Quantum AI

Quantum Era of Ferment:
From Counting Qubits to Assessing Capabilities

Adam Zalcman

2026年3月7日 (六)

Real World Post-Quantum Cryptography Workshop 2026

台北

Google

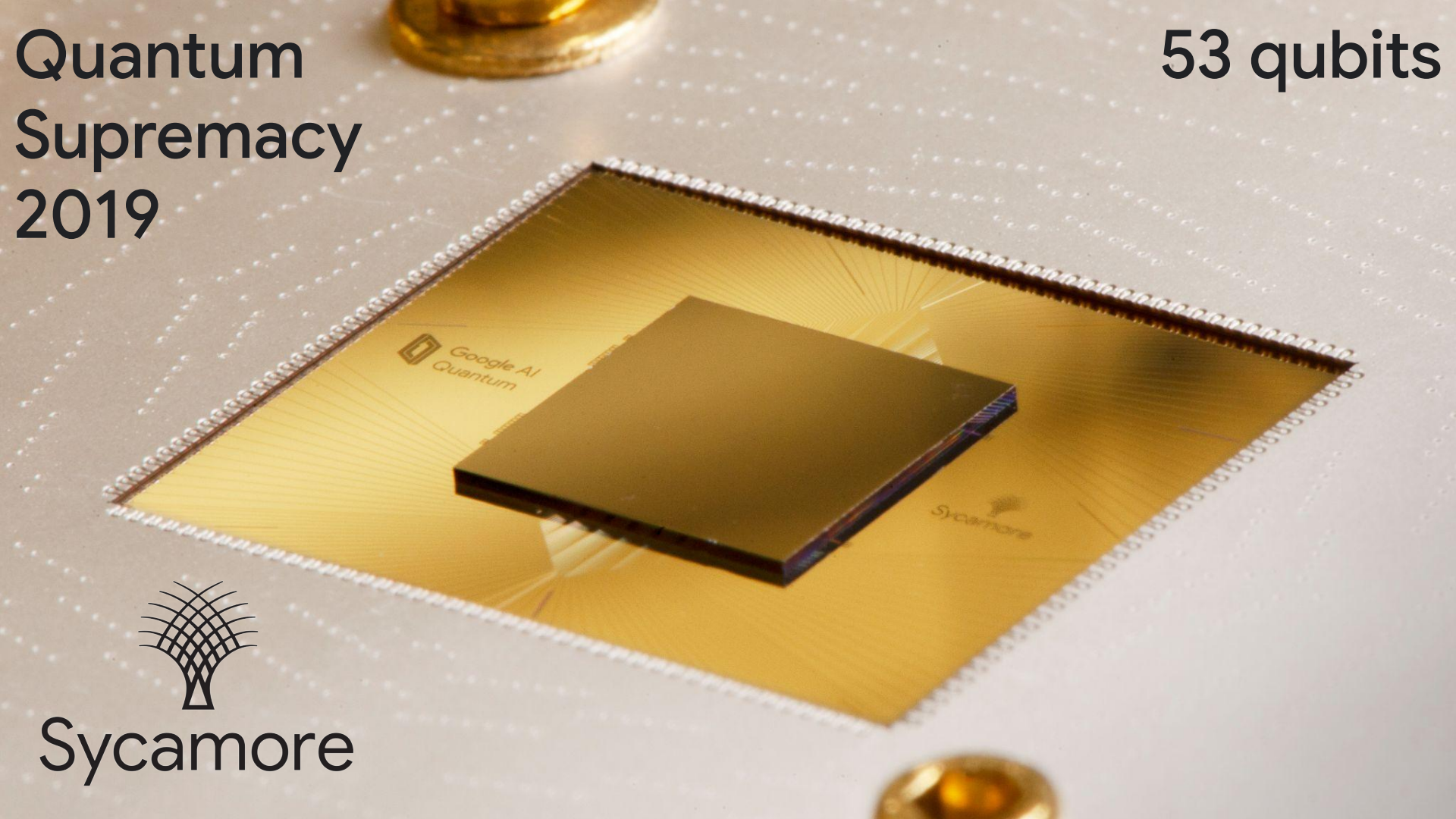
Quantum AI

#7210
djb, Tanja Lange
PQCHacks

PQCHacks, 32nd Chaos Communication Congress, Hamburg, **2015**

Quantum Supremacy 2019

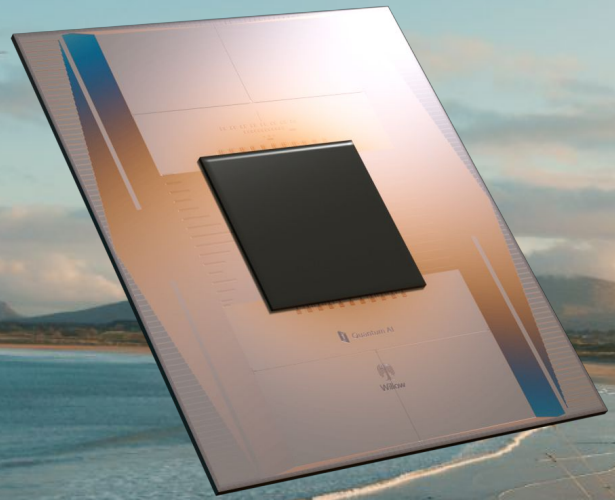
53 qubits



Sycamore

Quantum Error Correction 2024

105 qubits



Willow

Does it take five years to double the number of qubits?

Does it take five years to double the number of qubits?

No. Effort does not go into doubling the number of qubits. It goes into implementing capabilities.

Roadmap: Understanding Quantum Progress

How do we make progress?

- Assessing Capabilities vs. Counting Qubits
- Across Platforms: Anderson-Tushman Model
- Compiling: The finish line gets closer

What happens when we get there?

- Decoded Quantum Interferometry
- Quantum Algorithm for Planted Inference

Roadmap: Understanding Quantum Progress

How do we make progress?

- **Assessing Capabilities vs. Counting Qubits**
- Across Platforms: Anderson-Tushman Model
- Compiling: The finish line gets closer

What happens when we get there?

- Decoded Quantum Interferometry
- Quantum Algorithm for Planted Inference

Capabilities introduce performance discontinuities

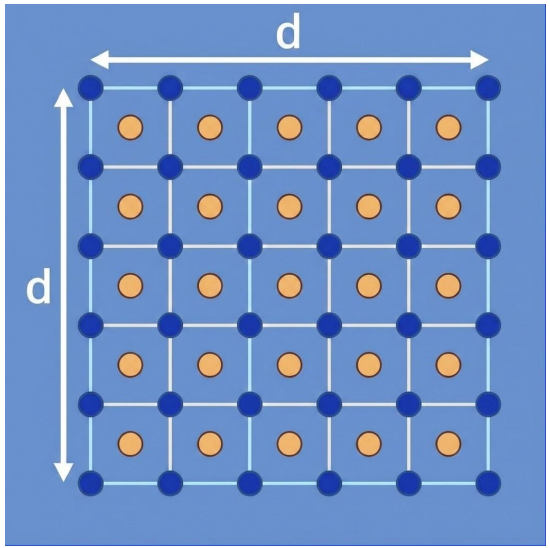
Example 1: Quantum Error Correction

Before: Quantum information in a superconducting circuit survives for ~100µs

Example 1: Quantum Error Correction

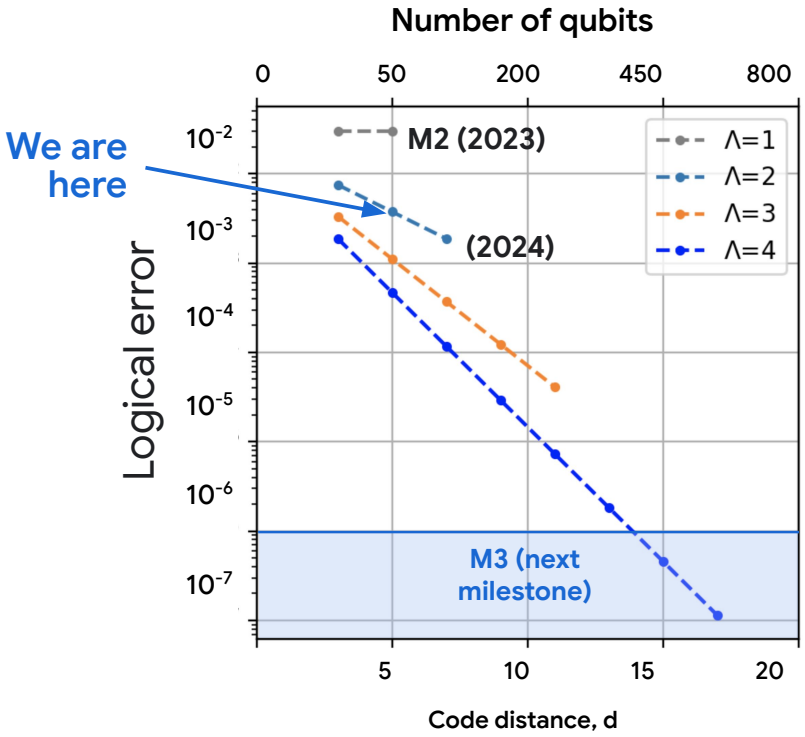
$$\varepsilon_d \propto \left(\frac{p}{p_{\text{thr}}}\right)^{(d+1)/2}$$

