DR JEFF SOFTWARE
INDIE APP DEVELOPER

Classroom

Rev.12-3-23

Quantum Computing

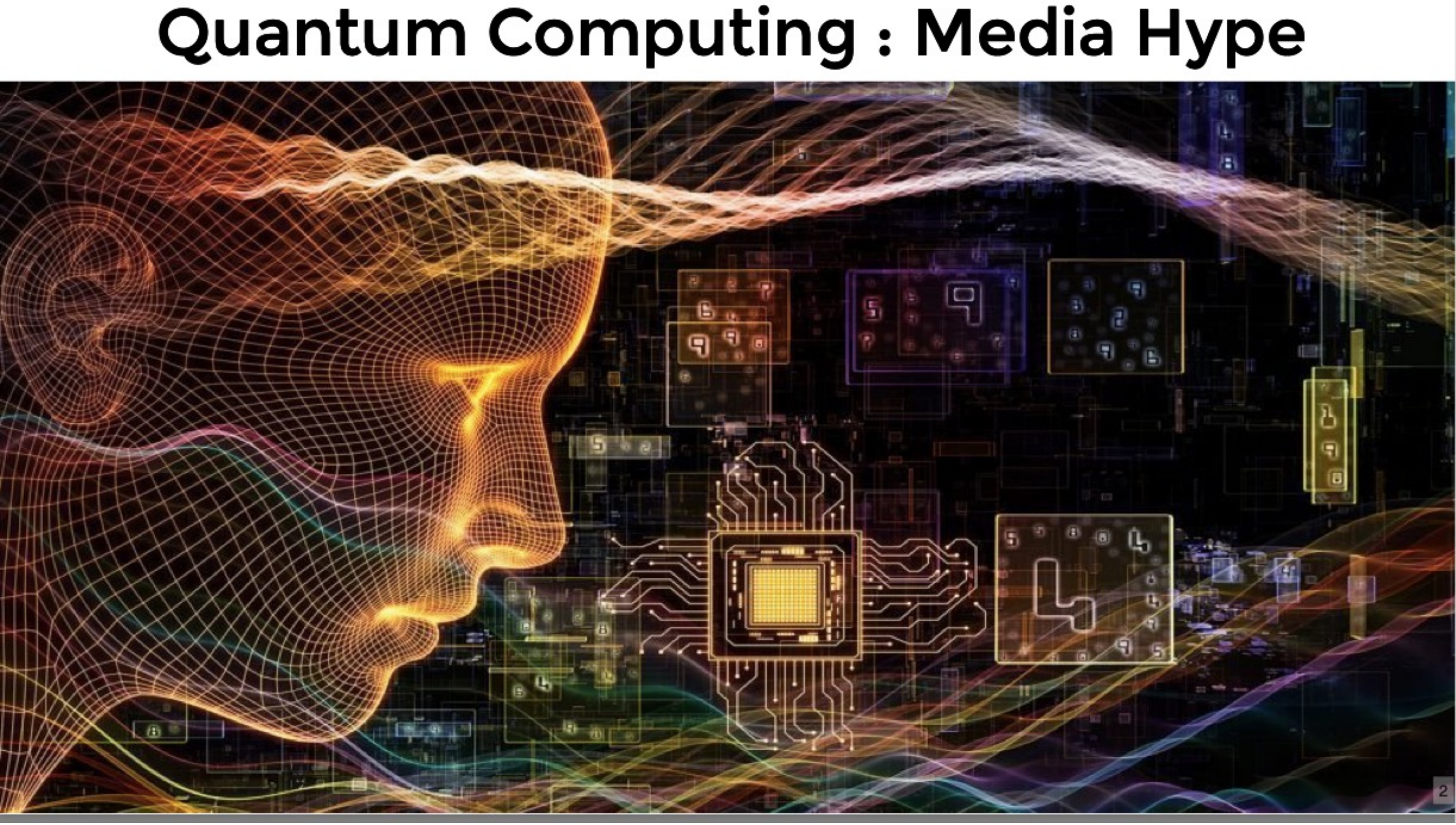
by

Dr Jeff Drobman

Dr Jeff Software

Lecturer, CSUN

Quantum Computing



Quantum Computing : Media Hype

QC's

How close are we to practical quantum computers?

We already have them! ... sort of

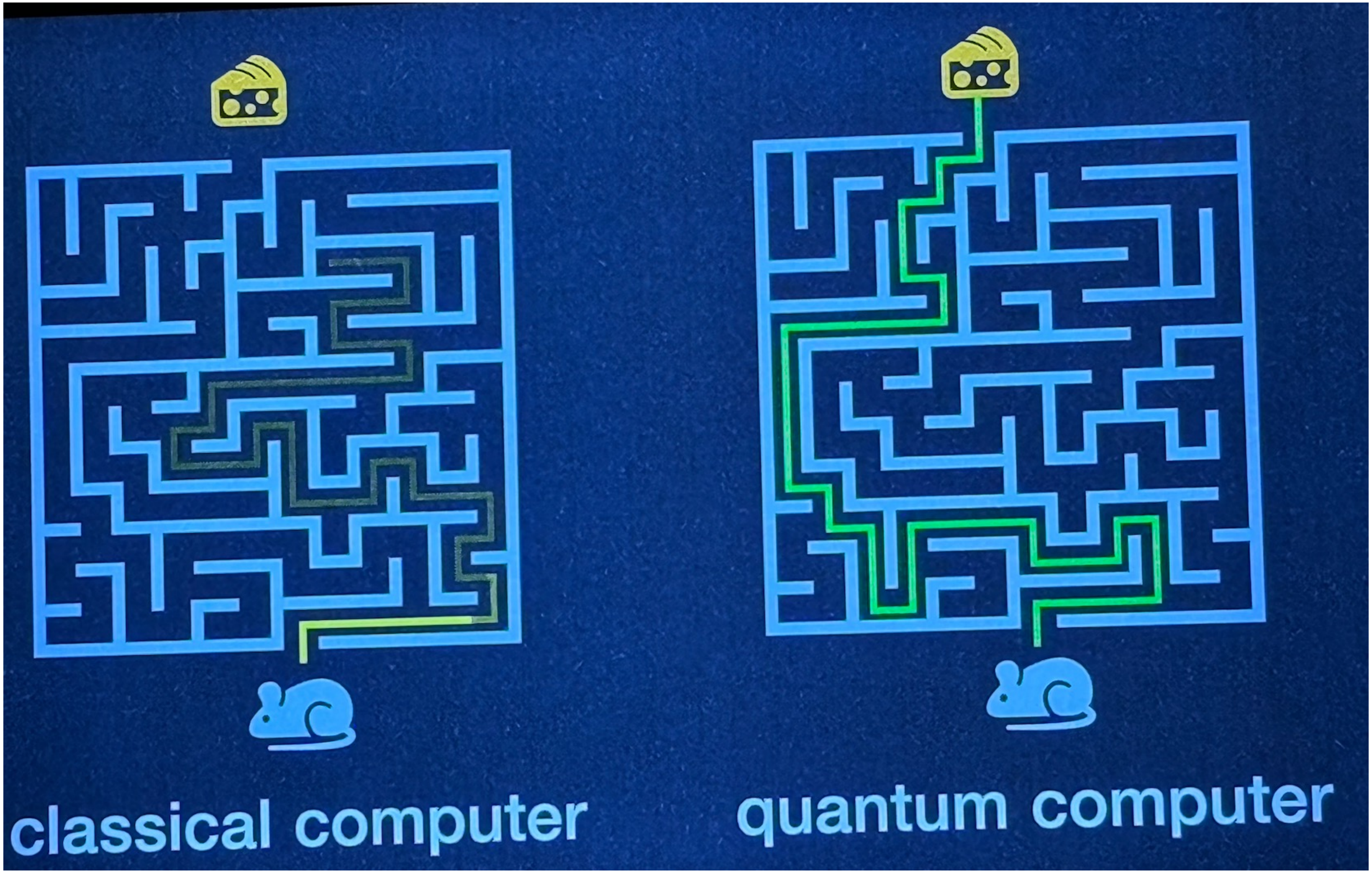
2 main competing implementations (others in development):

1. Trapped ions
   UMD : 53 qubits

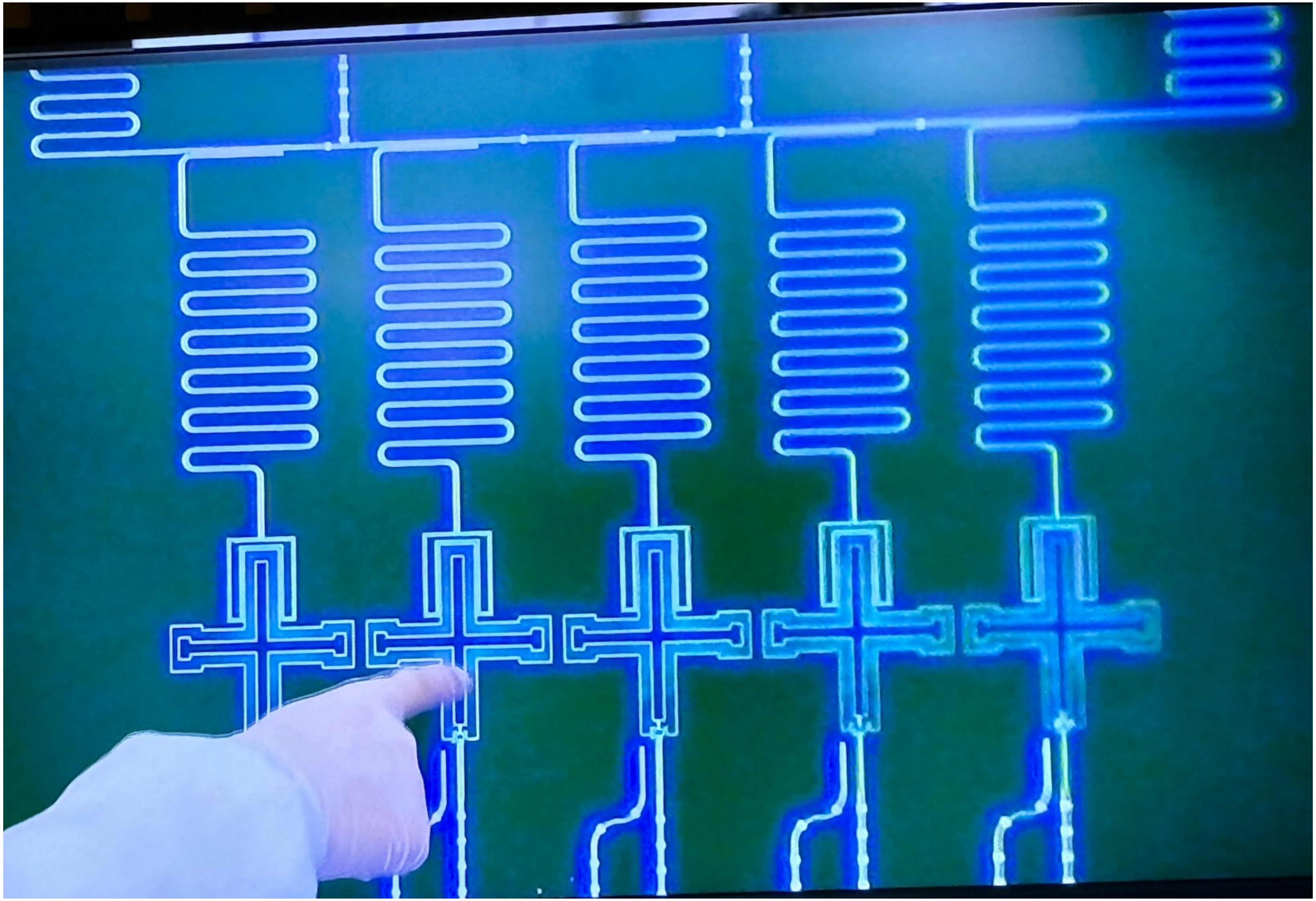
2. Superconducting circuits
   Google : 72 qubits
   IBM : 50 qubits
   Rigetti Computing : 19 qubits
   UC Berkeley : 10 qubits

But these numbers do not tell the complete story

QC vs Classical



QC Qubits



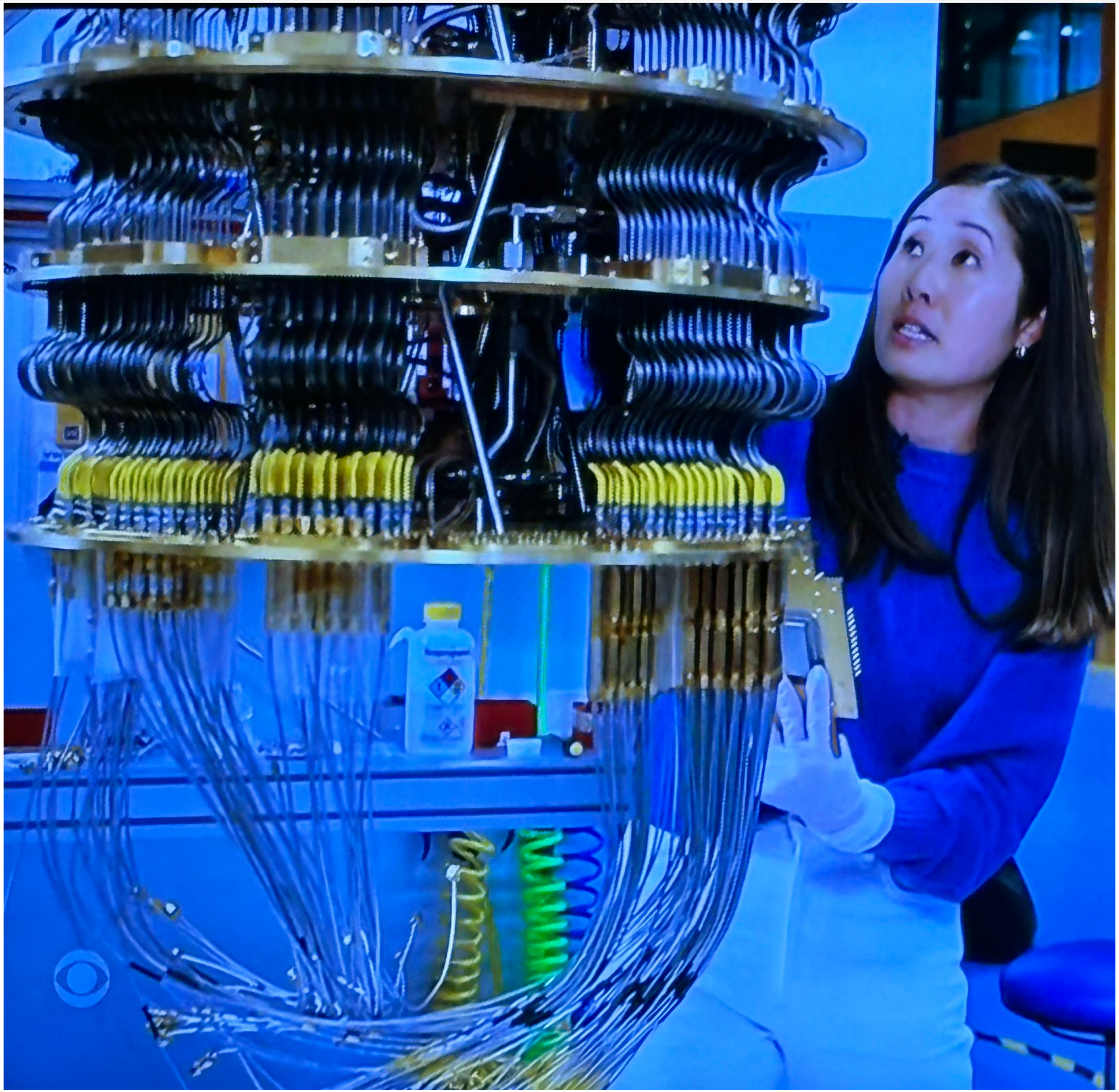
QC

QC 2023 Update

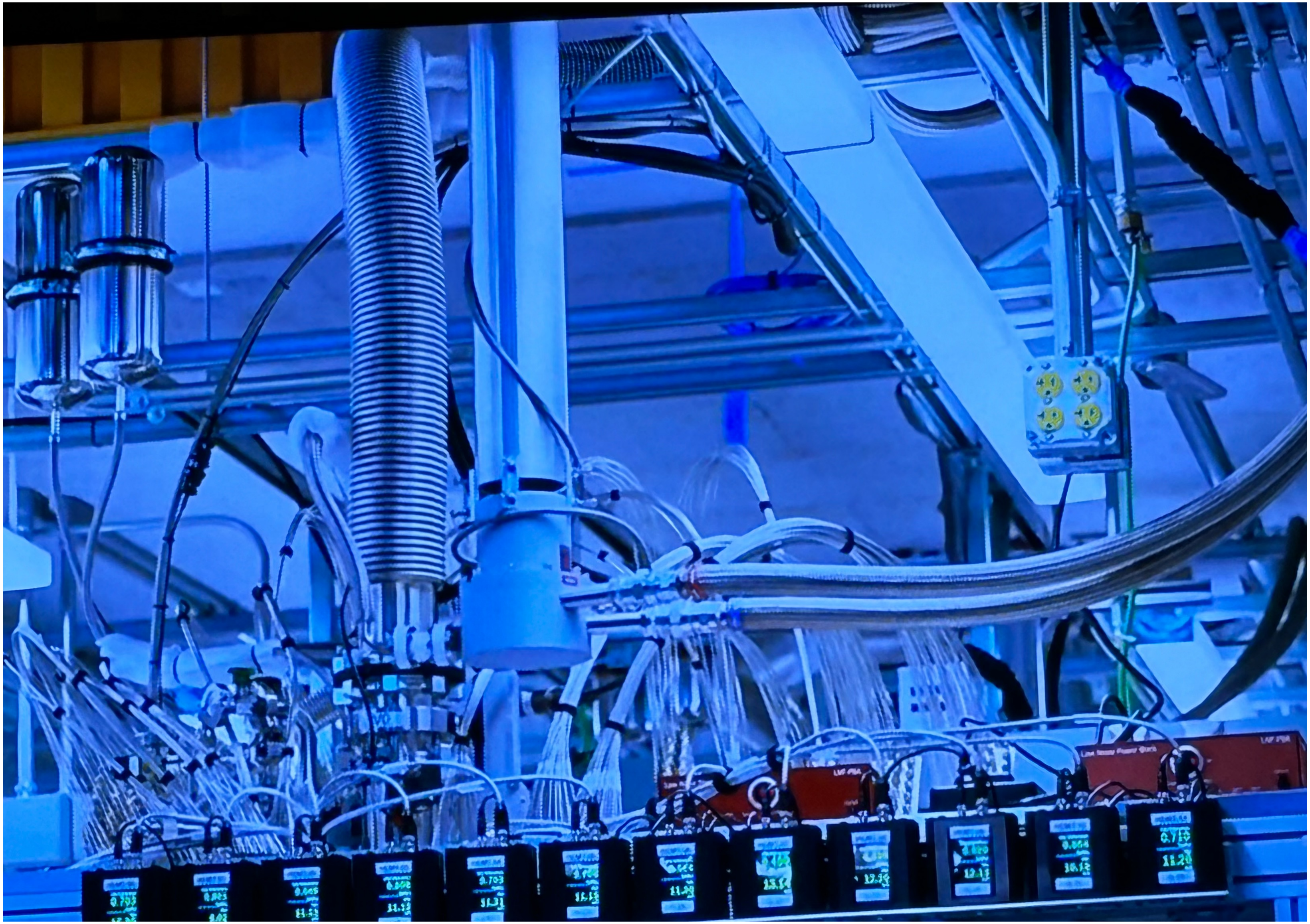
2nd Gen Machines

- Google
- IBM

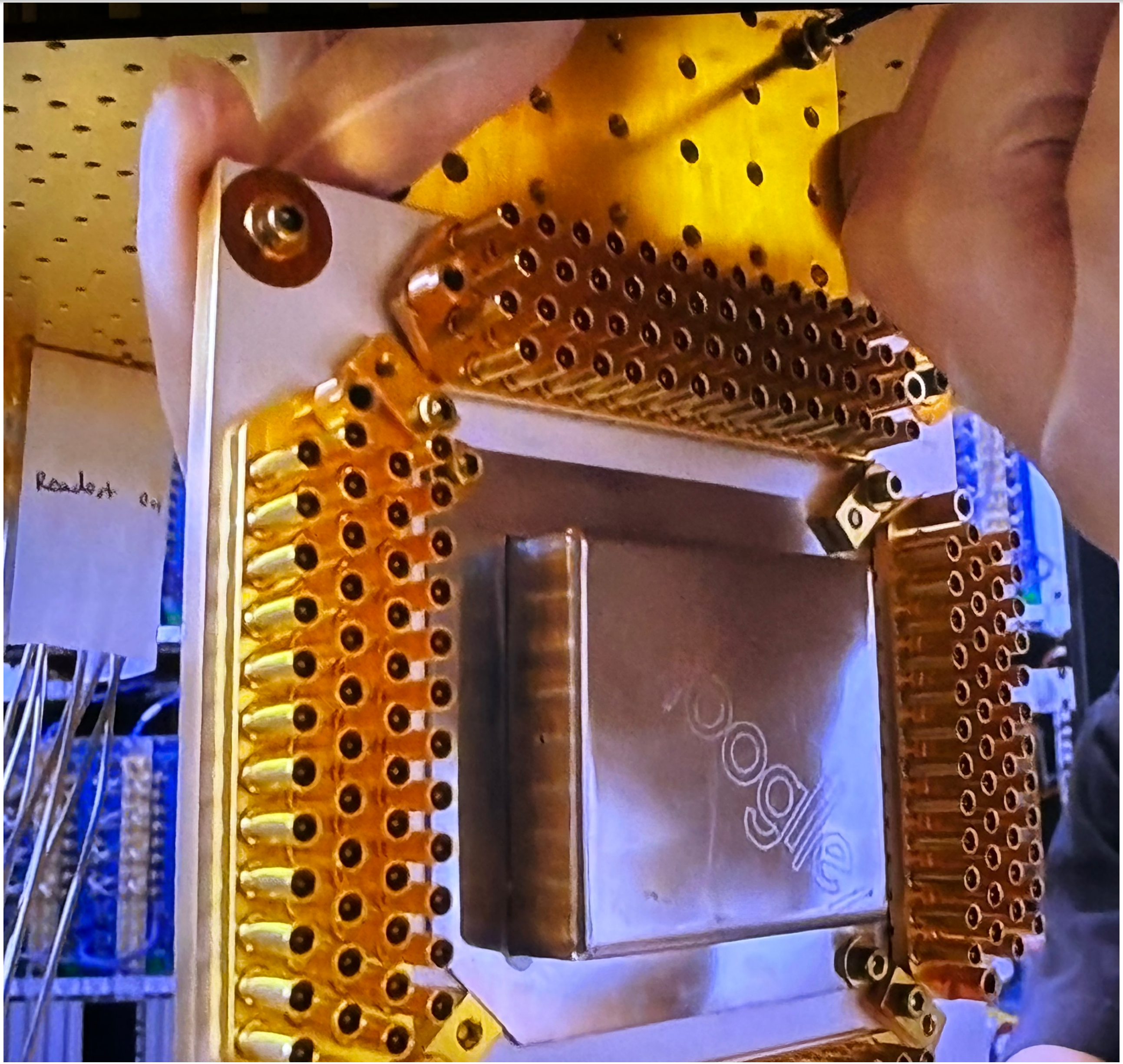
2nd Gen: Google



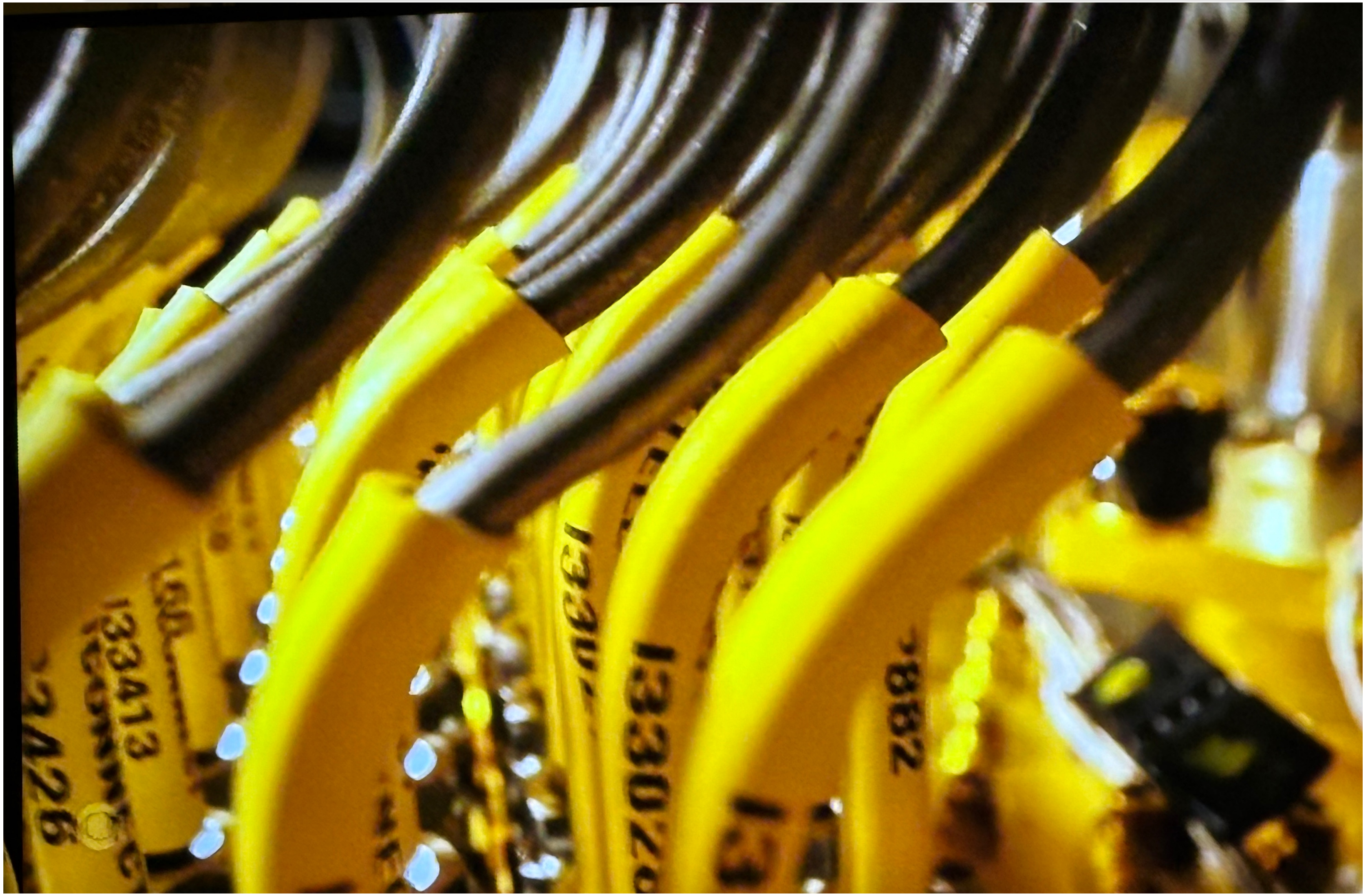
2nd Gen: Google



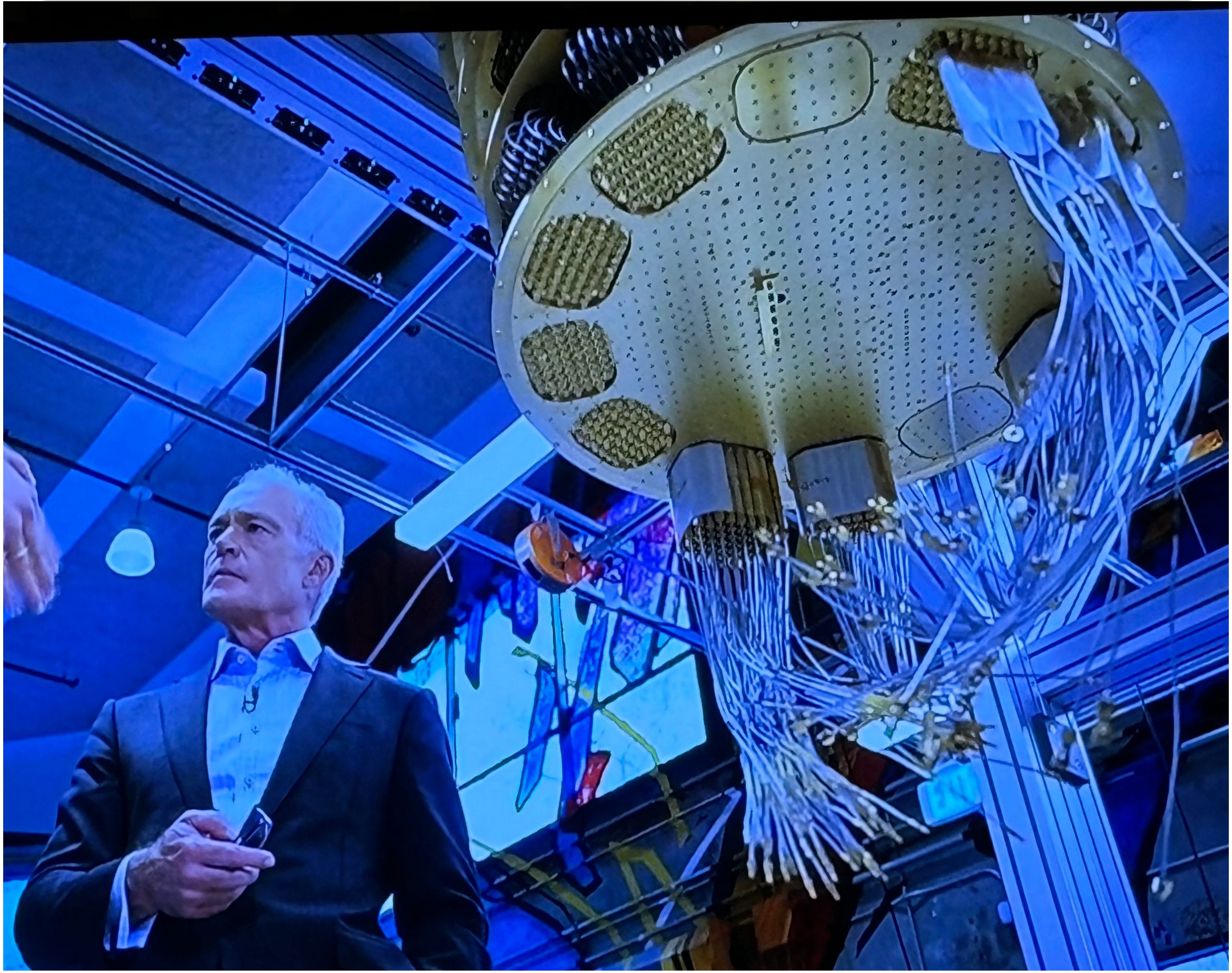
2nd Gen: Google



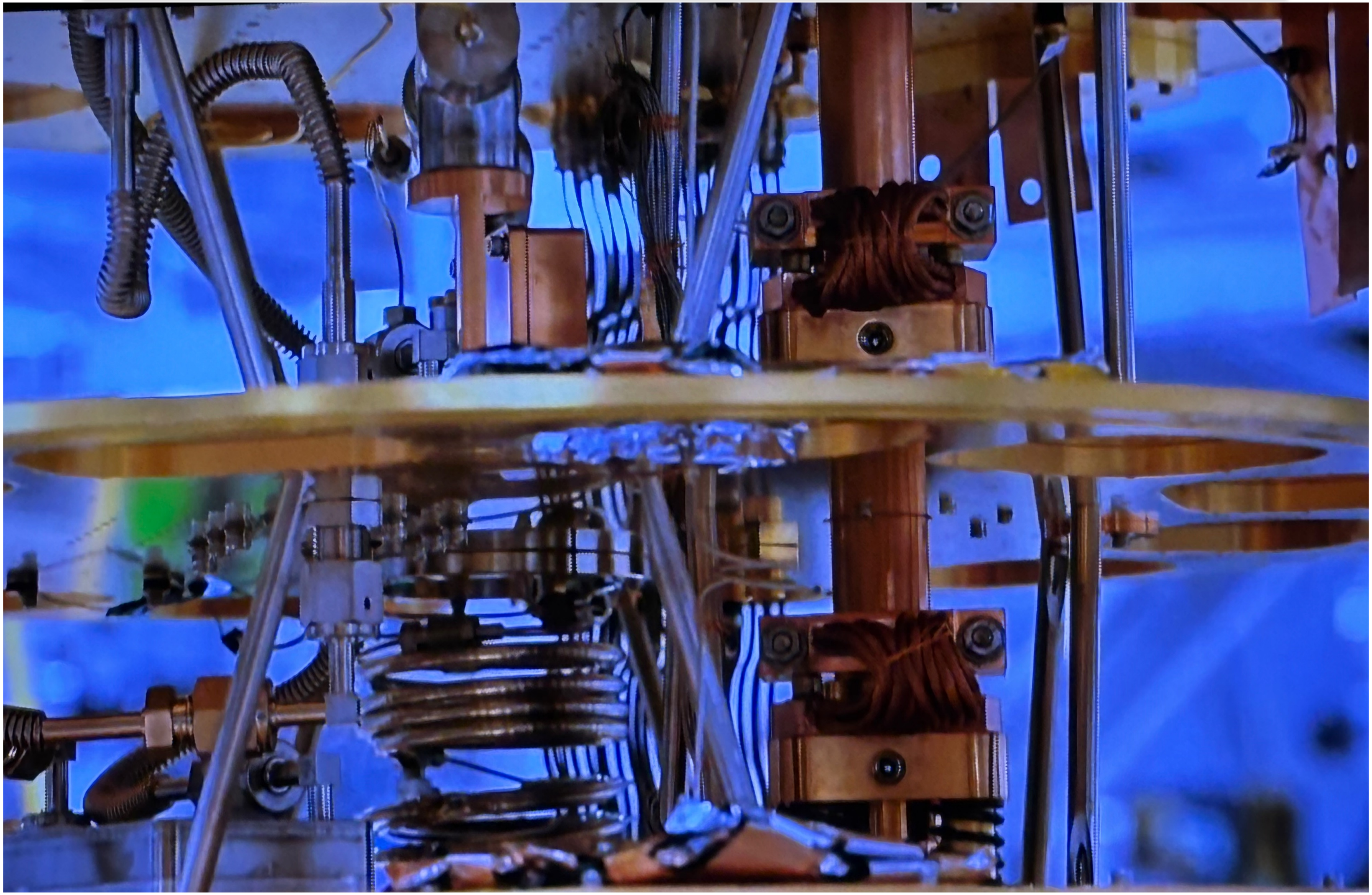
2nd Gen: Google



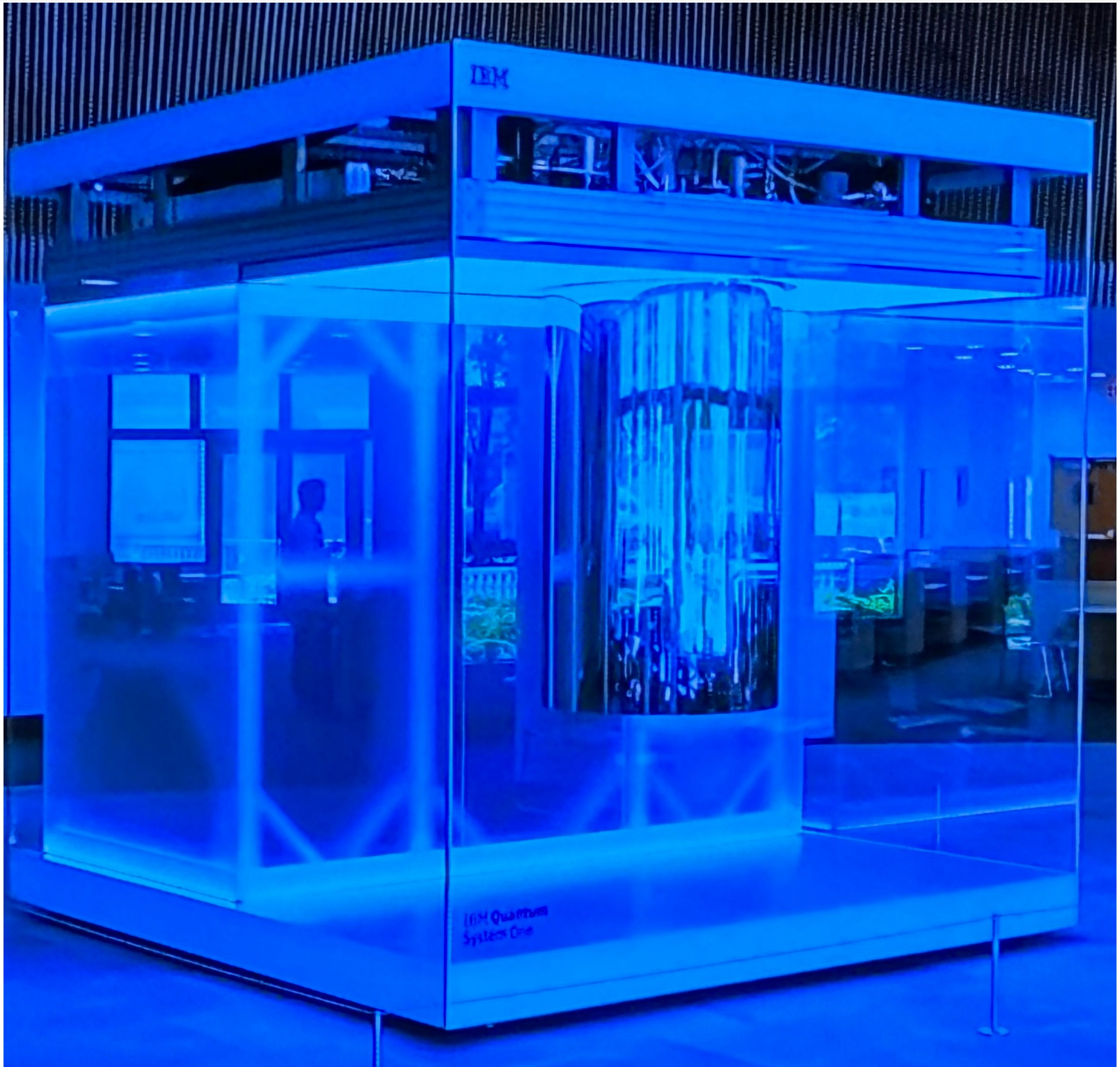
2nd Gen: Google



2^{nd} Gen: Google

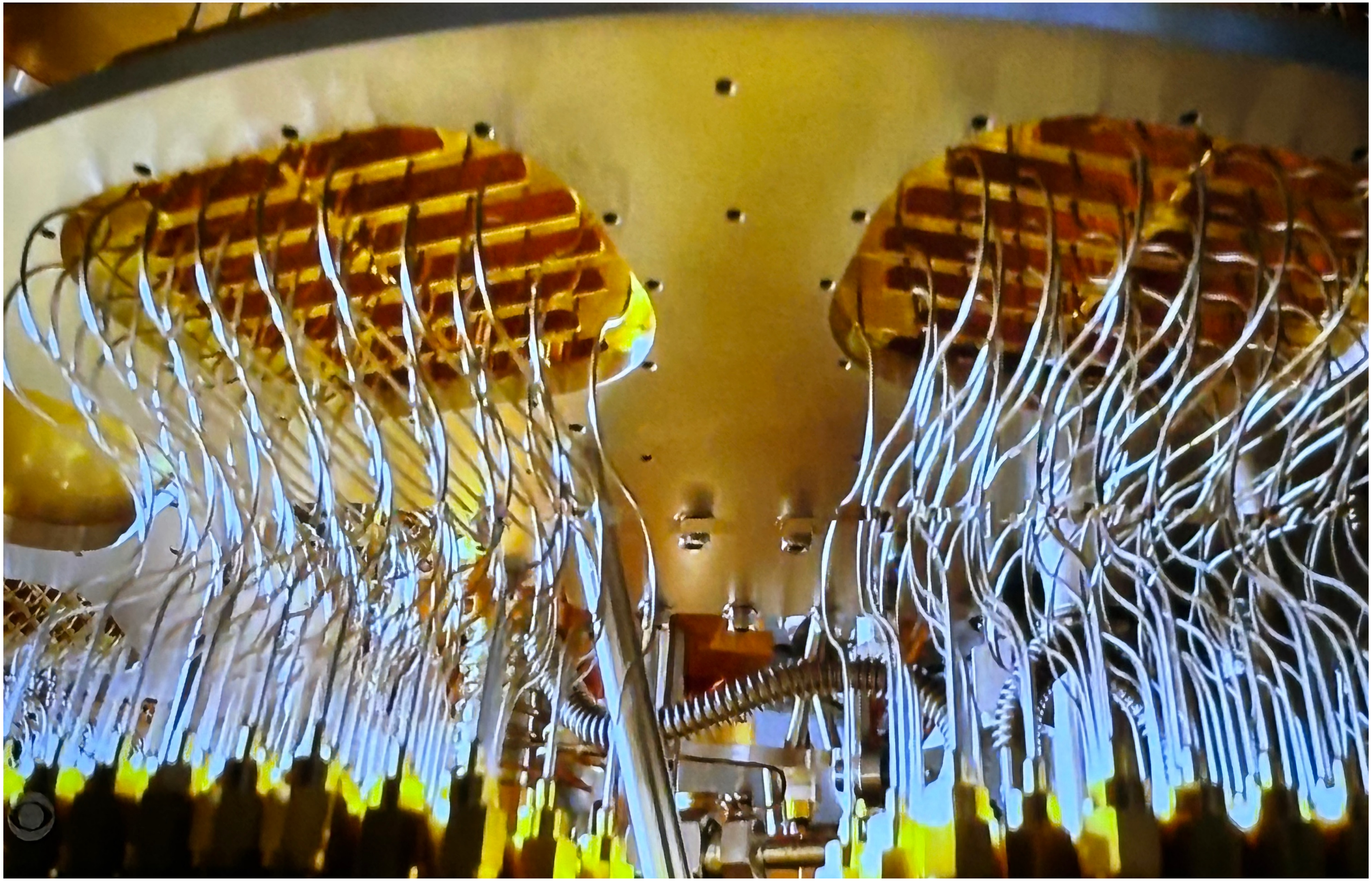


2nd Gen: IBM Q2

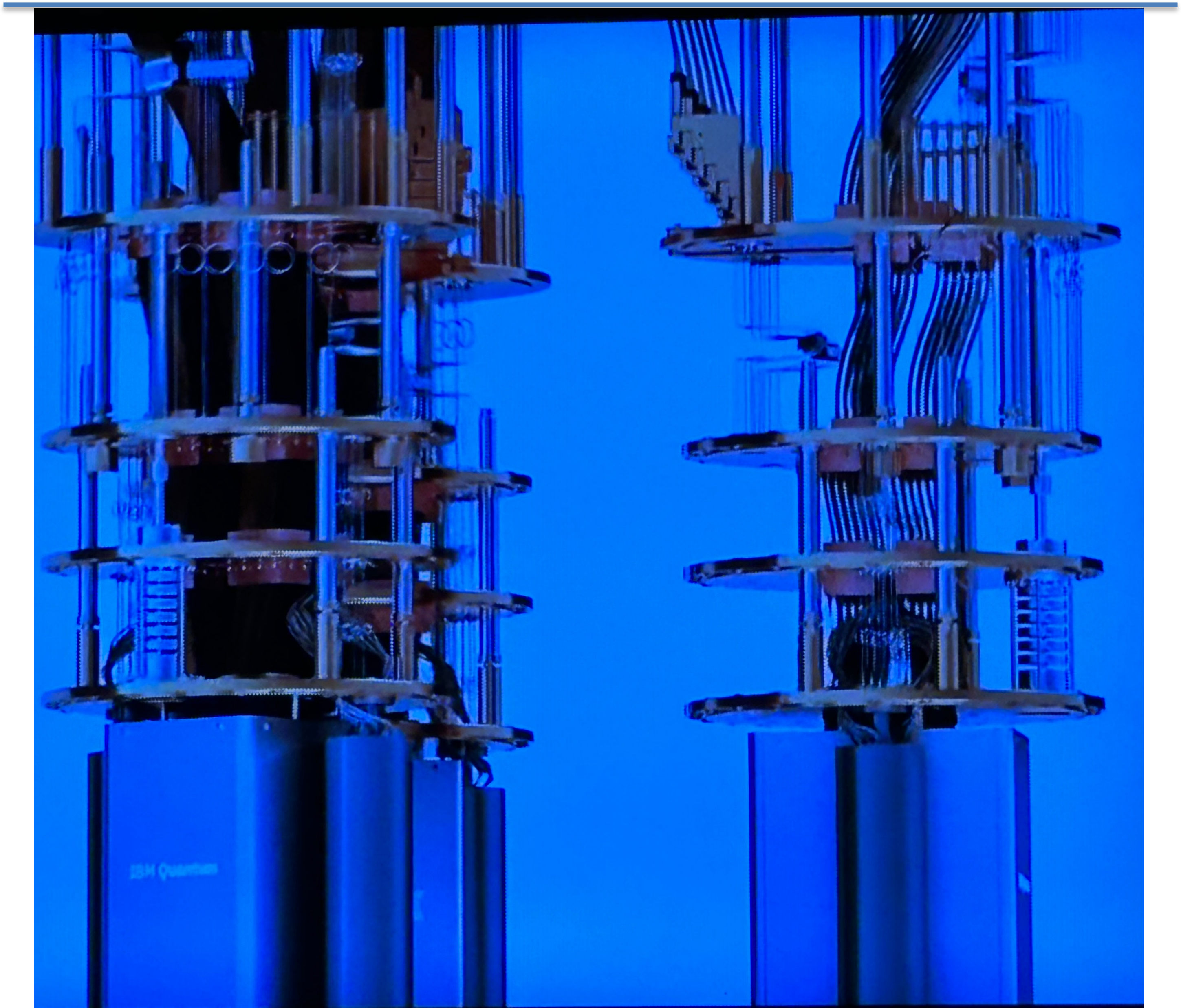


2nd Gen: IBM Q2

2^{nd} Gen: IBM Q2

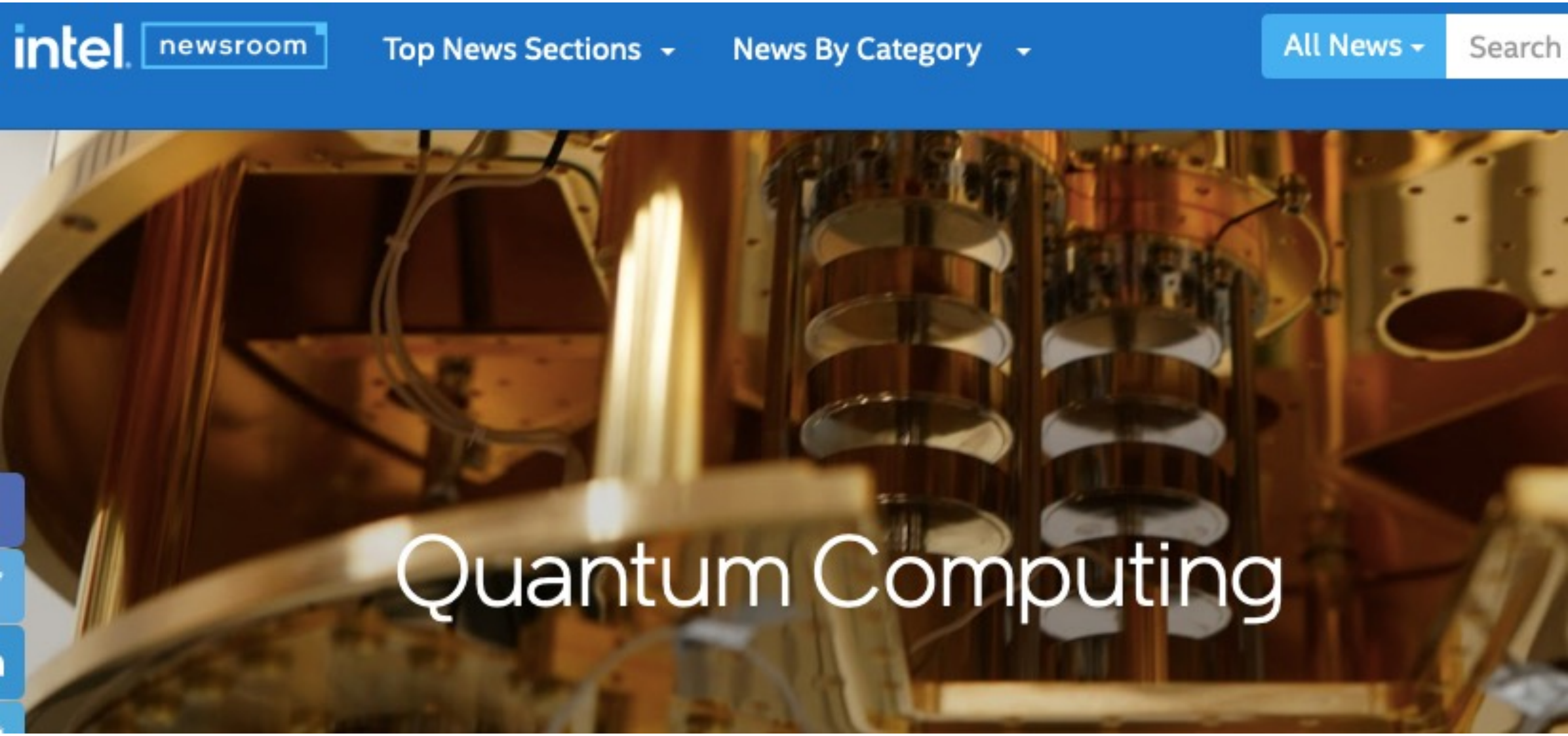


2^{nd} Gen: IBM Q2

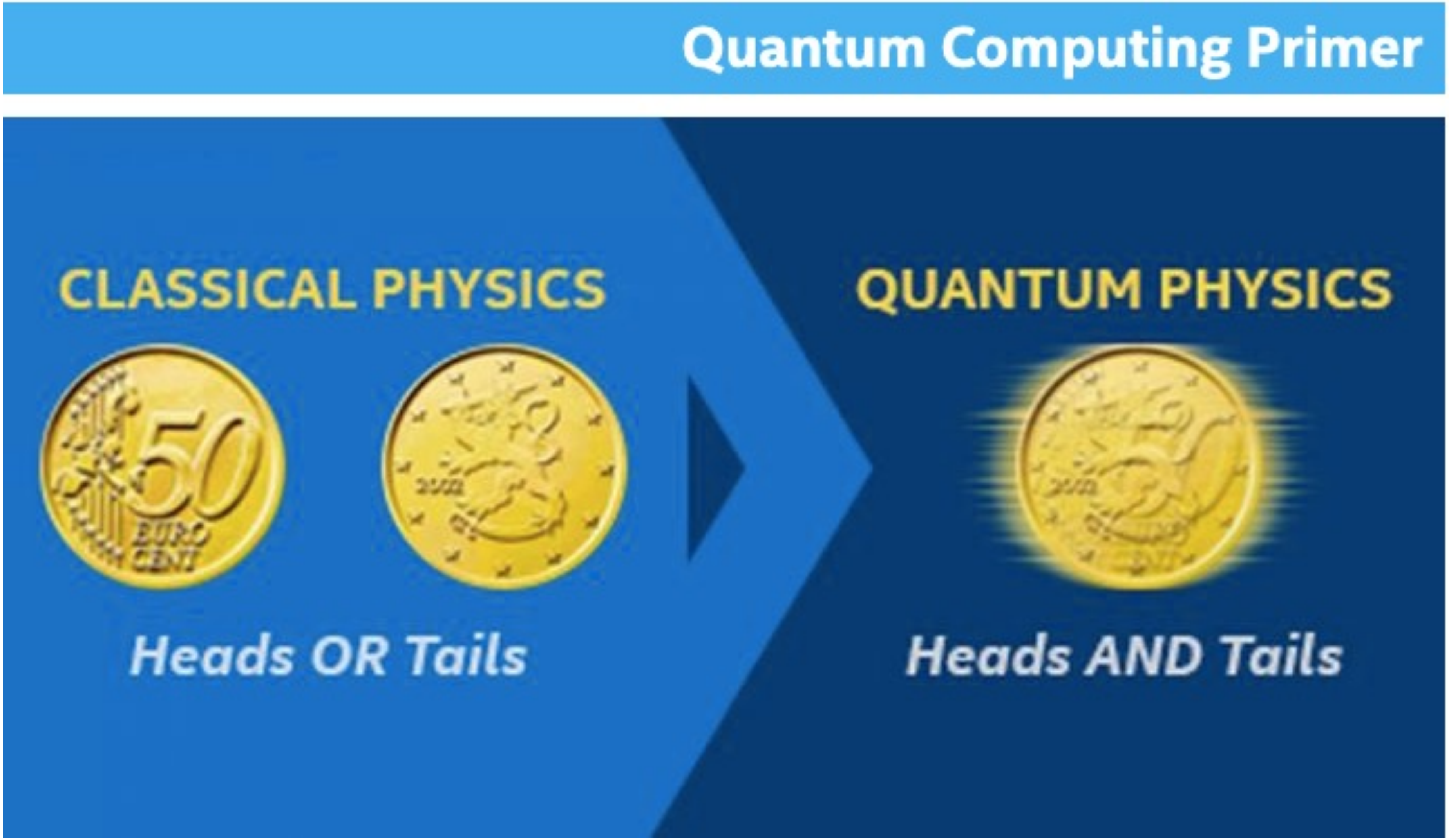


2nd Gen: IBM Q2

Intel Photo


intel newsroom
Top News Sections
News By Category
All News
Search
Quantum Computing

Intel Diagram



Quantum Computing Primer

CLASSICAL PHYSICS

Heads OR Tails

QUANTUM PHYSICS

Heads AND Tails

QC News

July 2020

UC to lead Group Awarded $25M by NSF to Launch Quantum Computing Institute

The National Science Foundation announced a five-year, $25 million award to UC Berkeley, UCLA and other universities to create an institute to study quantum computation. Computer science professor Jens Palsberg is part of the team.

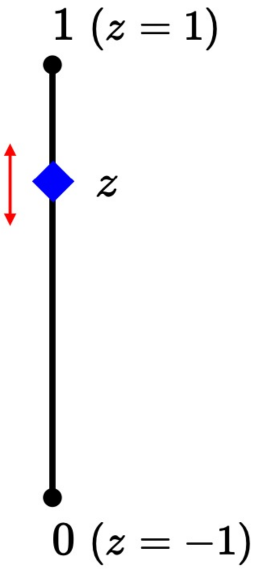
QC’s & Qubits

Probabilistic Bits vs. Quantum Bits

Classical Bit

Only 2 *definite* states: 0 or 1

z-axis connecting them is *indefinite*, or probabilistic



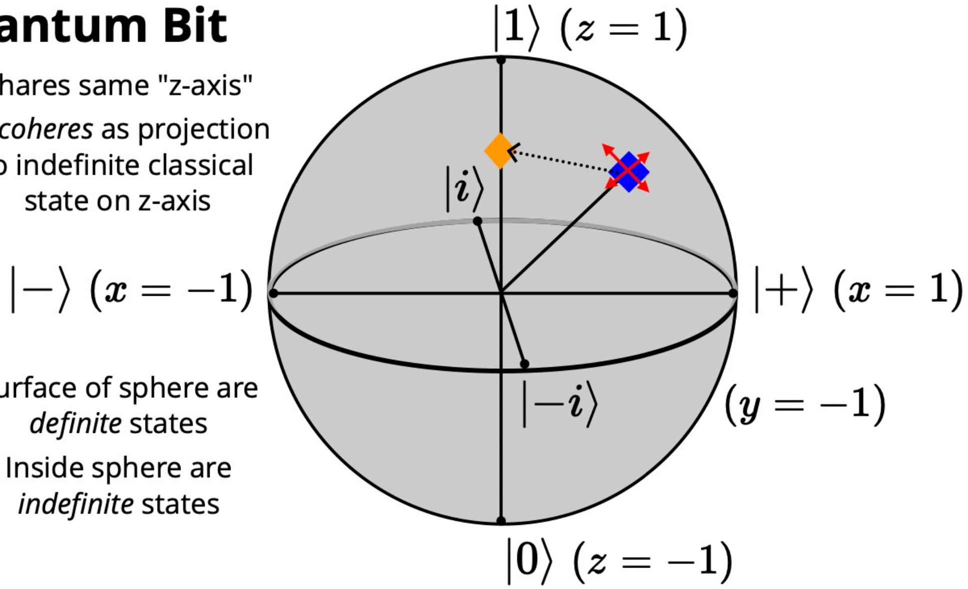
Quantum Bit

Shares same "z-axis"

Decoheres as projection to indefinite classical state on z-axis

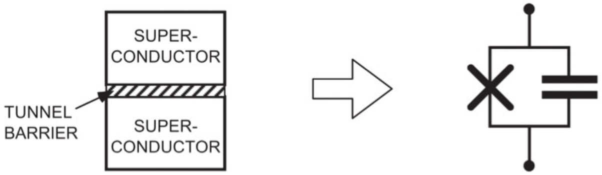
Surface of sphere are *definite* states

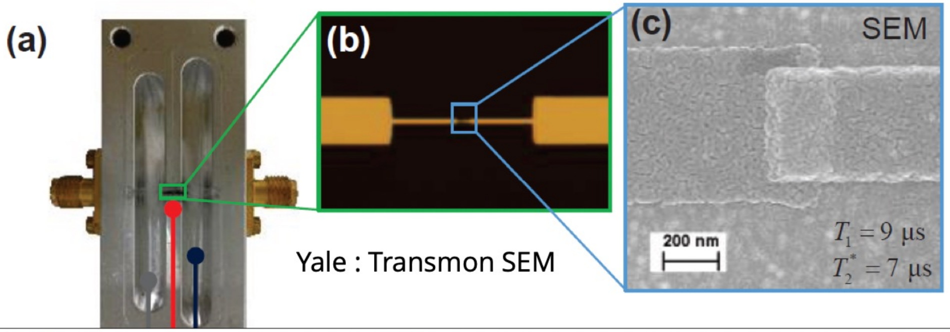
Inside sphere are *indefinite* states



Technology 2 : Superconducting Qubits

A superconducting (transmon) qubit is a superposition of the lowest two energy levels of a charge oscillation (an "artificial atom") across a nonlinear inductive tunnel barrier attached to a capacitive antenna

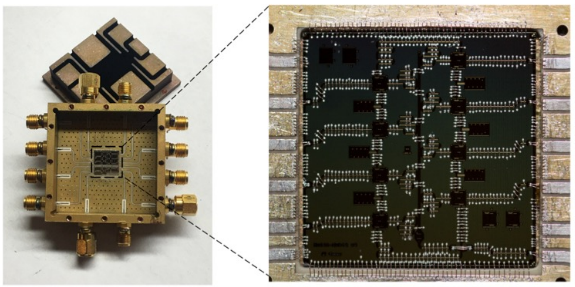


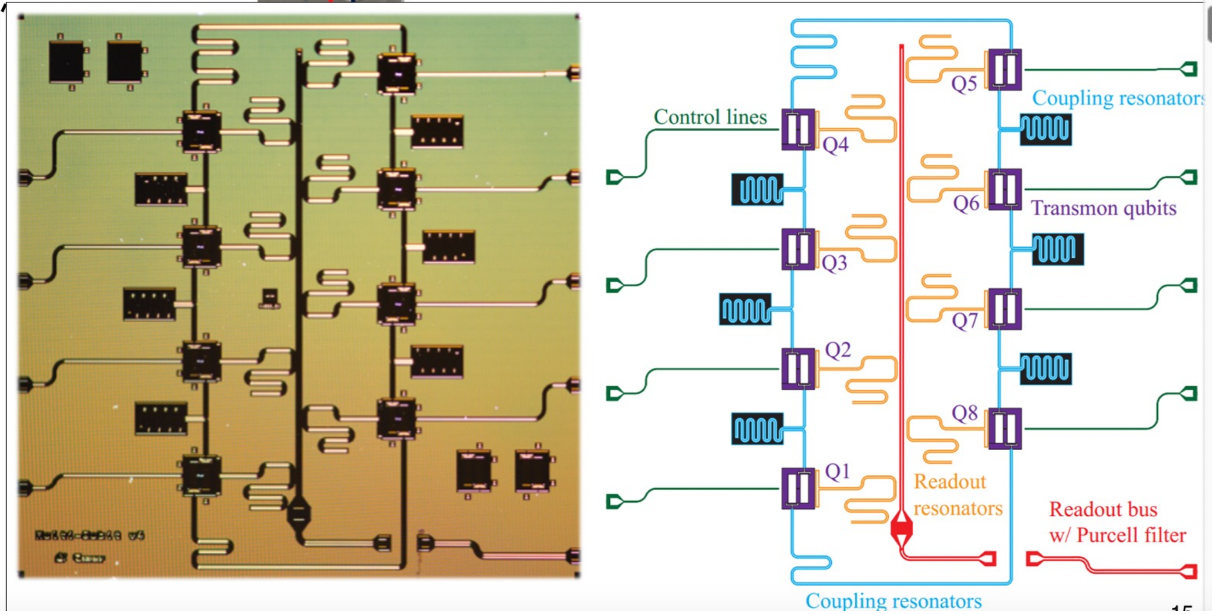


Yale : Transmon SEM

Controlled with all electrical AC signals at microwave frequencies

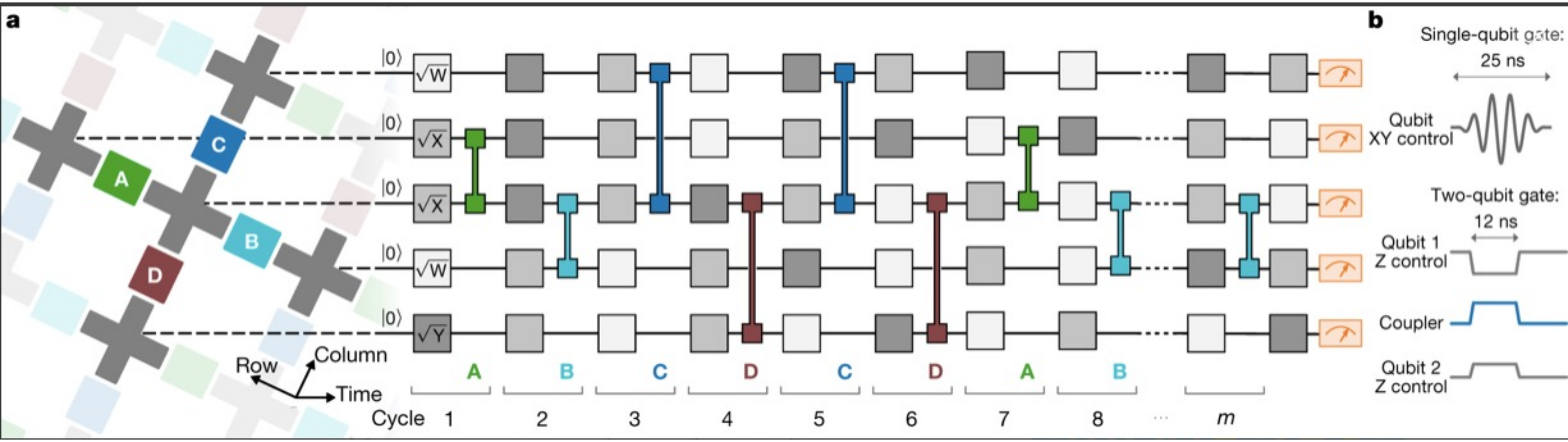
Cooled to mK temperatures





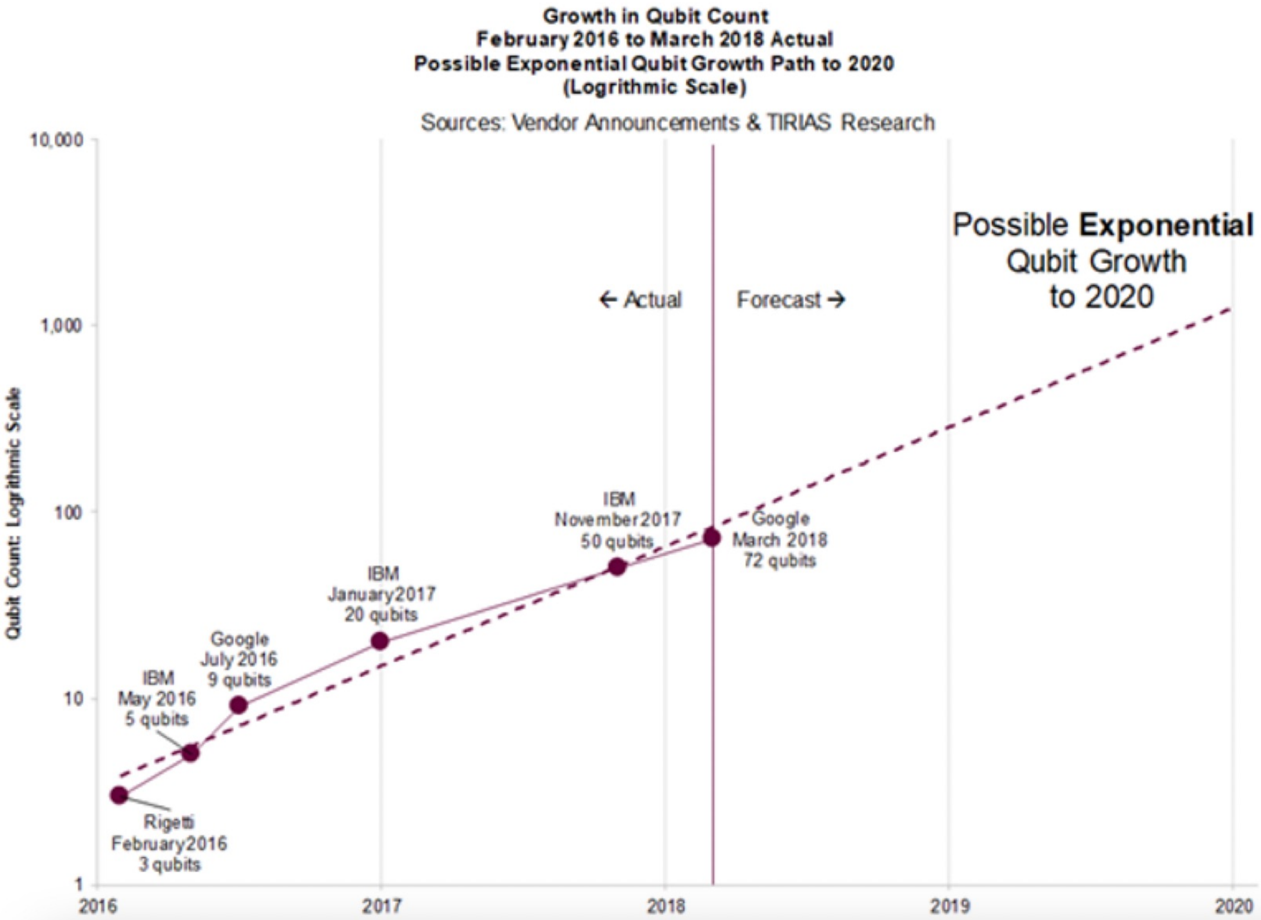
UC Berkeley : 8 qubit chip

Qubits


a
A
B
C
D
Row
Column
Time
|0⟩
√W
√X
√X
√W
√Y
A
B
C
D
C
D
A
B
Cycle 1
2
3
4
5
6
7
8
···
m
b
Single-qubit gate:
25 ns
Qubit
XY control
Two-qubit gate:
12 ns
Qubit 1
Z control
Coupler
Qubit 2
Z control

QC's & Qubits

How Long Until A Billion Qubits?



