INHA UNIVERSITY TASHKENT
DEPARTMENT OF CSE & ICE

SPRING SEMESTER 2019

SOC2060 - COMPUTER ARCHITECTURE

TERM PAPER REPORT

GOOGLE QUANTUM COMPUTING

Submitted by

Jamshid Khakimjonov (Leader)	U1710260
Sevara Amirullaeva	U1710142
Ziyoda Saydakhmatova	U1710259
Vladislav Podkovyrkin	U1710275
Sarvinoz Maksudova	U1710276

Group: 001 **SOPHOMORE**

GOOGLE QUANTUM COMPUTING

CONTENTS

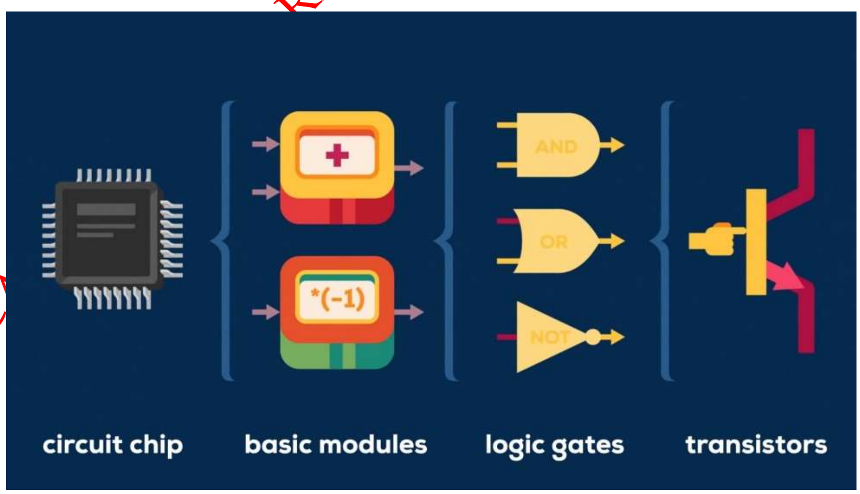
TITLE OF THE PROJECT:
GOOGLE QUANTUM COMPUTING

1. INTRODUCTION TO QUANTUM COMPUTING

1.1 Powerful new possibilities

Quantum computing sounds like something out of a sci-fi movie. But it's real, and scientists and engineers are working to make it a practical reality. Google engineers are creating chips the size of a quarter that could revolutionize the computers of tomorrow. But what is quantum computing, exactly?

Can you get more and more out of less and less? The smaller computers get, the more powerful they seem to become: there's more number-crunching ability in a 21st-century cell phone than you'd have found in a room-sized, military computer 50 years ago. Yet, despite such amazing advances, there are still plenty of complex problems that are beyond the reach of even the world's most powerful computers—and there's no guarantee we'll ever be able to tackle them. One problem is that the basic switching and memory units of computers, known as transistors, are now approaching the point where they'll soon be as small as individual atoms. If we want computers that are smaller and more powerful than today's, we'll soon need to do our computing in a radically different way. Entering the realm of atoms opens up powerful new possibilities in the shape of quantum computing, with processors that could work millions of times faster than the ones we use today.



Sounds amazing, but the trouble is that quantum computing is hugely more complex than traditional computing and operates in the Alice in Wonderland world of quantum physics, where the "classical," sensible, everyday laws of physics no longer apply. What is quantum computing and how does it work?

1.2 - Quantum theory

It's the branch of physics that deals with the world of atoms and the smaller (subatomic) particles inside them. You might think atoms behave the same way as everything else in the world, in their own tiny little way—but that's not true: on the atomic scale, the rules change and the "classical" laws of physics we take for granted in our everyday world no longer automatically apply. As **Richard P. Feynman**, one of the greatest physicists of the 20th century, once put it:

> *"Things on a very small scale behave like nothing you have any direct experience about... or like anything that you have ever seen."*
>
> **Richard Feynman**

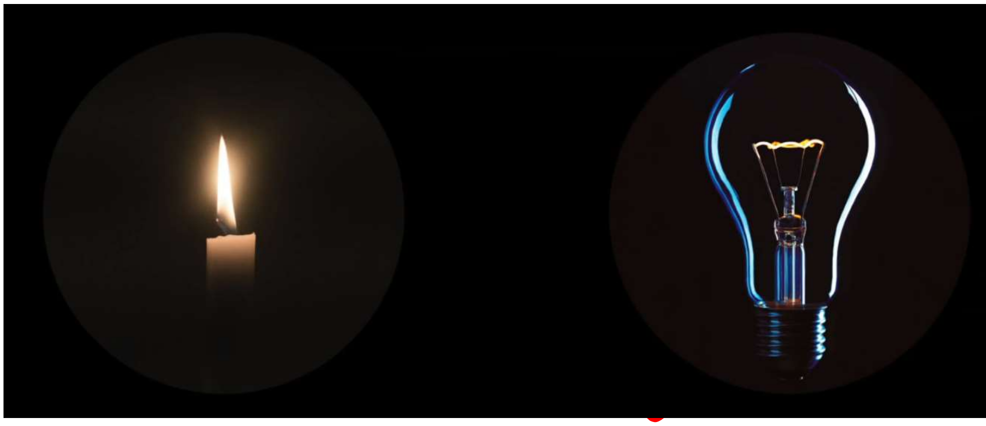
If you've studied light, you may already know a bit about quantum theory. You might know that a beam of light sometimes behaves as though it's made up of particles (like a steady stream of cannonballs), and sometimes as though it's waves of energy rippling through space (a bit like waves on the sea). That's called wave-particle duality and it's one of the ideas that comes to us from quantum theory. It's hard to grasp that something can be two things at once—a **particle** and a **wave**—because it's totally alien to our everyday experience: a car is not simultaneously a bicycle and a bus. In quantum theory, however, that's just the kind of crazy thing that can happen. The most striking example of this is the baffling riddle known as Schrödinger's cat. Briefly, in the weird world of quantum theory, we can imagine a situation where something like a cat could be alive and dead at the same time!
What does all this have to do with computers? Suppose we keep on pushing Moore's Law—keep on making transistors smaller until they get to the point where they obey not the ordinary laws of physics (like old-style transistors) but the more bizarre laws of quantum mechanics. The question is whether computers designed this way can do things our conventional computers can't. If we can predict mathematically that they might be able to, can we actually make them work like that in practice?

1.3- Scientists

People have been asking those questions for several decades. Among the first were IBM research physicists **Rolf Landauer** and **Charles H. Bennett**. Landauer opened the door for quantum computing in the **1960s** when he proposed that information is a physical entity that could be manipulated according to the laws of physics. One important consequence of this is that computers waste energy manipulating the bits inside them (which is partly why computers use so much energy and get so hot, even though they appear to be doing not very much at all). In the **1970s**, building on Landauer's work, Bennett showed how a computer could circumvent this problem by working in a "reversible" way, implying that a quantum computer could carry out massively complex computations without using **massive amounts of energy**. In **1981**, physicist Paul Benioff from Argonne National Laboratory tried to envisage a basic machine that would work in a similar way to an ordinary computer but according to the principles of quantum physics. The following year, Richard Feynman sketched out roughly how a machine using quantum principles could carry out basic computations. A few years later, Oxford University's David Deutsch (one of the leading lights in quantum computing) outlined the theoretical basis of a quantum computer in more detail. How did these great scientists imagine that quantum computers might work? [1]

2. WHAT IS QUANTUM COMPUTER?

2.1 What is a quantum computer? Well, quantum physics describes the behavior of atoms and fundamental particles, like electrons and photons. So quantum computer operates by controlling the behavior of these particles, but in a way that is completely difference from our regular computers. So quantum computer is not just a more powerful version of our current computers, just like a light bulb is not a more powerful candle.



You cannot build a light bulb by building better and better candles. A light bulb is a different technology, based on deeper scientific understanding. Similarly, a quantum computer is new kind of device, based on the science of quantum physics, and just like a light bulb transformed society, quantum computers have the potential to impact so many aspects of our lives, including our security needs, our health care and even the internet. So companies all around the world are working to build these devices, and Google company is one of them.