ε_d is the Logical Error Rate
p is the Physical Error Rate
p_{thr} is the Threshold Error Rate
d is the Code Distance



Example 1: Quantum Error Correction

$$\varepsilon_d \propto \left(\frac{p}{p_{\text{thr}}}\right)^{(d+1)/2}$$

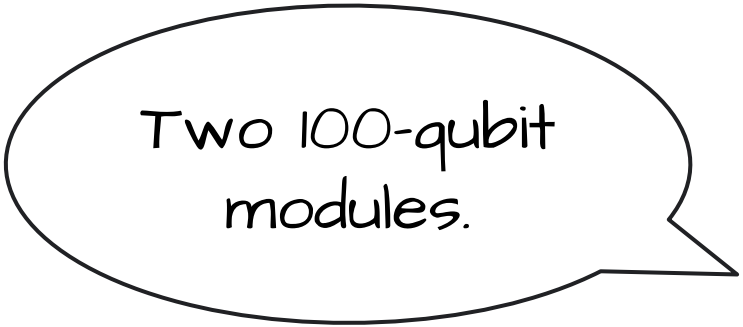


Example 1: Quantum Error Correction

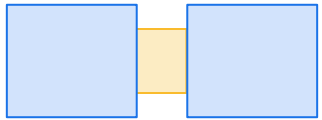
Before: Quantum information in a superconducting circuit survives for ~100μs

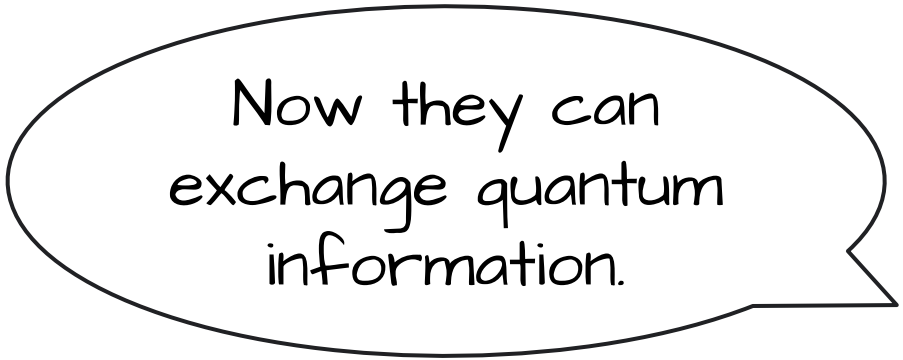
After: Quantum information in a superconducting circuit survives for essentially however long you wish

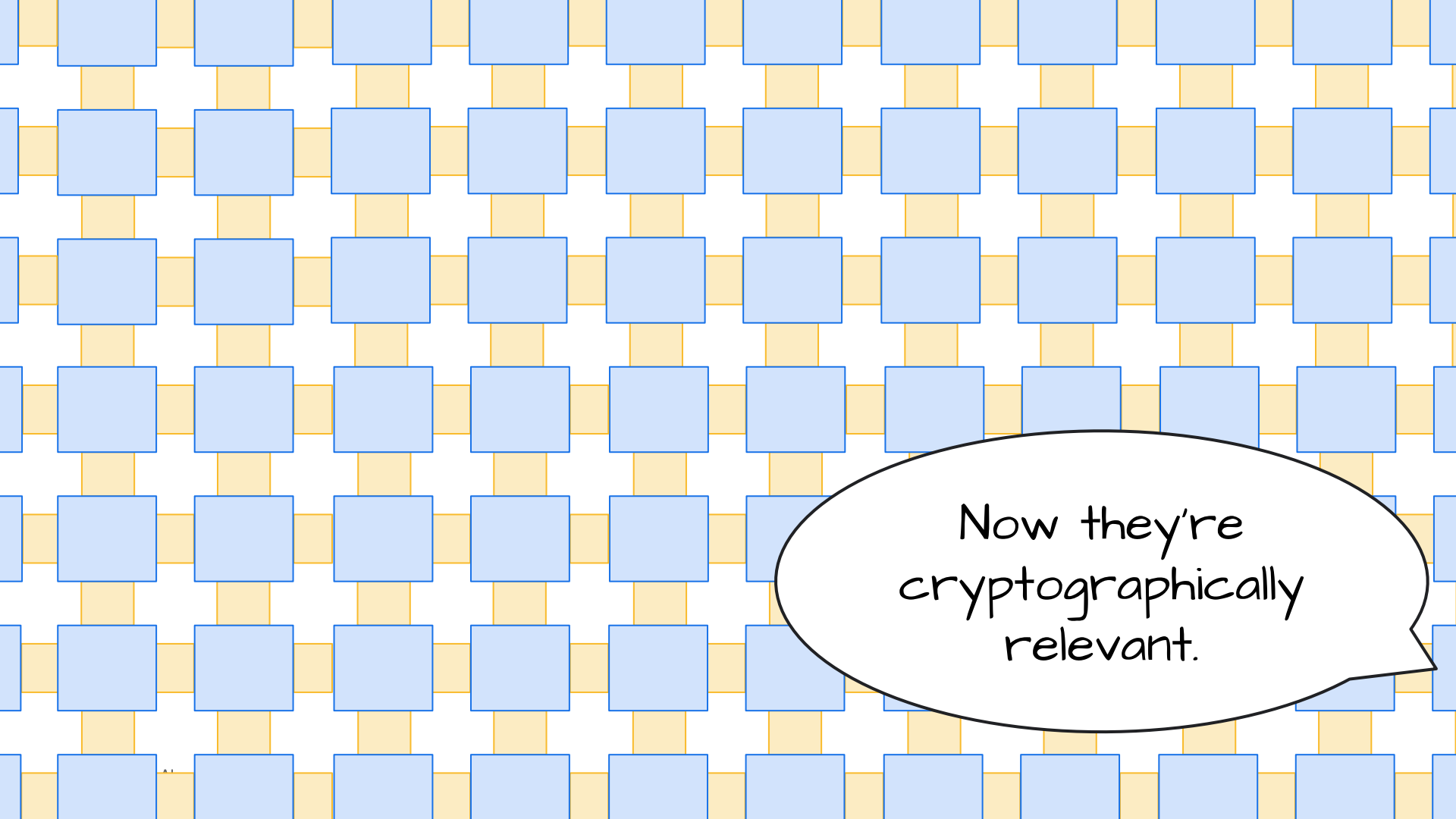
Example 2: Coherent Connections Between Processor Modules



