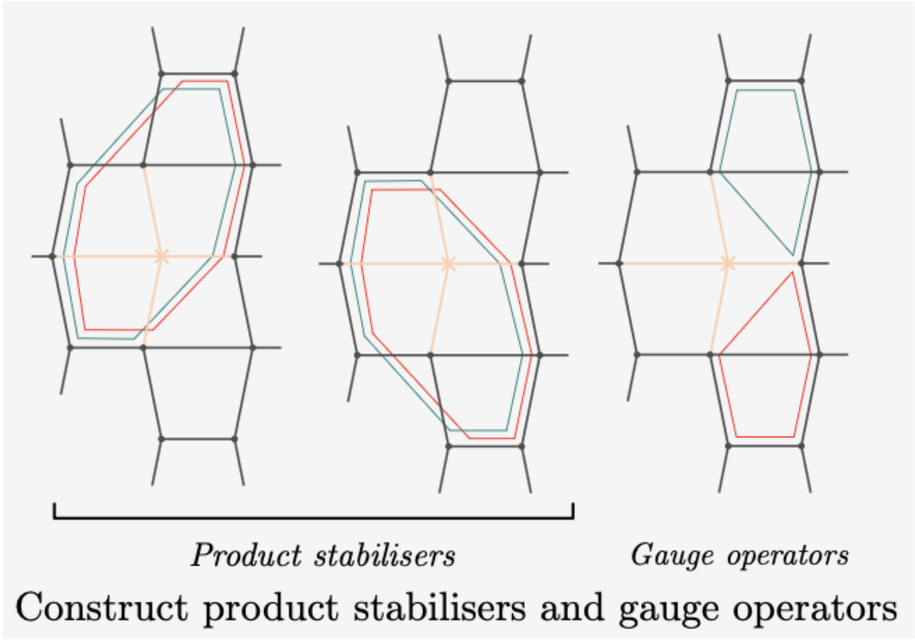

1. D. Debroy, et. al., "*LUCI in the Surface Code with Dropouts*", Quantum 9, 1936 (2025).
2. S. Bravyi, et. al., "*High-threshold and low-overhead fault-tolerant quantum memory*", Nature (2024).

# Hardware-aware Research at Riverlane

## I. Dealing with defective qubits

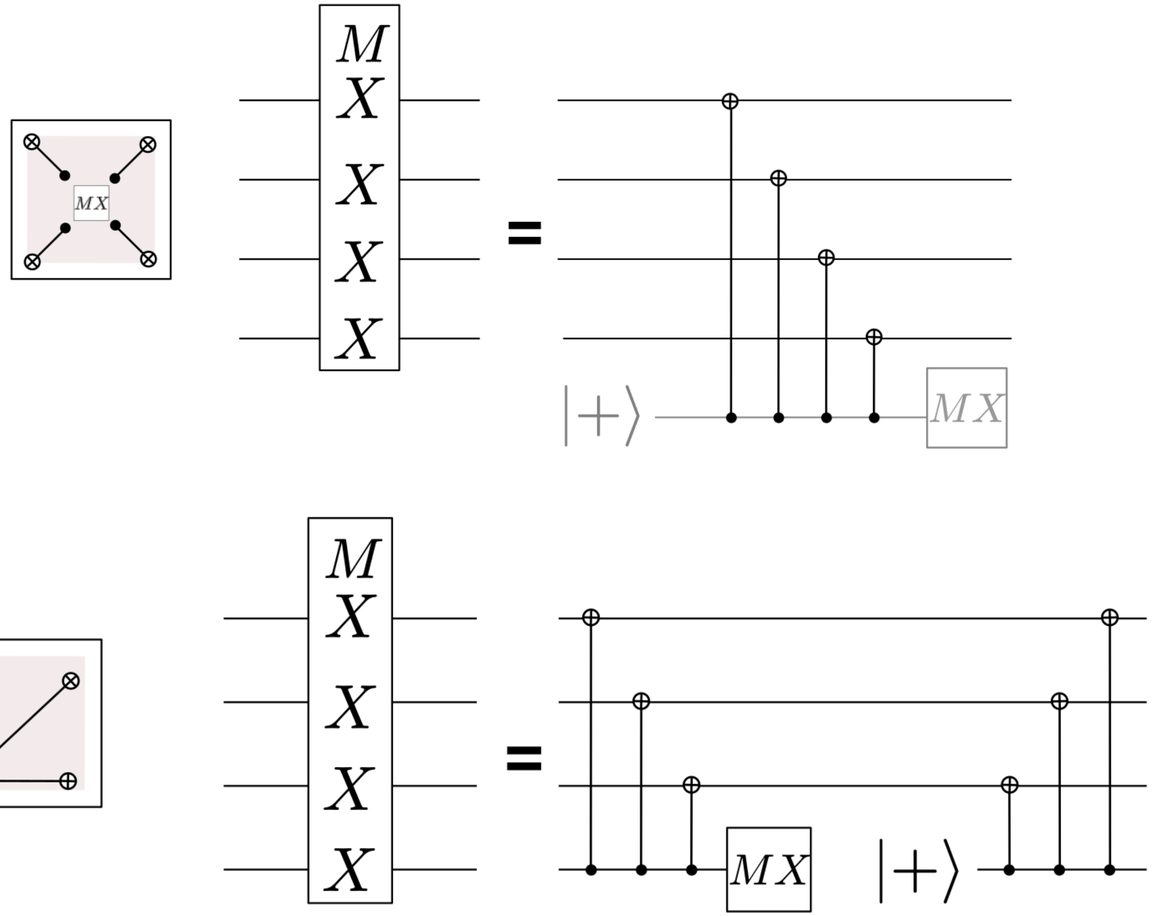
- Defects turn stabilisers into anti-commuting (**gauge**) operators.
- Can still construct product operators (**superstabilisers**) that commute with the original stabiliser group.
- Measure the gauge operators, and multiply outcomes to obtain **superstabiliser** measurements.
- The resulting subsystem code can still protect logical info, perhaps with slightly lower distance…



# Hardware-aware Research at Riverlane

## II. Taking advantage of the morphing[1] perspective of syndrome extraction

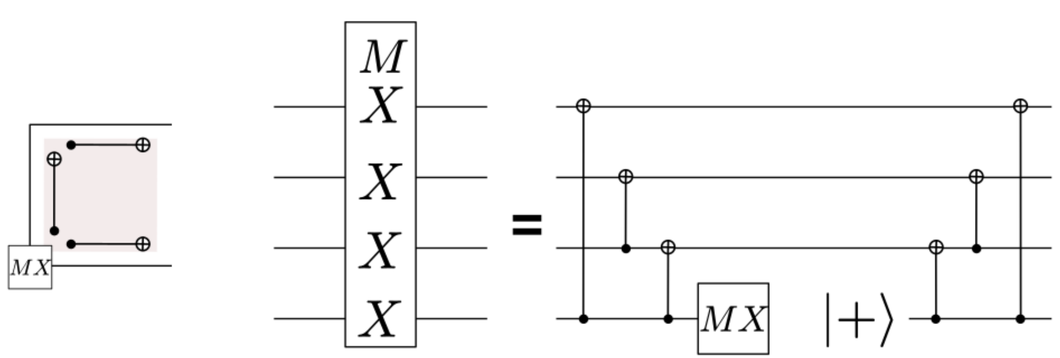


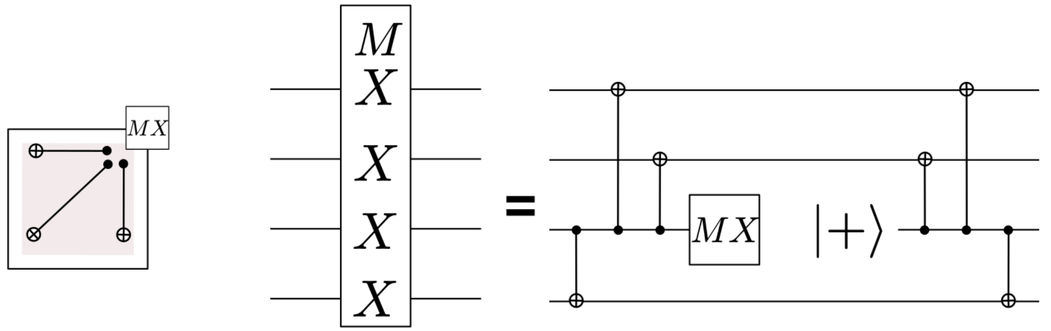

- Can contract a stabiliser onto any single data qubit in its support.
- No need for designated ancilla qubits.
- Can't contract both an X-type & Z-type stabiliser on a single qubit at the same time.
- Typically, need **at least 2 layers or "sub-rounds" to measure all X and Z stabilisers of a CSS code.**

1. M. Shaw, et. al., "***Lowering Connectivity Requirements for Bivariate Bicycle Codes Using Morphing Circuits***", PRL 134, 090602 (2025).

# Hardware-aware Research at Riverlane

## II. Dealing with defective couplers



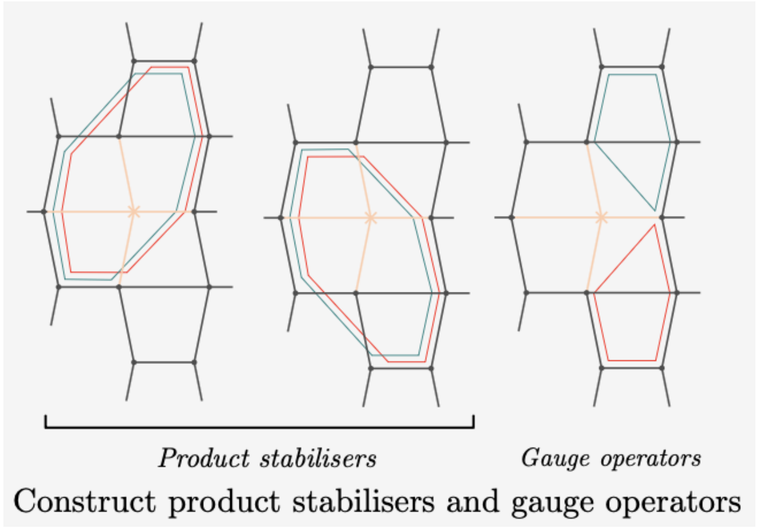

We can use any qubit as the measurement qubit…

Different possible contractions of the stabiliser can help avoid the use of defective couplers.

# Hardware-aware Research at Riverlane

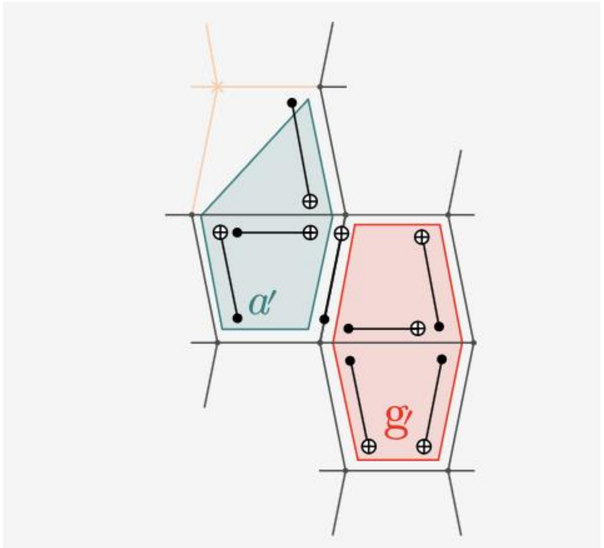
## ACID: Putting these together…

### 1. Deal with defective qubits

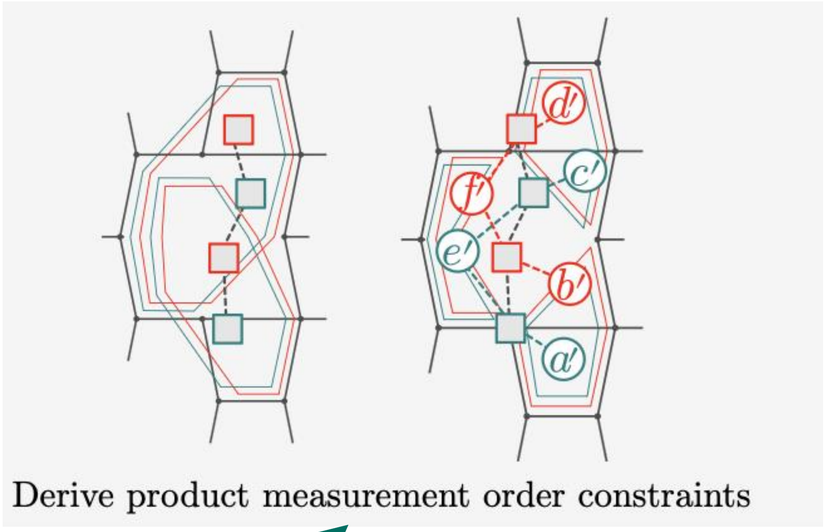



Construct product stabilisers and gauge operators

### 2. Deal with defective couplers




### 3. Find compatible schedules

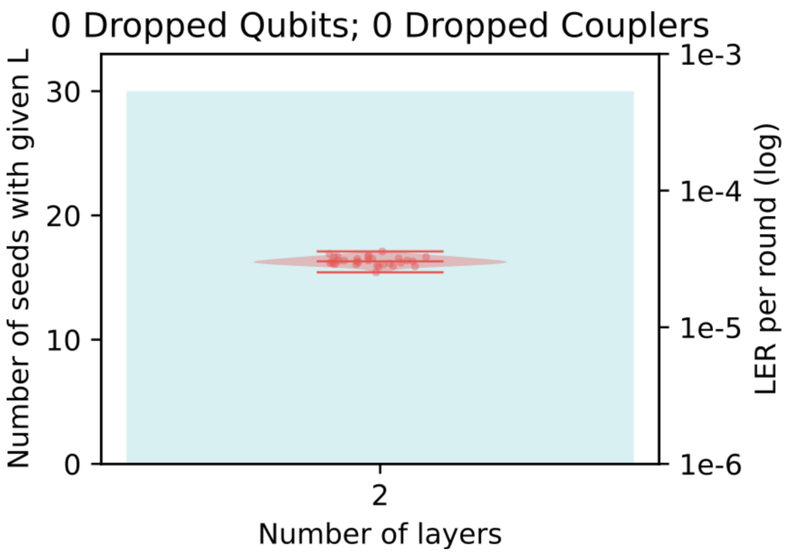


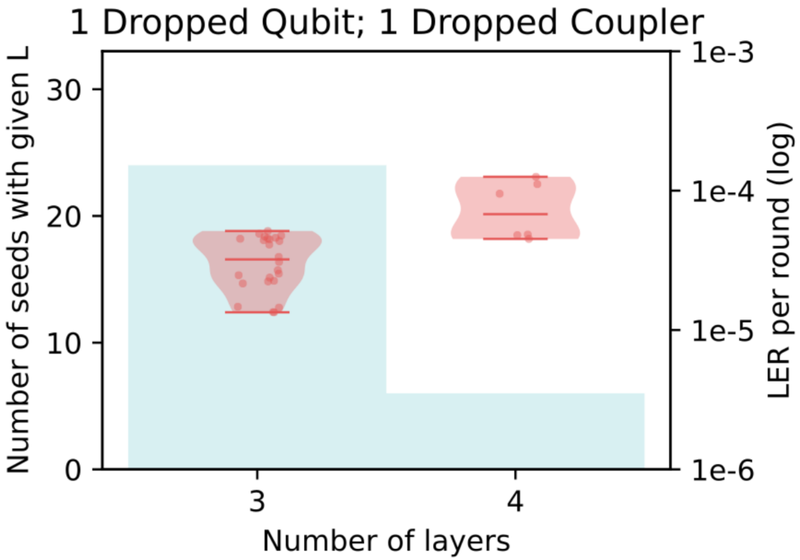

Derive product measurement order constraints

Using a SAT solver to optimise for low depth syndrome extraction circuits.

