GOOGLE – BUILDING QUANTUM COMPUTERS

a cura della Redazione del Notiziario Tecnico TIM

Quantum Computing merges two great scientific revolutions of the 20th century: computer science and quantum physics. Quantum physics is the theoretical basis of the transistor, the laser, and other technologies which enabled the computing revolution. But on the algorithmic level, today's computing machinery still operates on "classical" Boolean logic. Quantum Computing is the design of hardware and software that replaces Boolean logic by quantum law at the algorithmic level. For certain computations such as optimization, sampling, search or quantum simulation this promises dramatic speedups. We are particularly interested in applying quantum computing to artificial intelligence and machine learning. This is because many tasks in these areas rely on solving hard optimization problems or performing efficient sampling scenarios where optical-radio networks requires automatic real-time joint optimization of heterogeneous computation, communication, and memory/cache resources and high dimensional fast configurations (e.g., selecting and combining optimum network functions and inference techniques). Moreover, the nexus of EI with distributed ledger technologies will enable new collaborative ecosystems which can include, but are not limited to: network operators, platform providers, AI technology/software providers and Users.

What the quantum computing milestones means

On 23 October 2019, Nature [1] published the news that Google's team of researchers have achieved a big breakthrough in quantum computing known as quantum supremacy.

"It's a term of art that means we've used a quantum computer to solve a problem that would take a classical computer an impractically long amount of time.

This moment represents a distinct milestone in our effort to harness the principles of quantum mechanics to solve computational problems", said Sundar Pichai, Google CEO, in his note What our quantum computing milestone means. [2]

"While we're excited for what's ahead, we are also very humbled by the journey it took to get here.

And we're mindful of the wisdom left to us by the great Nobel Laureate Richard Feynman: "If you think you understand quantum mechanics, you don't understand quantum mechanics." [2]

In many ways, the exercise of building a quantum computer is one long lesson in everything we don't yet understand about the world around us.

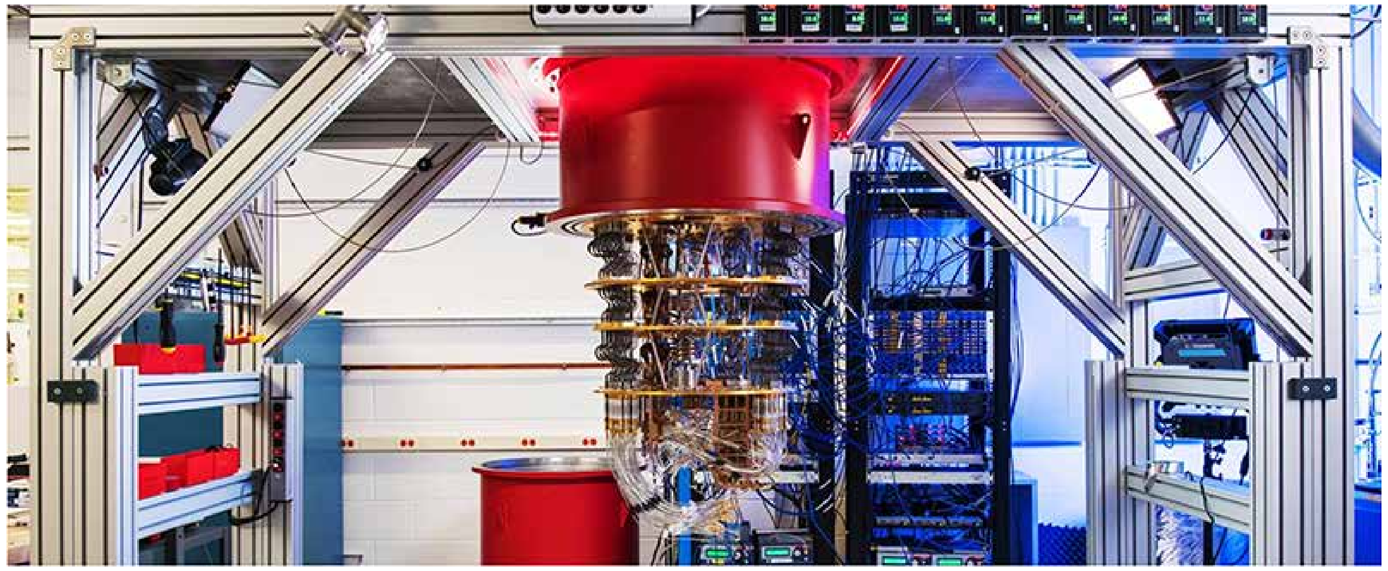
While the universe operates fundamentally at a quantum level, human beings don't experience it that way. In fact, many principles of quantum mechanics directly contradict our surface level observations about nature.