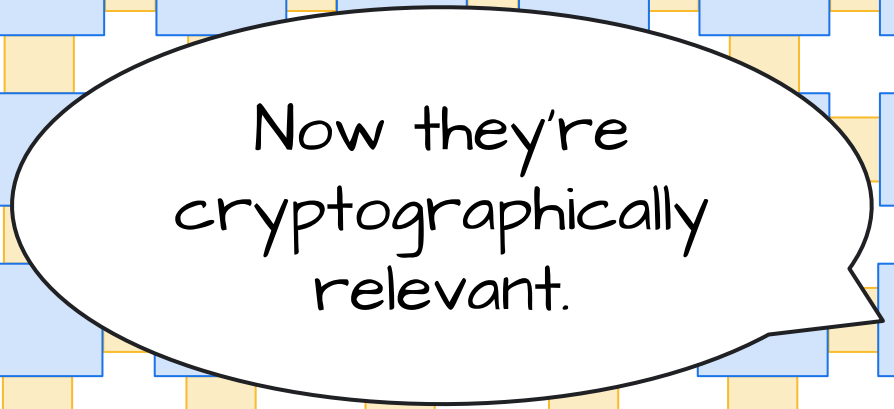
Example 2: Coherent Connections Between Processor Modules

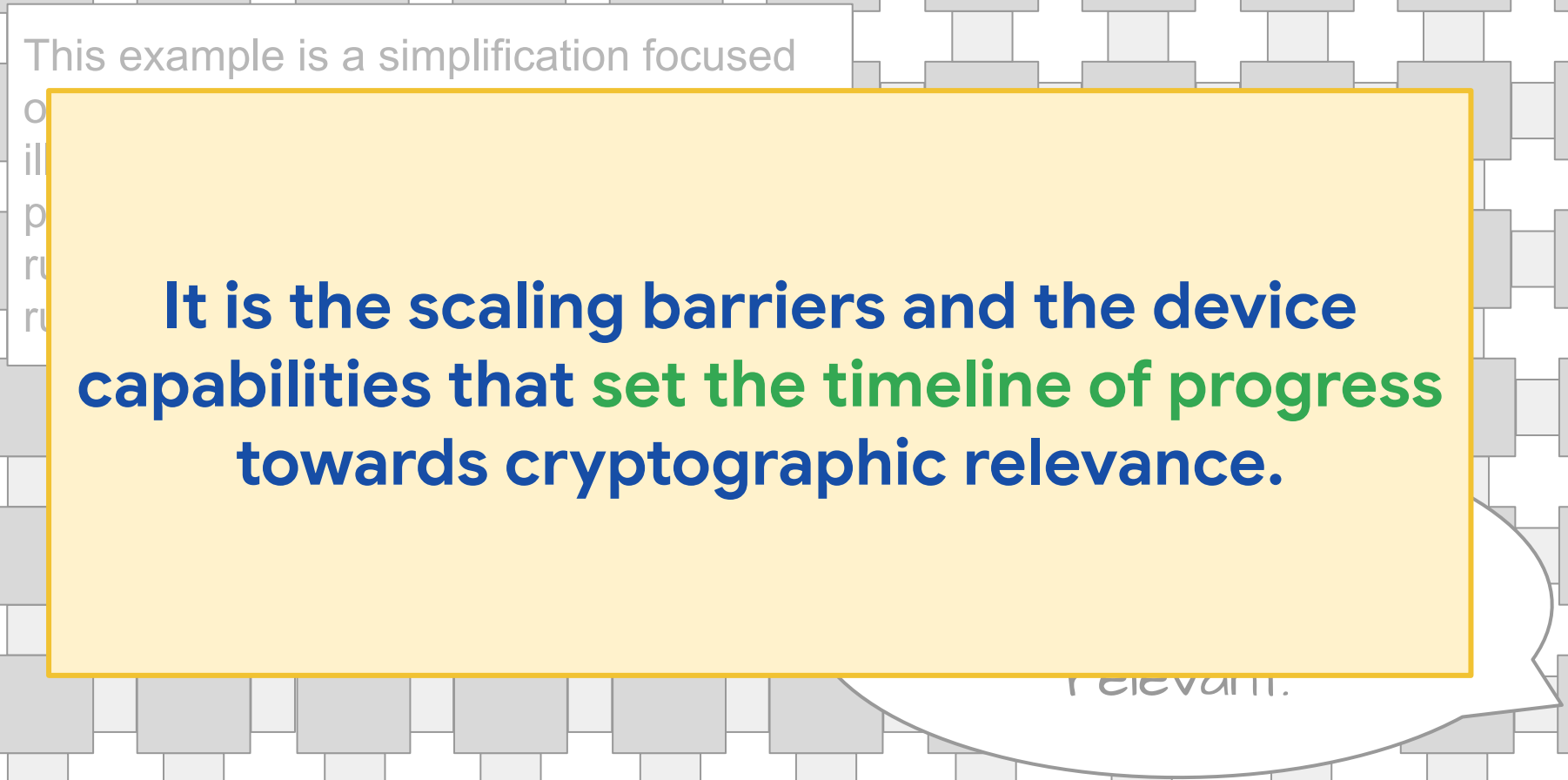





Now they're cryptographically relevant.

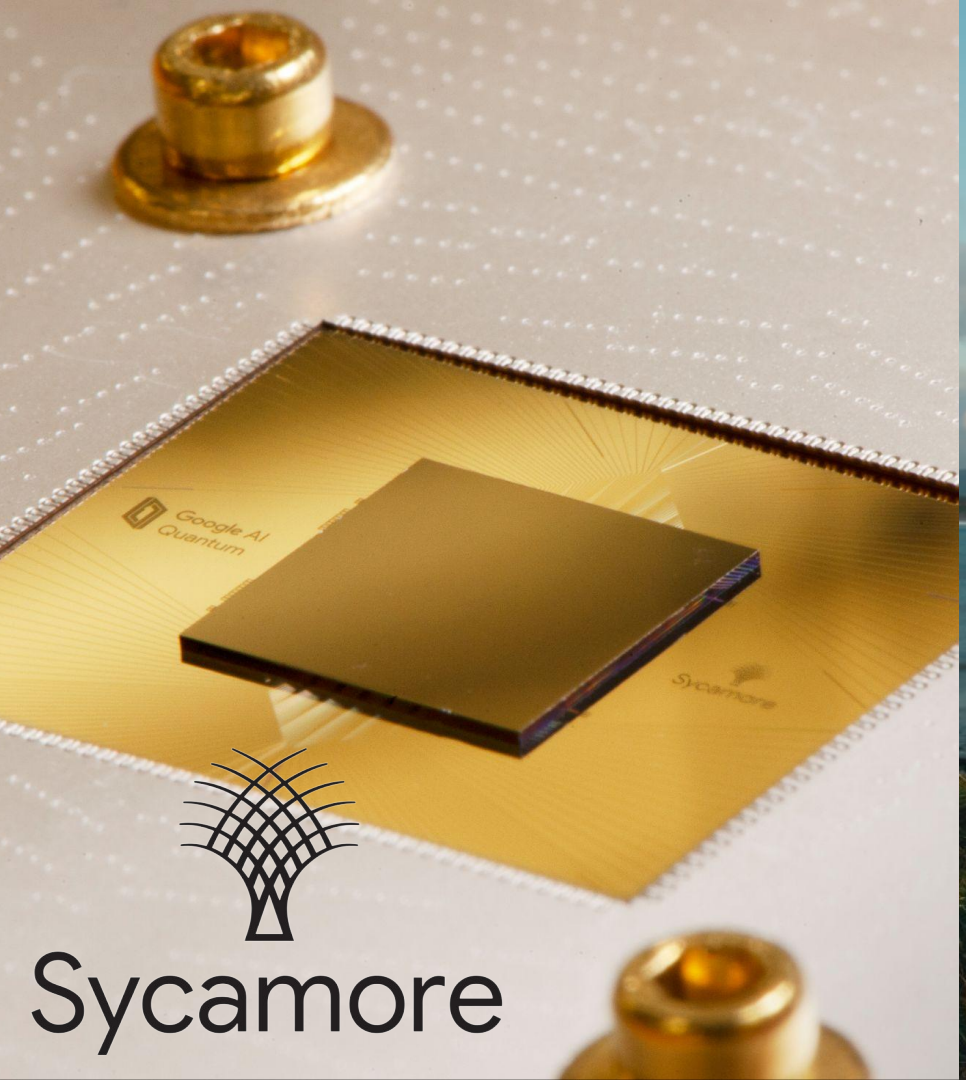
This example is a simplification focused on a single capability and is meant to illustrate how a new capability introduces performance singularity. In practice, runaway performance growth eventually runs into another scaling barrier.




This example is a simplification focused
It is the scaling barriers and the device capabilities that set the timeline of progress towards cryptographic relevance.

