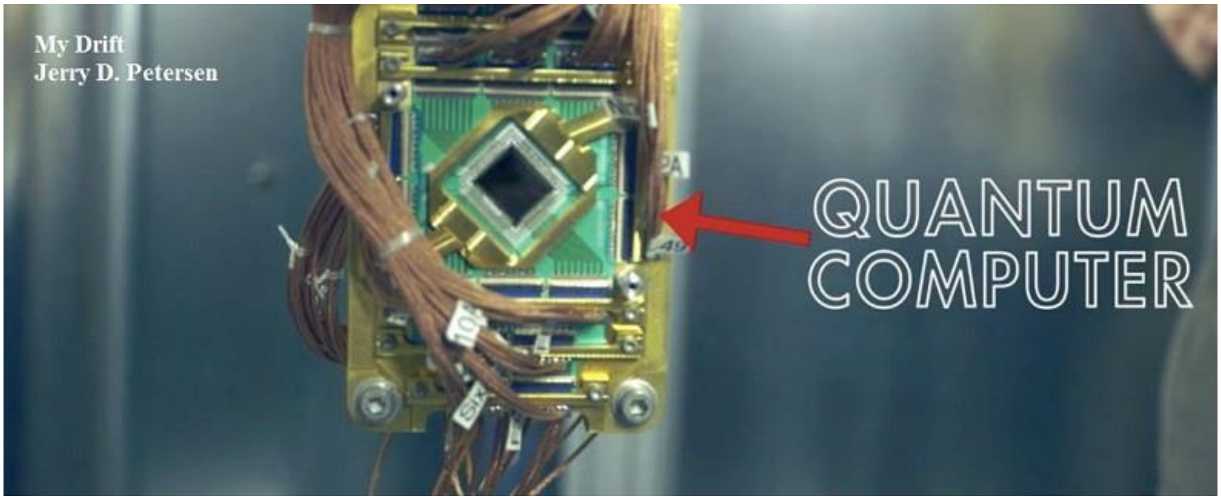



Monty

Quantum Computing

12 Jan 2017

246-2017-02

The next BIG THING in Information Technology and Computers is Quantum Computing. The Quantum Theory and the Quantum Computing concepts have been around for many years but up to now, nobody has been able to come up with a usable Quantum Computer that can help (or destroy) mankind. Up to this week, I knew nothing about Quantum Computing. The subject come up at the bar I was sitting at the other day and since I have worked with computers most of my life (since 1967), I have a great interest in new computing ideas and especially new programming languages. So, I went on the Internet to learn more about it but I must warn you that Quantum Processing is very difficult to understand and only computer Nerds, Geeks, and Techies will find this article interesting.

Every night I sit in our patio area to relax, watch sports on TV, have a cold beverage, and talk to our two Rottweiler dogs (Monty and Patti) about our day and the world's problems. Patti doesn't seem to care much about these things – she is more interested in when dinner is going to be served. But Monty is a deep thinker and after going through my findings with him, I think I know enough about Quantum Computing to write this article.

What is the most powerful supercomputer in the world?

China’s Sunway TaihuLight

The Sunway TaihuLight is indeed a monster: theoretical peak performance of 125 petaflops, 10,649,600 cores, and 1.31 petabytes of primary memory. That’s not just “big” - This is, like, mountain big. Shaq big. Jupiter big.

TaihuLight’s abilities are matched only by the ambition that drove its creation. Fifteen years ago, China claimed zero of the top 500 supercomputers in the world. The U.S. was King of Computer Technology. But today, China not only has more than everyone else—including the United States—but its best machine boasts speeds five times faster than the best the U.S. can muster. And, in a first, it achieves those speeds with purely China-made chips and materials.

The biggest U.S. supercomputer is the Titan that is the computational centerpiece of the Oak Ridge National Laboratory, pumping out 17.6 petaflops for the various research projects at the Department of Energy facility. It is the third biggest in the world. China also has the second biggest supercomputer called the Tianhe-2.