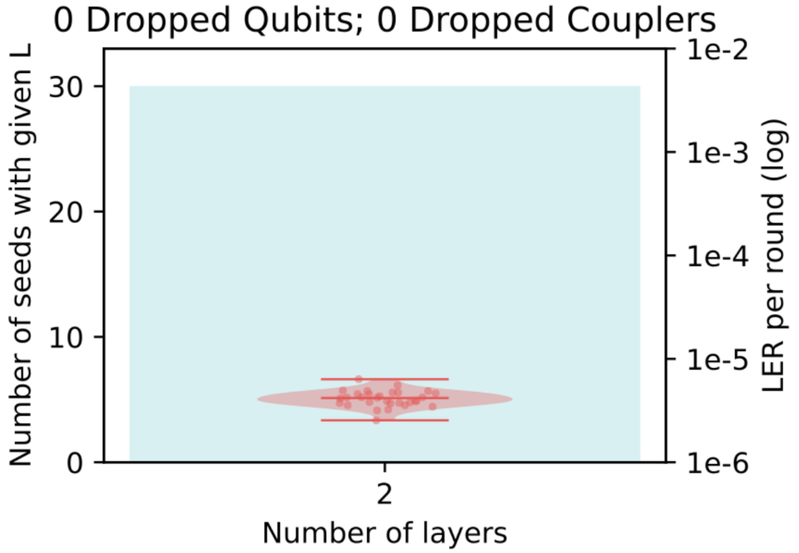
# The IBM BB code [[144, 12, 12]] on ACID

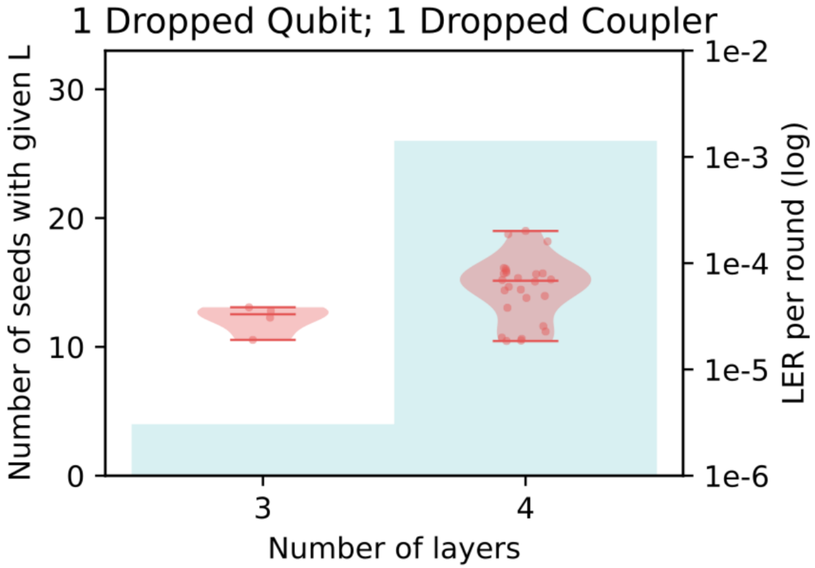
## Degree-6 Qubit connectivity graph




## Degree-5 Qubit connectivity graph



# Hardware-aware Research at Riverlane

## 2. **A**utomated **C**ompilation **I**ncluding **D**ropouts: Summary

Stasiu Wolanski, arXiv:2512.01943.

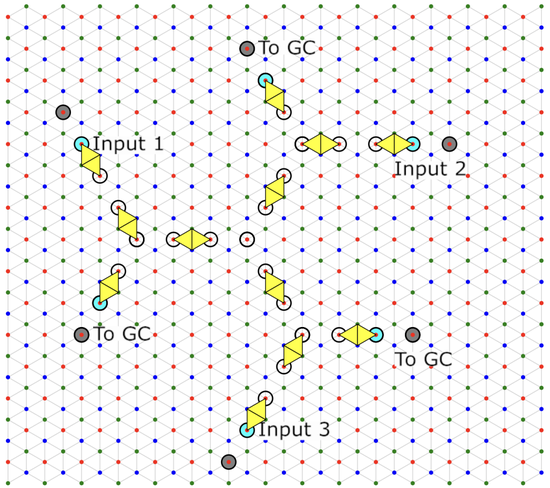
- A systematic way to design syndrome extraction circuits for arbitrary CSS codes in the presence of dropouts.
- Future scope:
  - **Code specific heuristics for improved circuit design, eg. for the BB codes.**

- Checkout the poster presentation **#44** by **Stasiu Wolanski** on Tuesday for more details!

# Hardware-aware Research at Riverlane

## 3. Minimum weight decoding of the colour code is NP-Hard

- Motivation:
  - Colour code: the “more-bang-for-your-buck” relative of the surface code.
  - Fewer physical qubits for similar LERs.
  - Transversal Clifford operations.
  - Permits magic state cultivation/distillation protocols.
  - Optimal decoding of the Colour code remained unsolved…




- Mark Walters and Mark Turner, arXiv:2603.04234.
  - Reduce the exact colour code decoding problem to 3-SAT, proving it is NP-hard.
  - Keep calm, and find good heuristics for polynomial-time approximate solutions ☺

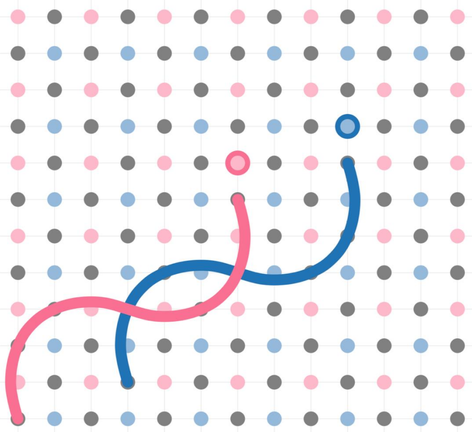
- Checkout talk by Mark Walters tomorrow @ 12pm for more details!

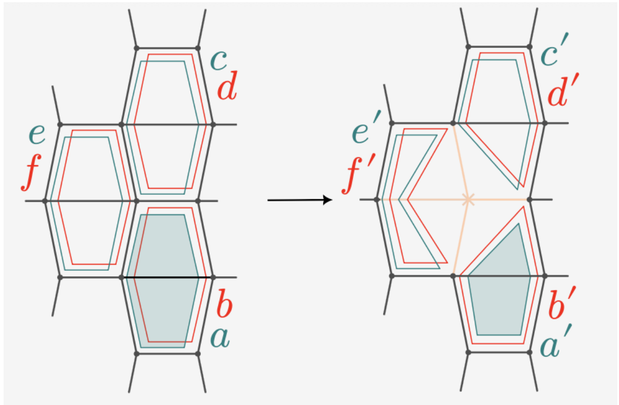
# QEC Research at Riverlane: Recap

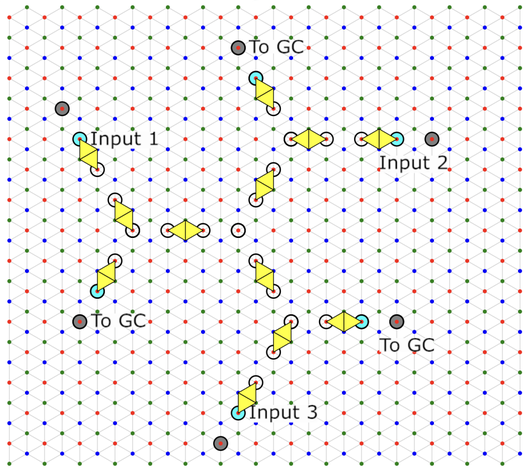
## Hardware-aware Research

*"Developing QEC strategies that address and exploit the idiosyncrasies of hardware"*

1. Directional codes
2. ACID
3. Colour Code Decoding



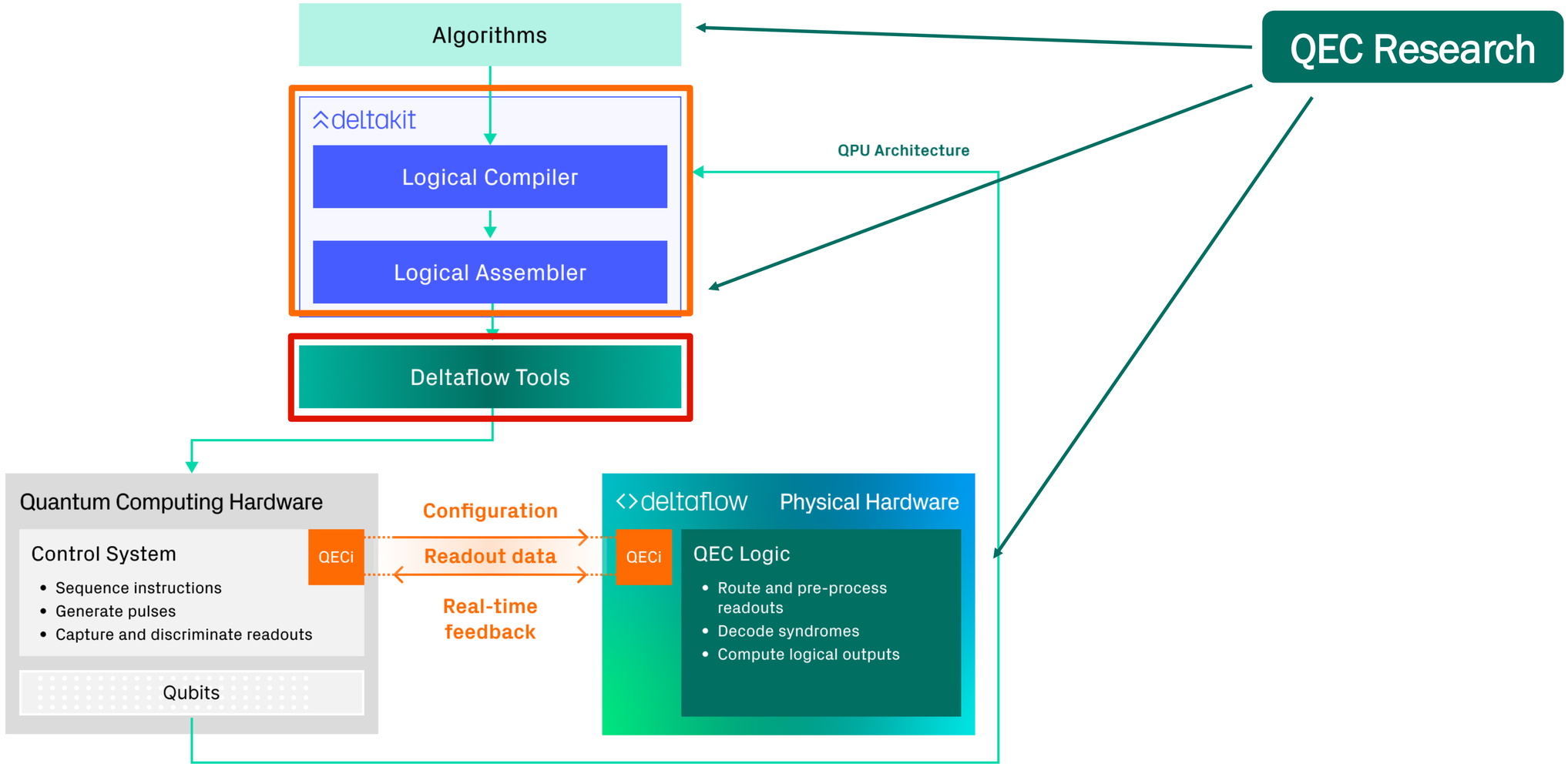




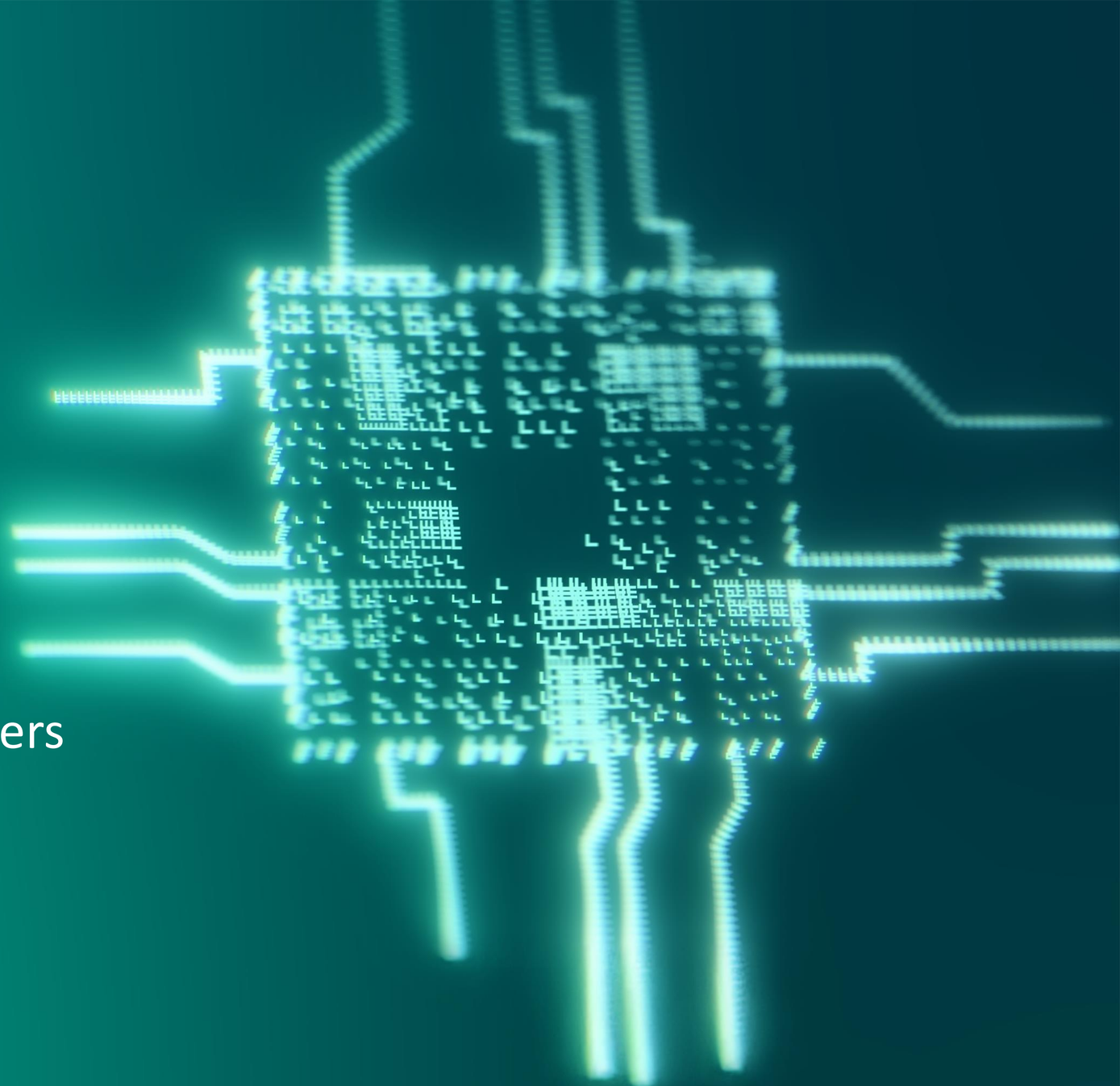
1. G. P. Geher, et. al., "*Directional Codes: a new family of quantum LDPC codes on hexagonal- and square-grid connectivity hardware*", arXiv:2507.19430.
2. S. Wolanski, "*Automated Compilation Including Dropouts: Tolerating Defective Components in Stabiliser Codes*", arXiv:2512.01943.
3. Mark Walters and Mark Turner, "*Minimum Weight Decoding in the Colour code is NP-hard*", arXiv:2603.04234.