A demonstration of Shor's algorithm
on a e.g. 64-bit integer
is not a wake-up call to deploy PQC.

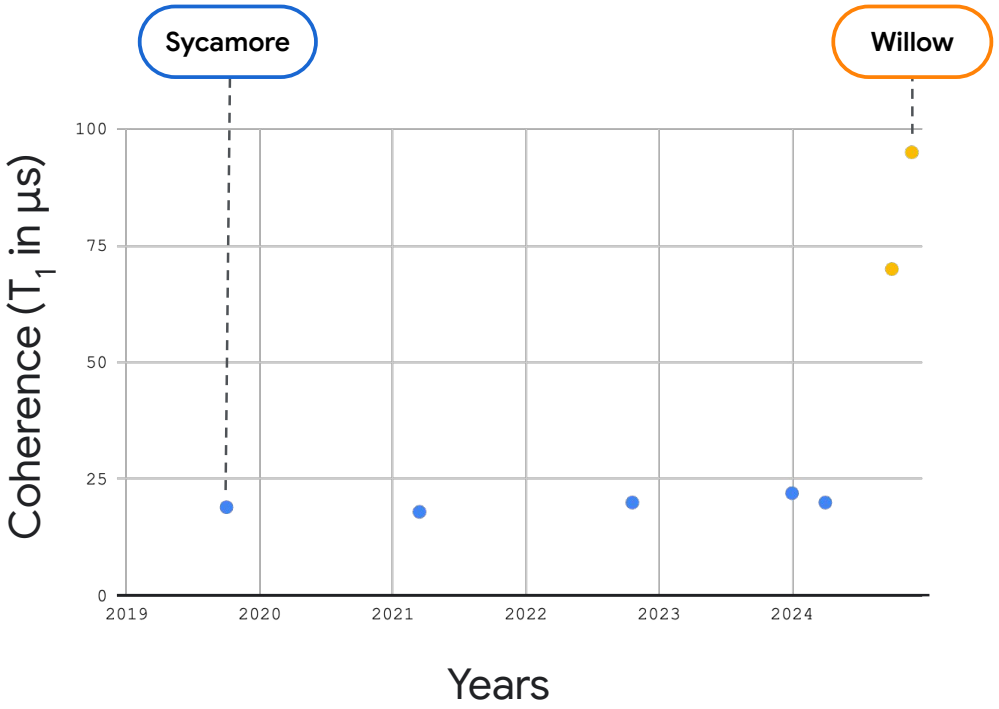
It is a potential signal that
PQC deployment has already failed.



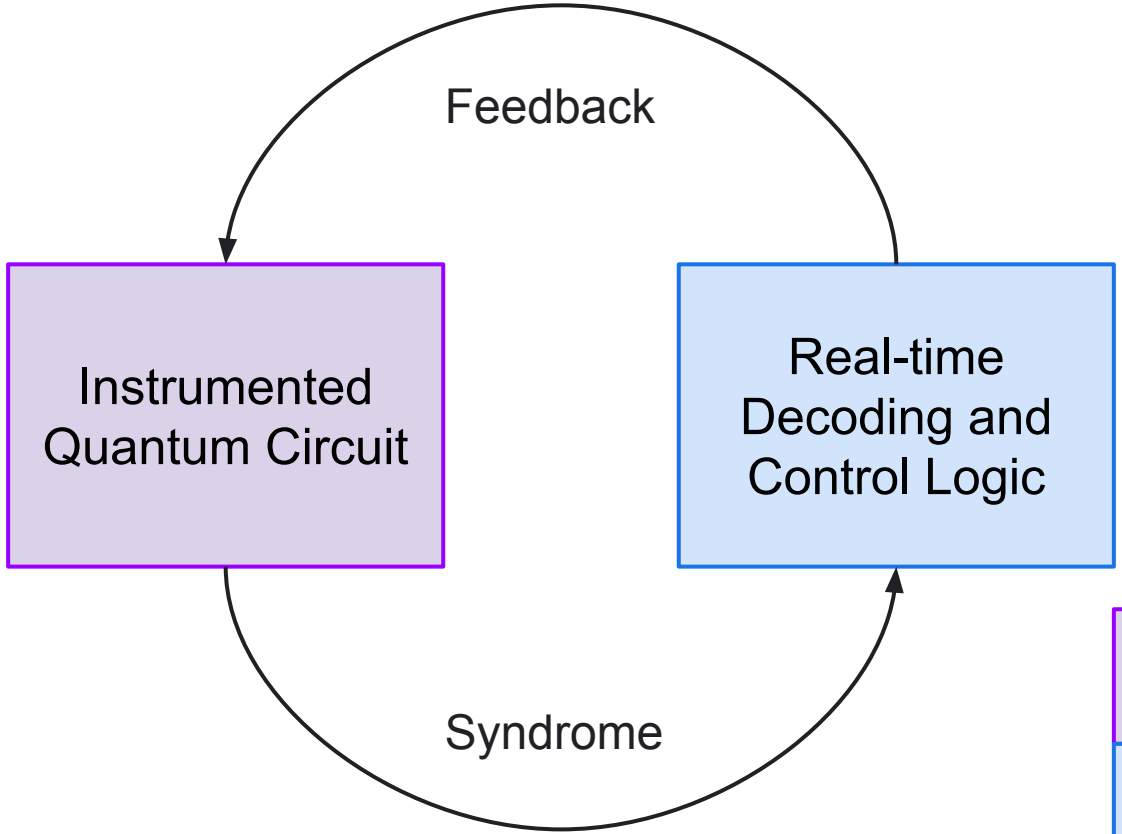
Sycamore

Willow

Sycamore vs Willow: Coherence Time



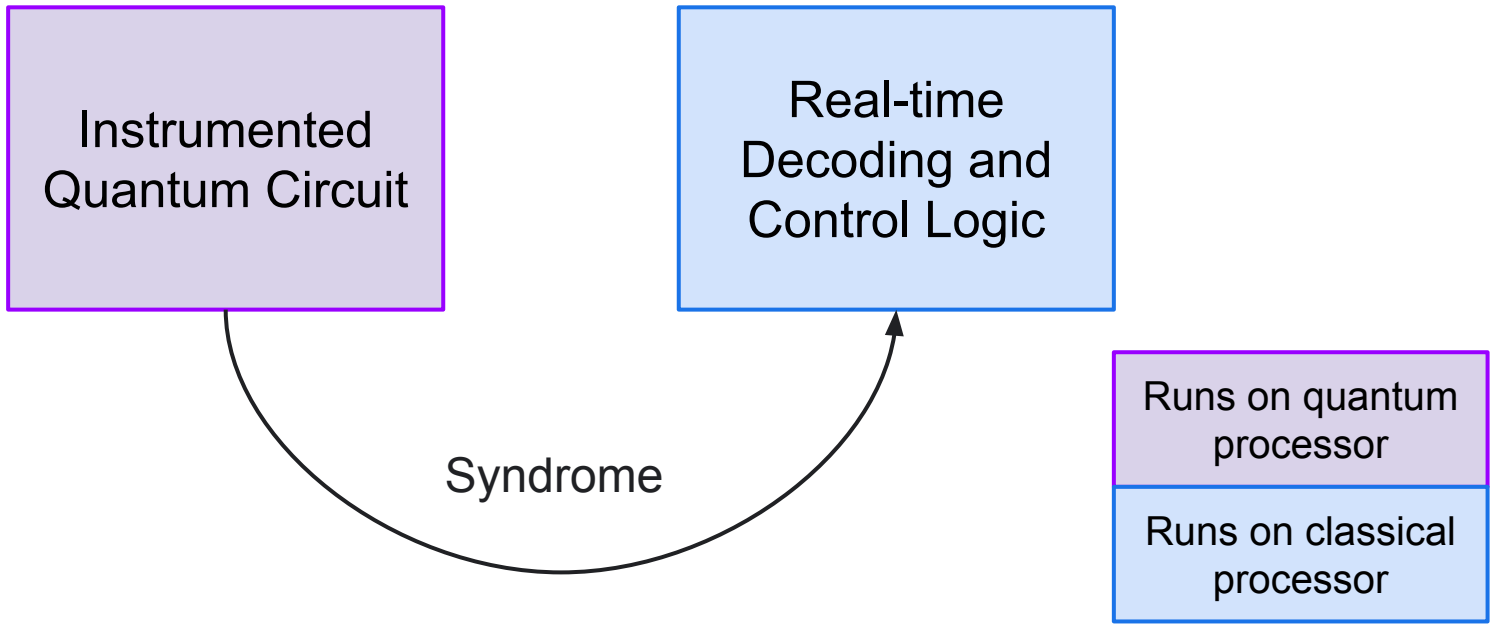
Sycamore vs Willow: Fault-Tolerance Control Loop



