Breaking News…

- Google claims quantum supremacy
- IBM announces Quantum Computation Center and 53-qubit system

FINANCIAL TIMES

Quantum technologies + Add to myFT

Google claims to have reached quantum supremacy

Researchers say their quantum computer has calculated an impossible problem for ordinary machines

https://www.ft.com/content/b9bb4e54-dbc1-11e9-8f9b-77216ebe1f17

Quantum technologies + Add to myFT

Rivals rubbish Google's claim of quantum supremacy

Researchers take aim at tech company for declaring computing milestone

https://www.ft.com/content/cede11e0-dd51-11e9-9743-db5a370481bc

NASA/TP-2019-220319

Quantum Supremacy Using a Programmable Superconducting Processor

Eleanor G. Rieffel
NASA Ames Research Center

Quantum supremacy using a programmable superconducting processor

Google AI Quantum and collaborators[†]

The tantalizing promise of quantum computers is that certain computational tasks might be executed exponentially faster on a quantum processor than on a classical processor. A fundamental challenge is to build a high-fidelity processor capable of running quantum algorithms in an exponentially large computational space. Here, we report using a processor with programmable superconducting qubits to create quantum states on 53 qubits, occupying a state space $2^{53} \sim 10^{16}$. Measurements from repeated experiments sample the corresponding probability distribution, which we verify using classical simulations. 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. This dramatic speedup relative to all known classical algorithms provides an experimental realization of quantum supremacy on a computational task and heralds the advent of a much-anticipated computing paradigm.

Rod Van Meter @rdviii · 16h

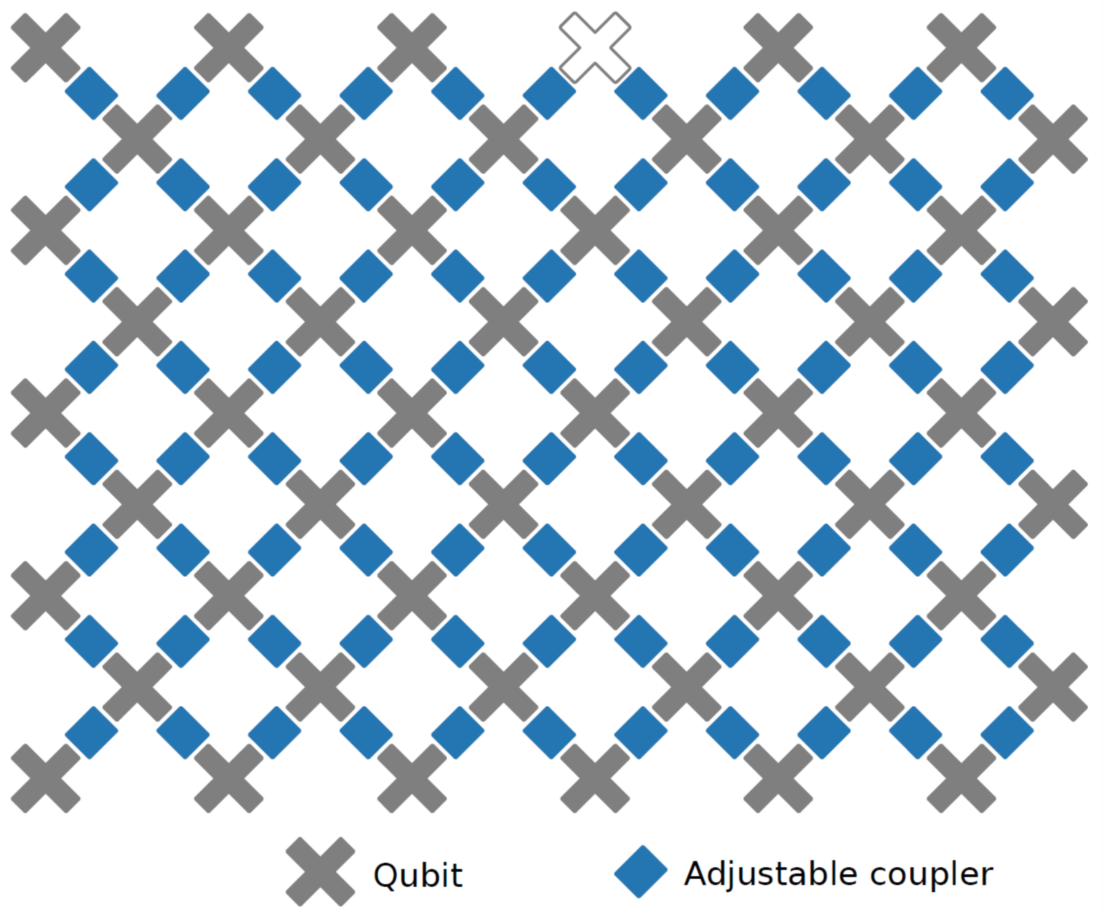
By my count, on the new Google supremacy paper, there are 1.43 authors per qubit.