Computers have slowly started to encroach on activities we previously believed only the brilliantly sophisticated human brain could handle. IBM’s Deep Blue supercomputer beat Grand Master Garry Kasparov at chess in 1997, and in 2011

IBM's Watson supercomputer beat former human champions Ken Jennings and Brad Rutter at the quiz game Jeopardy. The Watson supercomputer processes at a rate of 80 teraflops (trillion floating-point operations per second).

The United States needs to get going!! Other countries are passing us up in almost every area you can think of. This is not good. This might be why Donald Trump won the election - the American people are sick and tired of being number 2 or 3 or 4. We want to be #1 and GREAT AGAIN. AMEN!

Here is something to think about. China's Sunway TaihuLight supercomputer is NOT a Quantum Computer. A 2,000 qu-bits Quantum Processor is 3,600 times faster and 300,000 times more powerful than China's Sunway TaihuLight supercomputer!!!!!!

Do I have your attention now?

There are several companies working on developing quantum computer hardware and software including IBM, Google, Microsoft, NASA, CIA, and D-Wave. Nobody seems to know if China or Russia have the quantum computer technology or not. If they do not, we know for sure that they will be getting it one way or another soon. The U.S. needs to put whatever resources it takes to get back to #1 and stay there.

Over the last 75 years, there have been many astounding developments – the first electronic programmable computer, the first integrated circuit computer, the first microprocessor. But the next anticipated step may be the most revolutionary of all. Quantum computing is the technology that many scientists, entrepreneurs and big businesses expect to provide a, well, quantum leap into the future.

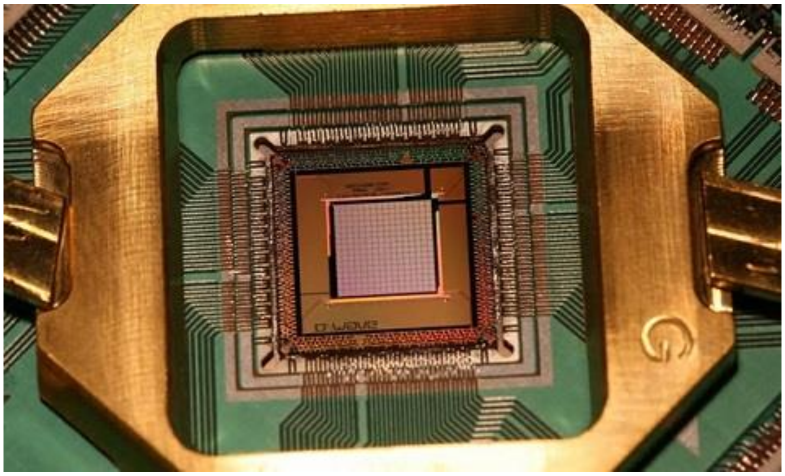
With normal computers, or classical computers as they're now called, there are only two options – on and off – for processing information. A computer "bit", the smallest unit into which all information is broken down, is either a "1" or a "0". And the computational power of a normal computer is dependent on the number of binary transistors – tiny power switches – that are contained within its microprocessor. Back in 1971 the first Intel processor was made up of 2,300 transistors. Intel now produce microprocessors with more than 5 billion transistors. However, they're still limited by their simple binary options. But with quantum computers the bits, or "qubits" as they are known, afford far more options owing to the uncertainty of their physical state.

In the mysterious subatomic realm of quantum physics, particles can act like waves, so that they can be particle or wave or particle and wave. This is what's known in

quantum mechanics as superposition. As a result of superposition, a qubit can be a 0 or 1 or 0 and 1. That means it can perform two equations at the same time. Two qubits can perform four equations. And three qubits can perform eight, and so on in an exponential expansion. That leads to some inconceivably large numbers, not to mention some mind-boggling working concepts.