Runs on quantum processor

Runs on classical processor

Willow: Supported Elements of the Fault-Tolerance Control Loop



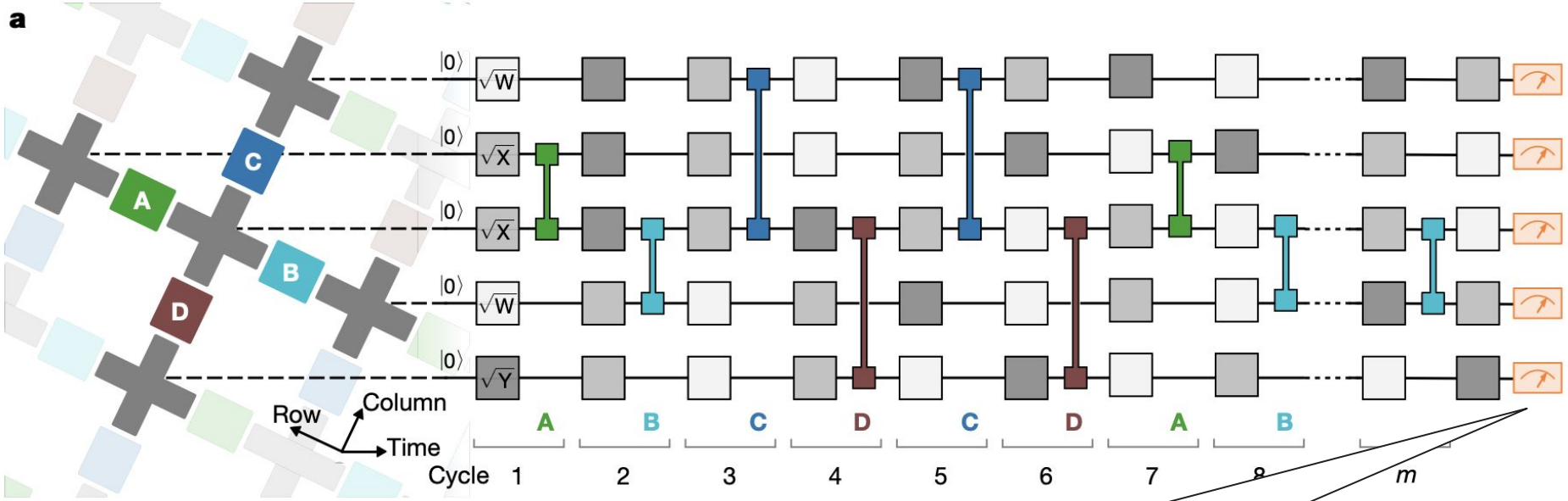
Sycamore: Supported Elements of the Fault-Tolerance Control Loop

Slide intentionally left blank

Runs on quantum processor

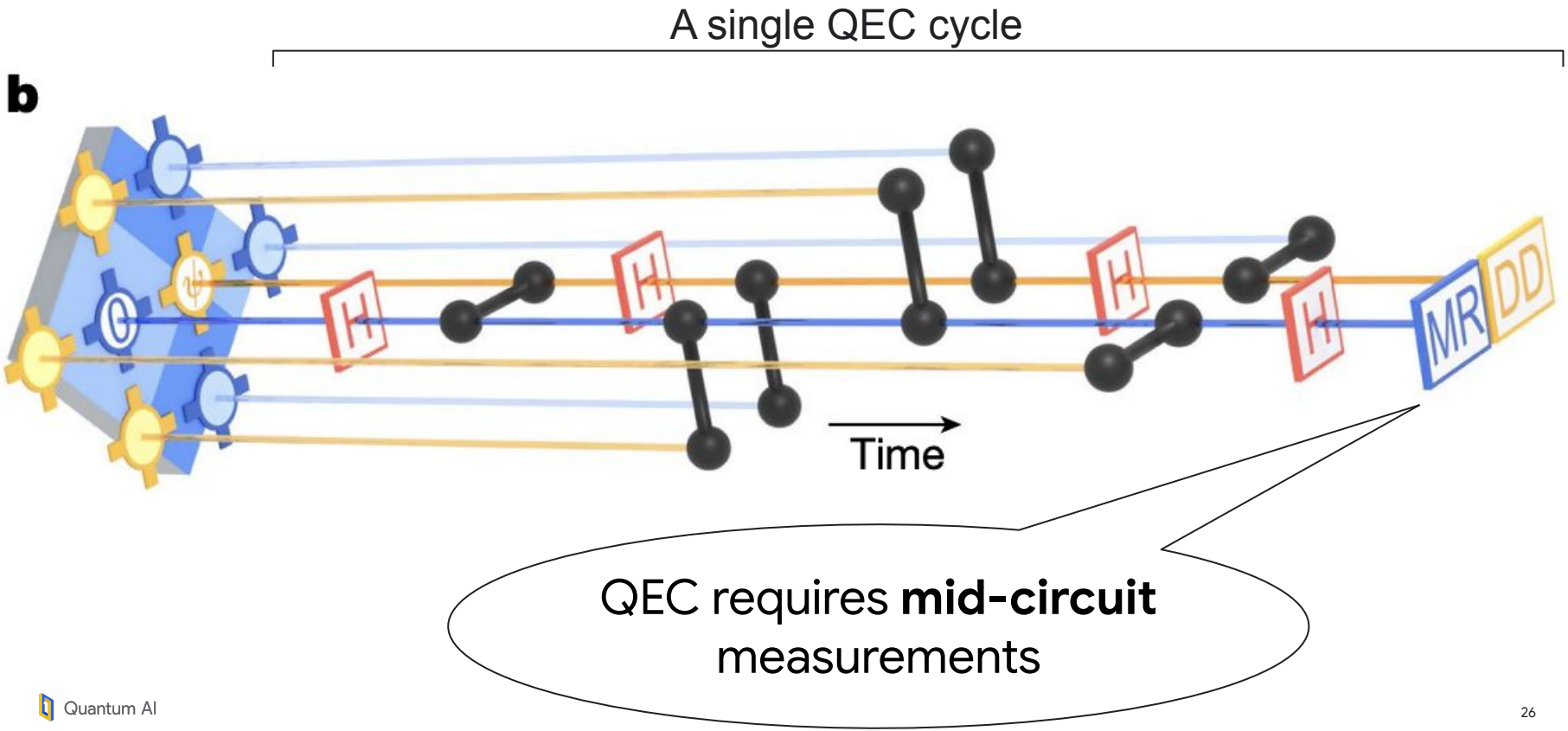
Runs on classical processor

Sycamore can run Quantum Supremacy circuits



Terminal measurements suffice for supremacy demo

Willow supports Quantum Error Correction circuits



Sycamore vs Willow: Sample of System-level Capabilities

Sample of capabilities	Sycamore system (2019)	Willow system (2024)
Is connectivity dense enough for surface code QEC?	Yes	Yes
Can it perform safe mid-circuit measurements?	No	Yes
Is error rate below threshold?	No	Yes
Can it decode syndrome in real-time?	No	Yes
Can it feed the result back to control quantum gates?	No	No
Can it produce magic?	No	No
Can it consume magic to execute T gates?	No	No
Can it exchange quantum information with other chips?	No	No

Roadmap: Understanding Quantum Progress

How do we make progress?

- Assessing Capabilities vs. Counting Qubits
- **Across Platforms: Anderson-Tushman Model**
- Compiling: The finish line gets closer

What happens when we get there?