The key features of an ordinary computer—bits, registers, logic gates, algorithms, and so on—have analogous features in a quantum computer. Instead of bits, a quantum computer has quantum bits or qubits, which work in a particularly intriguing way. Where a bit can store either a zero or a 1, a qubit can store a zero, a one, both zero and one, or an infinite number of values in between—and be in multiple states (store multiple values) at the same time! If that sounds confusing, think back to light being a particle and a wave at the same time, Schrödinger's cat being alive and dead, or a car being a bicycle and a bus. A gentler way to think of the numbers qubits store is through the physics concept of superposition (where two waves add to make a third one that contains both of the originals). If you blow on something like a flute, the pipe fills up with a standing wave: a wave made up of a fundamental frequency (the basic note you're playing) and lots of overtones or harmonics (higher-frequency multiples of the fundamental). The wave inside the pipe contains all these waves simultaneously: they're added together to make a combined wave that includes them all. Qubits use superposition to represent multiple states (multiple numeric values) simultaneously in a similar way.

Just as a quantum computer can store multiple numbers at once, so it can process them simultaneously. Instead of working in serial (doing a series of things one at a time in a sequence), it can work in parallel (doing multiple things at the same time). Only when you try to find out what state it's actually in at any given moment (by measuring it, in other words) does it "collapse" into one of its possible states—and that gives you the answer to your problem. Estimates suggest a quantum computer's ability to work in parallel would make it millions of times faster than any conventional computer... if only we could build it! So how would we do that?

2.2 What would a quantum computer be like in reality?

In reality, qubits would have to be stored by **atoms**, ions (atoms with too many or too few electrons), or even smaller things such as electrons and photons (energy packets), so a quantum computer would be almost like a table-top version of the kind of particle physics experiments they do at Fermi-lab or CERN. Now you wouldn't be racing particles round giant loops and smashing them together, but you would need mechanisms for containing atoms, ions, or subatomic particles, for putting them into certain states (so you can store information), knocking them into other states (so you can make them process information), and figuring out what their states are after particular operations have been performed.

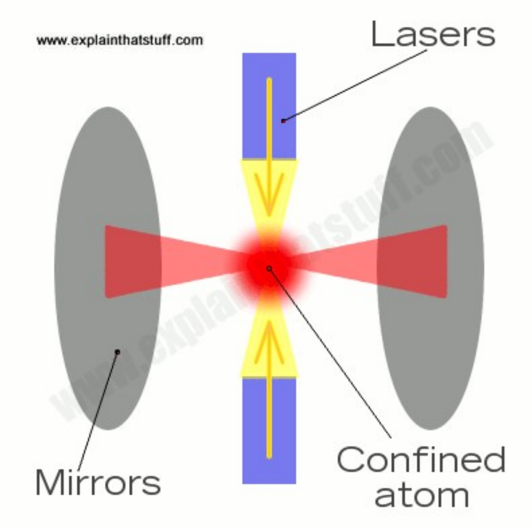


Photo: *A single atom or ion can be trapped in an optical cavity—the space between mirrors—and controlled by precise pulses from laser beams.*

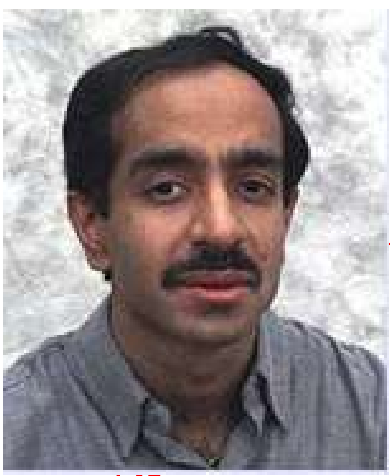
In practice, there are lots of possible ways of containing atoms and changing their states using laser beams, electromagnetic fields, radio waves, and an assortment of other techniques. One method is to make qubits using quantum dots, which are nanoscopically tiny particles of semiconductors inside which individual charge carriers, electrons and holes (missing electrons), can be controlled. Another method makes qubits from what are called ion traps: you add or take away electrons from an atom to make an ion, hold it steady in a kind of laser spotlight (so it's locked in place like a nanoscopic rabbit dancing in a very bright headlight), and then flip it into different states with laser pulses. In another technique, the qubits are photons inside optical cavities (spaces between extremely tiny mirrors). Don't worry if you don't understand; not many people do. Since the entire field of quantum computing is still largely abstract and theoretical, the only thing we really need to know is that qubits are stored by atoms or other quantum-scale particles that can exist in different states and be switched between them.

2.3 - What can quantum computers do that ordinary computers can't?

Although people often assume that quantum computers must automatically be better than conventional ones, that's by no means certain. So far, just about the only thing we know for certain that a quantum computer could do better than a normal one is **factorization**: finding two unknown prime numbers that, when multiplied together, give a third, known number. In 1994, while working at Bell Laboratories, mathematician Peter Shor demonstrated an algorithm that a quantum computer could follow to find the "prime factors" of a large number, which would speed up the problem enormously.

Peter Shor (1994)

Shor's algorithm really excited interest in quantum computing because virtually every modern computer (and every secure, online shopping and banking website) uses public-key **encryption** technology based on the virtual impossibility of finding prime factors quickly (it is, in other words, essentially an "intractable" computer problem). If quantum computers could indeed factor large numbers quickly, today's online security could be rendered obsolete at a stroke. But what goes around comes around, and some researchers believe quantum technology will lead to much stronger forms of encryption. (In 2017, Chinese researchers demonstrated for the first time how quantum encryption could be used to make a very secure video call from Beijing to Vienna.)



Lov Grover (1996)

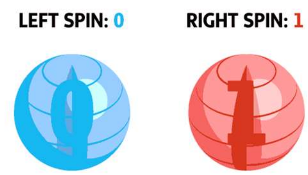
Does that mean quantum computers are better than conventional ones? Not exactly. Apart from Shor's algorithm, and a search method called **Grover's algorithm**, hardly any other algorithms have been discovered that would be better performed by quantum methods. Given enough time and computing power, conventional computers should still be able to solve any problem that quantum computers could solve, eventually. In other words, it remains to be proven that quantum computers are generally superior to conventional ones, especially given the difficulties of actually building them. Who knows how conventional computers might advance in the next 50 years, potentially making the idea of quantum computers irrelevant—and even absurd.

2.4 - Why is it so hard to make a quantum computer?

We have decades of experience building ordinary, transistor-based computers with conventional architectures; building quantum machines means reinventing the whole idea of a computer from the bottom up. First, there are the practical difficulties of making qubits, controlling them very precisely, and having enough of them to do really useful things. Next, there's a major difficulty with errors inherent in a quantum system—"**noise**" as this is technically called—which seriously compromises any calculations a quantum computer might make. There are ways around this ("quantum error correction"), but they introduce a great deal more complexity. There's also the fundamental issue of how you get data in and out of a quantum computer, which is, itself, a complex computing problem. Some critics believe these issues are insurmountable; others acknowledge the problems but argue the mission is too important to abandon.

2.5 - How far off are quantum computers?

Three decades after they were first proposed, quantum computers remain largely theoretical. Even so, there's been some encouraging progress toward realizing a quantum machine. There were two impressive breakthroughs in **2000**. First, Isaac Chuang (now an MIT professor, but then working at IBM's Almaden Research Center) used five fluorine atoms to make a crude, five-qubit quantum computer. The same year, researchers at Los Alamos National Laboratory figured out how to make a seven-qubit machine using a drop of liquid. Five years later, researchers at the University of Innsbruck added an extra qubit and produced the first quantum computer that could manipulate a qubyte (eight qubits).



These were tentative but important first steps. Over the next few years, researchers announced more ambitious experiments, adding progressively greater numbers of qubits. By 2011, a pioneering Canadian company called D-Wave Systems announced in Nature that it had produced a 128-qubit machine; the announcement proved highly controversial and there was a lot of debate over whether the company's machines had really demonstrated quantum behavior. Three years later, **Google** announced that it was hiring a team of academics (including University of California at Santa Barbara physicist John Martinis) to develop its own quantum computers based on D-Wave's approach. In March 2015, the **Google team** announced they were "a step closer to quantum computation," having developed a new way for qubits to detect and protect against errors. In **2016**, MIT's Isaac Chuang and scientists from the University of Innsbruck unveiled a five-qubit, ion-trap quantum computer that could calculate the factors of 15; one day, a scaled-up version of this machine might evolve into the long-promised, fully fledged encryption buster.

There's no doubt that these are hugely important advances. and the signs are growing steadily more encouraging that quantum technology will eventually deliver a computing revolution. In December 2017, **Microsoft** unveiled a complete quantum development kit, including a new computer language, Q#, developed specifically for quantum applications. In early 2018, **D-wave** announced plans to start rolling out quantum power to a cloud computing platform. A few weeks later, Google announced Bristlecone, a quantum processor based on a 72-qubit array, that might, one day, form the cornerstone of a quantum computer that could tackle real-world problems. All very exciting! Even so, it's early days for the whole field, and most researchers agree that we're unlikely to see practical quantum computers appearing for some years—and more likely several decades. The conclusion reached by an influential National Academies of Sciences, Medicine and Engineering report in December 2018 was that "it is still too early to be able to predict the time horizon for a practical quantum computer" and that "many technical challenges remain to be resolved before we reach this milestone."