A small Canadian company called D-Wave, a highly skilled collection of just 140 employees prides itself on building the world's first functioning quantum computer.

D-Wave Quantum Computers



Niobium Computer Chip Core Structure

This experimental processor itself looks like a large black fridge about 10 feet high. Within it is an elaborate structure of circuit boards with beautifully colorful niobium wafers as the centerpiece. It all looks fairly unremarkable, yet somewhere in here a multiplicity of different universes are thought to exist.

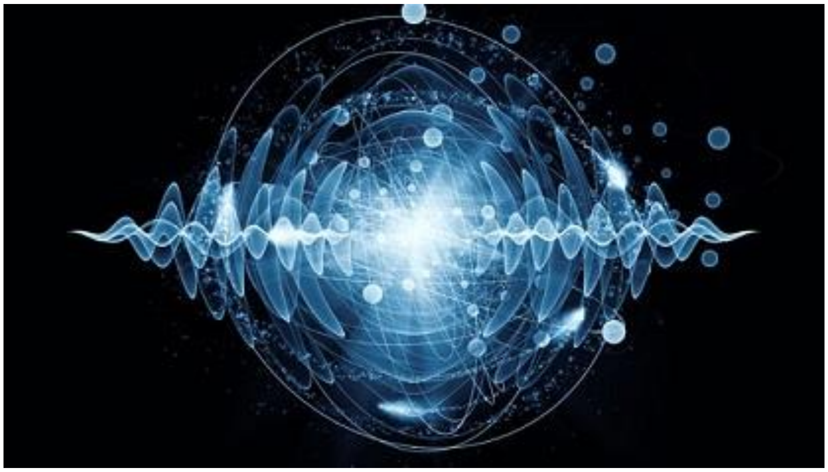
The quantum computer is contained within the large fridge-like casing. Actually, it is a fridge, the coldest fridge ever assembled. The cooling apparatus enables the niobium computer chip at its core to function at a temperature of just under –273C, or as close to absolute zero as the known universe can get. The super-cooled environment is necessary to maintain coherent quantum activity of superposition and entanglement, the state in which particles begin to interact – again rather mysteriously – co-dependently, and the qubits are linked by quantum mechanics regardless of their position in space. However, any intrusion of heat or light would corrupt the process and thus the effectiveness of the computer.

Well, I'm really confused now!

A one thousand qubit computer can be in 2 to the 1,000 states at one time, which is 10 to the 300th power. There's only 10 to the 80th atoms in the universe. Now does this mean it's in 10 to the 300th universes at the same time? Can billions of different universes coexist within one computer? That's the sort of questions that is probably best not grappled with before midnight and without the aid of illegal stimulants.

I will drink to that!! A beer for me and a Dasani water for Monty.

The more important question is – Do quantum computers work? At the moment, quantum computing still resides within a largely experimental, theoretical or speculative realm. The potential is staggering, involving a computational power many times the order of all the world's existing classical computers combined. But realizing that potential is a diabolically difficult task.