- Decoded Quantum Interferometry
- Quantum Algorithm for Planted Inference

Anderson-Tushman Model of Technological Change

JOURNAL ARTICLE

Technological Discontinuities and Dominant Designs: A Cyclical Model of Technological Change

Philip Anderson and Michael L. Tushman

Administrative Science Quarterly

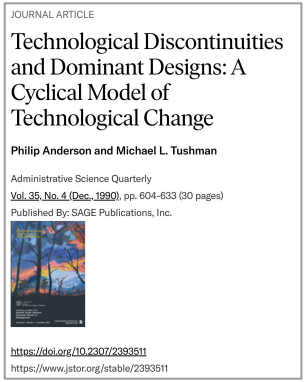
Vol. 35, No. 4 (Dec., 1990), pp. 604-633 (30 pages)

Published By: SAGE Publications, Inc.

https://doi.org/10.2307/2393511

https://www.jstor.org/stable/2393511

Anderson-Tushman Model of Technological Change


JOURNAL ARTICLE

Technological Discontinuities and Dominant Designs: A Cyclical Model of Technological Change

Philip Anderson and Michael L. Tushman

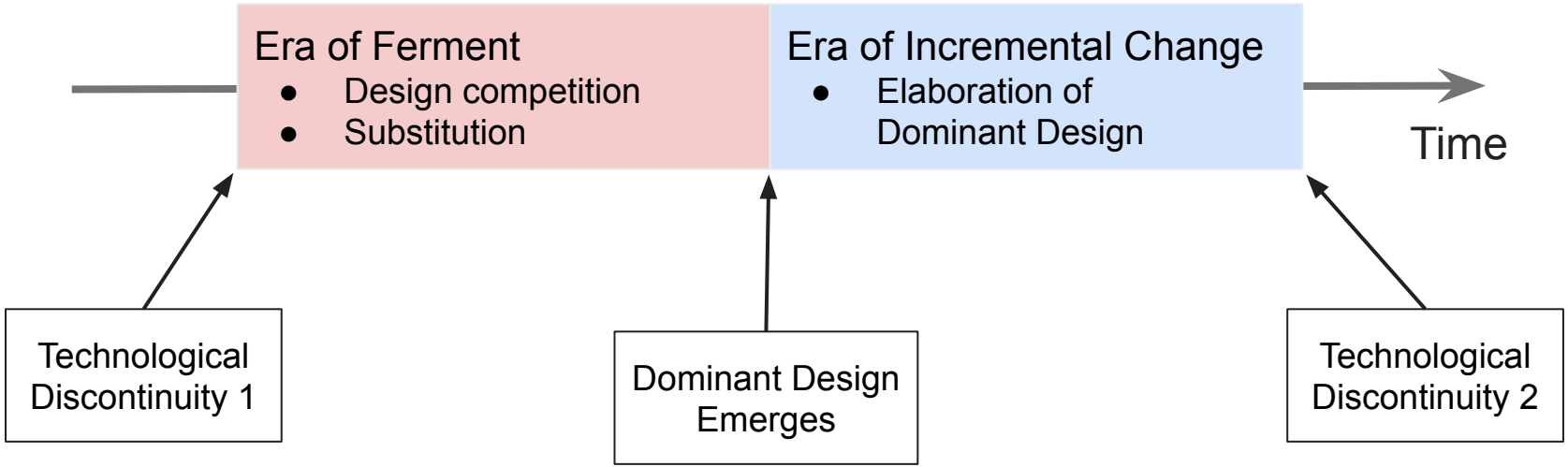
Administrative Science Quarterly

Vol. 35, No. 4 (Dec., 1990), pp. 604-633 (30 pages)

Published By: SAGE Publications, Inc.

https://doi.org/10.2307/2393511

https://www.jstor.org/stable/2393511

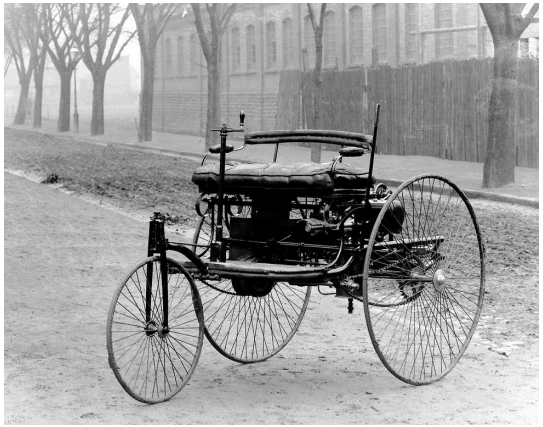


Car Industry in late 1800s

Steam

Jacquot Tonneau steam car (1878)

Internal Combustion



Benz Patent-Motorwagen (1886)

Electricity

Flocken Elektrowagen (1888)

~~How many miles per gallon?~~

What is the range?

Car Industry in late 1900s

Internal Combustion

Toyota Supra
(1985)

Internal Combustion

Corvette C4
(1990)

Internal Combustion

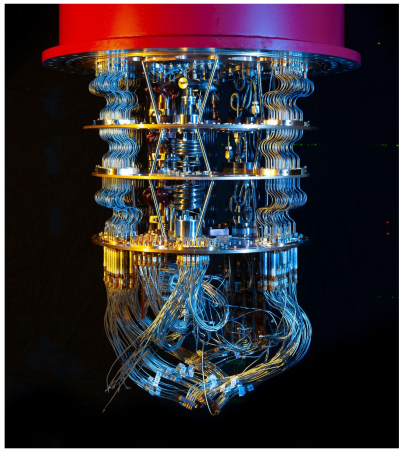
Alpha Romeo 155 GTA Stradale
(1992)

How many miles per gallon?

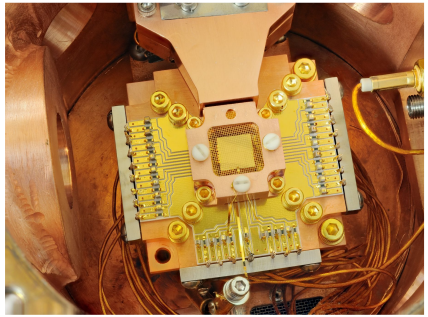
What is the range?

Quantum Computing in 2020s

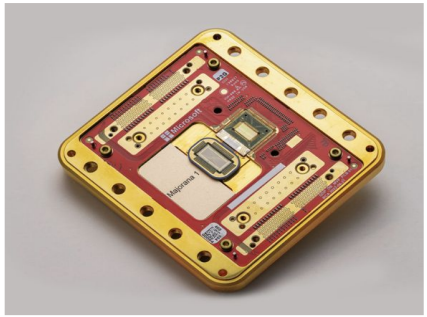
Superconductors



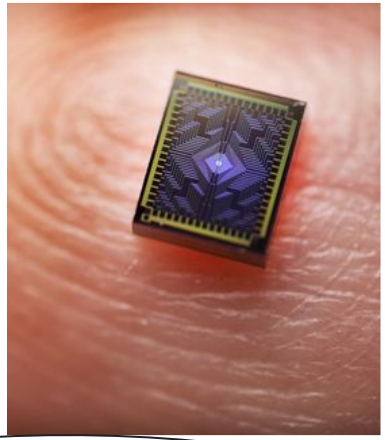
Ions



Majoranas



Spins



Photons

Neutral Atoms

~~How many qubits does it have?~~

What time and device size are needed to factor 2048-bit integer?

arXiv:2006.12326

Quantum Computing industry is going through its *Era of Ferment*

Meaningful comparisons involve system-level performance. Numbers of physical qubits between devices in different architectures are **not directly comparable**.

Roadmap: Understanding Quantum Progress

How do we make progress?