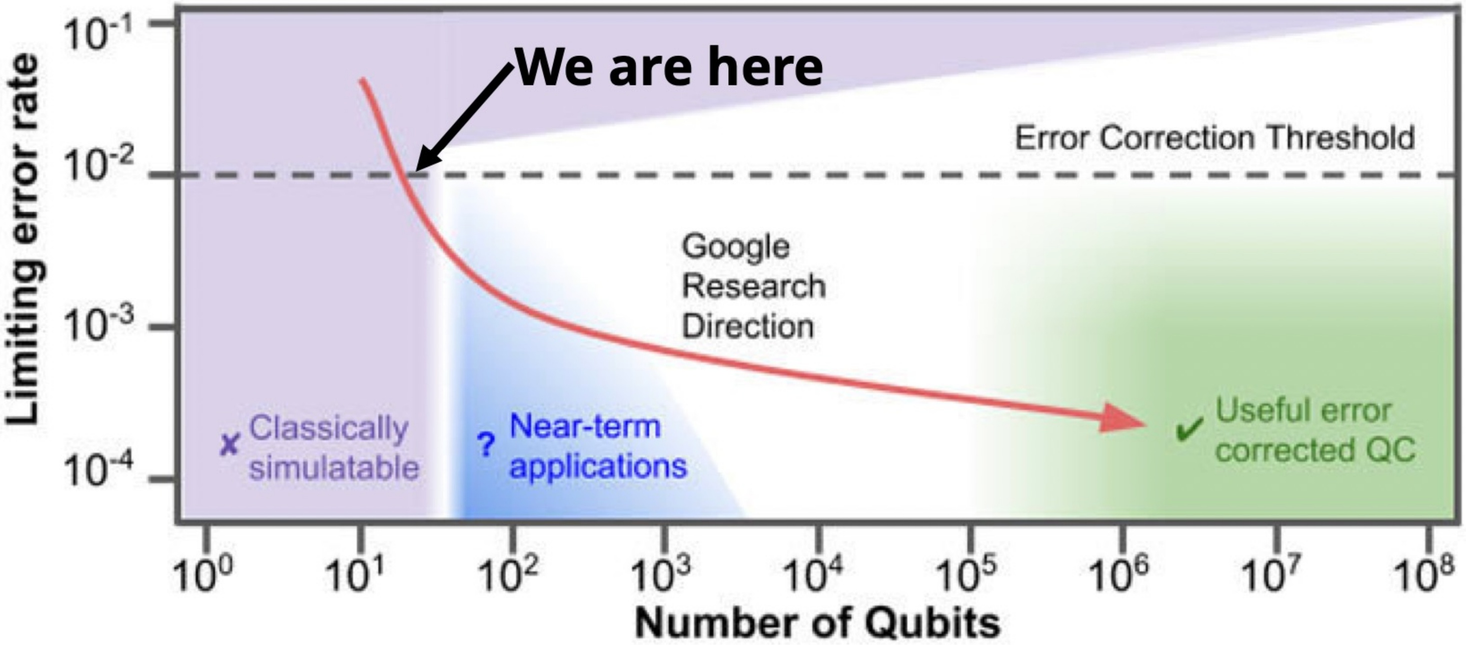
Growth in qubit number is currently **exponential**

If growth continues exponentially (with both fidelity and technical substrate scaling favorably) then we can expect chips with one billion qubits in:

~10-15 years

QC’s & Qubits

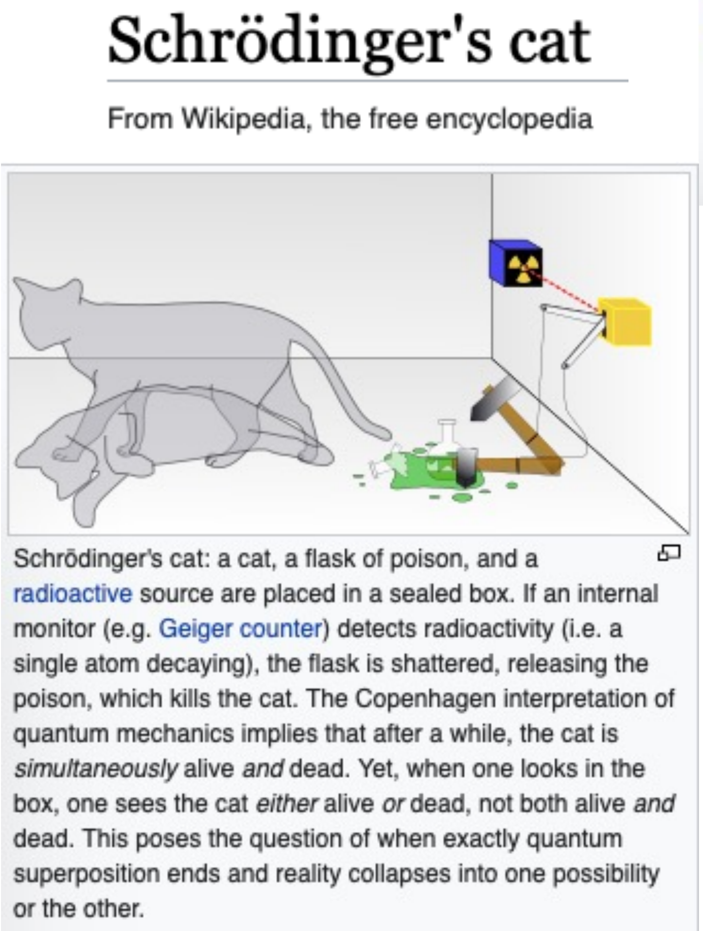
What can we do until then?



We are now reaching the scale that is no longer possible to simulate using classical supercomputers.

The current challenge is to find "near-term" applications for the existing quantum devices.

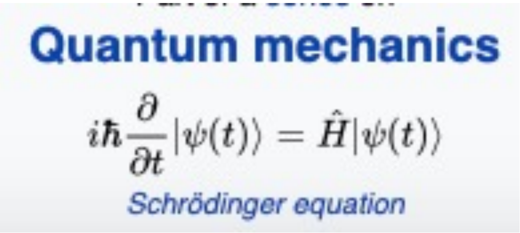
QC: Quantum Mechanics



Schrödinger's cat

From Wikipedia, the free encyclopedia

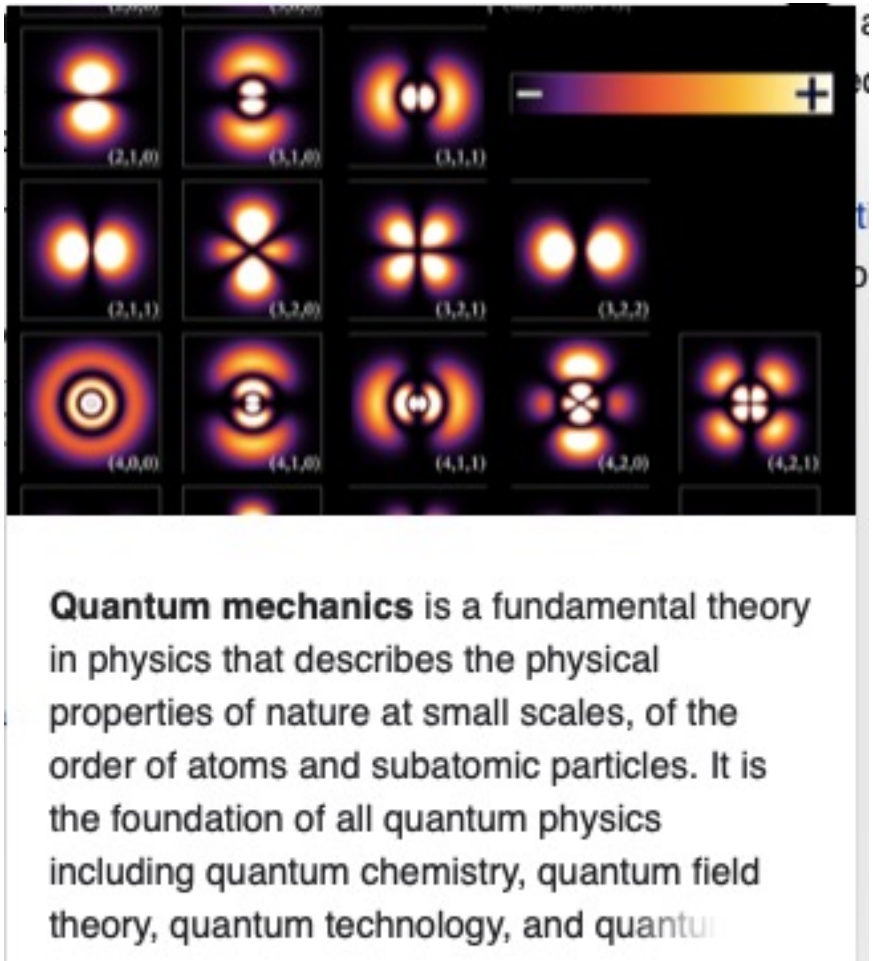
Schrödinger's cat: a cat, a flask of poison, and a radioactive source are placed in a sealed box. If an internal monitor (e.g. Geiger counter) detects radioactivity (i.e. a single atom decaying), the flask is shattered, releasing the poison, which kills the cat. The Copenhagen interpretation of quantum mechanics implies that after a while, the cat is *simultaneously* alive *and* dead. Yet, when one looks in the box, one sees the cat *either* alive *or* dead, not both alive *and* dead. This poses the question of when exactly quantum superposition ends and reality collapses into one possibility or the other.



Quantum mechanics

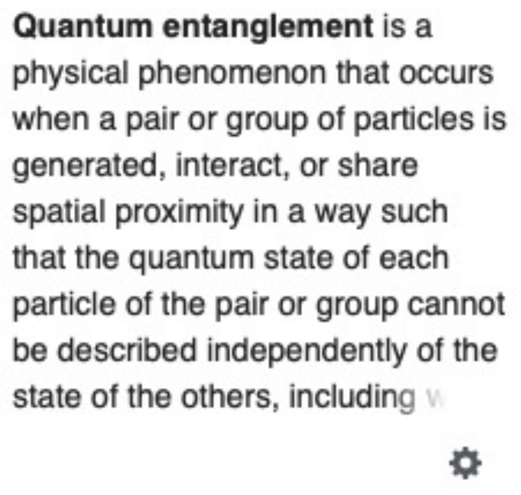
$$i\hbar\frac{\partial}{\partial t}|\psi(t)\rangle = \hat{H}|\psi(t)\rangle$$

Schrödinger equation

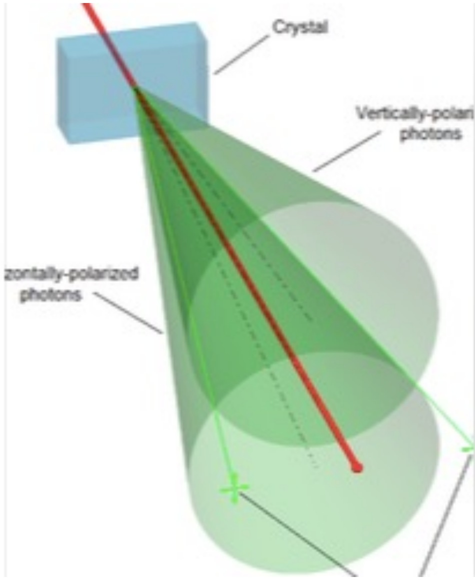


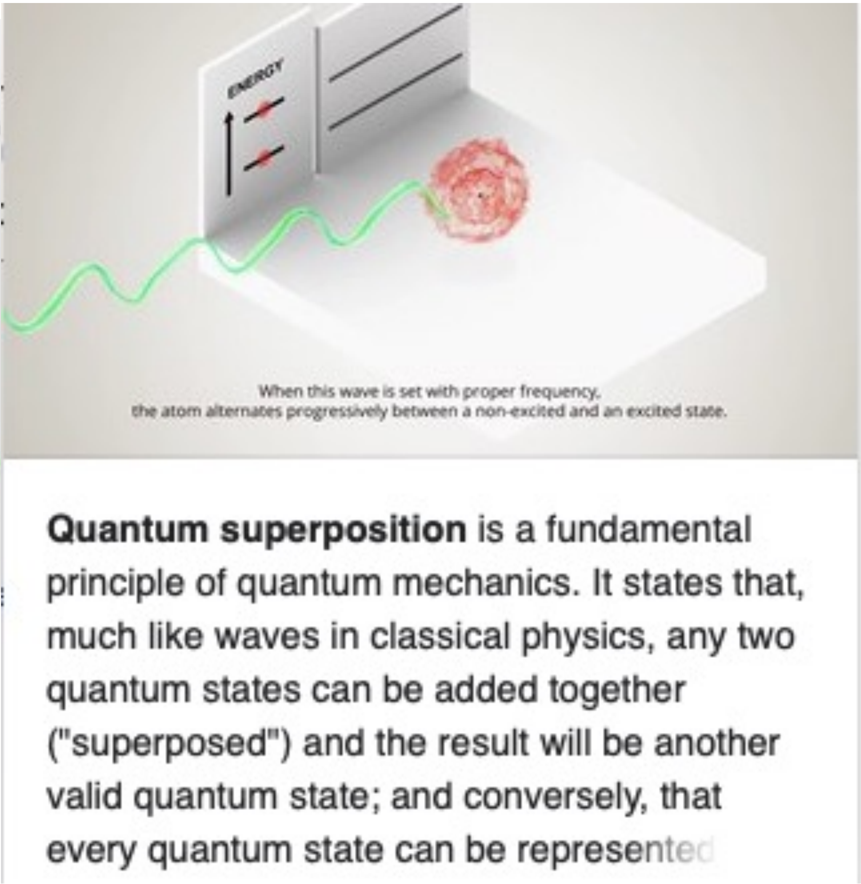
Quantum mechanics is a fundamental theory in physics that describes the physical properties of nature at small scales, of the order of atoms and subatomic particles. It is the foundation of all quantum physics including quantum chemistry, quantum field theory, quantum technology, and quantu

QC: Quantum Mechanics



Quantum entanglement is a physical phenomenon that occurs when a pair or group of particles is generated, interact, or share spatial proximity in a way such that the quantum state of each particle of the pair or group cannot be described independently of the state of the others, including w





When this wave is set with proper frequency,
the atom alternates progressively between a non-excited and an excited state.

Quantum superposition is a fundamental principle of quantum mechanics. It states that, much like waves in classical physics, any two quantum states can be added together ("superposed") and the result will be another valid quantum state; and conversely, that every quantum state can be represente

QC: Spinions & Chargons

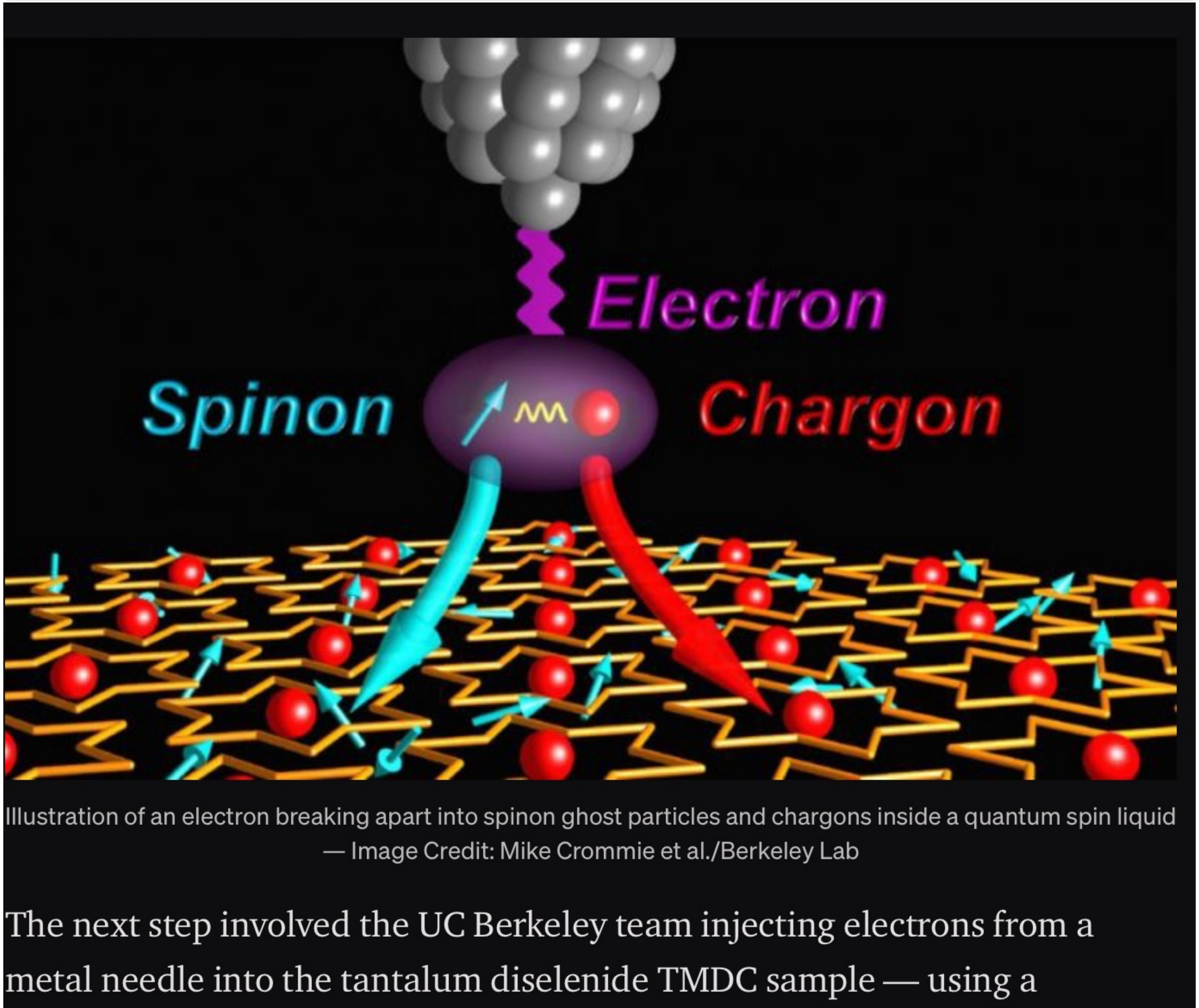
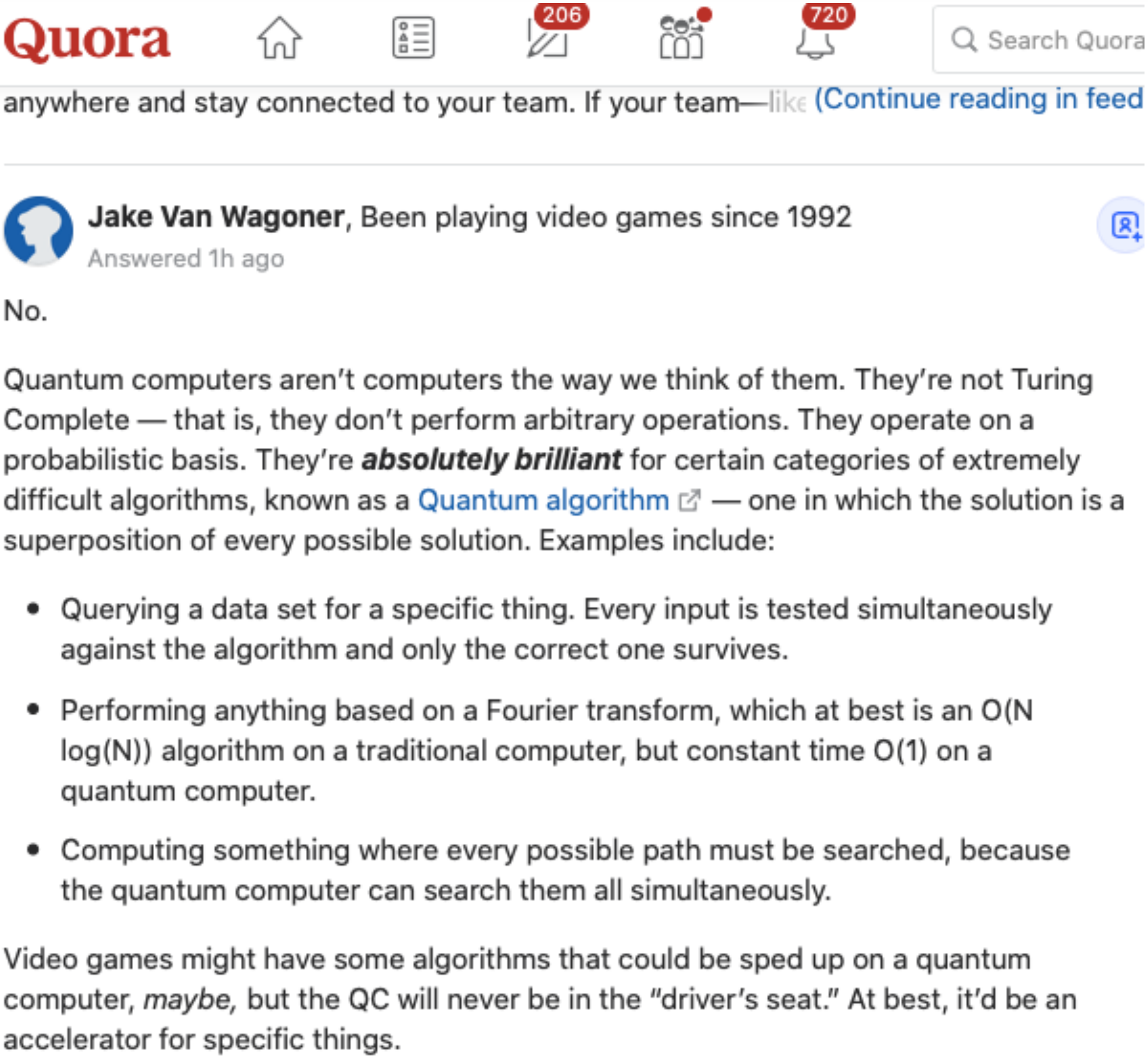


Illustration of an electron breaking apart into spinon ghost particles and chargons inside a quantum spin liquid — Image Credit: Mike Crommie et al./Berkeley Lab

The next step involved the UC Berkeley team injecting electrons from a metal needle into the tantalum diselenide TMDC sample — using a

QC’s



anywhere and stay connected to your team. If your team—like (Continue reading in feed

Jake Van Wagoner, Been playing video games since 1992
Answered 1h ago

No.

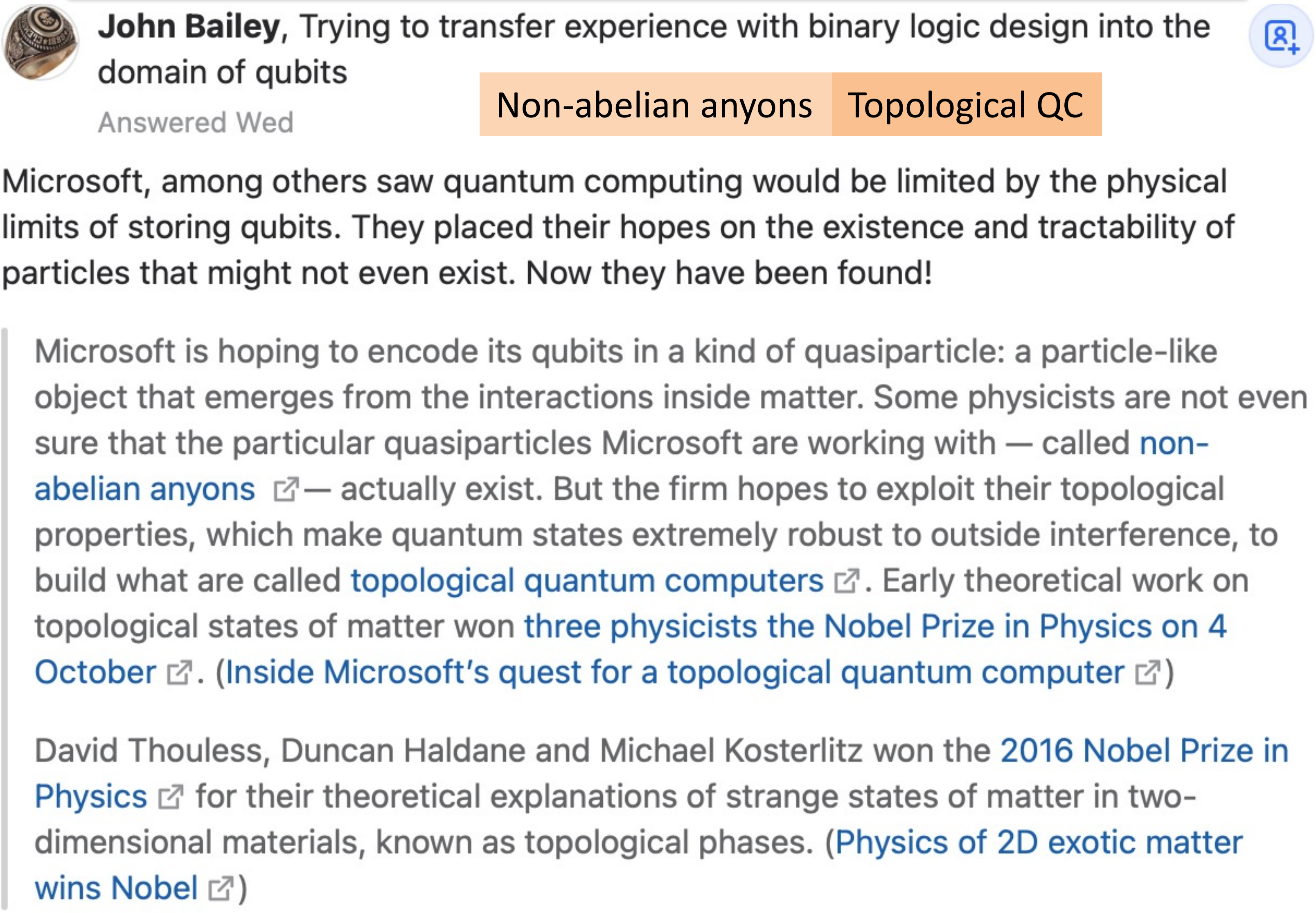
Quantum computers aren’t computers the way we think of them. They’re not Turing Complete — that is, they don’t perform arbitrary operations. They operate on a probabilistic basis. They’re ***absolutely brilliant*** for certain categories of extremely difficult algorithms, known as a Quantum algorithm — one in which the solution is a superposition of every possible solution. Examples include:

- Querying a data set for a specific thing. Every input is tested simultaneously against the algorithm and only the correct one survives.
- Performing anything based on a Fourier transform, which at best is an O(N log(N)) algorithm on a traditional computer, but constant time O(1) on a quantum computer.
- Computing something where every possible path must be searched, because the quantum computer can search them all simultaneously.

Video games might have some algorithms that could be sped up on a quantum computer, *maybe*, but the QC will never be in the “driver’s seat.” At best, it’d be an accelerator for specific things.

QC's

Quora



Non-abelian anyons

Topological QC

John Bailey, Trying to transfer experience with binary logic design into the domain of qubits

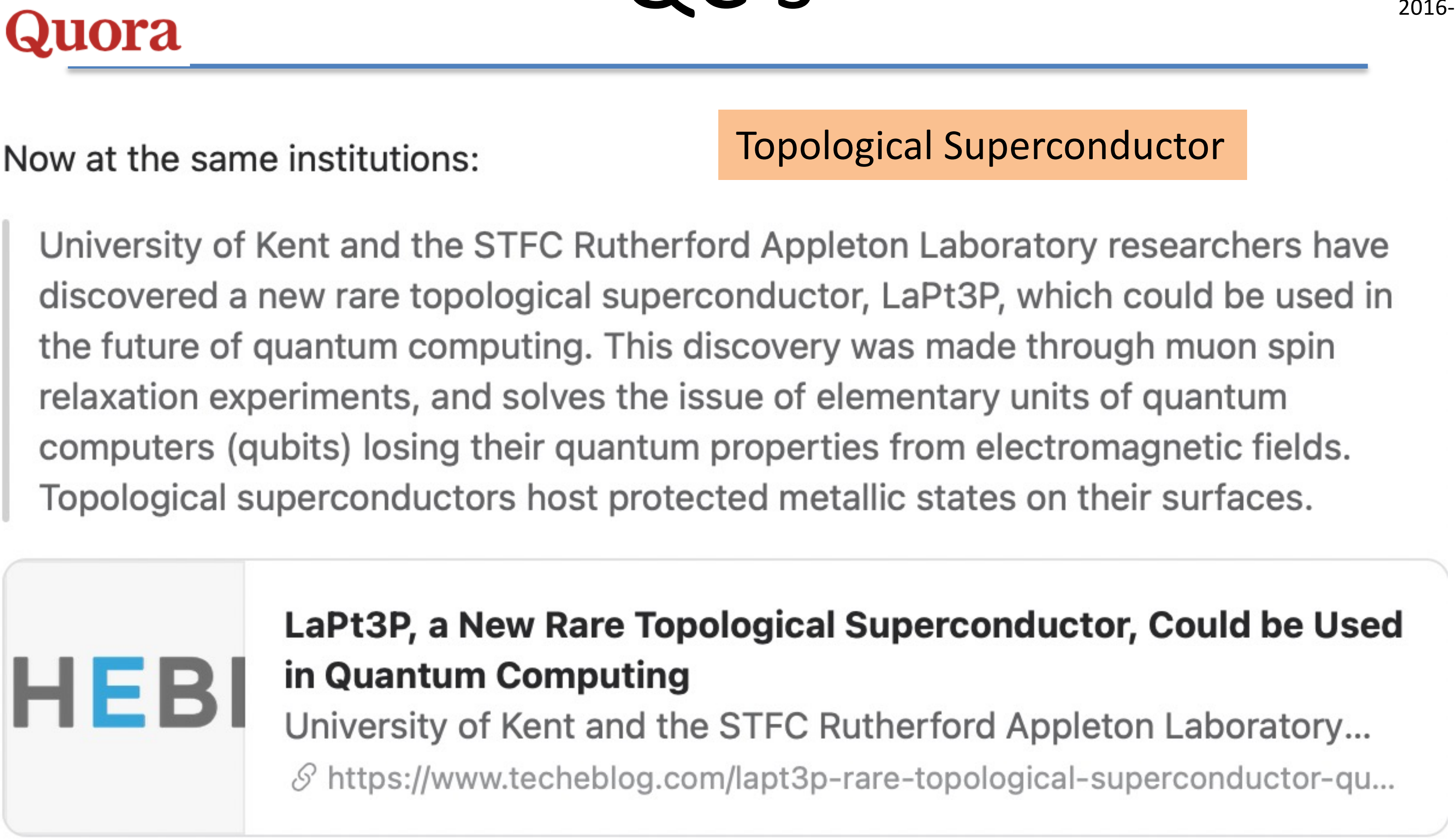
Answered Wed

Microsoft, among others saw quantum computing would be limited by the physical limits of storing qubits. They placed their hopes on the existence and tractability of particles that might not even exist. Now they have been found!

> Microsoft is hoping to encode its qubits in a kind of quasiparticle: a particle-like object that emerges from the interactions inside matter. Some physicists are not even sure that the particular quasiparticles Microsoft are working with — called non-abelian anyons — actually exist. But the firm hopes to exploit their topological properties, which make quantum states extremely robust to outside interference, to build what are called topological quantum computers. Early theoretical work on topological states of matter won three physicists the Nobel Prize in Physics on 4 October. (Inside Microsoft's quest for a topological quantum computer)
>
> David Thouless, Duncan Haldane and Michael Kosterlitz won the 2016 Nobel Prize in Physics for their theoretical explanations of strange states of matter in two-dimensional materials, known as topological phases. (Physics of 2D exotic matter wins Nobel)

QC's

Topological Superconductor



Quora

Now at the same institutions:

> University of Kent and the STFC Rutherford Appleton Laboratory researchers have discovered a new rare topological superconductor, LaPt3P, which could be used in the future of quantum computing. This discovery was made through muon spin relaxation experiments, and solves the issue of elementary units of quantum computers (qubits) losing their quantum properties from electromagnetic fields. Topological superconductors host protected metallic states on their surfaces.

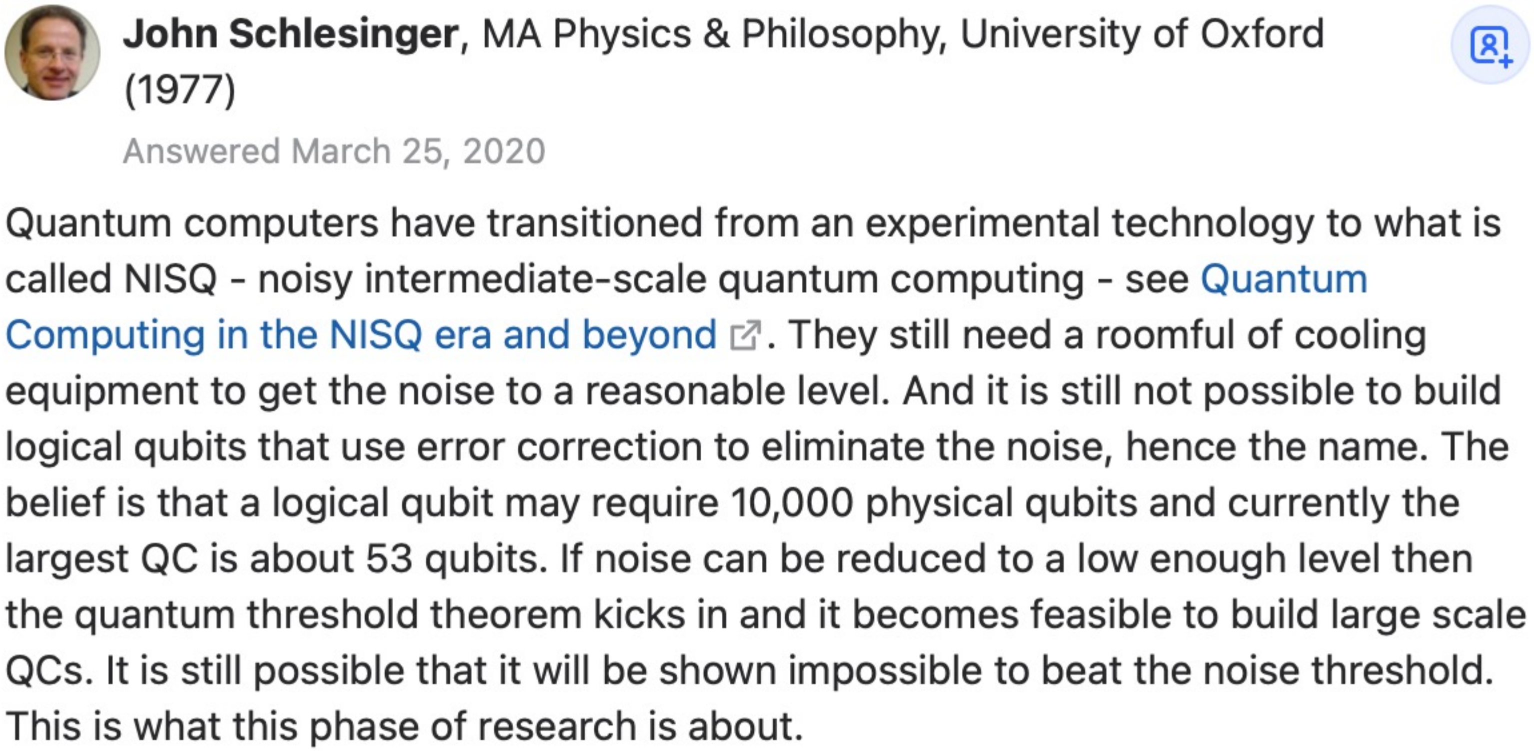
LaPt3P, a New Rare Topological Superconductor, Could be Used in Quantum Computing

University of Kent and the STFC Rutherford Appleton Laboratory...

https://www.techeblog.com/lapt3p-rare-topological-superconductor-qu...

QC’s

Quora



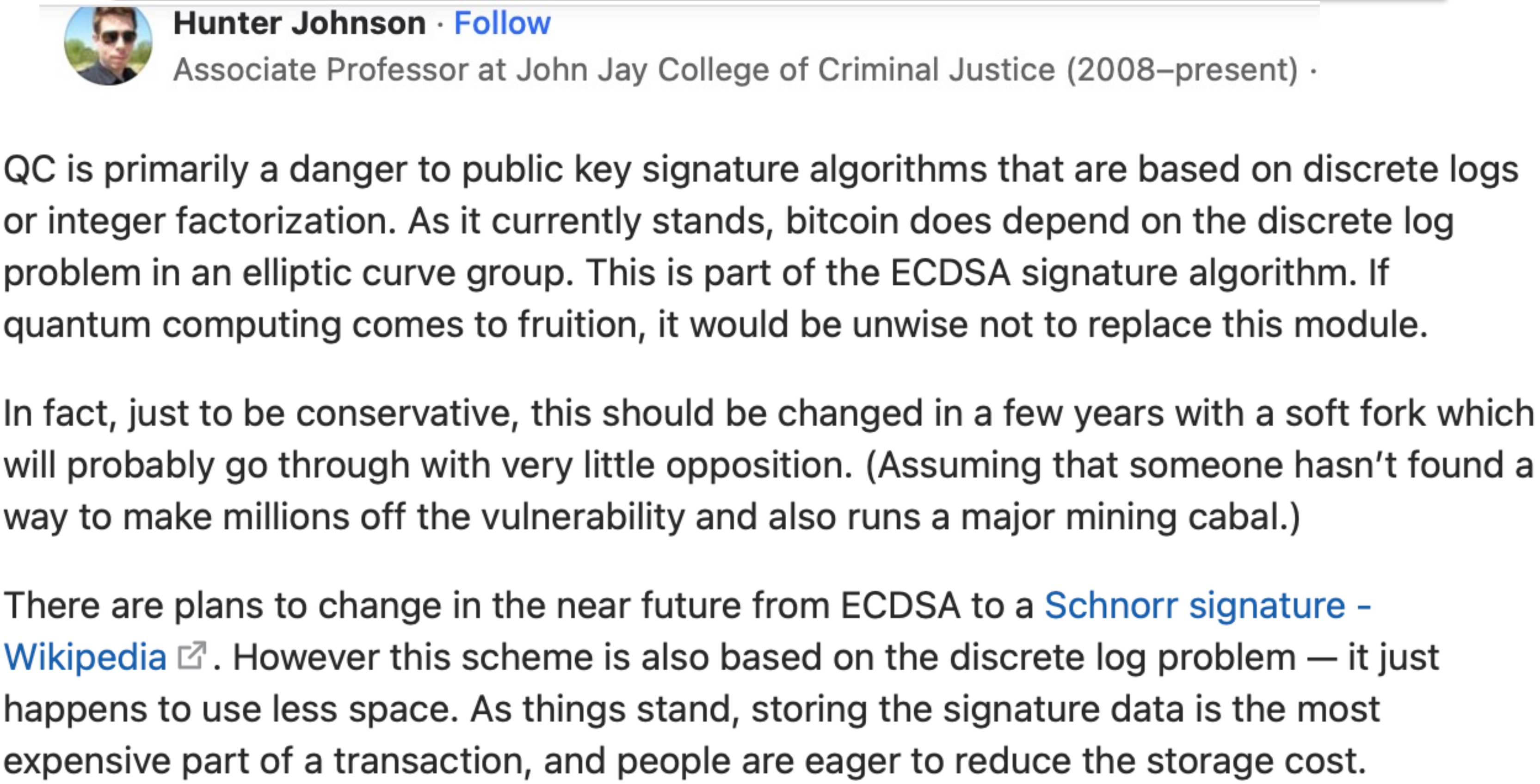
John Schlesinger, MA Physics & Philosophy, University of Oxford (1977)

Answered March 25, 2020

Quantum computers have transitioned from an experimental technology to what is called NISQ - noisy intermediate-scale quantum computing - see Quantum Computing in the NISQ era and beyond. They still need a roomful of cooling equipment to get the noise to a reasonable level. And it is still not possible to build logical qubits that use error correction to eliminate the noise, hence the name. The belief is that a logical qubit may require 10,000 physical qubits and currently the largest QC is about 53 qubits. If noise can be reduced to a low enough level then the quantum threshold theorem kicks in and it becomes feasible to build large scale QCs. It is still possible that it will be shown impossible to beat the noise threshold. This is what this phase of research is about.

QC’s

Quora



Hunter Johnson · Follow

Associate Professor at John Jay College of Criminal Justice (2008–present) ·

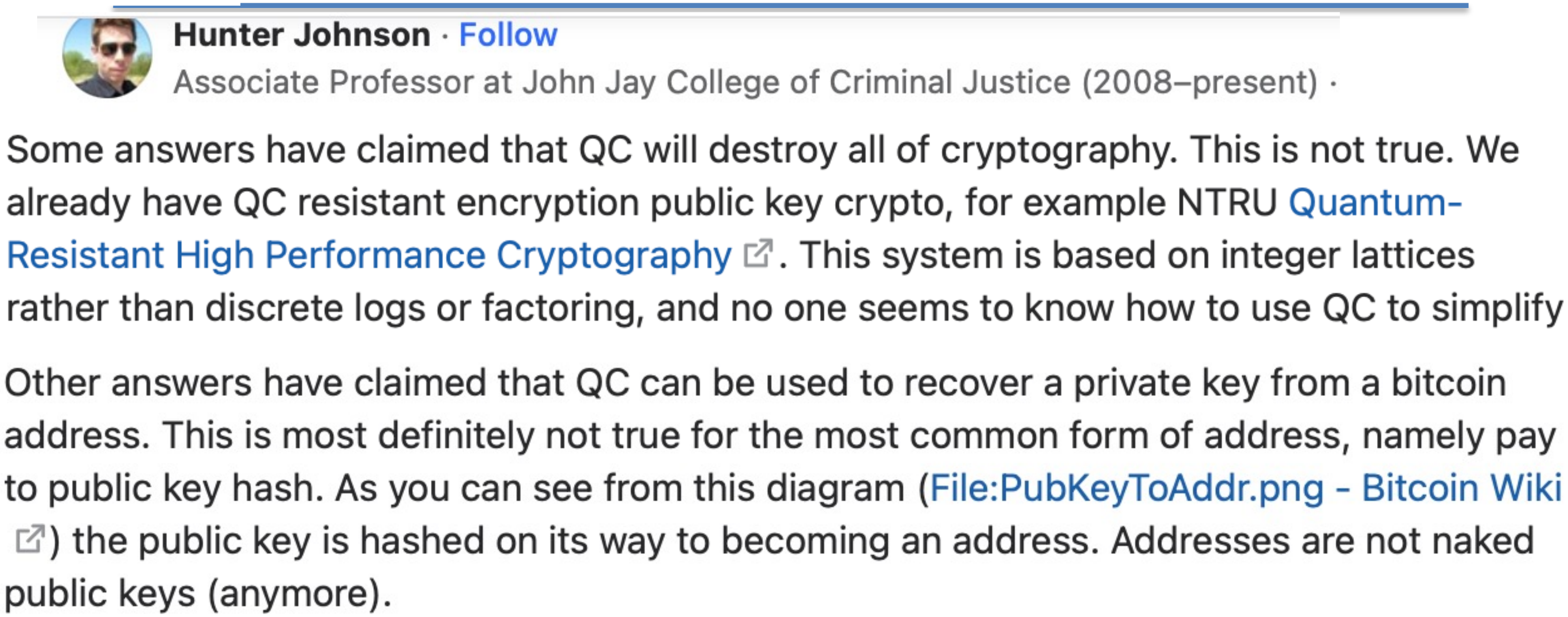
QC is primarily a danger to public key signature algorithms that are based on discrete logs or integer factorization. As it currently stands, bitcoin does depend on the discrete log problem in an elliptic curve group. This is part of the ECDSA signature algorithm. If quantum computing comes to fruition, it would be unwise not to replace this module.

In fact, just to be conservative, this should be changed in a few years with a soft fork which will probably go through with very little opposition. (Assuming that someone hasn’t found a way to make millions off the vulnerability and also runs a major mining cabal.)

There are plans to change in the near future from ECDSA to a Schnorr signature - Wikipedia. However this scheme is also based on the discrete log problem — it just happens to use less space. As things stand, storing the signature data is the most expensive part of a transaction, and people are eager to reduce the storage cost.

QC’s

Quora



Hunter Johnson · Follow

Associate Professor at John Jay College of Criminal Justice (2008–present) ·

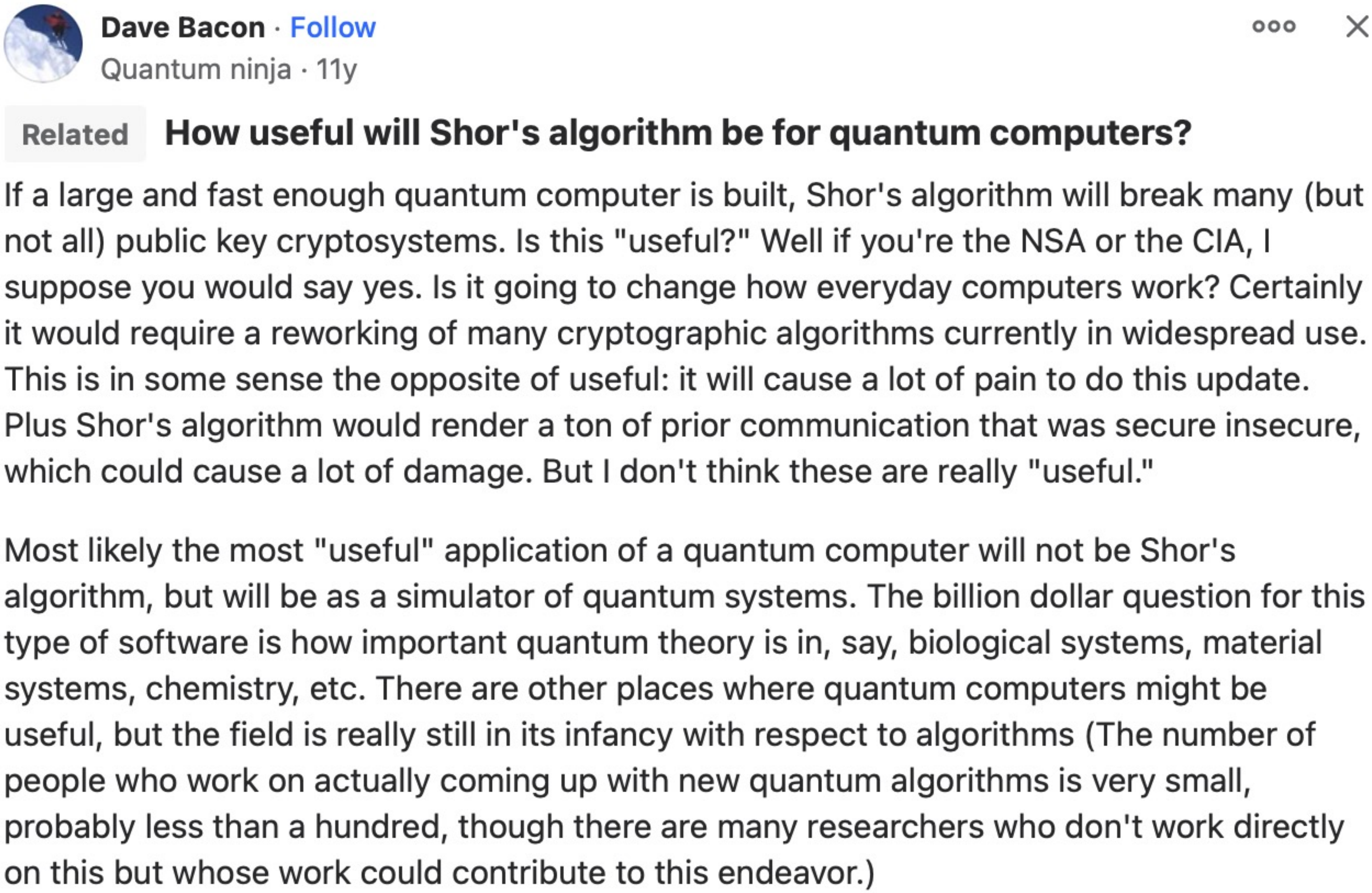
Some answers have claimed that QC will destroy all of cryptography. This is not true. We already have QC resistant encryption public key crypto, for example NTRU Quantum-Resistant High Performance Cryptography. This system is based on integer lattices rather than discrete logs or factoring, and no one seems to know how to use QC to simplify

Other answers have claimed that QC can be used to recover a private key from a bitcoin address. This is most definitely not true for the most common form of address, namely pay to public key hash. As you can see from this diagram (File:PubKeyToAddr.png - Bitcoin Wiki) the public key is hashed on its way to becoming an address. Addresses are not naked public keys (anymore).

QC's: Shor's Alg

DR JEFF SOFTWARE
INDIE APP DEVELOPER
© Jeff Drobman 2016-23

Quora



Dave Bacon · Follow
Quantum ninja · 11y

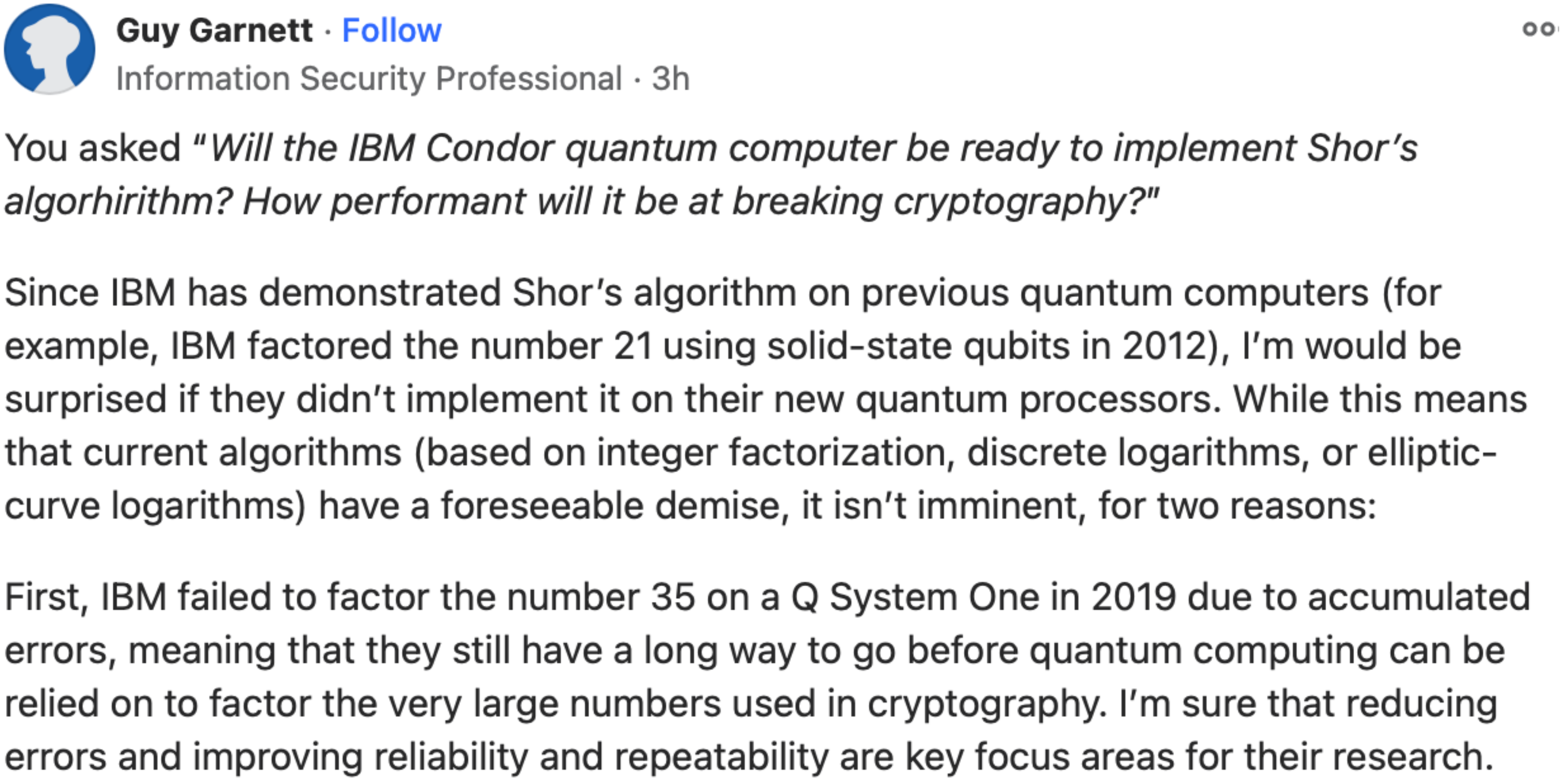
Related **How useful will Shor's algorithm be for quantum computers?**

If a large and fast enough quantum computer is built, Shor's algorithm will break many (but not all) public key cryptosystems. Is this "useful?" Well if you're the NSA or the CIA, I suppose you would say yes. Is it going to change how everyday computers work? Certainly it would require a reworking of many cryptographic algorithms currently in widespread use. This is in some sense the opposite of useful: it will cause a lot of pain to do this update. Plus Shor's algorithm would render a ton of prior communication that was secure insecure, which could cause a lot of damage. But I don't think these are really "useful."

Most likely the most "useful" application of a quantum computer will not be Shor's algorithm, but will be as a simulator of quantum systems. The billion dollar question for this type of software is how important quantum theory is in, say, biological systems, material systems, chemistry, etc. There are other places where quantum computers might be useful, but the field is really still in its infancy with respect to algorithms (The number of people who work on actually coming up with new quantum algorithms is very small, probably less than a hundred, though there are many researchers who don't work directly on this but whose work could contribute to this endeavor.)

QC's: Shor's Alg

Quora



Guy Garnett · Follow

Information Security Professional · 3h

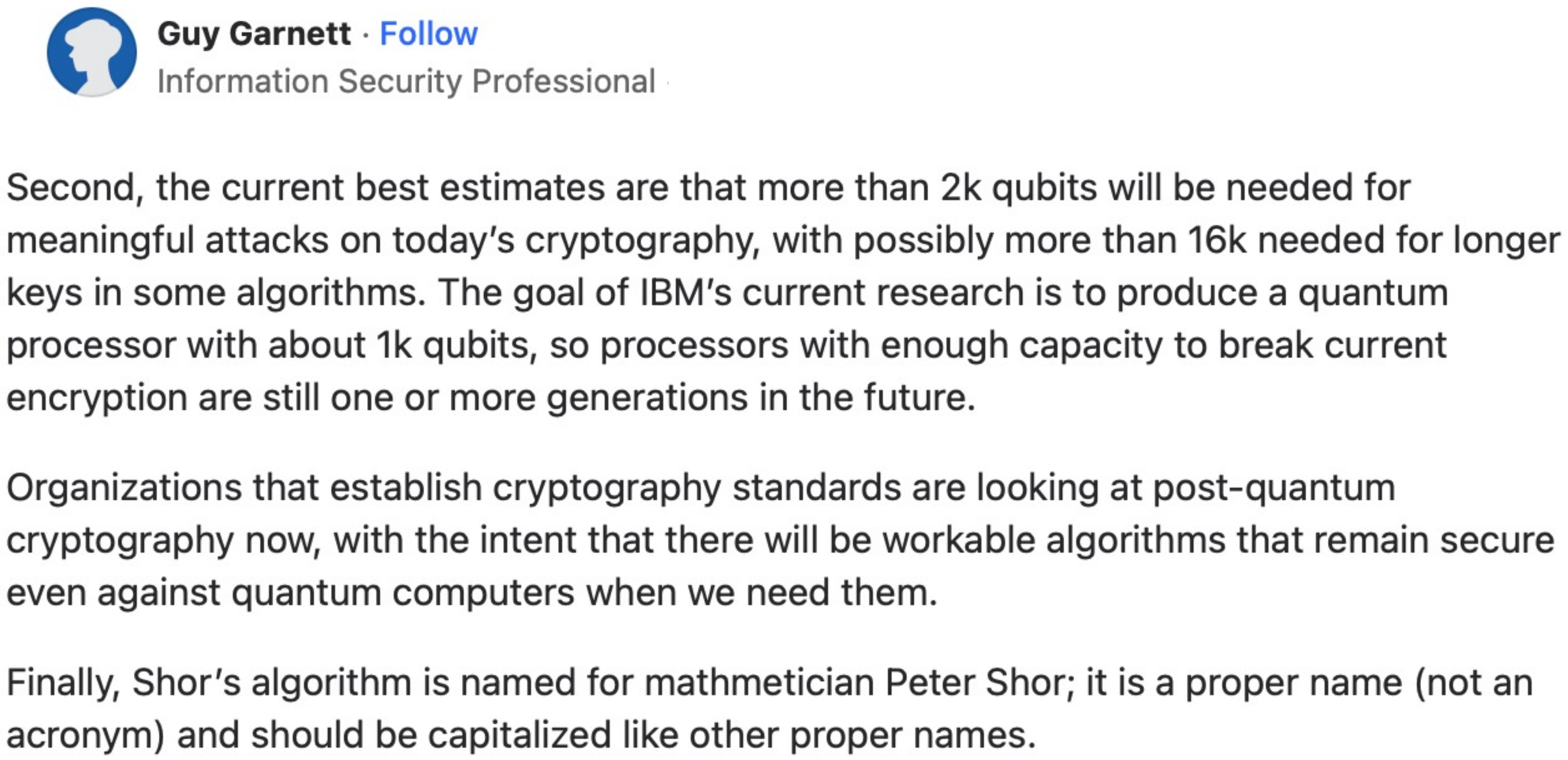
You asked "*Will the IBM Condor quantum computer be ready to implement Shor's algorhirithm? How performant will it be at breaking cryptography?*"

Since IBM has demonstrated Shor's algorithm on previous quantum computers (for example, IBM factored the number 21 using solid-state qubits in 2012), I'm would be surprised if they didn't implement it on their new quantum processors. While this means that current algorithms (based on integer factorization, discrete logarithms, or elliptic-curve logarithms) have a foreseeable demise, it isn't imminent, for two reasons:

First, IBM failed to factor the number 35 on a Q System One in 2019 due to accumulated errors, meaning that they still have a long way to go before quantum computing can be relied on to factor the very large numbers used in cryptography. I'm sure that reducing errors and improving reliability and repeatability are key focus areas for their research.

QC's: Shor's Alg

Quora



Guy Garnett · Follow
Information Security Professional

Second, the current best estimates are that more than 2k qubits will be needed for meaningful attacks on today's cryptography, with possibly more than 16k needed for longer keys in some algorithms. The goal of IBM's current research is to produce a quantum processor with about 1k qubits, so processors with enough capacity to break current encryption are still one or more generations in the future.

Organizations that establish cryptography standards are looking at post-quantum cryptography now, with the intent that there will be workable algorithms that remain secure even against quantum computers when we need them.

Finally, Shor's algorithm is named for mathmetician Peter Shor; it is a proper name (not an acronym) and should be capitalized like other proper names.

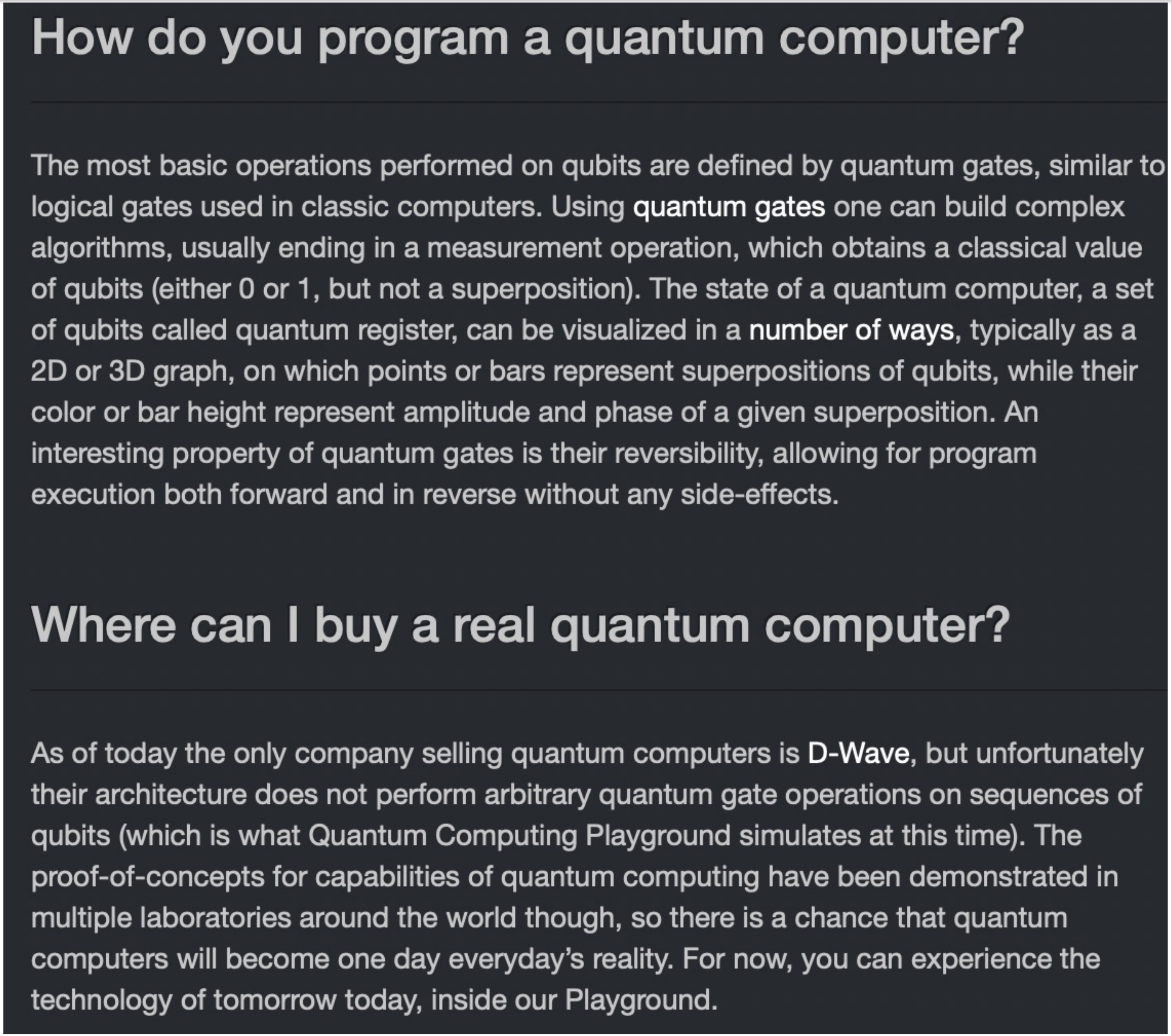
Software

DR JEFF SOFTWARE INDIE APP DEVELOPER

Programming

QC's

Quantum Computing



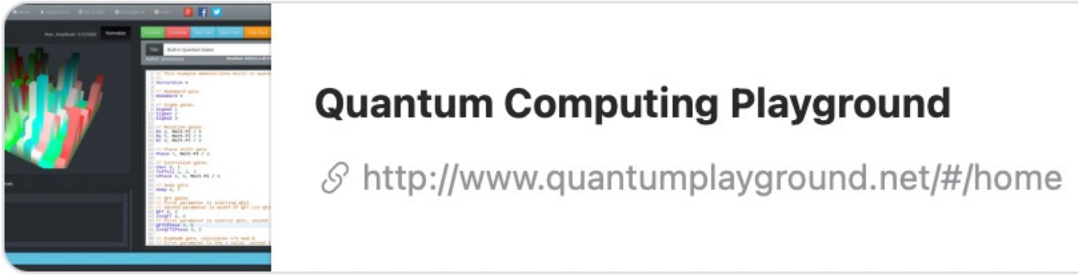
How do you program a quantum computer?

The most basic operations performed on qubits are defined by quantum gates, similar to logical gates used in classic computers. Using **quantum gates** one can build complex algorithms, usually ending in a measurement operation, which obtains a classical value of qubits (either 0 or 1, but not a superposition). The state of a quantum computer, a set of qubits called quantum register, can be visualized in a **number of ways**, typically as a 2D or 3D graph, on which points or bars represent superpositions of qubits, while their color or bar height represent amplitude and phase of a given superposition. An interesting property of quantum gates is their reversibility, allowing for program execution both forward and in reverse without any side-effects.

Where can I buy a real quantum computer?

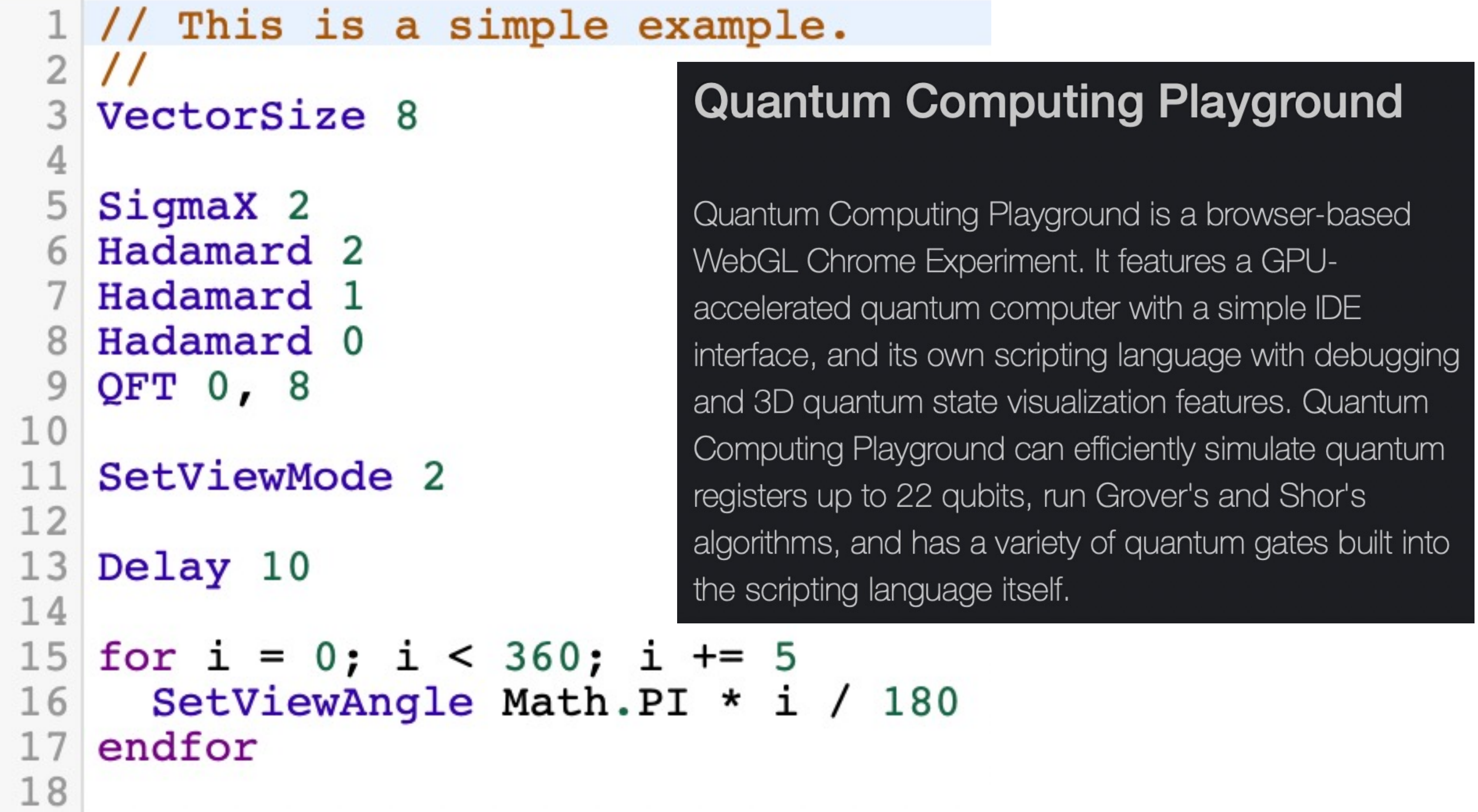
As of today the only company selling quantum computers is **D-Wave**, but unfortunately their architecture does not perform arbitrary quantum gate operations on sequences of qubits (which is what Quantum Computing Playground simulates at this time). The proof-of-concepts for capabilities of quantum computing have been demonstrated in multiple laboratories around the world though, so there is a chance that quantum computers will become one day everyday’s reality. For now, you can experience the technology of tomorrow today, inside our Playground.