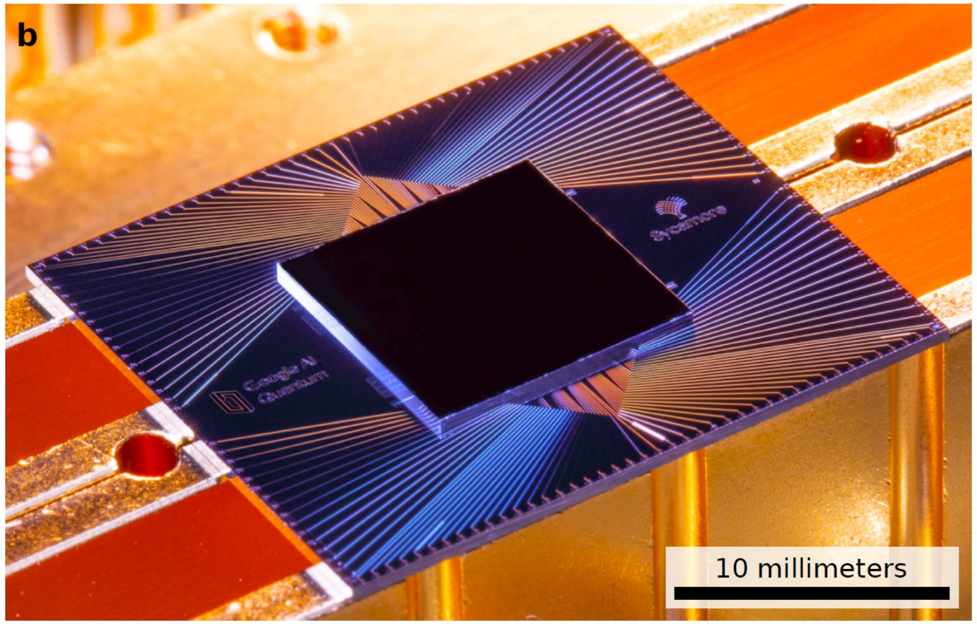
39

[†] Frank Arute[1], Kunal Arya[1], Ryan Babbush[1], Dave Bacon[1], Joseph C. Bardin[1,2], Rami Barends[1], Rupak Biswas[3], Sergio Boixo[1], Fernando G.S.L. Brandao[1,4], David Buell[1], Brian Burkett[1], Yu Chen[1], Zijun Chen[1], Ben Chiaro[5], Roberto Collins[1], William Courtney[1], Andrew Dunsworth[1], Edward Farhi[1], Brooks Foxen[5], Austin Fowler[1], Craig Gidney[1], Marissa Giustina[1], Rob Graff[1], Keith Guerin[1], Steve Habegger[1], Matthew P. Harrigan[1], Michael J. Hartmann[1,6], Alan Ho[1], Markus Hoffmann[1], Trent Huang[1], Travis S. Humble[7], Sergei V. Isakov[1], Evan Jeffrey[1], Zhang Jiang[1], Dvir Kafri[1], Kostyantyn Kechedzhi[1], Julian Kelly[1], Paul V. Klimov[1], Alexander Korotkov[1], Fedor Kostritsa[1], David Landhuis[1], Mike Lindmark[1], Erik Lucero[1], Dmitry Lyakh[7], Salvatore Mandrá[3], Jarrod R. McClean[1], Matt McEwen[5], Anthony Megrant[1], Xiao Mi[1], Kristel Michielsen[8], Masoud Mohseni[1], Josh Mutus[1], Ofer Naaman[1], Matthew Neeley[1], Charles Neill[1], Murphy Yuezhen Niu[1], Eric Ostby[1], Andre Petukhov[1], John C. Platt[1], Chris Quintana[1], Eleanor G. Rieffel[3], Pedram Roushan[1], Nicholas Rubin[1], Daniel Sank[1], Kevin J. Satzinger[1], Vadim Smelyanskiy[1], Kevin Sung[1], Matthew D. Trevithick[1], Amit Vainsencher[1], Benjamin Villalonga[1,9], Theodore White[1], Jamie Yao[1], Ping Yeh[1], Adam Zalcman[1], Hartmut Neven[1], John M. Martinis[1,5]