# Our mission is to make quantum computers useful sooner.

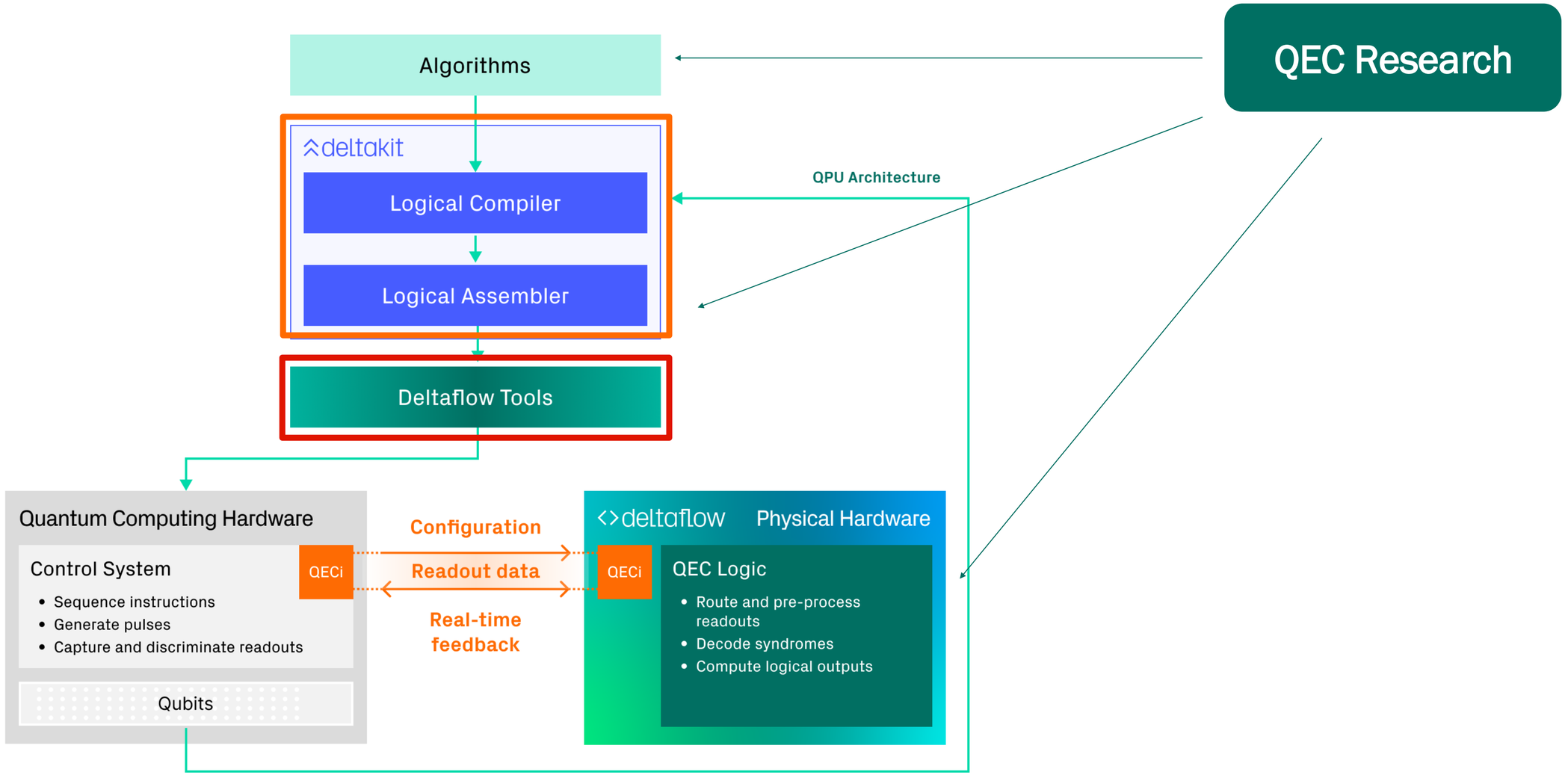


# Deltakit

An open-source QEC toolkit for researchers and companies



# Our mission is to make quantum computers useful sooner.



# Deltakit

## A QEC toolkit for researchers and companies

000 What is needed to compute interesting things?

001 Program generation in Deltakit

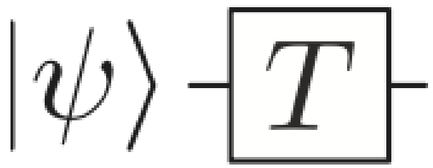
010 Deep-dive into a pipeline

011 Ways to improve your code/decoder/implementation/hardware

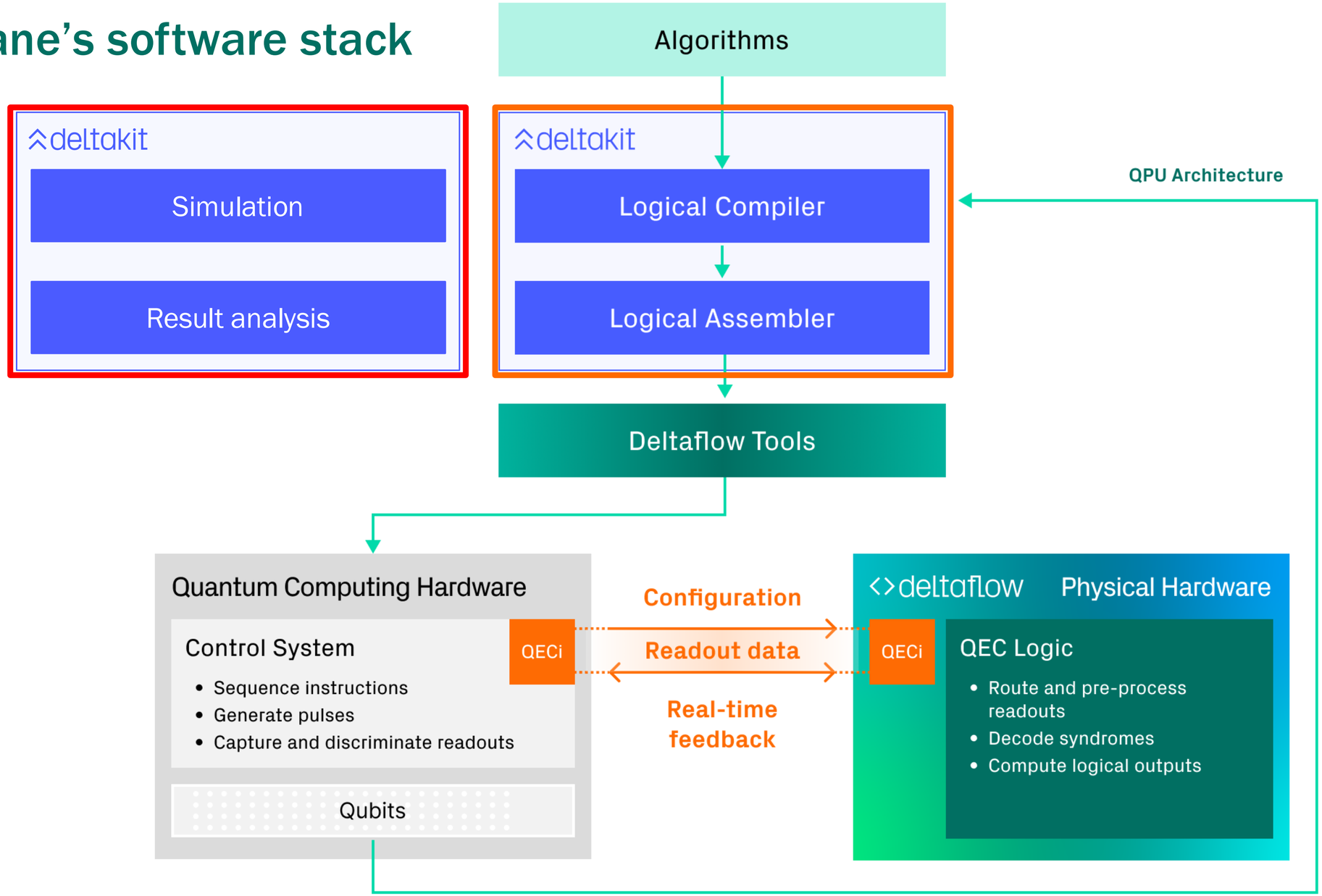
100 Spoilers

# Universal computation in QEC

## What it takes to compute?



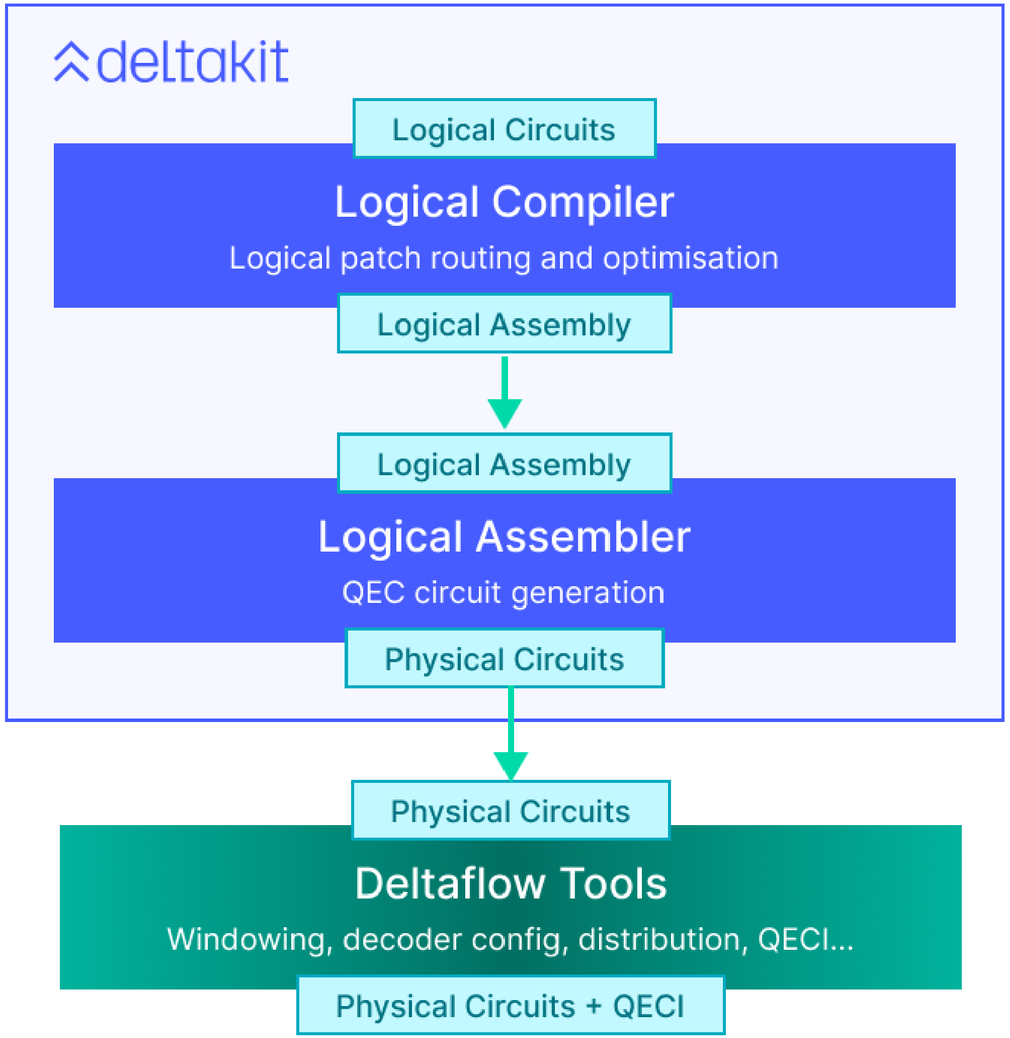
# Riverlane's software stack



# Riverlane's software stack

## Zooming on program generation

- Compiler architecture (inspired by MLIR)
- Python framework (xDSL)
- From logical programs to control-system instructions



# What is a compiler?

## Data representation(s)

**Intermediate Representation (IR):**

Representation of the state of the program at one point during the compilation.

**Dialect:**

Subset of an IR. Ideally small and mutually orthogonal.

**Example:**

stim.Circuit is an IR that could be decomposed as:

1. gates: for quantum gates (Clifford)
2. struct: for REPEAT blocks
3. det: for detectors
4. obs: for observables

## Data transformation(s)

**Pass:**

Code applied on an IR.

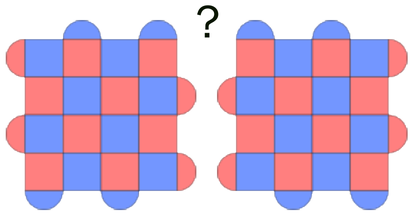
Can **modify** an IR (optimisation, lowering), or **annotate** after analysis.

**Examples:**

- A pass on det that would try to reduce the average number of measurements in each detector.
- A pass unrolling REPEAT blocks, removing the struct dialect from the IR.