Quantum Computing



Quantum Computing Playground

http://www.quantumplayground.net/#/home



```
// This is a simple example.
//
VectorSize 8

SigmaX 2
Hadamard 2
Hadamard 1
Hadamard 0
QFT 0, 8

SetViewMode 2

Delay 10

for i = 0; i < 360; i += 5
  SetViewAngle Math.PI * i / 180
endfor

```

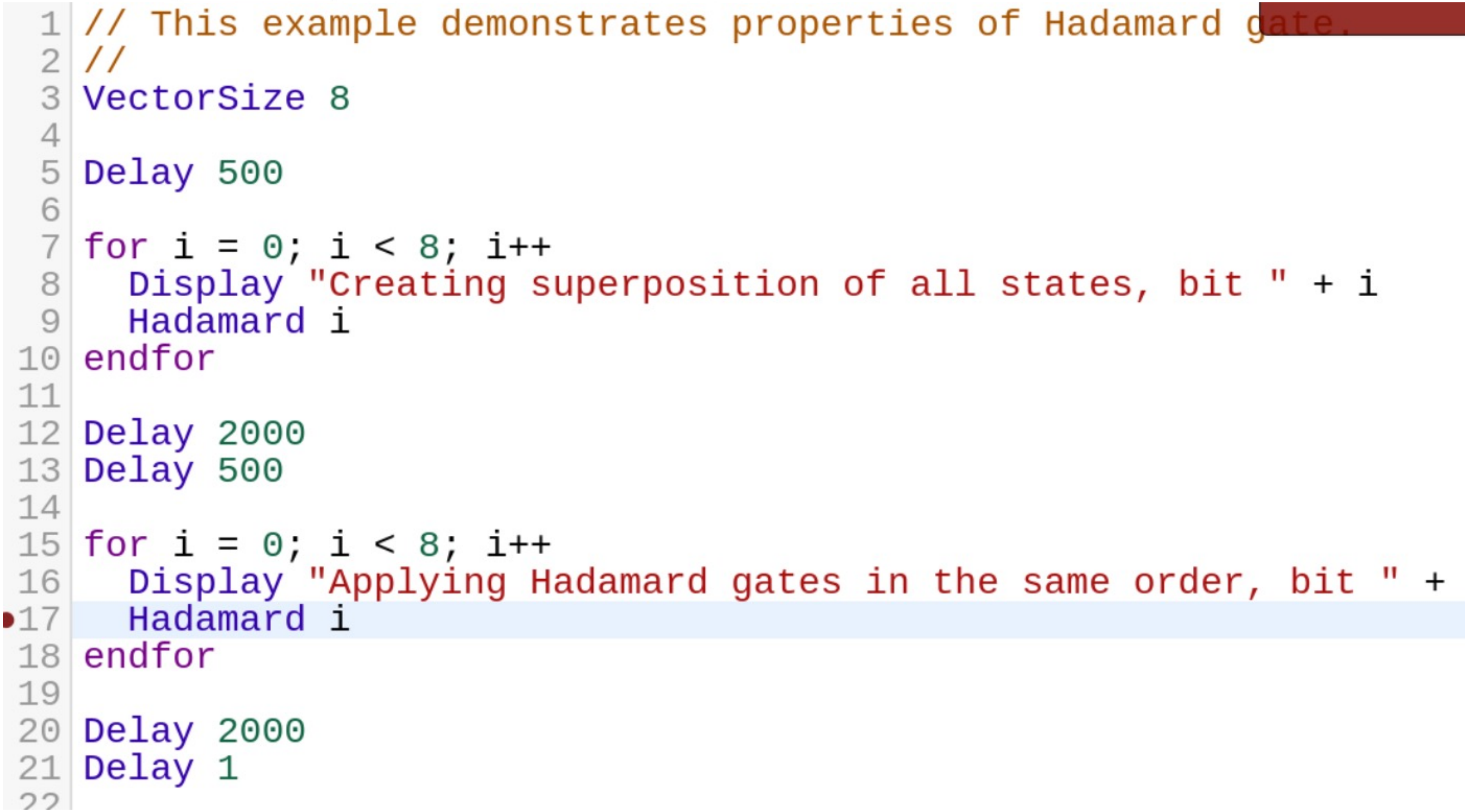
Quantum Computing Playground

Quantum Computing Playground is a browser-based WebGL Chrome Experiment. It features a GPU-accelerated quantum computer with a simple IDE interface, and its own scripting language with debugging and 3D quantum state visualization features. Quantum Computing Playground can efficiently simulate quantum registers up to 22 qubits, run Grover's and Shor's algorithms, and has a variety of quantum gates built into the scripting language itself.

Quantum Computing

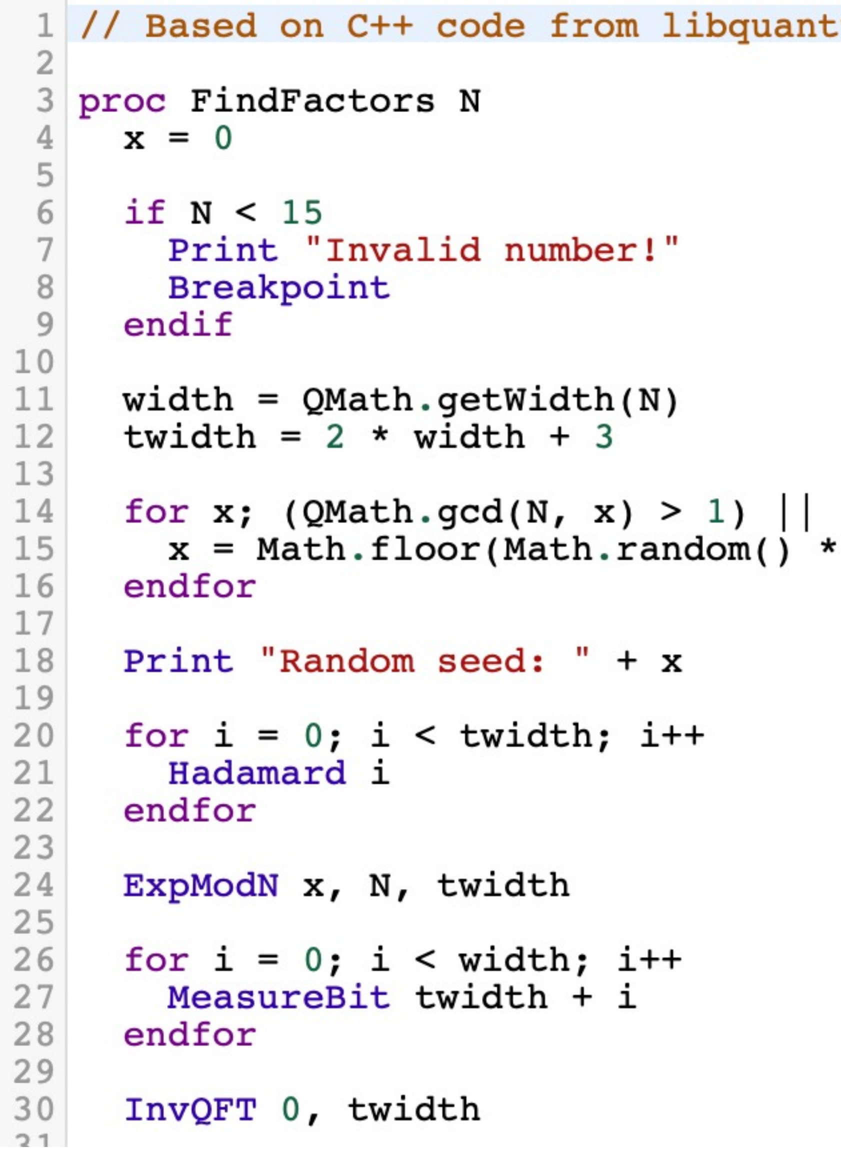
Quantum Computing Playground

http://www.quantumplayground.net/#/home



```
// This example demonstrates properties of Hadamard [redacted]
//
VectorSize 8

Delay 500

for i = 0; i < 8; i++
  Display "Creating superposition of all states, bit " + i
  Hadamard i
endfor

Delay 2000
Delay 500

for i = 0; i < 8; i++
  Display "Applying Hadamard gates in the same order, bit " +
  Hadamard i
endfor

Delay 2000
Delay 1
```

Quantum Computing



```
// Based on C++ code from libquantum library.

proc FindFactors N
  x = 0

  if N < 15
    Print "Invalid number!"
    Breakpoint
  endif

  width = QMath.getWidth(N)
  twidth = 2 * width + 3

  for x; (QMath.gcd(N, x) > 1) || (x < 2); x
    x = Math.floor(Math.random() * 10000) % N
  endfor

  Print "Random seed: " + x

  for i = 0; i < twidth; i++
    Hadamard i
  endfor

  ExpModN x, N, twidth

  for i = 0; i < width; i++
    MeasureBit twidth + i
  endfor

  InvQFT 0, twidth
```

Quantum Computing Playground

http://www.quantumplayground.net/#/home

Hardware

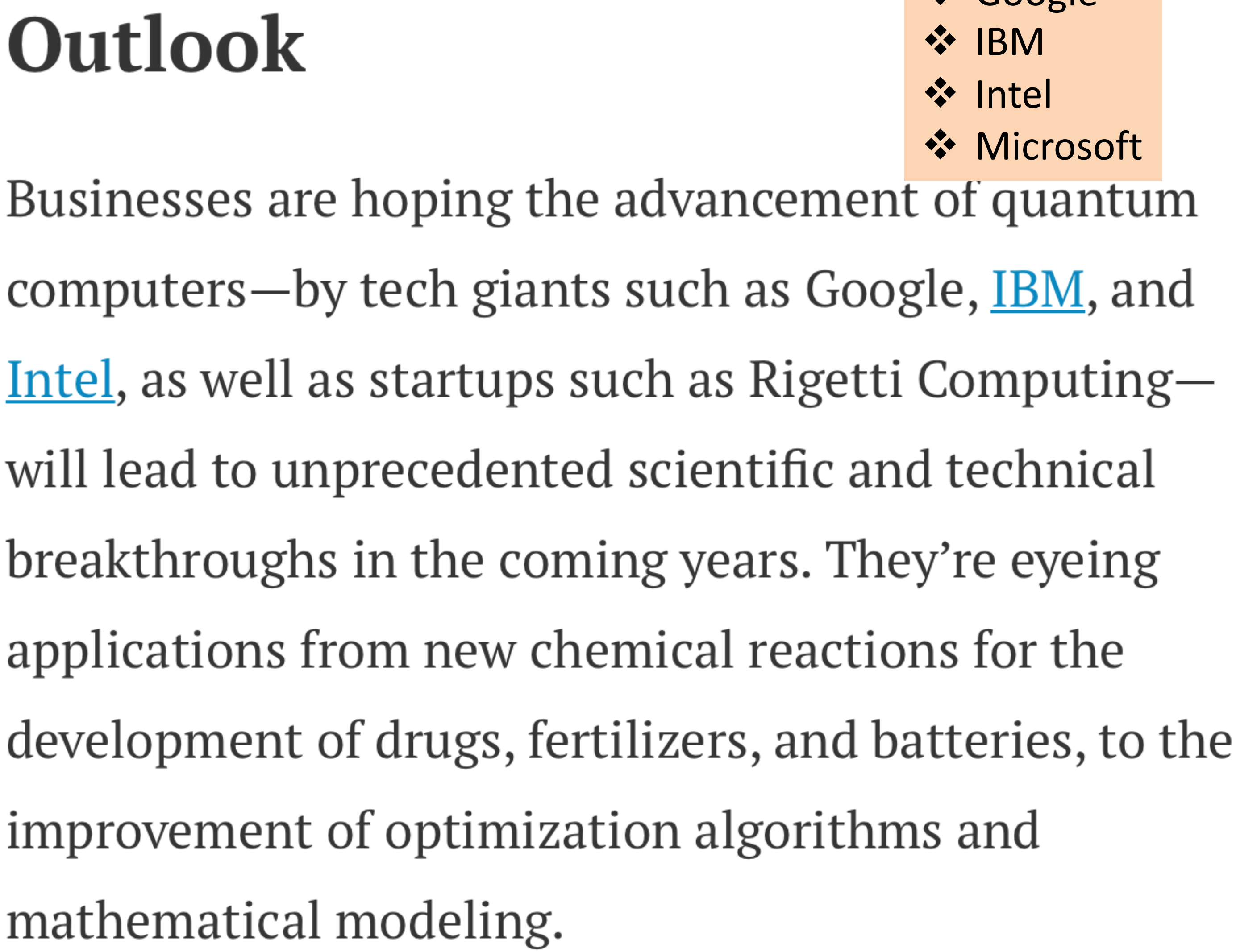
DR JEFF SOFTWARE
INDIE APP DEVELOPER
© Jeff Drobman 2016-23

Quantum

Computers

Top QC Companies

- ❖ Google
- ❖ IBM
- ❖ Intel
- ❖ Microsoft

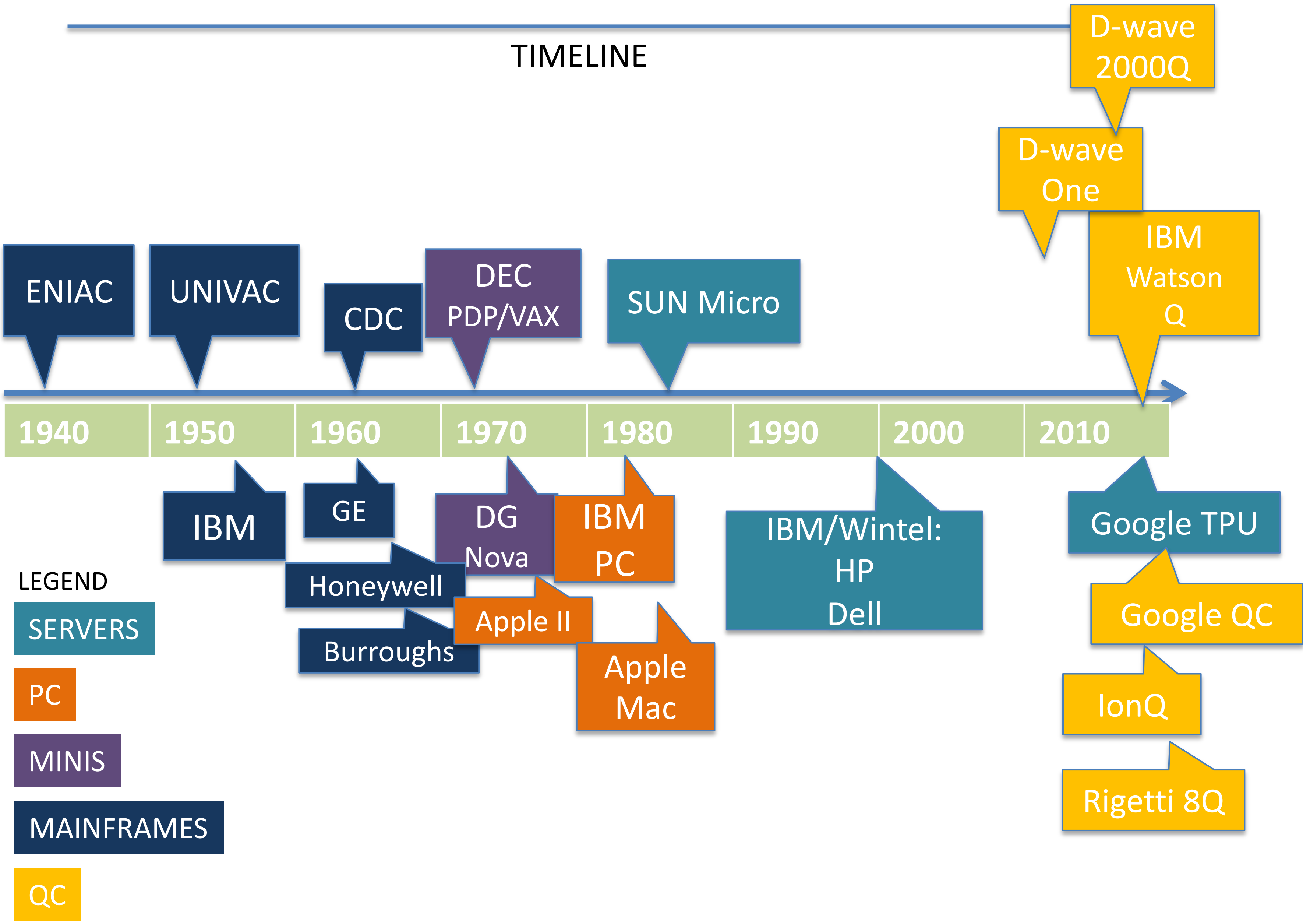


Outlook

Businesses are hoping the advancement of quantum computers—by tech giants such as Google, IBM, and Intel, as well as startups such as Rigetti Computing—will lead to unprecedented scientific and technical breakthroughs in the coming years. They're eyeing applications from new chemical reactions for the development of drugs, fertilizers, and batteries, to the improvement of optimization algorithms and mathematical modeling.

Computers & QC's

DR JEFF SOFTWARE
INDIE APP DEVELOPER



TIMELINE

D-wave 2000Q

D-wave One

IBM Watson Q

ENIAC

UNIVAC

CDC

DEC PDP/VAX

SUN Micro

1940

1950

1960

1970

1980

1990

2000

2010

IBM

GE

Honeywell

Burroughs

DG Nova

IBM PC

Apple II

Apple Mac

IBM/Wintel:
HP
Dell

Google TPU

Google QC

IonQ

Rigetti 8Q

LEGEND

SERVERS

PC

MINIS

MAINFRAMES

QC

Existing QC’s

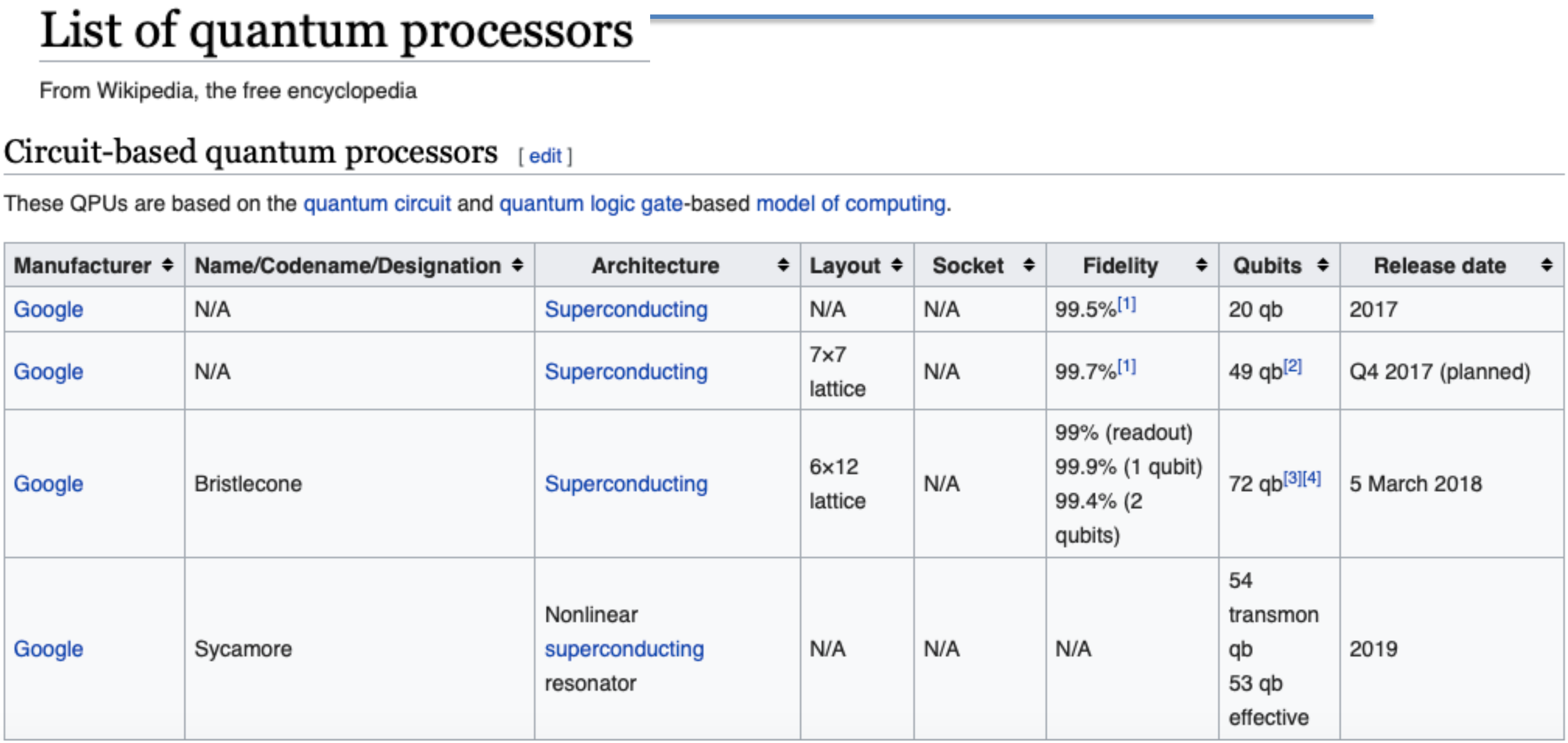
Up until now, there have been several quantum computers built with a range mostly under **100 qubits** or so. As of a little while ago, we also had these operational ***superconducting*** QC’s:

- □ Google *Bristlecone* at 72 qubits, *Sycamore* at 53 qubits
- □ IBM *Q* series up to 53 qubits
- □ Rigetti at 19 qubits
- □ UC Berkeley at 10 qubits

And this *trapped ion* version:

- □ IonQ at 53 qubits

Commercial QC's



List of quantum processors

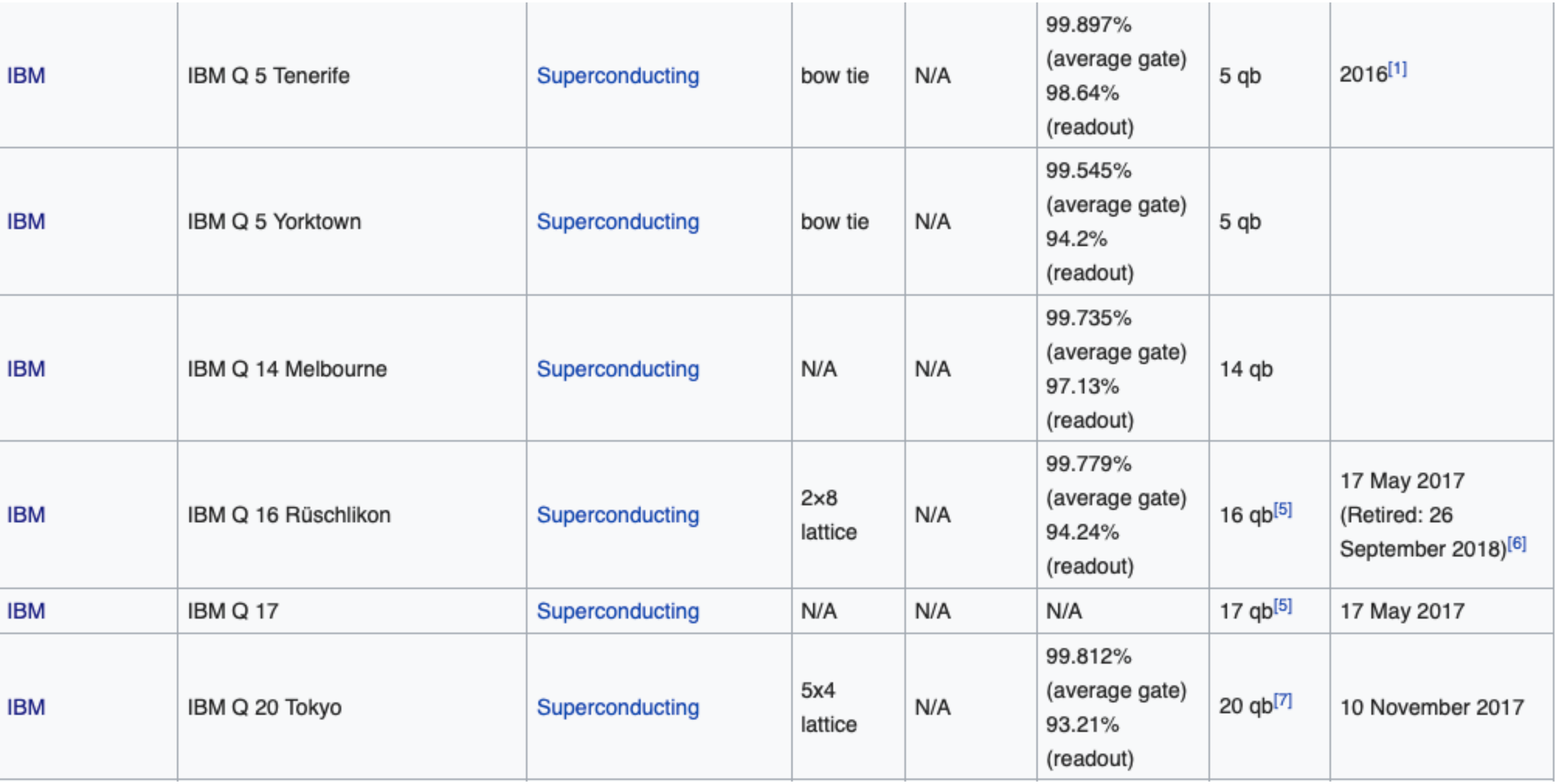
From Wikipedia, the free encyclopedia

Circuit-based quantum processors [edit]

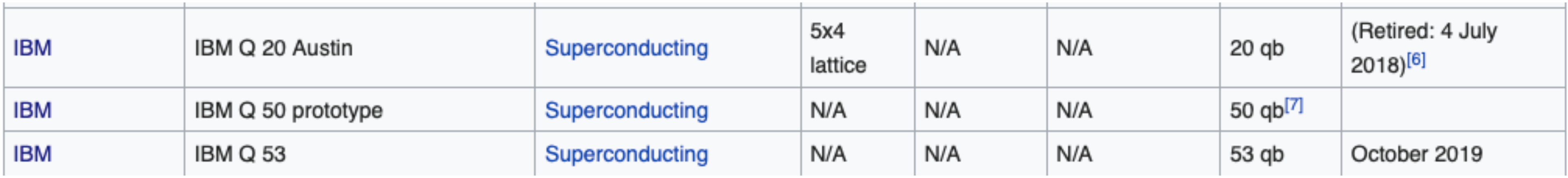
These QPUs are based on the quantum circuit and quantum logic gate-based model of computing.

Manufacturer	Name/Codename/Designation	Architecture	Layout	Socket	Fidelity	Qubits	Release date
Google	N/A	Superconducting	N/A	N/A	99.5%[1]	20 qb	2017
Google	N/A	Superconducting	7×7 lattice	N/A	99.7%[1]	49 qb[2]	Q4 2017 (planned)
Google	Bristlecone	Superconducting	6×12 lattice	N/A	99% (readout) 99.9% (1 qubit) 99.4% (2 qubits)	72 qb[3][4]	5 March 2018
Google	Sycamore	Nonlinear superconducting resonator	N/A	N/A	N/A	54 transmon qb 53 qb effective	2019

Commercial QC’s

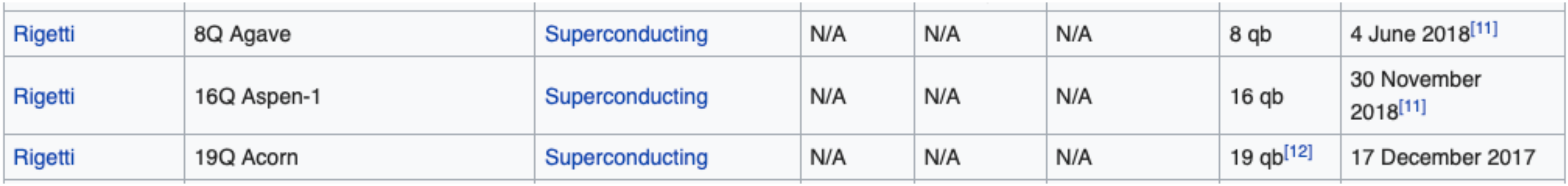


IBM	IBM Q 5 Tenerife	Superconducting	bow tie	N/A	99.897% (average gate) 98.64% (readout)	5 qb	2016[1]
IBM	IBM Q 5 Yorktown	Superconducting	bow tie	N/A	99.545% (average gate) 94.2% (readout)	5 qb	
IBM	IBM Q 14 Melbourne	Superconducting	N/A	N/A	99.735% (average gate) 97.13% (readout)	14 qb	
IBM	IBM Q 16 Rüschlikon	Superconducting	2x8 lattice	N/A	99.779% (average gate) 94.24% (readout)	16 qb[5]	17 May 2017 (Retired: 26 September 2018)[6]
IBM	IBM Q 17	Superconducting	N/A	N/A	N/A	17 qb[5]	17 May 2017
IBM	IBM Q 20 Tokyo	Superconducting	5x4 lattice	N/A	99.812% (average gate) 93.21% (readout)	20 qb[7]	10 November 2017

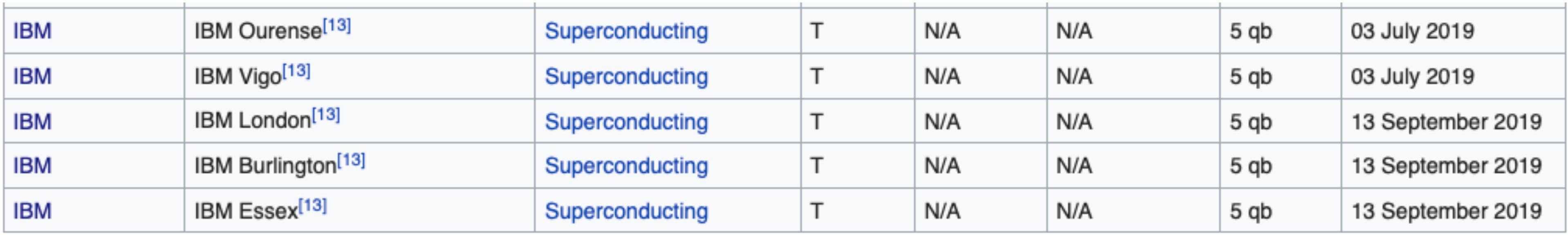


IBM	IBM Q 20 Austin	Superconducting	5x4 lattice	N/A	N/A	20 qb	(Retired: 4 July 2018)[6]
IBM	IBM Q 50 prototype	Superconducting	N/A	N/A	N/A	50 qb[7]	
IBM	IBM Q 53	Superconducting	N/A	N/A	N/A	53 qb	October 2019

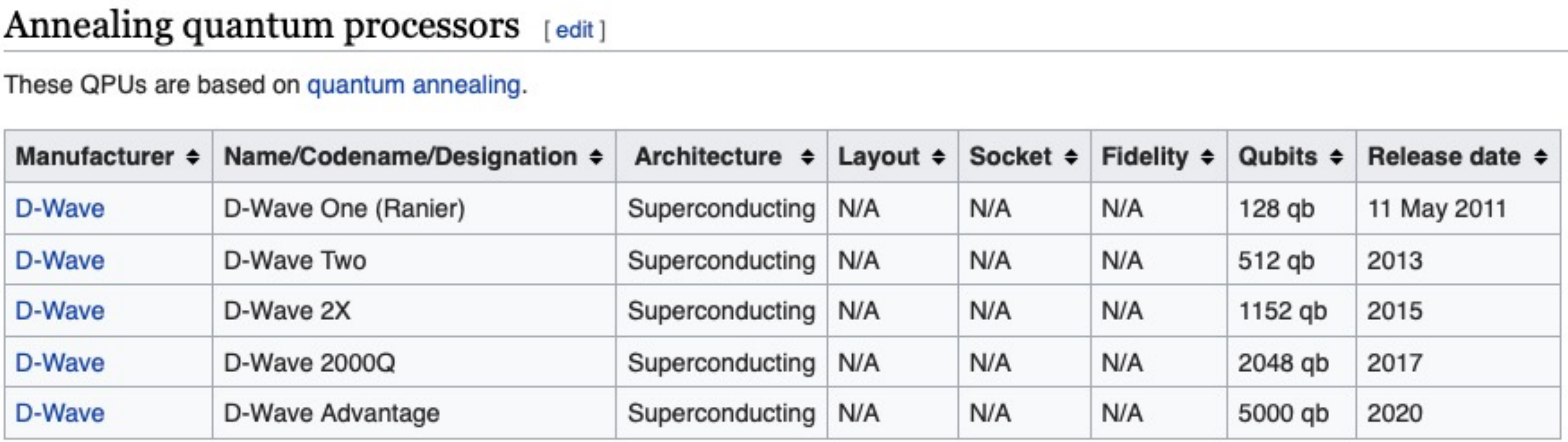
Commercial QC's



Rigetti	8Q Agave	Superconducting	N/A	N/A	N/A	8 qb	4 June 2018[11]
Rigetti	16Q Aspen-1	Superconducting	N/A	N/A	N/A	16 qb	30 November 2018[11]
Rigetti	19Q Acorn	Superconducting	N/A	N/A	N/A	19 qb[12]	17 December 2017



IBM	IBM Ourense[13]	Superconducting	T	N/A	N/A	5 qb	03 July 2019
IBM	IBM Vigo[13]	Superconducting	T	N/A	N/A	5 qb	03 July 2019
IBM	IBM London[13]	Superconducting	T	N/A	N/A	5 qb	13 September 2019
IBM	IBM Burlington[13]	Superconducting	T	N/A	N/A	5 qb	13 September 2019
IBM	IBM Essex[13]	Superconducting	T	N/A	N/A	5 qb	13 September 2019

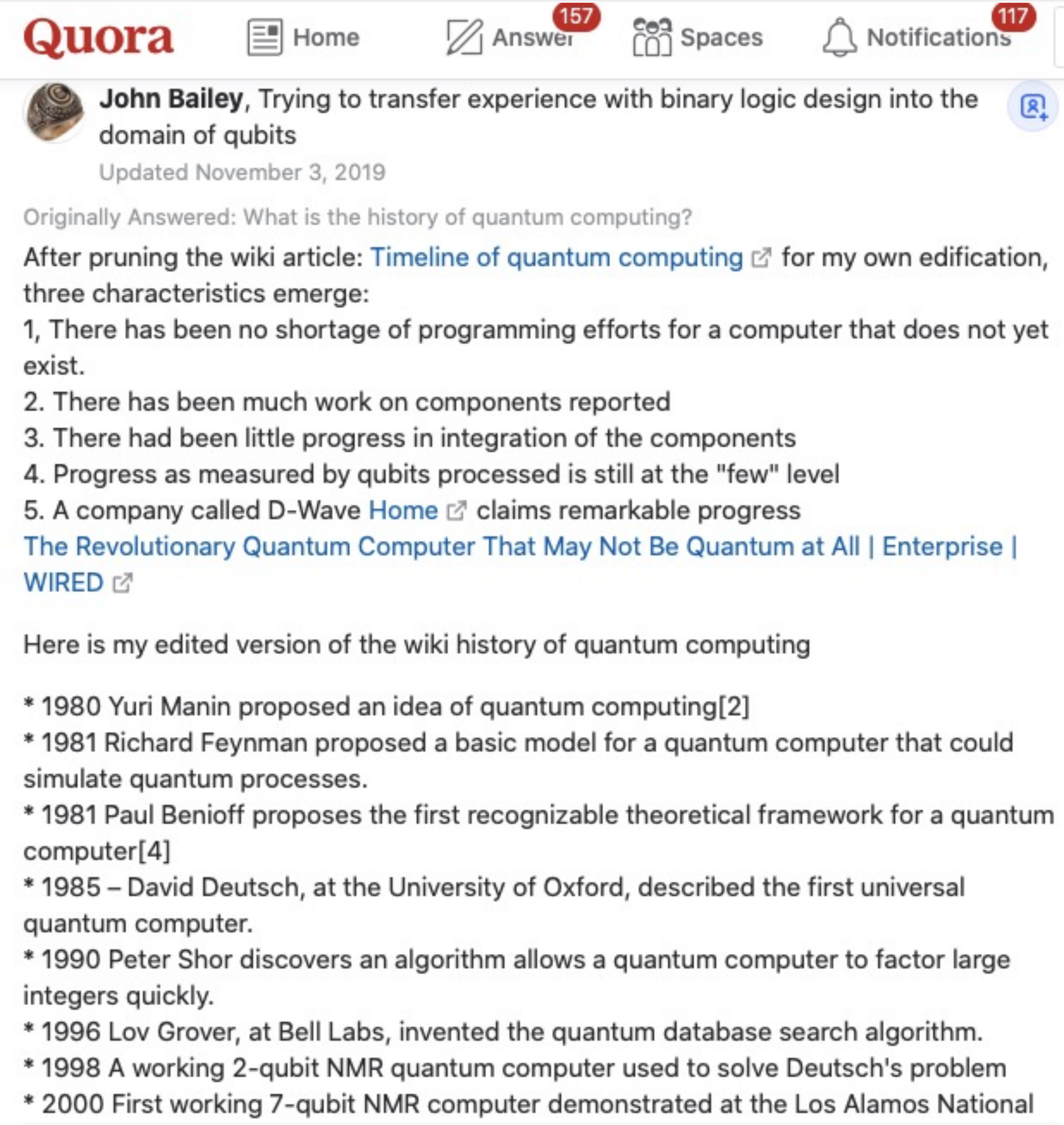


Annealing quantum processors [edit]

These QPUs are based on quantum annealing.

Manufacturer ⬍	Name/Codename/Designation ⬍	Architecture ⬍	Layout ⬍	Socket ⬍	Fidelity ⬍	Qubits ⬍	Release date ⬍
D-Wave	D-Wave One (Ranier)	Superconducting	N/A	N/A	N/A	128 qb	11 May 2011
D-Wave	D-Wave Two	Superconducting	N/A	N/A	N/A	512 qb	2013
D-Wave	D-Wave 2X	Superconducting	N/A	N/A	N/A	1152 qb	2015
D-Wave	D-Wave 2000Q	Superconducting	N/A	N/A	N/A	2048 qb	2017
D-Wave	D-Wave Advantage	Superconducting	N/A	N/A	N/A	5000 qb	2020

QC Timeline



John Bailey, Trying to transfer experience with binary logic design into the domain of qubits
Updated November 3, 2019

Originally Answered: What is the history of quantum computing?

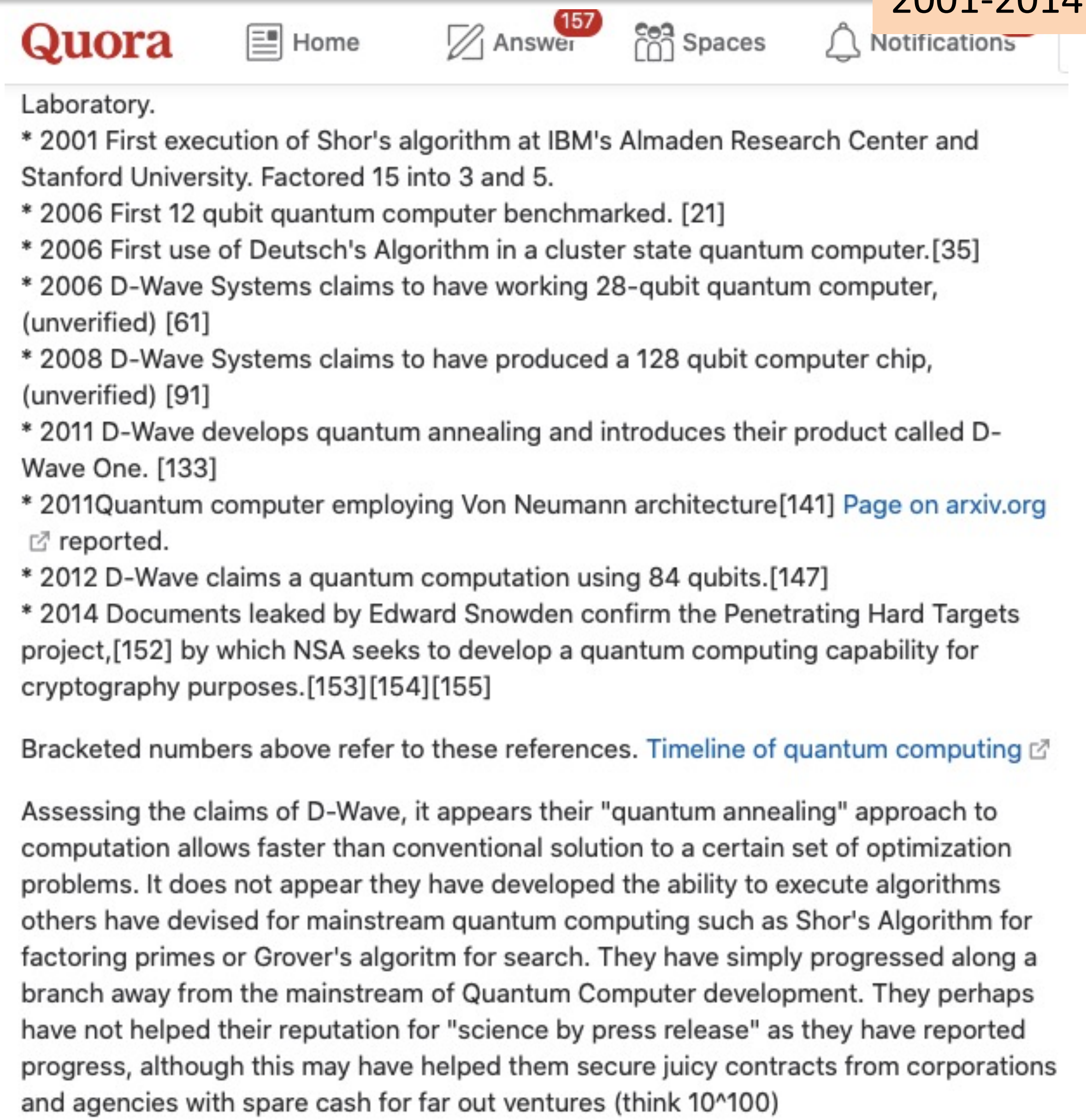
After pruning the wiki article: Timeline of quantum computing for my own edification, three characteristics emerge:
1, There has been no shortage of programming efforts for a computer that does not yet exist.
2. There has been much work on components reported
3. There had been little progress in integration of the components
4. Progress as measured by qubits processed is still at the "few" level
5. A company called D-Wave Home claims remarkable progress
The Revolutionary Quantum Computer That May Not Be Quantum at All | Enterprise | WIRED

Here is my edited version of the wiki history of quantum computing

* 1980 Yuri Manin proposed an idea of quantum computing[2]
* 1981 Richard Feynman proposed a basic model for a quantum computer that could simulate quantum processes.
* 1981 Paul Benioff proposes the first recognizable theoretical framework for a quantum computer[4]
* 1985 – David Deutsch, at the University of Oxford, described the first universal quantum computer.
* 1990 Peter Shor discovers an algorithm allows a quantum computer to factor large integers quickly.
* 1996 Lov Grover, at Bell Labs, invented the quantum database search algorithm.
* 1998 A working 2-qubit NMR quantum computer used to solve Deutsch's problem
* 2000 First working 7-qubit NMR computer demonstrated at the Los Alamos National

QC Timeline

2001-2014



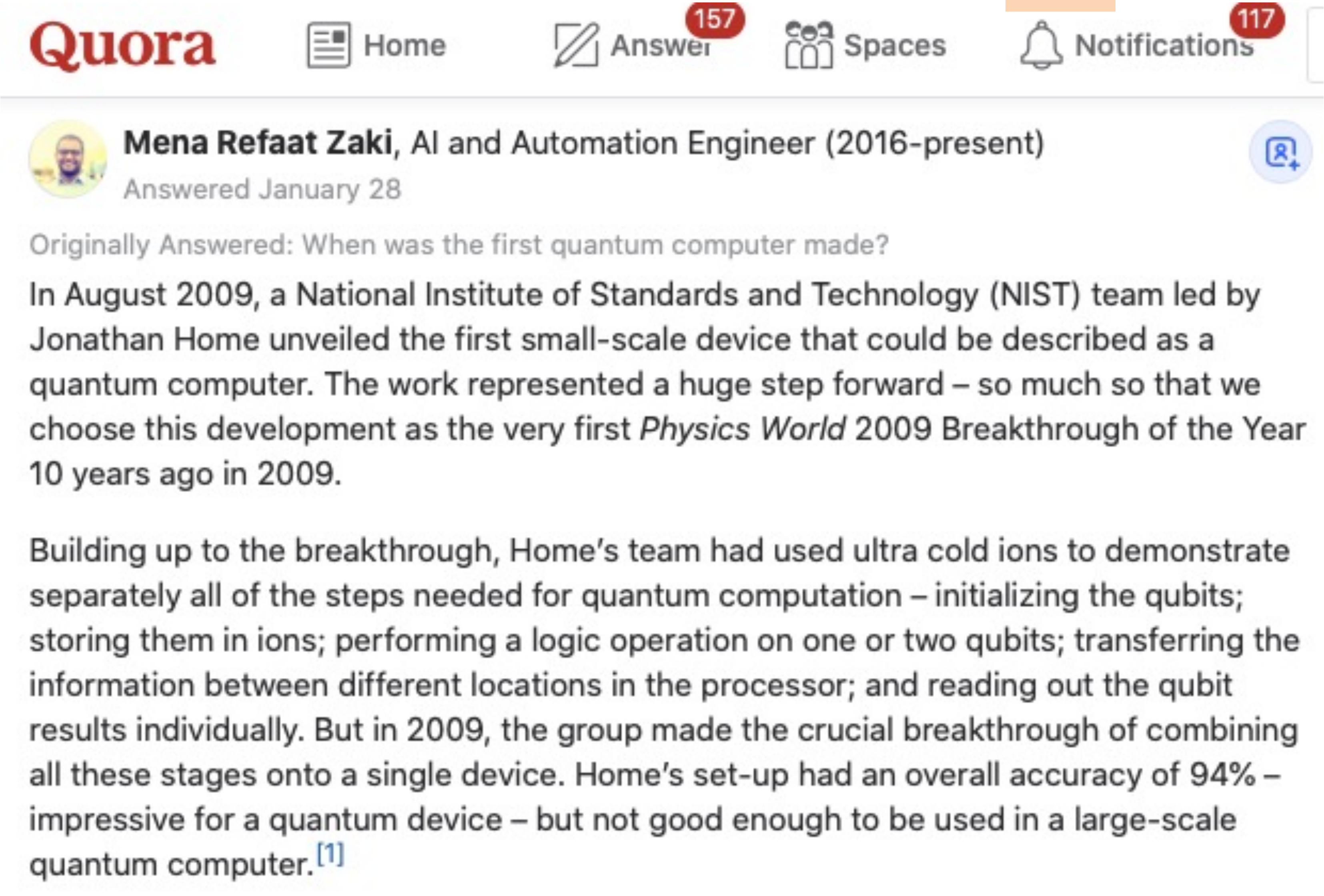
Laboratory.
* 2001 First execution of Shor's algorithm at IBM's Almaden Research Center and Stanford University. Factored 15 into 3 and 5.
* 2006 First 12 qubit quantum computer benchmarked. [21]
* 2006 First use of Deutsch's Algorithm in a cluster state quantum computer.[35]
* 2006 D-Wave Systems claims to have working 28-qubit quantum computer, (unverified) [61]
* 2008 D-Wave Systems claims to have produced a 128 qubit computer chip, (unverified) [91]
* 2011 D-Wave develops quantum annealing and introduces their product called D-Wave One. [133]
* 2011Quantum computer employing Von Neumann architecture[141] Page on arxiv.org reported.
* 2012 D-Wave claims a quantum computation using 84 qubits.[147]
* 2014 Documents leaked by Edward Snowden confirm the Penetrating Hard Targets project,[152] by which NSA seeks to develop a quantum computing capability for cryptography purposes.[153][154][155]

Bracketed numbers above refer to these references. Timeline of quantum computing

Assessing the claims of D-Wave, it appears their "quantum annealing" approach to computation allows faster than conventional solution to a certain set of optimization problems. It does not appear they have developed the ability to execute algorithms others have devised for mainstream quantum computing such as Shor's Algorithm for factoring primes or Grover's algoritm for search. They have simply progressed along a branch away from the mainstream of Quantum Computer development. They perhaps have not helped their reputation for "science by press release" as they have reported progress, although this may have helped them secure juicy contracts from corporations and agencies with spare cash for far out ventures (think 10^100)

QC Timeline

2009



Mena Refaat Zaki, AI and Automation Engineer (2016-present)

Answered January 28

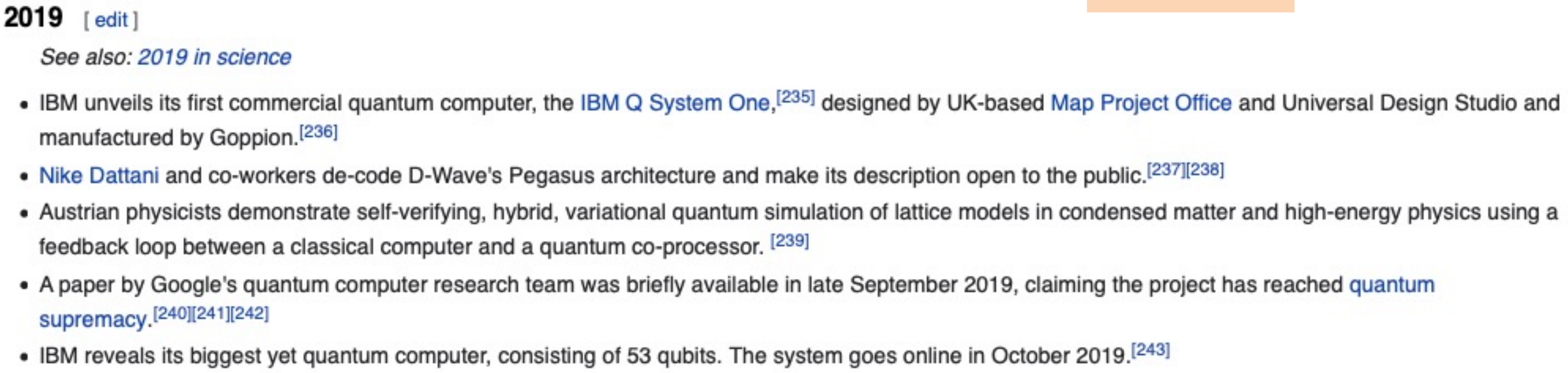
Originally Answered: When was the first quantum computer made?

In August 2009, a National Institute of Standards and Technology (NIST) team led by Jonathan Home unveiled the first small-scale device that could be described as a quantum computer. The work represented a huge step forward – so much so that we choose this development as the very first *Physics World* 2009 Breakthrough of the Year 10 years ago in 2009.

Building up to the breakthrough, Home's team had used ultra cold ions to demonstrate separately all of the steps needed for quantum computation – initializing the qubits; storing them in ions; performing a logic operation on one or two qubits; transferring the information between different locations in the processor; and reading out the qubit results individually. But in 2009, the group made the crucial breakthrough of combining all these stages onto a single device. Home's set-up had an overall accuracy of 94% – impressive for a quantum device – but not good enough to be used in a large-scale quantum computer.[1]

QC Timeline

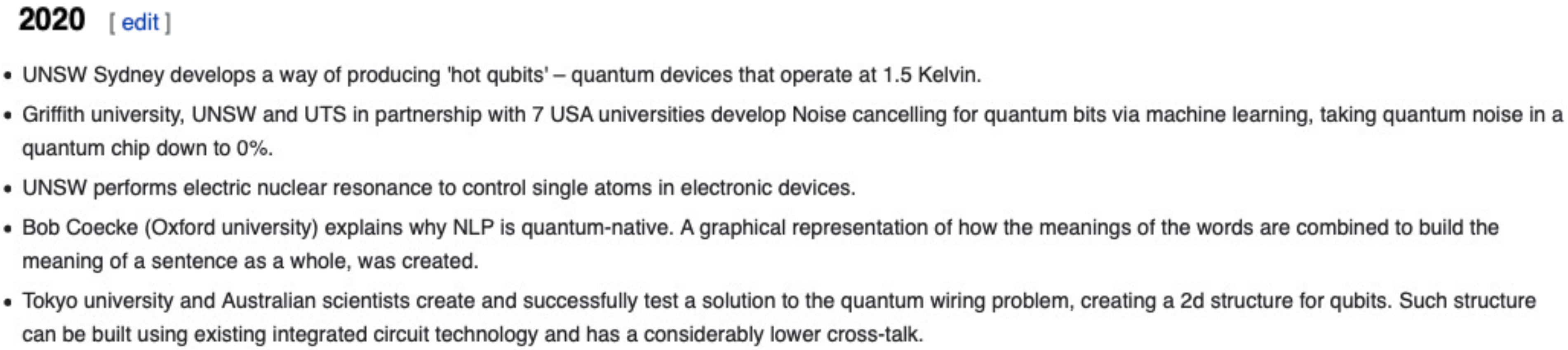
2019-2020



2019 [edit]

See also: 2019 in science

- IBM unveils its first commercial quantum computer, the IBM Q System One,[235] designed by UK-based Map Project Office and Universal Design Studio and manufactured by Goppion.[236]
- Nike Dattani and co-workers de-code D-Wave's Pegasus architecture and make its description open to the public.[237][238]
- Austrian physicists demonstrate self-verifying, hybrid, variational quantum simulation of lattice models in condensed matter and high-energy physics using a feedback loop between a classical computer and a quantum co-processor. [239]
- A paper by Google's quantum computer research team was briefly available in late September 2019, claiming the project has reached quantum supremacy.[240][241][242]
- IBM reveals its biggest yet quantum computer, consisting of 53 qubits. The system goes online in October 2019.[243]



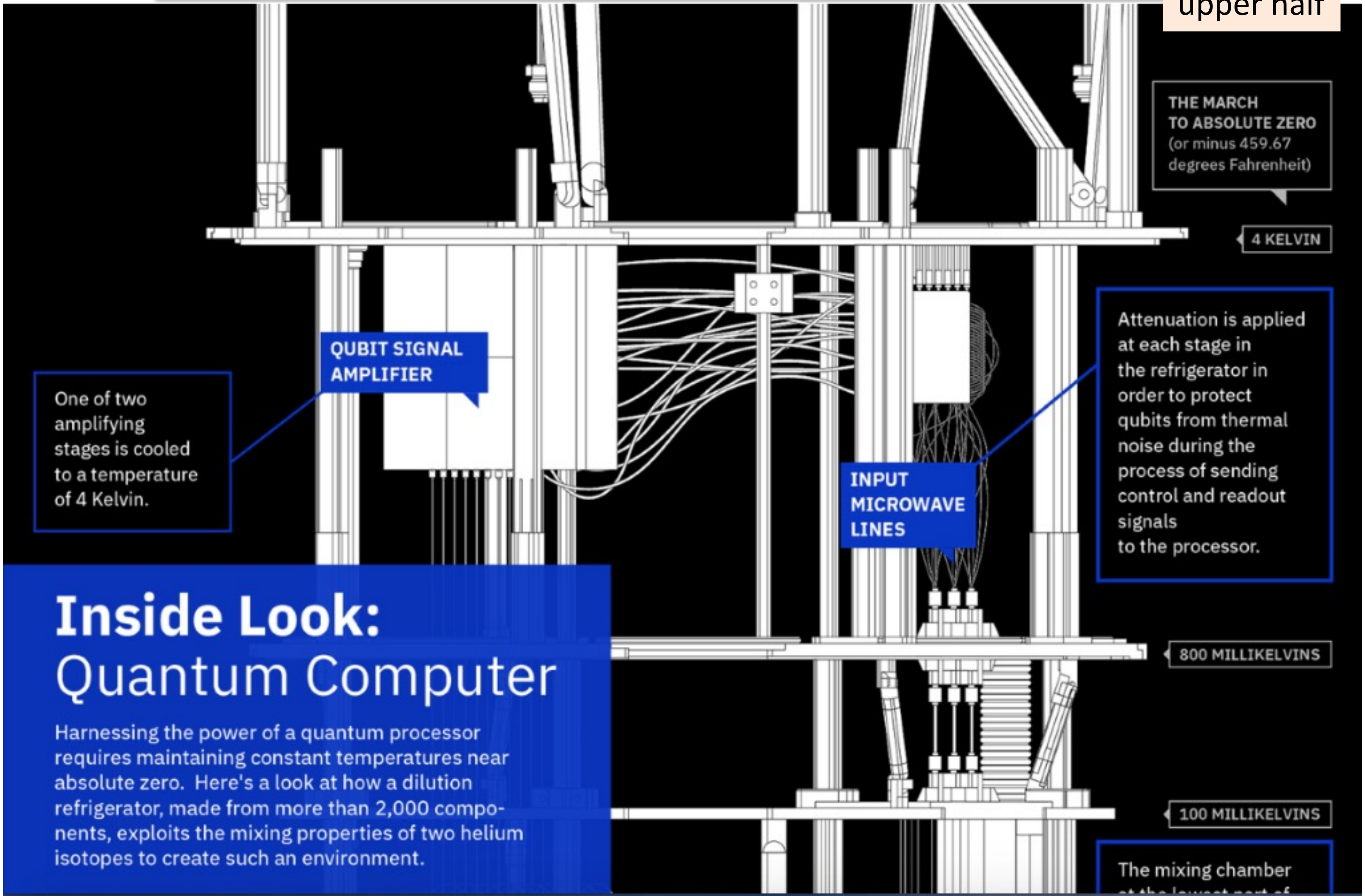
2020 [edit]

- UNSW Sydney develops a way of producing 'hot qubits' – quantum devices that operate at 1.5 Kelvin.
- Griffith university, UNSW and UTS in partnership with 7 USA universities develop Noise cancelling for quantum bits via machine learning, taking quantum noise in a quantum chip down to 0%.
- UNSW performs electric nuclear resonance to control single atoms in electronic devices.
- Bob Coecke (Oxford university) explains why NLP is quantum-native. A graphical representation of how the meanings of the words are combined to build the meaning of a sentence as a whole, was created.
- Tokyo university and Australian scientists create and successfully test a solution to the quantum wiring problem, creating a 2d structure for qubits. Such structure can be built using existing integrated circuit technology and has a considerably lower cross-talk.

News: IBM's Q

DR JEFF SOFTWARE
INDIE APP DEVELOPER
© Jeff Drobman
2016-23

upper half



One of two amplifying stages is cooled to a temperature of 4 Kelvin.

Attenuation is applied at each stage in the refrigerator in order to protect qubits from thermal noise during the process of sending control and readout signals to the processor.

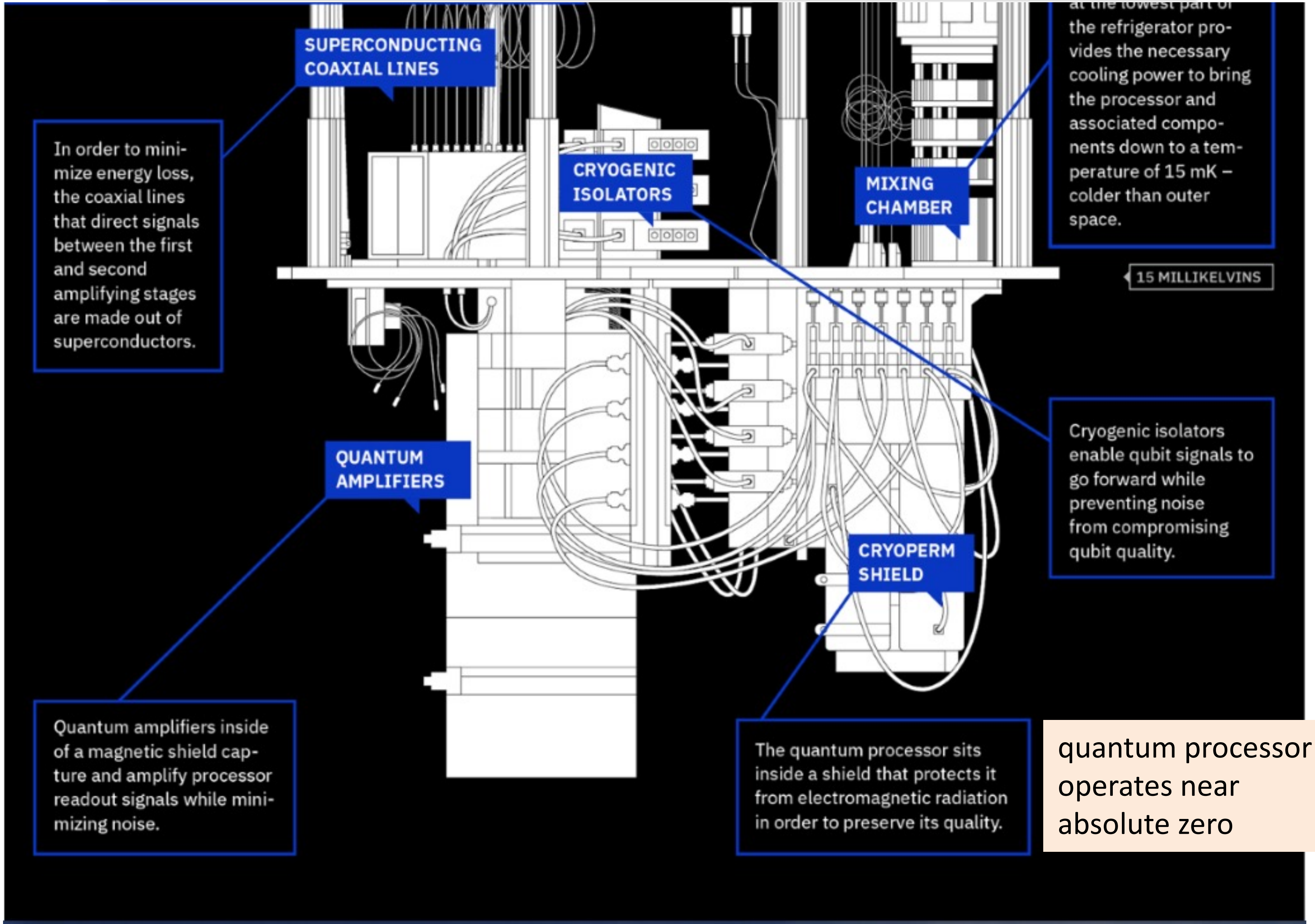
Inside Look: Quantum Computer

Harnessing the power of a quantum processor requires maintaining constant temperatures near absolute zero. Here's a look at how a dilution refrigerator, made from more than 2,000 components, exploits the mixing properties of two helium isotopes to create such an environment.

The mixing chamber

News: IBM's Q

lower half



at the lowest part of the refrigerator provides the necessary cooling power to bring the processor and associated components down to a temperature of 15 mK – colder than outer space.

In order to minimize energy loss, the coaxial lines that direct signals between the first and second amplifying stages are made out of superconductors.

Cryogenic isolators enable qubit signals to go forward while preventing noise from compromising qubit quality.

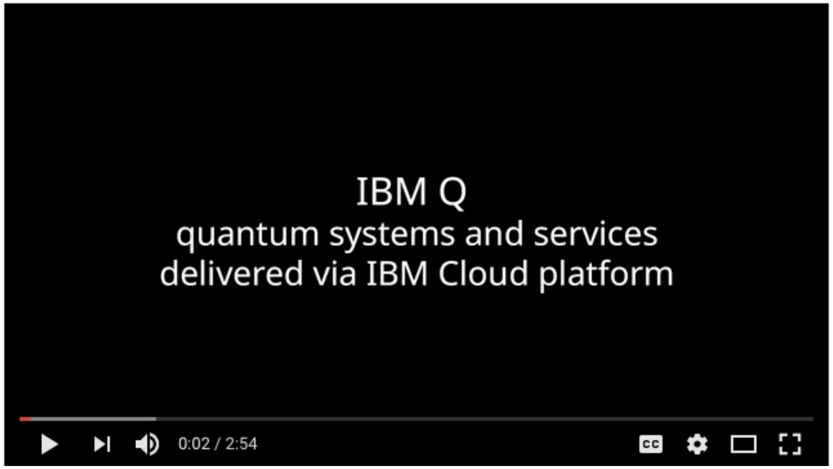
Quantum amplifiers inside of a magnetic shield capture and amplify processor readout signals while minimizing noise.

The quantum processor sits inside a shield that protects it from electromagnetic radiation in order to preserve its quality.

quantum processor operates near absolute zero

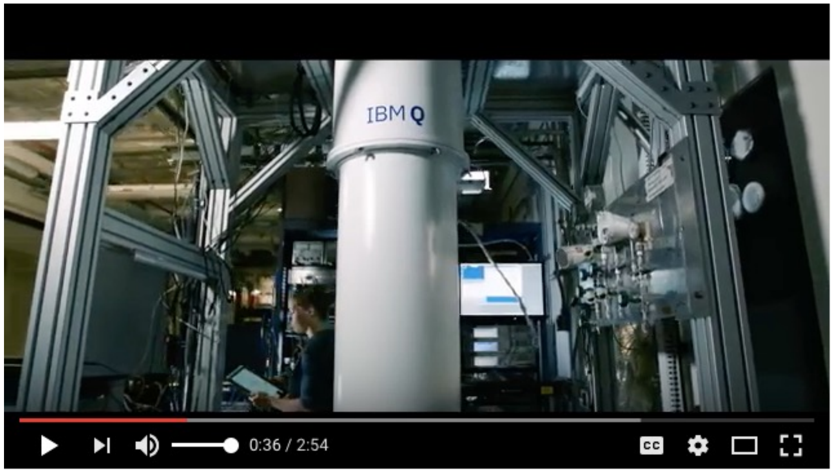
News: IBM's Q

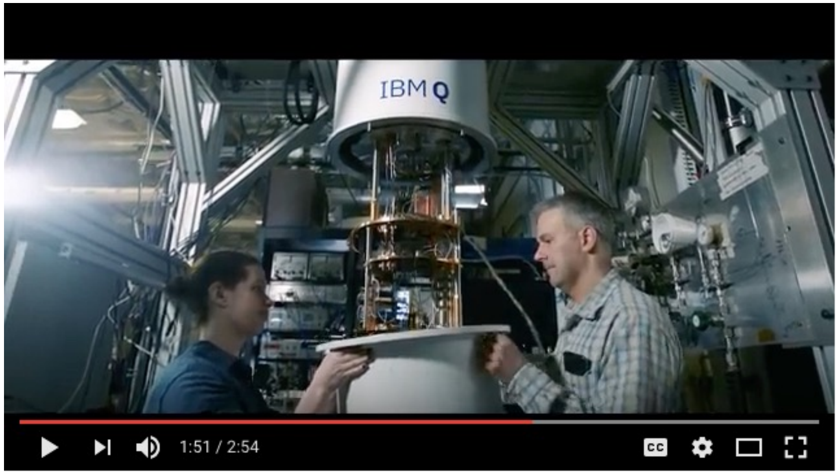
https://www.youtube.com/watch?time_continue=174&v=2B680d-qvhI



IBM Q
quantum systems and services
delivered via IBM Cloud platform

Enter quantum computing.