2.6 - Key ingredients of quantum mechanics

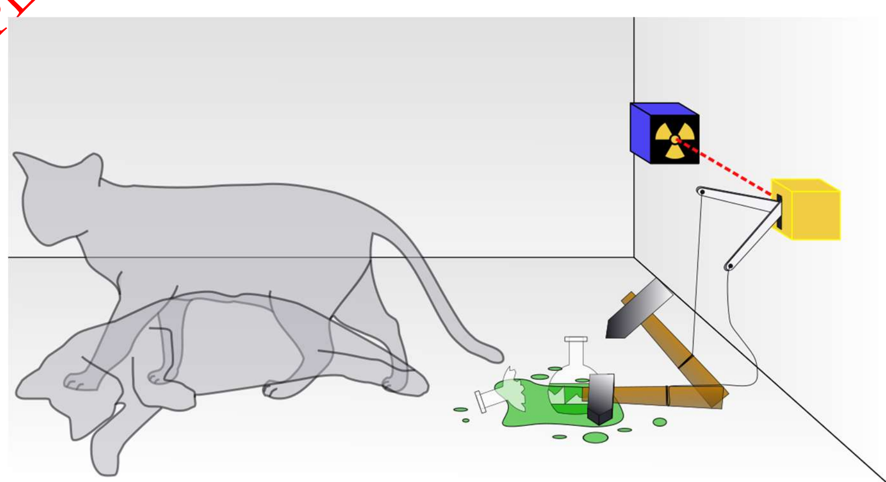
Quantum mechanics has certain bizarre features which do not occur in standard, or **"classical"** physics, such as:

1. **Superposition.** If a system can be in state A or state B, it can also be in a "mixture" of the two states. If we measure it, we see either A or B, probabilistically.

2. **Collapse.** Any further measurements will give the same result.

3. **Entanglement.** There exist systems of multiple parts which cannot be described only in terms of their constituent parts.

4. **Uncertainty.** There are pairs of measurements where greater certainty of the outcome of one measurement implies greater uncertainty of the outcome of the other measurement.

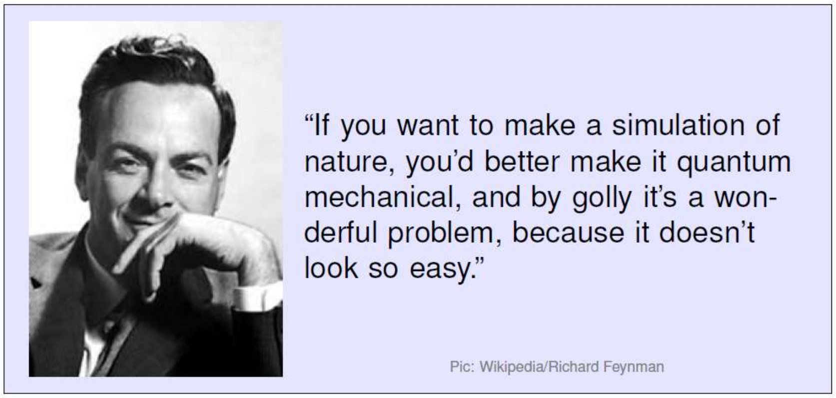


Example, Schrödinger's cat is a thought experiment, sometimes described as a paradox, devised by Austrian physicist Erwin Schrödinger in 1935. It illustrates what he saw as the problem of the Copenhagen interpretation of quantum mechanics applied to everyday objects. The scenario presents a hypothetical cat that may be simultaneously both alive and dead, a state known as a quantum superposition, as a result of being linked to a random subatomic event that may or may not occur.

3 - Historical Bibliography of Quantum Computing

3.1 Models of Computation

Richard Feynman was the first to suggest, in a talk in 1981, that quantum-mechanical systems might be more powerful than classical computers. Feynman asked what kind of computer could simulate physics and then argued that only a quantum computer could simulate quantum physics efficiently. He focused on quantum physics rather than classical physics because, as he colorfully put it:

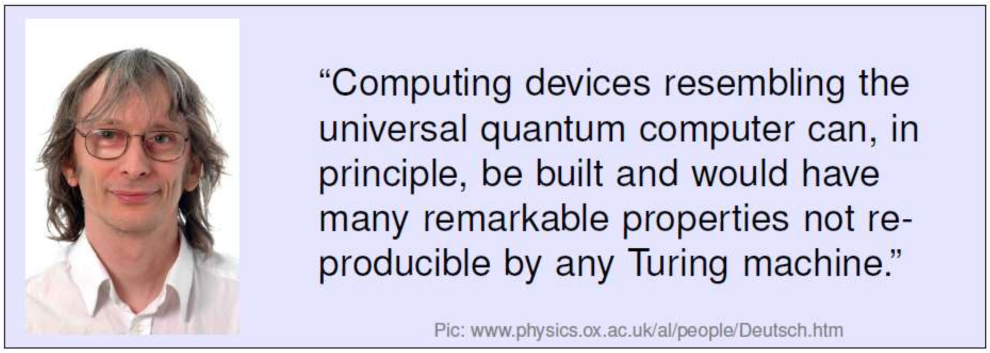


Around the same time, in "Quantum mechanical models of Turing machines that dissipate no energy" [3] and related articles, Paul Benioff demonstrated that quantum-mechanical systems could model Turing machines. In other words, he proved that quantum computation is at least as powerful as classical computation.

But is quantum computation more powerful than classical computation?

David Deutsch explored this question and more in his 1985 paper "Quantum theory, the Church–Turing principle and the universal quantum computer" [14]. First, he introduced quantum counterparts to both the Turing machine and the universal Turing machine. He then demonstrated that the universal quantum computer can do things that the universal Turing machine cannot, including generate genuinely random numbers, perform some parallel calculations in a single register, and perfectly simulate physical systems with finite dimensional state spaces.

In 1989, in “Quantum computational networks” [15], Deutsch described a second model for quantum computation: quantum circuits. He demonstrated that quantum gates can be combined to achieve quantum computation in the same way that Boolean gates can be combined to achieve classical computation. He then showed that quantum circuits can compute anything that the universal quantum computer can compute, and vice versa.

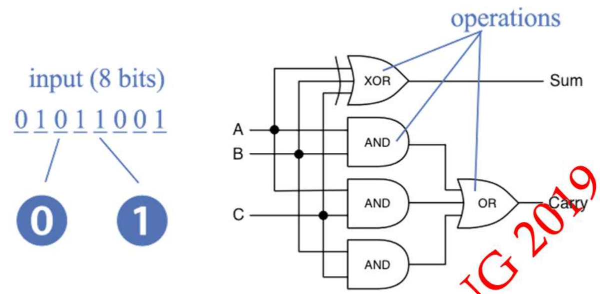


Andrew Chi-Chih Yao picked up where Deutsch left off and addressed the complexity of quantum computation in his 1993 paper “Quantum circuit complexity” [35]. Specifically, he showed that any function that can be computed in polynomial time by a quantum Turing machine can also be computed by a quantum circuit of polynomial size. This finding allowed researchers to focus on quantum circuits, which are easier than quantum Turing machines to design and analyze.

Also in 1993, Ethan Bernstein and Umesh Vazirani presented “Quantum complexity theory” [6], in which they described a universal quantum Turing machine that can efficiently simulate any quantum Turing machine. (As with so many quantum articles, the final version of the paper did not appear until several years later, in the SIAM Journal of Computing [7].) As its title suggests, Bernstein and Vazirani’s paper kick-started the study of quantum complexity theory.

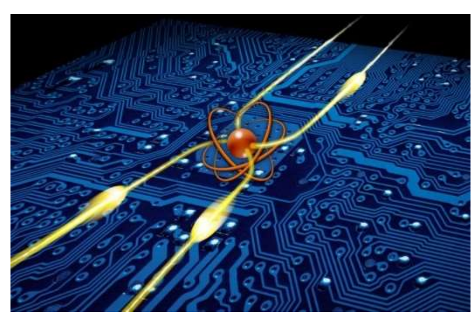
3.2 Quantum Gates

In 1995, a cluster of articles examined which sets of quantum gates are adequate for quantum computation—that is, which sets of gates are sufficient for creating any given quantum circuit. Of these papers, the one that was cited the most in later works was "Elementary gates for quantum computation" [1], in which Adriano Barenco et al. showed that any quantum circuit can be constructed using nothing more than quantum gates on one qubit and controlled exclusive-OR gates on two qubits. Though that paper was arguably the most influential, other articles were important as well, including "Two-bit gates are universal for quantum computation" [17], in which David DiVincenzo proved that two-qubit quantum gates are adequate; "Conditional quantum dynamics and logic gates" [2], in which Adriano Barenco, David Deutsch, and Artur Ekert showed that quantum controlled-NOT gates and one-qubit gates are together adequate; and "Almost any quantum logic gate is universal" [22], in which Seth Lloyd showed that almost any quantum gate with two or more inputs is universal (i.e., by itself adequate).