# Riverlane's software stack

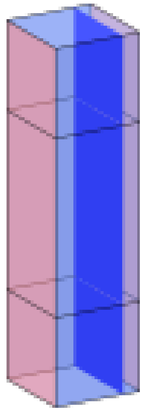
## Example of a pipeline



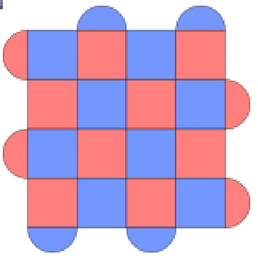

START

compute-boundary-parity

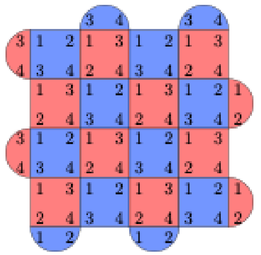
backpropagate-observables



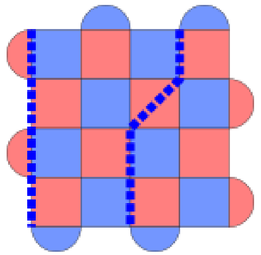
lower-patch-declaration



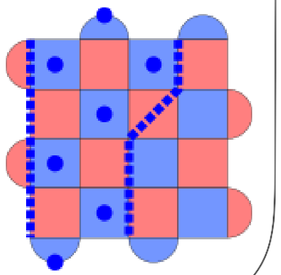
patch-to-plaquettes



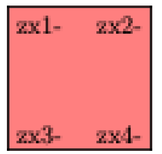
schedule-plaquettes

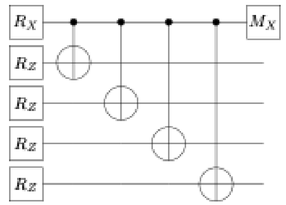


place-observables



plaquettes-to-circuits




transversal-op-to-circuit

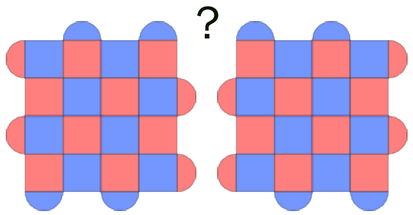
located-observable-to-move

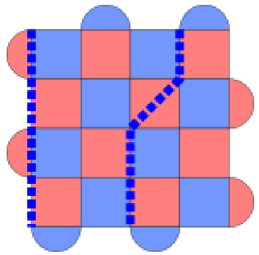
plaquette-to-qstruct

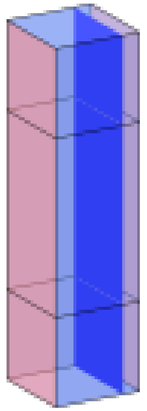
END

# Riverlane's software stack

## Control-flow awareness





START

compute-boundary-parity

backpropagate-observables

lower-patch-declaration

patch-to-plaquettes

schedule-plaquettes

plaquettes-to-circuits

plaquette-to-qstruct

place-observables

transversal-op-to-circuit

located-observable-to-move

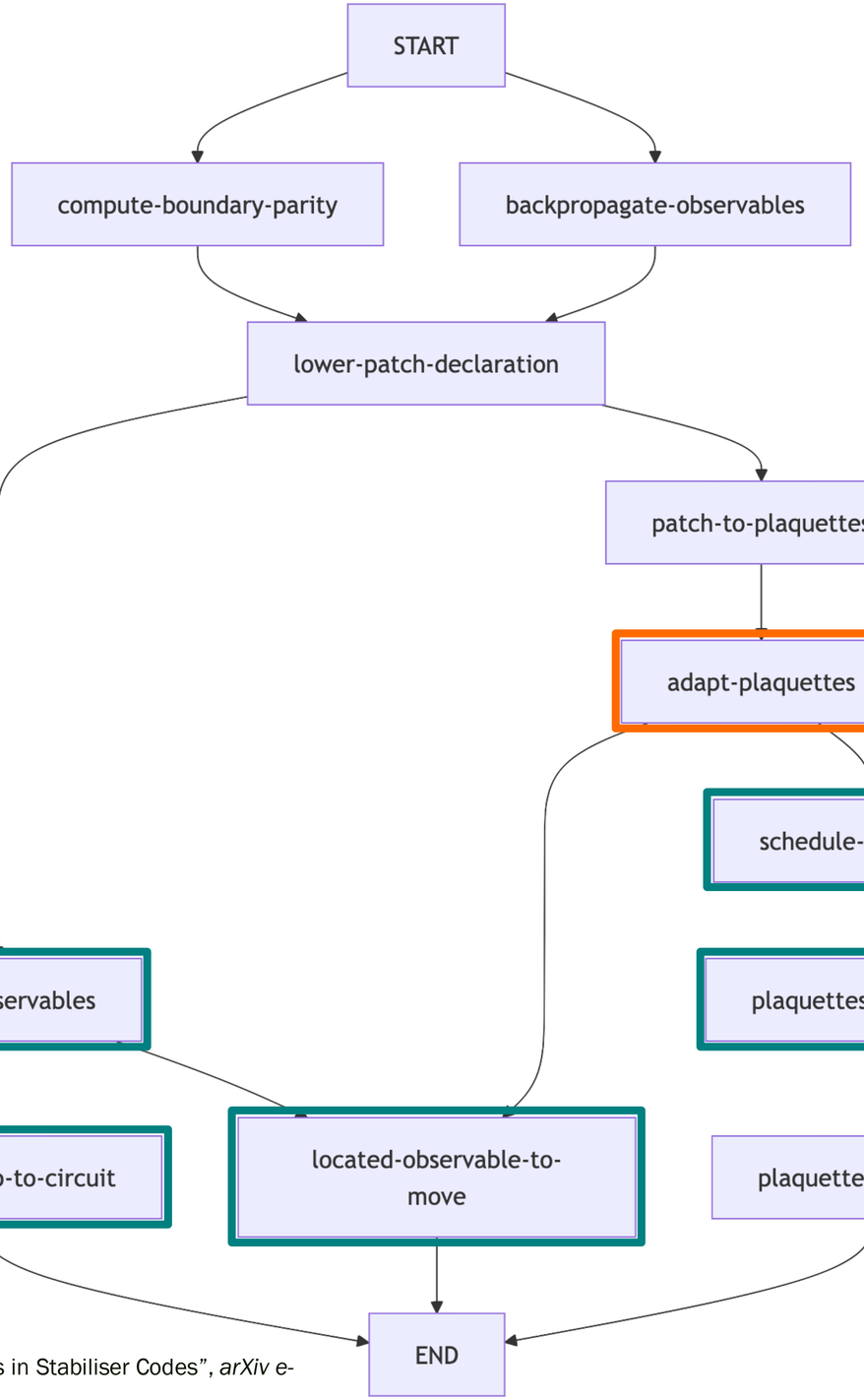
END

Significant changes

Small changes

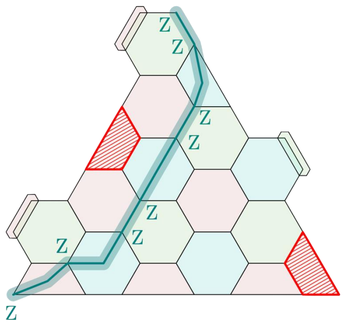
# Riverlane's software stack

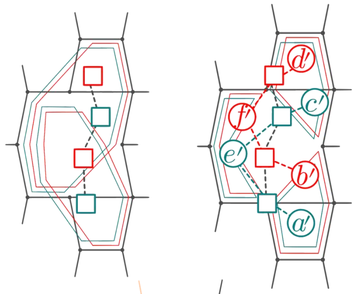
## Example of a pipeline on ACID

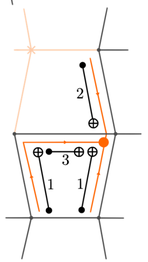



Significant changes

Small changes



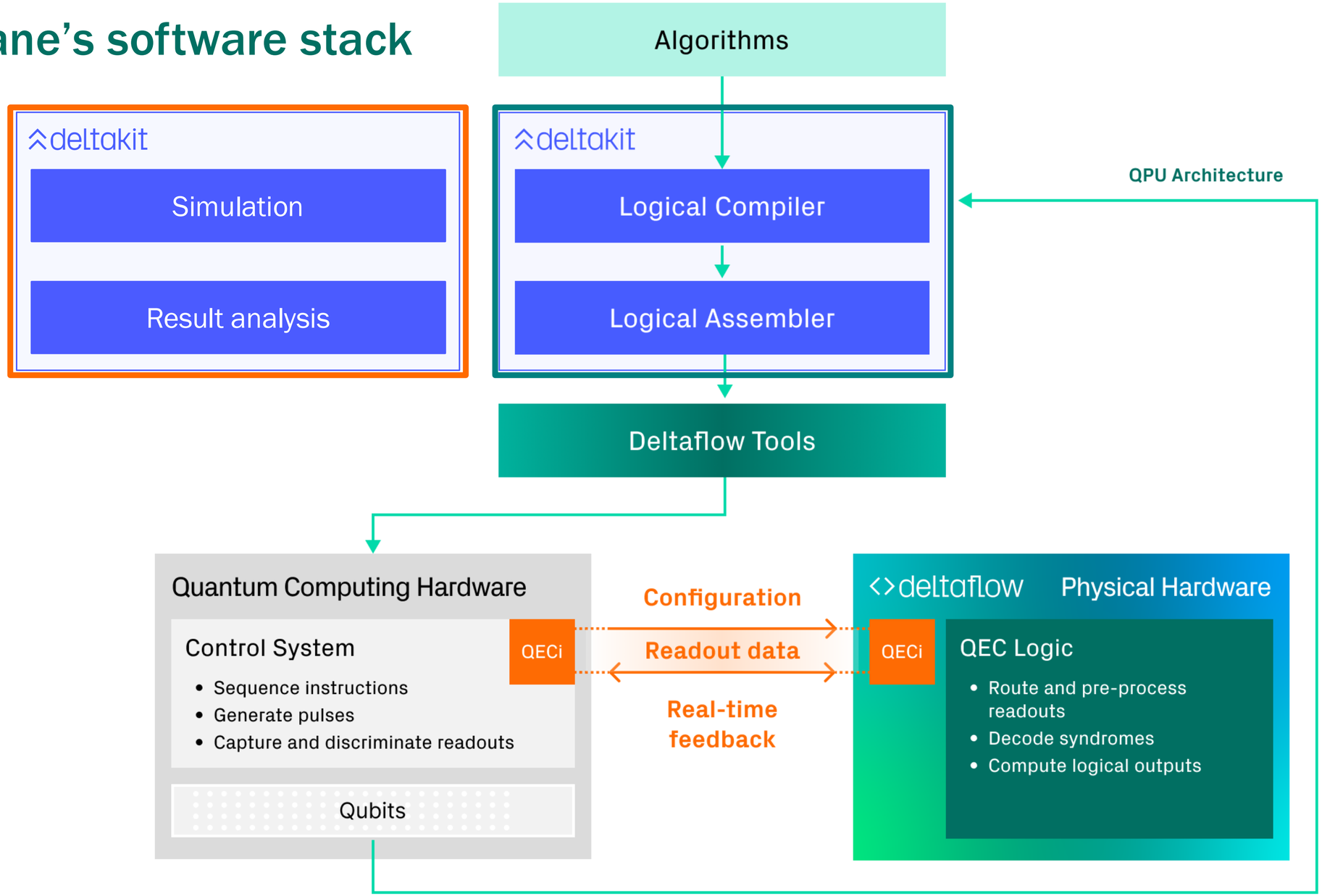






Wolanski, S., "Automated Compilation Including Dropouts: Tolerating Defective Components in Stabiliser Codes", *arXiv e-prints*, Art. no. arXiv:2512.01943, 2025. doi:10.48550/arXiv.2512.01943.

# Riverlane's software stack

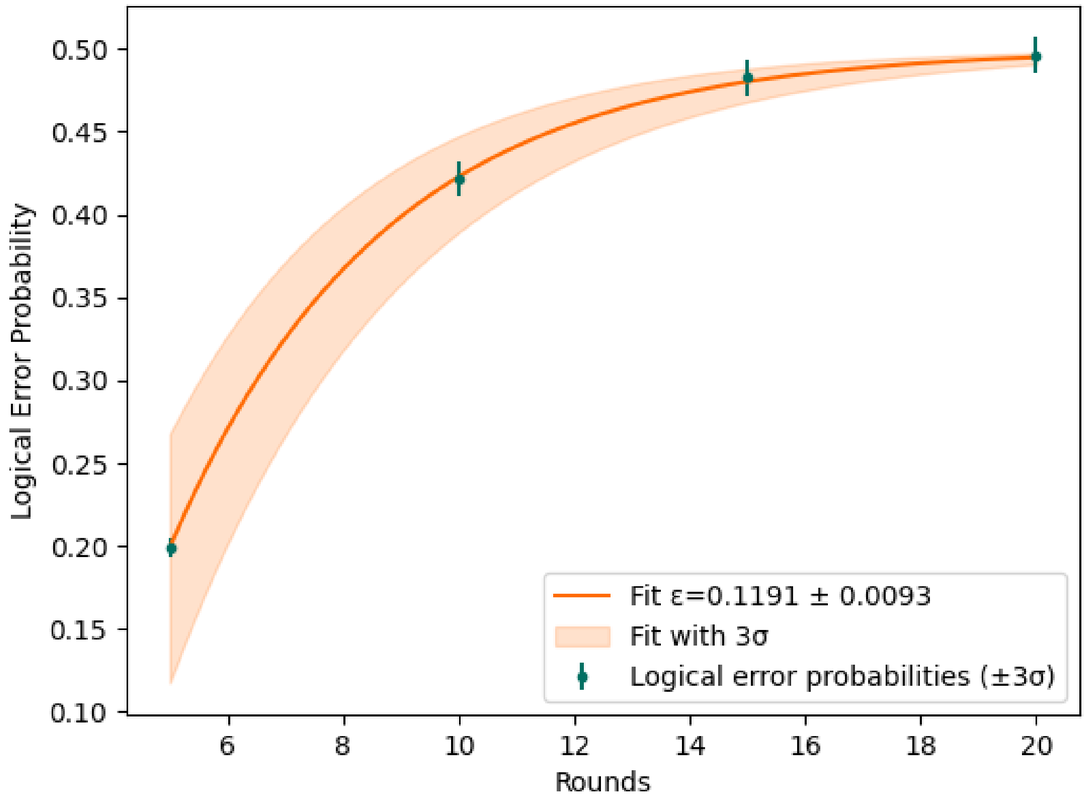


# Ways to improve performance?

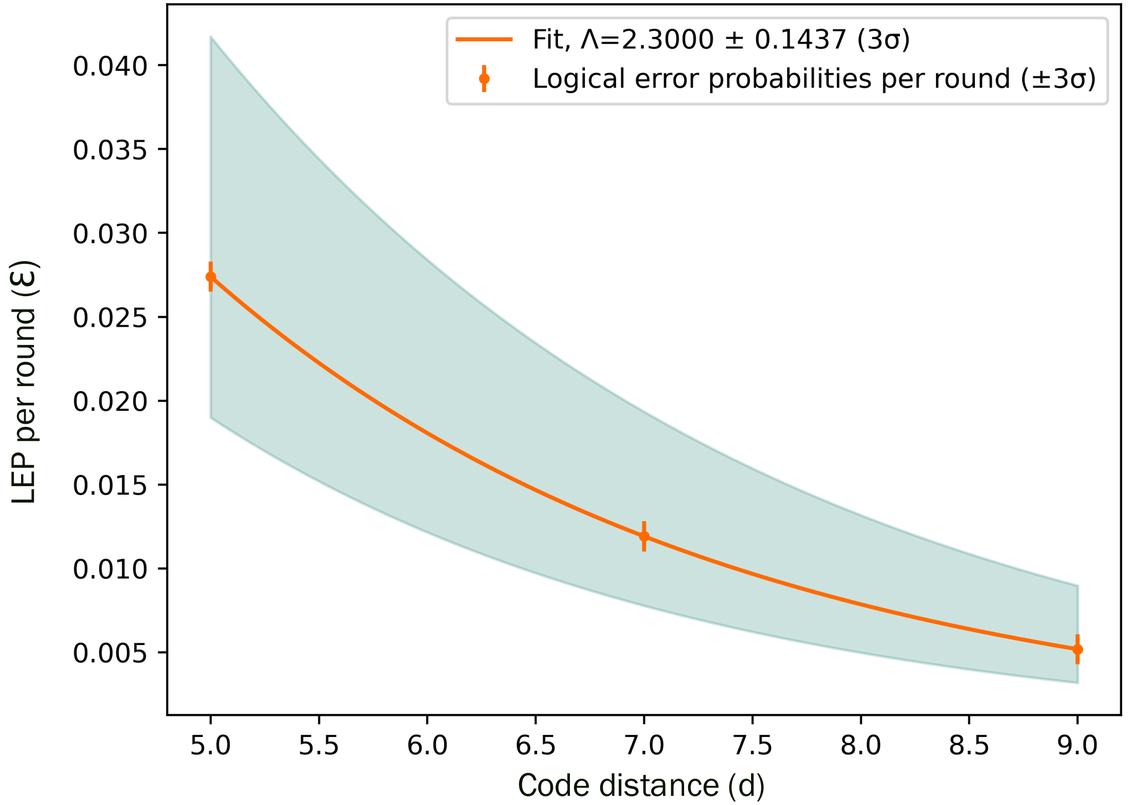
## What's the first step?