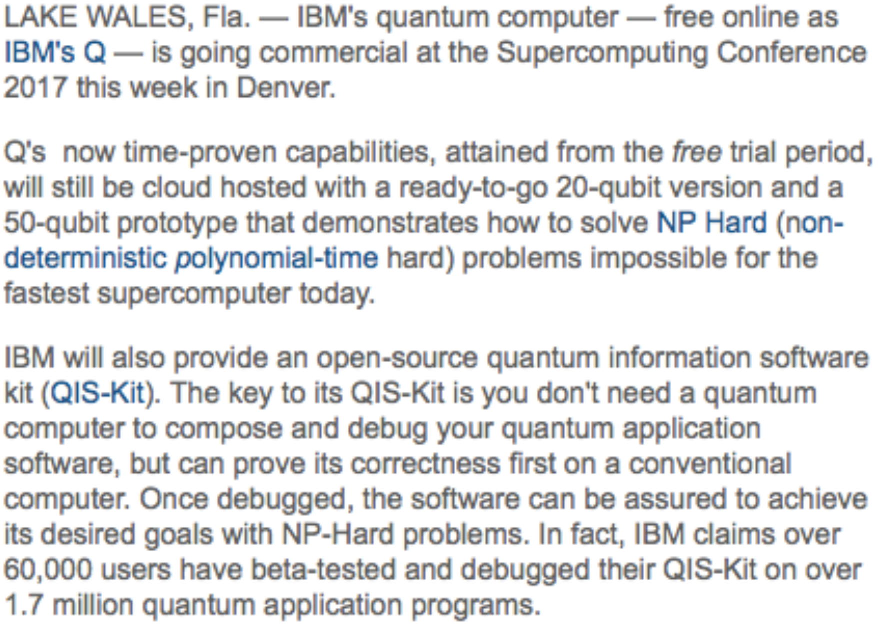




https://www.youtube.com/watch?v=yy6TV9Dntlw

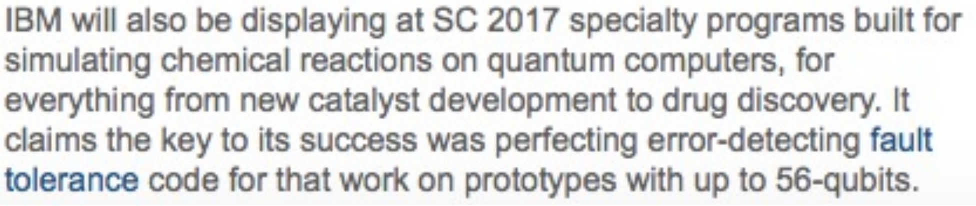
News: IBM's Q

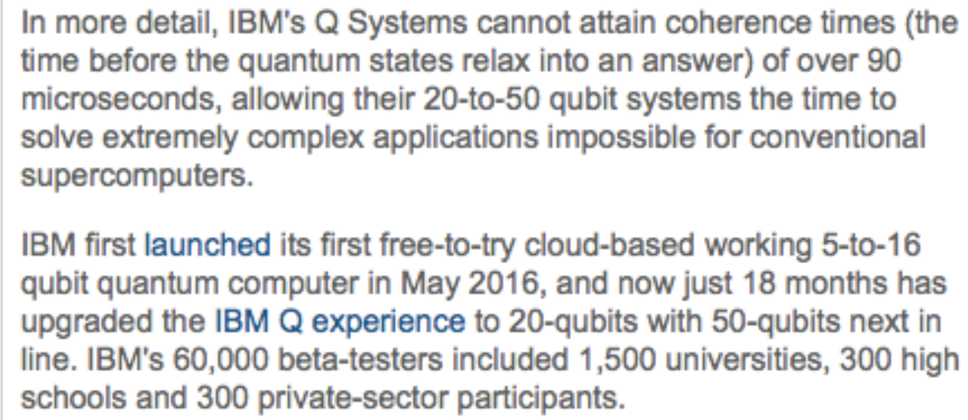
IBM's quantum computer in the cloud free to use for all comers.
Source: IBM


LAKE WALES, Fla. — IBM's quantum computer — free online as IBM's Q — is going commercial at the Supercomputing Conference 2017 this week in Denver.

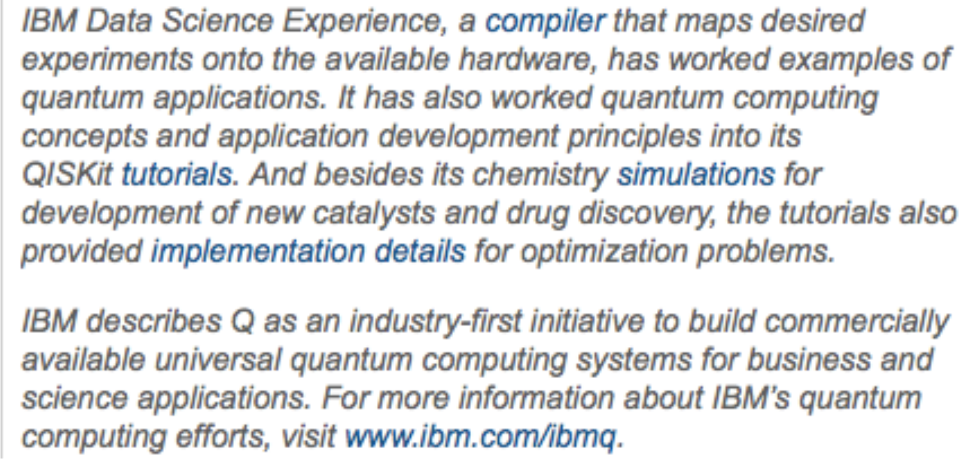
Q's now time-proven capabilities, attained from the *free* trial period, will still be cloud hosted with a ready-to-go 20-qubit version and a 50-qubit prototype that demonstrates how to solve NP Hard (non-deterministic polynomial-time hard) problems impossible for the fastest supercomputer today.

IBM will also provide an open-source quantum information software kit (QIS-Kit). The key to its QIS-Kit is you don't need a quantum computer to compose and debug your quantum application software, but can prove its correctness first on a conventional computer. Once debugged, the software can be assured to achieve its desired goals with NP-Hard problems. In fact, IBM claims over 60,000 users have beta-tested and debugged their QIS-Kit on over 1.7 million quantum application programs.


IBM will also be displaying at SC 2017 specialty programs built for simulating chemical reactions on quantum computers, for everything from new catalyst development to drug discovery. It claims the key to its success was perfecting error-detecting fault tolerance code for that work on prototypes with up to 56-qubits.


In more detail, IBM's Q Systems cannot attain coherence times (the time before the quantum states relax into an answer) of over 90 microseconds, allowing their 20-to-50 qubit systems the time to solve extremely complex applications impossible for conventional supercomputers.

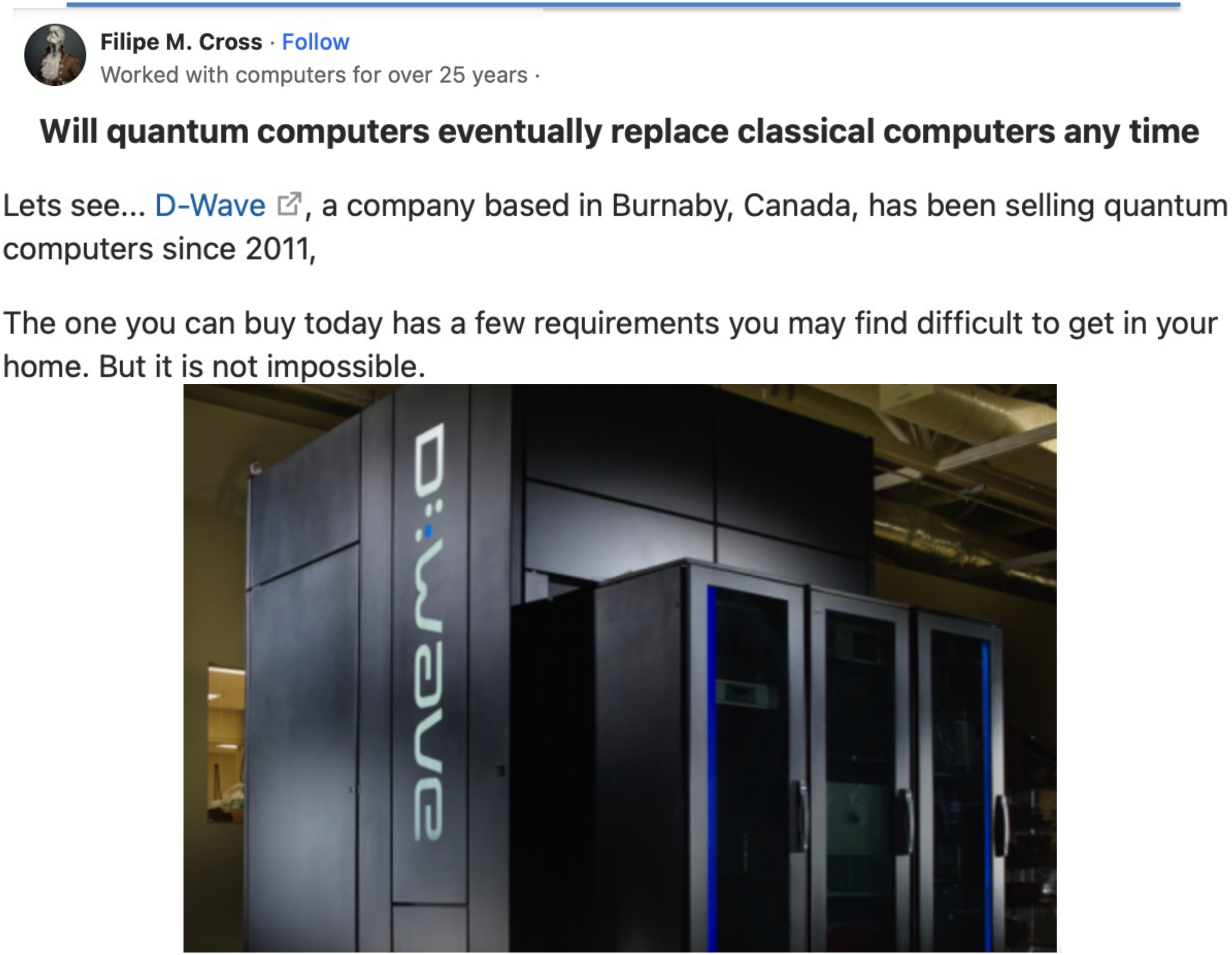
IBM first launched its first free-to-try cloud-based working 5-to-16 qubit quantum computer in May 2016, and now just 18 months has upgraded the IBM Q experience to 20-qubits with 50-qubits next in line. IBM's 60,000 beta-testers included 1,500 universities, 300 high schools and 300 private-sector participants.


IBM Data Science Experience, a compiler that maps desired experiments onto the available hardware, has worked examples of quantum applications. It has also worked quantum computing concepts and application development principles into its QISKit tutorials. And besides its chemistry simulations for development of new catalysts and drug discovery, the tutorials also provided implementation details for optimization problems.

IBM describes Q as an industry-first initiative to build commercially available universal quantum computing systems for business and science applications. For more information about IBM's quantum computing efforts, visit www.ibm.com/ibmq.

D-Wave

Quora



Filipe M. Cross · Follow

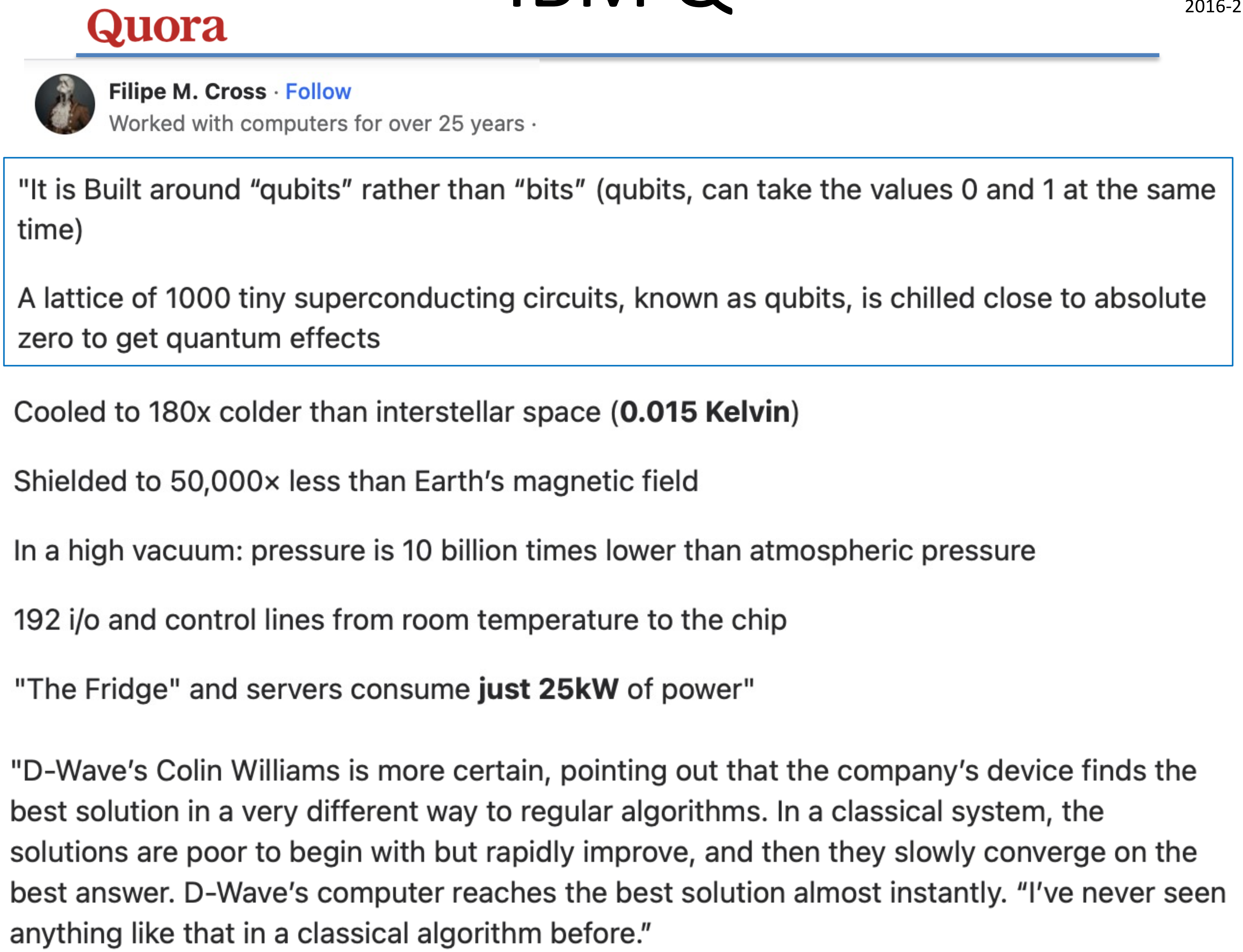
Worked with computers for over 25 years ·

Will quantum computers eventually replace classical computers any time

Lets see... D-Wave, a company based in Burnaby, Canada, has been selling quantum computers since 2011,

The one you can buy today has a few requirements you may find difficult to get in your home. But it is not impossible.

IBM Q



Quora

Filipe M. Cross · Follow
Worked with computers for over 25 years ·

"It is Built around “qubits” rather than “bits” (qubits, can take the values 0 and 1 at the same time)

A lattice of 1000 tiny superconducting circuits, known as qubits, is chilled close to absolute zero to get quantum effects

Cooled to 180x colder than interstellar space (**0.015 Kelvin**)

Shielded to 50,000× less than Earth’s magnetic field

In a high vacuum: pressure is 10 billion times lower than atmospheric pressure

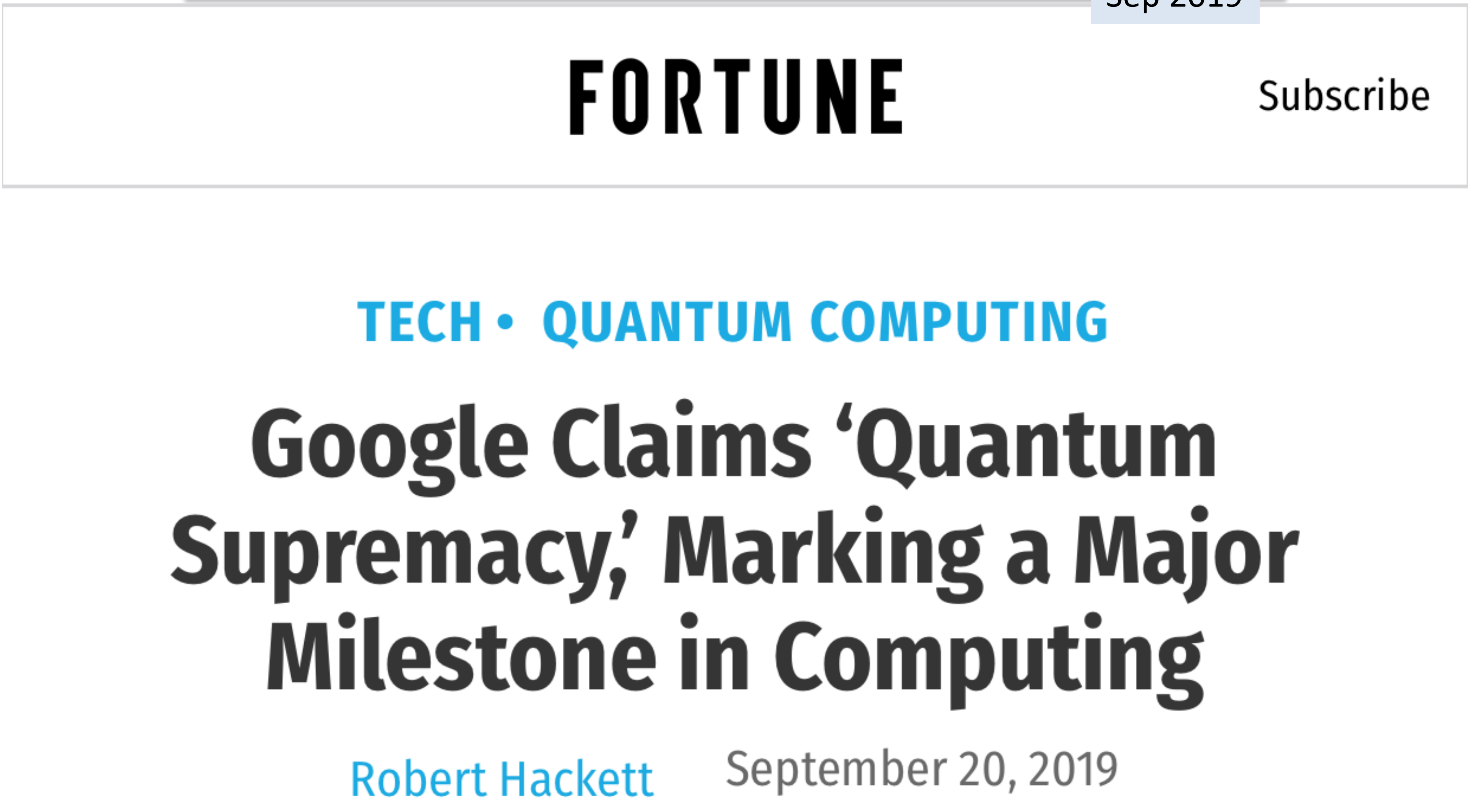
192 i/o and control lines from room temperature to the chip

"The Fridge" and servers consume **just 25kW** of power"

"D-Wave’s Colin Williams is more certain, pointing out that the company’s device finds the best solution in a very different way to regular algorithms. In a classical system, the solutions are poor to begin with but rapidly improve, and then they slowly converge on the best answer. D-Wave’s computer reaches the best solution almost instantly. “I’ve never seen anything like that in a classical algorithm before.”

Google QC

Sep 2019



FORTUNE

Subscribe

TECH • QUANTUM COMPUTING

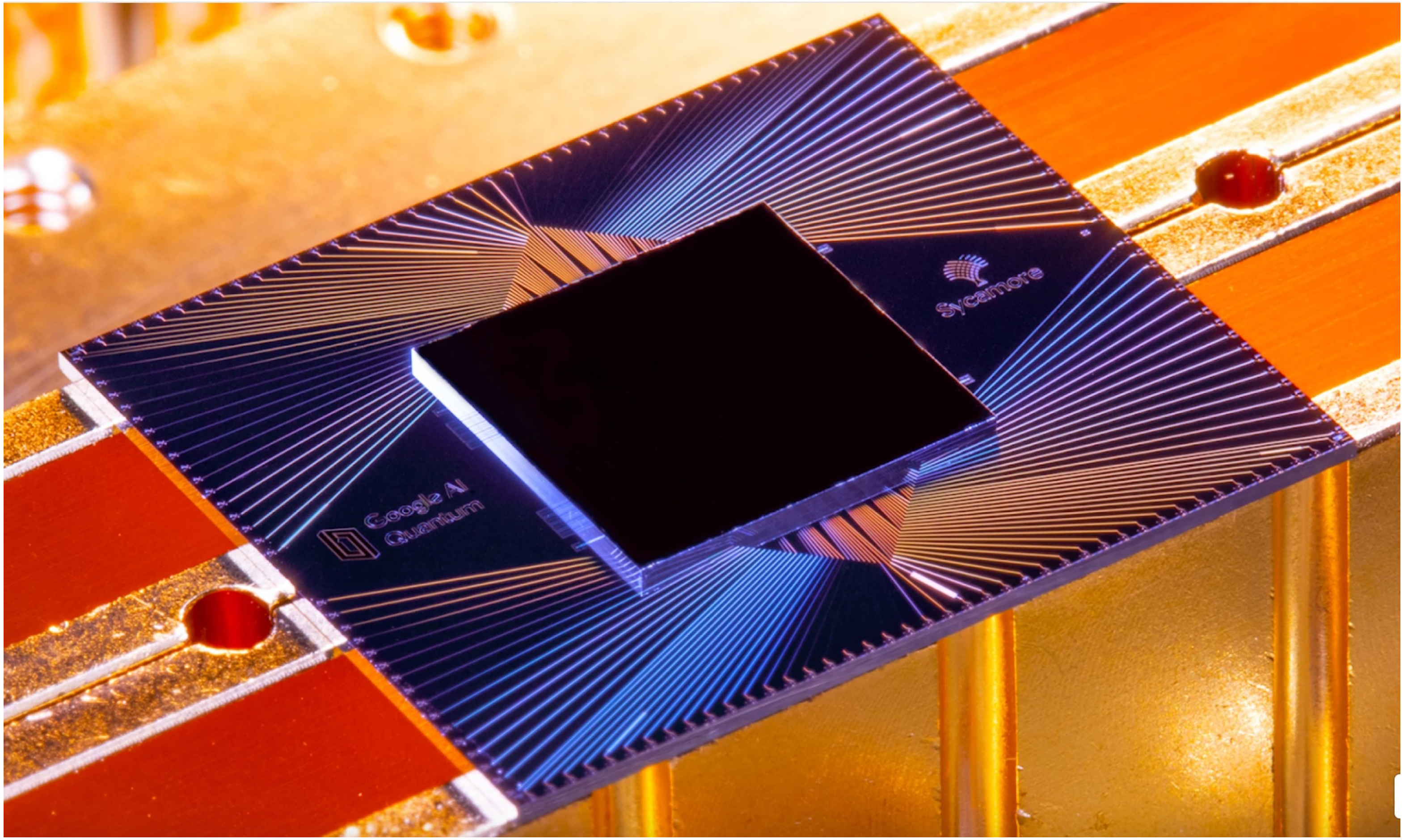
Google Claims 'Quantum Supremacy,' Marking a Major Milestone in Computing

Robert Hackett September 20, 2019

Google Sycamore QC

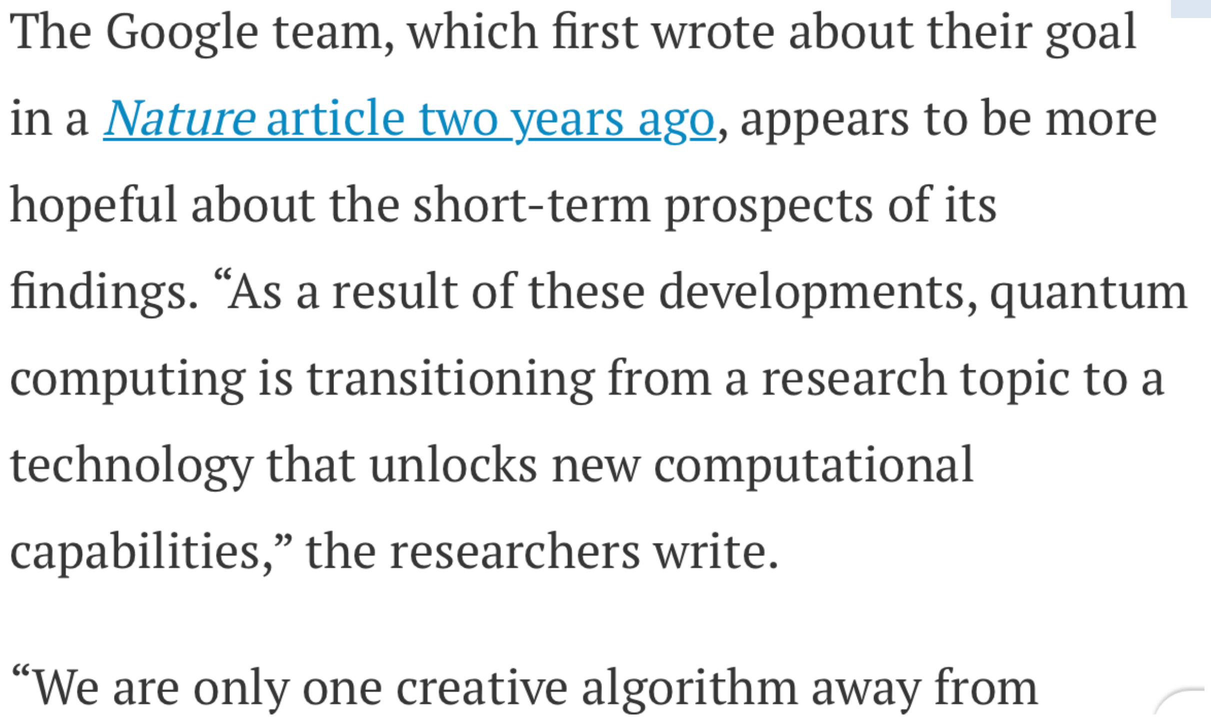
PHYSICS

Time crystals created in Google's quantum processor



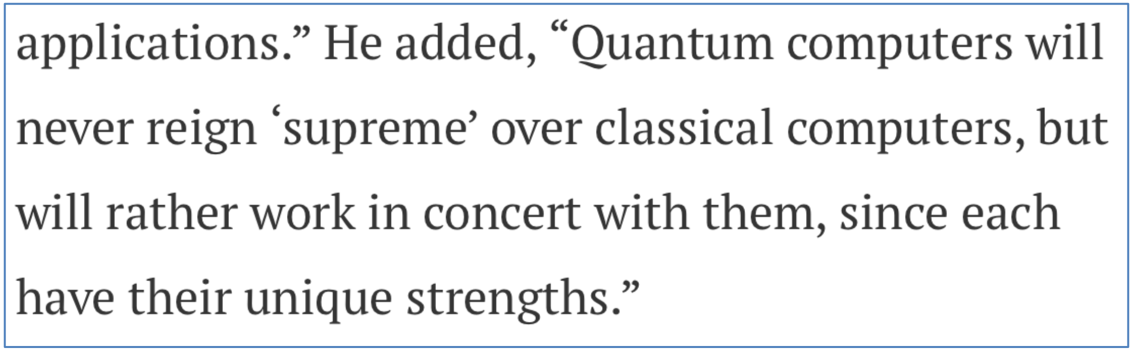
Google QC

Sep 2019



The Google team, which first wrote about their goal in a *Nature* article two years ago, appears to be more hopeful about the short-term prospects of its findings. "As a result of these developments, quantum computing is transitioning from a research topic to a technology that unlocks new computational capabilities," the researchers write.

"We are only one creative algorithm away from

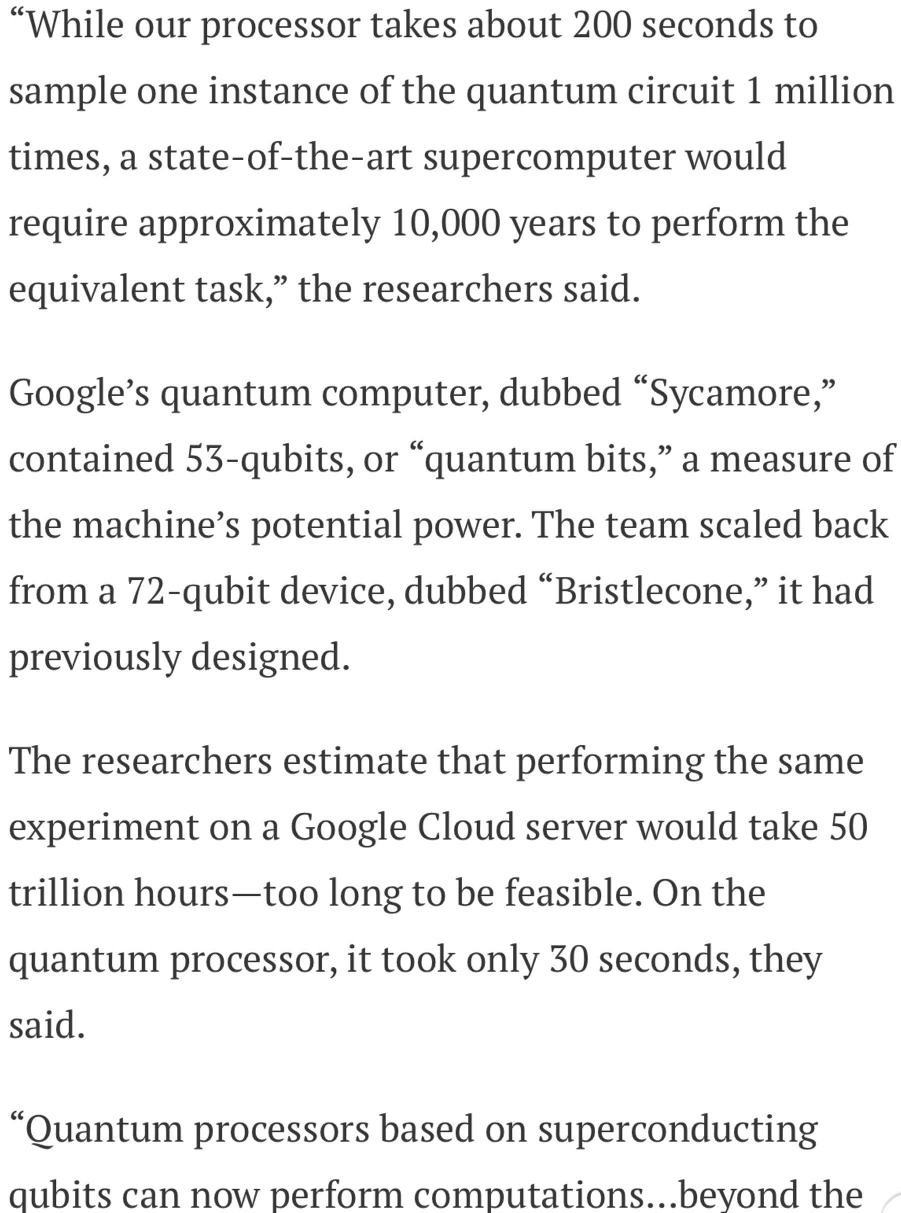


applications." He added, "Quantum computers will never reign 'supreme' over classical computers, but will rather work in concert with them, since each have their unique strengths."

Google QC

53-qubits

Sep 2019



“While our processor takes about 200 seconds to sample one instance of the quantum circuit 1 million times, a state-of-the-art supercomputer would require approximately 10,000 years to perform the equivalent task,” the researchers said.

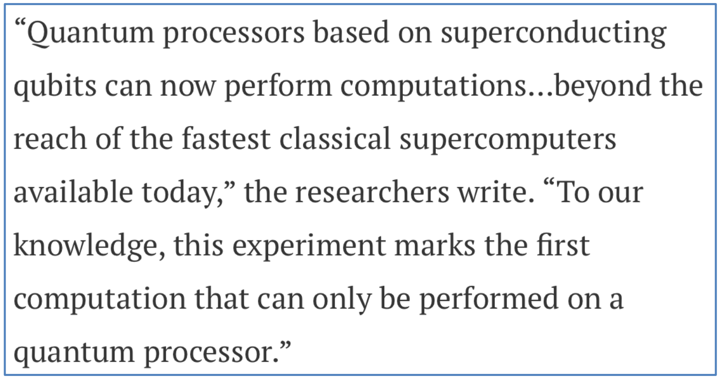
Google’s quantum computer, dubbed “Sycamore,” contained 53-qubits, or “quantum bits,” a measure of the machine’s potential power. The team scaled back from a 72-qubit device, dubbed “Bristlecone,” it had previously designed.

The researchers estimate that performing the same experiment on a Google Cloud server would take 50 trillion hours—too long to be feasible. On the quantum processor, it took only 30 seconds, they said.

“Quantum processors based on superconducting qubits can now perform computations…beyond the

200 sec << 10,000 years

Random number generation

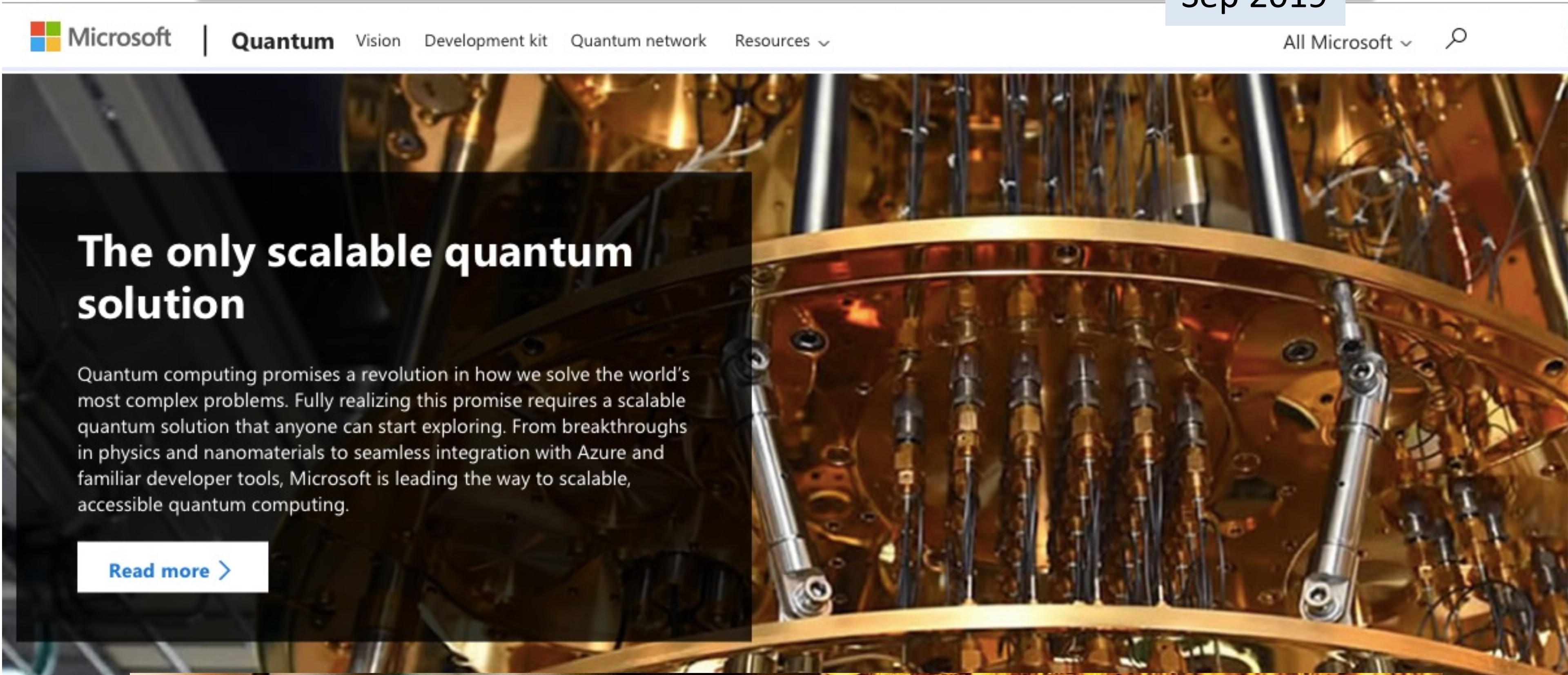


“Quantum processors based on superconducting qubits can now perform computations…beyond the reach of the fastest classical supercomputers available today,” the researchers write. “To our knowledge, this experiment marks the first computation that can only be performed on a quantum processor.”

Server = 50T hours

Microsoft QC

Sep 2019



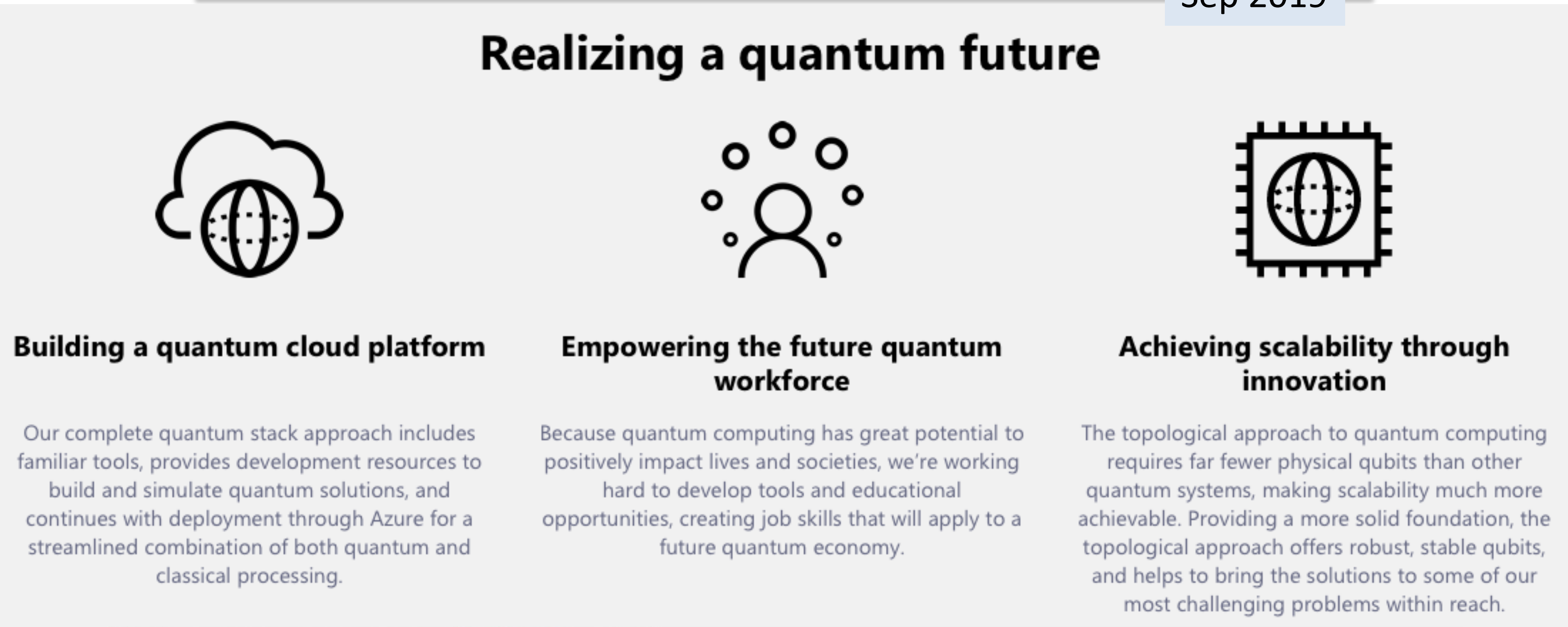
The only scalable quantum solution

Quantum computing promises a revolution in how we solve the world's most complex problems. Fully realizing this promise requires a scalable quantum solution that anyone can start exploring. From breakthroughs in physics and nanomaterials to seamless integration with Azure and familiar developer tools, Microsoft is leading the way to scalable, accessible quantum computing.

Read more >

Microsoft QC

Sep 2019



Realizing a quantum future

Building a quantum cloud platform

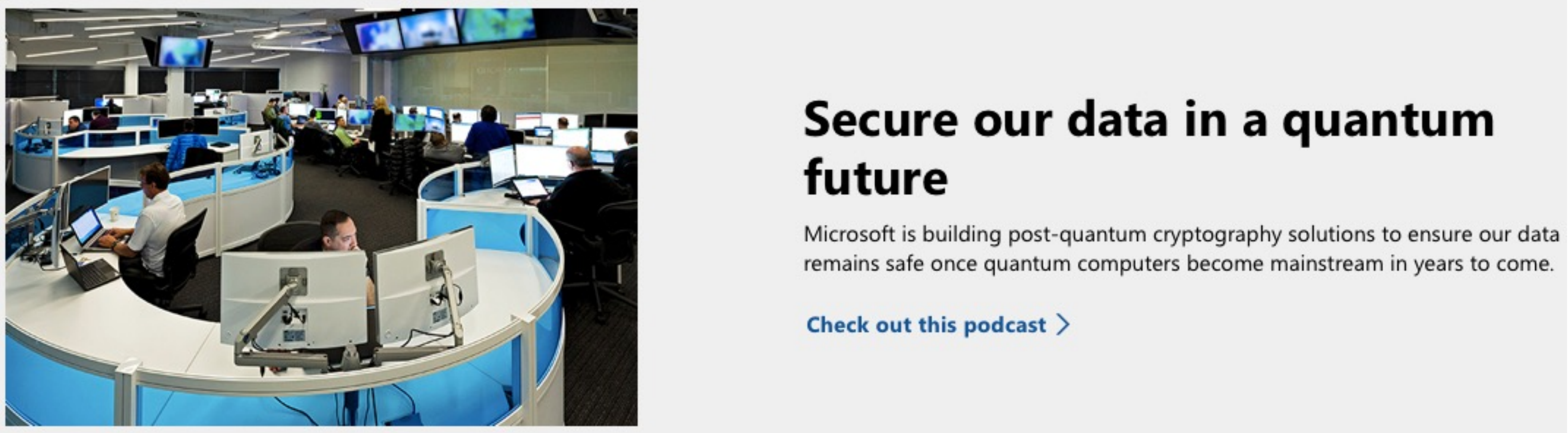
Our complete quantum stack approach includes familiar tools, provides development resources to build and simulate quantum solutions, and continues with deployment through Azure for a streamlined combination of both quantum and classical processing.

Empowering the future quantum workforce

Because quantum computing has great potential to positively impact lives and societies, we're working hard to develop tools and educational opportunities, creating job skills that will apply to a future quantum economy.

Achieving scalability through innovation

The topological approach to quantum computing requires far fewer physical qubits than other quantum systems, making scalability much more achievable. Providing a more solid foundation, the topological approach offers robust, stable qubits, and helps to bring the solutions to some of our most challenging problems within reach.



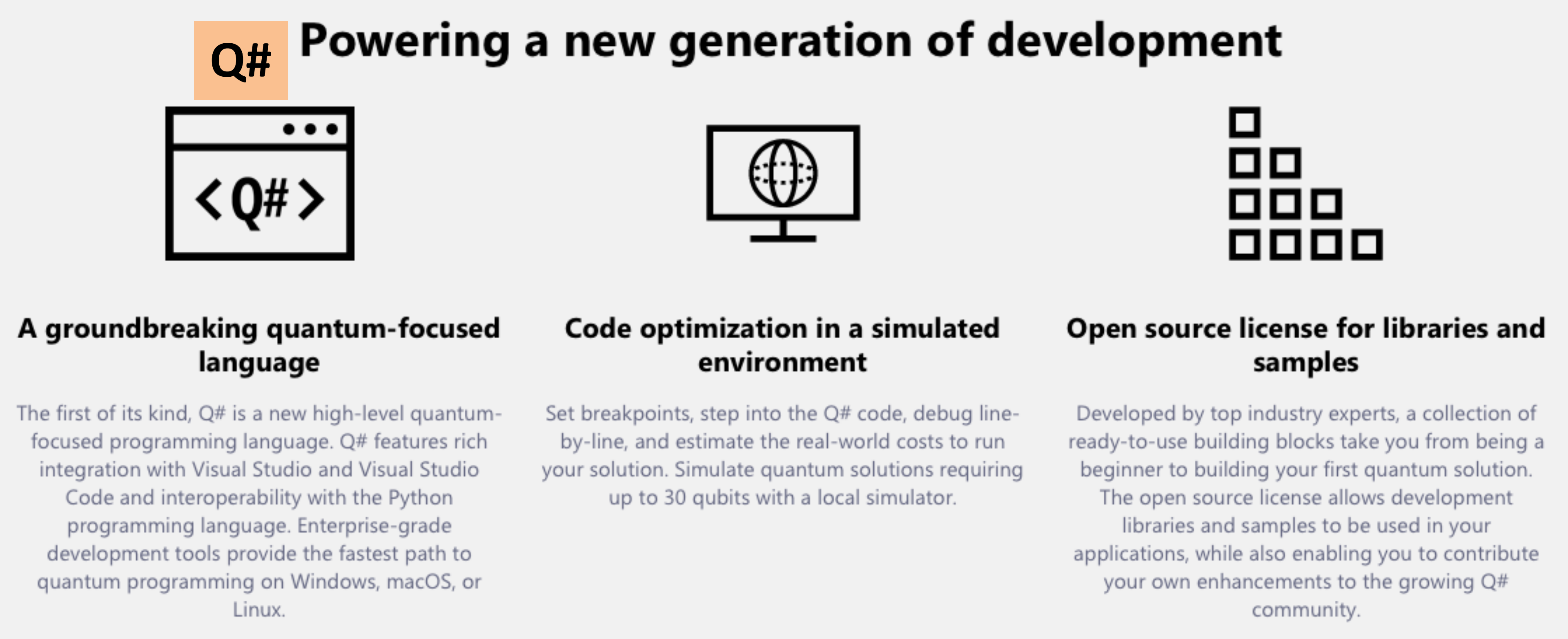
Secure our data in a quantum future

Microsoft is building post-quantum cryptography solutions to ensure our data remains safe once quantum computers become mainstream in years to come.

Check out this podcast >

Microsoft QC Tools

Sep 2019



Q#

Powering a new generation of development

A groundbreaking quantum-focused language

The first of its kind, Q# is a new high-level quantum-focused programming language. Q# features rich integration with Visual Studio and Visual Studio Code and interoperability with the Python programming language. Enterprise-grade development tools provide the fastest path to quantum programming on Windows, macOS, or Linux.

Code optimization in a simulated environment

Set breakpoints, step into the Q# code, debug line-by-line, and estimate the real-world costs to run your solution. Simulate quantum solutions requiring up to 30 qubits with a local simulator.

Open source license for libraries and samples

Developed by top industry experts, a collection of ready-to-use building blocks take you from being a beginner to building your first quantum solution. The open source license allows development libraries and samples to be used in your applications, while also enabling you to contribute your own enhancements to the growing Q# community.

Visual Studio + Python

Supports ***quantum inspired*** algorithms that *simulate*
<= 30 **Qubits**

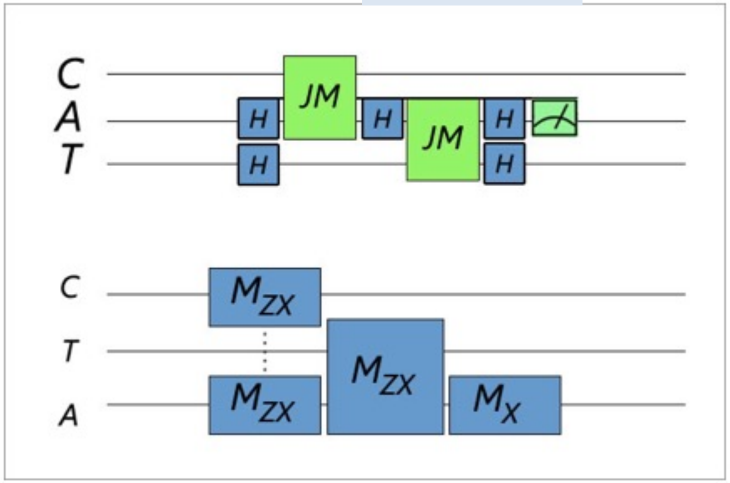
Microsoft QC Tools

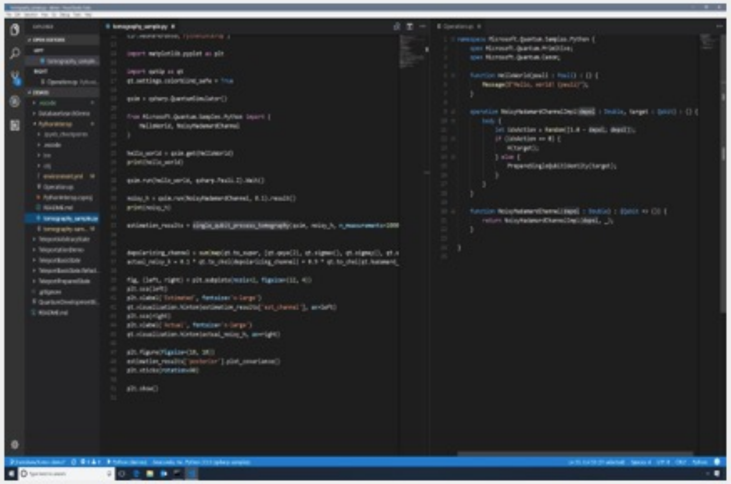
DR JEFF SOFTWARE
INDIE APP DEVELOPER
© Jeff Drobman 2016-23

Sep 2019

Runtime

To solve problems on a quantum computer, you need a runtime that executes a quantum algorithm while maintaining the state of the machine, operating the control system in a parallel real-time environment, and communicating from the device to the outside world. The runtime layer is the firmware and operating system of the quantum computer.



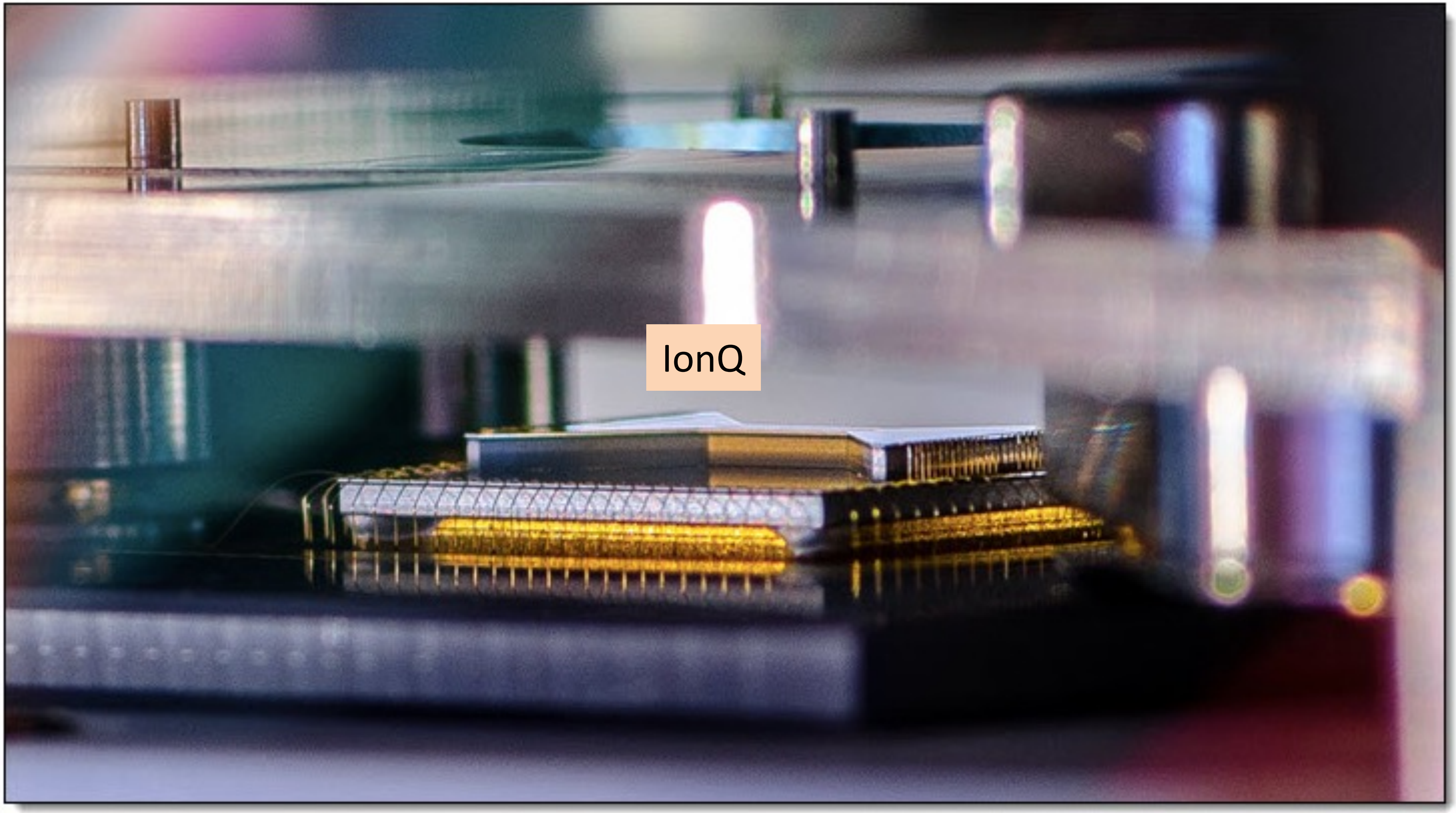


Quantum development tools

To help quantum developers build applications and algorithms, we've designed the **Quantum Development Kit**—a set of enterprise-grade tools to write, debug, and optimize quantum code. Microsoft has been focused on providing an integrated software experience for as long as we've been working on the hardware itself, and this kit includes everything you need to get started. Microsoft will also create quantum services in Azure, giving you a fast path from simulation to optimization to deployment on quantum hardware.

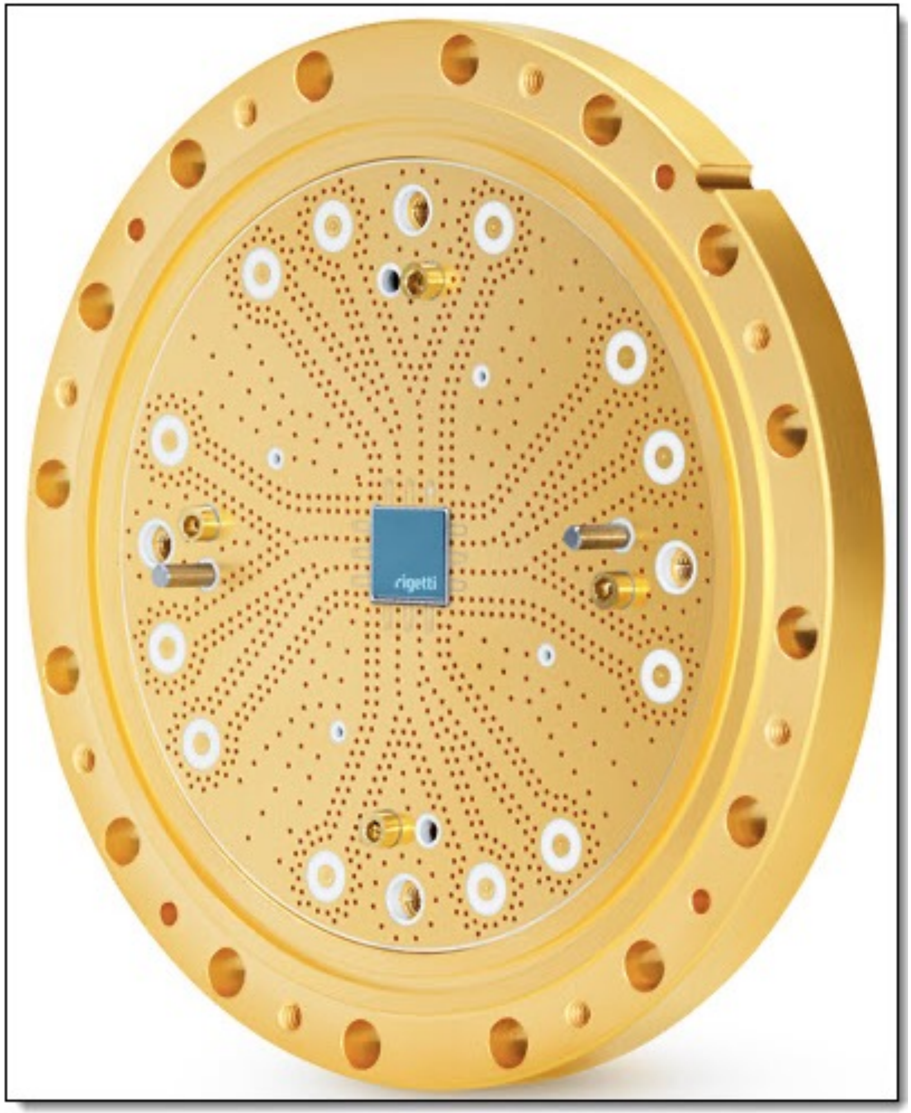
IonQ

And the IonQ linear ion trap:



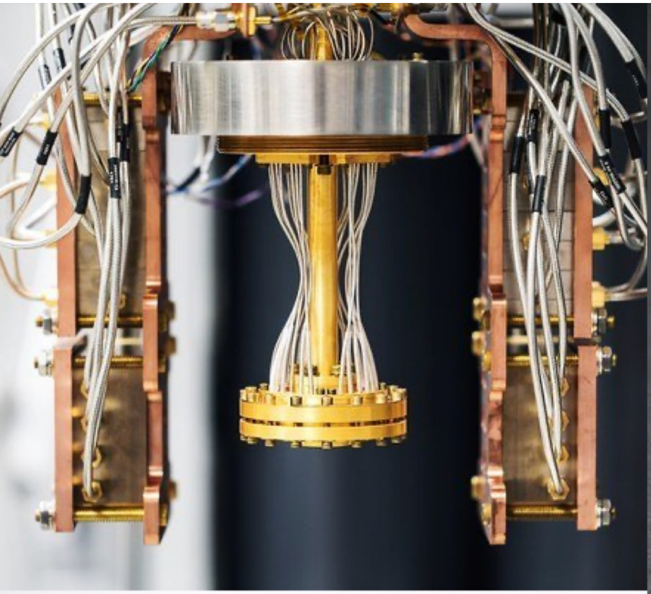
Rigetti

The Rigetti 16Q Aspen-4:

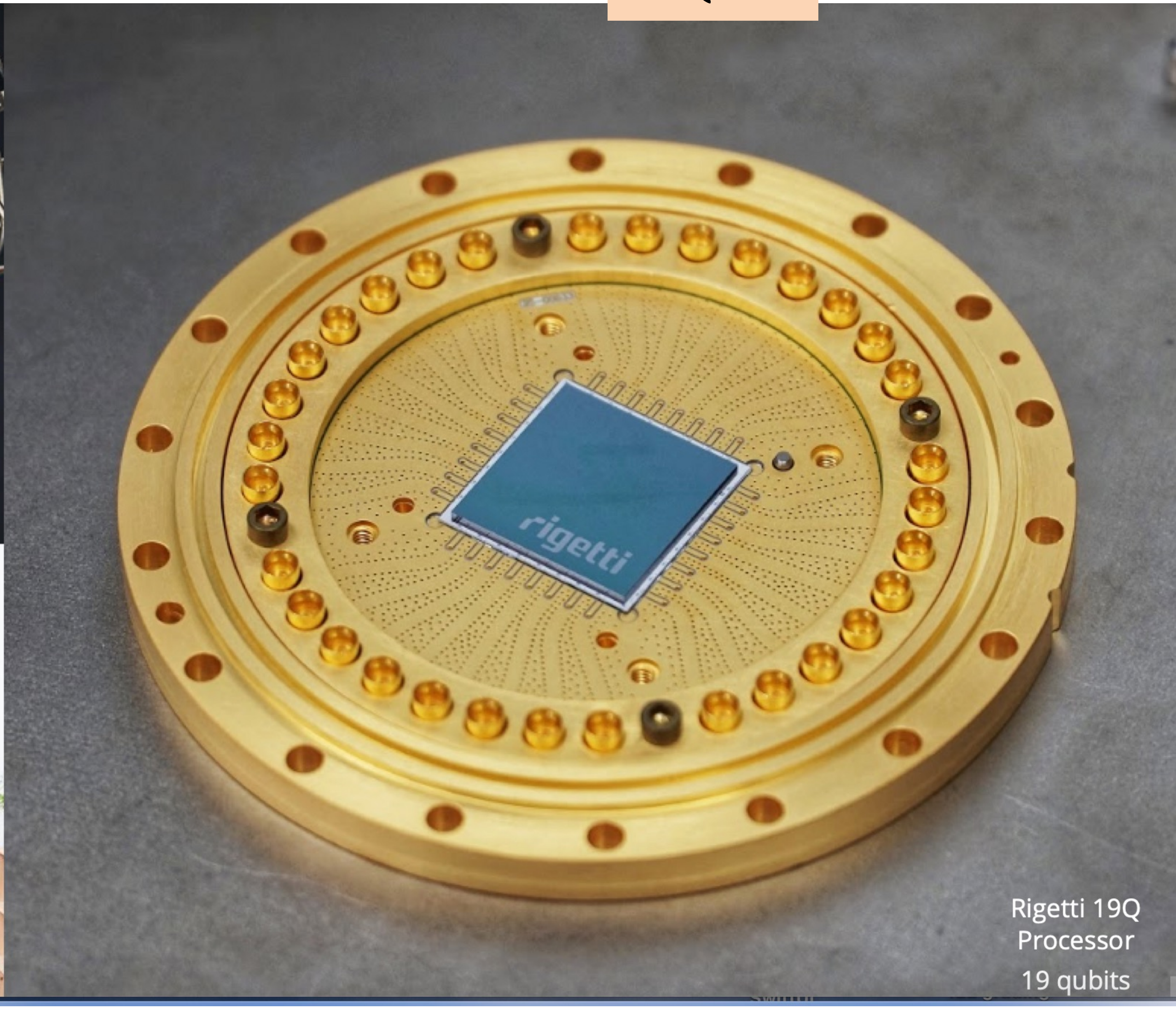


QC’s

19 Qubits



Rigetti Computing

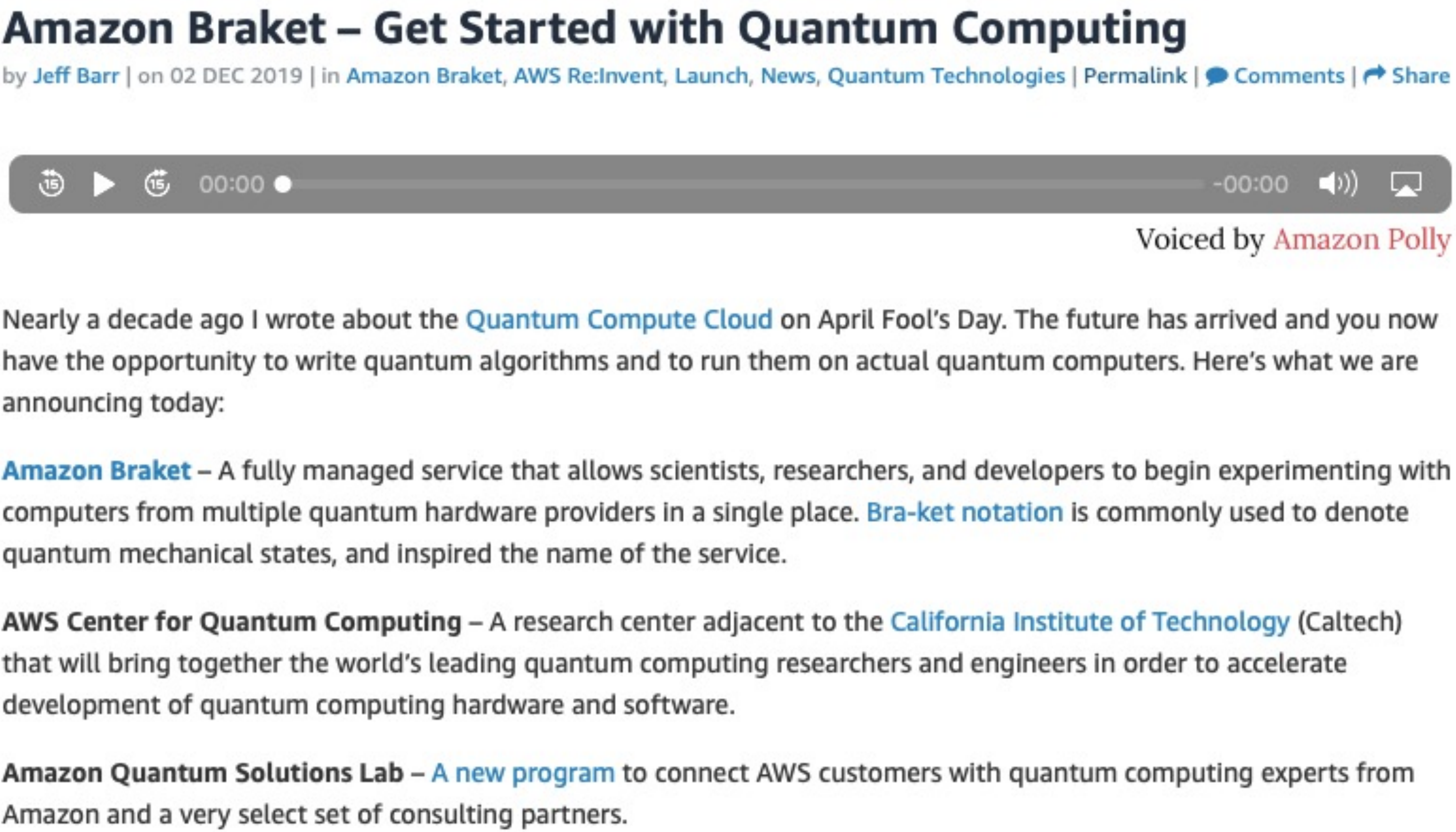


Rigetti 19Q Processor

19 qubits

Amazon Q1

Amazon Q1



Amazon Braket – Get Started with Quantum Computing

by Jeff Barr | on 02 DEC 2019 | in Amazon Braket, AWS Re:Invent, Launch, News, Quantum Technologies | Permalink | Comments | Share

Voiced by Amazon Polly

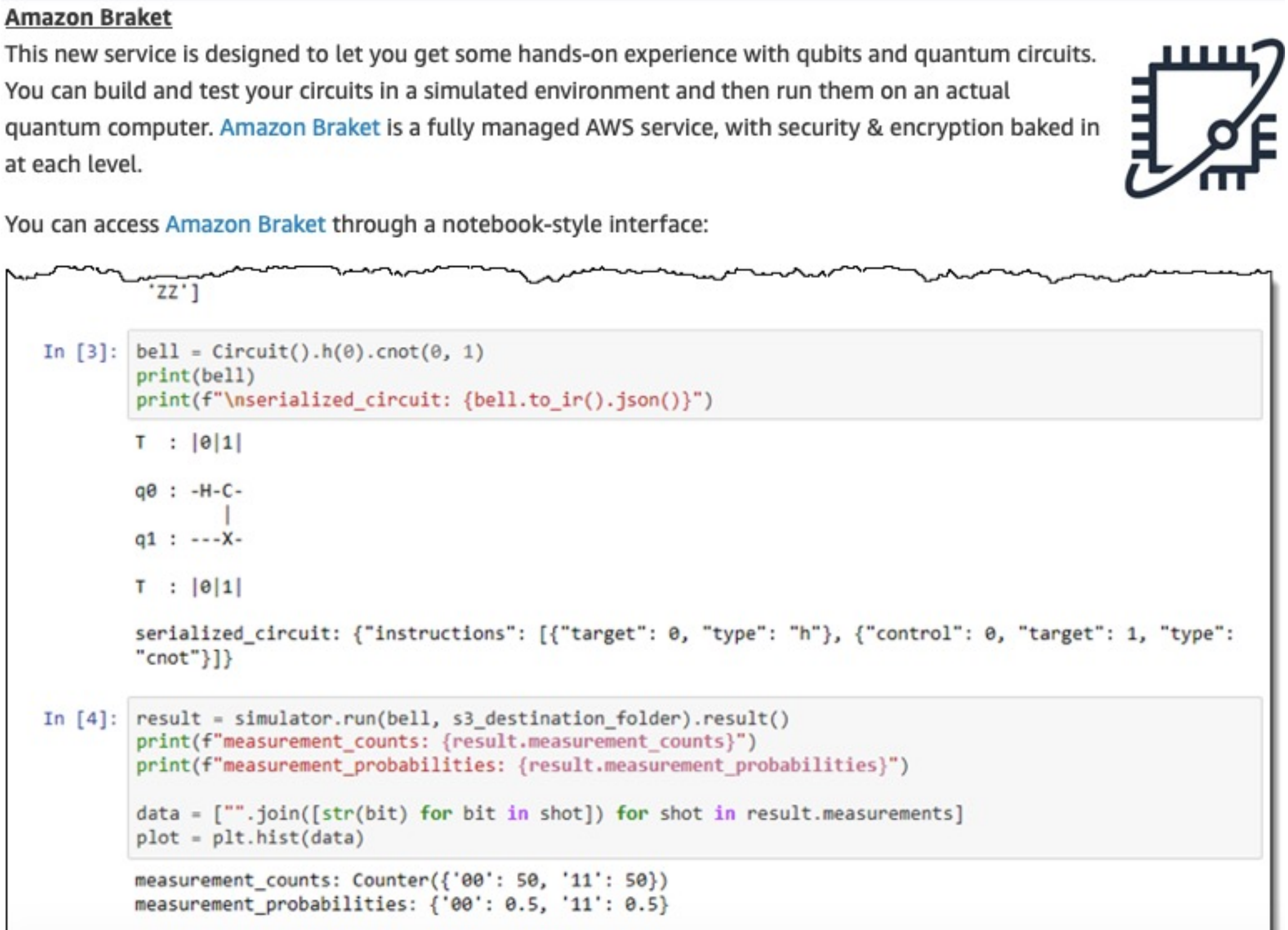
Nearly a decade ago I wrote about the Quantum Compute Cloud on April Fool's Day. The future has arrived and you now have the opportunity to write quantum algorithms and to run them on actual quantum computers. Here's what we are announcing today:

Amazon Braket – A fully managed service that allows scientists, researchers, and developers to begin experimenting with computers from multiple quantum hardware providers in a single place. Bra-ket notation is commonly used to denote quantum mechanical states, and inspired the name of the service.