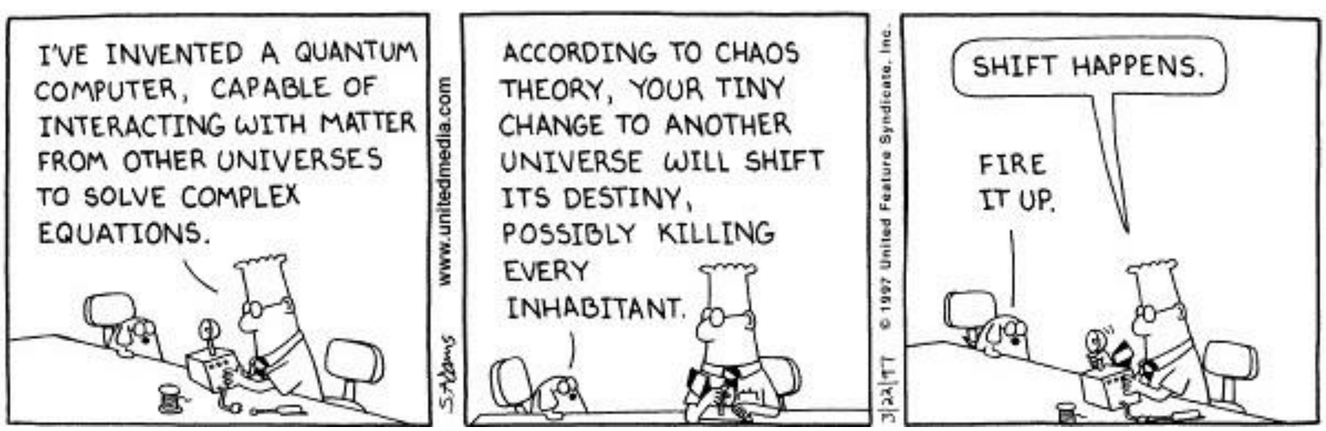
Most experimental quantum computers that had been assembled in labs followed the universal gate model, in which qubits substituted for transistors, without any notable success. D-Wave chose instead to develop an "adiabatic" quantum computer, which works by a process of what's called "quantum annealing" or "tunnelling". In essence, it means you develop an algorithm that assigns specific interactions between the qubits along the lines of the classical model – i.e. if this is a 0, that one is a 1 etc. Then create the conditions for quantum superposition, in which the qubits can realize their near infinite possibilities, before returning them to the classical state of 0s and 1s. The idea is the qubits will follow the path of least energy in relation to the algorithmic requirements, thus finding the most efficient answer.

If quantum mechanics is this hard to explain, imagine how difficult it is to build a fully functional quantum computer.

Scientists, engineers, and computer experts know that there are incredible potential applications of quantum computing. But, to this date, nobody has been able to fully take advantage of all this power. Experts agree that even if no one ever builds a useful quantum computer, we will still learn an immense amount by attempting to. This might sound like a cover up, but quantum mechanics itself is a theory so fundamental to our understanding of the universe, and is the seed to so many current and future other technologies

In areas such as artificial intelligence (AI) and cryptography, it's thought that quantum computing will transform the landscape, perhaps bringing about the breakthrough that will enable machines to "think" with the nuance and interpretative skill of humans.

Time for another beer and a little humor.



Even though they are not fully functional, D-Wave has started selling their 2X quantum computers for more than $15 million dollars each. A few companies and organizations have purchased one. They include Google, Lockheed Martin, NASA, and the CIA. It's clear that some of the world's most forward-looking institutions believe that the quantum computer has a future.

Okay, that was a brief overview of quantum theory and quantum computers. Now, let's look at quantum processing and programming in a little more detail.

What is Quantum Theory?

Quantum theory is 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 apply. Things on a very small scale behave like nothing you have any direct experience about... or like anything that you have ever seen.

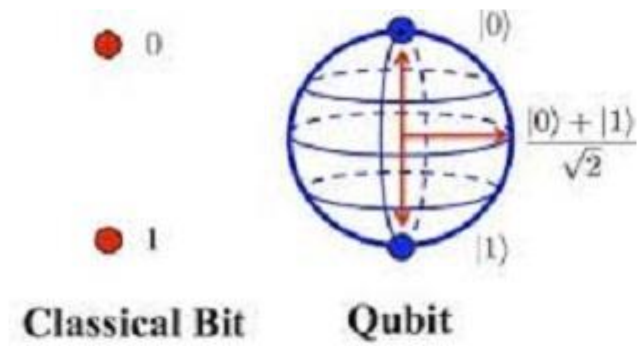
If you've studied light, you may already know a little bit about quantum theory. You might know that a beam of light sometimes behaves as though it's made up of particles (like a steady stream of cannonballs), and sometimes as though it's waves of energy rippling through space (a bit like waves on the 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 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? The answer is "maybe".