Benchmarking!

Logical error probability (LEP) per round: ε



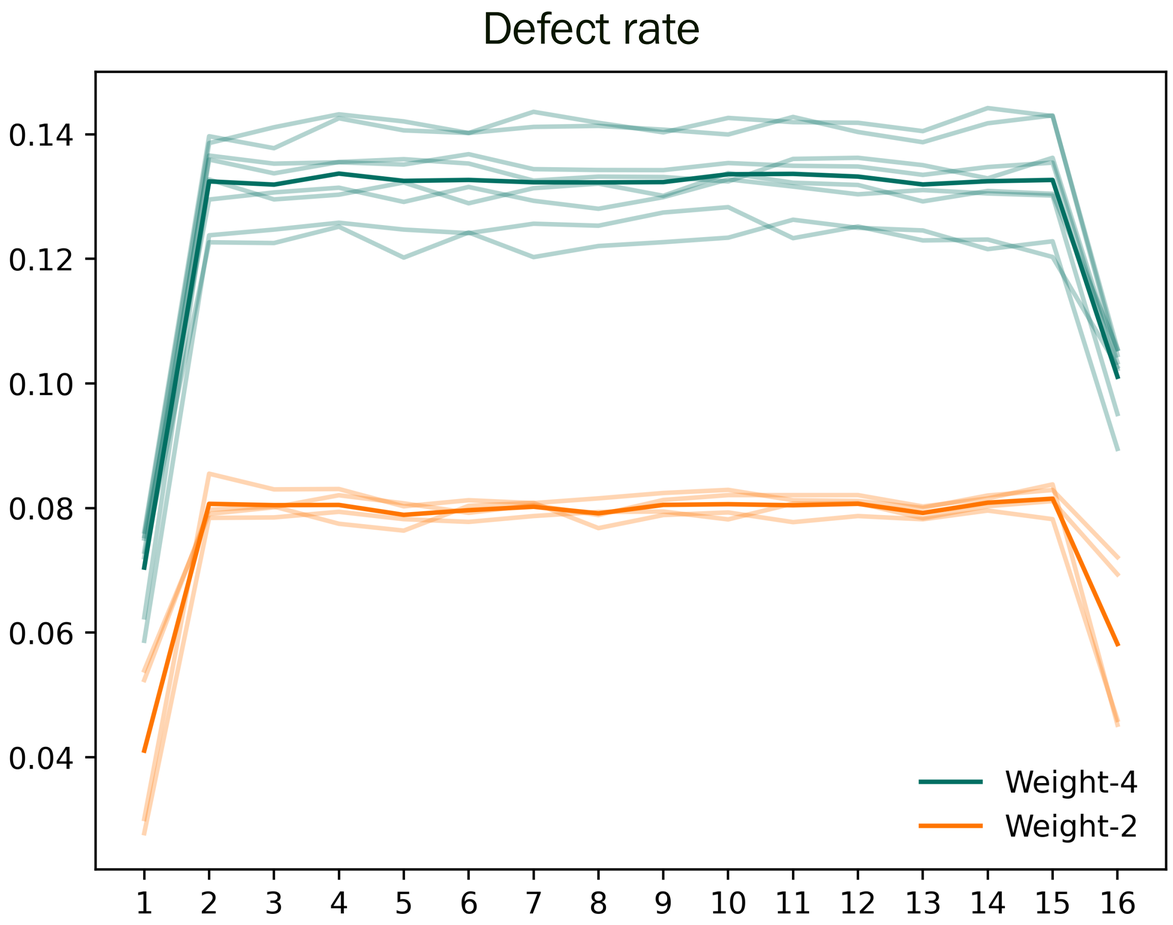

Error suppression factor Λ: $\varepsilon \propto \Lambda^{(d-1)/2}$

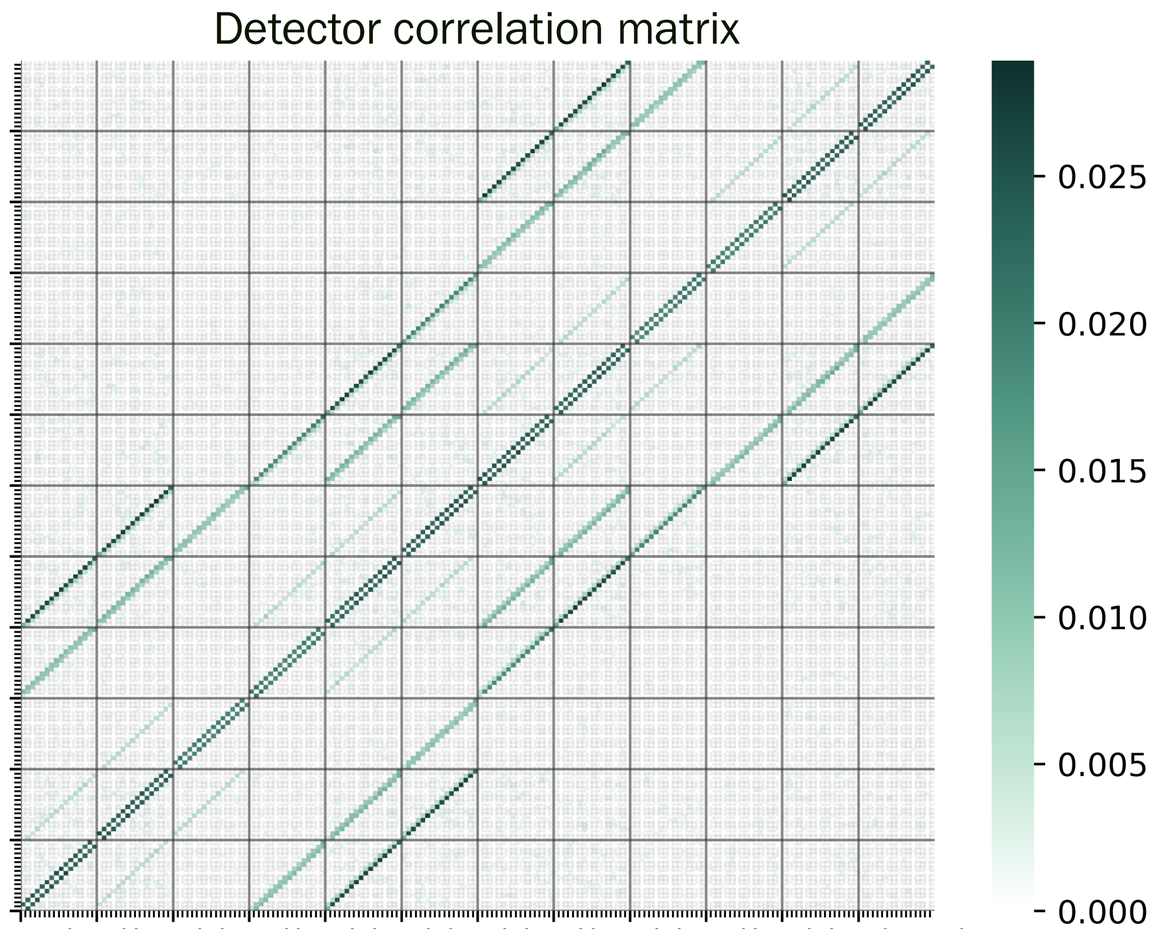


# Ways to improve performance?

## How to know what should be improved?

### One benchmark per improvable quantity?

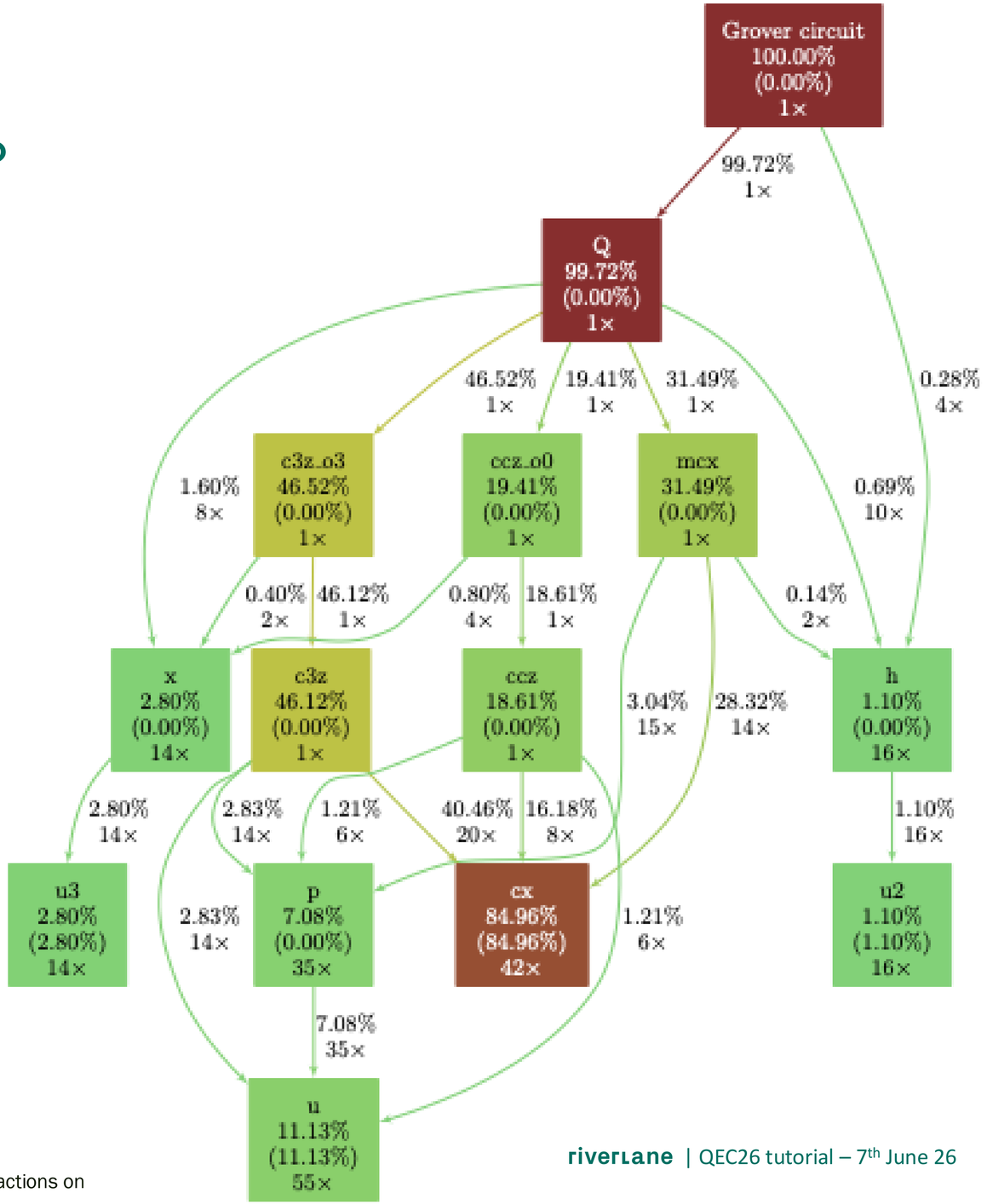


# Ways to improve performance?

## How to know what should be improved?

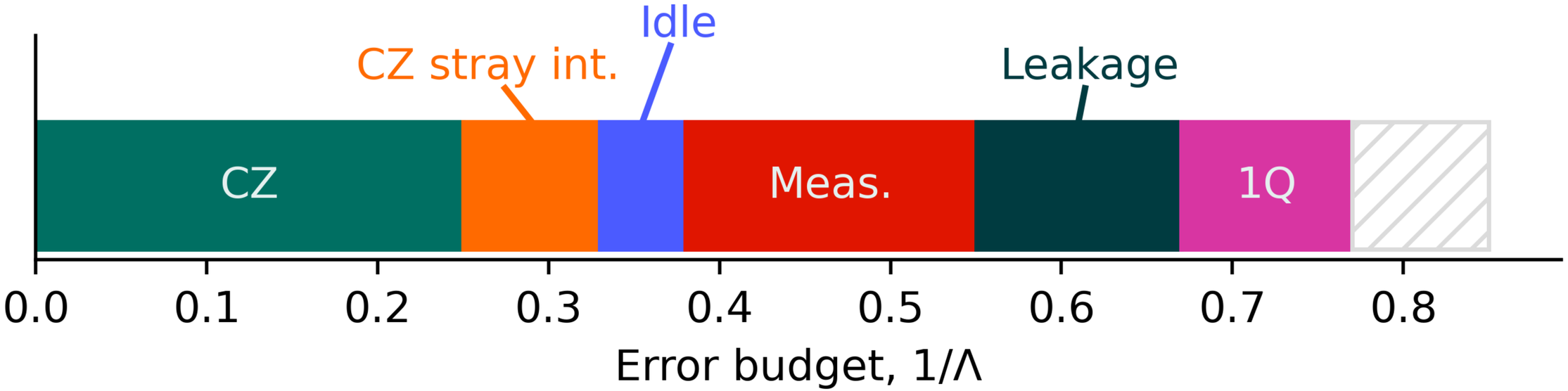
Can we have a generic, synthetic and useful picture for QEC?