AWS Center for Quantum Computing – A research center adjacent to the California Institute of Technology (Caltech) that will bring together the world's leading quantum computing researchers and engineers in order to accelerate development of quantum computing hardware and software.

Amazon Quantum Solutions Lab – A new program to connect AWS customers with quantum computing experts from Amazon and a very select set of consulting partners.

Amazon Q1



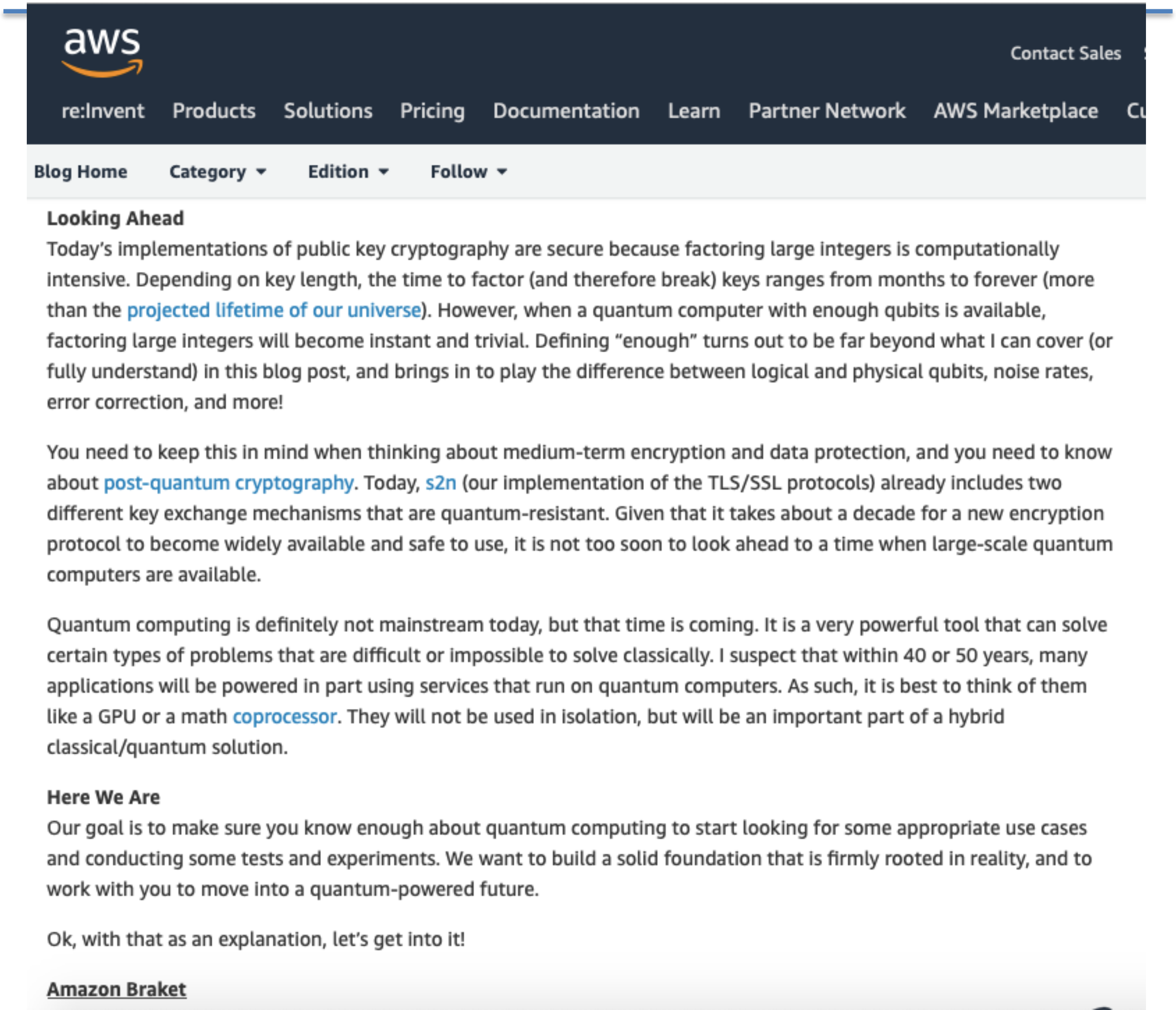
Amazon Braket

This new service is designed to let you get some hands-on experience with qubits and quantum circuits. You can build and test your circuits in a simulated environment and then run them on an actual quantum computer. Amazon Braket is a fully managed AWS service, with security & encryption baked in at each level.

You can access Amazon Braket through a notebook-style interface:

```
  'ZZ']

In [3]: bell = Circuit().h(0).cnot(0, 1)
        print(bell)
        print(f"\nserialized_circuit: {bell.to_ir().json()}")

        T  : |0|1|

        q0 : -H-C-
                |
        q1 : ---X-

        T  : |0|1|

        serialized_circuit: {"instructions": [{"target": 0, "type": "h"}, {"control": 0, "target": 1, "type":
        "cnot"}]}

In [4]: result = simulator.run(bell, s3_destination_folder).result()
        print(f"measurement_counts: {result.measurement_counts}")
        print(f"measurement_probabilities: {result.measurement_probabilities}")

        data = ["".join([str(bit) for bit in shot]) for shot in result.measurements]
        plot = plt.hist(data)

        measurement_counts: Counter({'00': 50, '11': 50})
        measurement_probabilities: {'00': 0.5, '11': 0.5}
```

Amazon Q1



Looking Ahead

Today's implementations of public key cryptography are secure because factoring large integers is computationally intensive. Depending on key length, the time to factor (and therefore break) keys ranges from months to forever (more than the projected lifetime of our universe). However, when a quantum computer with enough qubits is available, factoring large integers will become instant and trivial. Defining "enough" turns out to be far beyond what I can cover (or fully understand) in this blog post, and brings in to play the difference between logical and physical qubits, noise rates, error correction, and more!

You need to keep this in mind when thinking about medium-term encryption and data protection, and you need to know about post-quantum cryptography. Today, s2n (our implementation of the TLS/SSL protocols) already includes two different key exchange mechanisms that are quantum-resistant. Given that it takes about a decade for a new encryption protocol to become widely available and safe to use, it is not too soon to look ahead to a time when large-scale quantum computers are available.

Quantum computing is definitely not mainstream today, but that time is coming. It is a very powerful tool that can solve certain types of problems that are difficult or impossible to solve classically. I suspect that within 40 or 50 years, many applications will be powered in part using services that run on quantum computers. As such, it is best to think of them like a GPU or a math coprocessor. They will not be used in isolation, but will be an important part of a hybrid classical/quantum solution.

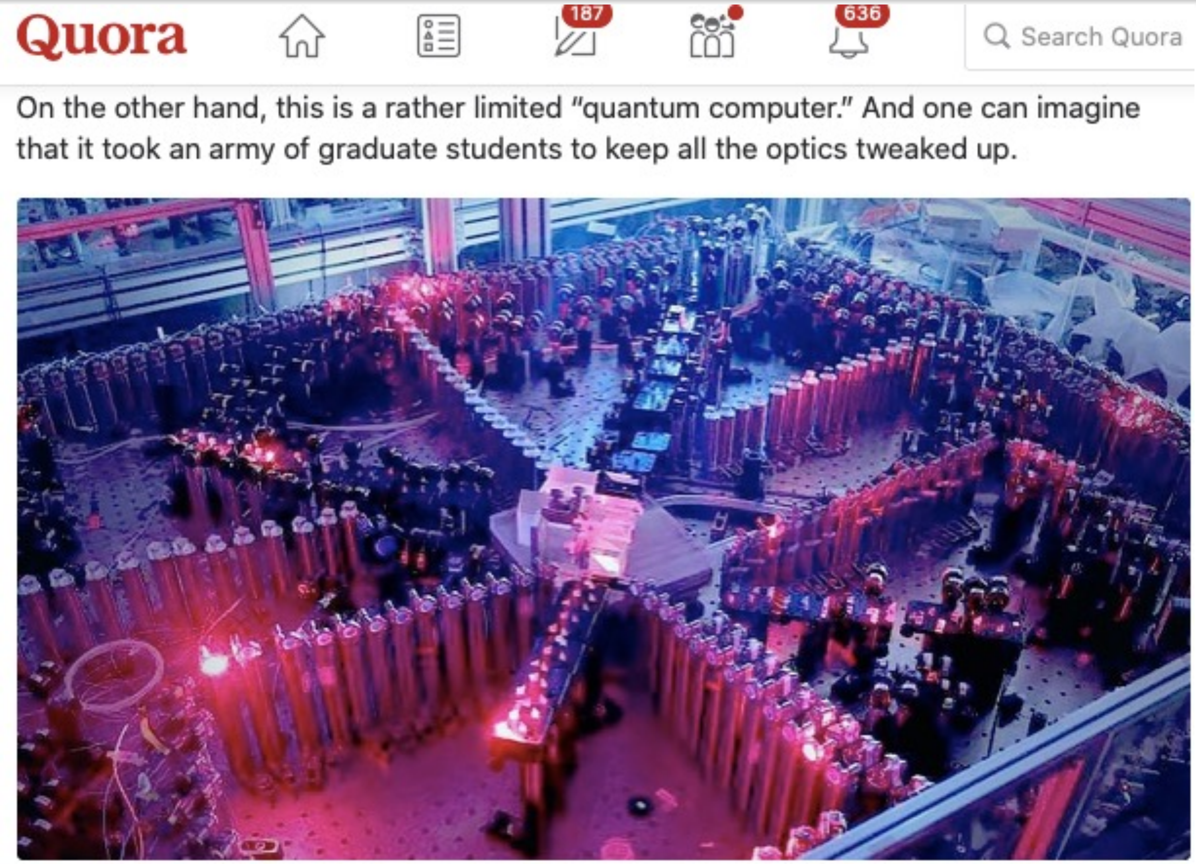
Here We Are

Our goal is to make sure you know enough about quantum computing to start looking for some appropriate use cases and conducting some tests and experiments. We want to build a solid foundation that is firmly rooted in reality, and to work with you to move into a quantum-powered future.

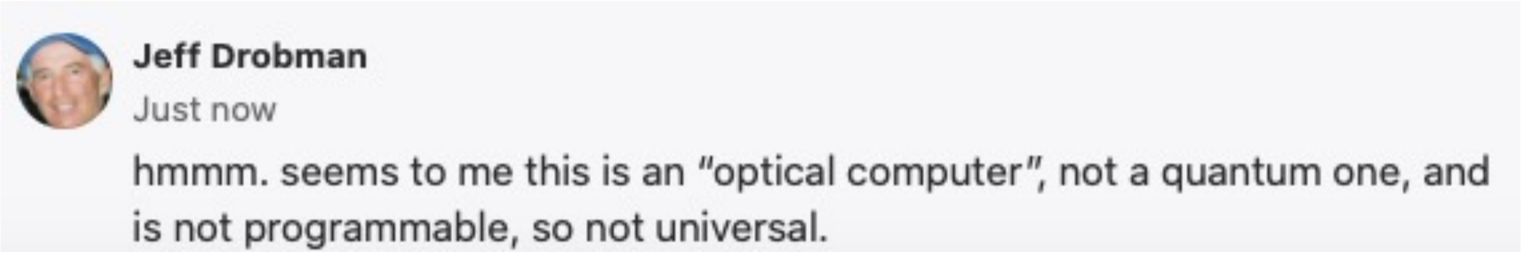
Ok, with that as an explanation, let's get into it!

Amazon Braket

Chinese QC/Optical



On the other hand, this is a rather limited "quantum computer." And one can imagine that it took an army of graduate students to keep all the optics tweaked up.



Jeff Drobman

Just now

hmmm. seems to me this is an "optical computer", not a quantum one, and is not programmable, so not universal.

QC

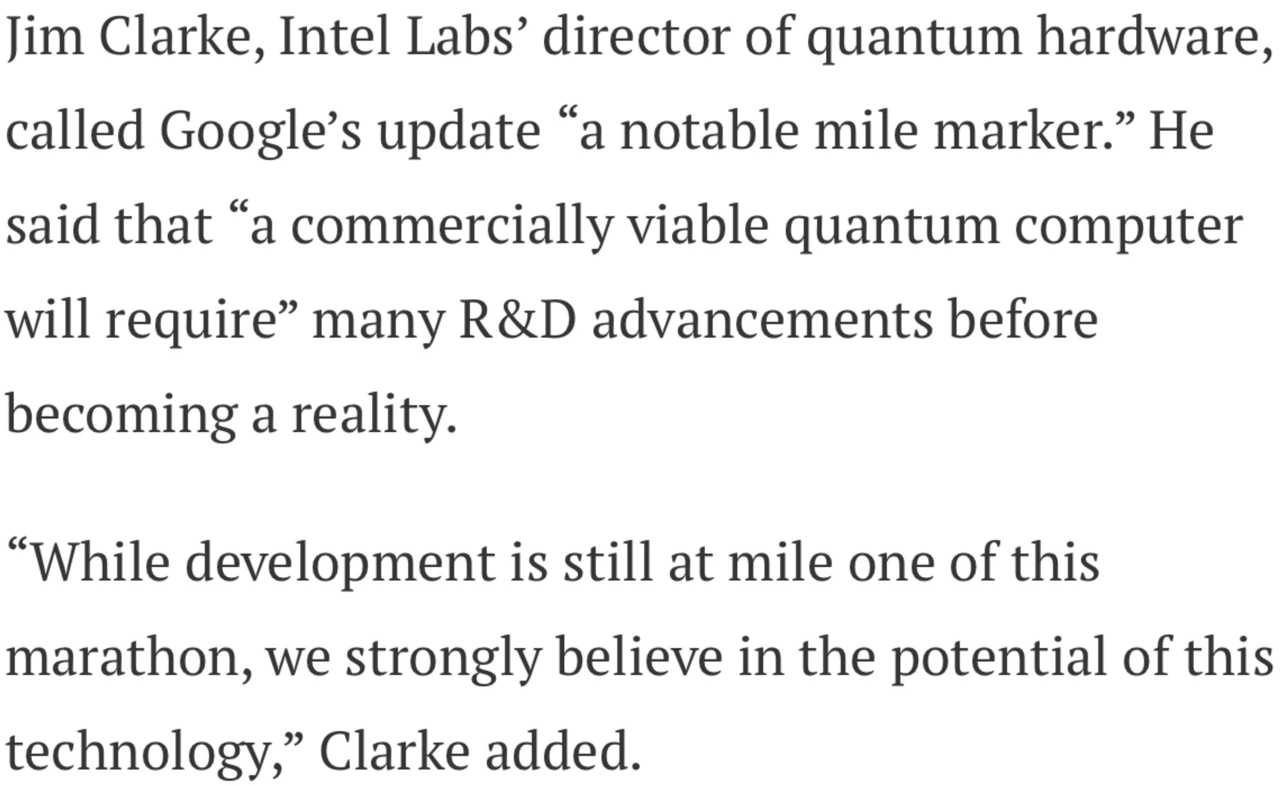
DR JEFF SOFTWARE
INDIE APP DEVELOPER

Intel

Building Blocks

Intel QC

Sep 2019

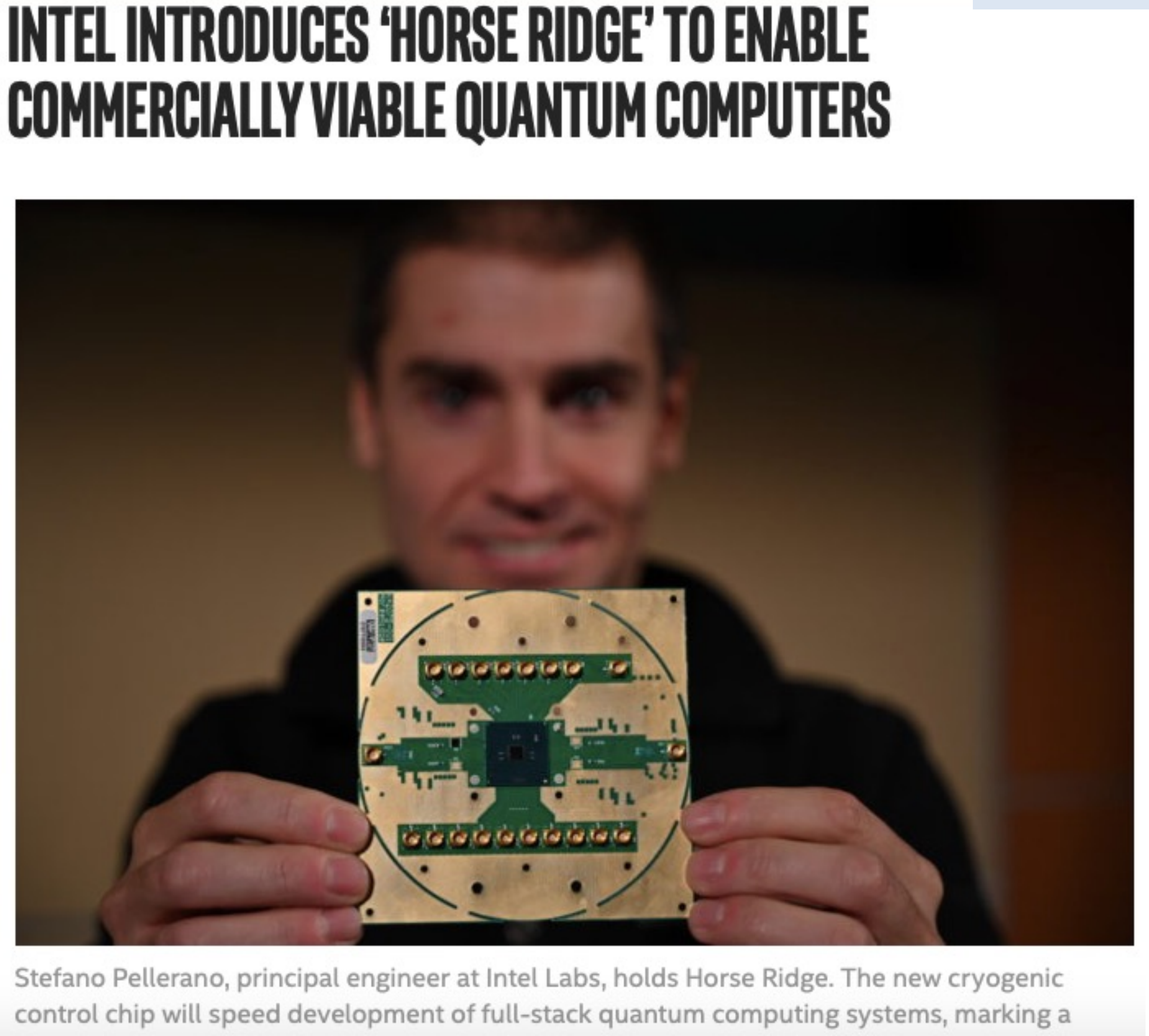


Jim Clarke, Intel Labs' director of quantum hardware, called Google's update "a notable mile marker." He said that "a commercially viable quantum computer will require" many R&D advancements before becoming a reality.

"While development is still at mile one of this marathon, we strongly believe in the potential of this technology," Clarke added.

Intel QC

Dec 2019

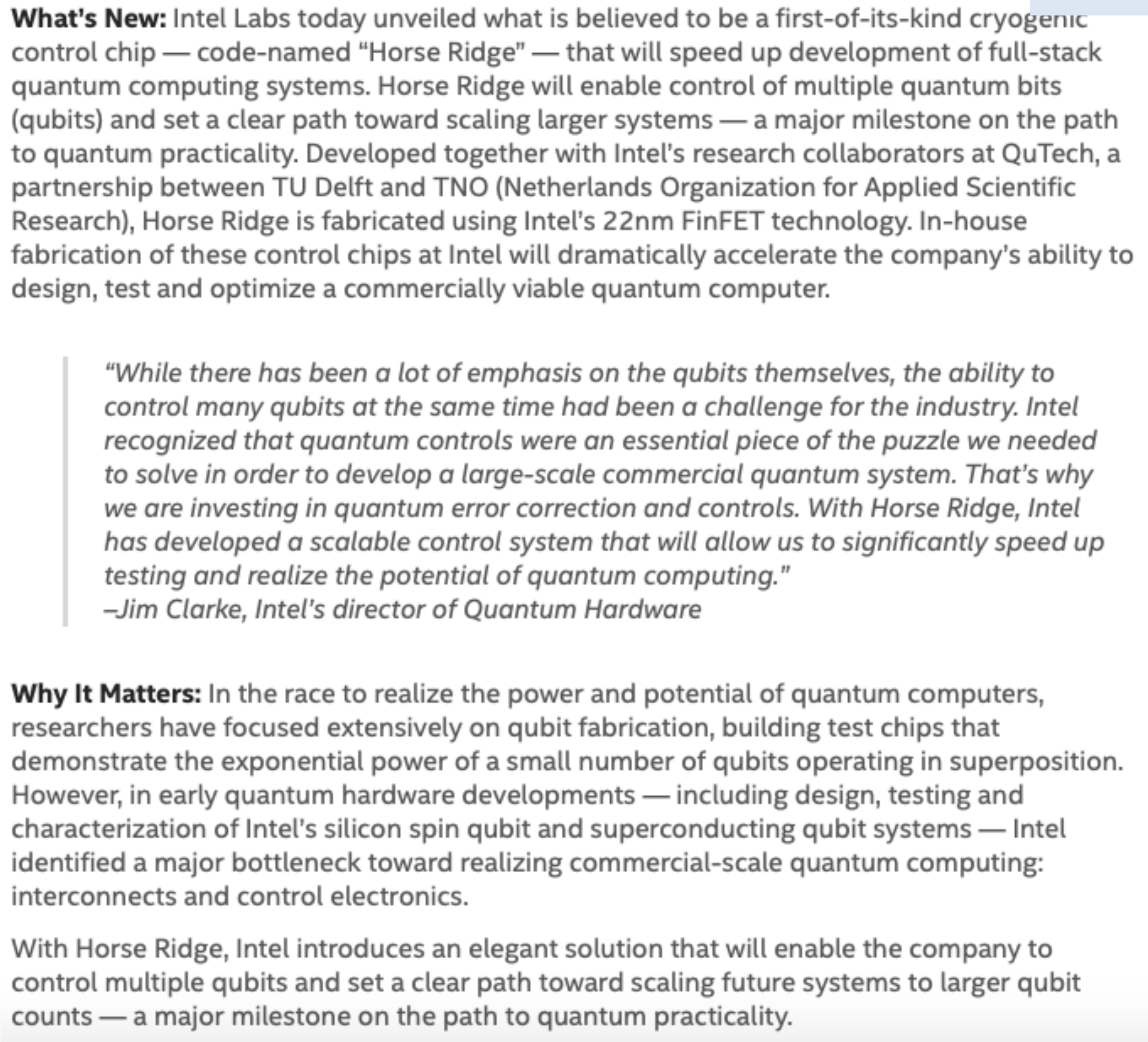


INTEL INTRODUCES 'HORSE RIDGE' TO ENABLE COMMERCIALLY VIABLE QUANTUM COMPUTERS

Stefano Pellerano, principal engineer at Intel Labs, holds Horse Ridge. The new cryogenic control chip will speed development of full-stack quantum computing systems, marking a

Intel QC

Dec 2019



What's New: Intel Labs today unveiled what is believed to be a first-of-its-kind cryogenic control chip — code-named "Horse Ridge" — that will speed up development of full-stack quantum computing systems. Horse Ridge will enable control of multiple quantum bits (qubits) and set a clear path toward scaling larger systems — a major milestone on the path to quantum practicality. Developed together with Intel's research collaborators at QuTech, a partnership between TU Delft and TNO (Netherlands Organization for Applied Scientific Research), Horse Ridge is fabricated using Intel's 22nm FinFET technology. In-house fabrication of these control chips at Intel will dramatically accelerate the company's ability to design, test and optimize a commercially viable quantum computer.

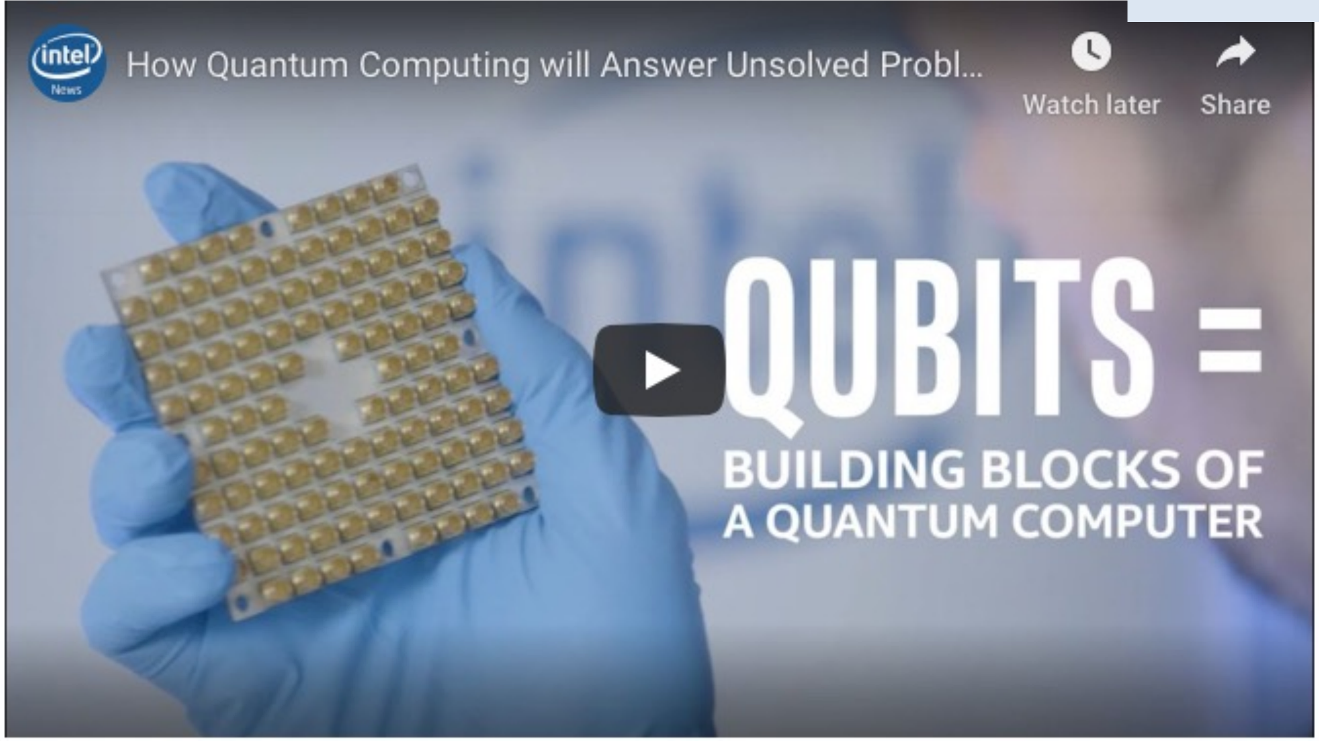
> *"While there has been a lot of emphasis on the qubits themselves, the ability to control many qubits at the same time had been a challenge for the industry. Intel recognized that quantum controls were an essential piece of the puzzle we needed to solve in order to develop a large-scale commercial quantum system. That's why we are investing in quantum error correction and controls. With Horse Ridge, Intel has developed a scalable control system that will allow us to significantly speed up testing and realize the potential of quantum computing."*
> *–Jim Clarke, Intel's director of Quantum Hardware*

Why It Matters: In the race to realize the power and potential of quantum computers, researchers have focused extensively on qubit fabrication, building test chips that demonstrate the exponential power of a small number of qubits operating in superposition. However, in early quantum hardware developments — including design, testing and characterization of Intel's silicon spin qubit and superconducting qubit systems — Intel identified a major bottleneck toward realizing commercial-scale quantum computing: interconnects and control electronics.

With Horse Ridge, Intel introduces an elegant solution that will enable the company to control multiple qubits and set a clear path toward scaling future systems to larger qubit counts — a major milestone on the path to quantum practicality.

Intel QC

Dec 2019

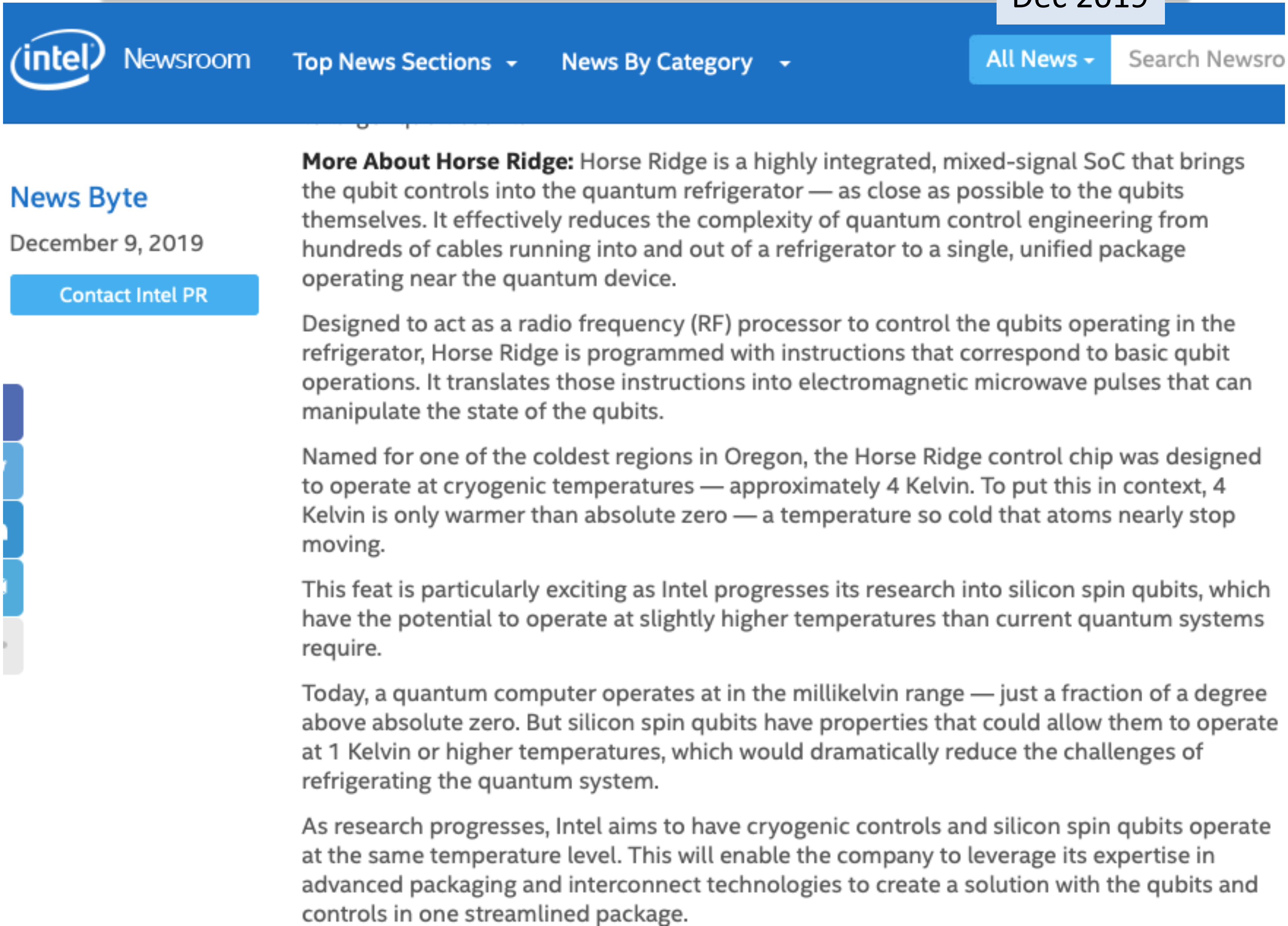


What Quantum Practicality is: Quantum computers promise the potential to tackle problems that conventional computers can't handle by leveraging a phenomena of quantum physics that allows qubits to exist in multiple states simultaneously. As a result, qubits can conduct a large number of calculations at the same time — dramatically speeding up complex problem-solving.

The quantum research community is still at mile one of a marathon toward demonstrating quantum practicality, a benchmark against which the quantum research community can determine whether a quantum system can deliver game-changing performance to solve real-world problems. Intel´s investment in quantum computing covers the full hardware and software stack in pursuit of the development and commercialization of a practical, commercially viable quantum system.

Intel QC

Dec 2019



News Byte

December 9, 2019

More About Horse Ridge: Horse Ridge is a highly integrated, mixed-signal SoC that brings the qubit controls into the quantum refrigerator — as close as possible to the qubits themselves. It effectively reduces the complexity of quantum control engineering from hundreds of cables running into and out of a refrigerator to a single, unified package operating near the quantum device.

Designed to act as a radio frequency (RF) processor to control the qubits operating in the refrigerator, Horse Ridge is programmed with instructions that correspond to basic qubit operations. It translates those instructions into electromagnetic microwave pulses that can manipulate the state of the qubits.

Named for one of the coldest regions in Oregon, the Horse Ridge control chip was designed to operate at cryogenic temperatures — approximately 4 Kelvin. To put this in context, 4 Kelvin is only warmer than absolute zero — a temperature so cold that atoms nearly stop moving.

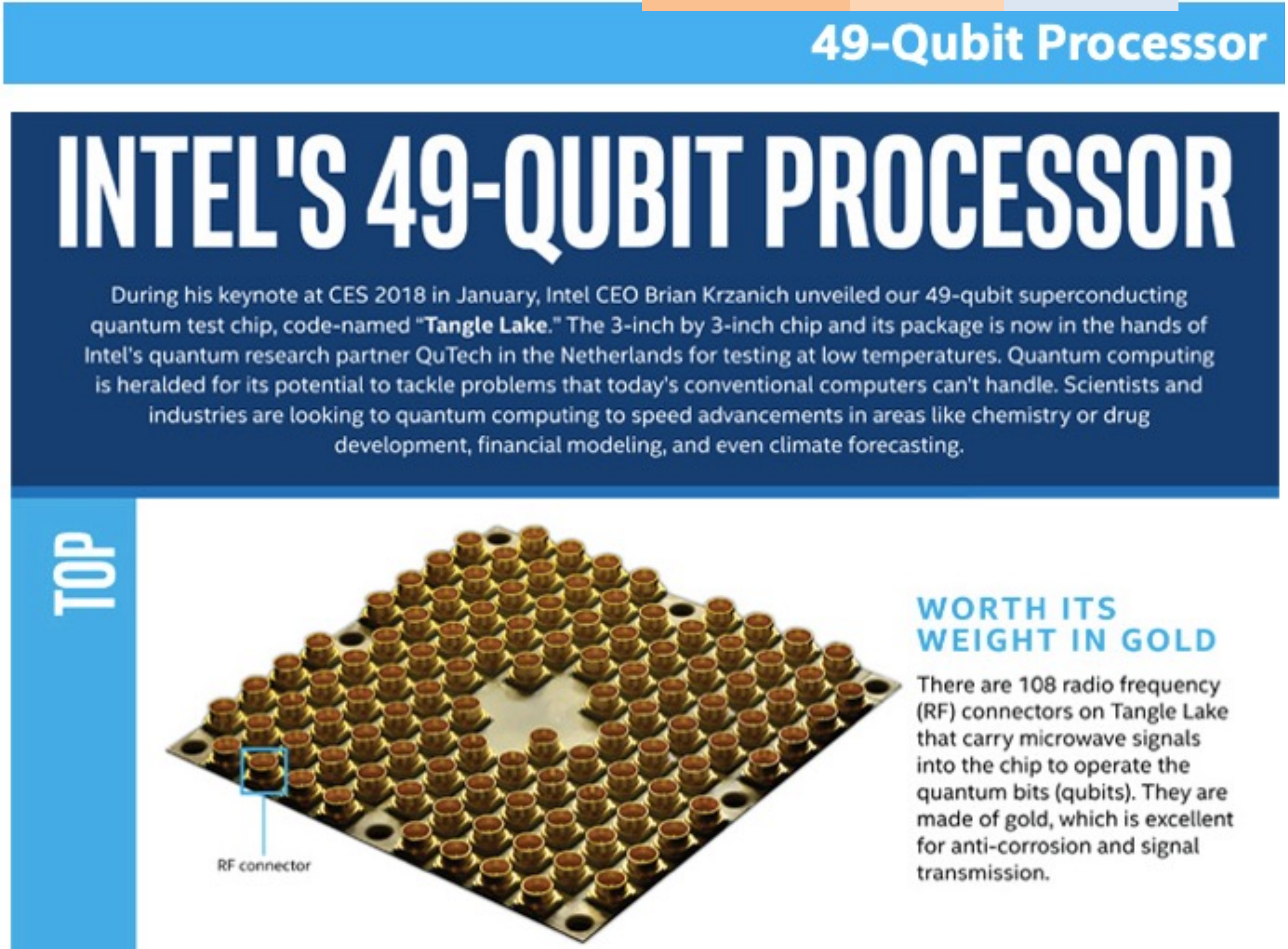
This feat is particularly exciting as Intel progresses its research into silicon spin qubits, which have the potential to operate at slightly higher temperatures than current quantum systems require.

Today, a quantum computer operates at in the millikelvin range — just a fraction of a degree above absolute zero. But silicon spin qubits have properties that could allow them to operate at 1 Kelvin or higher temperatures, which would dramatically reduce the challenges of refrigerating the quantum system.

As research progresses, Intel aims to have cryogenic controls and silicon spin qubits operate at the same temperature level. This will enable the company to leverage its expertise in advanced packaging and interconnect technologies to create a solution with the qubits and controls in one streamlined package.

Intel QC

Tangle Lake	49 qubit	Dec 2020



49-Qubit Processor

INTEL'S 49-QUBIT PROCESSOR

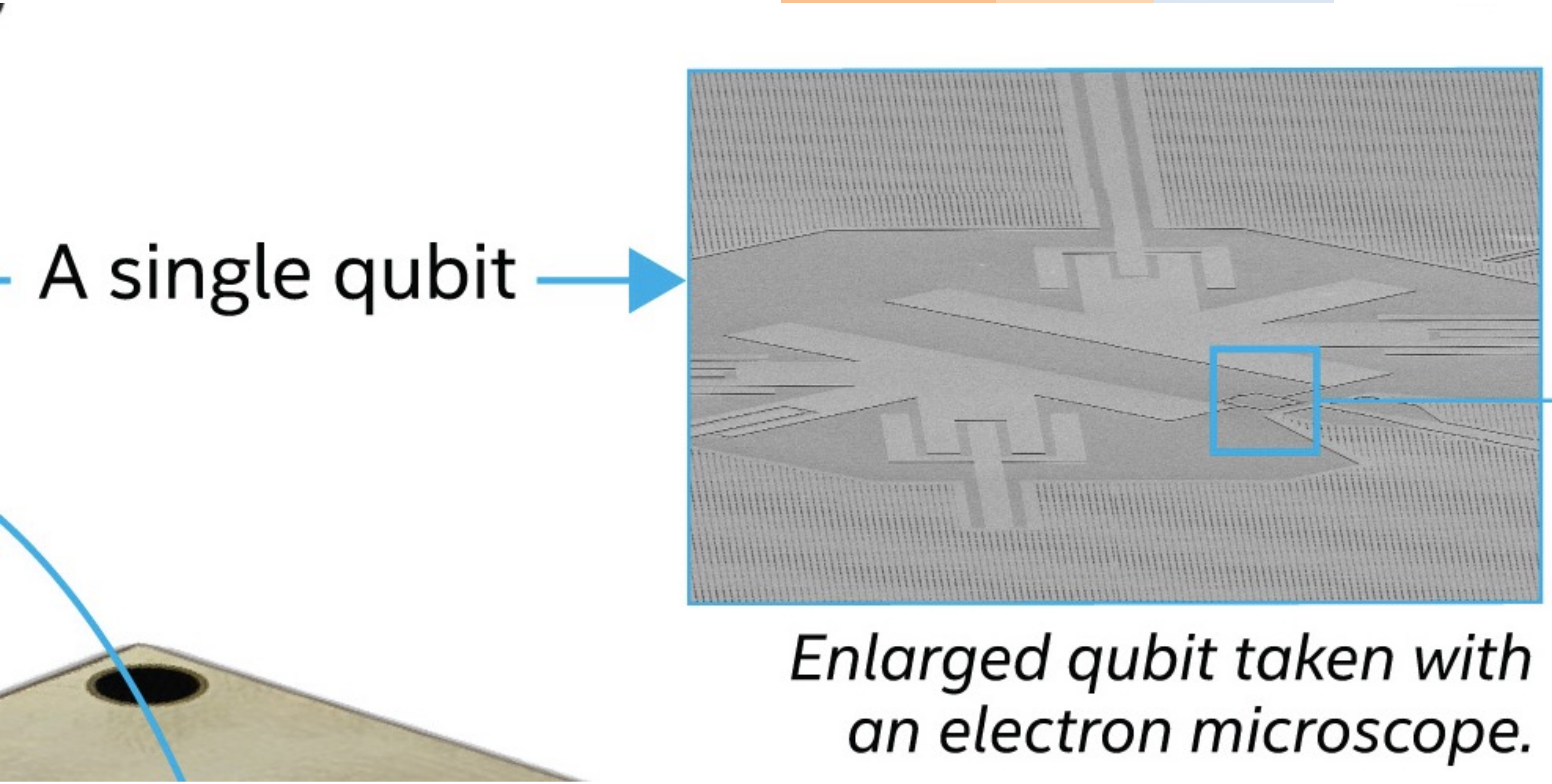
During his keynote at CES 2018 in January, Intel CEO Brian Krzanich unveiled our 49-qubit superconducting quantum test chip, code-named "**Tangle Lake**." The 3-inch by 3-inch chip and its package is now in the hands of Intel's quantum research partner QuTech in the Netherlands for testing at low temperatures. Quantum computing is heralded for its potential to tackle problems that today's conventional computers can't handle. Scientists and industries are looking to quantum computing to speed advancements in areas like chemistry or drug development, financial modeling, and even climate forecasting.

WORTH ITS WEIGHT IN GOLD

There are 108 radio frequency (RF) connectors on Tangle Lake that carry microwave signals into the chip to operate the quantum bits (qubits). They are made of gold, which is excellent for anti-corrosion and signal transmission.

Intel QC

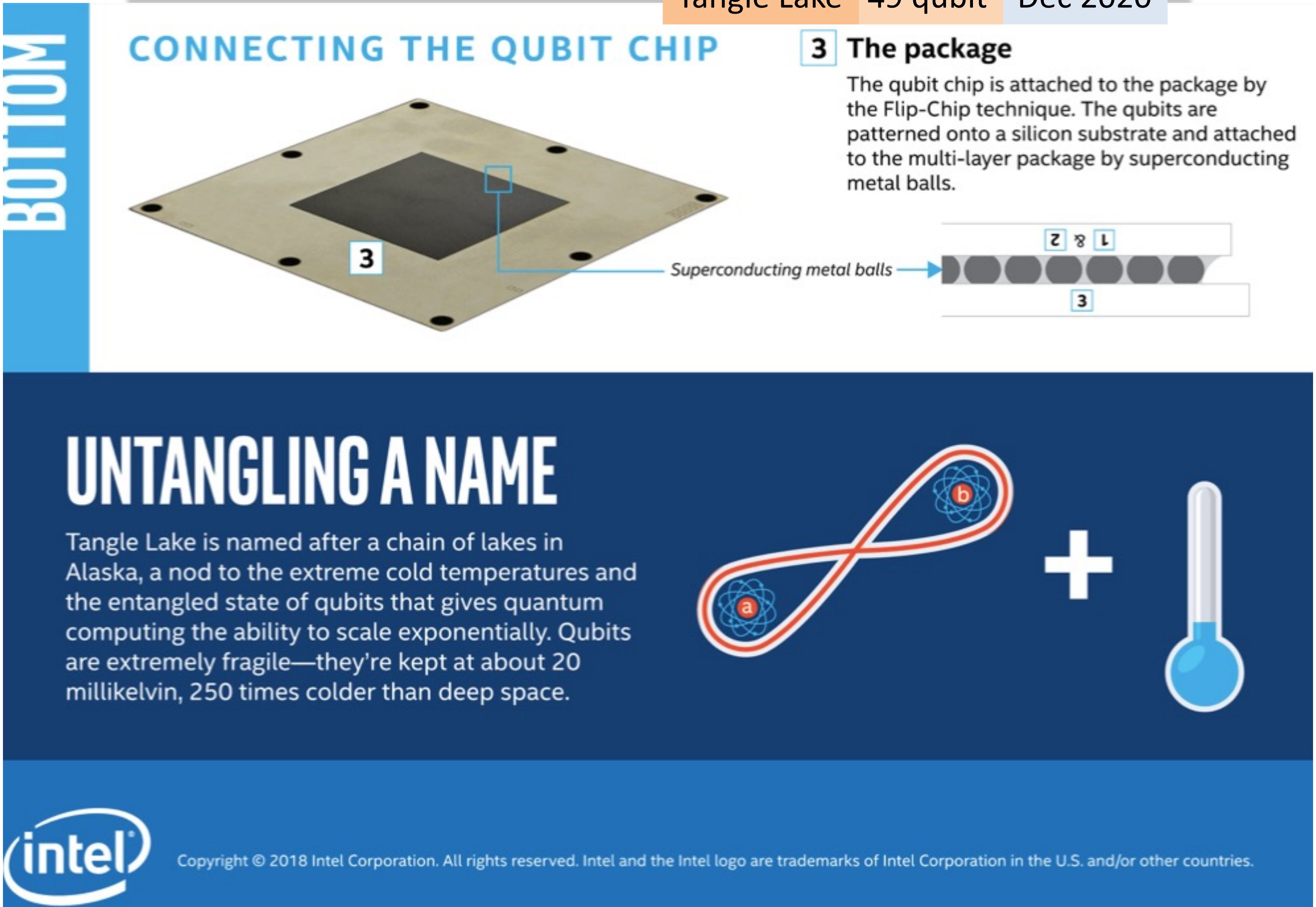
Tangle Lake | 49 qubit | Dec 2020



Enlarged qubit taken with an electron microscope.

Intel QC

Tangle Lake	49 qubit	Dec 2020



CONNECTING THE QUBIT CHIP

3 The package

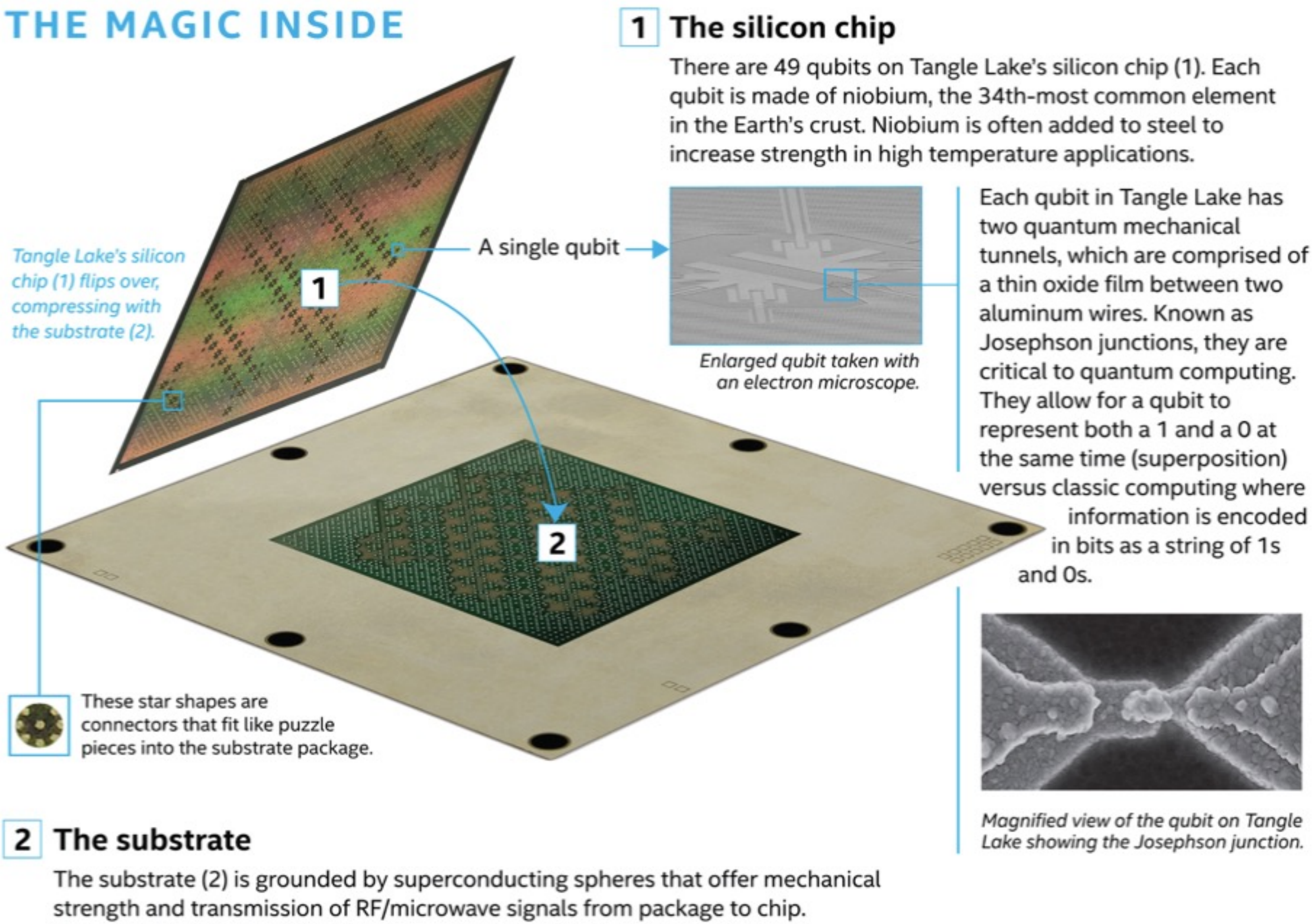
The qubit chip is attached to the package by the Flip-Chip technique. The qubits are patterned onto a silicon substrate and attached to the multi-layer package by superconducting metal balls.

UNTANGLING A NAME

Tangle Lake is named after a chain of lakes in Alaska, a nod to the extreme cold temperatures and the entangled state of qubits that gives quantum computing the ability to scale exponentially. Qubits are extremely fragile—they're kept at about 20 millikelvin, 250 times colder than deep space.

Intel QC

Tangle Lake | 49 qubit | Dec 2020



THE MAGIC INSIDE

Tangle Lake's silicon chip (1) flips over, compressing with the substrate (2).

These star shapes are connectors that fit like puzzle pieces into the substrate package.

1 The silicon chip

There are 49 qubits on Tangle Lake's silicon chip (1). Each qubit is made of niobium, the 34th-most common element in the Earth's crust. Niobium is often added to steel to increase strength in high temperature applications.

Enlarged qubit taken with an electron microscope.

Each qubit in Tangle Lake has two quantum mechanical tunnels, which are comprised of a thin oxide film between two aluminum wires. Known as Josephson junctions, they are critical to quantum computing. They allow for a qubit to represent both a 1 and a 0 at the same time (superposition) versus classic computing where information is encoded in bits as a string of 1s and 0s.

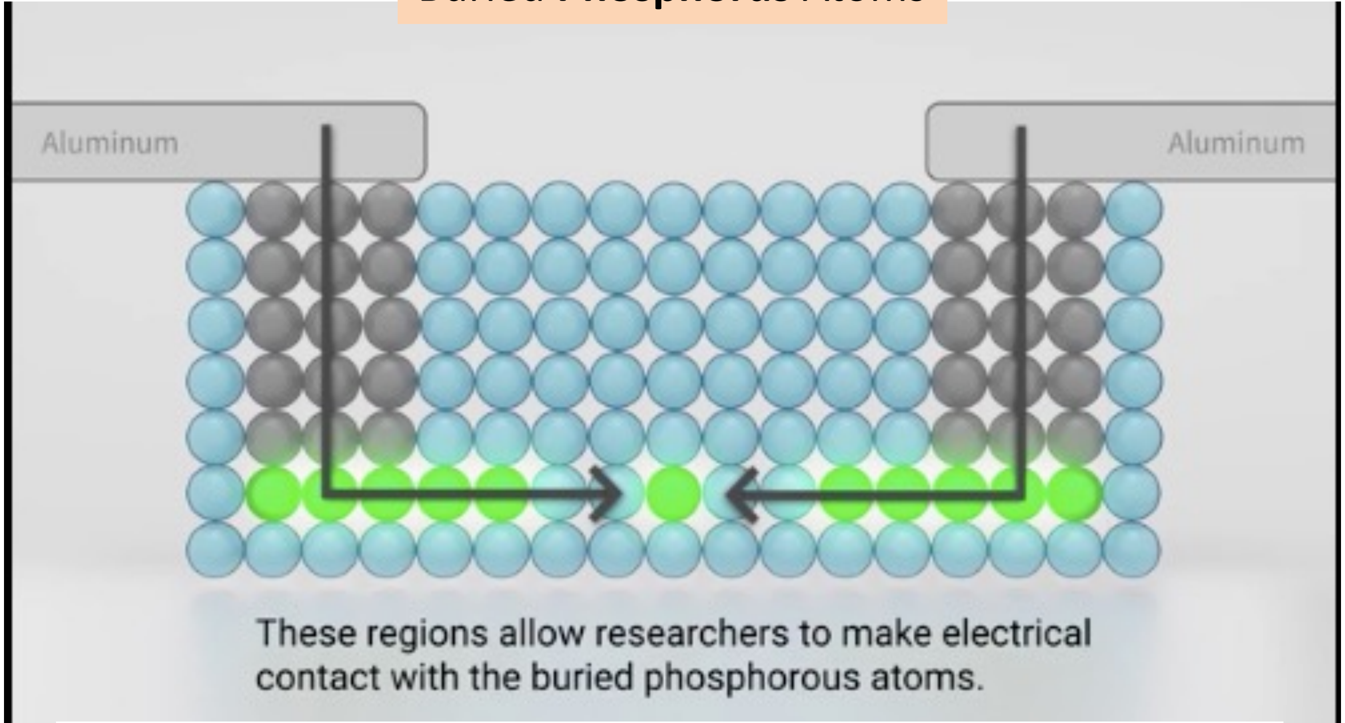
Magnified view of the qubit on Tangle Lake showing the Josephson junction.

2 The substrate

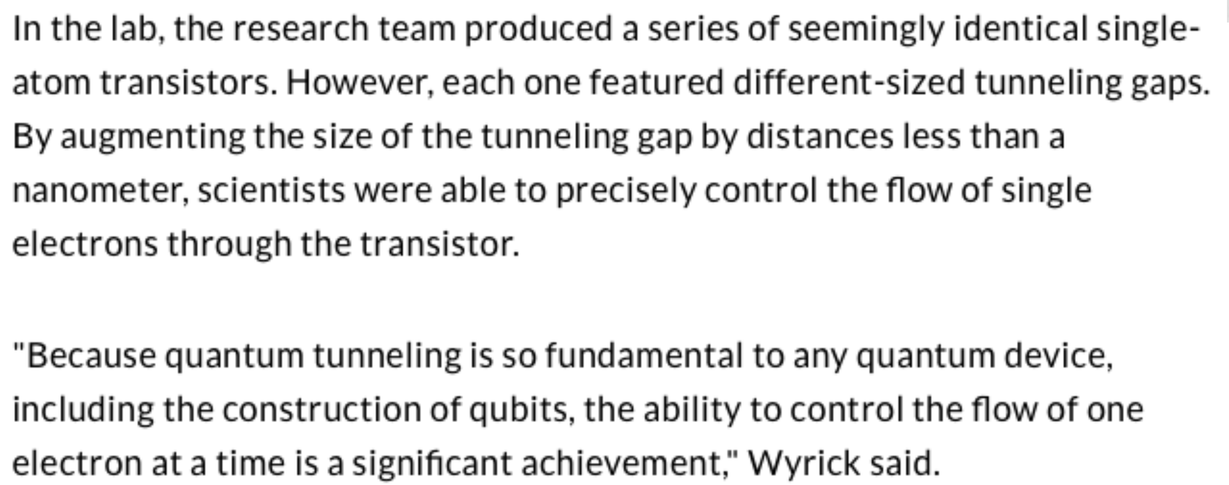
The substrate (2) is grounded by superconducting spheres that offer mechanical strength and transmission of RF/microwave signals from package to chip.

Single Atom Qubits

Buried **Phosphorus** Atoms



These regions allow researchers to make electrical contact with the buried phosphorous atoms.


In the lab, the research team produced a series of seemingly identical single-atom transistors. However, each one featured different-sized tunneling gaps. By augmenting the size of the tunneling gap by distances less than a nanometer, scientists were able to precisely control the flow of single electrons through the transistor.

"Because quantum tunneling is so fundamental to any quantum device, including the construction of qubits, the ability to control the flow of one electron at a time is a significant achievement," Wyrick said.

Quantum Computing

DR JEFF SOFTWARE
INDIE APP DEVELOPER

QC Presentation

CHAPMAN UNIVERSITY | INSTITUTE FOR QUANTUM STUDIES

QC Presentation

CHAPMAN UNIVERSITY | INSTITUTE FOR QUANTUM STUDIES

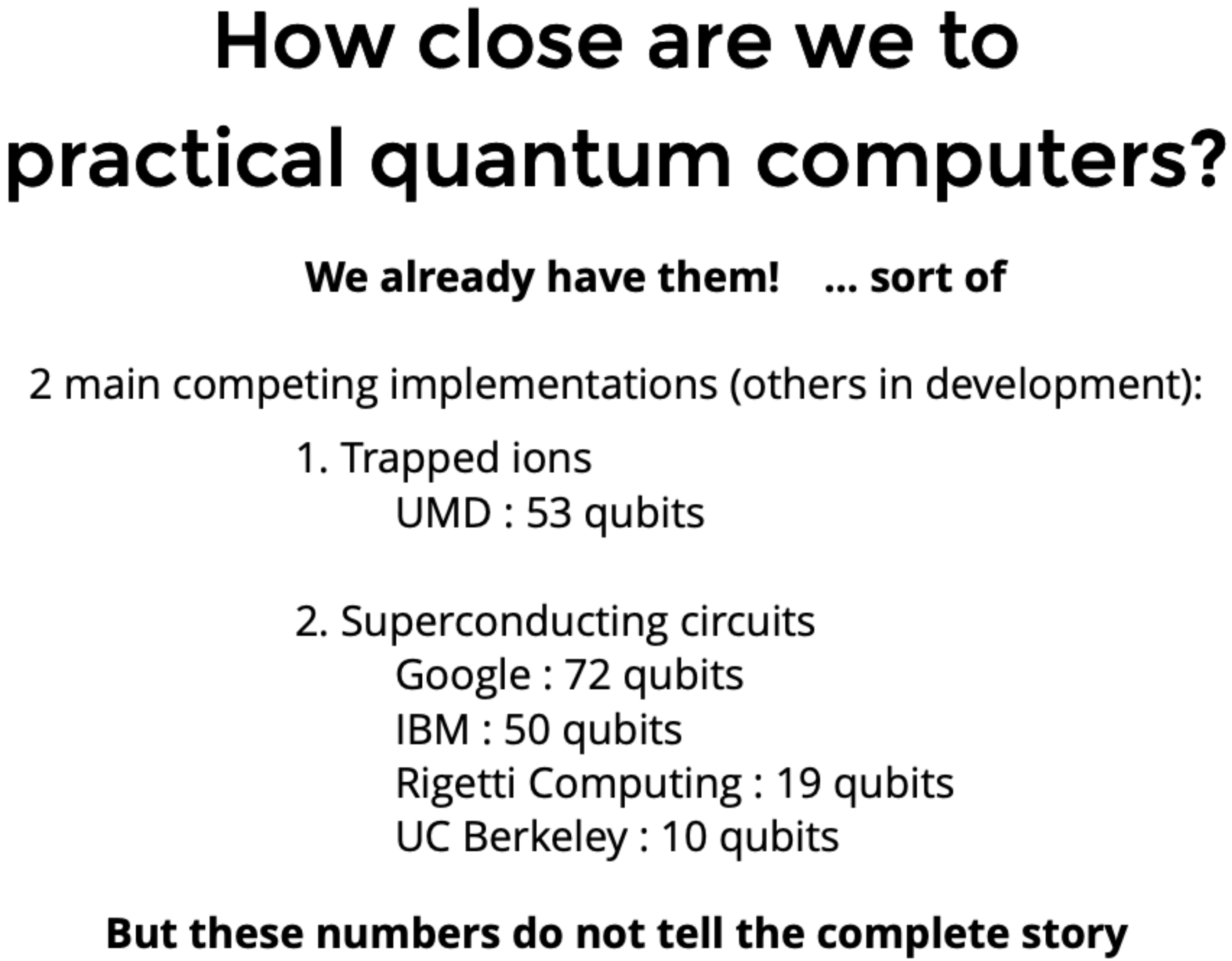
Quantum Computing:

State of Play

Justin Dressel, Ph.D.
Institute for Quantum Studies, Chapman University

OC ACM Chapter Meeting, May 16th, 2018

CHAPMAN UNIVERSITY | INSTITUTE FOR QUANTUM STUDIES



How close are we to practical quantum computers?

We already have them! ... sort of

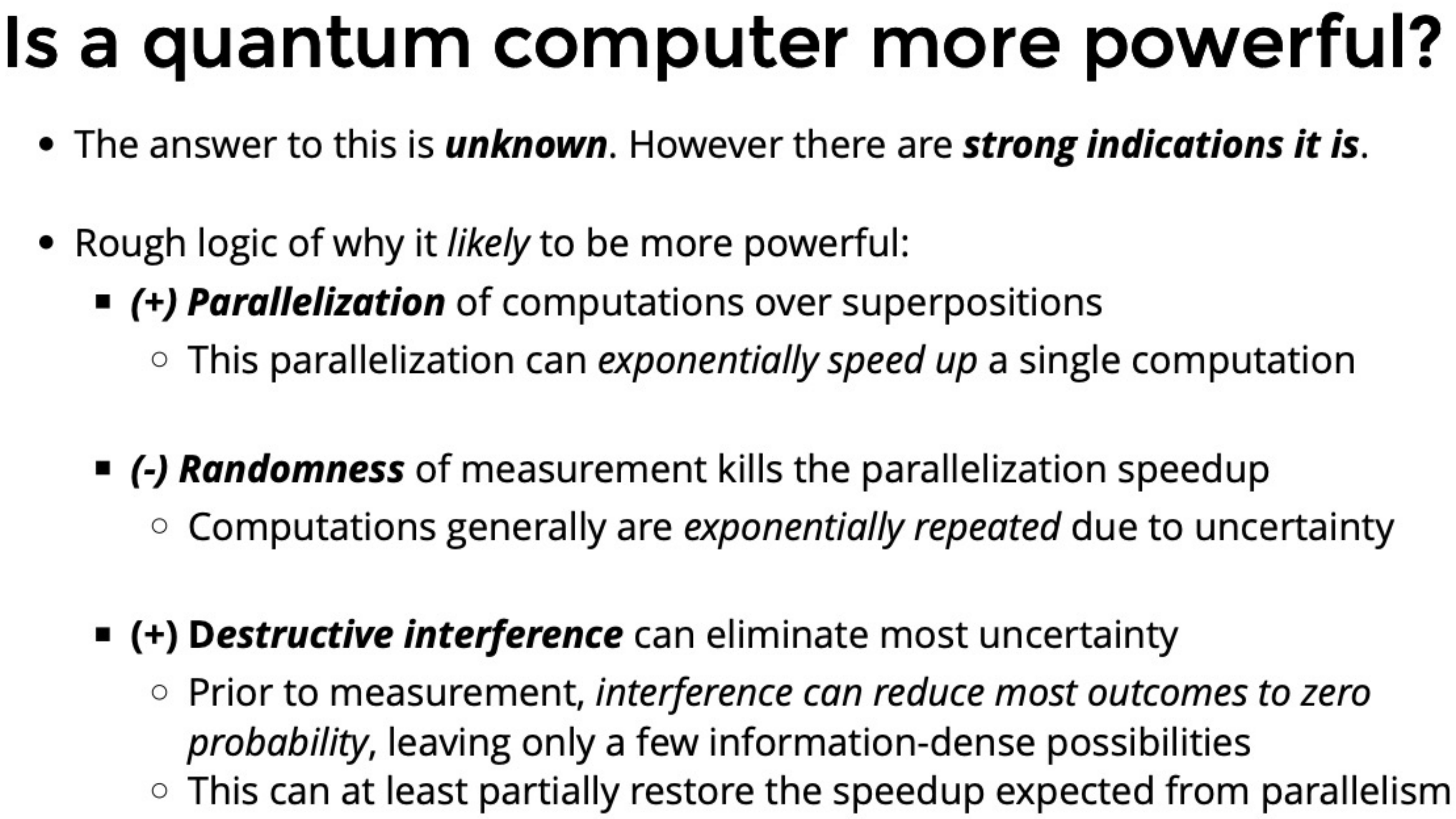
2 main competing implementations (others in development):

1. Trapped ions
   UMD : 53 qubits

2. Superconducting circuits
   Google : 72 qubits
   IBM : 50 qubits
   Rigetti Computing : 19 qubits
   UC Berkeley : 10 qubits

But these numbers do not tell the complete story

QC Presentation



Is a quantum computer more powerful?

- The answer to this is ***unknown***. However there are ***strong indications it is***.

- Rough logic of why it *likely* to be more powerful:
  - ***(+) Parallelization*** of computations over superpositions
    - This parallelization can *exponentially speed up* a single computation
  - ***(-) Randomness*** of measurement kills the parallelization speedup
    - Computations generally are *exponentially repeated* due to uncertainty
  - **(+) D*estructive interference*** can eliminate most uncertainty
    - Prior to measurement, *interference can reduce most outcomes to zero probability*, leaving only a few information-dense possibilities
    - This can at least partially restore the speedup expected from parallelism

CHAPMAN UNIVERSITY | INSTITUTE FOR QUANTUM STUDIES

Quantum Physics and Qubits

New "coherent" features for quantum bits (qubits)

- ***Superpositions*** of 0 and 1 can also be *definite*

  A bit has two possible definite states.