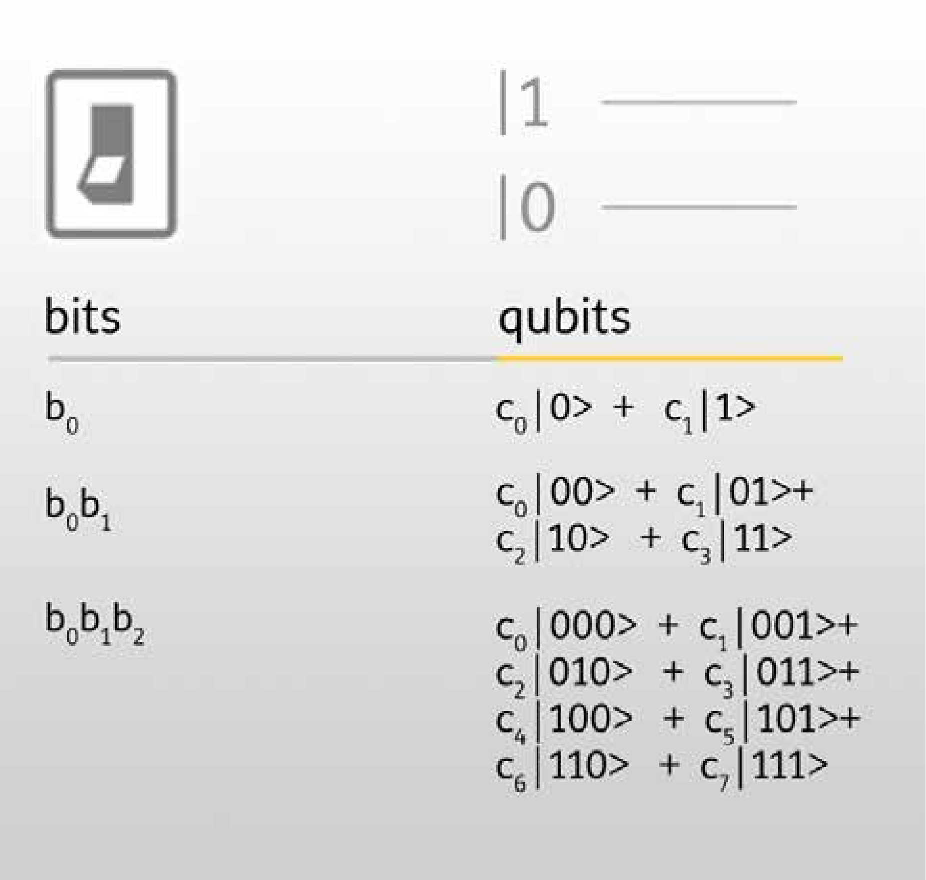
Yet the properties of quantum mechanics hold enormous potential for computing." [2]

What is a quantum computer?

The word quantum computer is a little bit misleading because it sounds like a computer, and when people think of computer, they think of a phone or a laptop.

The truth is the phone and the laptop and even a very powerful supercomputer all operate according to the same fundamental rules, and a quantum computer is fundamentally different [M1],[3].





Quantum hardware can be used as a tool for approaching certain kinds of computational problems. "Our ongoing efforts are both to develop the hardware and to develop algorithms that leverage this hardware", said Marissa Giustina, Research Scientist and Quantum Electronics Engineer at Google. [M1]

To better understand a Quantum Computer, we'll have to introduce for a moment the Quantum Mechanics.

"The most fundamental model of nature we know was developed in the early 20th century and is known as quantum mechanics. The word "mechanics" refers to the mechanisms by which things happen. The word "quantum" refers to discrete quantities of energy or some other physical quantity.

Within quantum mechanics, energy comes in packets, sometimes called photons. And you cannot have fractional packets", Giustina continues.

"The word "quantum" doesn't dictate an object's size. A quantum object is one that relates in a well-defined way to a single quantum of energy. For instance, the photon mentioned before is a quantum object; similarly, atoms are quantum objects.

In a nutshell, a quantum object is one whose observable behavior reflects that nature only offers energy in discrete packets.