1. Google Research, Mountain View, CA 94043, USA, 2. Department of Electrical and Computer Engineering, University of Massachusetts Amherst, Amherst, MA, USA, 3. Quantum Artificial Intelligence Lab. (QuAIL), NASA Ames Research Center, Moffett Field, USA, 4. Institute for Quantum Information and Matter, Caltech, Pasadena, CA, USA, 5. Department of Physics, University of California, Santa Barbara, CA, USA, 6. Friedrich-Alexander University Erlangen-Nürnberg (FAU), Department of Physics, Erlangen, Germany, 7. Quantum Computing Institute, Oak Ridge National Laboratory, Oak Ridge, TN, USA, 8. Institute for Advanced Simulation, Jülich Supercomputing Centre, Jülich, Germany, 9. Department of Physics, University of Illinois at Urbana-Champaign, Urbana, IL, USA

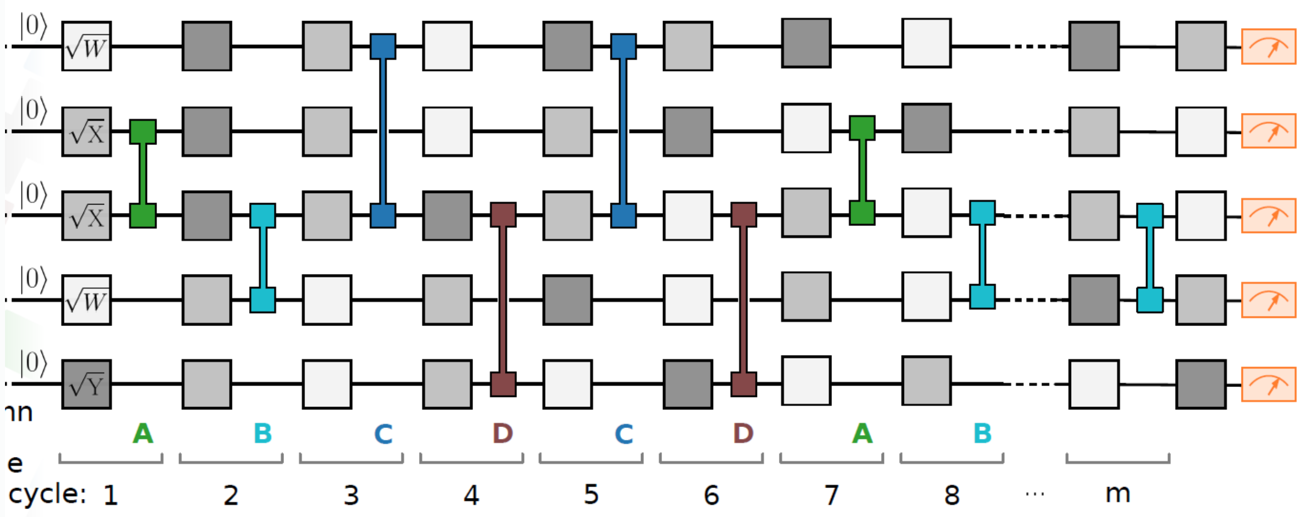
The Chip (Sycamore)