  A qubit has a definite state for each point on the surface of a unit sphere.

- ***Entanglement*** breaks modularity : *More is different*

  1 qubit requires 2 continuous angles to cover its spherical state space

  N qubits require 2^N continuous angles to cover their state space (not 2N)

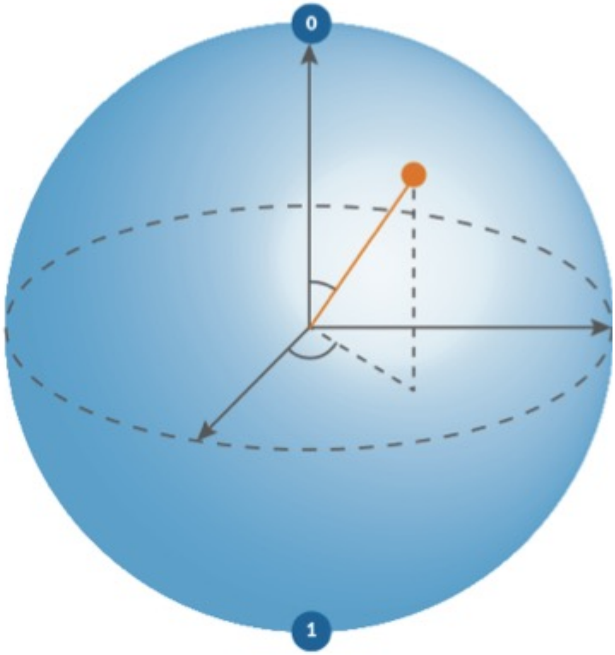
  Exponential scaling of parameters with qubit number, not linear!

- ***Time-symmetry*** : logic gates must be *reversible*

  Qubit states follow *smooth continuous orbits* on the unit sphere

- ***Measurement*** forces *probabilistic* description

  When measured, qubit *randomly* collapses to 0 or 1 based on state proximity



These coherent features wash out (or "decohere") on the macro-scale to produce the classical picture

QC Presentation

Classical Bit Error Correction

$$0 \mapsto 000 \qquad 1 \mapsto 111$$

If one bit flips, can detect and correct via majority-voting

Qubit Error Correction

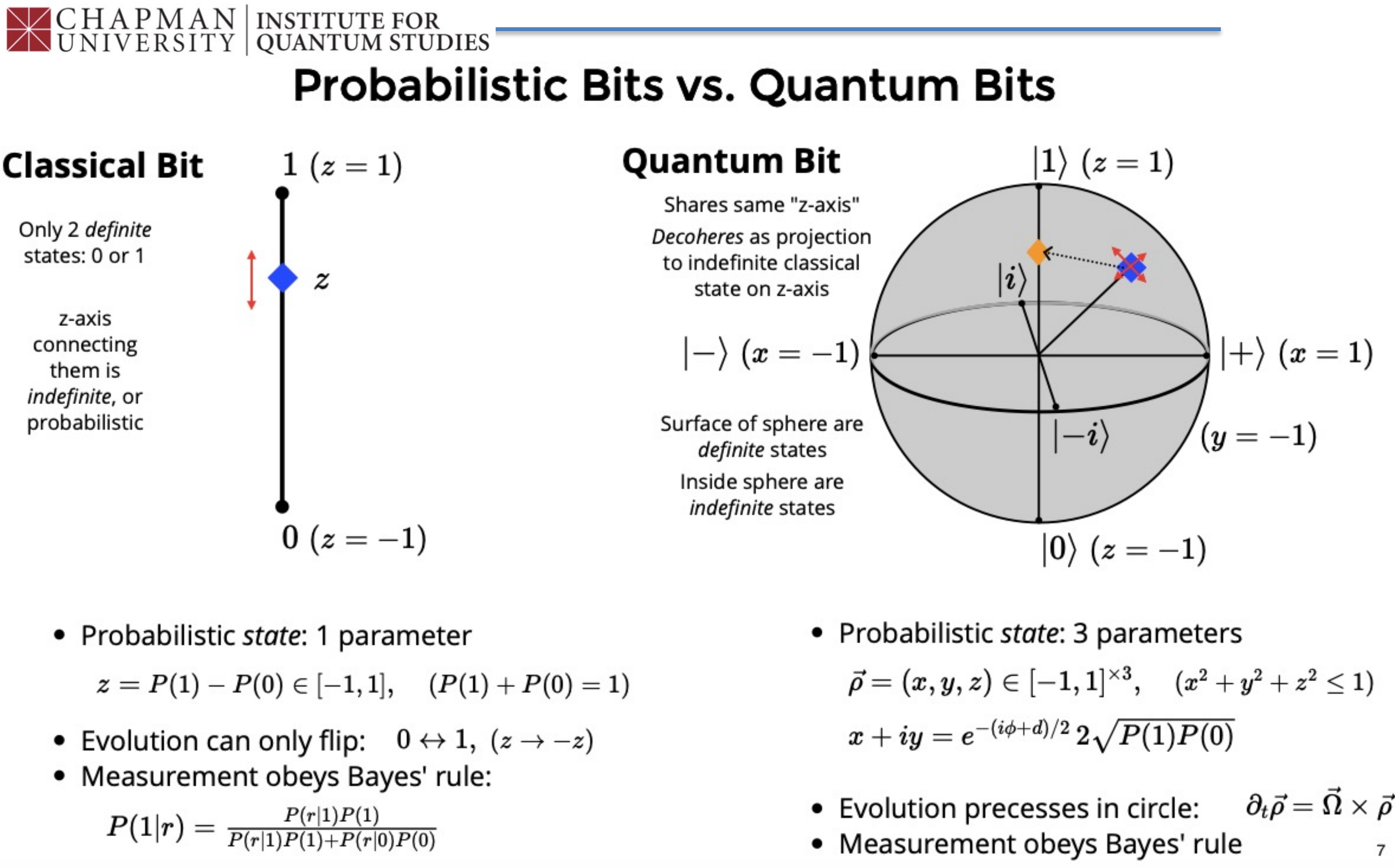
$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \mapsto \alpha|000\rangle + \beta|111\rangle$$

Same basic idea, but now applied to *superpositions*

Main problem: cannot "look" at the bits directly due to measurement collapse

Resolution: measure *parities* of bits instead

QC Presentation



CHAPMAN UNIVERSITY | INSTITUTE FOR QUANTUM STUDIES

Probabilistic Bits vs. Quantum Bits

Classical Bit

Only 2 *definite* states: 0 or 1

z-axis connecting them is *indefinite*, or probabilistic

Quantum Bit

Shares same "z-axis"

Decoheres as projection to indefinite classical state on z-axis

Surface of sphere are *definite* states

Inside sphere are *indefinite* states

- Probabilistic *state*: 1 parameter

$$z = P(1) - P(0) \in [-1,1], \quad (P(1)+P(0)=1)$$

- Evolution can only flip: $0 \leftrightarrow 1,\ (z \to -z)$
- Measurement obeys Bayes' rule:

$$P(1|r) = \frac{P(r|1)P(1)}{P(r|1)P(1)+P(r|0)P(0)}$$

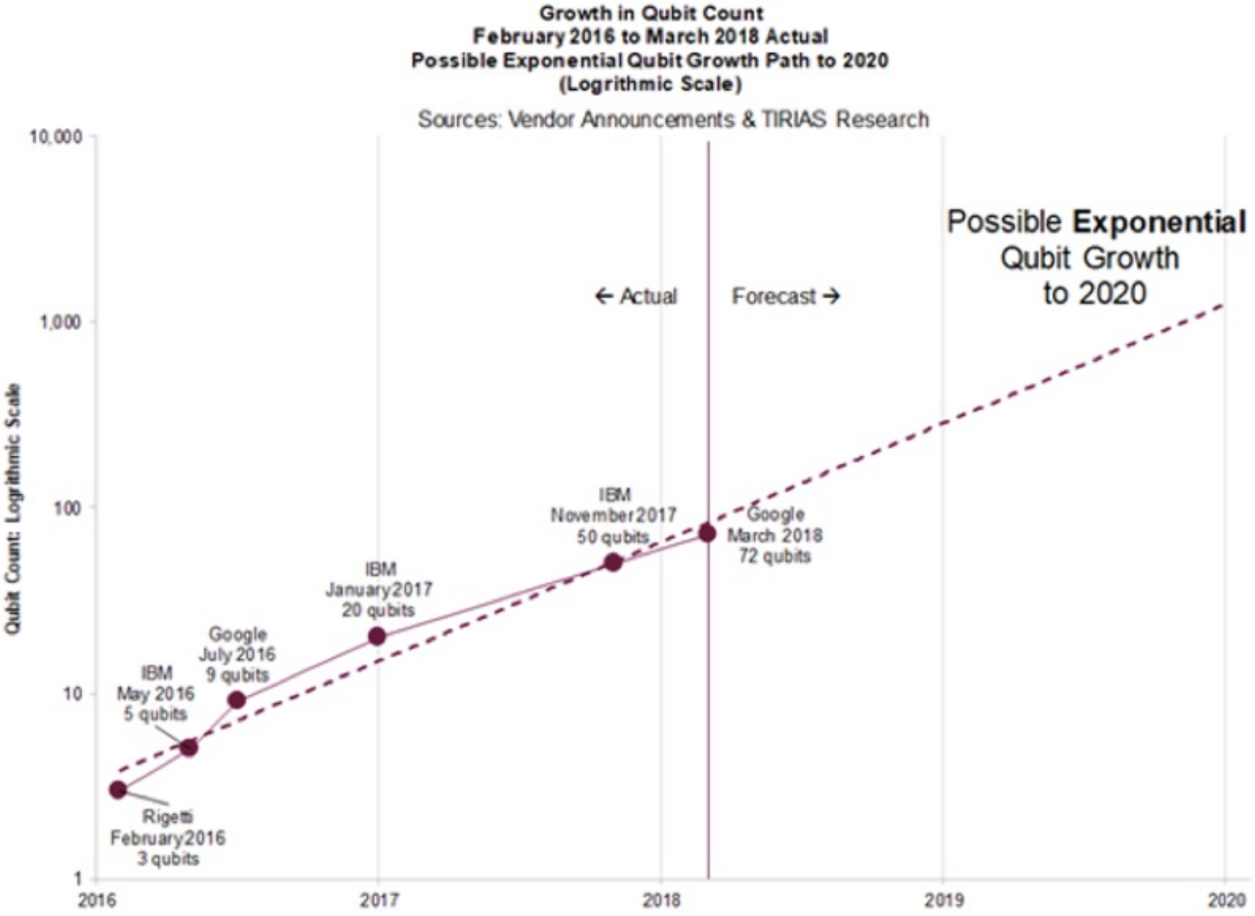
- Probabilistic *state*: 3 parameters

$$\vec{\rho} = (x,y,z) \in [-1,1]^{\times 3}, \quad (x^2+y^2+z^2 \leq 1)$$

$$x + iy = e^{-(i\phi+d)/2}\, 2\sqrt{P(1)P(0)}$$

- Evolution precesses in circle: $\partial_t \vec{\rho} = \vec{\Omega} \times \vec{\rho}$
- Measurement obeys Bayes' rule

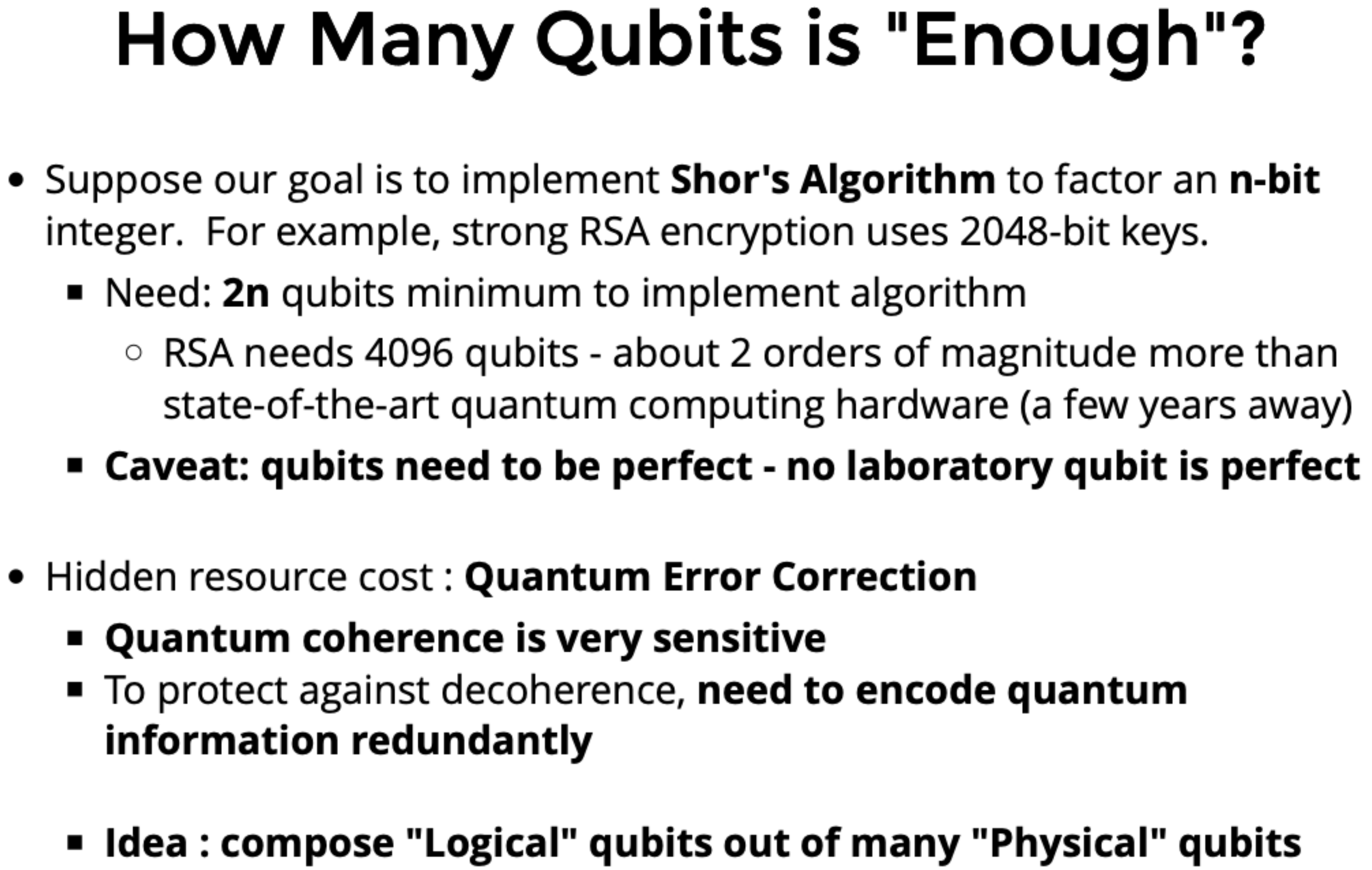
How Long Until A Billion Qubits?



Growth in qubit number is currently **exponential**

If growth continues exponentially (with both fidelity and technical substrate scaling favorably) then we can expect chips with one billion qubits in:

~10-15 years



How Many Qubits is "Enough"?

- Suppose our goal is to implement **Shor's Algorithm** to factor an **n-bit** integer. For example, strong RSA encryption uses 2048-bit keys.
  - Need: **2n** qubits minimum to implement algorithm
    - RSA needs 4096 qubits - about 2 orders of magnitude more than state-of-the-art quantum computing hardware (a few years away)
  - **Caveat: qubits need to be perfect - no laboratory qubit is perfect**

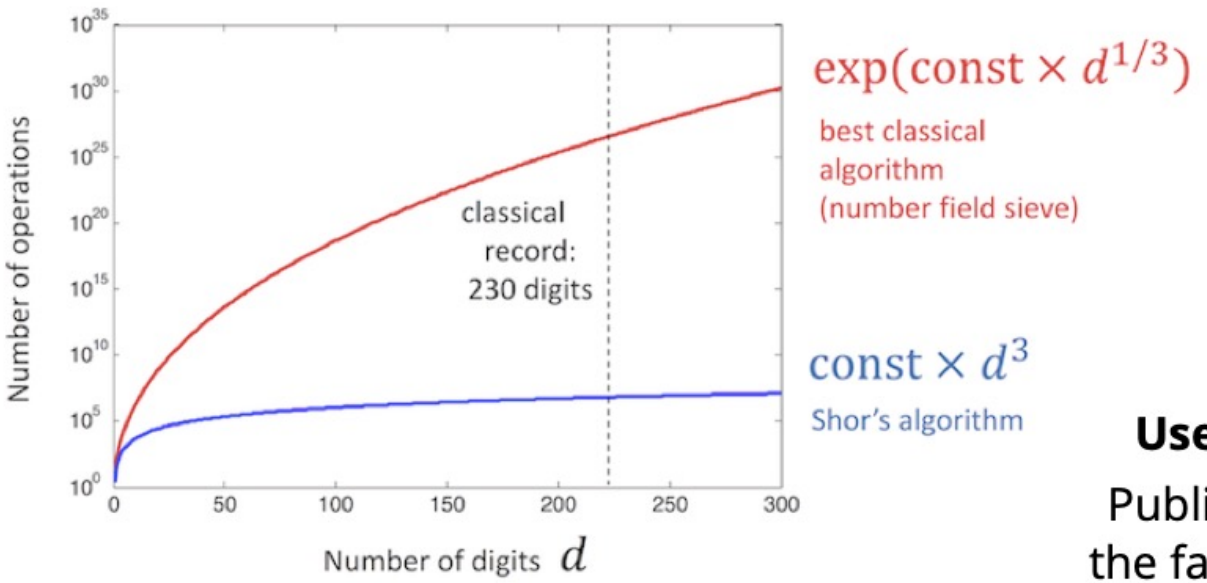
- Hidden resource cost : **Quantum Error Correction**
  - **Quantum coherence is very sensitive**
  - To protect against decoherence, **need to encode quantum information redundantly**

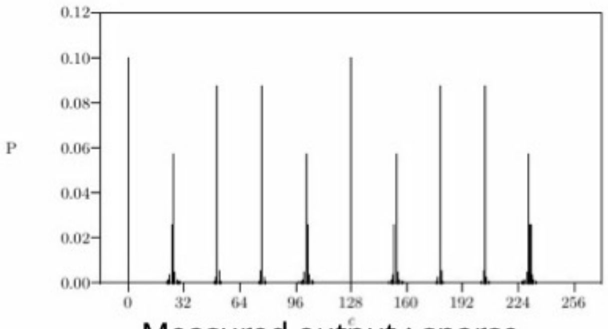
  - **Idea : compose "Logical" qubits out of many "Physical" qubits**

Example: Shor's Algorithm

To **factorize an n-bit integer**, reduce the problem to a period-finding problem, then apply the quantum Fourier transform to exponentially speed it up. Since the resulting superpositions are periodic by construction, the main caveat of the QFT is mitigated.

$$O(e^{1.7(\log n)^{1/3}(\log\log n)^{2/3}}) \text{ (number sieve) } \longrightarrow O((\log n)^2(\log\log n)(\log\log\log n)) \text{ (Shor)}$$





Measured output : sparse, easy to sample

Useful for breaking encryption!

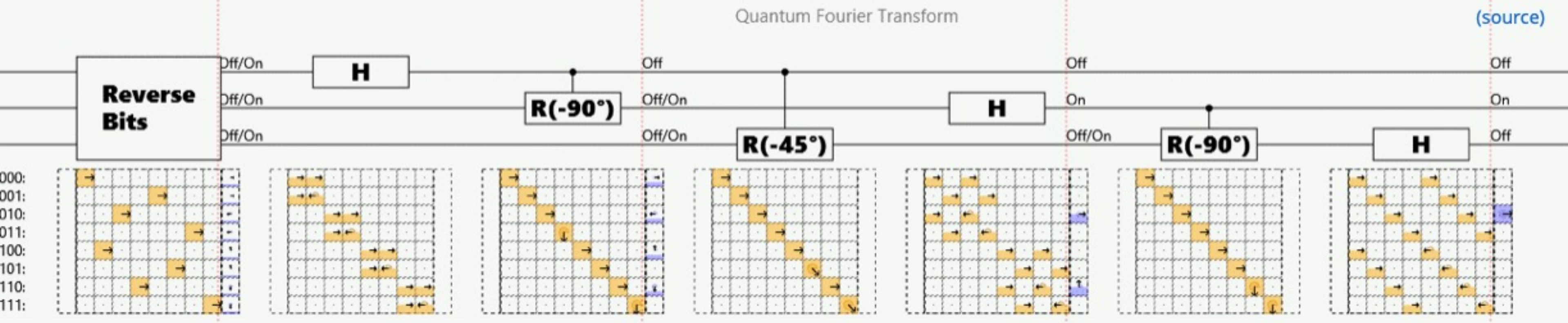
Public key encryption (RSA) relies on the factoring of integers to be difficult

Example: Quantum (Fast) Fourier Transform

Suppose a periodic sequence can be encoded as the amplitudes of a superposition

The quantum Fourier transform (QFT) finds periodicity in polynomial operations

steps per n bits: $2^n(2^{n+1}-1)\ (\mathrm{DFT}) \longrightarrow 3n2^n\ (\mathrm{FFT}) \longrightarrow (n^2+n)/2\ (\mathrm{QFT})$



$$|\psi\rangle = |000\rangle + i|001\rangle - |010\rangle - i|011\rangle + |100\rangle + i|101\rangle - |110\rangle - i|111\rangle$$

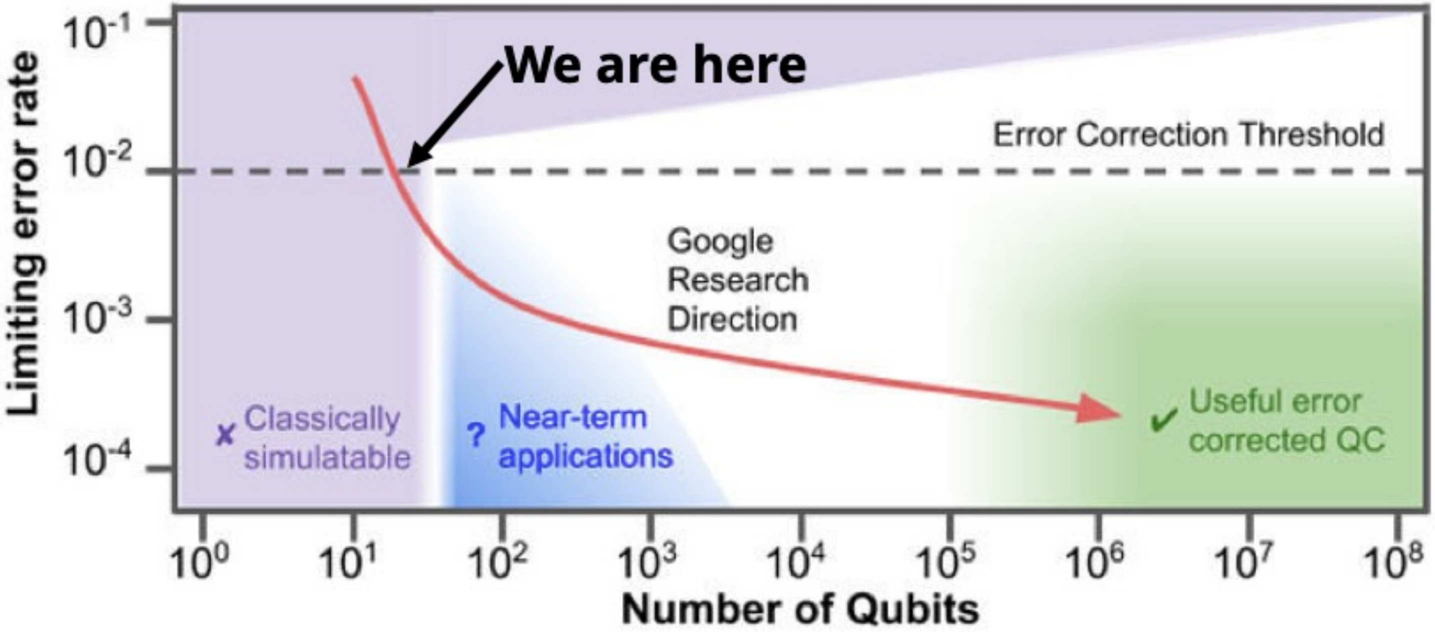
$\Rightarrow \hat{F}|\psi\rangle = |010\rangle$ Detects that each successive phase factor is: $(e^{2\pi i/8})^2$

Caveat: Answer stored as *superposition*. Must *randomly sample outputs* to measure.

QC Presentation

CHAPMAN UNIVERSITY | INSTITUTE FOR QUANTUM STUDIES

What can we do until then?



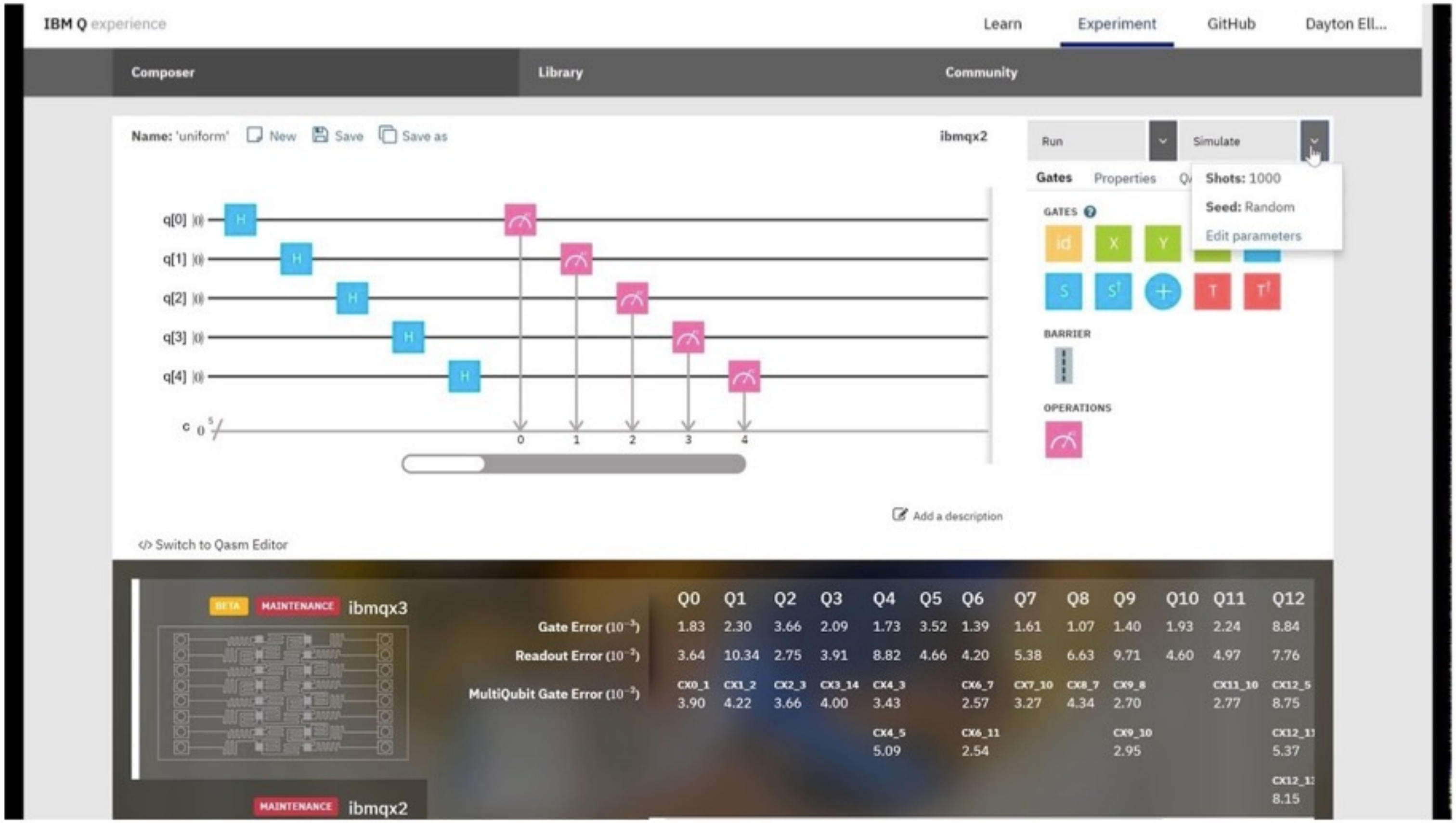
Google Slides

We are now reaching the scale that is no longer possible to simulate using classical supercomputers.

The current challenge is to find "near-term" applications for the existing quantum devices.

QC Presentation

Program a Quantum Computer Now



Name: 'uniform'

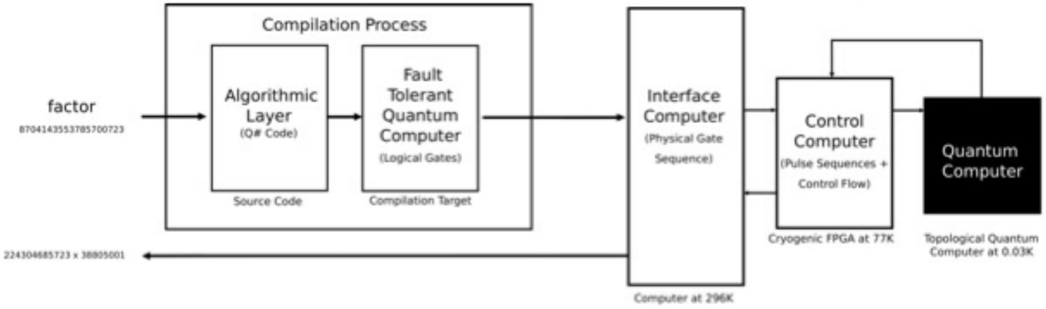
Shots: 1000
Seed: Random
Edit parameters

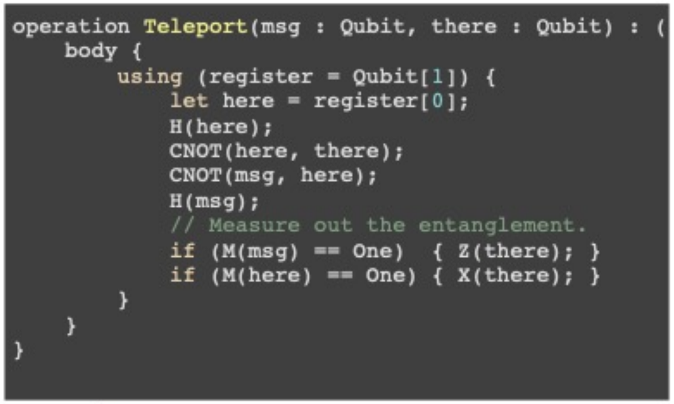
	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12
Gate Error (10^{-3})	1.83	2.30	3.66	2.09	1.73	3.52	1.39	1.61	1.07	1.40	1.93	2.24	8.84
Readout Error (10^{-2})	3.64	10.34	2.75	3.91	8.82	4.66	4.20	5.38	6.63	9.71	4.60	4.97	7.76
MultiQubit Gate Error (10^{-2})	CX0_1 3.90	CX1_2 4.22	CX2_3 3.66	CX3_14 4.00	CX4_3 3.43 CX4_5 5.09		CX6_7 2.57 CX6_11 2.54	CX7_10 3.27	CX8_7 4.34	CX9_8 2.70 CX9_10 2.95		CX11_10 2.77	CX12_5 8.75 CX12_1 5.37 CX12_1 8.15

IBM Quantum Experience : Cloud Computer (16 qubits free, 20+ paid)

Quantum Software Stacks

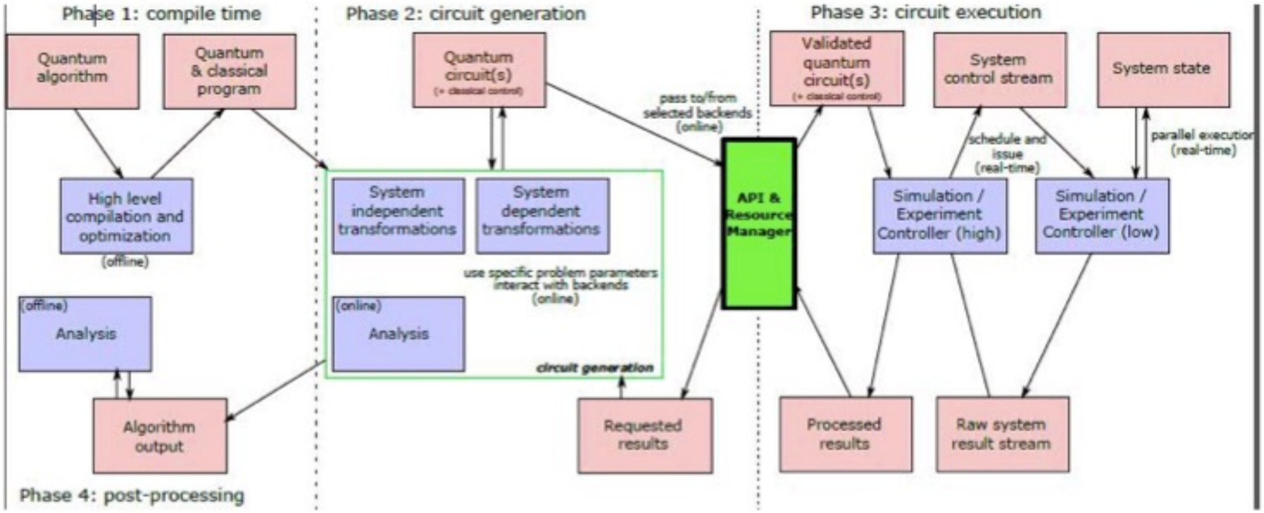
Microsoft : Q#, Quantum Dev Kit, LiQui|>

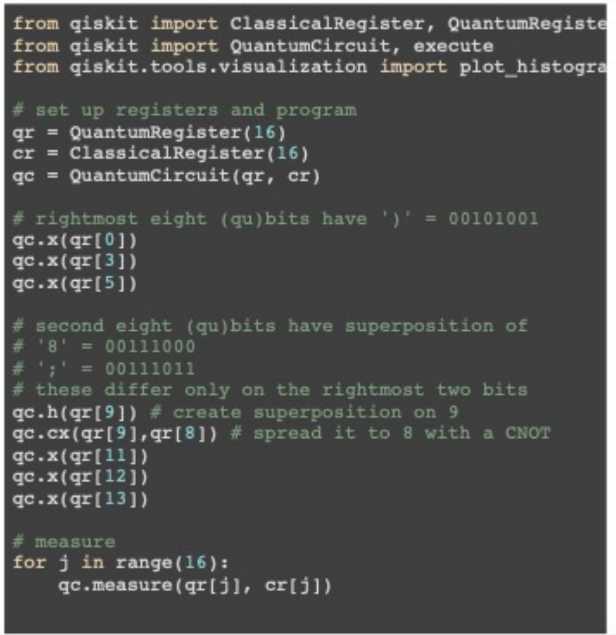




```
operation Teleport(msg : Qubit, there : Qubit) : (
    body {
        using (register = Qubit[1]) {
            let here = register[0];
            H(here);
            CNOT(here, there);
            CNOT(msg, here);
            H(msg);
            // Measure out the entanglement.
            if (M(msg) == One)  { Z(there); }
            if (M(here) == One) { X(there); }
        }
    }
}
```

IBM : QISKit SDK

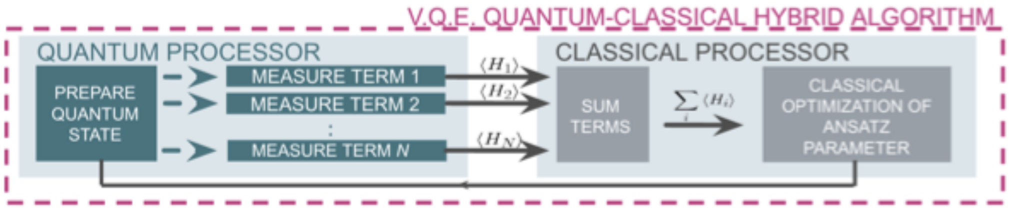

Phase 1: compile time
Phase 2: circuit generation
Phase 3: circuit execution
Phase 4: post-processing
Quantum algorithm
Quantum & classical program
High level compilation and optimization
(offline)
(offline)
Analysis
Algorithm output
Quantum circuit(s)
(+ classical control)
System independent transformations
System dependent transformations
use specific problem parameters
interact with backends
(online)
(online)
Analysis
circuit generation
Requested results
pass to/from selected backends (online)
API & Resource Manager
Validated quantum circuit(s)
(+ classical control)
System control stream
System state
schedule and issue (real-time)
parallel execution (real-time)
Simulation / Experiment Controller (high)
Simulation / Experiment Controller (low)
Processed results
Raw system result stream

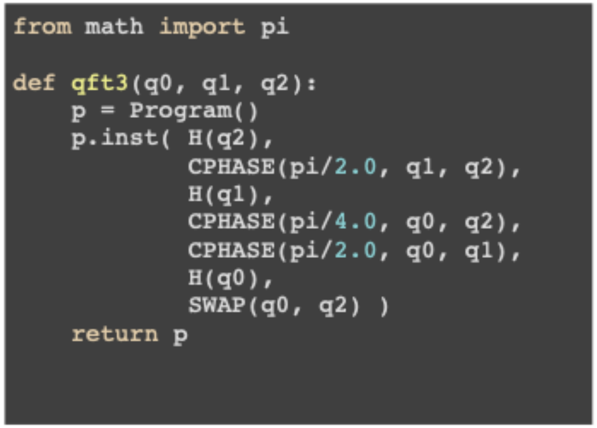


```
from qiskit import ClassicalRegister, QuantumRegiste
from qiskit import QuantumCircuit, execute
from qiskit.tools.visualization import plot_histogra

# set up registers and program
qr = QuantumRegister(16)
cr = ClassicalRegister(16)
qc = QuantumCircuit(qr, cr)

# rightmost eight (qu)bits have ')' = 00101001
qc.x(qr[0])
qc.x(qr[3])
qc.x(qr[5])

# second eight (qu)bits have superposition of
# '8' = 00111000
# ';' = 00111011
# these differ only on the rightmost two bits
qc.h(qr[9]) # create superposition on 9
qc.cx(qr[9],qr[8]) # spread it to 8 with a CNOT
qc.x(qr[11])
qc.x(qr[12])
qc.x(qr[13])

# measure
for j in range(16):
    qc.measure(qr[j], cr[j])
```

More Quantum Software Stacks

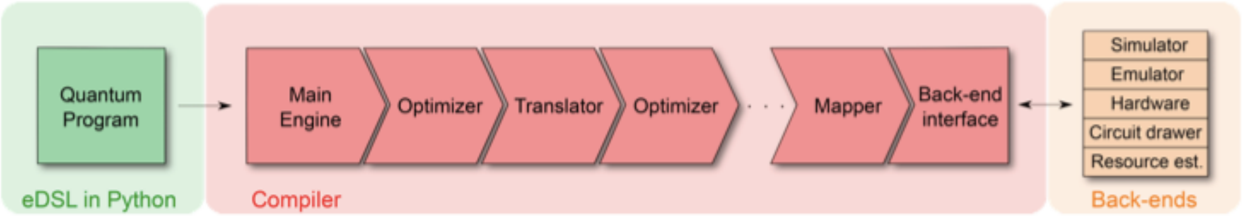
Rigetti Computing : Forest, Quil, PyQuil

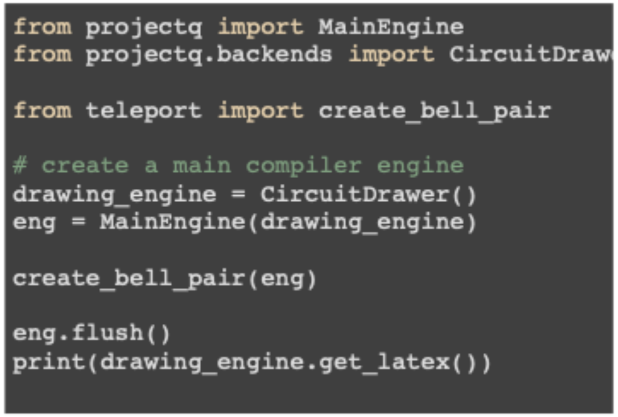




```
from math import pi

def qft3(q0, q1, q2):
    p = Program()
    p.inst( H(q2),
            CPHASE(pi/2.0, q1, q2),
            H(q1),
            CPHASE(pi/4.0, q0, q2),
            CPHASE(pi/2.0, q0, q1),
            H(q0),
            SWAP(q0, q2) )
    return p
```

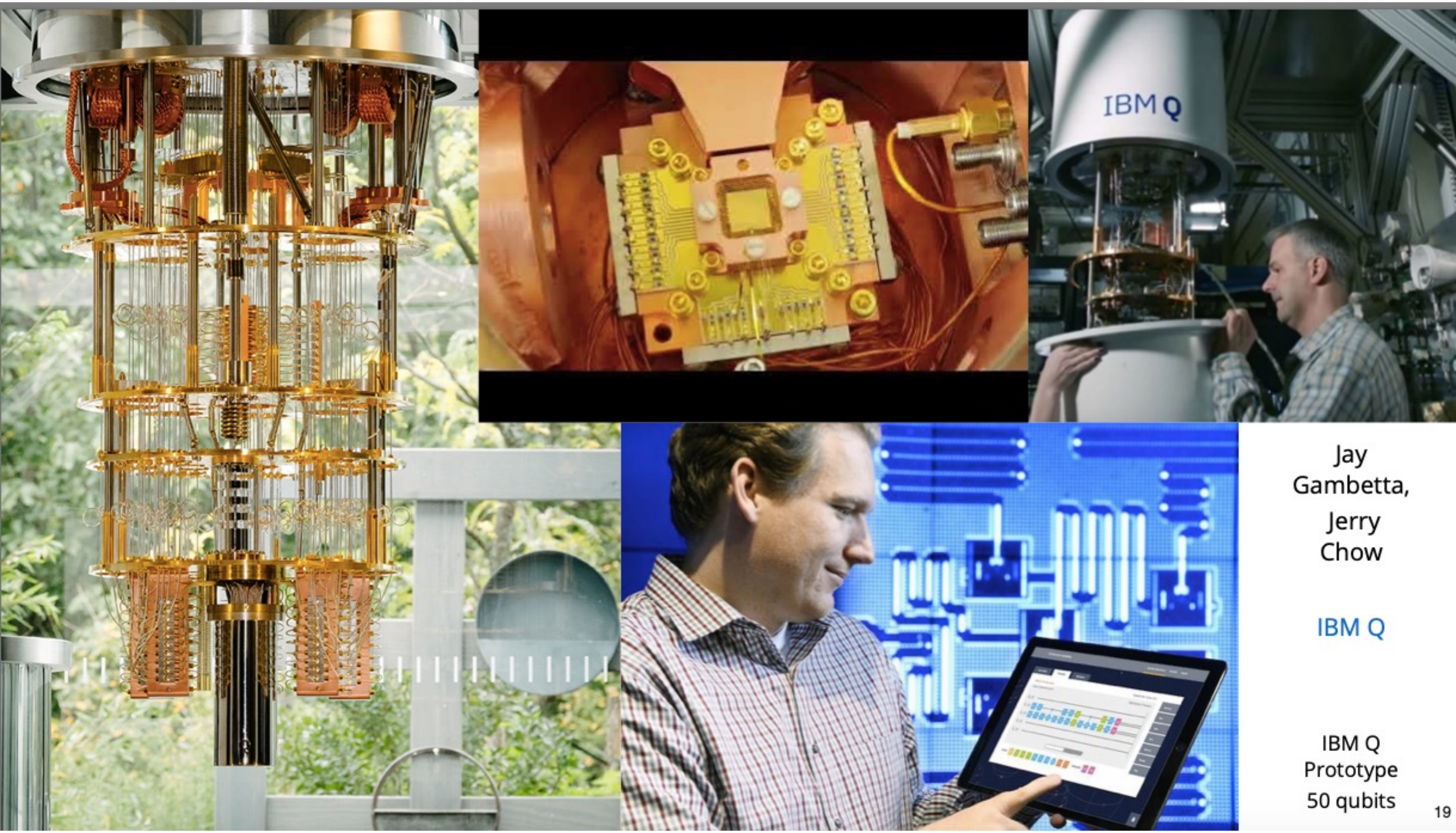
Opensource : ProjectQ





```
from projectq import MainEngine
from projectq.backends import CircuitDraw

from teleport import create_bell_pair

# create a main compiler engine
drawing_engine = CircuitDrawer()
eng = MainEngine(drawing_engine)

create_bell_pair(eng)

eng.flush()
print(drawing_engine.get_latex())
```

QC Presentation

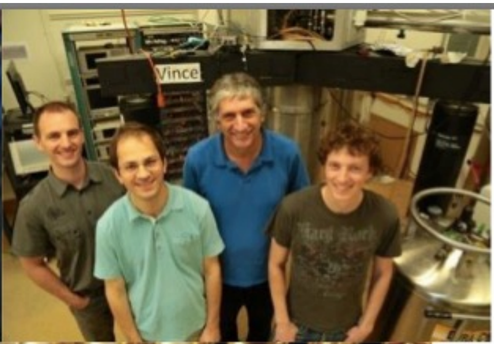


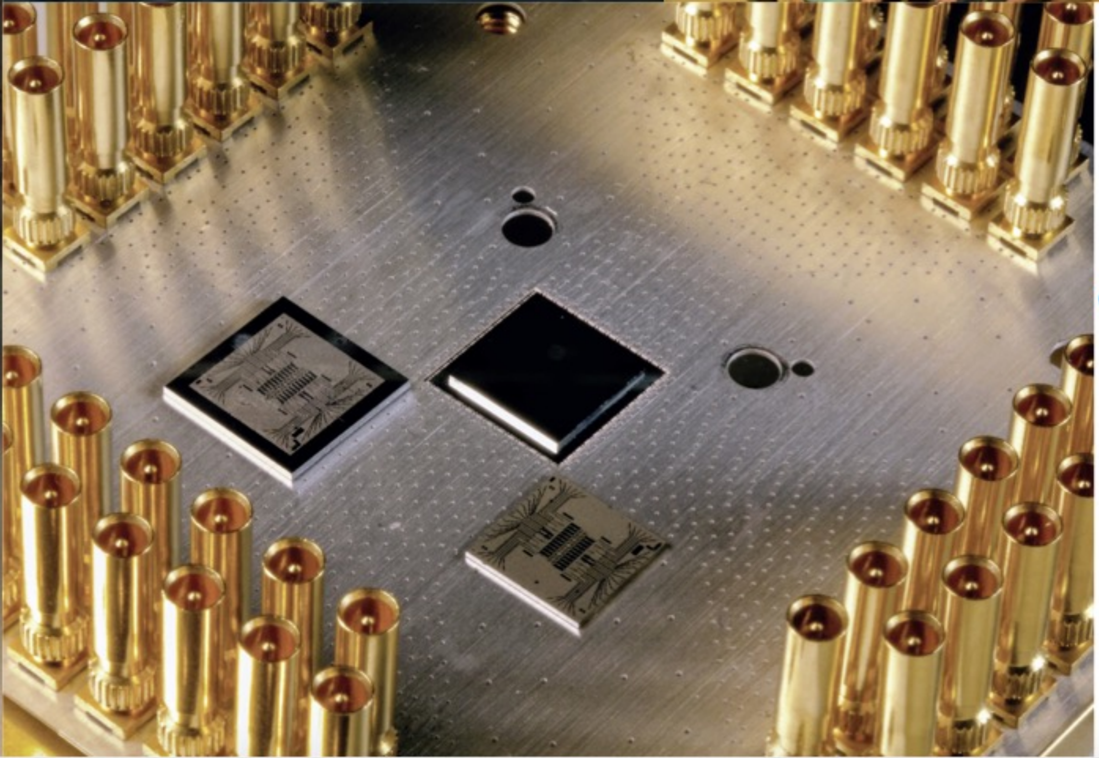
Jay Gambetta, Jerry Chow

IBM Q

IBM Q Prototype 50 qubits

QC Presentation





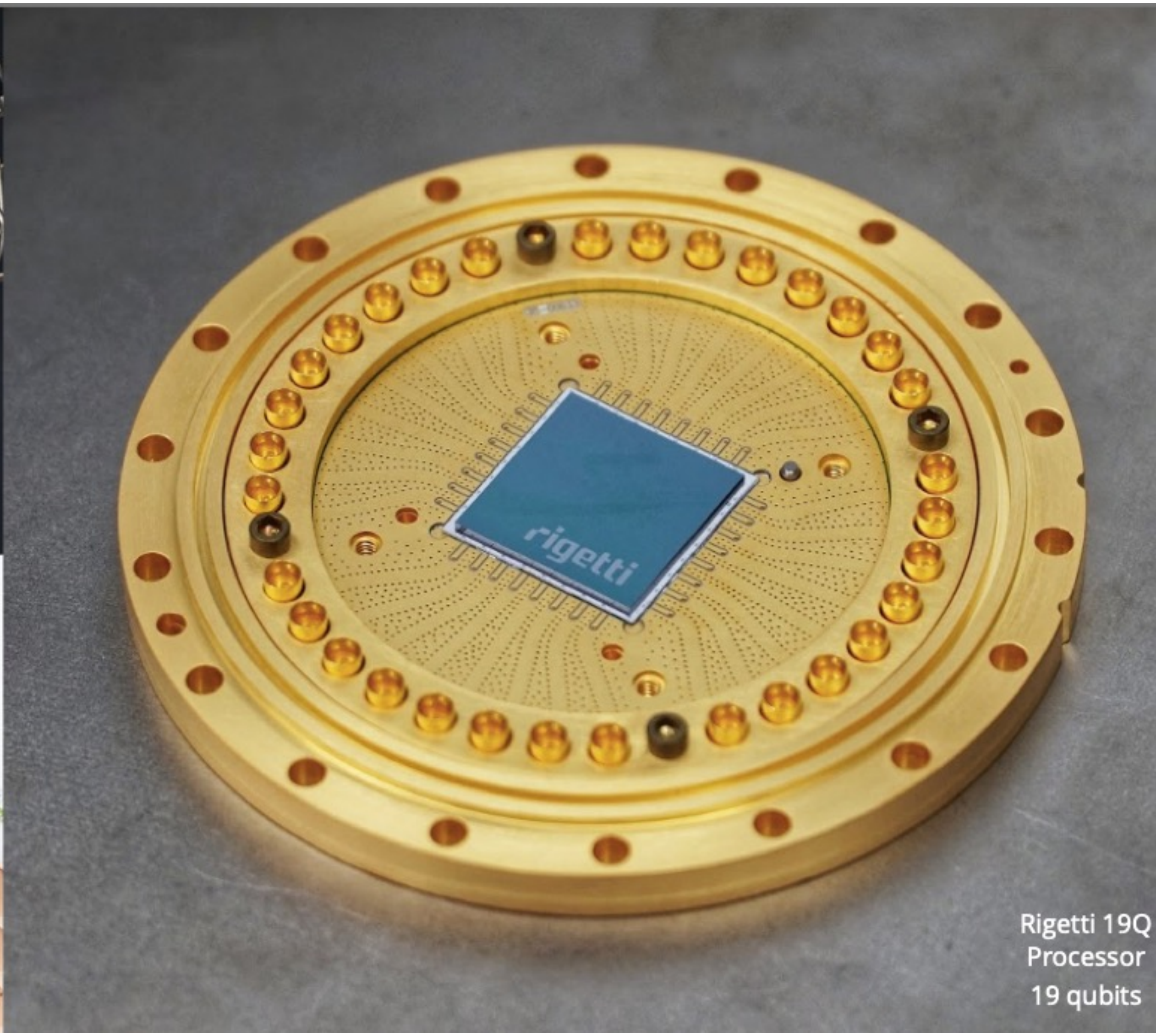
Hartmut Neven, John Martinis,

Google Quantum AI Lab

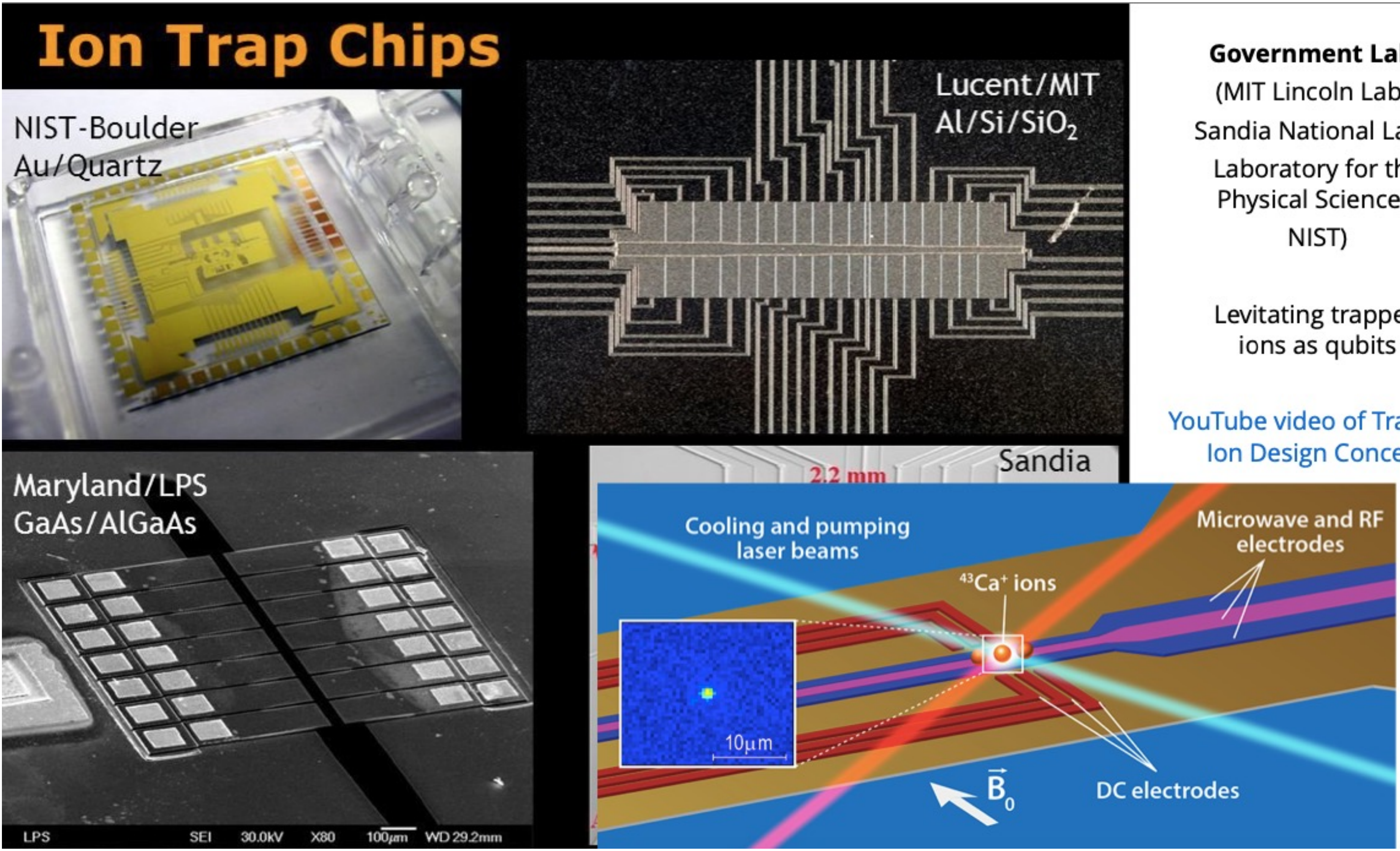
Bristlecone Chip
72 qubits

QC Presentation

Rigetti Computing



Rigetti 19Q Processor
19 qubits



Government Labs

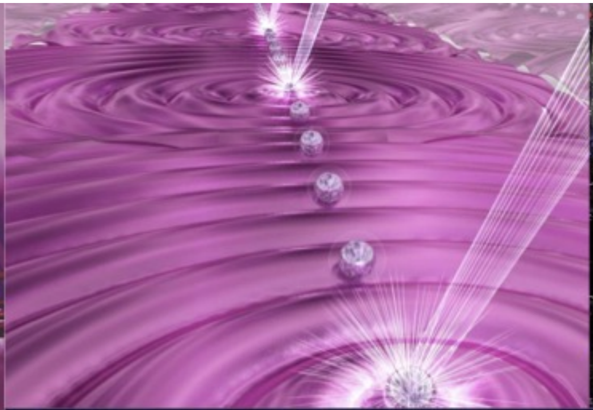
(MIT Lincoln Labs,
Sandia National Labs,
Laboratory for the
Physical Sciences,
NIST)

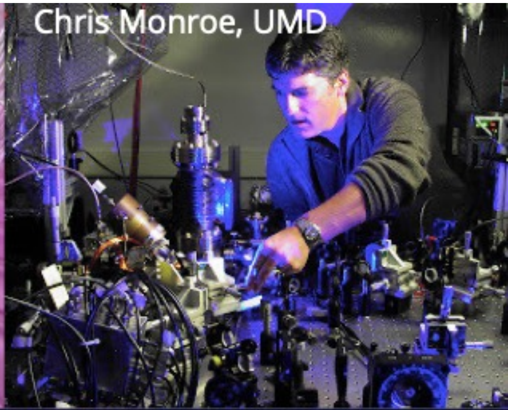
Levitating trapped ions as qubits

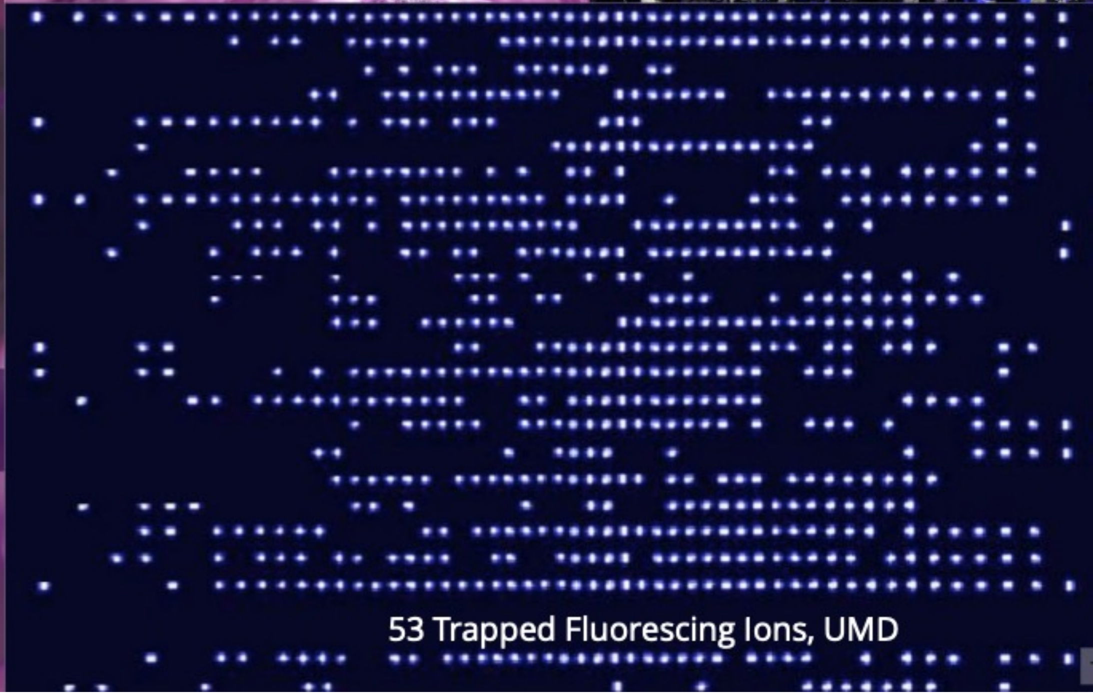
YouTube video of Trapped Ion Design Concept

QC Presentation

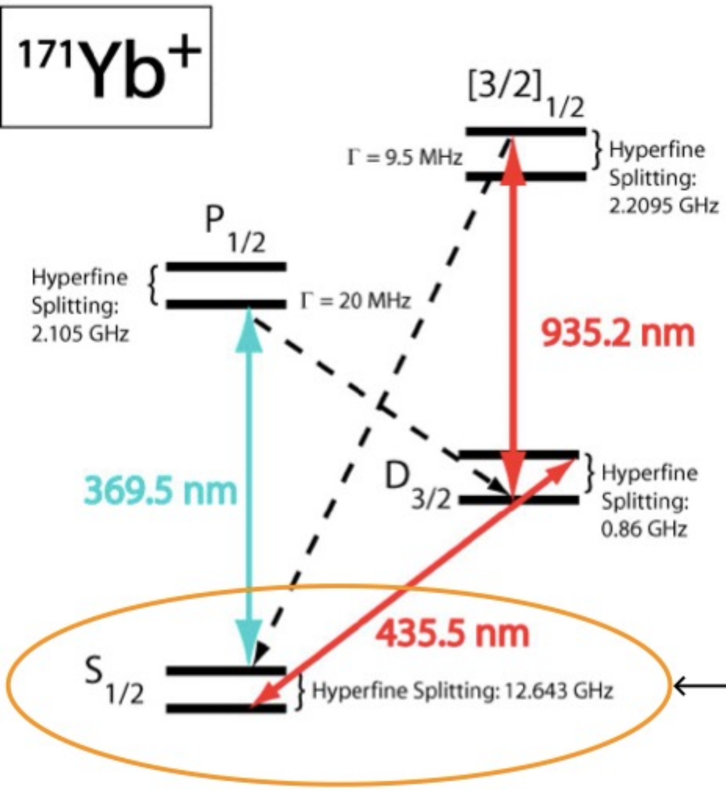
nature
THE INTERNATIONAL WEEKLY JOURNAL OF SCIENCE
MIGHTY ATOMS
A programmable quantum computer based on five atomic qubits PAGES 35 & 63
HUMAN PERFORMANCE
WE HAVE THE TECHNOLOGY
Bionic athletes prepare for 'cyborg Olympics' in Zurich
PAGE 20
ELECTRONIC WASTE
OFFSHORED POLLUTION
A global solution to the flood of toxic e-waste
PAGE 23
ECONOMICS
MINTING TROUBLE
Joseph Stiglitz on that conflicted currency, the euro
PAGE 26
NATUREASIA.COM
4 August 2016
Vol. 536, No. 7614

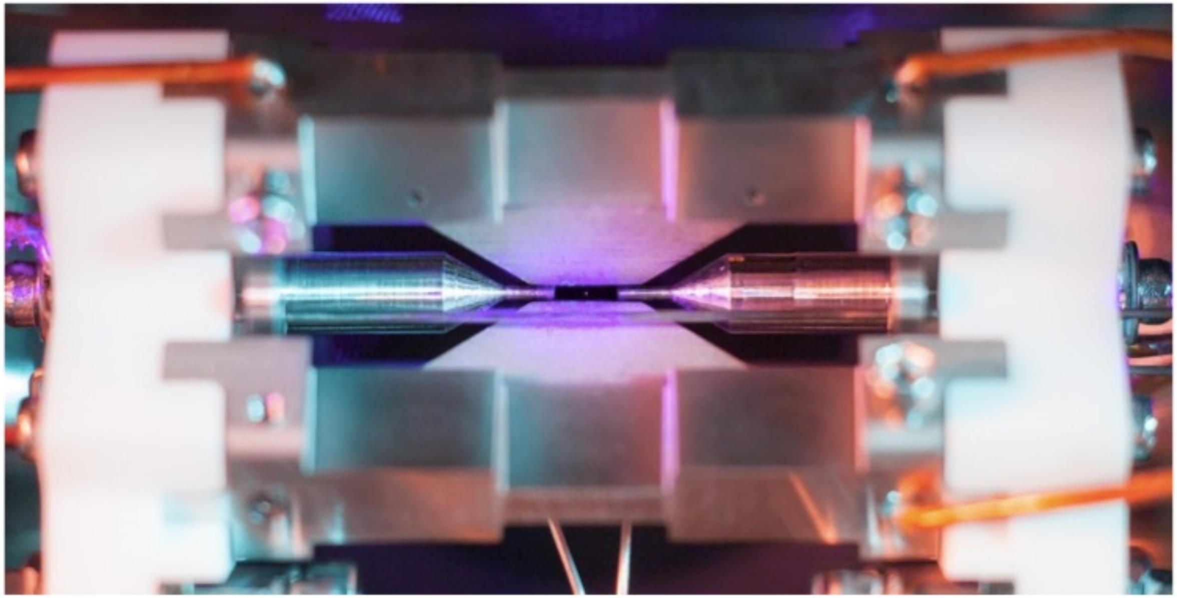



Chris Monroe, UMD


53 Trapped Fluorescing Ions, UMD

Technology 1 : Trapped Ions


$^{171}Yb^+$
$[3/2]_{1/2}$
Γ = 9.5 MHz
Hyperfine Splitting: 2.2095 GHz
$P_{1/2}$
Hyperfine Splitting: 2.105 GHz
Γ = 20 MHz
935.2 nm
369.5 nm
$D_{3/2}$
Hyperfine Splitting: 0.86 GHz
435.5 nm
$S_{1/2}$
Hyperfine Splitting: 12.643 GHz



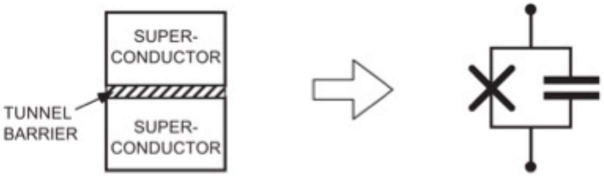
A trapped ion qubit is a superposition of the lowest two magnetic hyperfine energy levels of an ion (like Ytterbium or Calcium)

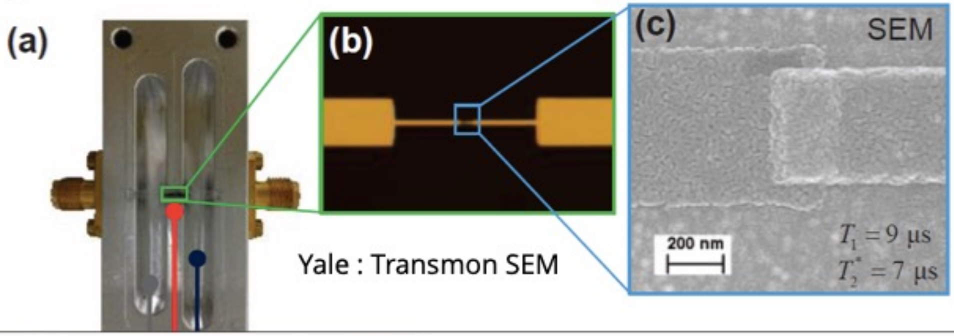
Such ions are trapped and cooled with lasers, then manipulated with more lasers

CHAPMAN UNIVERSITY | INSTITUTE FOR QUANTUM STUDIES

Technology 2 : Superconducting Qubits

A superconducting (transmon) qubit is a superposition of the lowest two energy levels of a charge oscillation (an "artificial atom") across a nonlinear inductive tunnel barrier attached to a capacitive antenna

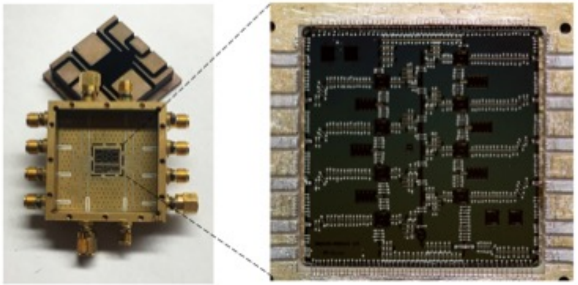




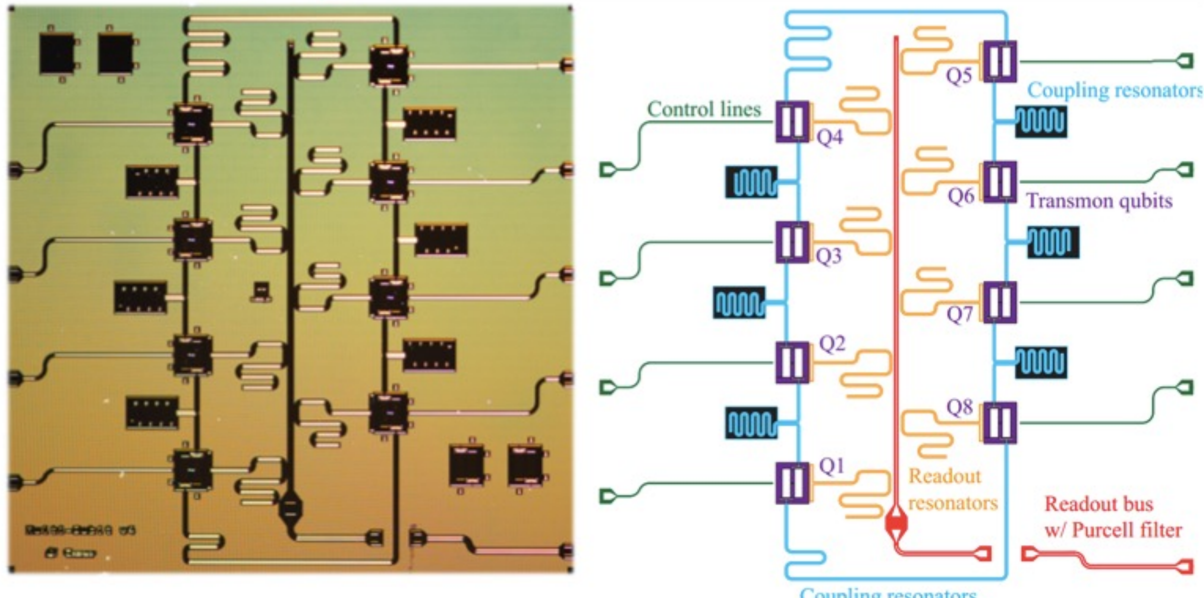
Yale : Transmon SEM

Controlled with all electrical AC signals at microwave frequencies

Cooled to mK temperatures



UC Berkeley : 8 qubit chip

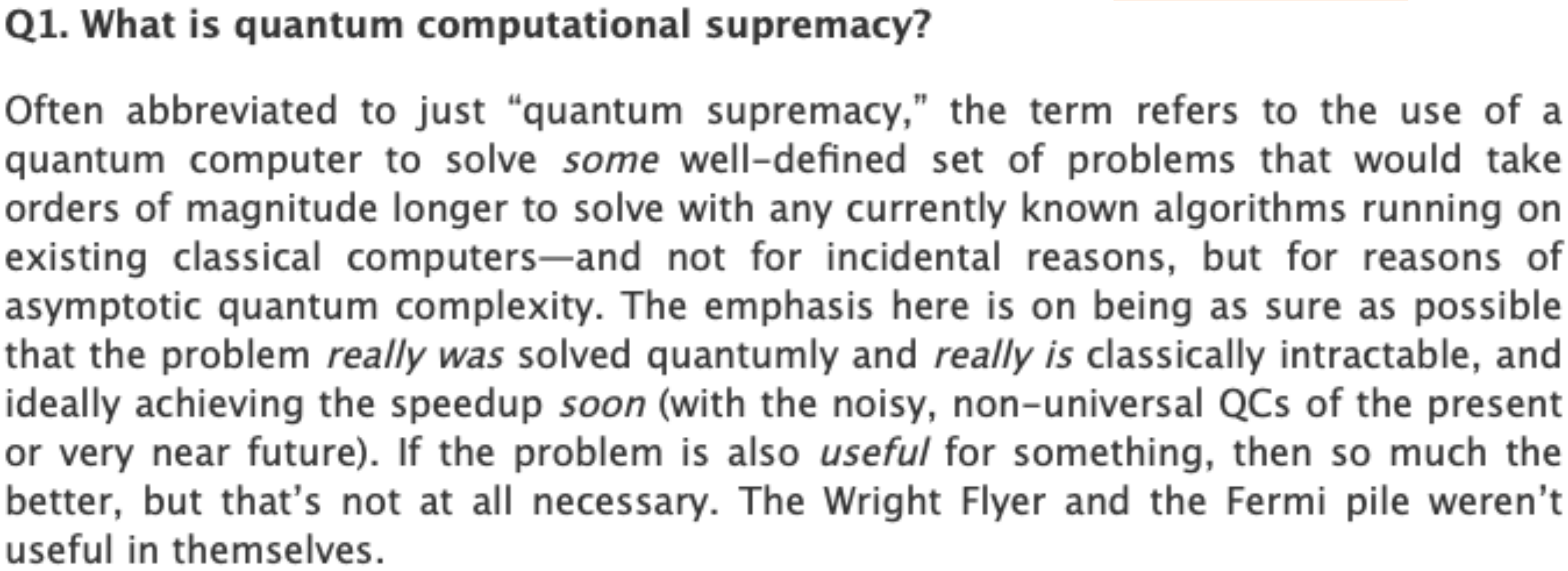


Hardware

Quantum Supremacy

Quantum Supremacy

Scott Aaronson

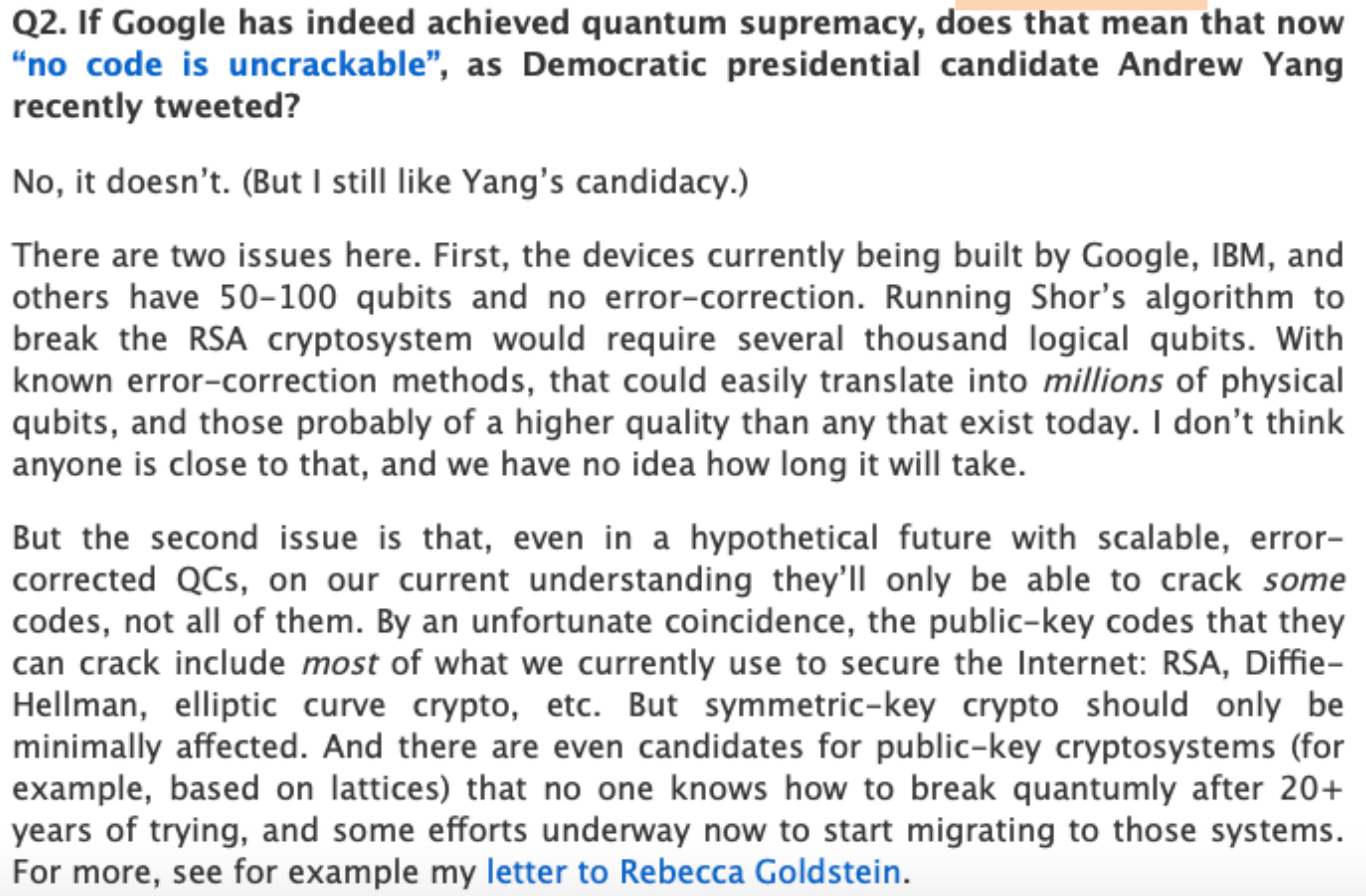


Q1. What is quantum computational supremacy?