What differentiates quantum computing hardware from a regular computer? In essence, quantum hardware lives in a richer world than its conventional counterpart. Let's consider a simple, abstract, quantum object, which is entirely described by the fact that it can be in one of two different energy levels. Let's call those levels 0 (/0>) and 1(/1>).

Because of the apparent similarity between our quantum object and that classical bit of information, we call this quantum analog a quantum bit, or qubit.

One peculiar feature about quantum mechanics is the existence of superpositions. A superposition is like a special mixture of the energy levels 0 and 1, where the weight of each energy level is given by complex constants C_0 and C_1 (C_0/0>) + C_1 (/1>).

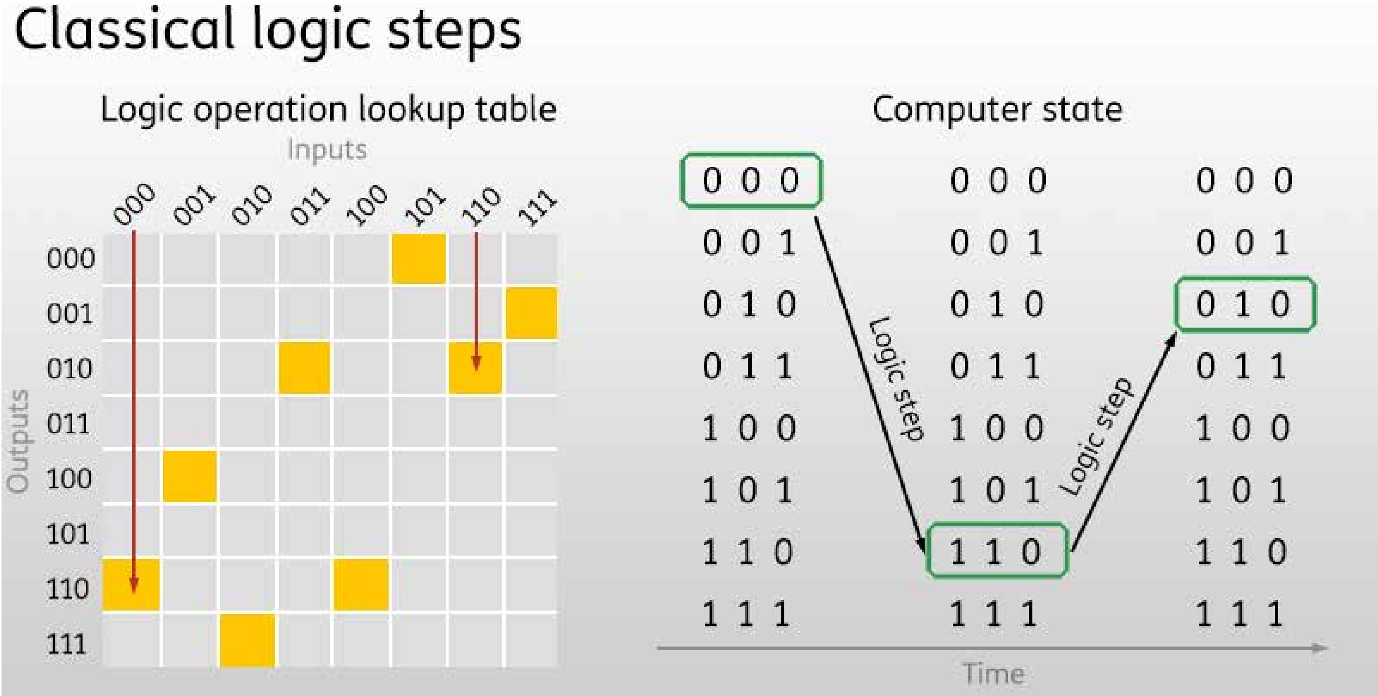
If we measure the energy of our qubit, we will sometimes observe 0, and sometimes 1, where the value of sometimes is given by the constants C. An individual measurement will yield an outcome of 0 or 1. There are no other options.

But before the measurement occurs, we know at most the chances of getting a 0 or a 1.

We can't know the actual outcome for sure until we measure it. Therefore, when we want to talk about the energy state of the qubit before we've made the measurement, we use this superposition to represent that the qubit hasn't decided yet which outcome to display, even though the chances of getting each outcome are fixed.

Suppose we add a second qubit. If these were conventional switches, we could think about each switch independently. But qubits are different. Just as one qubit can be in a superposition state, two qubits can share a superposition state, where, for instance, the measurement outcome is unknown but will certainly be the same for both objects or opposite for both objects.