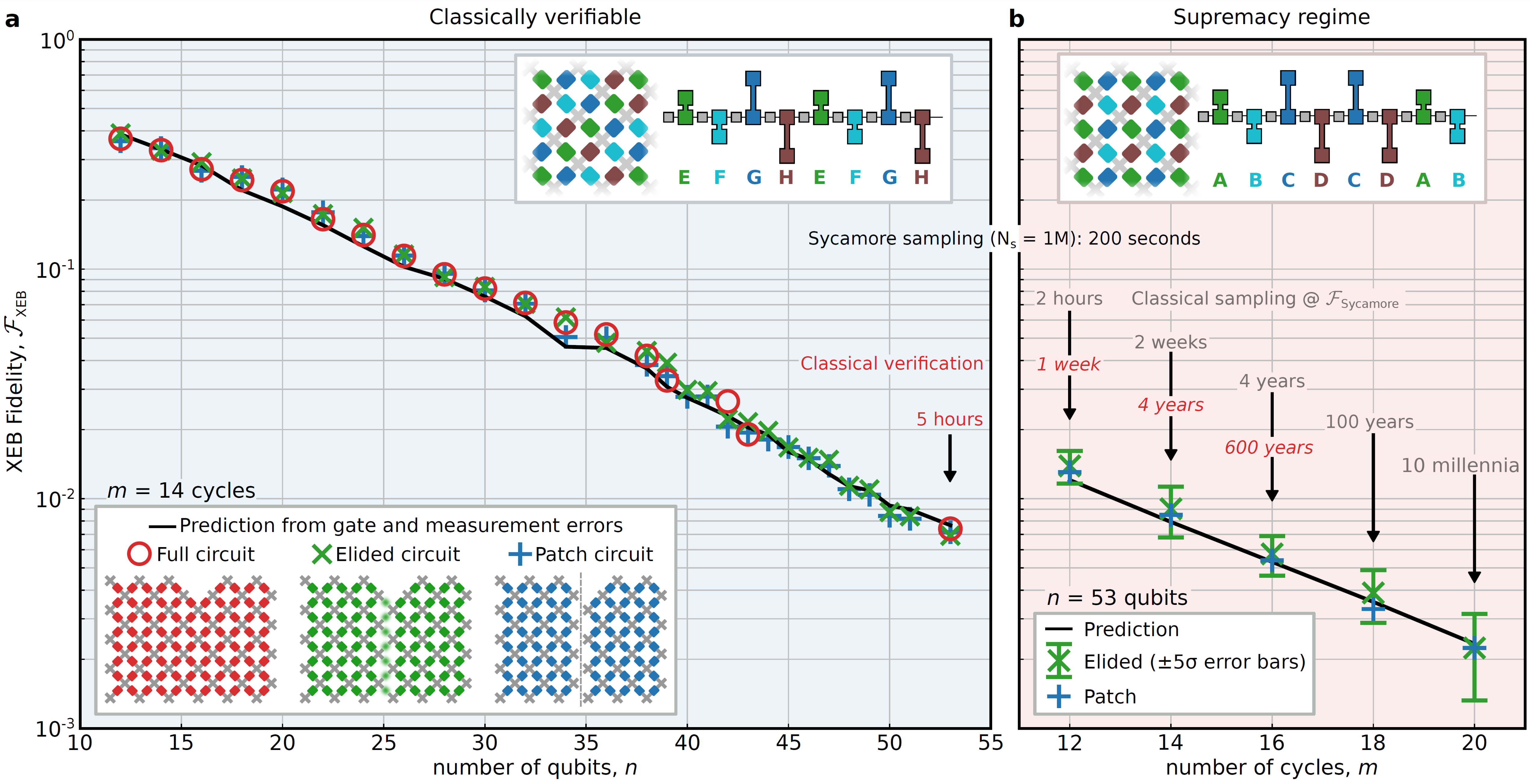
Quantum Supremacy

- What is the task?
  - sampling the output of a pseudo-random quantum circuit
  - probability distribution of the [measured] bitstrings resembles a speckled intensity pattern produced by light interference in laser scatter
  - classically computing […] becomes exponentially more difficult as the number of qubits (width) and number of gate cycles (depth) grows