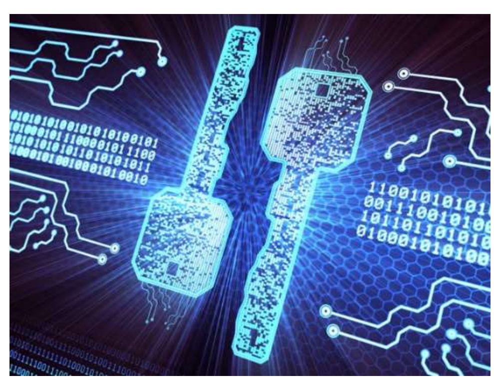
3.3 Quantum Information

Secure channels of communication are of course crucial, but security is not the only consideration in the transfer of information. Accordingly, quantum cryptography is just one of several topics in the burgeoning field of quantum information. Other topics include quantum error correction, fault-tolerant quantum computation, quantum data compression, and quantum teleportation. Information needs to be protected not just from eavesdroppers but also from errors caused by channel noise, implementation flaws, and, in the quantum case, decoherence. Peter W. Shor, a trailblazer not just of quantum algorithms but also of quantum error correction and fault-tolerant quantum computation, was the first to describe a quantum error-correcting method. In his 1995 article "Scheme for reducing decoherence in quantum computer memory" [27], he demonstrated that encoding each qubit of information into nine qubits could provide some protection against decoherence. At almost the same time but without



knowledge of Shor's article, Andrew M. Steane wrote "Error correcting codes in quantum theory" [32], which achieved similar results. Very shortly thereafter, Shor and A.R. Calderbank presented improved results in "Good quantum error-correcting codes exist" [11]. In the late 1990s, when research on quantum error correction and fault-tolerant 7 quantum computation ballooned, Shor, Steane, and Calderbank remained among the major contributors. Error is not the only thing information theorists strive to reduce; they also seek to reduce the space required to represent information. The landmark paper on the classical representation and compression of data was "A mathematical theory of communication" by Claude E. Shannon [25], the "father" of information theory. In this **1948** paper, Shannon showed that it is possible, up to a certain limit, to compress data without loss of information; beyond that limit, some information is necessarily lost. (Seminal in so many ways, this paper also laid the groundwork for classical error-correcting codes.) Almost 50 years later, Benjamin Schumacher developed a quantum version of Shannon's theorem. Schumacher first described his finding in an article called "Quantum coding," which he submitted to Physical Review A in 1993 but which was not published until 1995 [24]. In the (unfortunate but not uncommon) lag between submission and publication, he and Richard Jozsa published "A new proof of the quantum noiseless coding theorem" [21], which offered a simpler proof than the original article. Not everything in quantum information theory has a precedent in classical information theory. In 1993, Charles H. Bennett et al. dazzled the scientific community and delighted science fiction fans by showing that quantum teleportation is theoretically possible. In "Teleporting an unknown quantum state via dual classical and Einstein–Podolsky–Rosen Channels" [5], they described how an unknown quantum state could be disassembled and then reconstructed perfectly in another location. The first researchers to verify this method of teleportation experimentally were Dik Bouwmeester et al., who reported their achievement in 1997 in "Experimental quantum teleportation" [8].

3.4 Quantum Cryptography



As mentioned before, Shor's factorization algorithm has yet to be implemented on more than a few qubits. But if the efficient factorization of large numbers becomes possible, RSA cryptography will need to be replaced by a new form of cryptography, one that will not be foiled by classical or quantum computers. Conveniently, such a method already exists; in fact, it was developed before Shor invented his factorization algorithm. Coincidentally, it too relies on quantum mechanics.

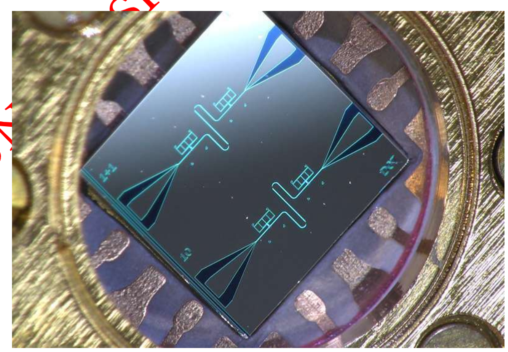
The cryptographic method in question is quantum key distribution, which was introduced in 1984 by Charles H. Bennett and Gilles Brassard in “Quantum cryptography: Public key distribution and coin tossing” [4] and is thus commonly referred to as BB84. In short, quantum key distribution is secure not because messages are encrypted in some difficultto-decrypt
way but rather because eavesdroppers cannot intercept messages undetected, regardless of computational resources.
Although quantum key distribution is the most famous cryptographic application of quantum mechanics, it is not the only one, and it was not the first. In the 1960s, Stephen Wiesner conceived of two applications: a way to send two messages, only one of which can be read, and a way to design money that cannot be counterfeited. His ideas were largely unknown until 1983, when he described them in an article called “Conjugate coding” [34].

4 - GOOGLE RESEARCH AREAS

Google Research Areas:

- Superconducting qubit processors
- Qubit metrology
- Quantum simulation
- Quantum assisted optimization
- Quantum neural networks

4.1 - Superconducting qubit processors



Superconductivity is the ability of certain metals to allow electricity to pass through them without any resistance at very low temperatures.

Superconductivity is the ability of certain metals to allow electricity to pass through them without any resistance at very low temperatures.

One area of research the **Google AI Quantum team** follows is building quantum processors from **superconducting** electrical circuits, which are tempting tools for implementing **quantum bits.** Even if superconducting circuits have shown extensibility to **modest processor sizes**, the next challenge is stabilizing their performance, which can fluctuate unpredictably. The origin of performance fluctuations is not well understood and impede the progress in stabilizing processor performance even though it has been observed in many superconducting qubit processors. [1]

Therefore, in spite of recent advances, today's qubits are still unstable, and hardware needs to be sturdy to run them. Quantum computers need to keep their processors extremely cold and protect them from external shocks. Even accidental sounds can cause the computer to make mistakes.

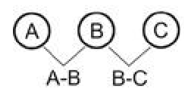
Finally, to operate better, quantum processors need to have an error rate of less than 0.5 percent for every two qubits. Google's best has been 0.6 percent using its much smaller 9-qubit hardware. But recently Bristlecone - newest 72-qubit quantum processor which has an error rate of less than 0.5 percent was launched. (Jamshid will tell about it) [2] [3]

Manufacturer	Name/Code name/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

4.2 - Qubit metrology

Today the biggest obstacle in building a fault-tolerant quantum computer is the need for high-fidelity qubit operations in a scalable architecture. Since only quantum error correction can overcome fundamental fragility of quantum information the challenge is to decrease the percentage of error. In a fault-tolerant quantum computer the qubits and their logic interactions must have errors below a threshold: scaling up with more and more qubits then brings the net error probability down to appropriate levels.

The key to quantum error correction is measuring qubit parities, which detects bit flips and phase flips in pairs of qubits. When parity changes, one of the two qubits had an error, but which one is not known. To identify, encoding must use larger numbers of qubits.



input bits			parity	
A	B	C	A-B	B-C
0	0	0	0	0
1	0	0	1	0
0	1	0	1	1
0	0	1	0	1
1	1	0	0	1
1	0	1	1	1
0	1	1	1	0
1	1	1	0	0

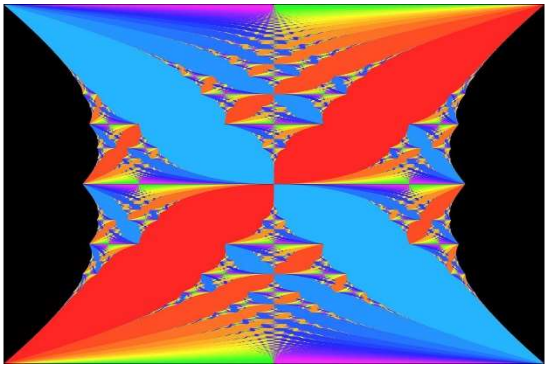
Reducing error requires solving problems in physics, control, materials and fabrication, which differ for every implementation. [4] Now Google is working on a quantum supremacy experiment, to approximately sample a quantum circuit beyond the capabilities of state-of-the-art classical computers and algorithms. [3]

4.3- Quantum simulation

What is a quantum simulator? One possible definition is that of an experimental system that reproduces the physics of a precisely defined Hamiltonian*. However, the quantum simulator might provide an analogue that allows us to experimentally address intriguing questions that are not directly tractable in the laboratory. A second class of simulators deals with objectives that are not accessible via classical computation and may be employed to better understand and enable the design of exotic materials.