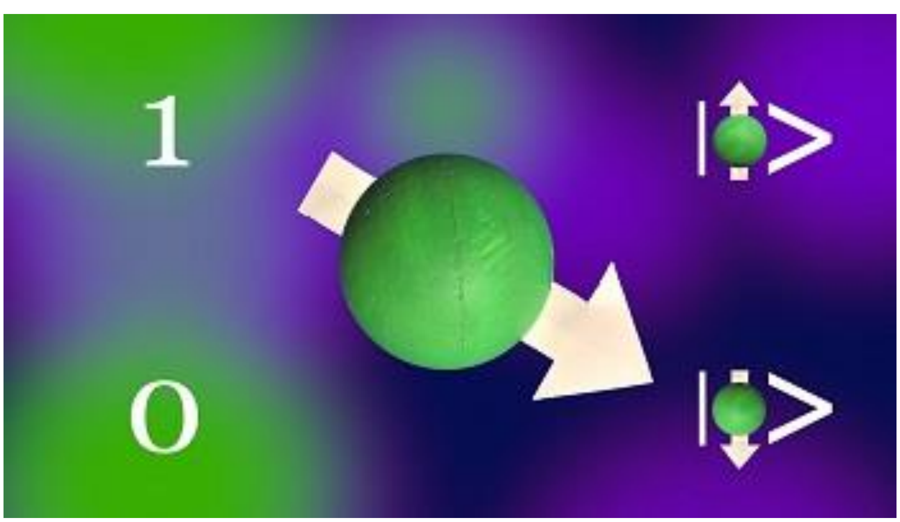
Quantum Computing

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). Qubits use superposition to represent multiple states (multiple numeric values) simultaneously in a similar way.

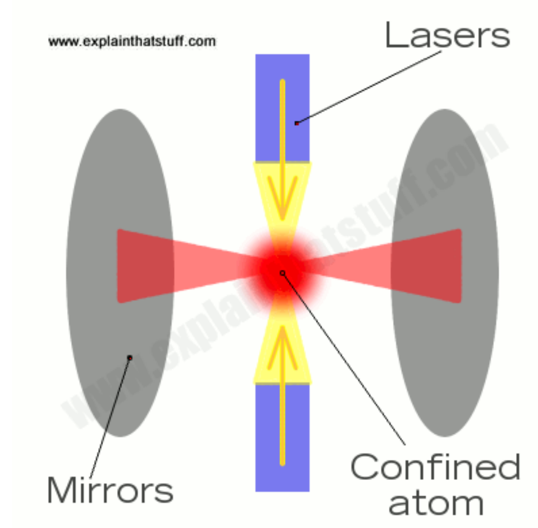


Just as a quantum computer can store multiple numbers at once, so it can process them simultaneously. Instead of working in serial (doing a series of things one at a time in a sequence), it can work in parallel (doing multiple things at the same time). Only when you try to find out what state it's actually in at any given moment (by measuring it) 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!