This means that in order to fully describe two qubits, we need to consider C's for all possible measurement outcomes we could see. To describe three qubits, we need eight C's. Describing four qubits takes 16 C's, and so on. Each time we add another qubit, it takes twice as much information to describe the whole pile of them.” [M1]

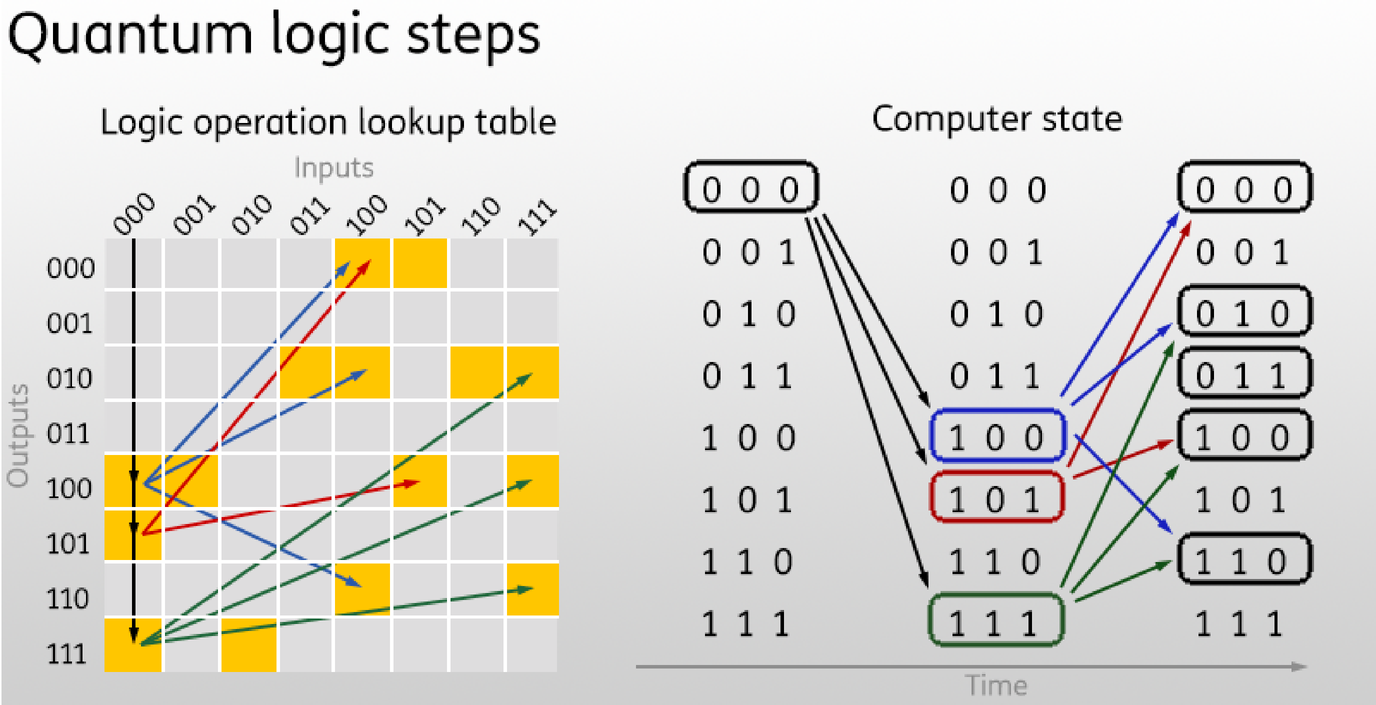


Classical logic steps

Logic operation lookup table

Outputs \ Inputs	000	001	010	011	100	101	110	111
000						■		
001								■
010				■			■	
011								
100		■						
101								
110	■				■			
111			■					

The logic operation shown in the picture takes the state 0-0-0 to 1-1-0. If we were to apply the same operation again, we'd go from 1-1-0 to 0-1-0.



Quantum logic steps

Logic operation lookup table

Outputs \ Inputs	000	001	010	011	100	101	110	111
000					■	■		
001								
010				■	■		■	■
011								
100	■	■				■		■
101	■							
110					■			■
111	■		■					

For example, starting in 0-0-0, we go to a superposition of 1-0-0, 1-0-1, and 1-1-1.Then each of those three parts of the super post state branches out to even more states.