- Assessing Capabilities vs. Counting Qubits
- Across Platforms: Anderson-Tushman Model
- **Compiling: The finish line gets closer**

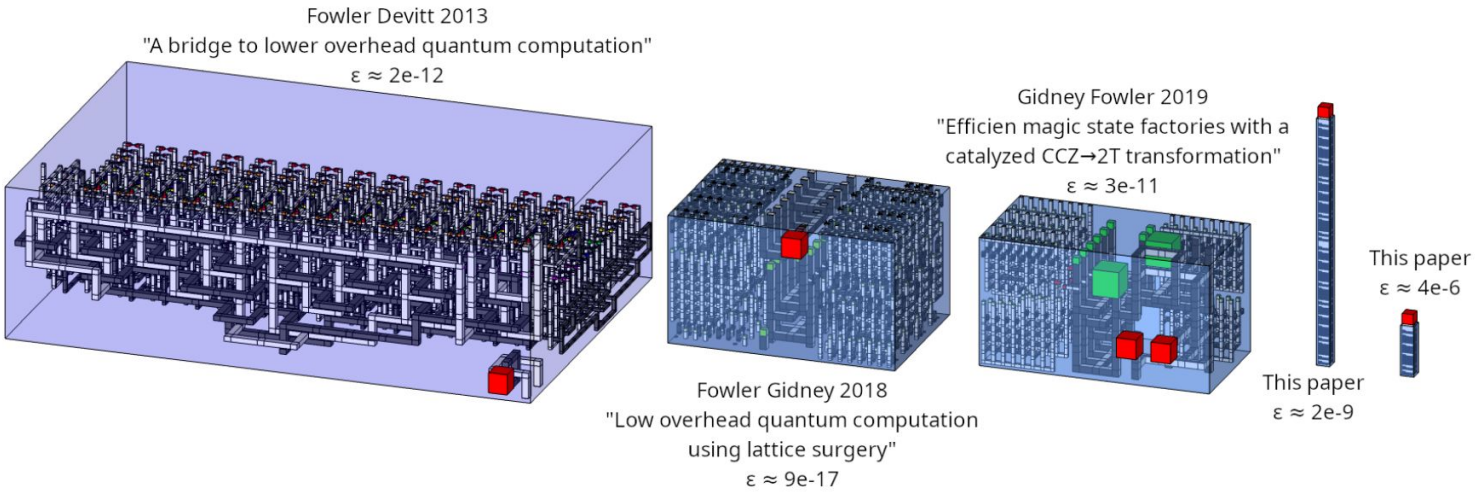
What happens when we get there?

- Decoded Quantum Interferometry
- Quantum Algorithm for Planted Inference

Progress in Optimizing Magic Production

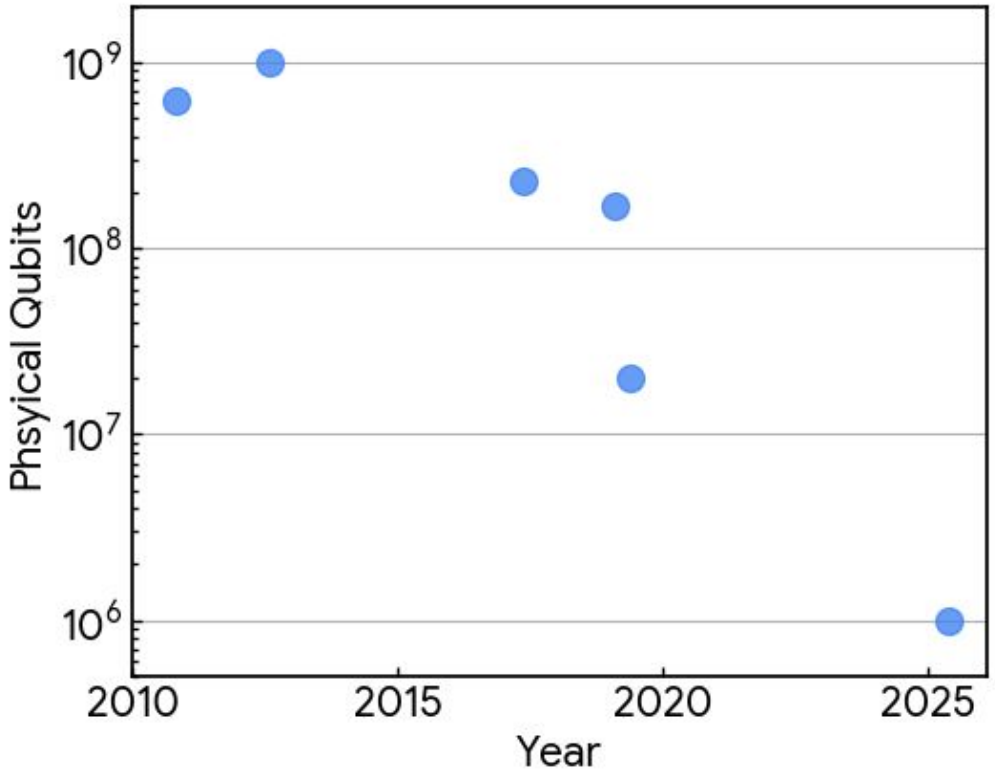
2012 → 2024

Computational volume required for T state creation

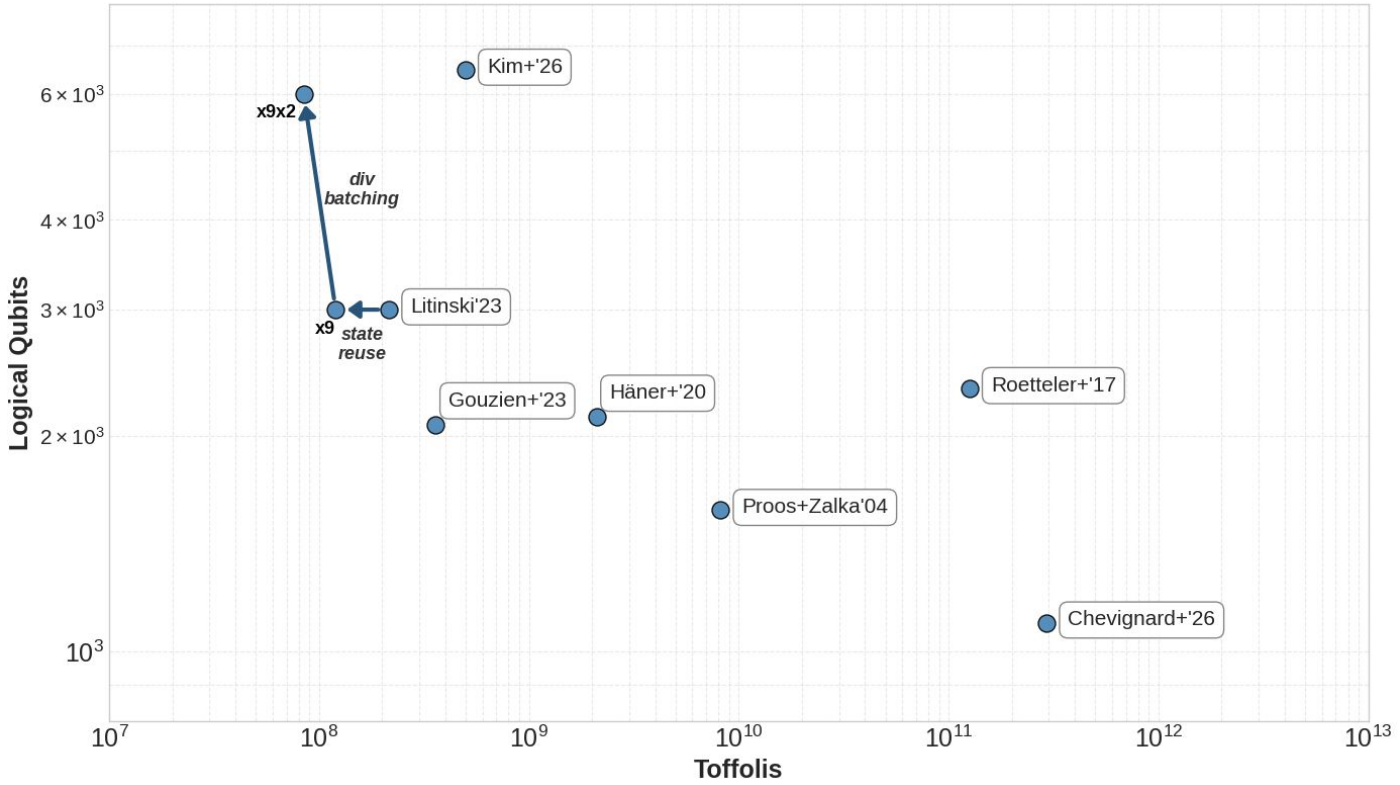


C. Gidney et al, arXiv:2409.17595 (2024)

Progress in Compiling Shor's Algorithm for 2048-bit RSA integers



Progress in Compiling Shor's Algorithm for 256-bit ECDLP



Roadmap: Understanding Quantum Progress

How do we make progress?