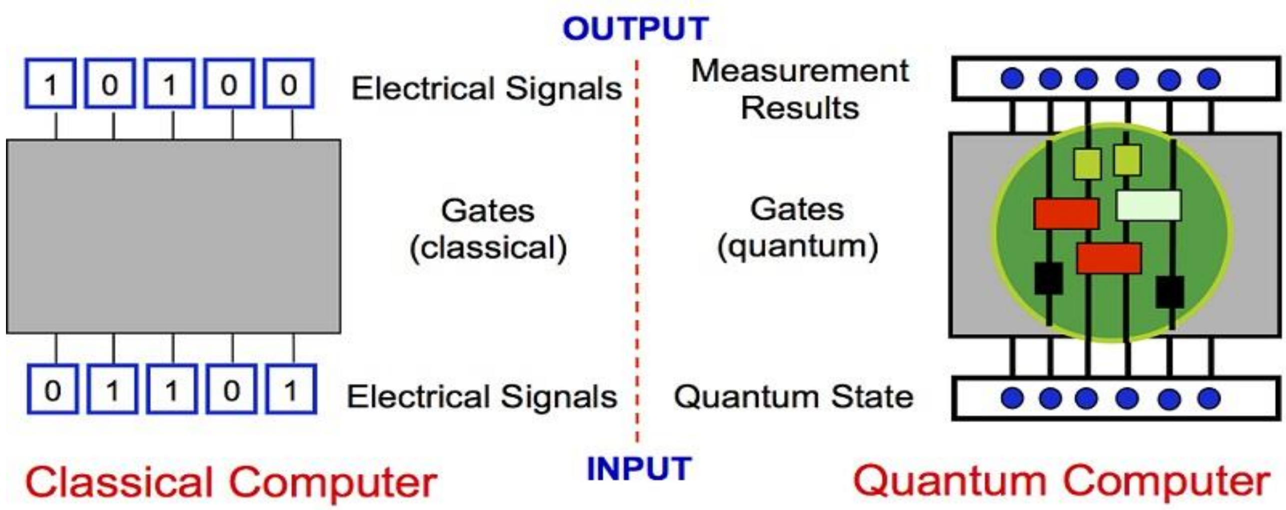
Quantum Computer Principles

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). 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.



A single atom 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 and then flip it into different states with laser pulses.

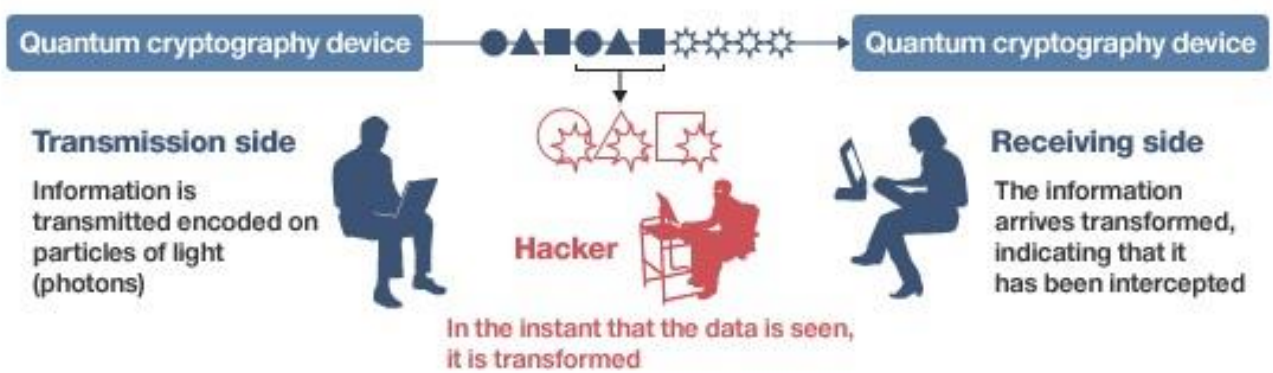


If you are still with me – It is time for another beer and a little more humor

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

Factoring large numbers, for starters. Multiplying two large numbers is easy for any computer. But calculating the factors of a very large (say, 500-digit) number, on the other hand, is considered impossible for any classical computer.

In fact, the difficulty of factoring big numbers is the basis for much of our present day cryptography. It's based on math problems that are too tough to solve. RSA encryption, the method used to encrypt your credit card number when you're shopping online, relies completely on the factoring problem. The website you want to purchase from gives you a large "public" key (which anyone can access) to encode your credit card information.