That is the crux of what differentiates quantum hardware. The quantum system lives in a richer space, so that representing n qubits with a classical computer requires 2n bits

A classical computer's state is the value of its memory bits and computer programs determine how the computer goes from one state to the next. But because we're based in classical physics, the state of the computer is just one of these states at each point in time. On each step of a classical algorithm, we go from one state to the next. [M2]

“Compared to classical states, quantum states are more rich. They can have weight in all possible classical states, a situation physicists call superposition.

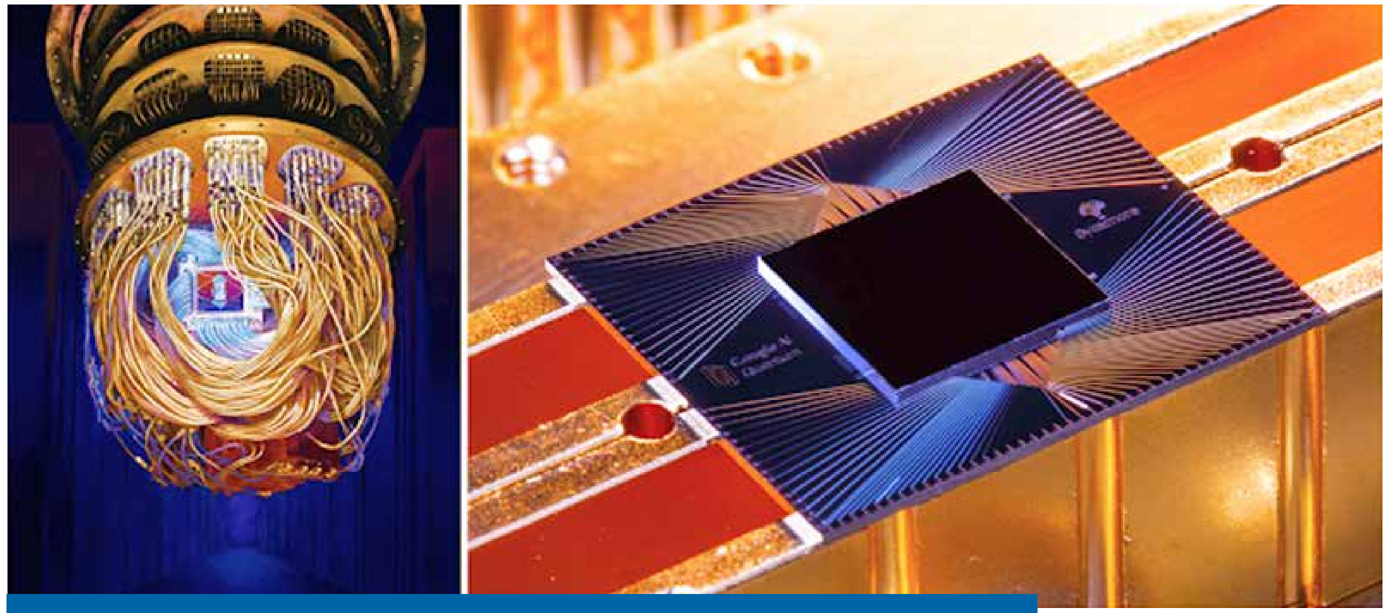
Each step of a quantum algorithm mixes the states into complex superpositions”, explains Daniel Sank, Quantum Electronics Engineer at Google. [M2] The extra complexity of quantum computers allows them to solve some problems faster than a classical computer ever could.

But does this mean that a quantum memory with 100 qubits corresponds to a conventional memory with 2100 bits? Not so fast...

Quantum hardware is very effective at encoding and processing certain kinds of information. But it cannot efficiently mimic many useful aspects of its classical counterpart. [M1]

The exponentially growing complexity of quantum systems also gives a clue about where quantum hardware could be useful.

In the fields of chemistry and materials development, simulation of molecules could be a powerful technique to learn about the properties of a new molecule before fully synthesizing it in the lab.



Left: Artist's rendition of the Sycamore processor mounted in the cryostat. (Forest Stearns, Google AI Quantum Artist in Residence) Right: Photograph of the Sycamore processor. (Erik Lucero, Research Scientist and Lead Production Quantum Hardware)

However, our ability to simulate chemistry on computers is limited. At its heart, chemistry is an application of quantum mechanics. In fact, chemistry and materials simulations have appeared as an appealing near-term problem to approach using quantum hardware. [M1]

Demonstrating quantum supremacy

“Physicists have been talking about the power of quantum computing for over 30 years, but the questions have always been: will it ever do something useful and is it worth investing in?” said John Martinis, Chief Scientist Quantum Hardware and Sergio Boixo, Chief Scientist Quantum Computing Theory, Google AI Quantum.