*Hamiltonian is an operator corresponding to the sum of the kinetic energies plus the potential energies for all the particles in the system

Quantum simulation of the electronic structure problem is one of the most researched applications of quantum computing. Google especially focus on quantum algorithms for modelling systems of interacting electrons with applications in chemistry and materials science. [5][3]



It's called the Hofstadter butterfly (1976) and it's actually a map of how electrons behave in a strong magnetic field. Every split and shift of these subatomic particles are rendered by the photons inside of Google's quantum chip.

This was made possible by quantum simulators, which are special-purpose quantum computers. Even if quantum simulators can't solve any problem like theoretical quantum computers could, they can be used to solve specific problems. [6]

4.4 Quantum assisted optimization

Optimization deals with finding the best solution to a problem (according to some criteria) from a set of possible solutions.[7]

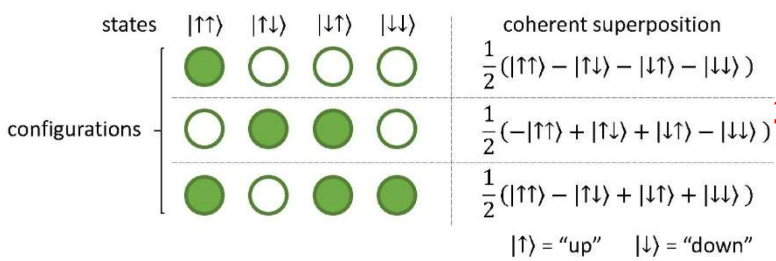
Now Google is developing hybrid quantum-classical solvers for approximate optimization. Because discrete optimizations in aerospace, automotive, and other industries may benefit from hybrid quantum-classical optimization. [3]

Traditional quantum algorithms, such as Shor's factorization algorithm and Grover's search algorithm, can be statically compiled with a high level of optimization using known input parameters. With hybrid algorithms, some of a quantum program's input parameters can change each iteration. For example, a compiler may spend hours optimizing for quantum instructions that include quantum rotations for specific input angles to solve a chemistry problem, but now we find that the angles change every iteration. This suggests that we need a partial compilation strategy in which programs are optimized for unchanging parameters, but then quickly re-optimized each iteration for parameters that change. [8]

4.5 Quantum neural networks

In "Classification with Quantum Neural Networks on Near Term Processors", Google constructs a model of quantum neural networks that is specifically designed to work on quantum processors that are expected to be available in the near term. [3]

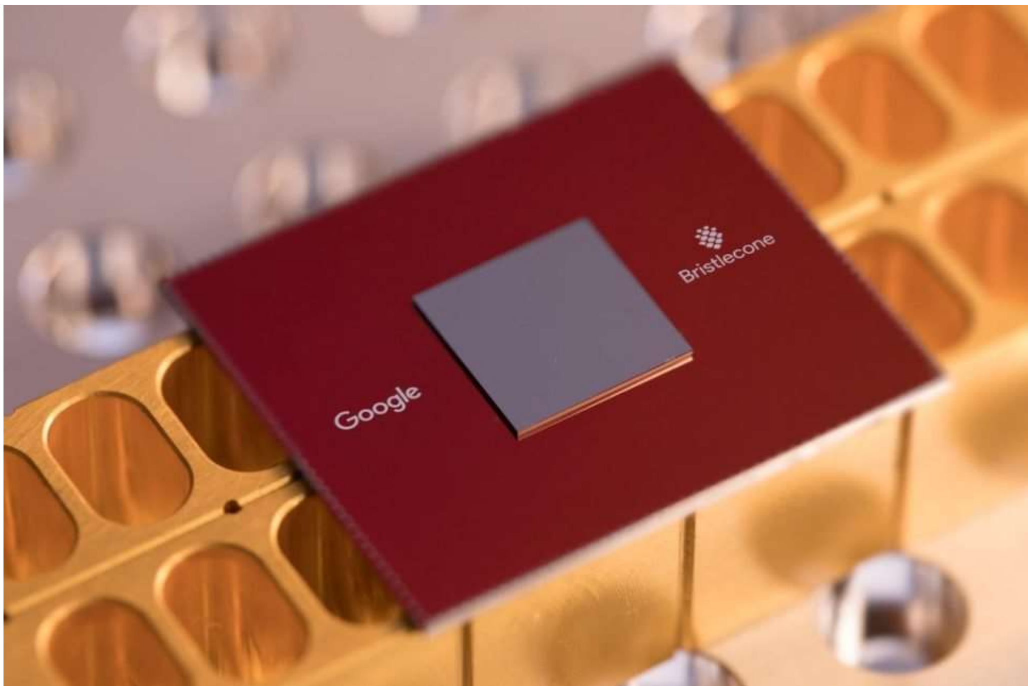
Using quantum coherence, where a system can concurrently exist in a combination of states—in what we call a coherent superposition. The trick is then to associate each neuron with a state of the system: if the neuron is “on” then its corresponding state appears with a positive sign in the superposition, while if the neuron is “off” then there is a negative sign in the superposition. We have focused on systems of multiple quantum bits (qubits), each of which can either be “up” or “down”. By looking at all of the combinations of “up” and “down” possible in our collection of qubits, you can see that an exponential number of neuron configurations can be stored within small number of qubits. For example, the diagram below shows that we can store any configuration of 4 neurons in only 2 qubits! [9]



Since the technological implementation of a quantum computer is still in a premature stage, such quantum neural network models are mostly theoretical proposals that await their full implementation in physical experiments.

5. Bristlecone Google's 72-Qubit Quantum Processor

Tech giant Google introduced to the world its newest quantum processor: the Bristlecone chip. The company is reportedly optimistic that their processor would pave the way for quantum computing to go mainstream.



Google is all set to dominate the race for quantum supremacy with its Bristlecone chip.

The processor was presented by the **Google Quantum AI Lab** at the annual American Physical Society meeting in Los Angeles, California. In a blog post written by Julian Kelly, a research scientist at the Quantum AI Lab, he said:

"Today we presented Bristlecone, our new quantum processor, at the annual American Physical Society meeting in Los Angeles. The purpose of this gate-based superconducting system is to provide a testbed for research into system error rates and scalability of our qubit technology, as well as applications in quantum simulation, optimization, and machine learning."

The Bristlecone chip reportedly features a 72-qubit gate-based superconducting system. Before its arrival, the most powerful **quantum computing chip** was the 50-qubit processor developed by IBM. It is now clear that the crown has been snatched away from the latter.

Kelly explained how the Google Quantum AI Lab team developed the Bristlecone chip. The processor was reportedly created by scaling the company's former 9-qubit system which has a 1 percent low error rate for readout. The new processor uses the same ***"scheme for coupling, control, and readout, but is scaled to a square array of 72 qubits."***

"Our strategy is to explore near-term applications using systems that are forward compatible to a large-scale universal error-corrected quantum computer," Kelly further stated.

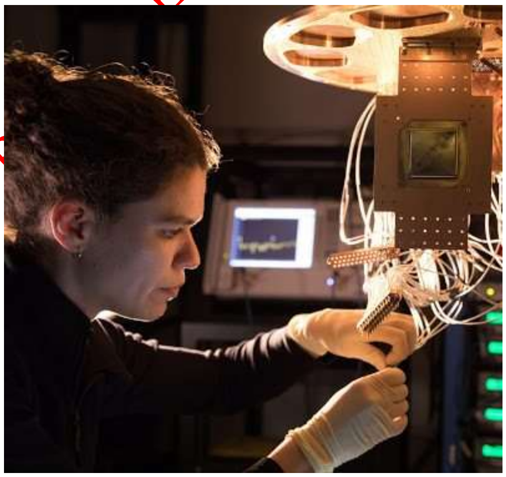
The Bristlecone chip has outdone its 9-qubit chip predecessor when it comes to quantum error correction, a significant factor that keeps hypersensitive **qubits** from getting corrupted.