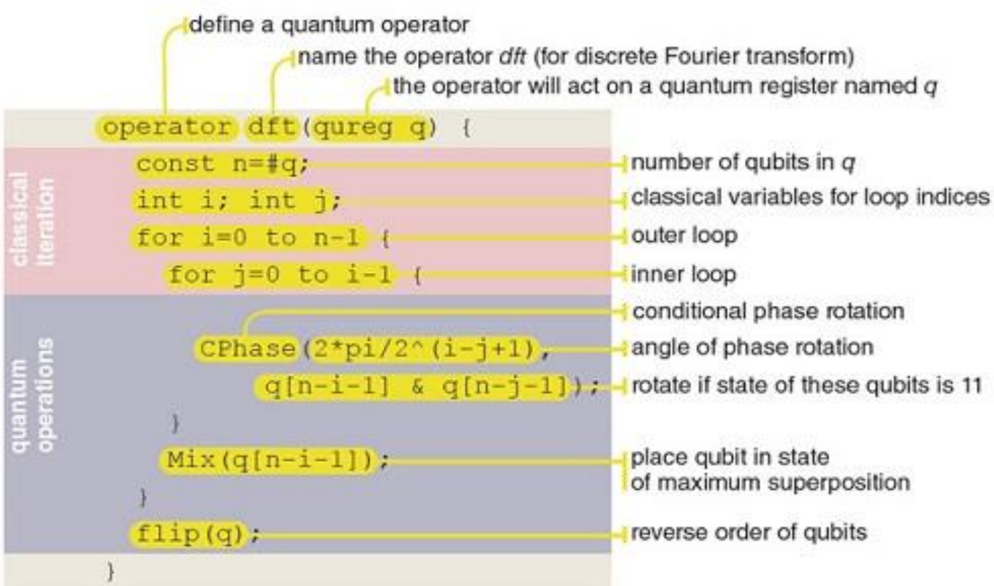
Wait… so a quantum computer will be able to hack into my private data? That's not good! Don't worry — classical cryptography is not completely jeopardized. Although certain aspects of classical cryptography would be jeopardized by quantum computing, quantum mechanics also allows for a new type of highly secure cryptography. So, harnessing the quantum world can break and make codes.

Can quantum computers help mankind?

Yes. For example, quantum computers will be able to efficiently simulate quantum systems, which has been said to be a "holy grail" of quantum computing: it will allow us to study, in remarkable detail, the interactions between atoms and molecules. This could help us design new drugs and new materials, such as superconductors that work at room temperature. Another of the many tasks for which the quantum computer is inherently faster than a classical computer is at searching through a space of potential solutions for the best solution. Researchers are constantly working on new quantum algorithms and applications. But the true potential of quantum computers likely hasn't even been imagined yet. The inventors of the laser surely didn't envision supermarket checkout scanners, CD players and eye surgery. Similarly, the future uses of quantum computers are bound only by one's imagination.

Finally, we have reached the part of this subject that I'm most interested in – How do you program this beast? Are there programming languages available for quantum computers?

The fully functional hardware doesn't yet exist, but languages for quantum coding are ready to go. Well, maybe they are ready? Both Microsoft and IBM are working on new and improved Quantum Programming Languages. Here is some sample code and some of the current programming languages:



A sample program for computing the discrete Fourier transform (above) is written in the Quantum Computing Language (QCL). The language combines elements that require a classical—that is, nonquantum—computer (above code in pink) with operations that are unique to quantum processors (above code in blue).

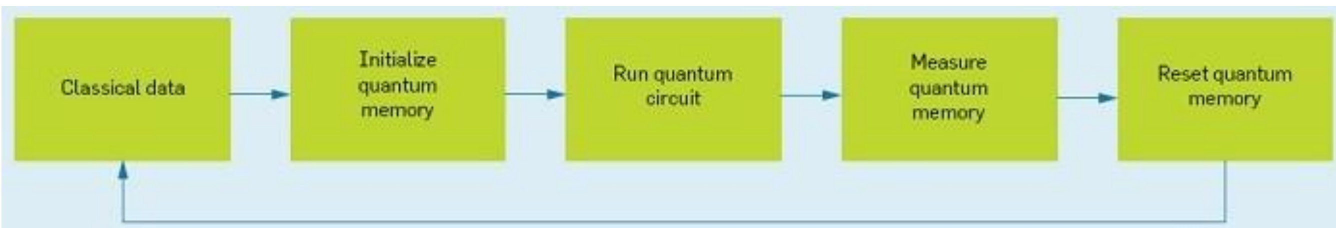
Quantum computing language (QCL)