$$\mathcal{F}_{\mathrm{XEB}} = 2^n \langle P(x_i) \rangle_i - 1$$

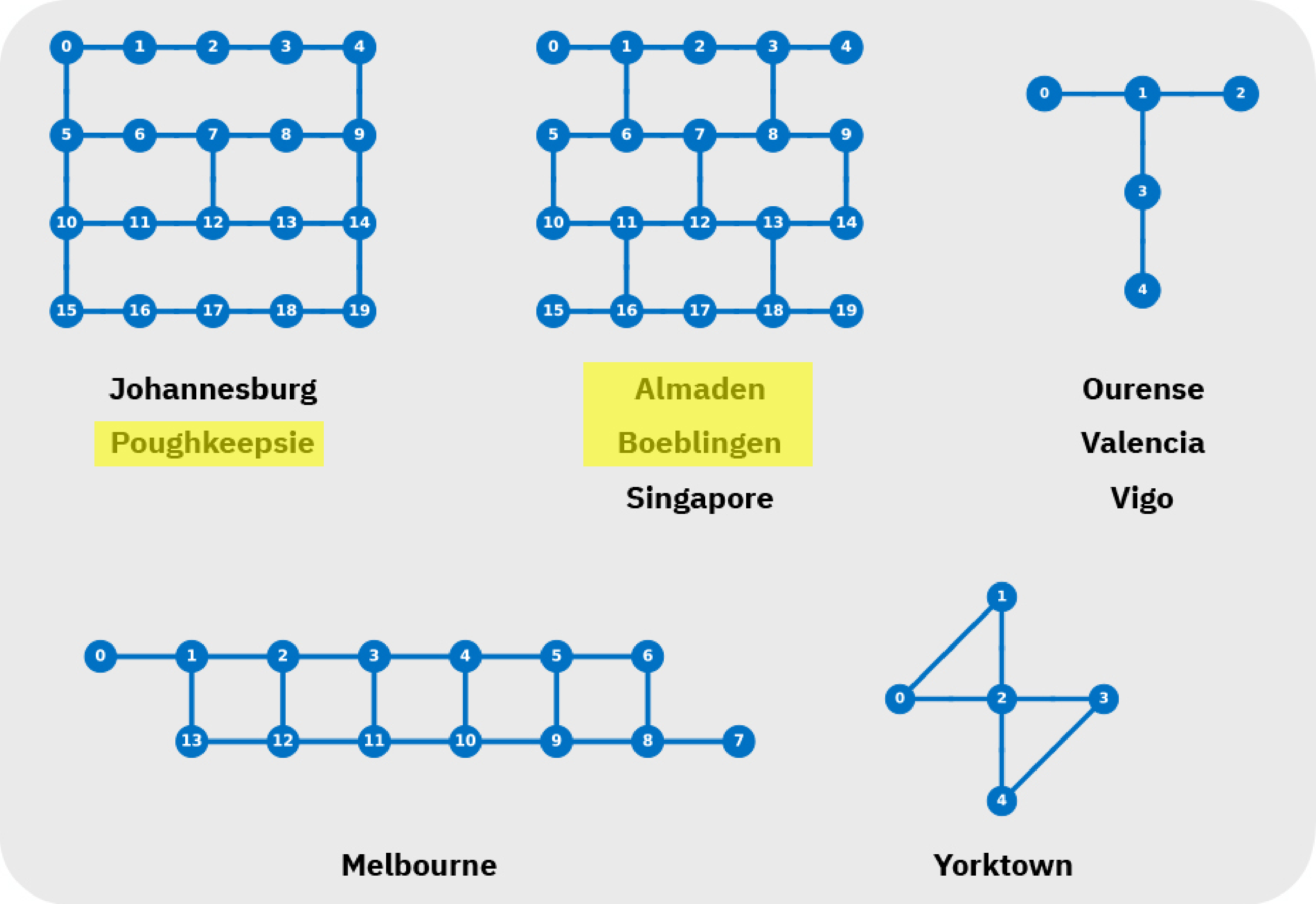
53 qubits, 1113 single-qubit gates, 430 two-qubit gates = 0.002