"We are cautiously optimistic that quantum supremacy can be achieved with Bristlecone." ~ Julian Kelly

According to Kelly, Bristlecone was primarily developed as a testbed for research works involving **quantum system error rates**, Google qubit technology's scalability, quantum simulation, optimization, and machine learning.
With the Bristlecone chip's 72-qubit size, the team aims to demonstrate quantum supremacy in the future, use surface code to investigate first and second order error-correction, and be able to facilitate **quantum algorithm** development on actual hardware.

However, before investigating specific applications, Kelly pointed out that it is vital to quantify a processor's capabilities first. For this, Google's Quantum AI Lab theory team developed a benchmarking tool capable of doing such complicated task.

"We can assign a single system error by applying random quantum circuits to the device and checking the sampled output distribution against a classical simulation," Kelly went on to say.



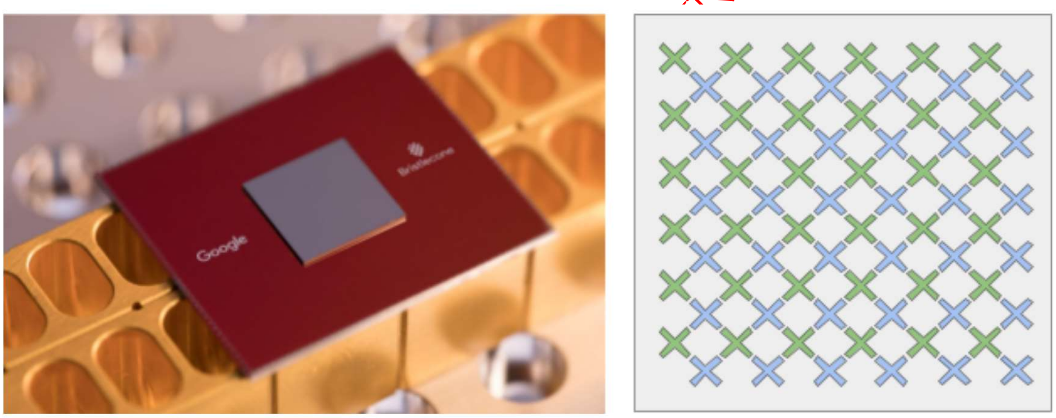
A Bristlecone chip being installed by Research Scientist Marissa Giustina at the Quantum AI Lab in Santa Barbara | Google

While the size of the Bristlecone chip may somehow help mitigate **qubit error rates**, Google said that quantum computing is not just about qubits.

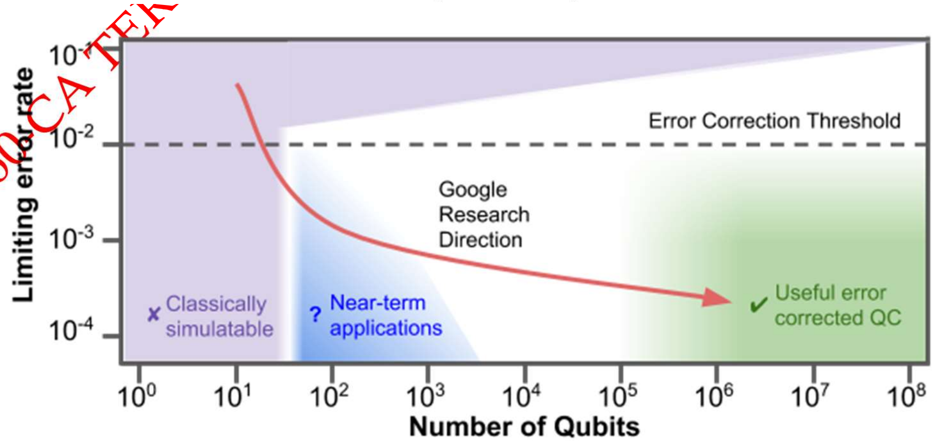
Apparently, operating devices like the Bristlecone at low system error also requires a balance between a full stack of technology from software and control electronics to the processor itself. This needs careful system engineering over several iterations, Kelly said.

Aside from Google, other companies like IBM and Microsoft have been vocal about their quantum computing efforts. For instance, IBM has reportedly been working on a general-purpose quantum computer for utilization by businesses in the future.

With Google's latest quantum processor, the race for quantum supremacy has become tighter than ever before.



Bristlecone is Google's newest quantum processor (left). On the right is a cartoon of the device: each "X" represents a qubit, with nearest neighbor connectivity.



2D conceptual chart showing the relationship between **error rate** and **number** of qubits. The intended research direction of the **Quantum AI Lab** is shown in red, where we hope to access near-term applications on the road to building an error corrected quantum computer.

We have finally reached the last question, what does a quantum computer look like? Let's take a look at the actual hardware they are building at Google.

Their qubits are resonant electrical circuits, made of patterned aluminum on a silicon chip that slosh electrical current back and forth at two different energy levels to encode the quantum 0 to 1 states.

Here is an example of one of quantum chips. Each chip features 72 qubits. As you can see, it about a size of a quarter. They want each qubit to behave as one single quantum object, with two levels. Any other particle interacting qubit from it's environment pulls away from this two-level idea.

So creating a clean qubit environment is a critical challenge at the same time, we want to be able to control the qubits efficiently, adding and removing quanta of energy and letting pairs of qubits interact to exchange the energy with each other on demand. These requirements seem to oppose each other. Ideal qubits should be perfectly clean to interact with nothing. But when in specific cases, we want them to interact very strongly. This gives one insight into tensions and challenges of building good quantum hardware. A first step toward building clean qubits is to build the qubit circuits out of superconducting materials, which experiences ne electrical loss. Superconductors perform only at very low temperatures. And they operate qubits in a cryostat at less then 50 millikelvin, just a fraction of degree above absolute zero. The cold temperatures and vacuum inside a cryostat also contribute to keeping the qubit environment clean. The cryostat consist of a series of nested plates and cans. The warmest stage is at top, and it get's colder, as you go down. All the equipment is the central core of the cryostat is responsible for getting things cold. Their hardware is installed around the edges and on the bottom coldest plate.

Each qubit chip must be mounted in a package, which holds the chip at millikelvin temperatures and bridges the gap between cables and a small chip.

To address the packed chip, electronics outside the cryostat send signals. Each carry electrical signals from room temperature all the way down to the coldest stage, while leaking only the small amount of heat. A large heat load would prevent the cryostat from reaching its millikelvin base temperature. A collection of filters and amplifiers outfits each cable for its specific task. The electronics outside cryostat are controlled by code running on a computer. They generate precisely calibrated electrical signals, shaped pulses of microwave radiation, which are send to control and read the qubits. This entire system – from chip to cryostat, cables to code – is all necessary to run our quantum hardware.

6. QUANTUM MACHINE LEARNING

Google is at the forefront of innovation in **Machine Intelligence**, with active research exploring virtually all aspects of machine learning, including deep learning and more classical algorithms. Exploring theory as well as application, much of our work on language, speech, translation, visual processing, ranking and prediction relies on Machine Intelligence. In all of those tasks and many others, we gather large volumes of direct or indirect evidence of relationships of interest, applying learning algorithms to understand and generalize.

Machine Intelligence at Google raises deep scientific and engineering challenges, allowing us to contribute to the broader academic research community through technical talks and publications in major conferences and journals. Contrary to much of current theory and practice, the statistics of the data we observe shifts rapidly, the features of interest change as well, and the volume of data often requires enormous computation capacity. When learning systems are placed at the core of interactive services in a fast changing and sometimes adversarial environment, combinations of techniques including deep learning and statistical models need to be combined with ideas from control and game theory.

Supervised learning on molecules has incredible potential to be useful in chemistry, drug discovery, and materials science. Luckily, several promising and closely related neural network models invariant to molecular symmetries have already been described in the literature. These models learn a message passing algorithm and aggregation function to compute a function of their entire input graph. At this point, the next step is to find a particularly effective variant of this general approach and apply it to chemical prediction benchmarks until we either solve them or reach the limits of the approach. In this paper, we reformulate existing models into a single common framework we call Message Passing **Neural Networks** (MPNNs) and explore additional novel variations within this framework. Using MPNNs we demonstrate state of the art results on an important molecular property prediction benchmark, results we believe are strong enough to justify retiring this benchmark.