Often abbreviated to just "quantum supremacy," the term refers to the use of a quantum computer to solve *some* well-defined set of problems that would take orders of magnitude longer to solve with any currently known algorithms running on existing classical computers—and not for incidental reasons, but for reasons of asymptotic quantum complexity. The emphasis here is on being as sure as possible that the problem *really was* solved quantumly and *really is* classically intractable, and ideally achieving the speedup *soon* (with the noisy, non-universal QCs of the present or very near future). If the problem is also *useful* for something, then so much the better, but that's not at all necessary. The Wright Flyer and the Fermi pile weren't useful in themselves.

Quantum Supremacy

Scott Aaronson



Q2. If Google has indeed achieved quantum supremacy, does that mean that now “no code is uncrackable”, as Democratic presidential candidate Andrew Yang recently tweeted?

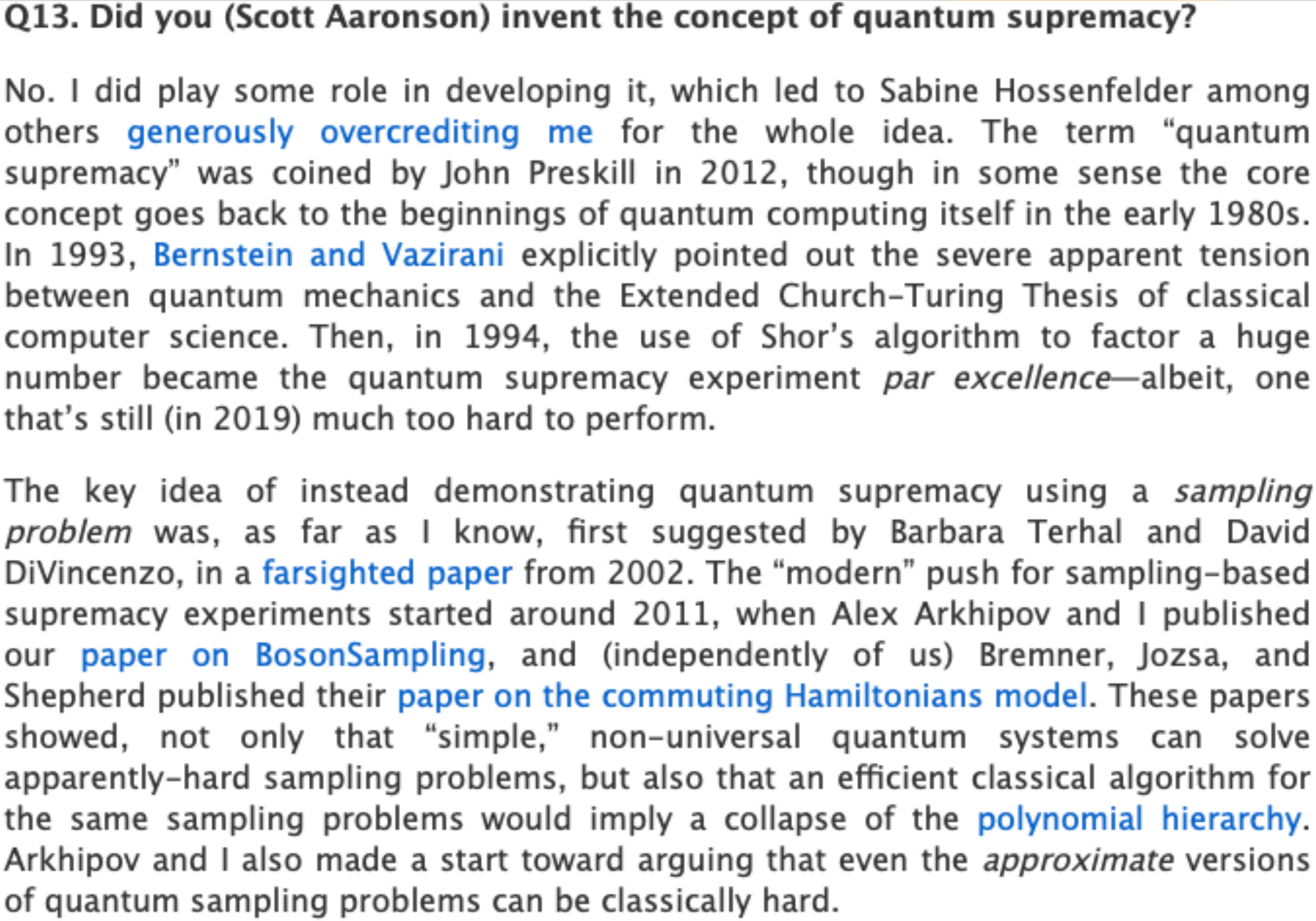
No, it doesn’t. (But I still like Yang’s candidacy.)

There are two issues here. First, the devices currently being built by Google, IBM, and others have 50-100 qubits and no error-correction. Running Shor’s algorithm to break the RSA cryptosystem would require several thousand logical qubits. With known error-correction methods, that could easily translate into *millions* of physical qubits, and those probably of a higher quality than any that exist today. I don’t think anyone is close to that, and we have no idea how long it will take.

But the second issue is that, even in a hypothetical future with scalable, error-corrected QCs, on our current understanding they’ll only be able to crack *some* codes, not all of them. By an unfortunate coincidence, the public-key codes that they can crack include *most* of what we currently use to secure the Internet: RSA, Diffie-Hellman, elliptic curve crypto, etc. But symmetric-key crypto should only be minimally affected. And there are even candidates for public-key cryptosystems (for example, based on lattices) that no one knows how to break quantumly after 20+ years of trying, and some efforts underway now to start migrating to those systems. For more, see for example my letter to Rebecca Goldstein.

Quantum Supremacy

Scott Aaronson



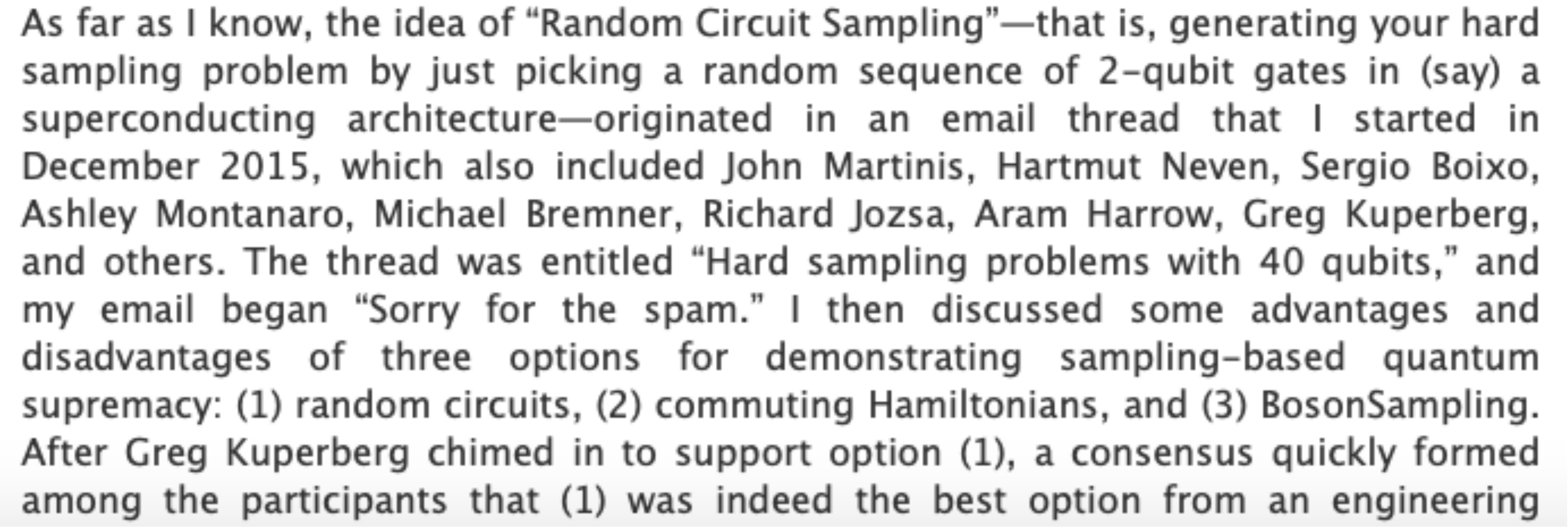
Q13. Did you (Scott Aaronson) invent the concept of quantum supremacy?

No. I did play some role in developing it, which led to Sabine Hossenfelder among others generously overcrediting me for the whole idea. The term "quantum supremacy" was coined by John Preskill in 2012, though in some sense the core concept goes back to the beginnings of quantum computing itself in the early 1980s. In 1993, Bernstein and Vazirani explicitly pointed out the severe apparent tension between quantum mechanics and the Extended Church-Turing Thesis of classical computer science. Then, in 1994, the use of Shor's algorithm to factor a huge number became the quantum supremacy experiment *par excellence*—albeit, one that's still (in 2019) much too hard to perform.

The key idea of instead demonstrating quantum supremacy using a *sampling problem* was, as far as I know, first suggested by Barbara Terhal and David DiVincenzo, in a farsighted paper from 2002. The "modern" push for sampling-based supremacy experiments started around 2011, when Alex Arkhipov and I published our paper on BosonSampling, and (independently of us) Bremner, Jozsa, and Shepherd published their paper on the commuting Hamiltonians model. These papers showed, not only that "simple," non-universal quantum systems can solve apparently-hard sampling problems, but also that an efficient classical algorithm for the same sampling problems would imply a collapse of the polynomial hierarchy. Arkhipov and I also made a start toward arguing that even the *approximate* versions of quantum sampling problems can be classically hard.

Quantum Supr.: Random Sampling

Scott Aaronson



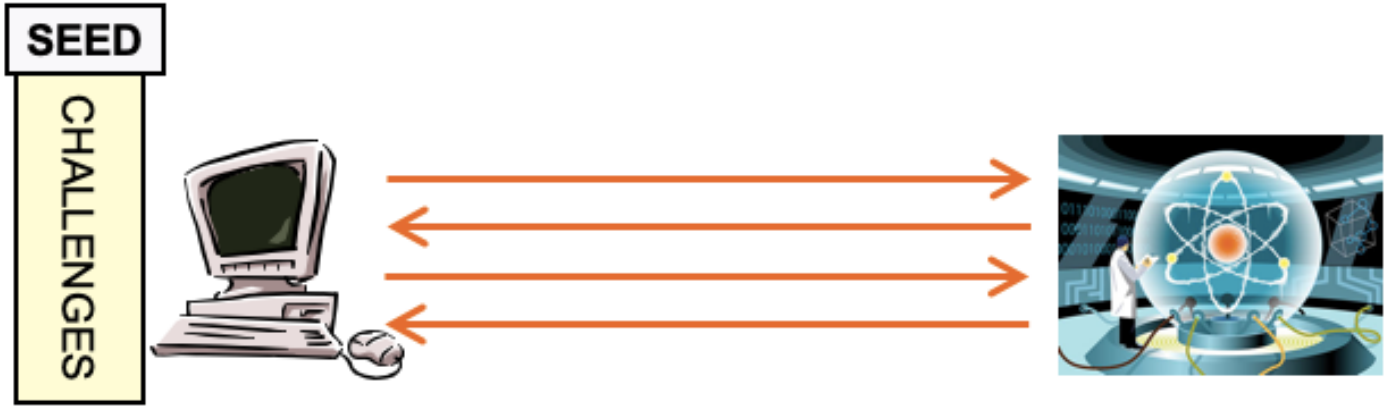
As far as I know, the idea of "Random Circuit Sampling"—that is, generating your hard sampling problem by just picking a random sequence of 2-qubit gates in (say) a superconducting architecture—originated in an email thread that I started in December 2015, which also included John Martinis, Hartmut Neven, Sergio Boixo, Ashley Montanaro, Michael Bremner, Richard Jozsa, Aram Harrow, Greg Kuperberg, and others. The thread was entitled "Hard sampling problems with 40 qubits," and my email began "Sorry for the spam." I then discussed some advantages and disadvantages of three options for demonstrating sampling-based quantum supremacy: (1) random circuits, (2) commuting Hamiltonians, and (3) BosonSampling. After Greg Kuperberg chimed in to support option (1), a consensus quickly formed among the participants that (1) was indeed the best option from an engineering

Quantum Supr.: Random Sampling

Scott Aaronson

The Randomness Protocol

"Born from complexity theory. Somehow became first planned application for Bristlecone / Sycamore..."

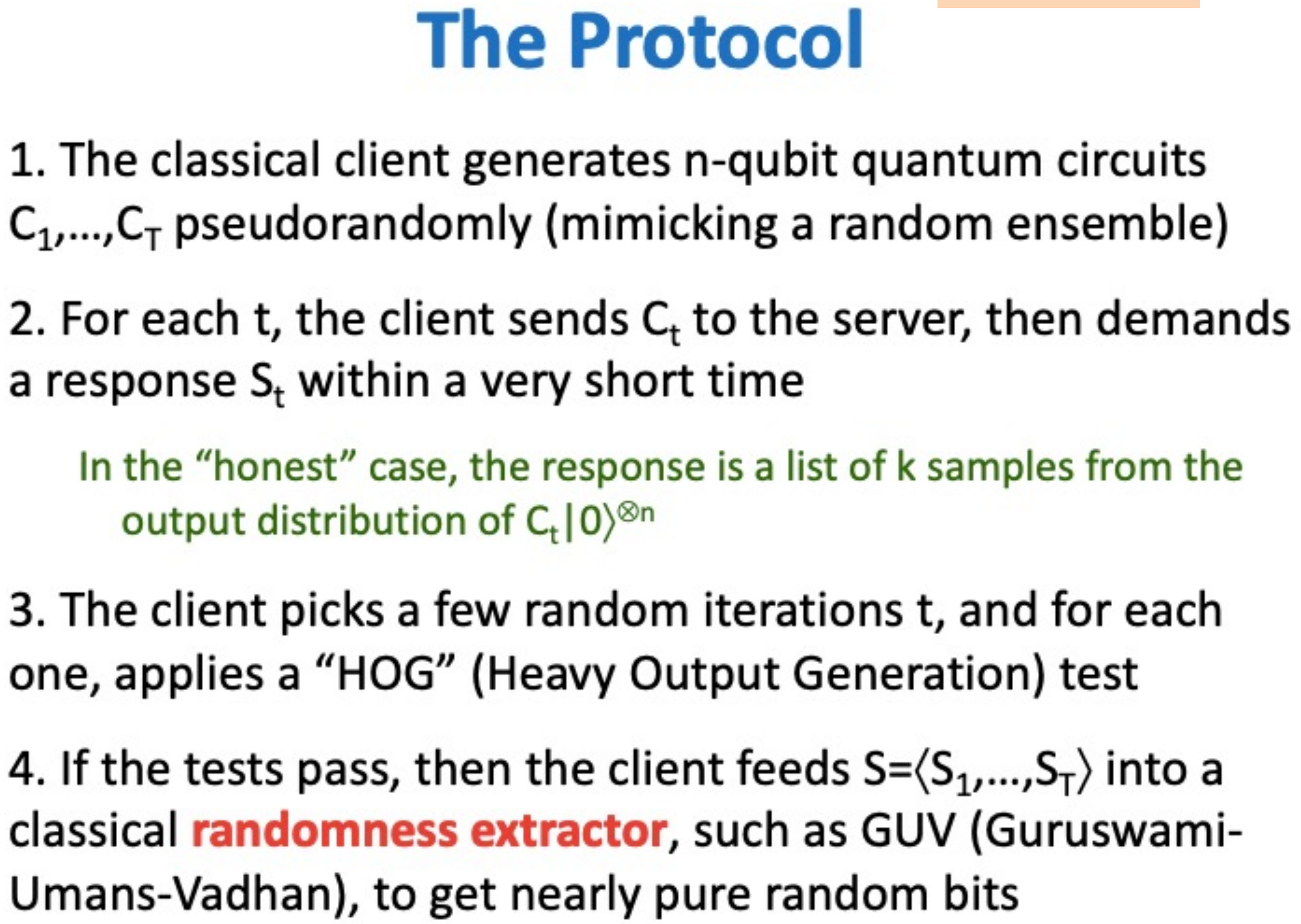


Goal: By interacting with a NISQ QC remotely, force it to generate fresh random bits, which no one (not even the QC) knew beforehand. **Place no trust in the QC!**

"Proof of Sampling." Modest quantum speedups, not for their own sake, but as proof of some other property

Quantum Supr.: Random Sampling

Scott Aaronson



The Protocol

1. The classical client generates n-qubit quantum circuits $C_1,\ldots,C_T$ pseudorandomly (mimicking a random ensemble)

2. For each t, the client sends C_t to the server, then demands a response S_t within a very short time

   In the "honest" case, the response is a list of k samples from the output distribution of $C_t|0\rangle^{\otimes n}$

3. The client picks a few random iterations t, and for each one, applies a "HOG" (Heavy Output Generation) test

4. If the tests pass, then the client feeds $S=\langle S_1,\ldots,S_T\rangle$ into a classical **randomness extractor**, such as GUV (Guruswami-Umans-Vadhan), to get nearly pure random bits

Hardware

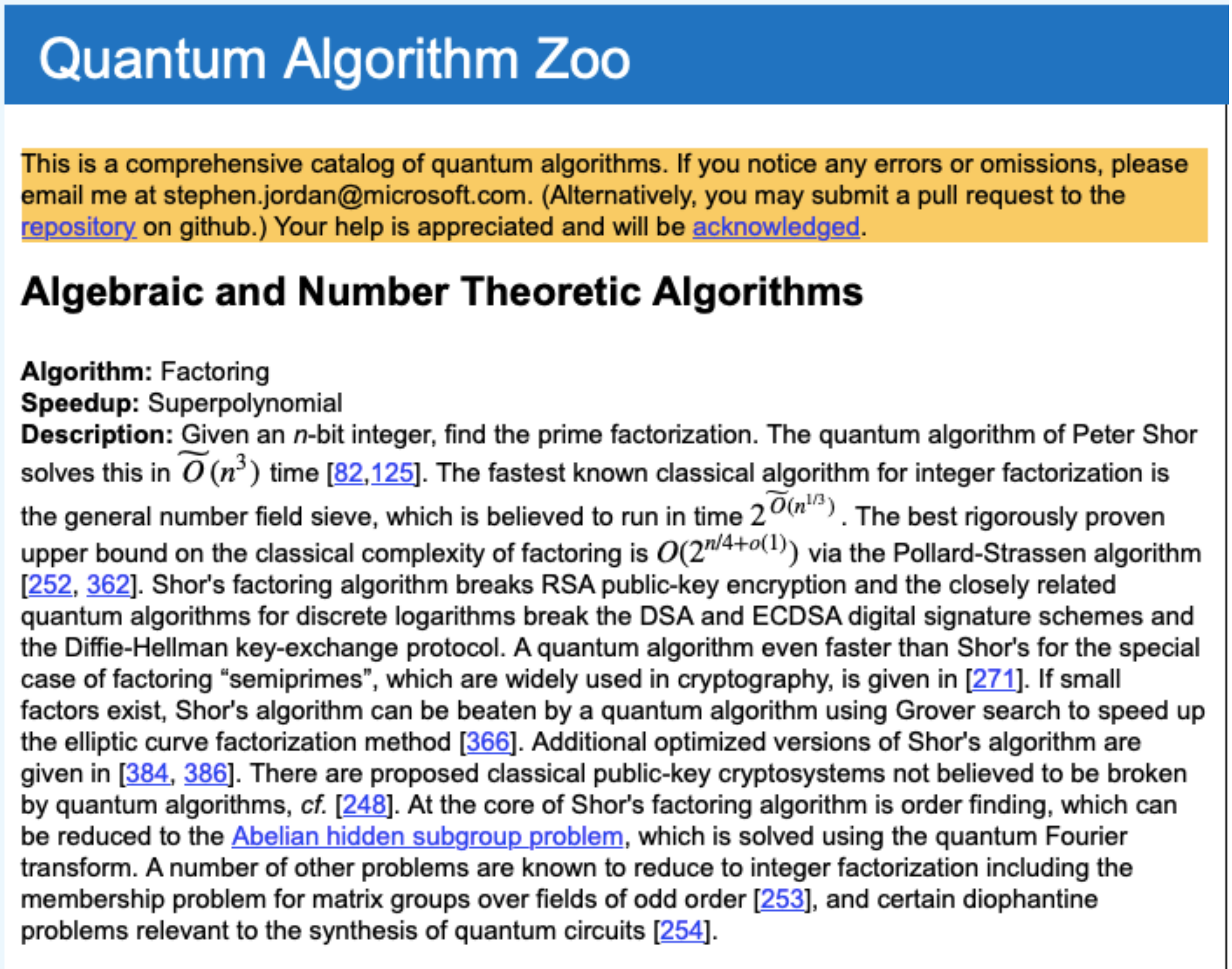
Other
Algorithms

Lab Experiments in QC

By a PhD researcher: **Patrick Banner** Physics PhD student

In my experiment, rubidium atoms are loaded into a magneto-optical trap (MOT), cooled using optical molasses, and then trapped finally in an optical dipole trap (ODT); we then run our experiment, which usually means sending a probe laser and a control laser through our cloud of about 10,000 atoms, and measuring in one way or another the probe light that exits the cloud. All of this happens in a fraction of a second, with the interesting part happening in tens of milliseconds or less. The time period of an experiment happening is audibly defined by laser shutters in our lab clicking on and off within a second. An entire experimental cycle is called a “shot,” and gives effectively one data point for every parameter.

QC Algorithms



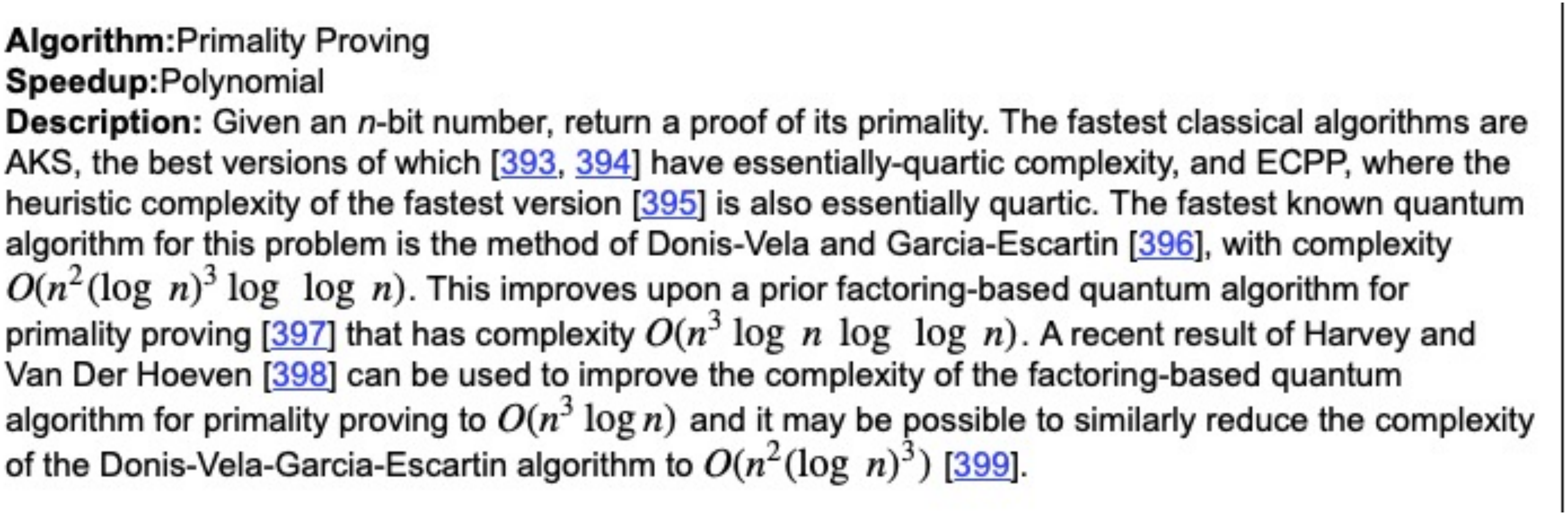
Quantum Algorithm Zoo

This is a comprehensive catalog of quantum algorithms. If you notice any errors or omissions, please email me at stephen.jordan@microsoft.com. (Alternatively, you may submit a pull request to the repository on github.) Your help is appreciated and will be acknowledged.

Algebraic and Number Theoretic Algorithms

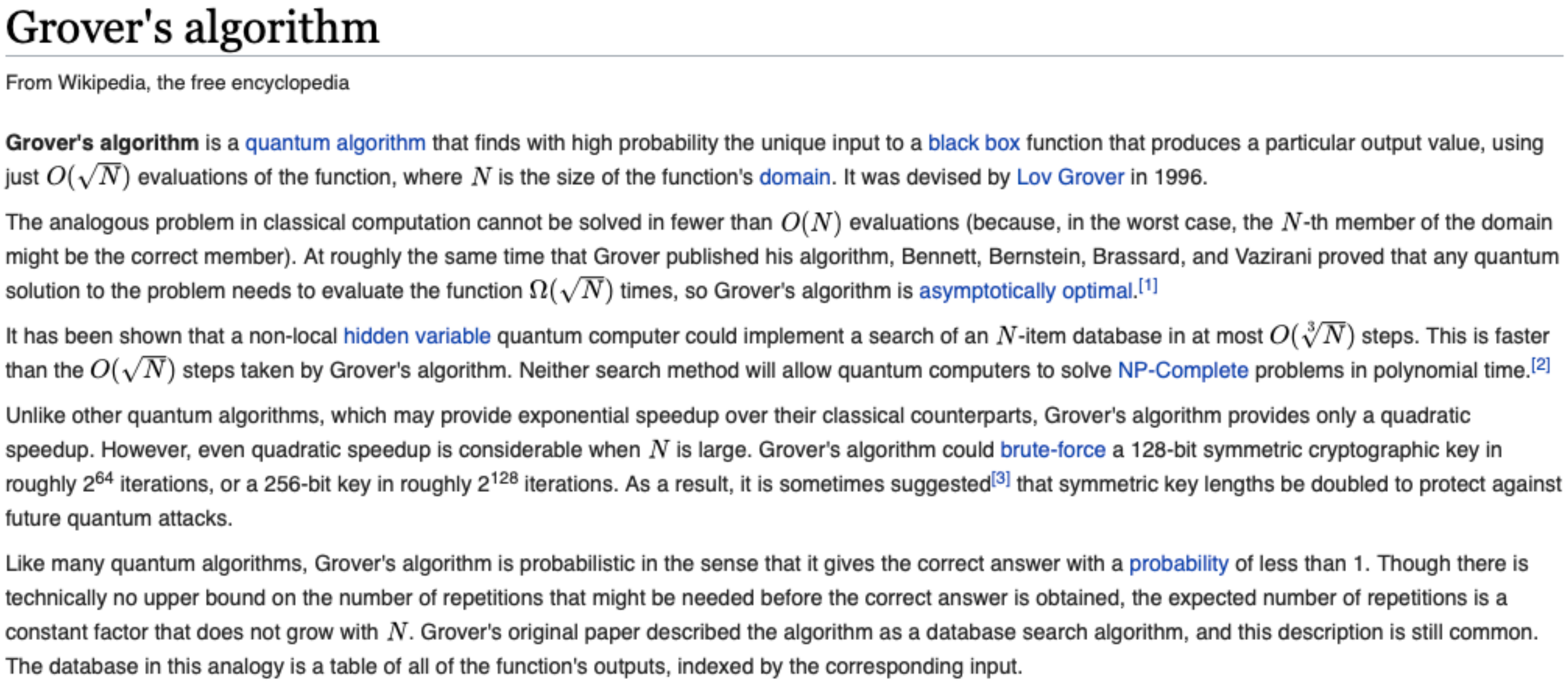
Algorithm: Factoring
Speedup: Superpolynomial
Description: Given an *n*-bit integer, find the prime factorization. The quantum algorithm of Peter Shor solves this in $\widetilde{O}(n^3)$ time [82,125]. The fastest known classical algorithm for integer factorization is the general number field sieve, which is believed to run in time $2^{\widetilde{O}(n^{1/3})}$. The best rigorously proven upper bound on the classical complexity of factoring is $O(2^{n/4+o(1)})$ via the Pollard-Strassen algorithm [252, 362]. Shor's factoring algorithm breaks RSA public-key encryption and the closely related quantum algorithms for discrete logarithms break the DSA and ECDSA digital signature schemes and the Diffie-Hellman key-exchange protocol. A quantum algorithm even faster than Shor's for the special case of factoring "semiprimes", which are widely used in cryptography, is given in [271]. If small factors exist, Shor's algorithm can be beaten by a quantum algorithm using Grover search to speed up the elliptic curve factorization method [366]. Additional optimized versions of Shor's algorithm are given in [384, 386]. There are proposed classical public-key cryptosystems not believed to be broken by quantum algorithms, *cf.* [248]. At the core of Shor's factoring algorithm is order finding, which can be reduced to the Abelian hidden subgroup problem, which is solved using the quantum Fourier transform. A number of other problems are known to reduce to integer factorization including the membership problem for matrix groups over fields of odd order [253], and certain diophantine problems relevant to the synthesis of quantum circuits [254].

QC Algorithms-Primality



Algorithm: Primality Proving
Speedup: Polynomial
Description: Given an *n*-bit number, return a proof of its primality. The fastest classical algorithms are AKS, the best versions of which [393, 394] have essentially-quartic complexity, and ECPP, where the heuristic complexity of the fastest version [395] is also essentially quartic. The fastest known quantum algorithm for this problem is the method of Donis-Vela and Garcia-Escartin [396], with complexity $O(n^2(\log\ n)^3 \log\ \ \log\ n)$. This improves upon a prior factoring-based quantum algorithm for primality proving [397] that has complexity $O(n^3 \log\ n\ \log\ \ \log\ n)$. A recent result of Harvey and Van Der Hoeven [398] can be used to improve the complexity of the factoring-based quantum algorithm for primality proving to $O(n^3 \log n)$ and it may be possible to similarly reduce the complexity of the Donis-Vela-Garcia-Escartin algorithm to $O(n^2(\log\ n)^3)$ [399].

Grover’s Algorithm



Grover's algorithm

From Wikipedia, the free encyclopedia

Grover's algorithm is a quantum algorithm that finds with high probability the unique input to a black box function that produces a particular output value, using just $O(\sqrt{N})$ evaluations of the function, where N is the size of the function's domain. It was devised by Lov Grover in 1996.

The analogous problem in classical computation cannot be solved in fewer than $O(N)$ evaluations (because, in the worst case, the N-th member of the domain might be the correct member). At roughly the same time that Grover published his algorithm, Bennett, Bernstein, Brassard, and Vazirani proved that any quantum solution to the problem needs to evaluate the function $\Omega(\sqrt{N})$ times, so Grover's algorithm is asymptotically optimal.[1]

It has been shown that a non-local hidden variable quantum computer could implement a search of an N-item database in at most $O(\sqrt[3]{N})$ steps. This is faster than the $O(\sqrt{N})$ steps taken by Grover's algorithm. Neither search method will allow quantum computers to solve NP-Complete problems in polynomial time.[2]

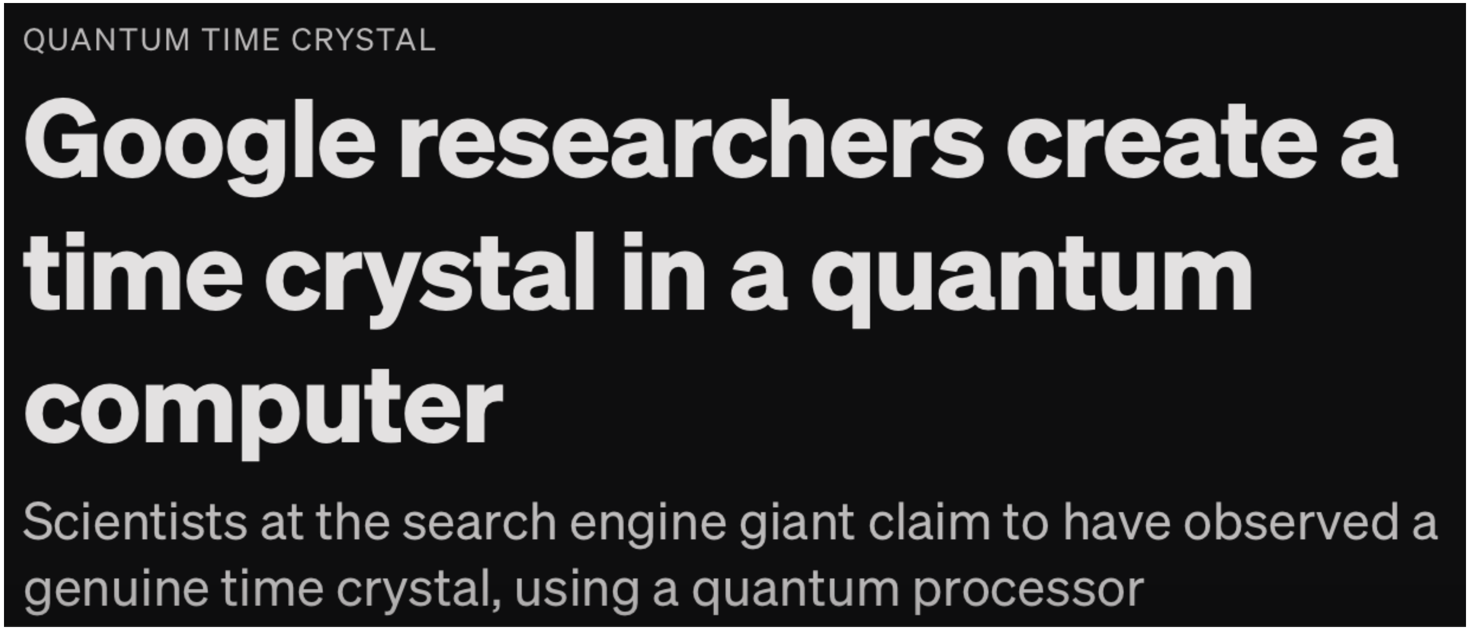
Unlike other quantum algorithms, which may provide exponential speedup over their classical counterparts, Grover's algorithm provides only a quadratic speedup. However, even quadratic speedup is considerable when N is large. Grover's algorithm could brute-force a 128-bit symmetric cryptographic key in roughly 2^{64} iterations, or a 256-bit key in roughly 2^{128} iterations. As a result, it is sometimes suggested[3] that symmetric key lengths be doubled to protect against future quantum attacks.

Like many quantum algorithms, Grover's algorithm is probabilistic in the sense that it gives the correct answer with a probability of less than 1. Though there is technically no upper bound on the number of repetitions that might be needed before the correct answer is obtained, the expected number of repetitions is a constant factor that does not grow with N. Grover's original paper described the algorithm as a database search algorithm, and this description is still common. The database in this analogy is a table of all of the function's outputs, indexed by the corresponding input.

Hardware

Time Crystals

Time Crystals



QUANTUM TIME CRYSTAL

Google researchers create a time crystal in a quantum computer

Scientists at the search engine giant claim to have observed a genuine time crystal, using a quantum processor

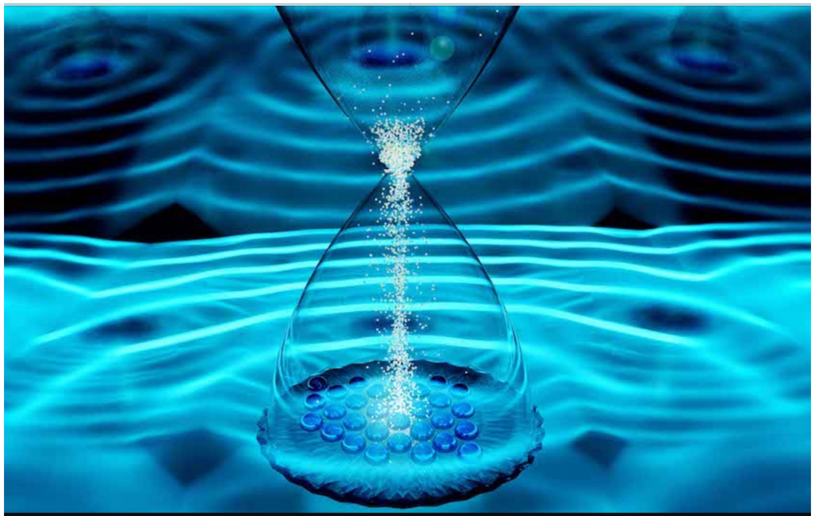


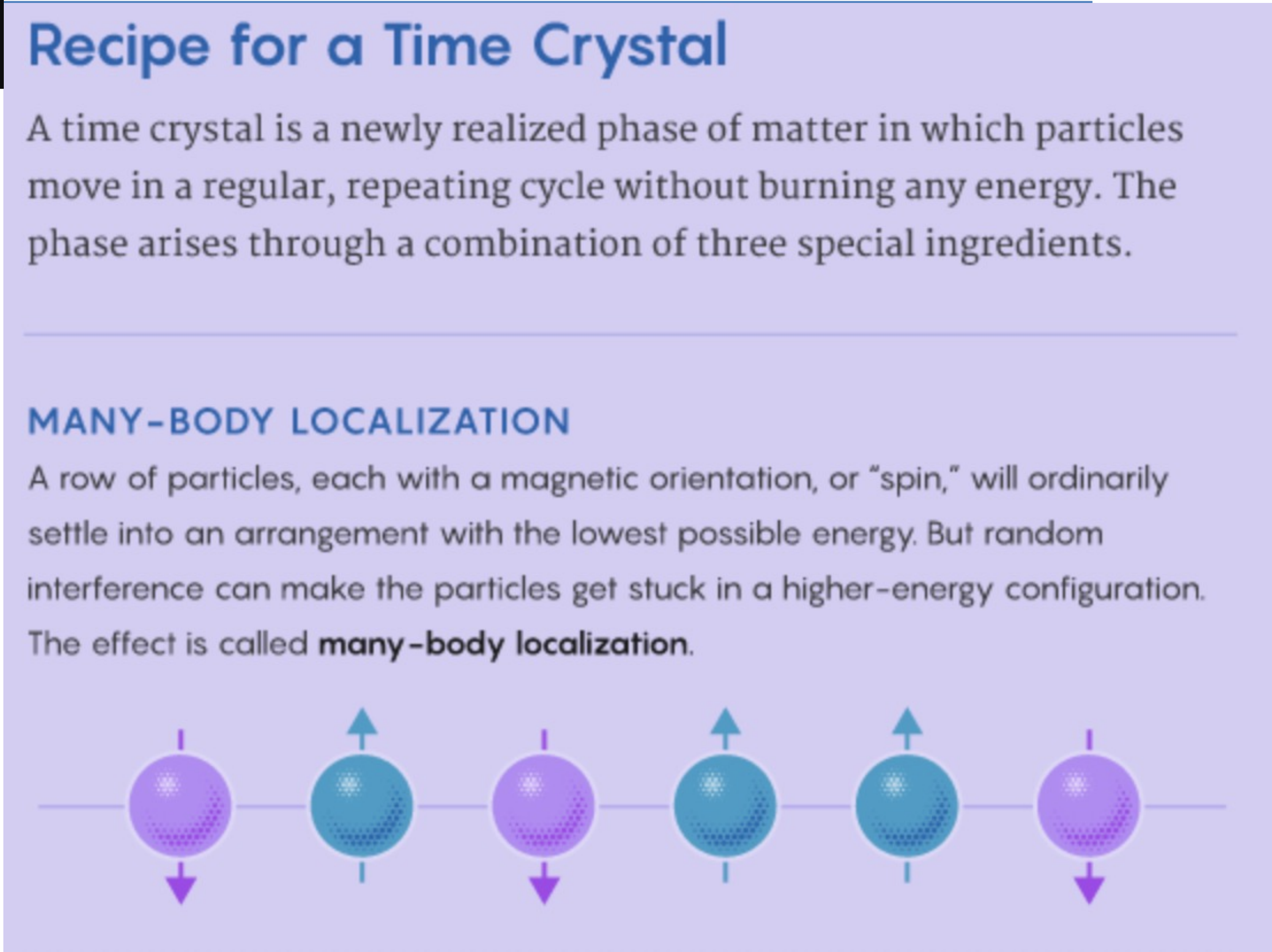
Image Credit: E. Edwards/JQI

Time Crystals

DR JEFF SOFTWARE
INDIE APP DEVELOPER
© Jeff Drobman
2016-23

Faisal Khan

A devout futurist keeping a keen eye on the latest in Emerging Tech, Global Economy, Space, Science, Cryptocurrencies & more



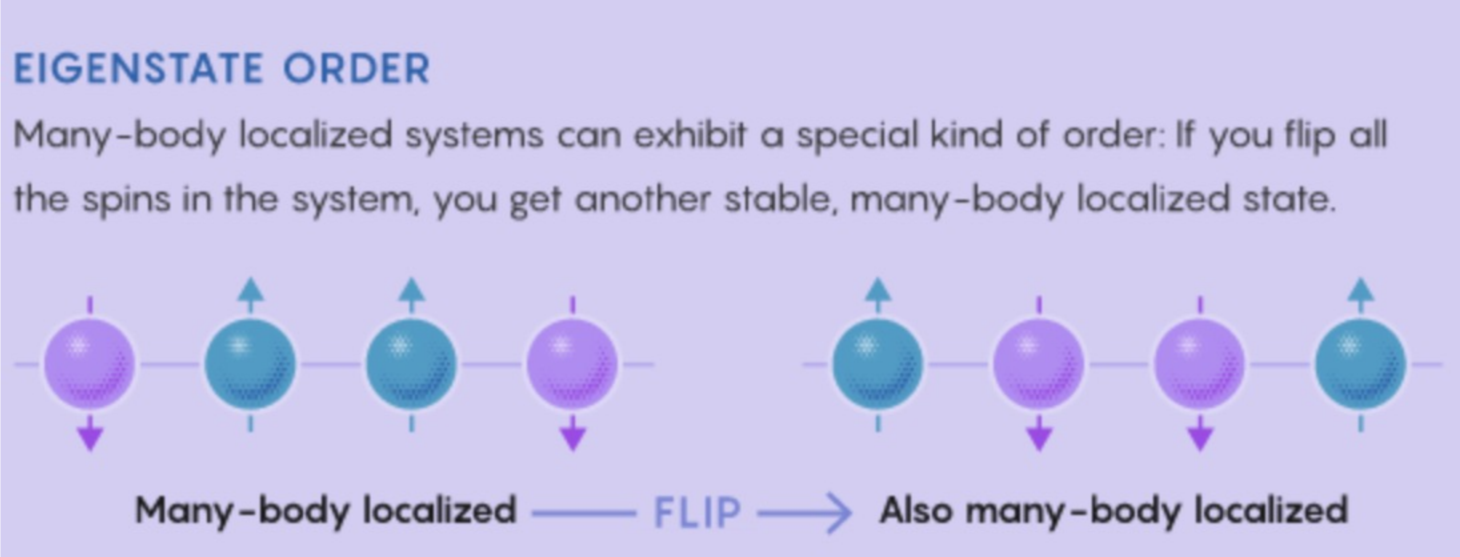
Recipe for a Time Crystal

A time crystal is a newly realized phase of matter in which particles move in a regular, repeating cycle without burning any energy. The phase arises through a combination of three special ingredients.

MANY-BODY LOCALIZATION

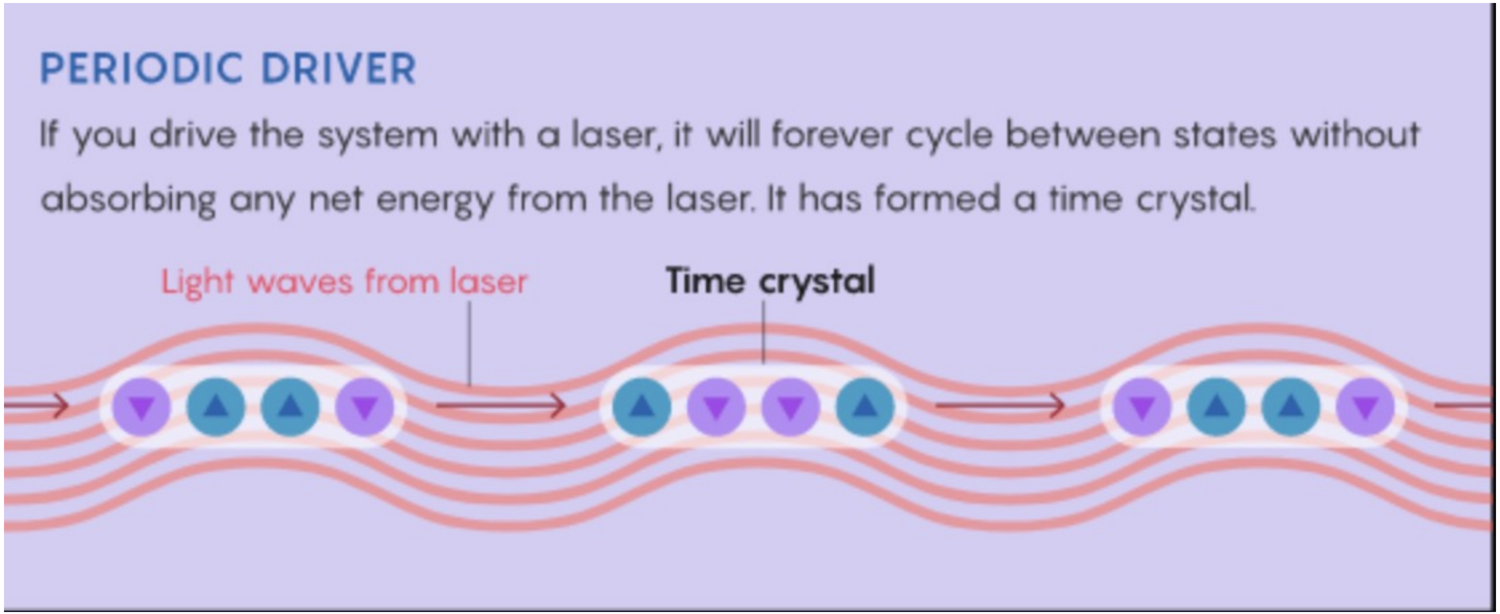
A row of particles, each with a magnetic orientation, or "spin," will ordinarily settle into an arrangement with the lowest possible energy. But random interference can make the particles get stuck in a higher-energy configuration. The effect is called **many-body localization**.

Time Crystals



EIGENSTATE ORDER

Many-body localized systems can exhibit a special kind of order: If you flip all the spins in the system, you get another stable, many-body localized state.



PERIODIC DRIVER

If you drive the system with a laser, it will forever cycle between states without absorbing any net energy from the laser. It has formed a time crystal.

ACM Tech Talk

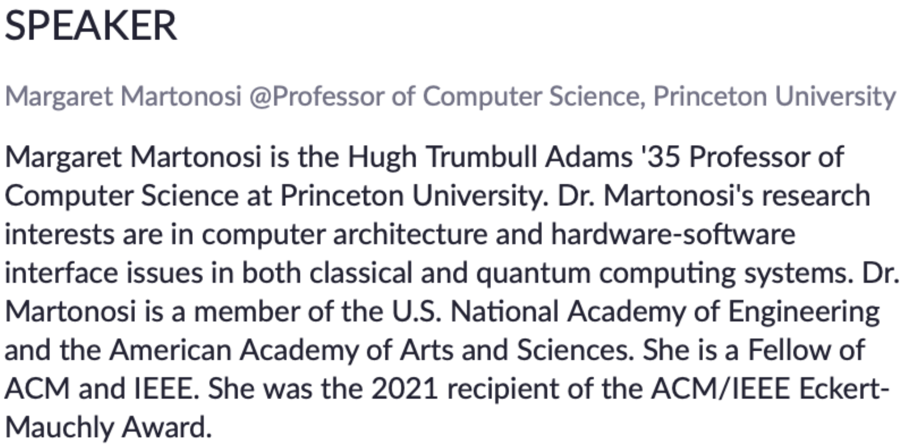
Quantum Computing

Seismic Shifts: Challenges and Opportunities in the 'Post-ISA' Era of Computer Systems Design

Education
ACM Learning Center

Compiled by Dr Jeff Drobman

- ➢ Focus on a hybrid *classical – quantum* distributed architecture



SPEAKER

Margaret Martonosi @Professor of Computer Science, Princeton University

Margaret Martonosi is the Hugh Trumbull Adams '35 Professor of Computer Science at Princeton University. Dr. Martonosi's research interests are in computer architecture and hardware-software interface issues in both classical and quantum computing systems. Dr. Martonosi is a member of the U.S. National Academy of Engineering and the American Academy of Arts and Sciences. She is a Fellow of ACM and IEEE. She was the 2021 recipient of the ACM/IEEE Eckert-Mauchly Award.

The Learning Continues…

TechTalk Discourse Forum: https://on.acm.org

TechTalk Inquiries: learning@acm.org

TechTalk Archives: https://learning.acm.org/techtalks

Learning Center: https://learning.acm.org

ACM Selects: https://selects.acm.org/

ACM ByteCast: https://learning.acm.org/bytecast

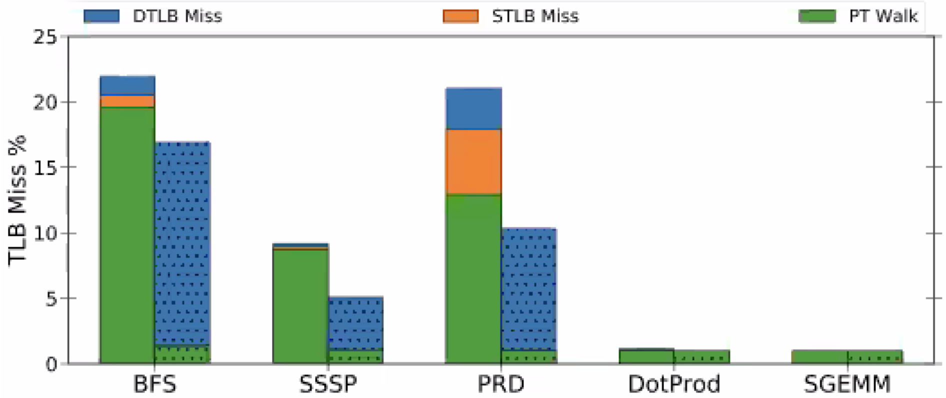
Professional Ethics: https://ethics.acm.org

Queue Magazine: https://queue.acm.org

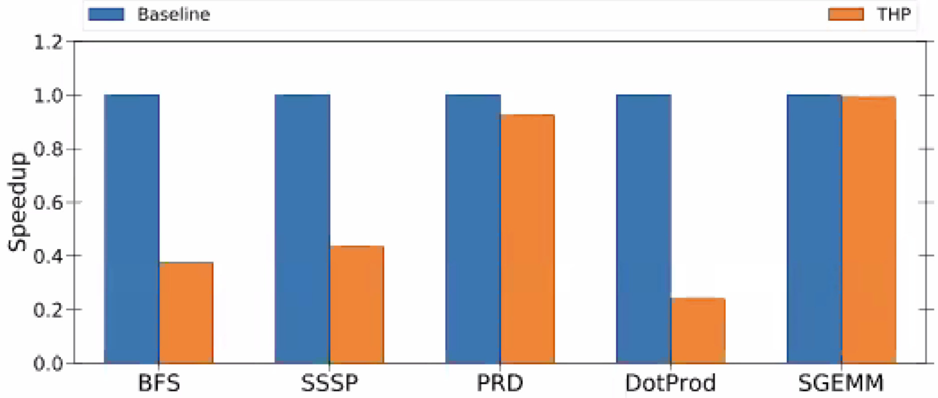
Page Size

Example 1: OS Page Size Management Tailored Graph Analytics

- Graph analytics have high TLB miss rates that cause address translation overheads
- Huge pages (2MB on x86) can alleviate such overheads with increased TLB reach
- Modern OS policies greedily (over)allocate huge pages due to lack of app knowledge
- Need: OS techniques to intelligently manage huge pages tailored for graph analytics

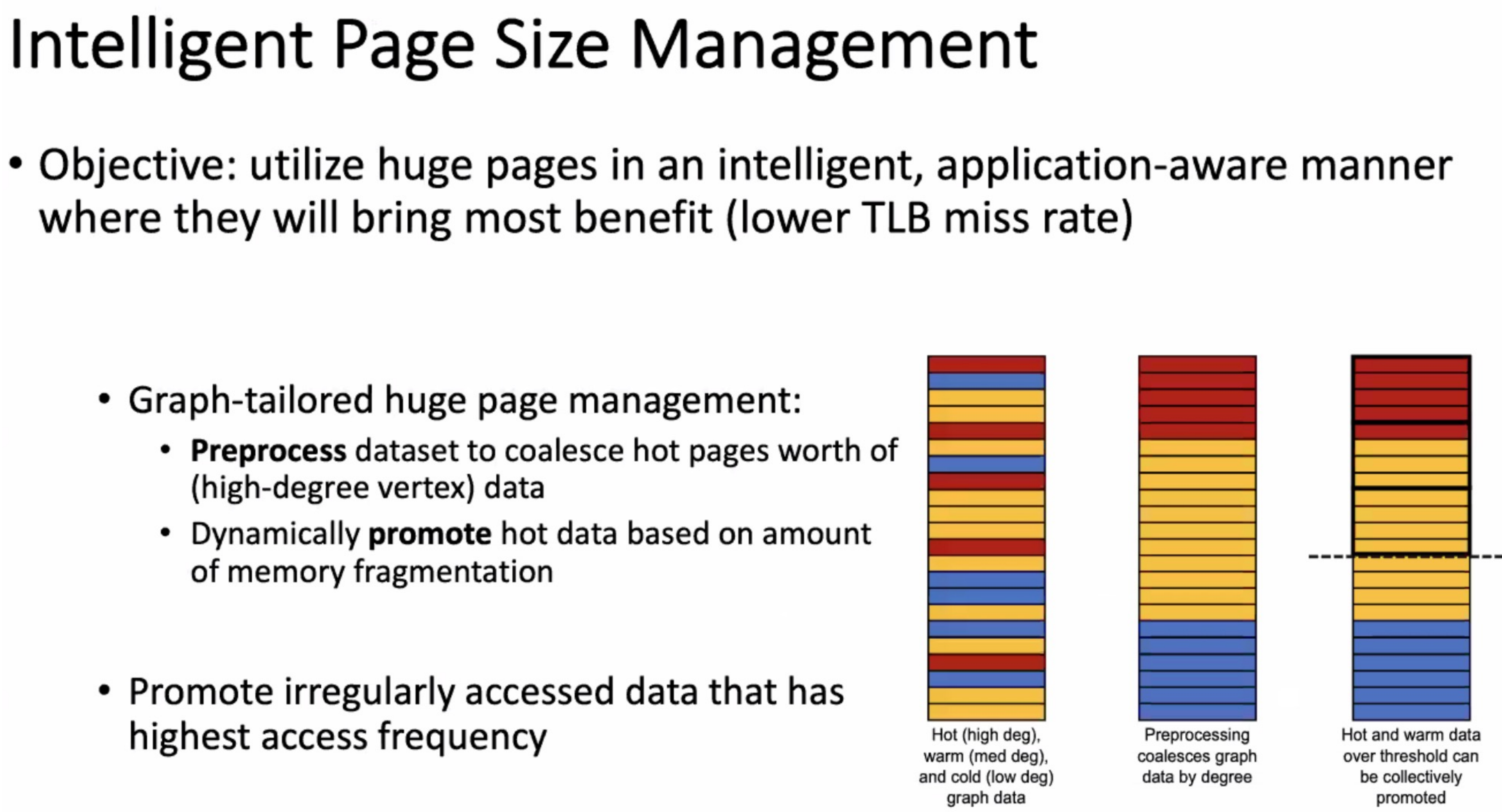


TLB miss rates without (left) and with (right) THPs; graph analytics have high miss rates compared to dense apps



Runtime speedups without (left) and with (right) THPs; Linux THP causes slowdown when memory is constrained

Page Size



Intelligent Page Size Management

- Objective: utilize huge pages in an intelligent, application-aware manner where they will bring most benefit (lower TLB miss rate)

- Graph-tailored huge page management:
  - **Preprocess** dataset to coalesce hot pages worth of (high-degree vertex) data
  - Dynamically **promote** hot data based on amount of memory fragmentation
- Promote irregularly accessed data that has highest access frequency

Hot (high deg), warm (med deg), and cold (low deg) graph data

Preprocessing coalesces graph data by degree

Hot and warm data over threshold can be collectively promoted

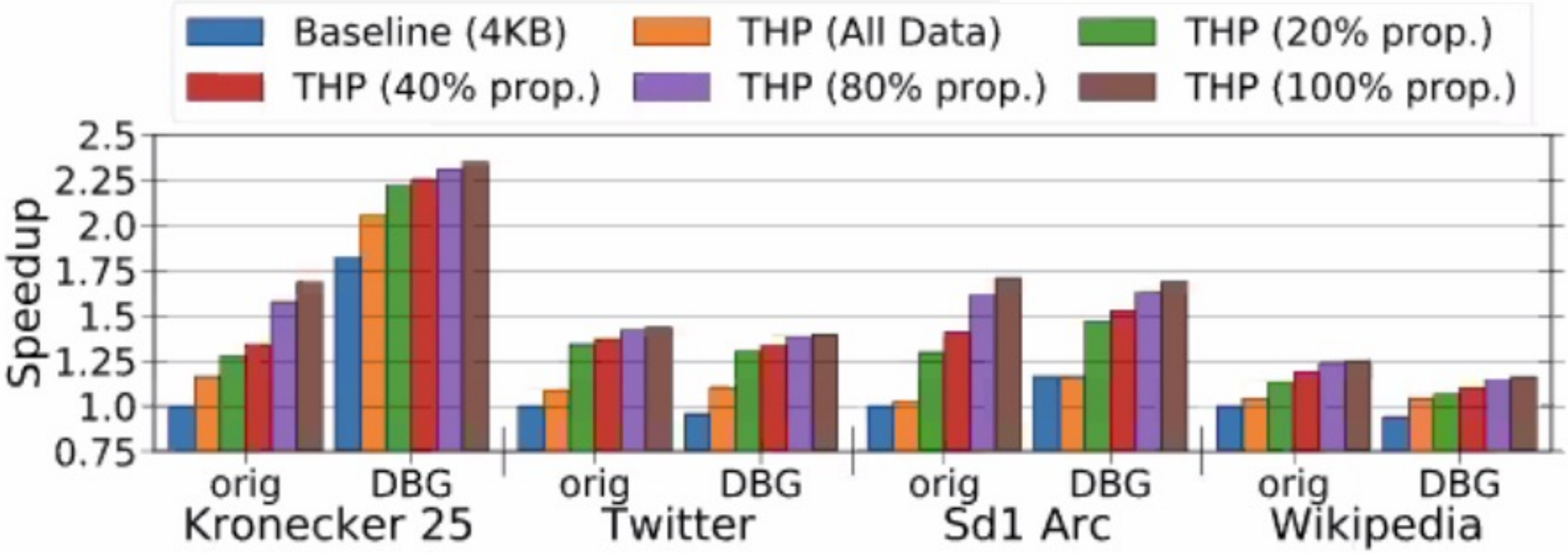
Page Size

Results and Takeaways

- Leveraging application knowledge for huge page allocation and placement best optimizes performance improvements from huge pages in real systems

- For graph analytics, utilize huge pages selectively for hottest percentage of **property array** (frequently and irregularly accessed data)

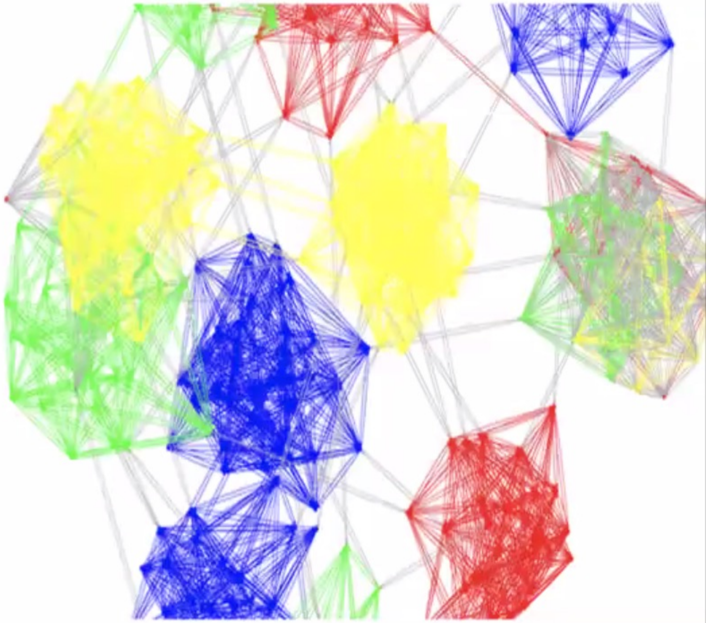
- **1.26-1.57x** speedup over 4KB pages
  - **77.3-96.3%** of ideal THP performance
  - Requires only **0.58-2.52%** of application footprint to be backed by huge pages



Runtime speedups comparing THPs applied system-wide vs. selectively to percentage of preprocessed TLB-sensitive prop. array

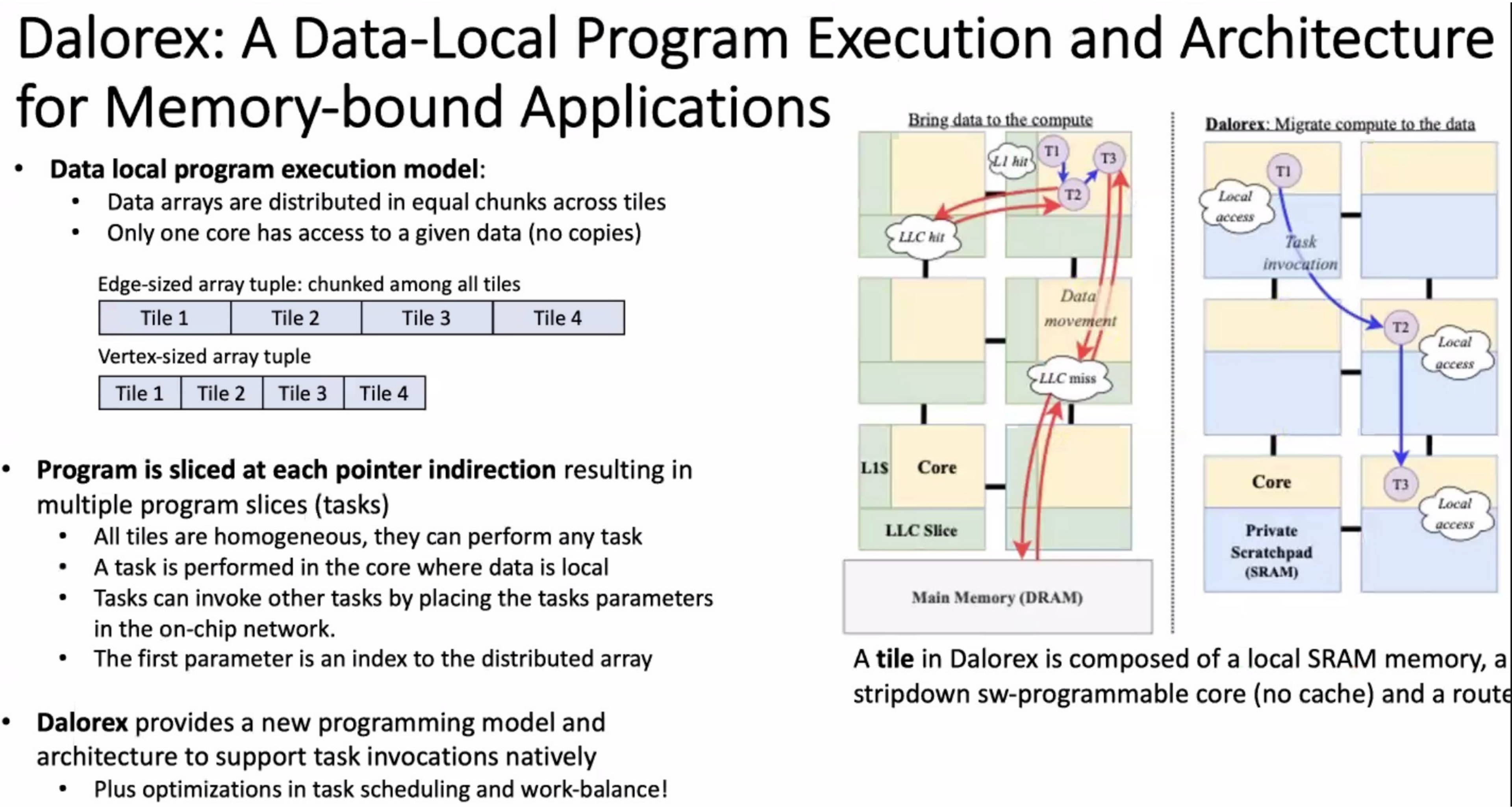
Example 2: Hardware and Programming Models for Sparse/Graph Applications

- Graph analytics and memory bottlenecks
- Challenges:
  - Little compute per loaded cache line
  - Little data reuse
  - >50% of accesses go to main memory
  - >95% of total energy spent on memory operations
- Prior work mitigates the memory latency, but bandwidth and synchronization remains a problem when scaling to high core counts



Orenes-Vera, Tureci, Wentzlaff, Martonosi. Dalorex: A Data-Local Program Execution and Architecture for Memory-Bound Applications". ArXiv July 2022

Dalorex



Dalorex: A Data-Local Program Execution and Architecture for Memory-bound Applications

- **Data local program execution model**:
  - Data arrays are distributed in equal chunks across tiles
  - Only one core has access to a given data (no copies)

Edge-sized array tuple: chunked among all tiles

Tile 1	Tile 2	Tile 3	Tile 4

Vertex-sized array tuple

Tile 1	Tile 2	Tile 3	Tile 4

- **Program is sliced at each pointer indirection** resulting in multiple program slices (tasks)
  - All tiles are homogeneous, they can perform any task
  - A task is performed in the core where data is local
  - Tasks can invoke other tasks by placing the tasks parameters in the on-chip network.
  - The first parameter is an index to the distributed array
- **Dalorex** provides a new programming model and architecture to support task invocations natively
  - Plus optimizations in task scheduling and work-balance!