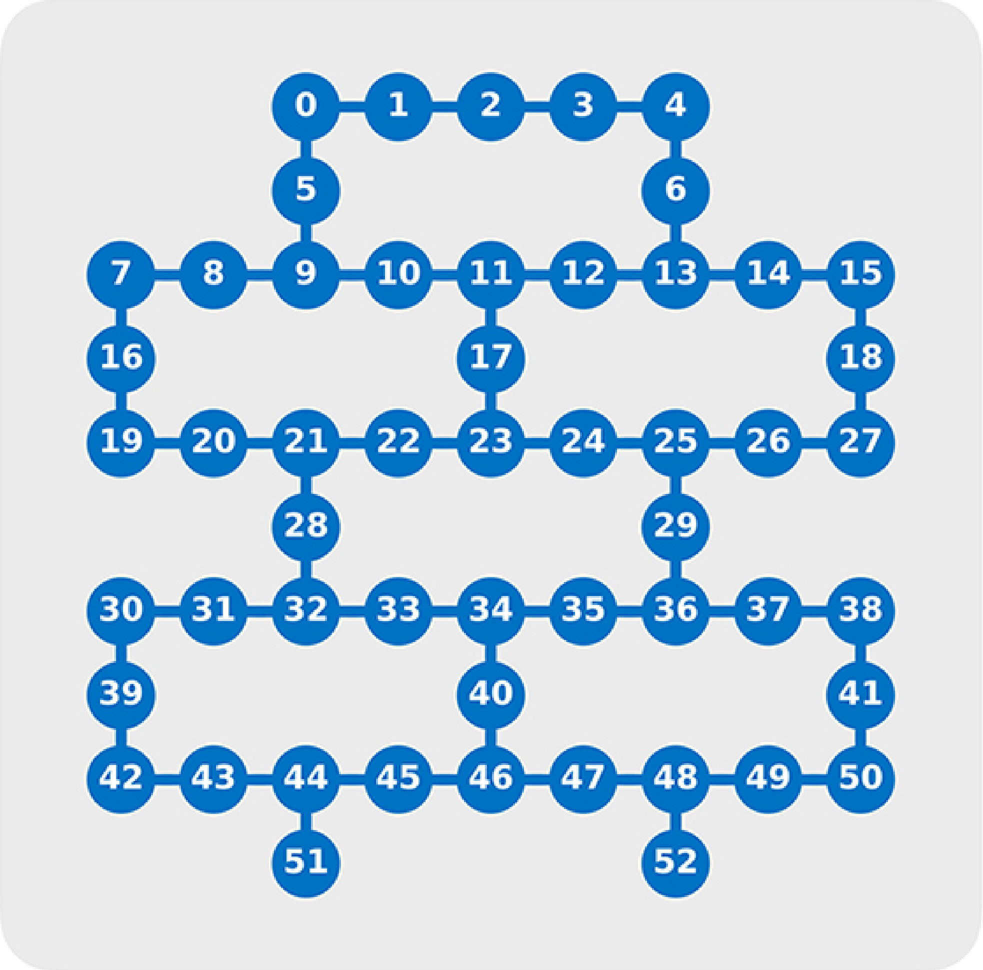
a
Classically verifiable
b
Supremacy regime
E F G H E F G H
A B C D C D A B
Sycamore sampling (N$_s$ = 1M): 200 seconds
Classical verification
5 hours
m = 14 cycles
Prediction from gate and measurement errors
Full circuit
Elided circuit
Patch circuit
XEB Fidelity, $\mathcal{F}_{XEB}$
number of qubits, n
2 hours
Classical sampling @ $\mathcal{F}_{Sycamore}$
1 week
2 weeks
4 years
4 years
600 years
100 years
10 millennia
n = 53 qubits
Prediction
Elided (±5σ error bars)
Patch
number of cycles, m

IBM Quantum Computing Center

IBM's 10 Quantum Device Lineup



53 Qubit Rochester Device



Deutsch-Josza Algorithm

Problem: Given an *n*-bit Boolean function (mapping *n* bits to 1) that is known to be either constant or balanced, determine whether it is balanced or constant. A function is "balanced" if an equal number of input values return 0 and 1.

Apply phase shift of π to negate elements where $f(x) = 1$.
Apply Walsh-Hadamard to the result.