6.1 What is quantum machine learning?

Quantum machine learning is an emerging interdisciplinary research area at the intersection of quantum physics and machine learning. The most common use of the term refers to machine learning algorithms for the analysis of classical data executed on a quantum computer. While machine learning algorithms are used to compute immense quantities of data, quantum machine learning increases such capabilities intelligently, by creating opportunities to conduct analysis on quantum states and systems. This includes hybrid methods that involve both classical and quantum processing, where computationally difficult subroutines are outsourced to a quantum device. These routines can be more complex in nature and executed faster with the assistance of quantum devices. Furthermore, quantum algorithms can be used to analyze quantum states instead of classical data. Beyond quantum computing, the term "quantum machine learning" is often associated with machine learning methods applied to data generated from quantum experiments, such as learning quantum phase transitions or creating new quantum experiments. Quantum machine learning also extends to a branch of research that explores methodological and structural similarities between certain physical systems and learning systems, in particular neural networks. For example, some mathematical and numerical techniques from quantum physics are applicable to classical deep learning and vice versa. Finally, researchers investigate more abstract notions of learning theory with respect to quantum information, sometimes referred to as "quantum learning theory".

6.2 - Machine learning with quantum computers

Quantum-enhanced machine learning refers to quantum algorithms that solve tasks in machine learning, thereby improving and often expediting classical machine learning techniques. Such algorithms typically require one to encode the given classical data set into a quantum computer to make it accessible for quantum information processing. Subsequently, quantum information processing routines are applied and the result of the quantum computation is read out by measuring the quantum system. For example, the outcome of the measurement of a qubit reveals the result of a binary classification task. While many proposals of quantum machine learning algorithms are still purely theoretical and require a full-scale universal quantum computer to be tested, others have been implemented on small-scale or special purpose quantum devices.

6.3 - Linear algebra simulation with quantum amplitudes

A number of quantum algorithms for machine learning are based on the idea of *amplitude encoding*, that is, to associate the amplitudes of a quantum state with the inputs and outputs of computations. Since a state of qubits is described by complex amplitudes, this information encoding can allow for an exponentially compact representation. Intuitively, this corresponds to associating a discrete probability distribution over binary random variables with a classical vector. The goal of algorithms based on amplitude encoding is to formulate quantum algorithms whose resources grow polynomially in the number of qubits , which amounts to a logarithmic growth in the number of amplitudes and thereby the dimension of the input.

Many quantum machine learning algorithms in this category are based on variations of the **quantum algorithm for linear systems of equations** (colloquially called HHL, after the paper's authors) which, under specific conditions, performs a matrix inversion using an amount of physical resources growing only logarithmically in the dimensions of the matrix. One of these conditions is that a Hamiltonian which entry wise corresponds to the matrix can be simulated efficiently, which is known to be possible if the matrix is sparse or low rank. For reference, any known classical algorithm for **matrix inversion** requires a number of operations that grows **at least quadratically in the dimension of the matrix.**

Quantum matrix inversion can be applied to machine learning methods in which the training reduces to solving a **linear system of equations**, for example in least-squares linear regression, the least-squares version of **support vector machines**, and Gaussian processes.

A crucial bottleneck of methods that simulate linear algebra computations with the amplitudes of quantum states is state preparation, which often requires one to initialize a quantum system in a state whose amplitudes reflect the features of the entire dataset. Although efficient methods for state preparation are known for specific cases, this step easily hides the complexity of the task.

7. CIRQ and OPENFERMION by Google

7.1 - CIRQ – Googles' initiative

Google is taking initiative to join the "quantum race" and released a new software.

The software developed by Google is called Cirq, it was recently released this summer. It's an open source platform to develop Noisy Intermediate-Scale Quantum (NISQ) algorithms. As coined by John Preskill, the NISQ era describes our current quantum computers that require error-correction and only have a small number of qubits. Cirq allows access for anyone to develop and conduct their own research into near-term applications for quantum computers.

Before we reach fault-tolerant quantum computers with millions of coherent qubits that can run Shor's algorithm and DESTROY ALL PREMISES OF SECURITY AND BLOW THE WORLD UP IN FLAMES, (just kidding, luckily there is post-quantum cryptography) we need to work with noisy models. This means error-correction for qubits is a MUST or else many of our computations will output weird and strange answers under the affects of decoherence.

So it makes sense to do a small demo using Cirq on a error-correcting code. Specifically encoding the bit-flip code onto 3 qubits. This can be simulated on IBM's Q experience but in this example I will use Cirq to re-create the same circuit. In the process you can learn the basic syntax for this specific software and make your own projects!

Unlike classical computers, quantum information cannot be copied and thus repetition methods of error-correction will not work on quantum states.

Instead information stored in a qubit is distributed among other qubits in an entangled state. This protects the superposition state and helps discover errors as qubits decohere over time.

7.2 - OpenFermion - Google's Chemistry Package

"The underlying physical laws necessary for the mathematical theory of a large part of physics and the whole of chemistry are thus completely known, and the difficulty is only that the exact application of these laws leads to equations much too complicated to be soluble."

-Paul Dirac, Quantum Mechanics of Many-Electron Systems (1929)

In this passage, physicist Paul Dirac laments that while quantum mechanics accurately models all of chemistry, exactly the associated equations appears intractably complicated. Not until 1982 would Richard Feynman suggest that instead of surrendering to the complexity of quantum mechanics, we might harness it as a computational resource. Hence, the original motivation for quantum computing: by operating a computer according to the laws of quantum mechanics, one could efficiently unravel exact simulations of nature. Such simulations could lead to breakthroughs in areas such as photovoltaics, batteries, new materials, pharmaceuticals and superconductivity. And while we do not yet have a quantum computer large enough to solve classically intractable problems in these areas, rapid progress is being made. Last year, Google published this paper detailing the first quantum computation of a molecule using a superconducting qubit quantum computer. Building on that work, the quantum computing group at IBM scaled the experiment to larger molecules, which made the cover of Nature last month.

Today, we announce the release of OpenFermion, the first open source platform for translating problems in chemistry and materials science into quantum circuits that can be executed on existing platforms. OpenFermion is a library for simulating the systems of interacting electrons (fermions) which give rise to the properties of matter. Prior to OpenFermion, quantum algorithm developers would need to learn a significant amount of chemistry and write a large amount of code hacking apart other codes to put together even the most basic quantum simulations. While the project began at Google, collaborators at ETH Zurich, Lawrence Berkeley National Labs, University of Michigan, Harvard University, Oxford University, Dartmouth College, Rigetti Computing and NASA all contributed to alpha releases. You can learn more details about this release in our paper, OpenFermion: The Electronic Structure Package for Quantum Computers.

One way to think of OpenFermion is as a tool for generating and compiling physics equations which describe chemical and material systems into representations which can be interpreted by a quantum computer1. The most effective quantum algorithms for these problems build upon and extend the power of classical quantum chemistry packages used and developed by research chemists across government, industry and academia. Accordingly, we are also releasing OpenFermion-Psi4 and OpenFermion-PySCF which are plugins for using OpenFermion in conjunction with the classical electronic structure packages Psi4 and PySCF.

While quantum simulation is widely recognized as one of the most important applications of quantum computing in the near term, very few quantum computer scientists know quantum chemistry and even fewer chemists know quantum computing. Our hope is that OpenFermion will help to close the gap between these communities and bring the power of quantum computing to chemists and material scientists.

The core OpenFermion library is designed in a quantum programming framework agnostic way to ensure compatibility with various platforms being developed by the community. This allows OpenFermion to support external packages which compile quantum assembly language specifications for diverse hardware platforms. We hope this decision will help establish OpenFermion as a community standard for putting quantum chemistry on quantum computers. To see how OpenFermion is used with diverse quantum programming frameworks, take a look at OpenFermion-ProjectQ and Forest-OpenFermion - plugins which link OpenFermion to the externally developed circuit simulation and compilation platforms known as ProjectQ and Forest.

CONCLUSION:

Google discovered a lot of things in quantum computing sphere and the goals of Google's Quantum computing LAB are possible to achieve in terms of 21st century. Even their newest superconducting quantum processor with 72 qubits impressed public a lot.

But for many years, quantum computers were not much more than an idea. Today, companies, governments and intelligence agencies are investing in the development of quantum technology. Robert König, professor for the theory of complex quantum systems at the TUM, in collaboration with David Gosset from the Institute for Quantum Computing at the University of Waterloo and Sergey Bravyi from IBM, has now placed a cornerstone in this promising field.

As a conclusion we want to mention that quantum computers in theory can do certain types of problems effectively impossible to normal computers, but are not good at some tasks normal computers do well. E.g. a quantum computer should in theory be good at finding first prime number bigger than ten billion, but take ages to add three ten digit numbers. In practice the most powerful computers today probably use a mix of classical, quantum and pseudo-quantum processing. I think a handful of quantum bits have enough theoretical processing power to solve some problems that would take a room full of regular computers to attempt.