“For such large-scale endeavours it is good engineering practice to formulate decisive short-term goals that demonstrate whether the designs are going in the right direction.

So, we devised an experiment as an important milestone to help answer these questions.

This experiment, referred to as a quantum supremacy experiment [4], provided direction for our team to overcome the many technical challenges inherent in quantum systems engineering to make a computer that is both programmable and powerful.

To test the total system performance we selected a sensitive computational benchmark that fails if just a single component of the computer is not good enough.

We developed a new 54-qubit processor, named “Sycamore”, that is comprised of fast, high-fidelity quantum logic gates, in order to perform the benchmark testing.

Our machine performed the target computation in 200 seconds, and from measurements in our experiment we determined that it would take the world’s fastest supercomputer 10,000 years to produce a similar output." [4]

The Experiment

To actually demonstrate quantum supremacy, we have these three steps: first, pick a circuit, second, run it on the quantum computer, third, simulate what the quantum computer is doing on a classical computer and we gradually increase the complexity of that circuit.At some point, it becomes completely impossible for the classical computer to keep up.

Then we say we have achieved quantum supremacy. [M3]

To get a sense of how this benchmark works, imagine enthusiastic quantum computing neophytes visiting our lab in order to run a quantum algorithm on our new processor.

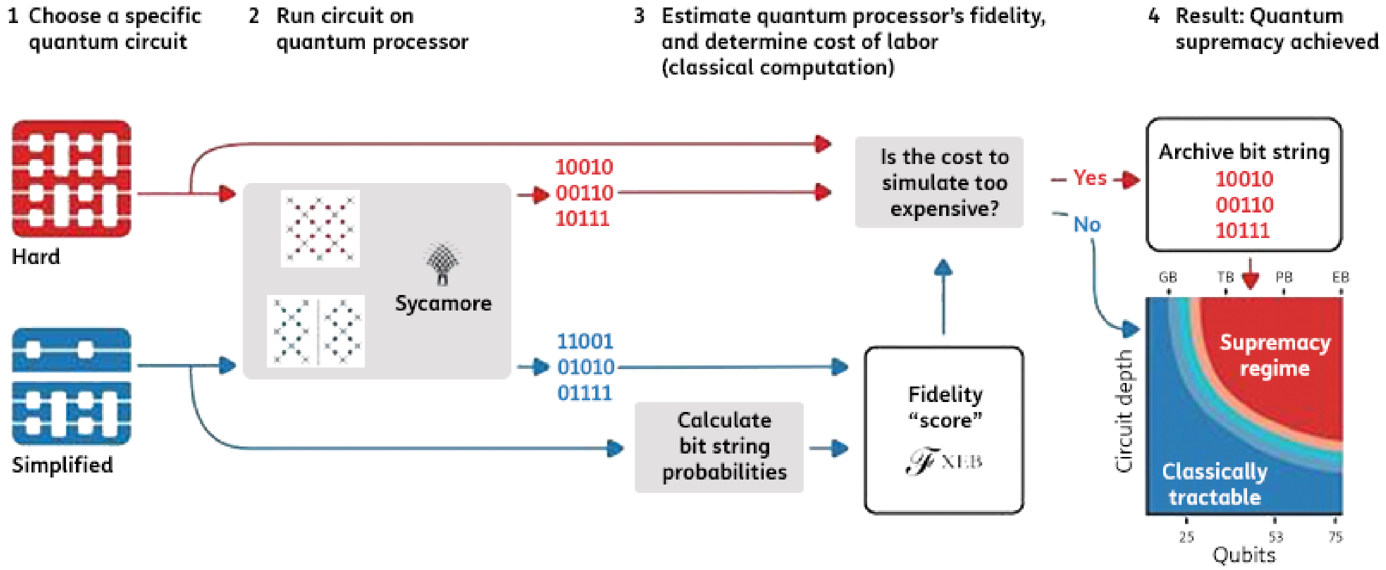
They can compose algorithms from a small dictionary of elementary gate operations. Since each gate has a probability of error, our guests would want to limit themselves to a modest sequence with about a thousand total gates.