Adrien Suau, Gabriel Staffelbach, and Aida Todri-Sanial. 2022. Qprof: A gprof-Inspired Quantum Profiler. ACM Transactions on Quantum Computing 4, 1, Article 4 (March 2023), 28 pages. https://doi.org/10.1145/3529398

# Ways to improve performance?

## How to know what should be improved?




Google Quantum AI. Exponential suppression of bit or phase errors with cyclic error correction. *Nature* **595**, 383–387 (2021). https://doi.org/10.1038/s41586-021-03588-y
Google Quantum AI and Collaborators. Quantum error correction below the surface code threshold. *Nature* **638**, 920–926 (2025). https://doi.org/10.1038/s41586-024-08449-y

# Deltakit

## A QEC toolkit for researchers and companies

000 What is needed to compute interesting things?

001 Program generation in Deltakit

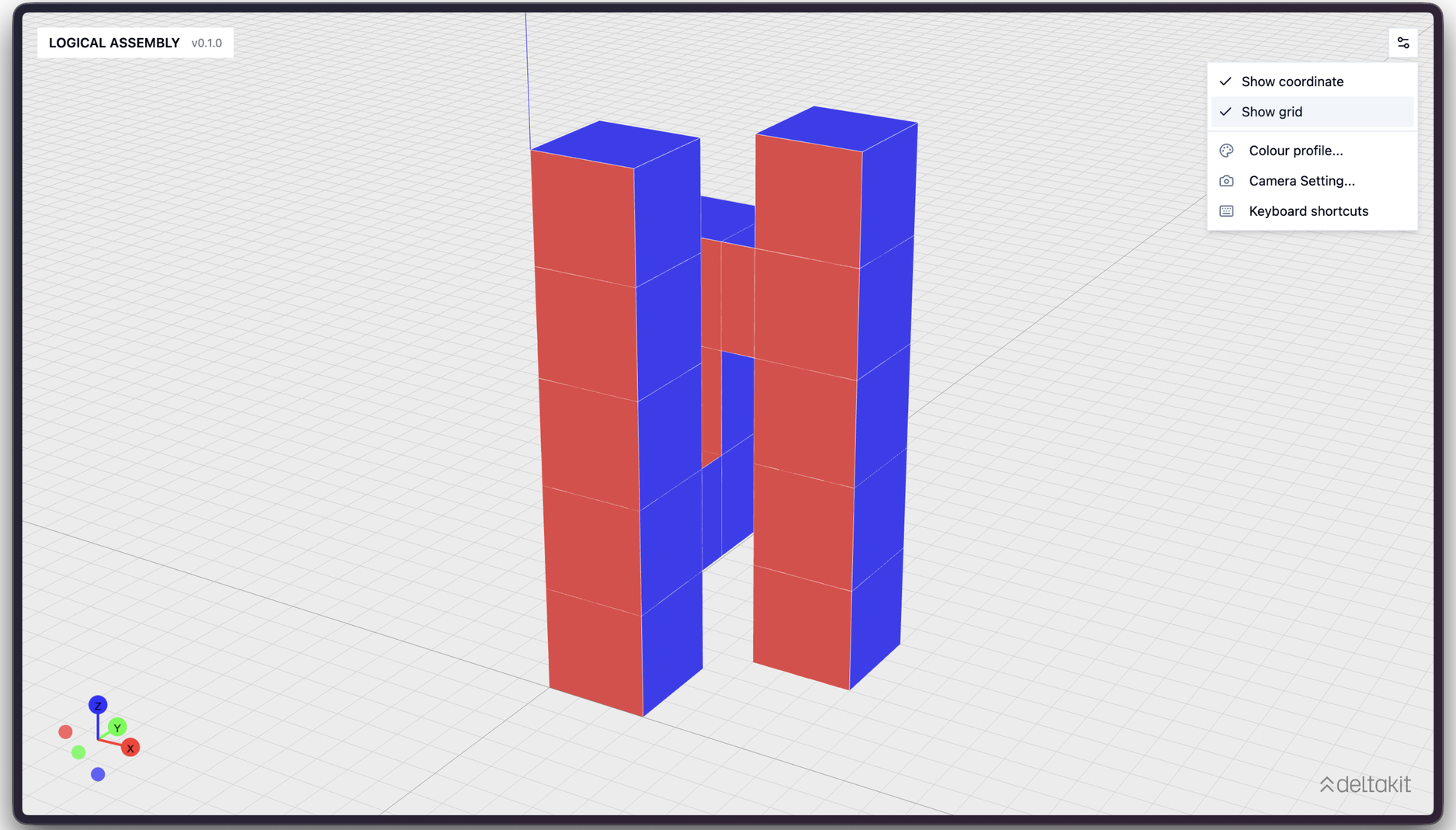
010 Deep-dive into a pipeline

011 Ways to improve your code/decoder/implementation/hardware

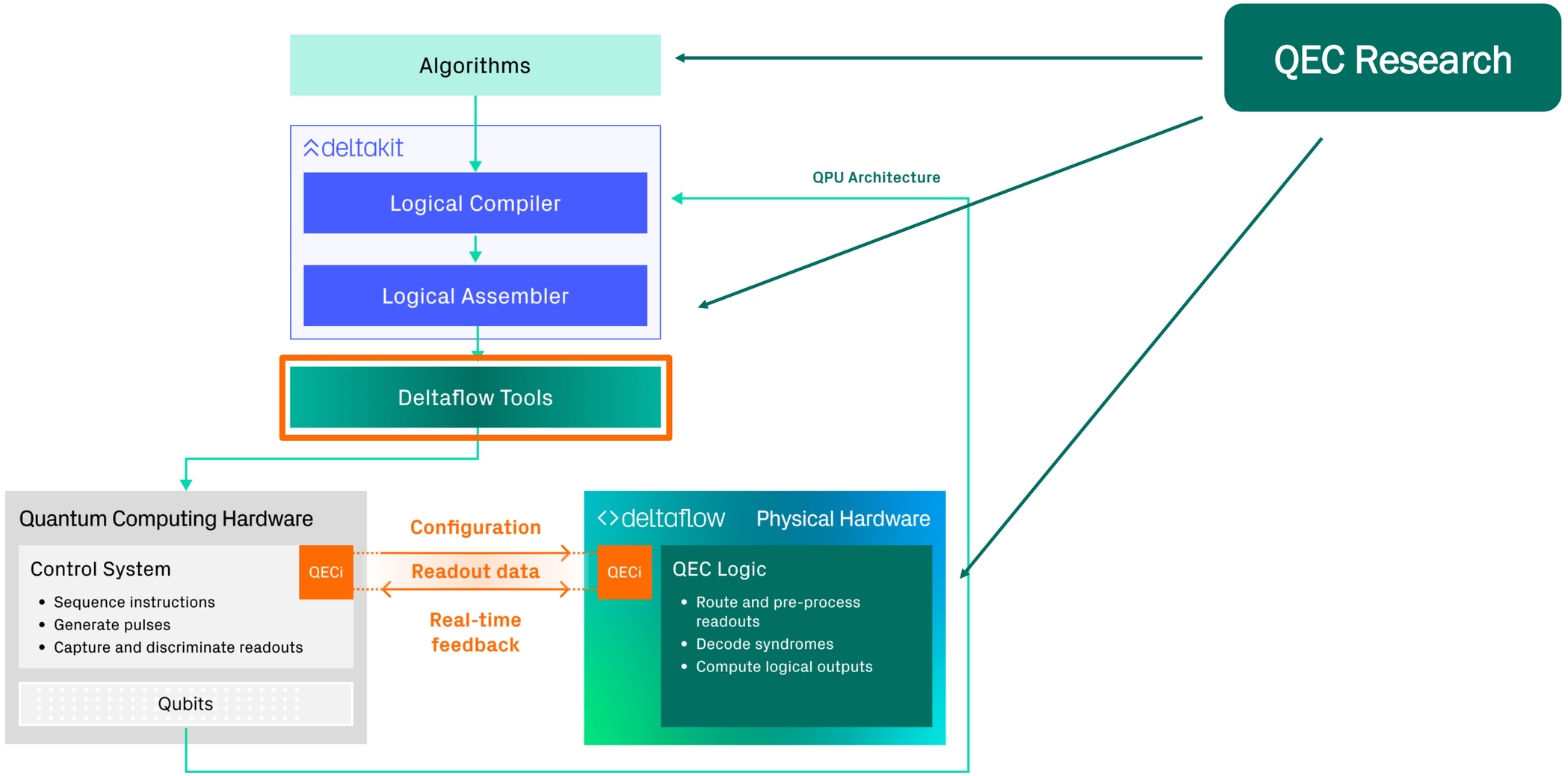
**100 Spoilers**

# Visualisation

## Spoiler on visualisation



# Our mission is to make quantum computers useful sooner.



# deltaflow

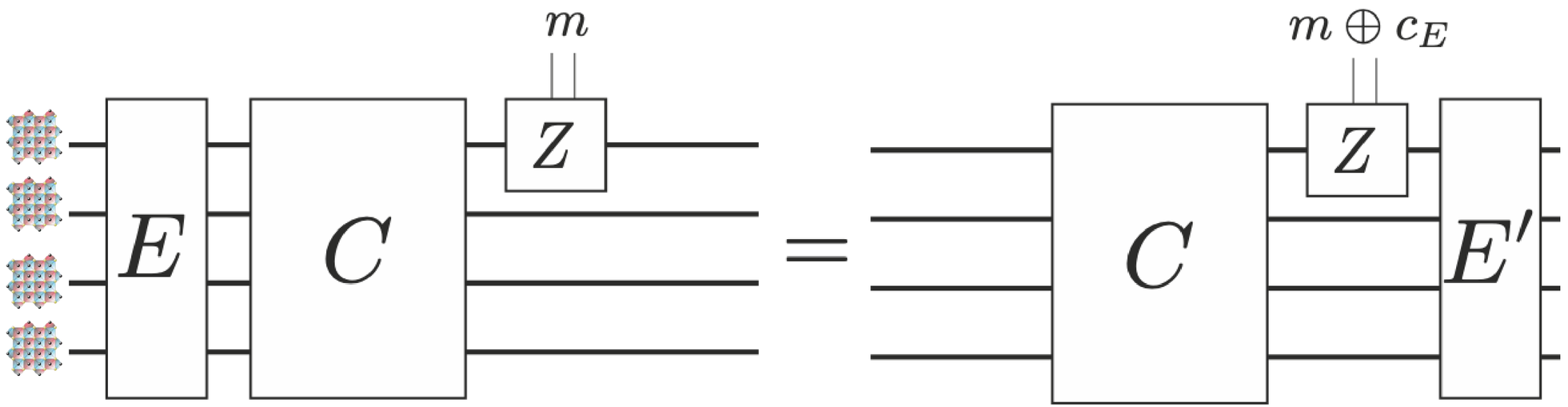
**Real-time decoder**
accelerating the world's quantum computers to utility scale

# deltakit

**The essential toolkit**
for learning, developing and adopting QEC

# Why do we need real-time decoding? – Clifford circuits

Fault-tolerant computing using Cliffords C (e.g. quantum memory with the surface code)

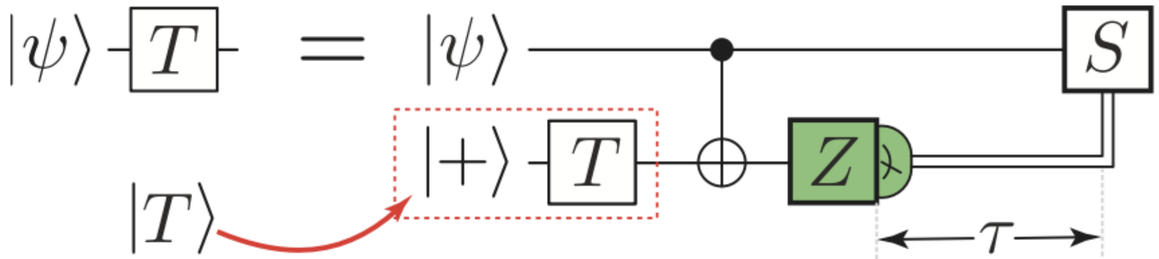


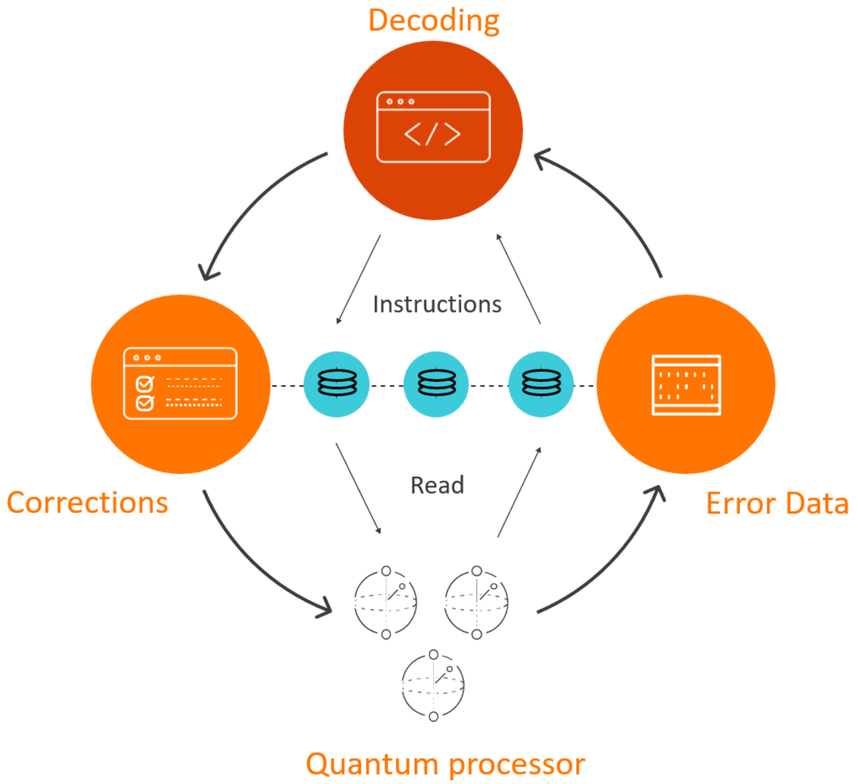

Pauli errors E in the past, are equivalent to Pauli errors E’ at later times.
Pauli measurements outcomes are just re-interpreted.

Decoding in post-processing is sufficient

# Why do we need real-time decoding? – non-Clifford circuits

- QEC codes alone cannot implement a universal and transversal gate set



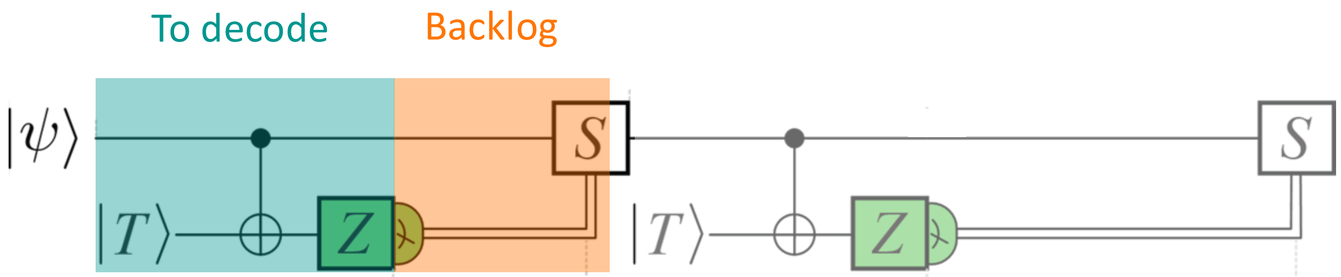




- Logical branching – logical operation conditional on a corrected observable
- Need live feedback ⇒ Real-time decoding necessary
- But how quick does a real-time need to be?