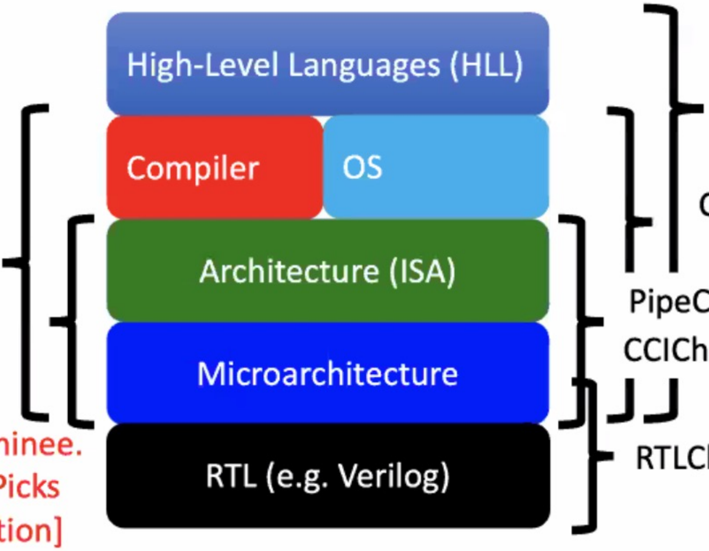
A **tile** in Dalorex is composed of a local SRAM memory, a stripdown sw-programmable core (no cache) and a route

Margaret Martonosi

Example 3: The Check Suite: An Ecosystem of Tools For Early-Stage Verification and Example Synthesis

CheckMate [Micro '18] [IEEE Micro Top Picks]

PipeProof [Micro '18] [Best Paper Nominee. IEEE Micro Top Picks Honorable Mention]



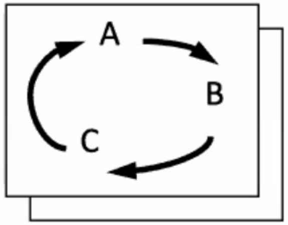
TriCheck [ASPLOS '17] [IEEE MICRO Top Picks]

COATCheck [ASPLOS '16] [IEEE MICRO Top Picks]

PipeCheck [Micro '14] [IEEE MICRO Top Picks]

CCICheck [Micro '15] [Nominated for Best Paper Award]

RTLCheck [Micro '17] [IEEE MICRO Top Picks Honorable Mention]



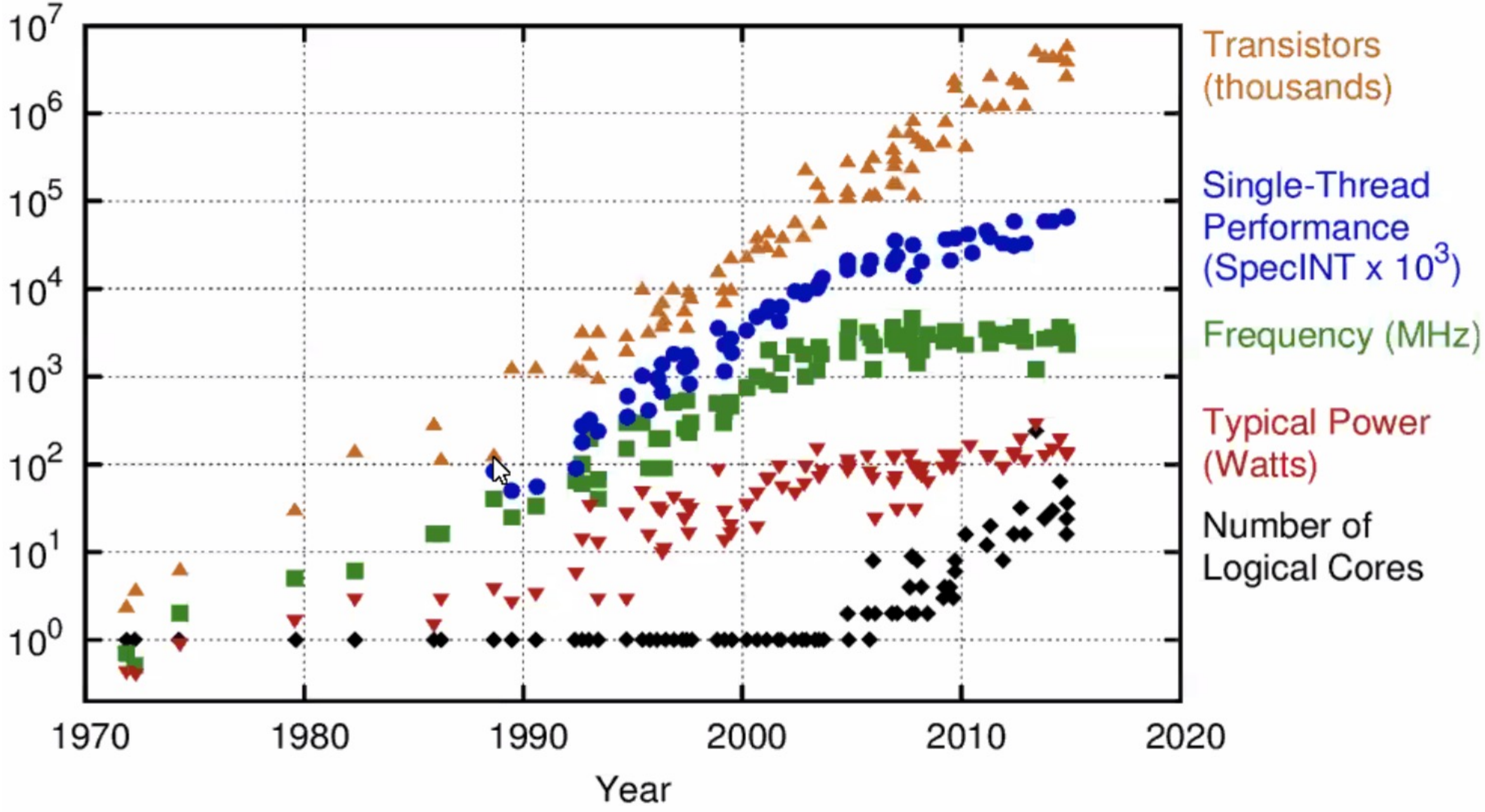
Our Approach

- Axiomatic specifications -> Happens-before graphs
- Check Happens-Before Graphs via Efficient SMT solvers
  - Cyclic => A->B->C->A... Can't happen

Moore's Law

Decades of Moore's Law scaling

40 Years of Microprocessor Trend Data



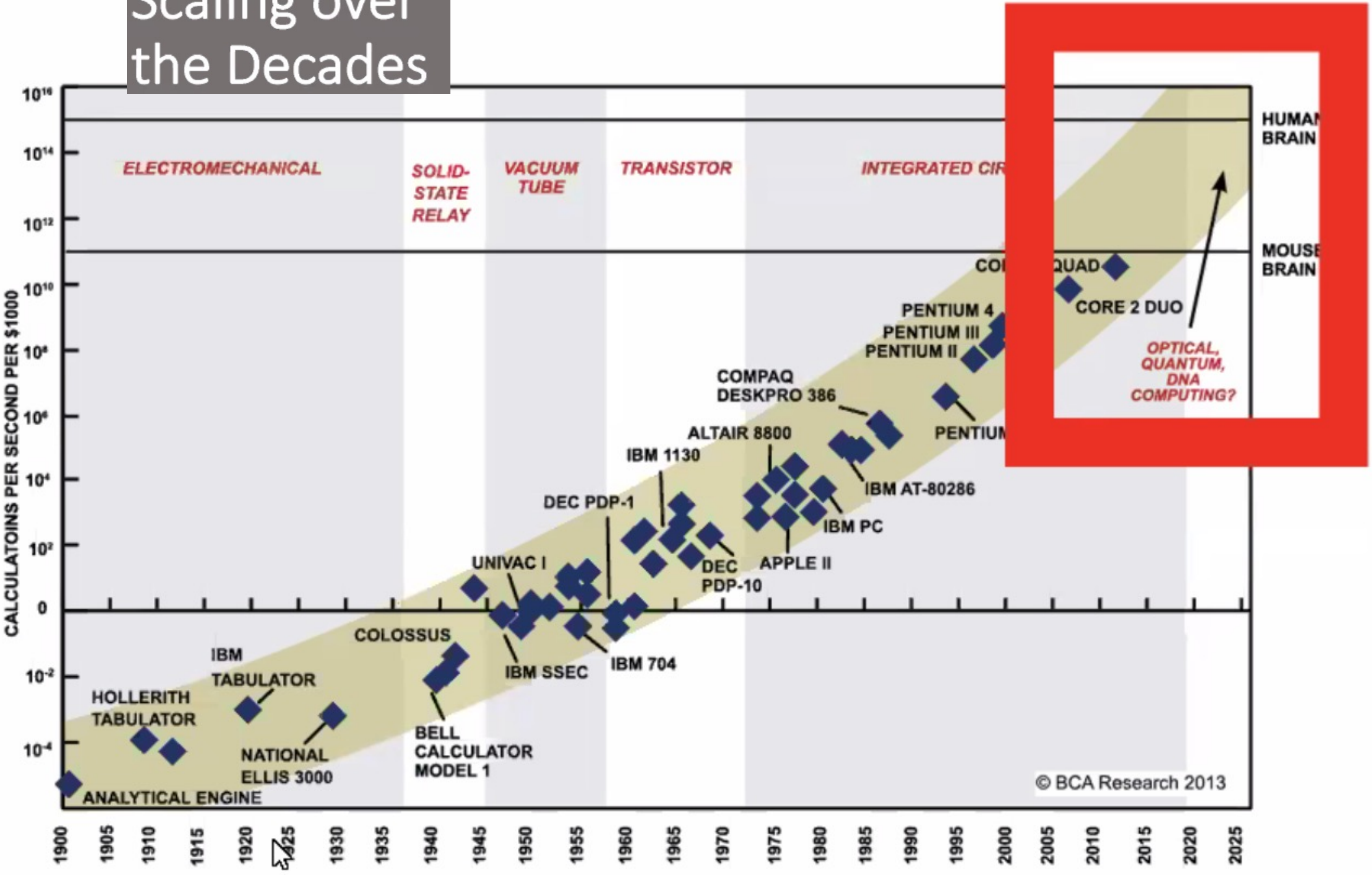
Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten
New plot and data collected for 2010-2015 by K. Rupp

1947 75 2022 acm Association for Computing Machinery

ACM Tech Talk

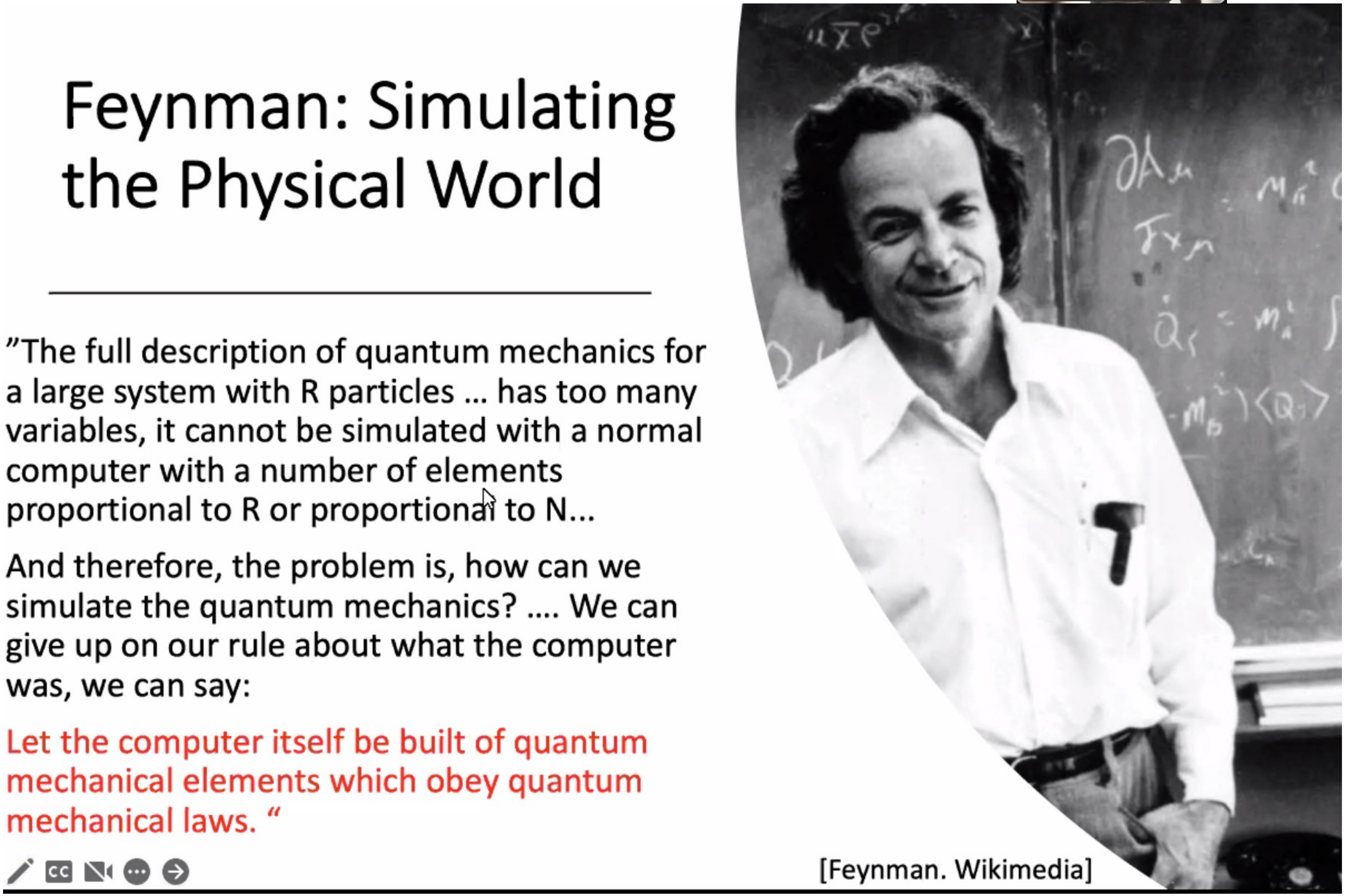
DSJ Dr Jeff DR JEFF SOFTWARE INDIE APP DEVELOPER
© Jeff Drobman 2016-23

Performance Scaling over the Decades



SOURCE: RAY KURZWEIL, "THE SINGULARITY IS NEAR: WHEN HUMANS TRANSCEND BIOLOGY", P.67, *THE VIKING PRESS*, 2006. DATAPOINTS BETWEEN 2000 AND 2012 REPRESENT BCA ESTIMATES.

Margaret Martonosi



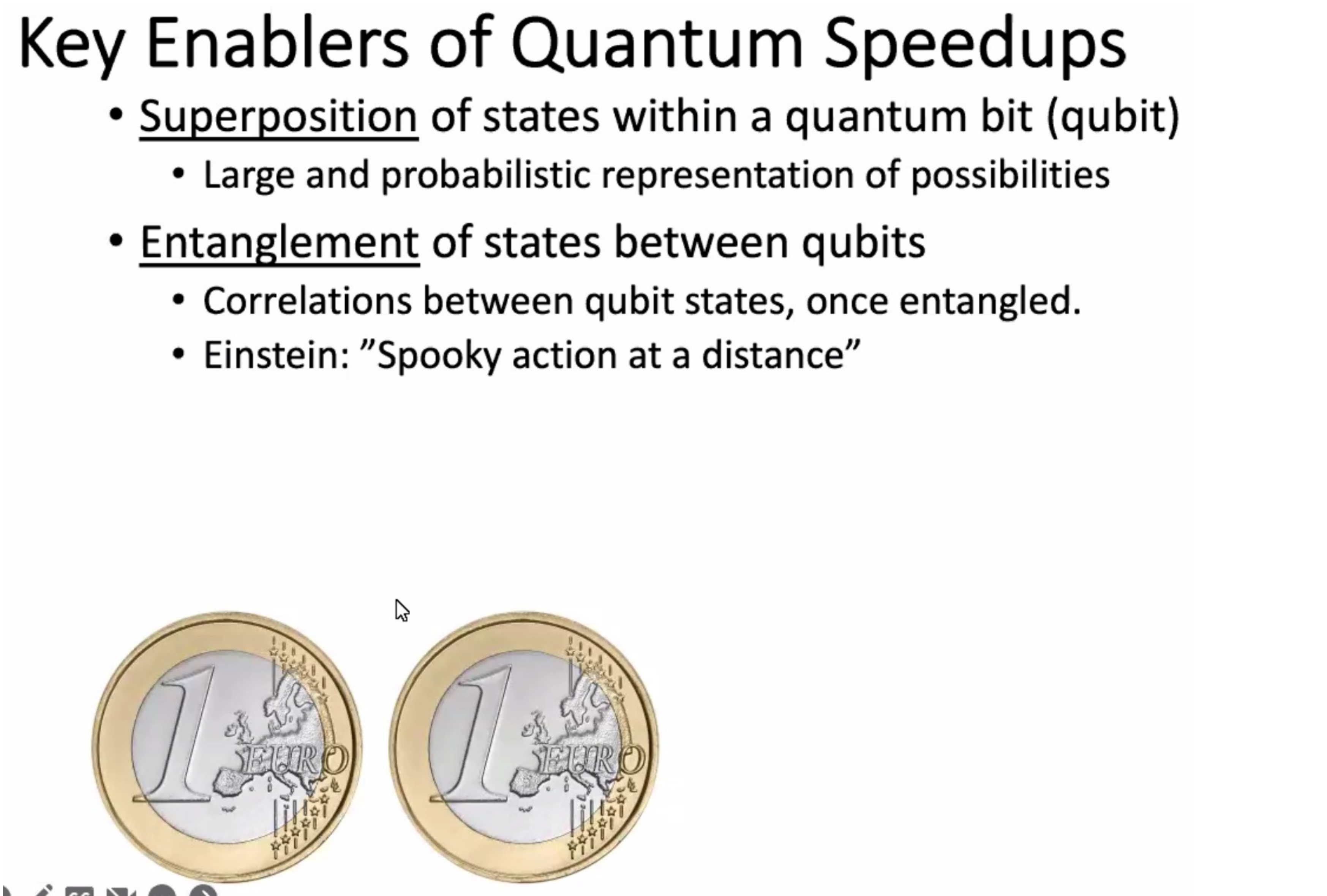
Feynman: Simulating the Physical World

”The full description of quantum mechanics for a large system with R particles … has too many variables, it cannot be simulated with a normal computer with a number of elements proportional to R or proportional to N…

And therefore, the problem is, how can we simulate the quantum mechanics? …. We can give up on our rule about what the computer was, we can say:

Let the computer itself be built of quantum mechanical elements which obey quantum mechanical laws. “

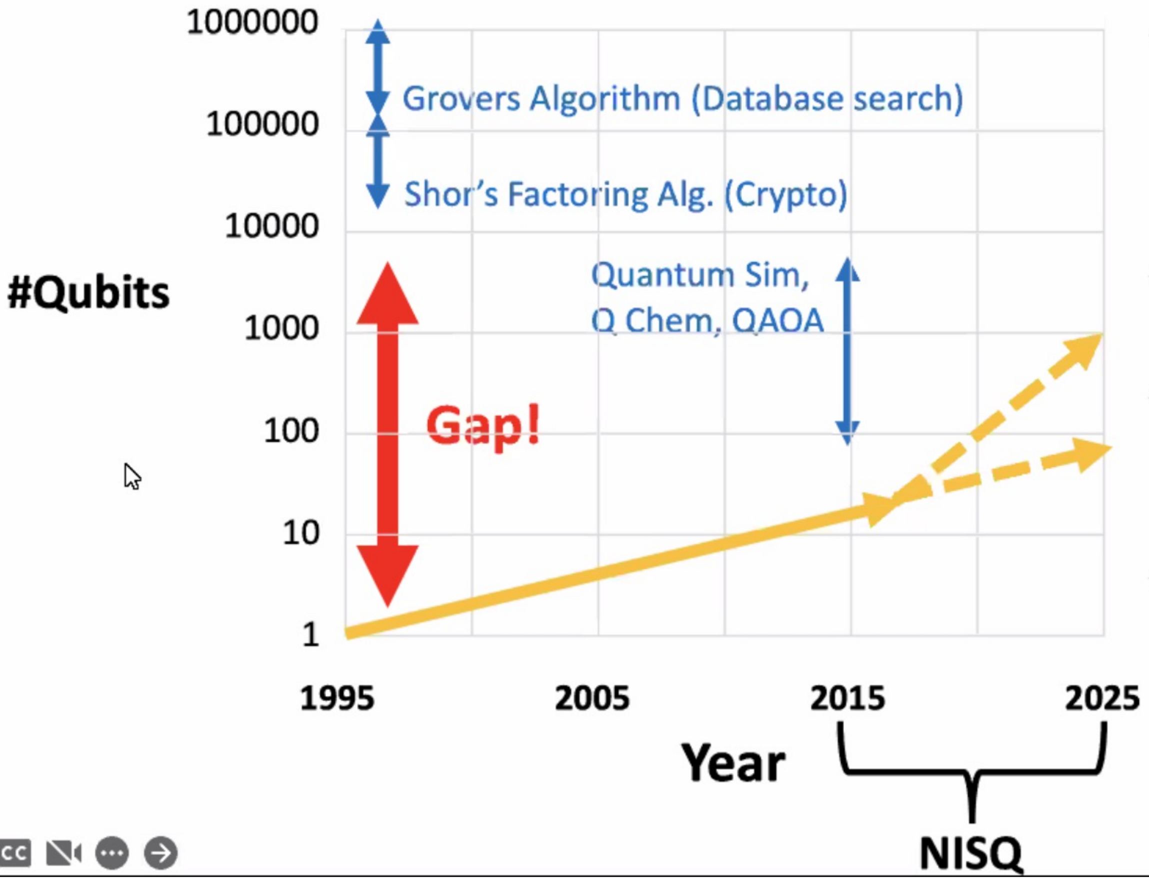
[Feynman. Wikimedia]



Key Enablers of Quantum Speedups

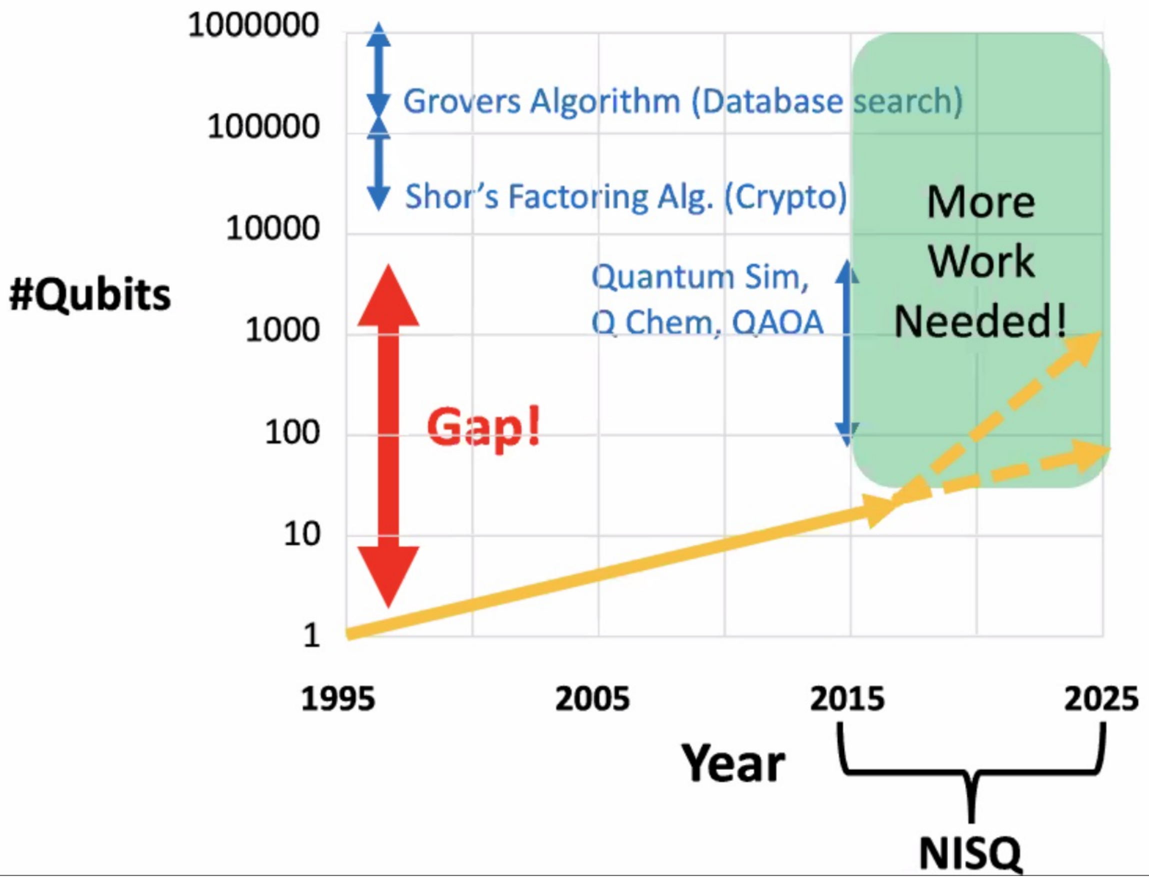
- Superposition of states within a quantum bit (qubit)
  - Large and probabilistic representation of possibilities
- Entanglement of states between qubits
  - Correlations between qubit states, once entangled.
  - Einstein: ”Spooky action at a distance”

QC Algorithms to Machines Gap: The NISQ Era



- Noisy Intermediate-Scale Quantum (NISQ)
  - Preskill, Jan 2018
  - 10-1000 qubits
- Too small for known algorithms with exponential speedup
- Too small for ECC
- Large enough to support interesting experiments!

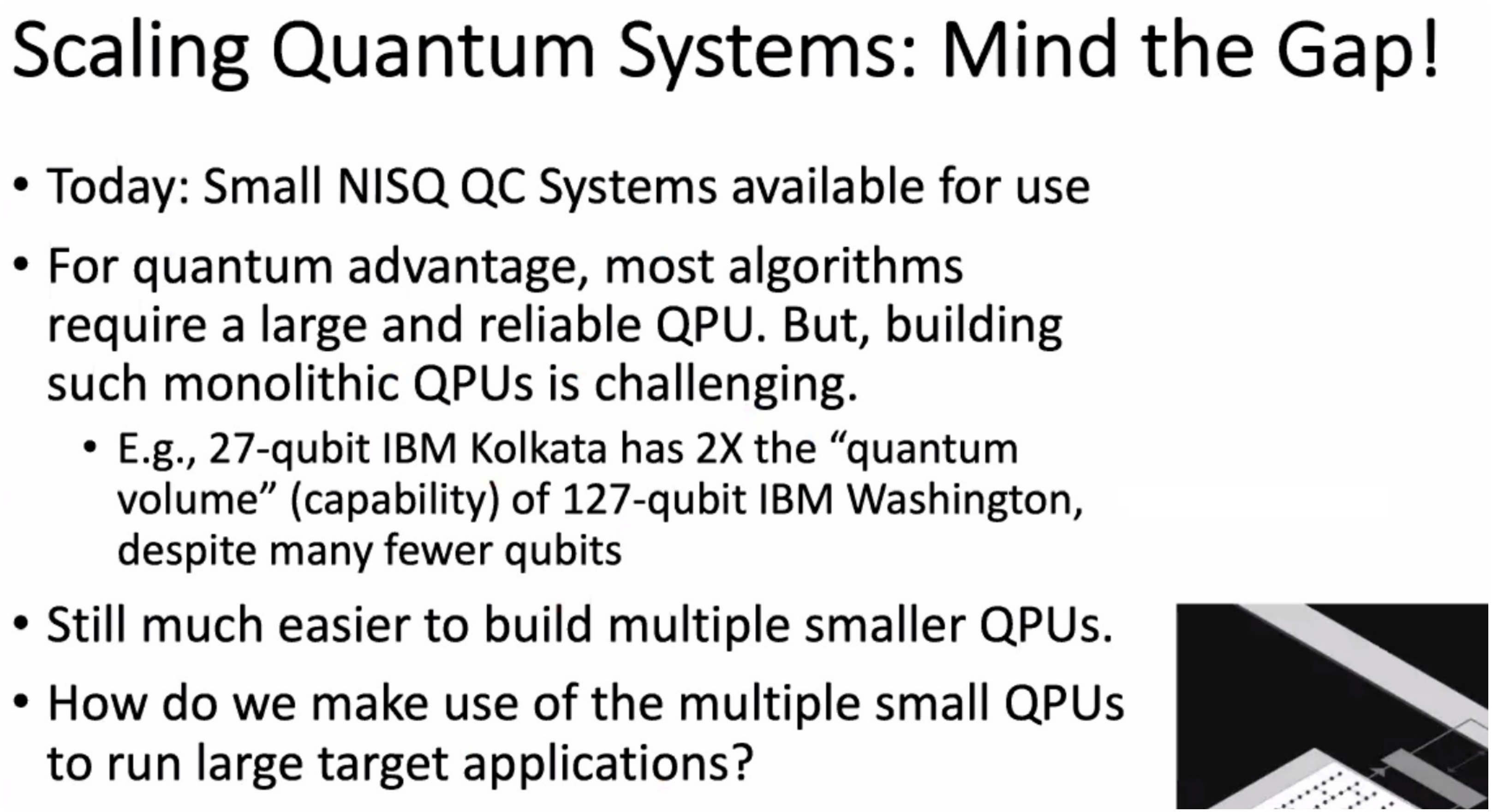
QC Algorithms to Machines Gap: Opportunity



QC programming and design tools that shrink the gap can move the feasibility point years sooner!

- Reduce algorithm qubit requirements
- Improve effectiveness of hardware qubits

Margaret Martonosi



Scaling Quantum Systems: Mind the Gap!

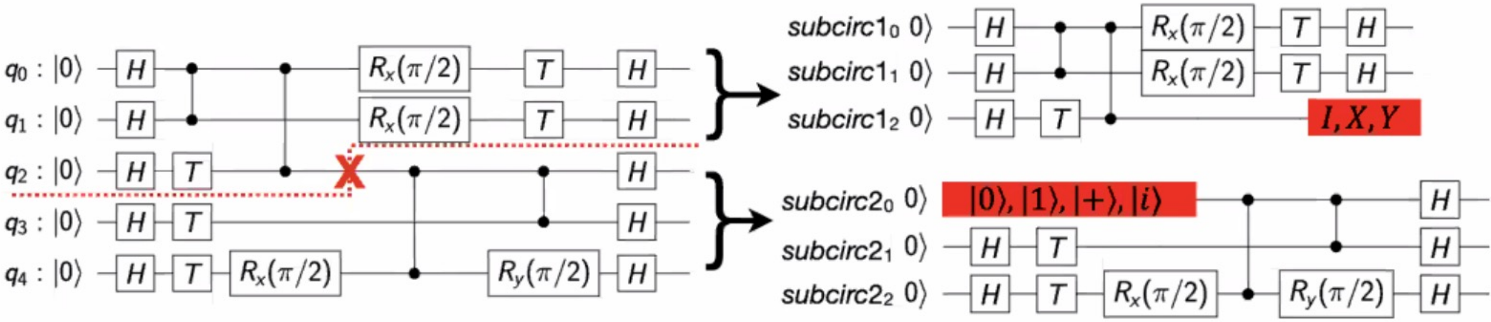
- Today: Small NISQ QC Systems available for use
- For quantum advantage, most algorithms require a large and reliable QPU. But, building such monolithic QPUs is challenging.
  - E.g., 27-qubit IBM Kolkata has 2X the “quantum volume” (capability) of 127-qubit IBM Washington, despite many fewer qubits
- Still much easier to build multiple smaller QPUs.
- How do we make use of the multiple small QPUs to run large target applications?

Margaret Martonosi

Example 4: CutQC: Combining Classical and Quantum Computation to Run QC algorithms at Larger Scale

- Approach: Cut quantum circuits into smaller subcircuits that fit and reconstruct the results classically afterward.
- Challenge: Classical reconstruction scales exponentially!
- Solution: parallel processing[1] and GPU[2].

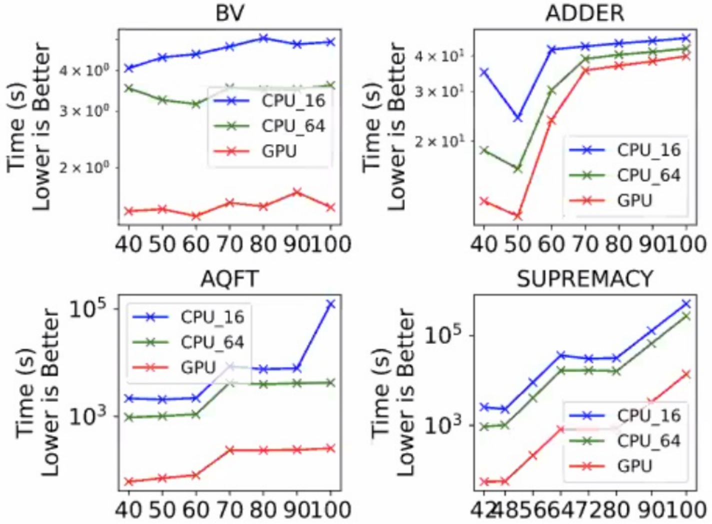
Example: Cut one edge to split a 5-qubit circuit into two smaller (3-qubit each) subcircuits.



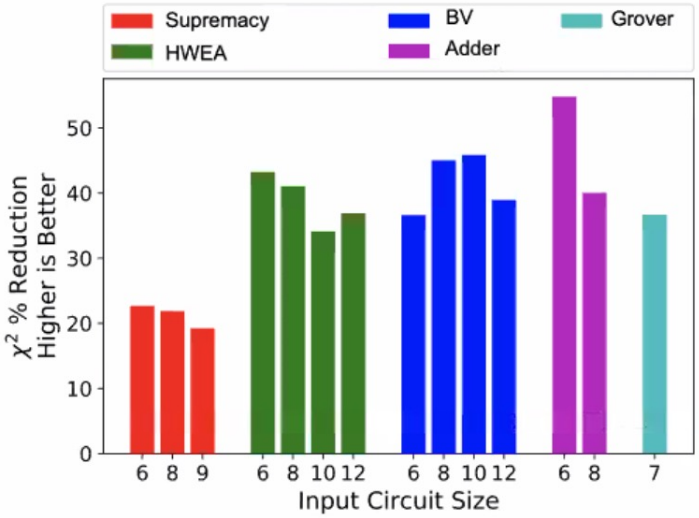
[1] Tang, Wei, Teague Tomesh, Martin Suchara, Jeffrey Larson, and Margaret Martonosi. "Cutqc: using small quantum computers for large quantum circuit evaluations." In *Proceedings of the 26th ACM International conference on architectural support for programming languages and operating systems*, pp. 473-486. 2021.

[2] Tang, Wei, and Margaret Martonosi. "Cutting Quantum Circuits to Run on Quantum and Classical Platforms." *arXiv preprint arXiv:2205.05836* (2022).

Result: Runtime and Fidelity Improvements



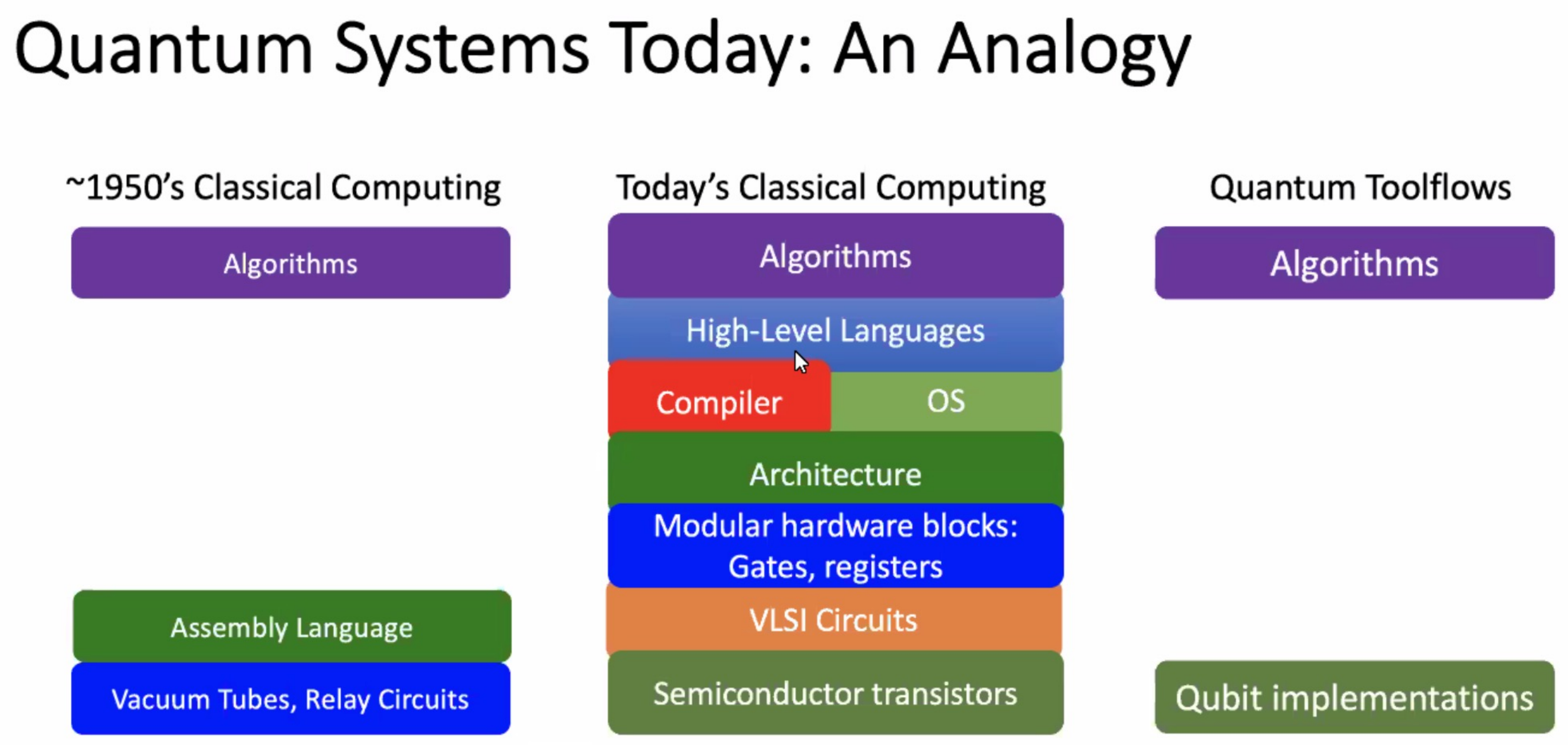
Faster than classical.



Higher fidelity than large monolithic QPUs.

- Cut and run benchmarks with up to 75% of number of qubits in input circuits.
- Runtime shows the reconstruction of 2^{30} bins. GPU is the fastest backend as expected.
- CutQC achieves an average of 21% to 47% fidelity improvement

Margaret Martonosi

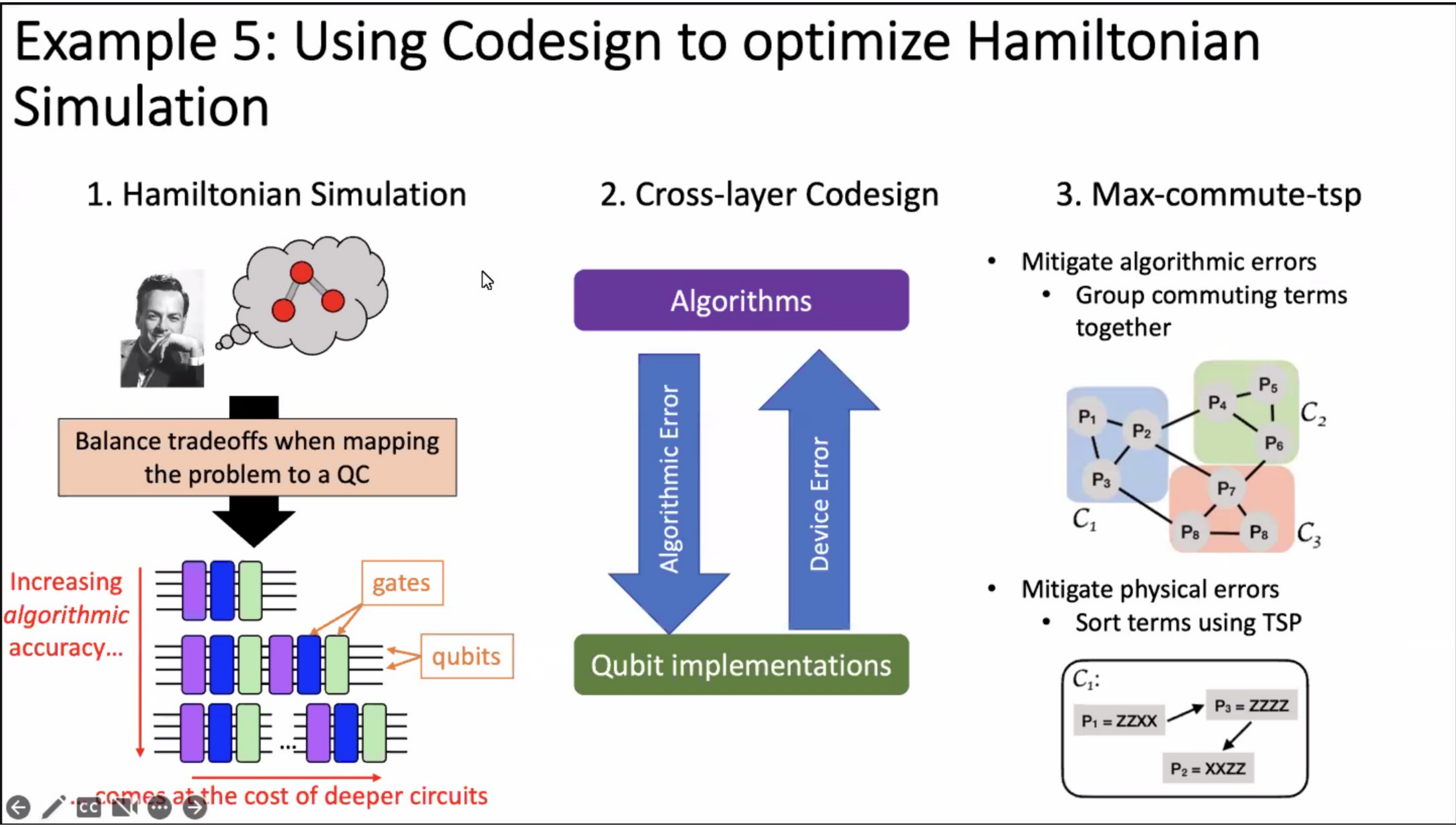


Quantum Systems Today: An Analogy

~1950's Classical Computing
- Algorithms
- Assembly Language
- Vacuum Tubes, Relay Circuits

Today's Classical Computing
- Algorithms
- High-Level Languages
- Compiler | OS
- Architecture
- Modular hardware blocks: Gates, registers
- VLSI Circuits
- Semiconductor transistors

Quantum Toolflows
- Algorithms
- Qubit implementations



Example 5: Using Codesign to optimize Hamiltonian Simulation

1. Hamiltonian Simulation

Balance tradeoffs when mapping the problem to a QC

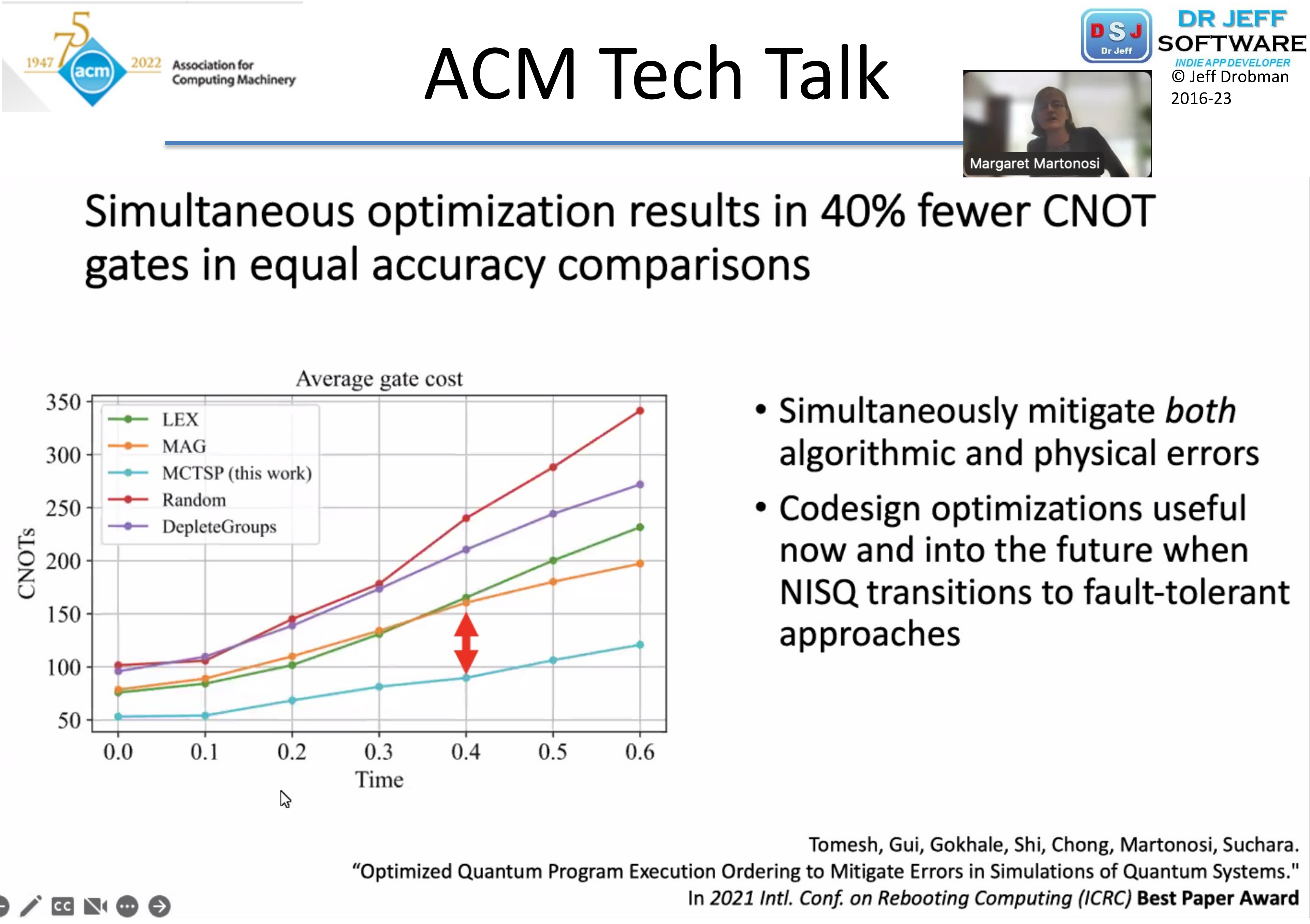
…comes at the cost of deeper circuits

2. Cross-layer Codesign

3. Max-commute-tsp

- Mitigate algorithmic errors
  - Group commuting terms together

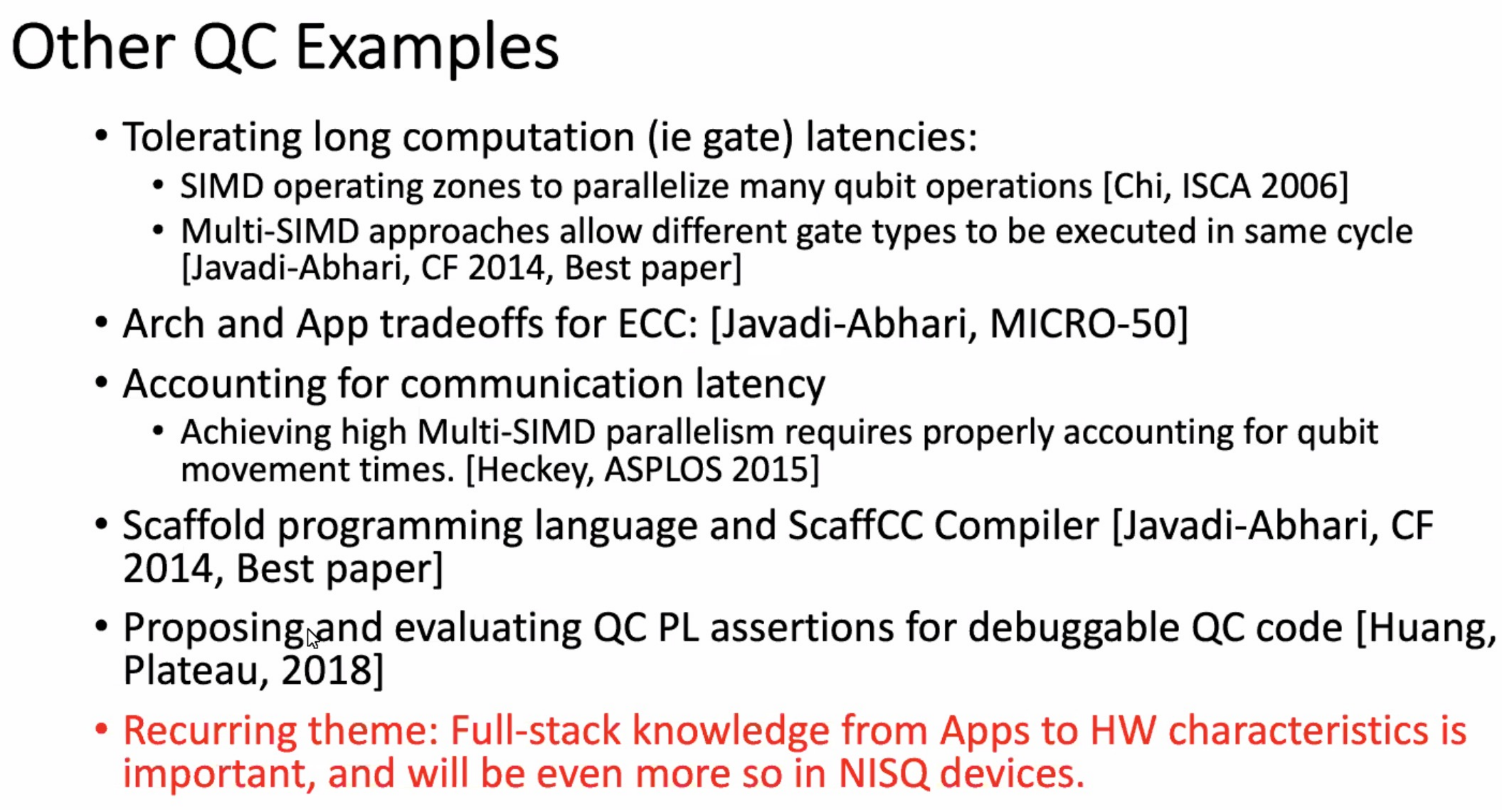
- Mitigate physical errors
  - Sort terms using TSP



Simultaneous optimization results in 40% fewer CNOT gates in equal accuracy comparisons

- Simultaneously mitigate *both* algorithmic and physical errors
- Codesign optimizations useful now and into the future when NISQ transitions to fault-tolerant approaches

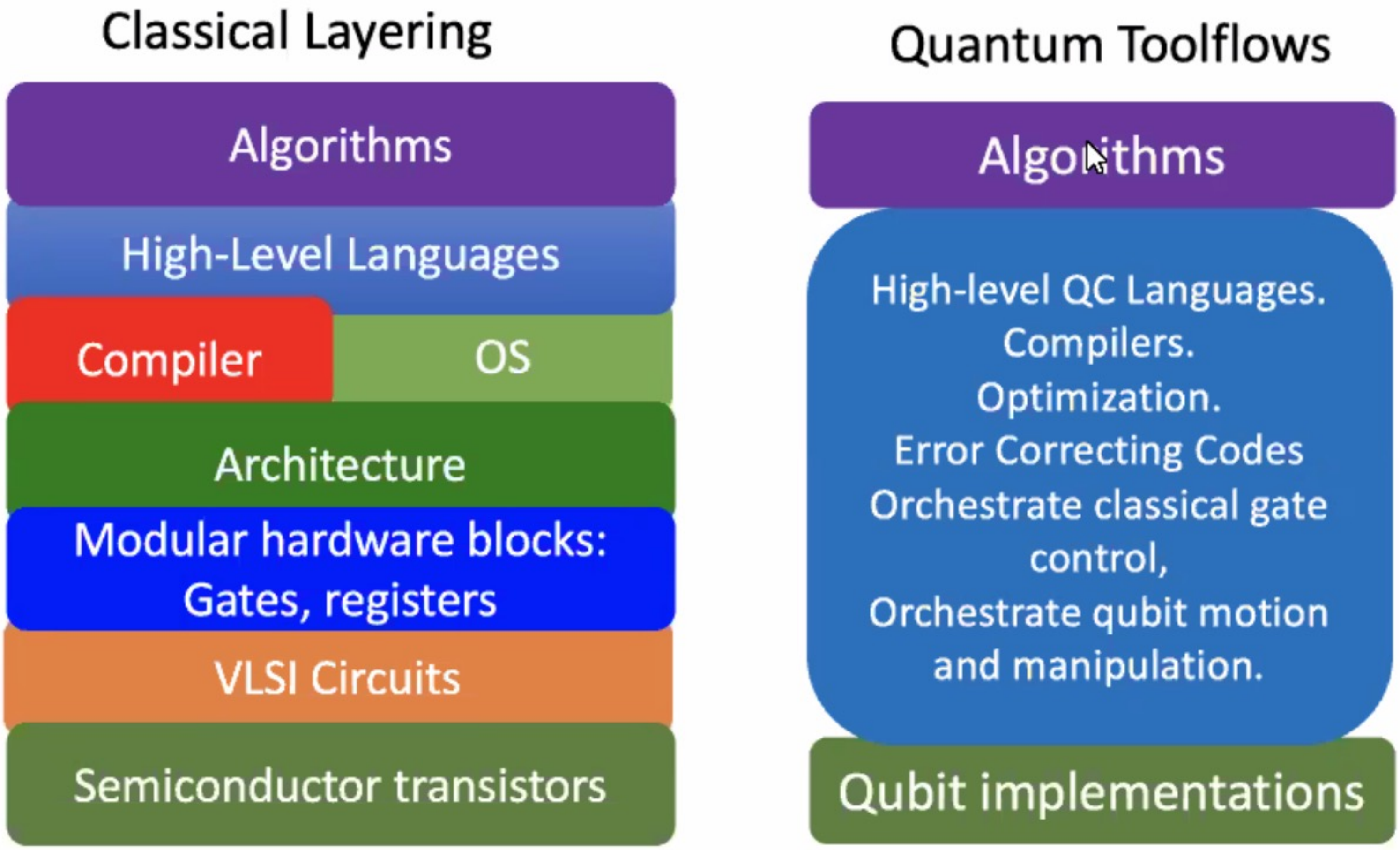
Tomesh, Gui, Gokhale, Shi, Chong, Martonosi, Suchara.
"Optimized Quantum Program Execution Ordering to Mitigate Errors in Simulations of Quantum Systems."
In *2021 Intl. Conf. on Rebooting Computing (ICRC)* **Best Paper Award**



Other QC Examples

- Tolerating long computation (ie gate) latencies:
  - SIMD operating zones to parallelize many qubit operations [Chi, ISCA 2006]
  - Multi-SIMD approaches allow different gate types to be executed in same cycle [Javadi-Abhari, CF 2014, Best paper]
- Arch and App tradeoffs for ECC: [Javadi-Abhari, MICRO-50]
- Accounting for communication latency
  - Achieving high Multi-SIMD parallelism requires properly accounting for qubit movement times. [Heckey, ASPLOS 2015]
- Scaffold programming language and ScaffCC Compiler [Javadi-Abhari, CF 2014, Best paper]
- Proposing and evaluating QC PL assertions for debuggable QC code [Huang, Plateau, 2018]
- Recurring theme: Full-stack knowledge from Apps to HW characteristics is important, and will be even more so in NISQ devices.

Quantum Systems: Layering Options

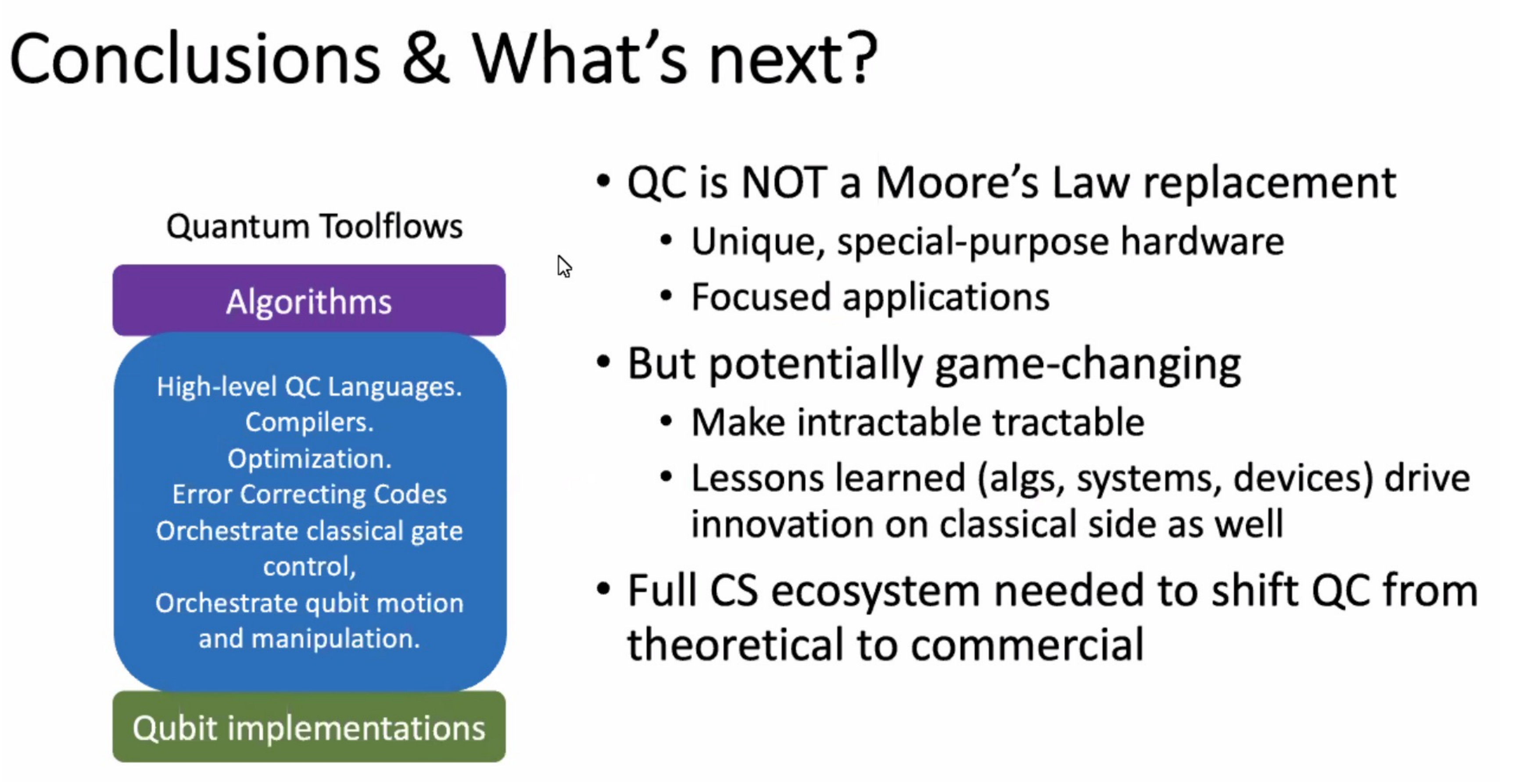


Classical Layering

- Algorithms
- High-Level Languages
- Compiler | OS
- Architecture
- Modular hardware blocks: Gates, registers
- VLSI Circuits
- Semiconductor transistors

Quantum Toolflows

- Algorithms
- High-level QC Languages. Compilers. Optimization. Error Correcting Codes Orchestrate classical gate control, Orchestrate qubit motion and manipulation.
- Qubit implementations



Conclusions & What's next?

Quantum Toolflows

Algorithms

High-level QC Languages.
Compilers.
Optimization.
Error Correcting Codes
Orchestrate classical gate control,
Orchestrate qubit motion and manipulation.

Qubit implementations

- QC is NOT a Moore's Law replacement
  - Unique, special-purpose hardware
  - Focused applications
- But potentially game-changing
  - Make intractable tractable
  - Lessons learned (algs, systems, devices) drive innovation on classical side as well
- Full CS ecosystem needed to shift QC from theoretical to commercial