For constant f, the final output is $|0\rangle$ with probability 1.
For balanced f, the final output is non-zero with probability 1.

(Details on next slides.)

Requires only a single call to black box U_f, while classical algorithm requires at least $2^{n-1} + 1$ calls.

Background: Hamming Distance

The Hamming distance $\boldsymbol{d_H(x,y)}$ between two bit strings x and y is the number of bits in which the two strings differ.

The Hamming weight $\boldsymbol{d_H(x)}$ of a bit string x is the number of 1 bits.

For two bit strings x and y, the operator $\boldsymbol{x \cdot y}$ gives the number of common 1 bits.

Some interesting notes:

$$x \cdot y = d_H(x \wedge y)$$

$$\sum_{x=0}^{2^n-1} (-1)^{x \cdot x} = 0$$

$$\sum_{x=0}^{2^n-1} (-1)^{x \cdot y} = \begin{cases} 2^n & \text{if } y = 0 \\ 0 & \text{otherwise} \end{cases}$$

More on Walsh-Hadamard

$$W|0\rangle = \frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}|x\rangle$$

$$\begin{aligned}
W|r\rangle &= (H\otimes\cdots\otimes H)(|r_{n-1}\rangle\otimes\cdots\otimes|r_0\rangle) \\
&= \frac{1}{\sqrt{2^n}}(|0\rangle + (-1)^{r_{n-1}}|1\rangle)\otimes\cdots\otimes(|0\rangle + (-1)^{r_0}|1\rangle) \\
&= \frac{1}{\sqrt{2^n}}\sum_{s=0}^{2^n-1}(-1)^{s_{n-1}r_{n-1}}|s_{n-1}\rangle\otimes\cdots\otimes(-1)^{s_0 r_0}|s_0\rangle \\
&= \frac{1}{\sqrt{2^n}}\sum_{s=0}^{2^n-1}(-1)^{s\cdot r}|s\rangle
\end{aligned}$$

Deutsch-Josza Algorithm

Problem: Given an *n*-bit Boolean function (mapping *n* bits to 1) that is known to be either constant or balanced, determine whether it is balanced or constant. A function is "balanced" if an equal number of input values return 0 and 1.

Apply phase shift of π to negate elements where $f(x) = 1$.
Apply Walsh-Hadamard to the result.

For constant f, the final output is $|0\rangle$ with probability 1.
For balanced f, the final output is non-zero with probability 1.

(Details on next slides.)

Quirk Circuit

Requires only a single call to black box U_f, while classical algorithm requires at least $2^{n-1} + 1$ calls.

First, prepare a complete superposition, and then apply the phase shift algorithm to negate the terms corresponding to vectors $|x\rangle$ where $f(x) = 1$.

$$|\psi\rangle = \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} (-1)^{f(i)} |i\rangle$$

Next, apply the Walsh-Hadamard transform to obtain:

$$|\phi\rangle = \frac{1}{N} \sum_{i=0}^{N-1} \left((-1)^{f(i)} \sum_{j=0}^{N-1} (-1)^{i \cdot j} |j\rangle \right)$$

$$|\phi\rangle = \frac{1}{N}\sum_{i=0}^{N-1}\left((-1)^{f(i)}\sum_{j=0}^{N-1}(-1)^{i\cdot j}|j\rangle\right)$$

For constant f, the $(-1)^{f(i)} = (-1)^{f(0)}$ is simply a global phase, and the state $|\phi\rangle$ is $|0\rangle$:

$$\begin{aligned}
|\phi\rangle &= (-1)^{f(0)}\frac{1}{N}\sum_{j}\left(\sum_{i}(-1)^{i\cdot j}\right)|j\rangle \\
&= (-1)^{f(0)}\frac{1}{N}\sum_{i}(-1)^{i\cdot 0}|0\rangle \\
&= (-1)^{f(0)}|0\rangle
\end{aligned}$$

because $\sum_i(-1)^{i\cdot j} = 0$ for $j \neq 0$.

$$|\phi\rangle = \frac{1}{N}\sum_{i=0}^{N-1}\left((-1)^{f(i)}\sum_{j=0}^{N-1}(-1)^{i\cdot j}|j\rangle\right)$$

For balanced f,

$$|\phi\rangle = \frac{1}{N}\sum_{j}\left(\sum_{i\in X_0}(-1)^{i\cdot j} - \sum_{i\notin X_0}(-1)^{i\cdot j}\right)|j\rangle, \text{where} \quad X_0 = \{x|f(x) = 0\}$$

In this case, when $j = 0$, the amplitude is zero.
Therefore, measuring $|\phi\rangle$ in the standard basis will return a non-zero j with probability 1.