Assuming these programmers have no prior experience, they might create what essentially looks like a random sequence of gates, which one could think of as the “hello world” program for a quantum computer. Because there is no structure in random circuits that classical algorithms can exploit, emulating such quantum circuits typically takes an enormous amount of classical supercomputer effort. [4]

Each run of a random quantum circuit on a quantum computer produces a bitstring, for example 0000101.

Owing to quantum interference, some bitstrings are much more likely to occur than others when we repeat the experiment many times. However, finding the most likely bitstrings for a random quantum circuit on a classical computer becomes exponentially more difficult as the number of qubits (width) and number of gate cycles (depth) grow. [4]

Process for demonstrating quantum supremacy.



In the experiment, we first ran random simplified circuits from 12 up to 53 qubits, keeping the circuit depth constant.

We checked the performance of the quantum computer using classical simulations and compared with a theoretical model.

Once we verified that the system was working, we ran random hard circuits with 53 qubits and increasing depth, until reaching the point where classical simulation became infeasible. [4]

This result is the first experimental challenge against the extended Church-Turing thesis [6], which states that classical computers can efficiently implement any "reasonable" model of computation.
With the first quantum computation that cannot reasonably be emulated on a classical computer, we have opened up a new realm of computing to be explored. [4]

The sycamore processor

"The quantum supremacy experiment was run on a fully programmable 54-qubit processor named "Sycamore." It's comprised of a two-dimensional grid where each qubit is connected to four other qubits.

As a consequence, the chip has enough connectivity that the qubit states quickly interact throughout the entire processor, making the overall state impossible to emulate efficiently with a classical computer.

The success of the quantum supremacy experiment was due to our improved two-qubit gates with enhanced parallelism that reliably achieve record performance, even when operating many gates simultaneously.

We achieved this performance using a new type of control knob that is able to turn off interactions between neighboring qubits. This greatly reduces the errors in such a multi-connected qubit system.

We made further performance gains by optimizing the chip design to lower crosstalk, and by developing new control calibrations that avoid qubit defects.

We designed the circuit in a two-dimensional square grid, with each qubit connected to four other qubits. This architecture is also forward compatible for the implementation of quantum error-correction. We see our 54-qubit Sycamore processor as the first in a series of ever more powerful quantum processors."

"To ensure the future utility of quantum computers, we also needed to verify that there are no fundamental roadblocks coming from quantum mechanics." John Martinis, Chief Scientist Quantum Hardware and Sergio Boixo, Chief Scientist Quantum Computing Theory, Google AI Quantum continue.

"Physics has a long history of testing the limits of theory through experiments, since new phenomena often emerge when one starts to explore new regimes characterized by very different physical parameters. Prior experiments showed that quantum mechanics works as expected up to a state-space dimension of about 1000.

Here, we expanded this test to a size of 10 quadrillion and find that everything still works as expected. We also tested fundamental quantum theory by measuring the errors of two-qubit gates and finding that this accurately predicts the benchmarking results of the full quantum supremacy circuits. This shows that there is no unexpected physics that might degrade the performance of our quantum computer.

Our experiment therefore provides evidence that more complex quantum computers should work according to theory, and makes us feel confident in continuing our efforts to scale up." [4]