# High throughput – avoid the backlog problem

The backlog problem: The rate at which the decoder processes data (*throughput*) needs to be greater than the rate at which data is generated

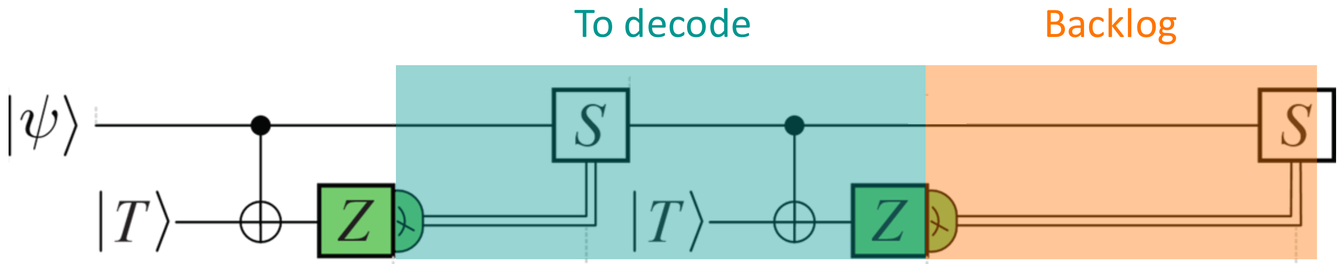
- Decoder slower than data received -> logical qubit idles -> more data accumulates to be decoded next round -> wait longer for decoding -> logical qubit idles even longer
- Each subsequent conditional operation will be exponentially slower



# High throughput – avoid the backlog problem

The backlog problem: The rate at which the decoder processes data (*throughput*) needs to be greater than the rate at which data is generated

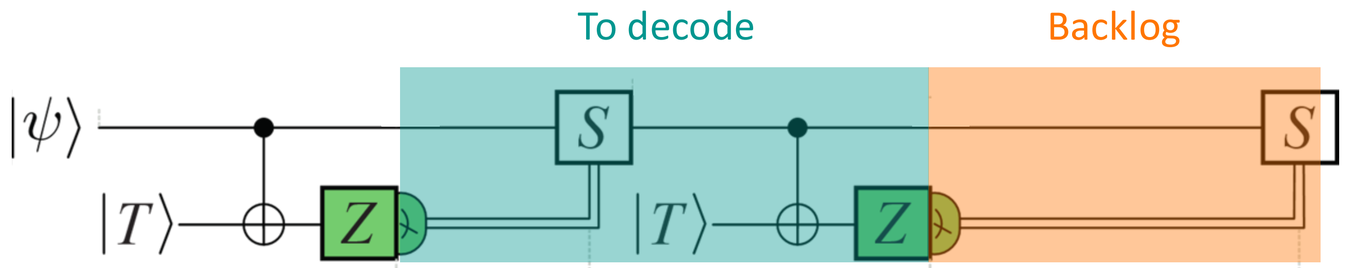
- Decoder slower than data received -> logical qubit idles -> more data accumulates to be decoded next round -> wait longer for decoding -> logical qubit idles even longer
- Each subsequent conditional operation will be **exponentially** slower



# High throughput – avoid the backlog problem

The backlog problem: The rate at which the decoder processes data (*throughput*) needs to be greater than the rate at which data is generated

- $1\mu s$/round superconducting QEC time
- Solution: fast decoder or parallelize decoding over many decoder threads




nature communications

Explore content | About the journal | Publish with us

nature > nature communications > articles > article

Article | Open access | Published: 03 November 2023

**Parallel window decoding enables scalable fault tolerant quantum computation**

Luka Skoric, Dan E. Browne, Kenton M. Barnes, Neil I. Gillespie & Earl T. Campbell

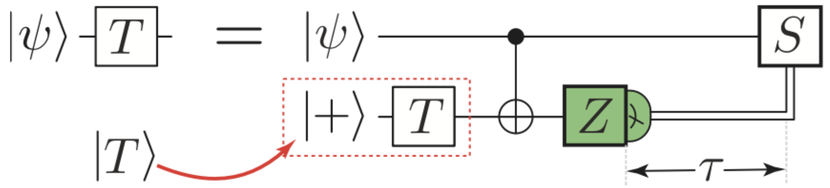
*Nature Communications* **14**, Article number: 7040 (2023) | Cite this article

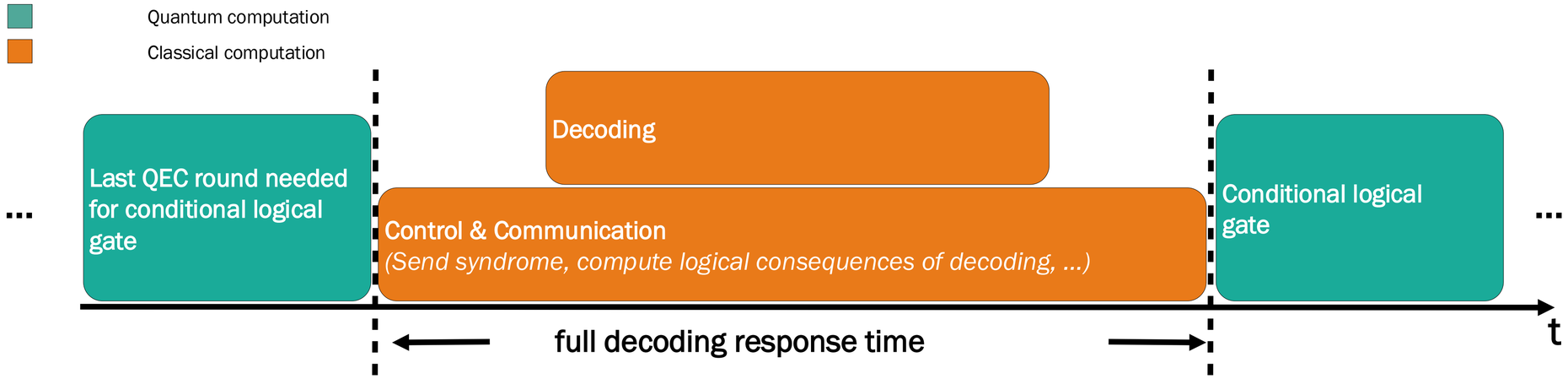
**14k** Accesses | **83** Citations | **53** Altmetric | Metrics

# Low-latency decoding – fast full decoding response time

**Full decoding response time:** The time between the final data extraction round and the application of a logical conditional gate

- Decoding time + communication and control latency times
- Can be a speed bottleneck for operations with logical branching
- Affects the *logical clock rate* – number of logical non-Clifford gates per unit time
- 10 µs full decoding response time assumed to factor 2048-bit RSA integers in 8h using 20 million noisy superconducting qubits




Gidney, C. & Ekera, M. *How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits.* Quantum **5**, 433 (2021).

# Low-latency decoding – fast full decoding response time

Full decoding response time: The time between the final data extraction round and the application of a logical conditional gate

Solution: real-time decoding and minimize control and communication latencies

nature communications

Explore content ⌄ About the journal ⌄ Publish with us ⌄

nature › nature communications › articles › article

Article | Open access | Published: 17 December 2025

## Local clustering decoder as a fast and adaptive hardware decoder for the surface code

Abbas B. Ziad ✉, Ankit Zalawadiya, Canberk Topal, Joan Camps ✉, György P. Gehér, Matthew P. Stafford & Mark L. Turner

*Nature Communications* **16**, Article number: 11048 (2025) | Cite this article

**5383** Accesses | **7** Citations | **16** Altmetric | Metrics

# Real-time decoding

Three aspects need to be satisfied to show *true* real-time decoding

**High-throughput streaming decoding:**
Throughput (how fast decoder processes a single round of QEC data) needs to be faster than the rate at which data to be decoded is generated to **avoid exponential slow-down of computation time**

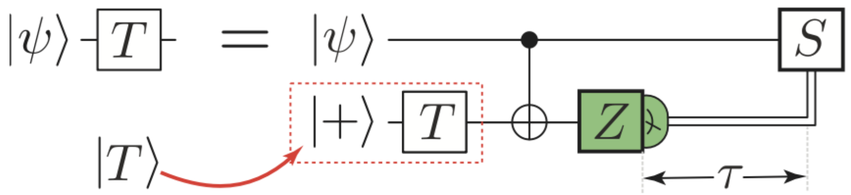
Barbara M. Terhal, *Quantum error correction for quantum memories*. Rev Mod Phys. **87** 307 (2015)

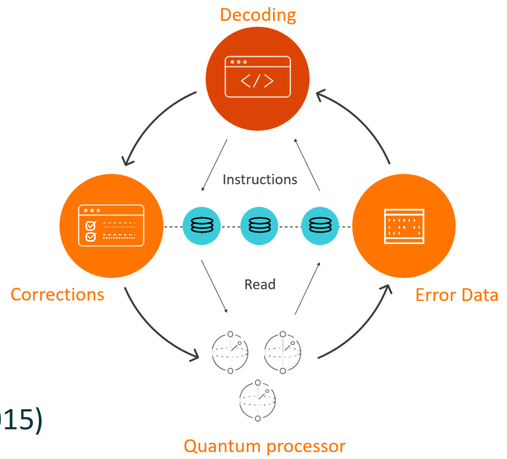
# Real-time decoding

Three aspects need to be satisfied to show *true* real-time decoding

**High-throughput streaming decoding:**
Throughput (how fast decoder processes a single round of QEC data) needs to be faster than the rate at which data to be decoded is generated to **avoid exponential slow-down of computation time**

**Low-latency decoding:**
The total decode time needs to be low-latency to not add significantly to the reaction time and slow down **logical clock rate**



# Real-time decoding

Three aspects need to be satisfied to show *true* real-time decoding

**High-throughput streaming decoding:**
Throughput (how fast decoder processes a single round of QEC data) needs to be faster than the rate at which data to be decoded is generated to **avoid exponential slow-down of computation time**

**Low-latency decoding:**
The total decode time needs to be low-latency to not add significantly to the reaction time and slow down **logical clock rate**

**Fast feedback:**
Minimize additional control and communication latencies towards your reaction time and **logical clock rate**

Barbara M. Terhal, *Quantum error correction for quantum memories*. Rev Mod Phys. **87** 307 (2015)

# Deltaflow 2

## Goal: demonstrate all three aspects of real-time decoding

Deltaflow 2 performs real-time error correction on up to 250 physical qubits with Riverlane's high-accuracy proprietary Local Clustering Decoder (LCD) on FPGA.

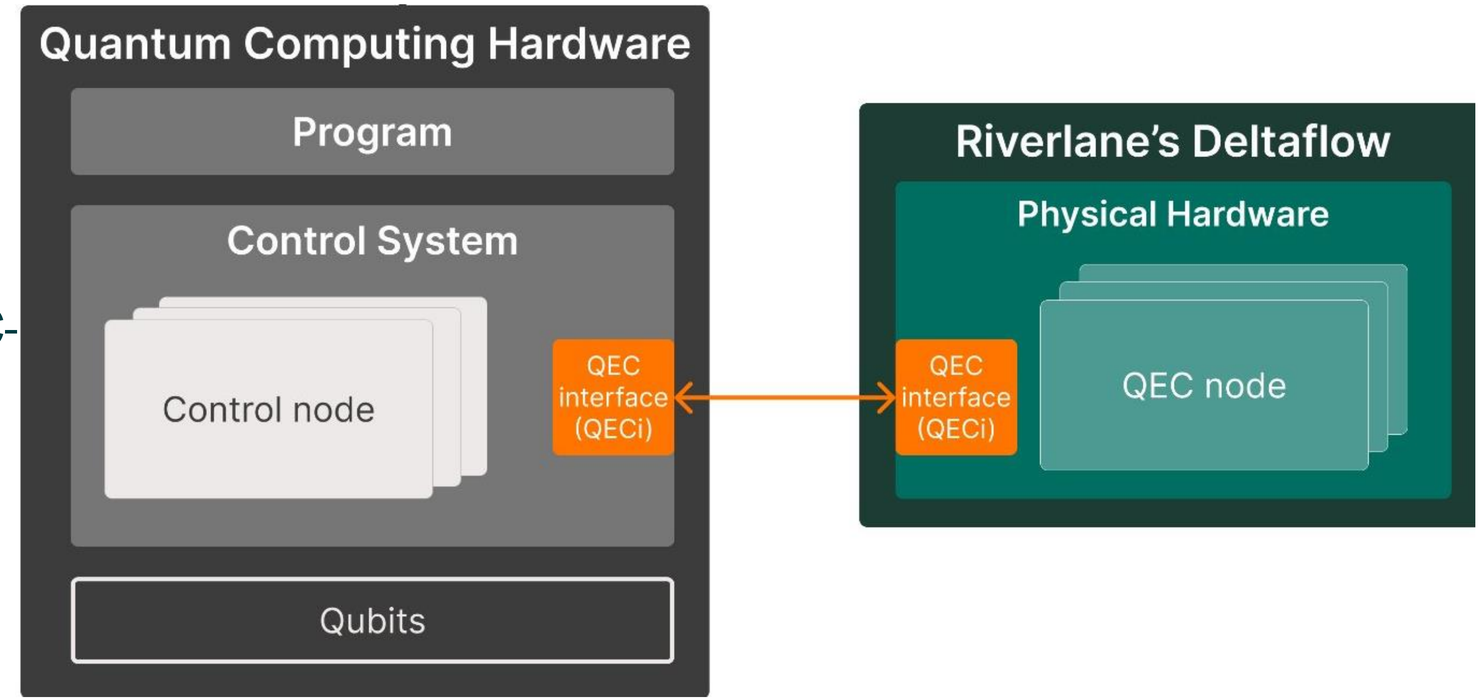
| | |
|---|---|
| **High Accuracy** | 10,000 QuOps |
| **Low-latency response time** | <14us |
| **High-throughput** | 1us/round |
| **Adaptive** | Leakage aware-decoding |
| **QEC Capabilities** | Streaming quantum memory/stability |
| **Decoding capabilities** | Surface code, repetition code |

All metrics measured for a d=9 rotated planar code quantum memory experiment with p=10^-3 and independent windowing