Links to Quirk Circuits

- Deutsch
- Selective Phase Change
- Deutsch-Josza

Simon's Algorithm

Problem: Given a 2-to-1 function f, such that $f(x) = f(x \oplus a)$, find the hidden string a.

Create superposition $|x\rangle|f(x)\rangle$

Measure the right part, which projects the left state into $\frac{1}{\sqrt{2}}(|x_0\rangle + |x_0 \oplus a\rangle)$.

Apply Walsh-Hadamard. (Details next slide.)

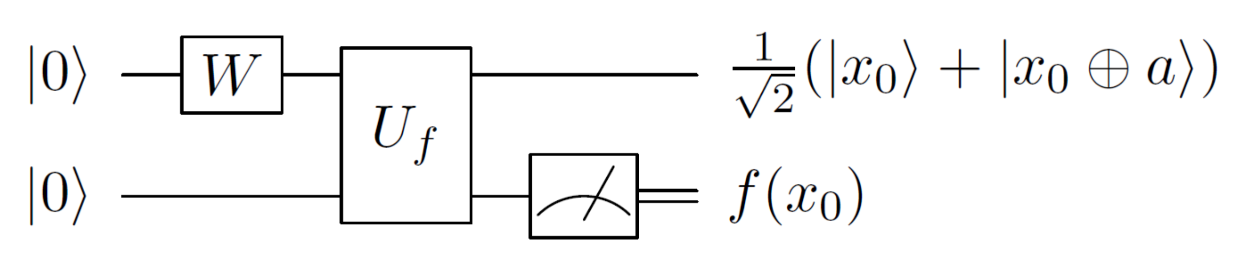
Measurement yields a random y such that $y \cdot a = 0$ (mod 2).

Computation is repeated until n independent equations – about $2n$ times.

Solve for a in $O(n^2)$ steps.

Requires $O(n)$ calls to U_f, followed by $O(n^2)$ steps to solve for a.

Classical approach requires $O(2^{n/2})$ calls to f.



$$
\begin{aligned}
W\left(\frac{1}{\sqrt{2}}(|x_0\rangle + |x_0 \oplus a\rangle)\right) &= \frac{1}{\sqrt{2}}\left(\frac{1}{\sqrt{2^n}}\sum_y ((-1)^{x_0 \cdot y} + (-1)^{(x_0 \oplus a)\cdot y)}\, |y\rangle\right) \\
&= \frac{1}{\sqrt{2^{n+1}}}\sum_y (-1)^{x_0 \cdot y}(1 + (-1)^{a \cdot y})\, |y\rangle \\
&= \frac{2}{\sqrt{2^{n+1}}}\sum_{y \cdot a \ \text{even}} (-1)^{x_0 \cdot y}\, |y\rangle
\end{aligned}
$$

Measurement yields random y such that $y \cdot a = 0 \mod 2$, so the unknown bits of a_i of a must satisfy this equation:

$$y_0 \cdot a_0 \oplus \cdots \oplus y_{n-1} \cdot a_{n-1} = 0$$

Computation is repeated until n linearly independent equations have been found. Each time, the resulting equation has at least a 50% probability of being linearly independent of the previous equations. After repeating $2n$ times, there is a 50% chance that n linearly independent equations have been found. These equations can be solved to find a in $O(n^2)$ steps.

Summary

- Any efficient reversible classical circuit can be efficiently implemented as a quantum circuit.
  - Use inverse function to reduce space and unentangle temporary bits.
- For quantum advantage, add some non-classical operations.
  - E.g, phase change.
- Are these algorithms really useful?
  - Perhaps not directly, but they illustrate ways in which quantum computing may have an advantage over classical computing.