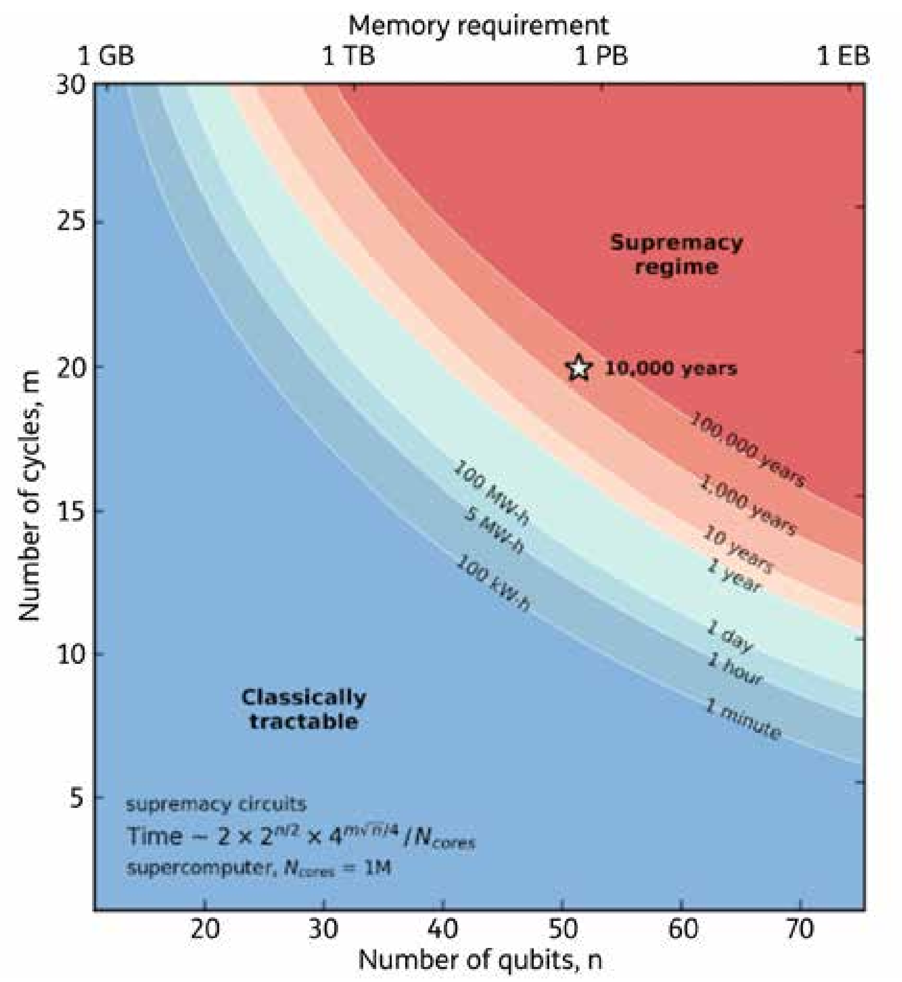
What's next?

"Our team has two main objectives going forward, both towards finding valuable applications in quantum computing", said John Martinis, Chief Scientist Quantum Hardware and Sergio Boixo, Chief Scientist Quantum Computing Theory, Google AI Quantum.
"First, in the future we will make our supremacy-class processors available to collaborators and academic researchers, as well as companies that are interested in developing algorithms and searching for applications for today's NISQ processors.

Creative researchers are the most important resource for innovation — now that we have a new computational resource, we hope more researchers will enter the field motivated by trying to invent something useful.

Second, we're investing in our team and technology to build a fault-tolerant quantum computer as quickly as possible. Such a device promises a number of valuable applications.

For example, we can envision quantum computing helping to design new materials — lightweight bat-



Schrödinger-Feynman algorithm

Estimate of the equivalent classical computation time assuming 1M CPU cores for quantum supremacy circuits as a function of the number of qubits and number of cycles for the Schrödinger-Feynman algorithm. The star shows the estimated computation time for the largest experimental circuits.

teries for cars and airplanes, new catalysts that can produce fertilizer more efficiently (a process that today produces over 2% of the world's carbon emissions), and more effective medicines. Achieving the necessary computational capabilities will still require years of hard engineering and scientific work. But we see a path clearly now, and we're eager to move ahead." [4] ■

Reference

1. Arute, F., Arya, K., Babbush, R. et al. Quantum supremacy using a programmable superconducting processor. Nature 574, 505–510 (2019). https://doi.org/10.1038/s41586-019-1666-5
2. Sundar Pichai, "What our quantum computing milestone means", https://www.blog.google/perspectives/sundar-pichai/what-our-quantum-computing-milestone-means/
3. Hartmut Neven, "Computing takes a quantum leap forward", https://www.blog.google/technology/ai/computing-takes-quantum-leap-forward/
4. John Martinis, Sergio Boixo, Quantum Supremacy Using a Programmable Superconducting Processor
5. https://research.google/teams/applied-science/quantum/
6. https://en.wikipedia.org/wiki/Church%E2%80%93Turing_thesis#Variations

Media

1. [M3] Demonstrating Quantum Supremacy https://www.youtube.com/watch?v=-ZNEzzDcllU
2. [M2] Building a quantum computer with superconducting qubits https://www.youtube.com/watch?v=uPw9nkJAwDY
3. [M1] What is a quantum computer? https://www.youtube.com/watch?v=k-21vRCC0RM
4. [M4] Quantum supremacy explained https://www.youtube.com/watch?v=gylmjTOUfCQ
5. [M5] Programming a quantum computer with Cirq https://www.youtube.com/watch?v=16ZfkPRVf2w