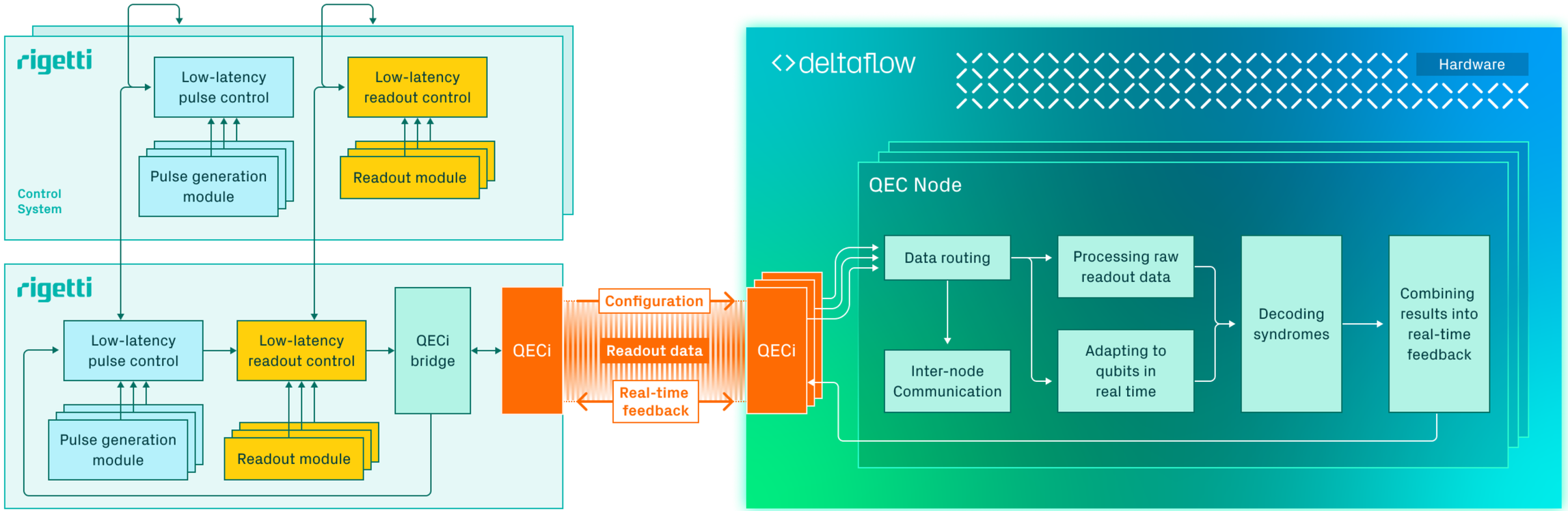
# QEC interface

**QEC interface (QECi):**

- Defines the interface between the QEC stack and the control system
- Describes data format, runtime states and parameters for communicating QEC-specific messages
- Reduces integration effort and minimizes duplication of effort



# Integrate Deltaflow 2 with Rigetti's Ankaa-3-36Q



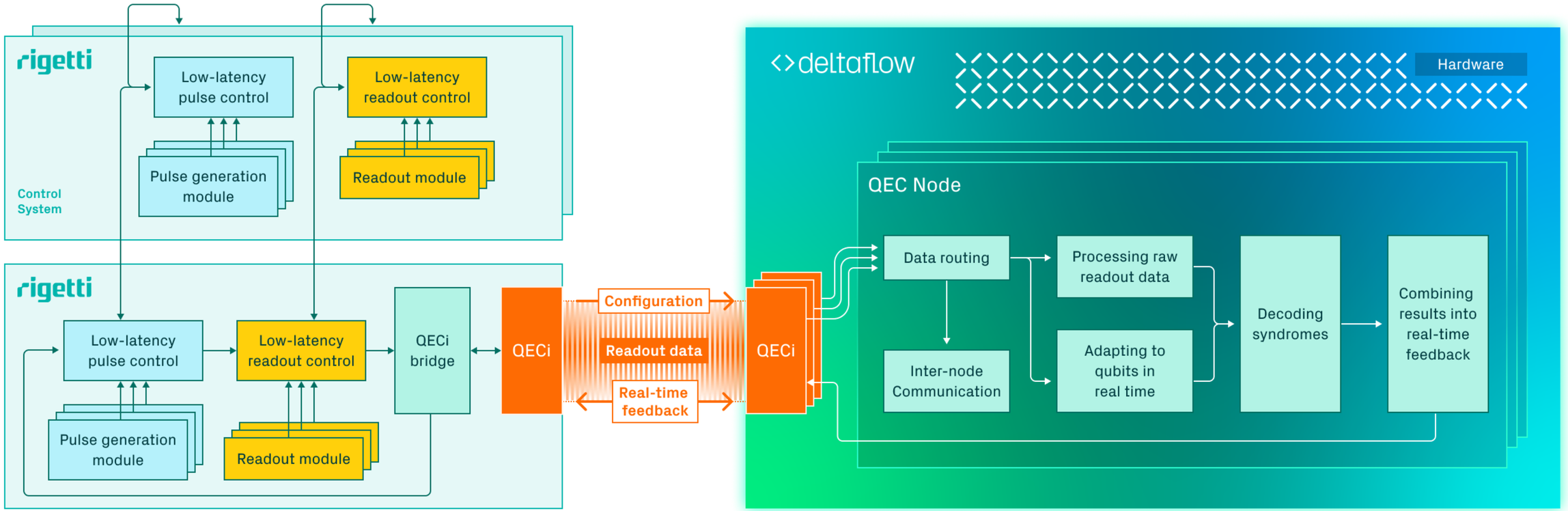

riverlane rigetti

# Integrate Deltaflow with Rigetti's Ankaa-3-36Q

Streaming decoding
(high throughput / no backlog)

Low decoding latency

Fast feedback
(physical circuit branches conditionally on)




riverlane rigetti

# A Decoding Algorithm ≠ A QEC System

### What traditional QEC research delivers

- Batch syndromes
- Fixed Decoding Graph
- Unlimited compute time
- Single/few code instance
- Output: a correction

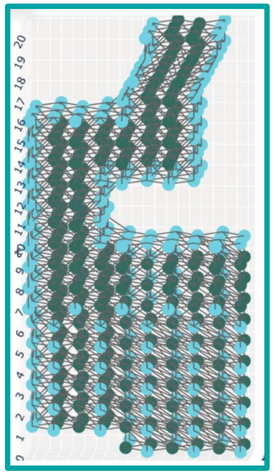
### What real-time execution demands

- Streaming continuous measurements
- Graph changes during computation
- Hard deadlines with bounded compute and memory
- Multiple codes + distances
- Feedback drives computation

**The space between these columns is where systems research lives**

# Frontier of research

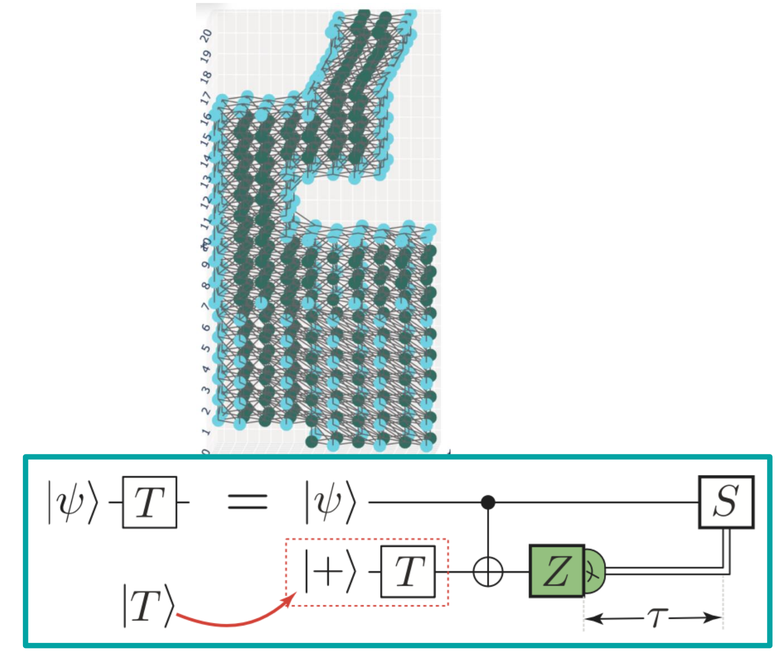
**1. Dynamic Decoding Graphs**



# Frontier of research

**1. Dynamic Decoding Graphs**

**2. Control Flow Effect On Streaming**

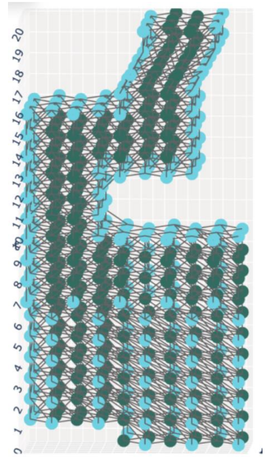


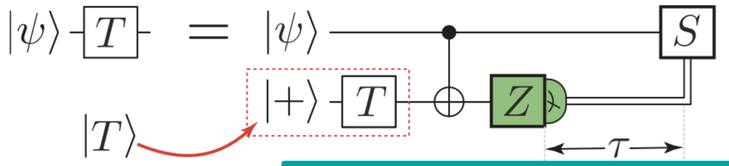
# Frontier of research

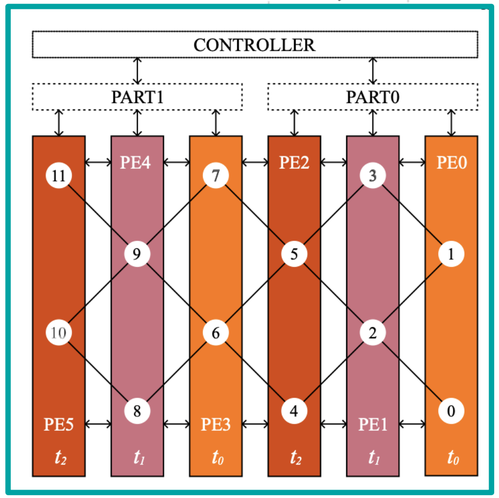
**1. Dynamic Decoding Graphs**

**2. Control Flow Effect On Streaming**

**3. Hardware Graph Embedding**





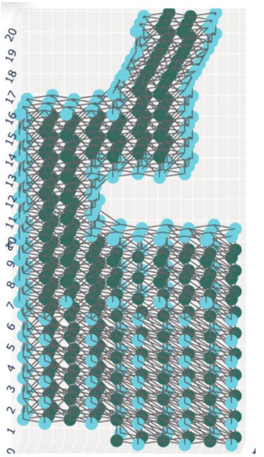
# Frontier of research

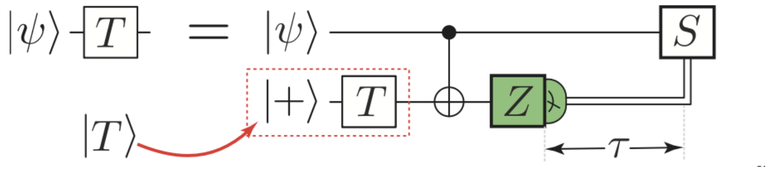
**1. Dynamic Decoding Graphs**

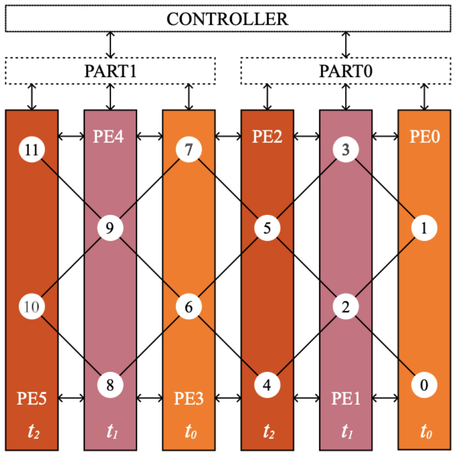
**2. Control Flow Effect On Streaming**

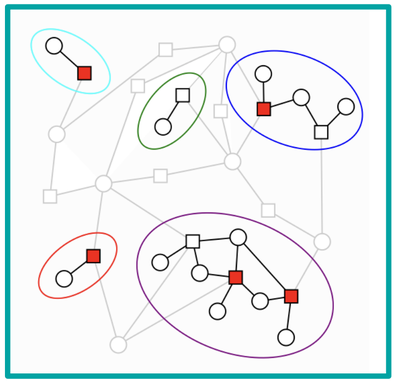
**3. Hardware Graph Embedding**

**4. Real time QLDPC decoding**





# …and that's just the start

**Multi-decoder orchestration**

N decoders coordinate during multi qubit operation

**Soft information**

Beliefs between adjacent windows

**Online weight adaptation**

Live calibration data

**Compile-time timing verification**

WCET analogue for QEC

**Latency–accuracy tradeoff optimization**

Optimal window params under deadline

**Observable tracking**

DEM observable propagation across windows

**QLDPC real-time decoding**

Non-local codes on local hardware

**Adaptive window sizing**

Reactive scheduling under load

**Decoder verification**

Can't exhaustively test at scale for small error rates

**Leakage in streaming**

Heralding corrupts future windows

**Streaming resource estimation**

Overhead depends on decode system

**Erasure errors**

Erasure of qubits during computation

**Decoder-aware compilation**

Schedule to maximize decode utilization

**These are not just engineering problems**

they need new theory, algorithms, and hardware co-design

**Solving them means directly enabling useful quantum computers.**

# Our mission is to make quantum computers useful sooner.

| 2025 | 2026–2027 | 2028–2032 | | | 2033+ |
|---|---|---|---|---|---|
| First full Quantum Error Correction system | Enables first fault-tolerant quantum computers | Enables first fault-tolerant circuits | | | Enables first full fault-tolerant quantum computers at utility scale |
| **Delivered product** | **In development** | **Future technology** | | | |
| **Quantum memory** | **Quantum gates** | **Quantum circuits** | **Beyond classical** | **First algorithms** | **Beginning of utility scale** |
| PRODUCT | | | | | |
| **Deltaflow 2**<br>• Continuous real-time decoding for high-fidelity quantum memory<br>• Leakage aware decoding<br>• First universal interfacing | **Deltaflow 3**<br>• Perpetual Clifford logic<br>• Fast logic by lattice surgery<br>• Higher error suppression rates | **Deltaflow 4**<br>• Perpetual universal logic<br>• Fault-tolerant circuits with lattice surgery<br>• Improved decoding accuracy via real-time adaptations<br>• Real-time logical branching for non-Clifford operations | PRODUCT INNOVATION | **MegaQuOp**<br>• Fault-tolerant execution of circuits beyond classical capabilities<br>• Low logical error rates through optimised decoders<br>• Expanded support for multiple qubit modalities and logical architectures | **GigaQuOp**<br>• Faster logical clock rates for large-scale fault-tolerant circuits<br>• Ultra-low logical error rates through system-wide error suppression<br>• Support for all leading qubit modalities, fault-tolerant codes and logical architectures | **TeraQuOp**<br>• Fault-tolerant quantum computing that surpasses classical computing capabilities across multiple domains |
| INNOVATION | | | | | | |
| • Transversal logic for topological codes in neutral-atom and trapped-ion qubits | • High accuracy decoding for neutral-atom and trapped-ion qubit systems<br>• Efficient MegaQuOp Applications | • Logical compilation for qLDPC codes<br>• Flexible real-time decoding for neutral-atom and trapped-ion qubits | | | | |

## We are hiring

- QEC Researchers
- QEC Application Scientists
- Principal Investigator, QEC
- Quantum Scientists
- Quantum Engineers

Roles in Cambridge UK, and Boston, US

View our latest vacancies:
www.riverlane.com/jobs