- Assessing Capabilities vs. Counting Qubits
- Across Platforms: Anderson-Tushman Model
- Compiling: The finish line gets closer

What happens when we get there?

- **Decoded Quantum Interferometry**
- Quantum Algorithm for Planted Inference

Decoded Quantum Interferometry

Article

Optimization by decoded quantum interferometry

https://doi.org/10.1038/s41586-025-09527-5

Received: 4 December 2024

Accepted: 13 August 2025

Published online: 22 October 2025

Open access

Check for updates

Stephen P. Jordan[1]✉, Noah Shutty[1]✉, Mary Wootters[2,3], Adam Zalcman[1], Alexander Schmidhuber[1,4], Robbie King[1,5], Sergei V. Isakov[1], Tanuj Khattar[1] & Ryan Babbush[1]

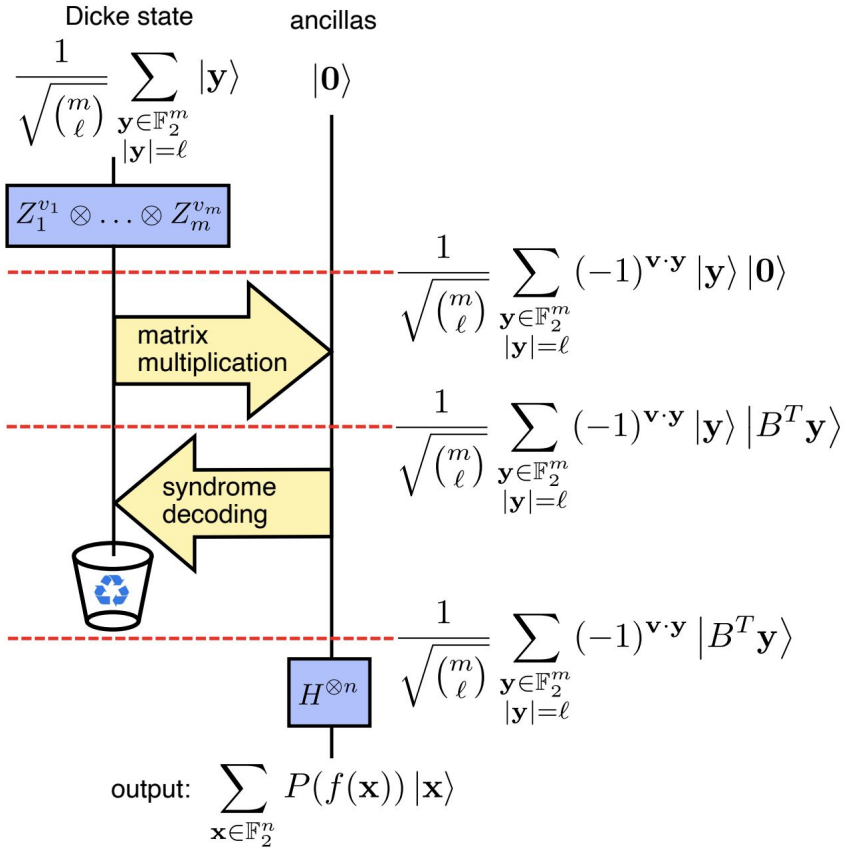
Achieving superpolynomial speed-ups for optimization has long been a central goal for quantum algorithms[1]. Here we introduce decoded quantum interferometry (DQI), a quantum algorithm that uses the quantum Fourier transform to reduce optimization problems to decoding problems. When approximating optimal polynomial fits over finite fields, DQI achieves a superpolynomial speed-up over known classical algorithms. The speed-up arises because the algebraic structure of the problem is reflected in the decoding problem, which can be solved efficiently. We then investigate whether this approach can achieve a speed-up for optimization problems that lack an algebraic structure but have sparse clauses. These problems reduce to decoding low-density parity-check codes, for which powerful decoders are known[2,3]. To test this, we construct a max-XORSAT instance for which DQI finds an approximate optimum substantially faster than general-purpose classical heuristics, such as simulated annealing. Although a tailored classical solver can outperform DQI on this instance, our results establish that combining quantum Fourier transforms with powerful decoding primitives provides a promising new path towards quantum speed-ups for hard optimization problems.

Decoded Quantum Interferometry

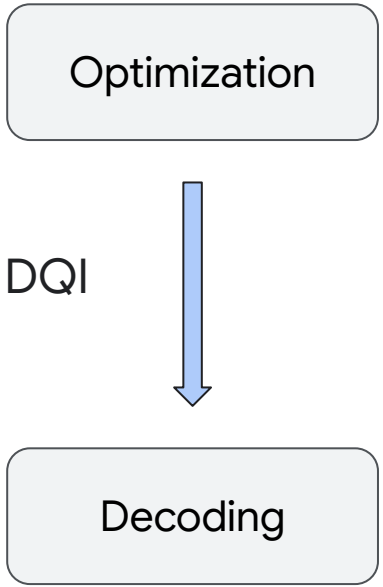
Optimization

DQI

Decoding



Decoded Quantum Interferometry



- DQI uses Quantum Fourier transform to reduce optimization problems to decoding problems
- DQI's performance can be **characterized rigorously**. Indeed, the fraction of satisfied constraints s/m can be expressed in terms of the performance the decoder subroutine ℓ/m and how restrictive the constraints are r/p:

$$\frac{\langle s \rangle}{m} = \left(\sqrt{\frac{\ell}{m}\left(1 - \frac{r}{p}\right)} + \sqrt{\frac{r}{p}\left(1 - \frac{\ell}{m}\right)} \right)^2$$

- It is based on Regev's reduction and has connections to **Learning With Errors** and **Oblivious Polynomial Evaluation**.

Roadmap: Understanding Quantum Progress

How do we make progress?

- Assessing Capabilities vs. Counting Qubits
- Across Platforms: Anderson-Tushman Model
- Compiling: The finish line gets closer

What happens when we get there?

- Decoded Quantum Interferometry
- **Quantum Algorithm for Planted Inference**

Quantum Algorithm for Planted Inference

PHYSICAL REVIEW X **15,** 021077 (2025)

Quartic Quantum Speedups for Planted Inference

Alexander Schmidhuber,[1,2] Ryan O'Donnell,[3] Robin Kothari,[1] and Ryan Babbush[1]
[1]*Google Quantum AI, Venice, California, USA*
[2]*Center for Theoretical Physics, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA*
[3]*Computer Science Department, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA*

(Received 11 November 2024; revised 23 April 2025; accepted 2 May 2025; published 2 June 2025)

We describe a quantum algorithm for the Planted Noisy kXOR Problem (also known as Sparse Learning Parity with Noise) that achieves a nearly *quartic* (fourth-power) speedup over the best known classical algorithm while using exponentially less space. Our work generalizes and simplifies prior work of Hastings [Quantum **4**, 237 (2020)], by building on his quantum algorithm for the tensor principal component analysis (PCA) problem. We achieve our quantum speedup using a general framework based on the Kikuchi method (recovering the quartic speedup for Tensor PCA), and we anticipate it will yield similar speedups for further planted inference problems. These speedups rely on the fact that planted inference problems naturally instantiate the guided sparse Hamiltonian problem. Since the Planted Noisy kXOR Problem has been used as a component of certain cryptographic constructions, our work suggests that some of these are susceptible to superquadratic quantum attacks.

DOI: 10.1103/PhysRevX.15.021077 Subject Areas: Quantum Information

Quantum Algorithm for Planted Inference

- Quartic (square of square) quantum speedup and exponential space reduction over classical algorithms.
- Maps planted inference problems to the problem of estimating the ground-state energy of a certain mean-field Hamiltonian and solves it by preparing a *guiding* state with significant overlap with the ground state of the Hamiltonian.
- Solves sparse variant of **Learning Parity with Noise**.

Research into quantum algorithms based on Regev's reduction has led to results in quantum optimization.

Research into quantum algorithms based on the Kikuchi method has led to quartic quantum speedup for sparse Learning Parity with Noise.

Quantum AI

Thank you!

Google