REFERENCES :

REFERENCES FOR "HISTORY" SECTION:

[1] A. Barenco, C.H. Bennett, R. Cleve, D.P. DiVincenzo, N. Margolus, P.W. Shor, T. Sleator, J.A. Smolin, and H. Weinfurter. Elementary gates for quantum computation. Physical Review A, 52(5):3457–3467, 1995.

[2] A. Barenco, D. Deutsch, A. Ekert, and R. Jozsa. Conditional quantum dynamics and logic gates. Physical Review Letters, 74(20):4083–4086, 1995.

[3] P. Benioff. Quantum mechanical models of Turing machines that dissipate no energy. Physical Review Letters, 48(23):1581–1585, 1982.

[4] C.H. Bennett and G. Brassard. Quantum cryptography: Public key distribution and coin tossing. In Proceedings of IEEE International Conference on Computers, Systems, and Signal Processing, pages 175–179, Bangalore, India, December 1984.

[5] C.H. Bennett, G. Brassard, C. Cr´epeau, R. Jozsa, A. Peres, and W.K. Wootters. Teleporting an unknown quantum state via dual classical and Einstein–Podolsky–Rosen channels. Physical Review Letters, 70(13):1895–1899, March 1993.

[6] E. Bernstein and U. Vazirani. Quantum complexity theory. In STOC '93: Proceedings of the Twenty-Fifth Annual ACM Symposium on Theory of Computing, pages 11–20, 1993.

[7] E. Bernstein and U. Vazirani. Quantum complexity theory. SIAM Journal on Computing, 26(5):1411–1473, 1997.

[8] D. Bouwmeester, J.-W. Pan, K. Mattle, M. Eibl, H. Weinfurter, and A. Zeilinger. Experimental quantum teleportation. Nature, 390:575–579, 1997.

[9] G. Brassard. Cryptology column – quantum cryptography: A bibliography. SIGACT News, 24(3):16–20, 1993.

[10] G. Brassard and C. Cr´epeau. Cryptology column – 25 years of quantum cryptography. SIGACT News, 27 (3):13–24, 1993.

[11] A.R. Calderbank and P.W. Shor. Good quantum error-correcting codes exist. Physical Review A, 54(2):1098–1105, 1996.

9

[12] I.L. Chuang, N. Gershenfeld, and M. Kubinec. Experimental implementation of fast quantum searching. Physical Review Letters, 80(15):3408–3411, 1998.

[13] I.L. Chuang, L.M.K. Vandersypen, X. Zhou, D.W. Leung, and S. Lloyd. Experimental realization of a quantum algorithm. Nature, 393:143–146, 1998.

[14] D. Deutsch. Quantum theory, the Church–Turing principle and the universal quantum computer. Proceedings of the Royal Society of London, Series A, 400(1818):97–117, 1985.

[15] D. Deutsch. Quantum computational networks. Proceedings of the Royal Society of London, Series A, 425(1868):73–90, 1989.

[16] D. Deutsch and R. Jozsa. Rapid solution of problems by quantum computation. Proceedings of the Royal Society of London, Series A, 439(1907):553–558, 1992.

[17] D.P. DiVincenzo. Two-bit gates are universal for quantum computation. Physical Review A, 51(2):1015–1022, 1995.

[18] R.P. Feynman. Simulating physics with computers. International Journal of Theoretical Physics, 21(6/7):467–488, 1982.
[19] L.K. Grover. A fast quantum mechanical algorithm for database search. In STOC ’96: Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing, pages 212–219, 1996.
[20] L.K. Grover. Quantum mechanics helps in searching for a needle in a haystack. Physical Review Letters, 79(2):325–328, 1997.
[21] R. Jozsa and B. Schumacher. A new proof of the quantum noiseless coding theorem. Journal of Modern Optics, 41(12):2343–2349, 1994.
[22] S. Lloyd. Almost any quantum logic gate is universal. Physical Review Letters, 75(2):346–349, 1995.
[23] R.L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM, 21(2):120–126, 1978.
[24] B. Schumacher. Quantum coding. Physical Review A, 51(4):2738–2747, 1995.
10
[25] C.E. Shannon. A mathematical theory of communication. Bell System Technical Journal, 27:379–423, 623–656, 1948.
[26] P.W. Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In Proceedings of the 35th Annual IEEE Symposium on the Foundations of Computer Science, pages 124–134, 1994.
[27] P.W. Shor. Scheme for reducing decoherence in quantum computer memory. Physical Review A, 52(4):R2493–R2496, 1995.
[28] P.W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM Journal on Computing, 26(5):1484–1509, 1997.
[29] P.W. Shor. Why haven’t more quantum algorithms been found? Journal of the ACM, 50(1):87–90, 2003.
[30] D.R. Simon. On the power of quantum computation. In Proceedings of the 35th Annual IEEE Symposium on Foundations of Computer Science, pages 116–123, 1994.
[31] D.R. Simon. On the power of quantum computation. SIAM Journal on Computing, 26(5):1474–1483, 1997.
[32] A.M. Steane. Error correcting codes in quantum theory. Physical Review Letters, 77(5):793–797, 1996.
[33] L.M.K. Vandersypen, G. Breyta, M. Steffen, C.S. Yannoni, M.H. Sherwood, and I.L. Chuang. Experimental realization of Shor’s quantum factoring algorithm using nuclear magnetic resonance. Nature, 414(6866):883–887, 2001.
[34] S. Wiesner. Conjugate coding. SIGACT News, 15(1):78–88, 1983.
[35] A.C-C. Yao. Quantum circuit complexity. In Proceedings of 34th Annual IEEE Symposium on Foundations of Computer Science, pages 352–361, 1993.

REFERENCES FOR "RESEARCH AREA" SECTION:

1. https://ai.googleblog.com/2018/08/understanding-performance-fluctuations.html
2. https://futurism.com/googles-new-quantum-processor
3. https://ai.google/research/teams/applied-science/quantum-ai/
4. https://www.semanticscholar.org/paper/Qubit-metrology-for-building-a-fault-tolerant-Martinis/cf8cc56f5c4eec4976349372af5b2ea29a438a59
5. https://ai.google/research/pubs/pub46225
6. https://www.inverse.com/article/38917-google-quantum-simulator-creates-a-butterfly
7. https://en.wikipedia.org/wiki/Quantum_optimization_algorithms
8. https://www.sigarch.org/hybrid-quantum-classical-computing/
9. https://medium.com/xanaduai/making-a-neural-network-quantum-34069e284bcf

REFERENCES FOR OTHER SECTIONS:

1. https://ai.google/research/teams/applied-science/quantum-ai/
2. https://gizmodo.com/googles-new-cirq-project-aims-to-make-quantum-computers-1827751628
3. https://towardsdatascience.com/why-quantum-computing-is-here-to-stay-d4ada7ddbd8d
4. https://en.wikipedia.org/wiki/Quantum_computing
5. https://en.wikipedia.org/wiki/Quantum
6. https://www.youtube.com/watch?v=JhHMJCUmq28
7. https://towardsdatascience.com/the-need-promise-and-reality-of-quantum-computing-4264ce15c6c0
8. https://medium.com/@jayanzaman/quantum-computing-landscape-b69c10e29704
9. https://www.youtube.com/watch?v=HdSmIUuGf-I
10. https://www.youtube.com/watch?v=OWJCfOvochA
11. https://www.youtube.com/watch?v=WVv5OAR4Nik
12. http://www.physics.open.ac.uk/quantum/qc.html
13. http://wavewatching.net/2012/03/24/a-brief-history-of-quantum-computing/
14. https://www.youtube.com/watch?v=CMdHDHEuOUE
15. https://www.technologyreview.com/s/611673/google-wants-to-make-programming-quantum-computers-easier/
16. https://www.technologyreview.com/s/612381/google-has-enlisted-nasa-to-help-it-prove-quantum-supremacy-within-months/
17. https://pdfs.semanticscholar.org/d7f5/03eddd3f39db53acd8495f52f357ebb707cd.pdf
18. https://hackernoon.com/quantum-programming-with-google-cirq-3209805279bc

TEAM DETAILS:

Team Leader :

Name	Student ID	Signature with date
1. Jamshid Khakimjonov	U1710260	3 May 2019

Team Members :

Name	Student ID	Signature with date
2. Sevara Amirullaeva	U1710142	3 May 2019
3. Ziyoda Saydakhmatova	U1710259	3 May 2019
4. Vladislav Podkovyrkin	U1710275	3 May 2019
5. Sarvinoz Maksudova	U1710276	3 May 2019