QCL (Quantum Computation Language) is one of the first implemented quantum programming languages. Its syntax resembles the syntax of the C Programming Language and its classical data types are similar to primitive data types in C. One can combine classical code and quantum code in the same program.

Q language

Q Language is the second implemented imperative quantum programming language. It was implemented as an extension of C++ programming language. New operators can be defined using C++ class mechanism.

QFC and QPL are two closely related functional quantum programming languages defined by Peter Selinger. They differ only in their syntax: QFC uses a flow chart syntax, whereas QPL uses a textual syntax. These languages have classical control flow but can operate on quantum or classical data.



Quipper Programming Language

Quipper is an embedded functional programming language that provides:

- **A high-level circuit description language - This includes gate-by-gate descriptions of circuit fragments.**
- **A monadic semantics, allowing for a mixture of procedural and declarative programming styles.**
- **Built-in facilities for automatic synthesis of reversible quantum circuits, including from classical code.**
- **Extensible quantum data types.**
- **Programmable circuit transformers.**
- **Support for three execution phases: compile time, circuit generation time, and circuit execution time.**
- **A dynamic lifting operation to allow circuit generation to be parametric on values generated at circuit execution time.**
- **Extensive libraries of quantum functions. i.e. Pre-coded routines that perform a particular function.**

The Quantum Mystique

Programming a Quantum Computer is HARD!! It requires a completely different mindset than us Computer Programmers are used to. The following paragraphs explain some of these differences:

A computer is a physical object, which has to obey the laws of nature as well as any rules of logic or mathematics that the designer wants to impose. You can't entirely ignore the physical substrate (the construction of integrated circuits)—and that goes double for a quantum computer.

The bits of a classical computer are just binary digits, with a value of either 0 or 1. Almost any device with two distinct states can serve to represent a classical bit: a switch, a valve, a magnet, or a coin. Qubits, partaking of the quantum mystique, can occupy a superposition of 0 and 1 states. What does that mean? It's not that the qubit can have an intermediate value, such as 0.63; when the state of the qubit is measured, the result is always 0 or 1. But, in the course of a computation a qubit can act as if it were a mixture of states—say, 63 percent 0 and 37 percent 1. Only a few physical systems exhibit superposition clearly enough to function as qubits. Examples include photons with two directions of polarization, atomic nuclei with two spin orientations, and superconducting loops with clockwise and counterclockwise electric currents.

Another key aspect of qubit behavior is interference, a phenomenon well known in the physics of waves. When two waves overlap, they can either reinforce each other (if the peaks and valleys of the undulations coincide) or they can cancel (if the waves are out of phase). Mathematically, the intensity of the combined waves at any point is given by the square of the sum of the individual wave amplitudes. When the two amplitudes have the same sign, the interference is constructive; when one amplitude is positive and the other negative, the resulting destructive interference yields an intensity less than that of either wave alone.

Like waves, the 0 and 1 states of a qubit have amplitudes that can be either positive or negative. Depending on the signs of the amplitudes, quantum interference can either increase or decrease the probability that a specific state will be observed when the qubit is measured.

Interference plays a role in all interesting algorithms for quantum computers—that is, the algorithms that might enable such a machine to outperform a classical computer. The general idea is to arrange the evolution of the quantum system so that wrong answers are suppressed by destructive interference and right answers are enhanced by constructive interference. In this way, the algorithms exploit a form of

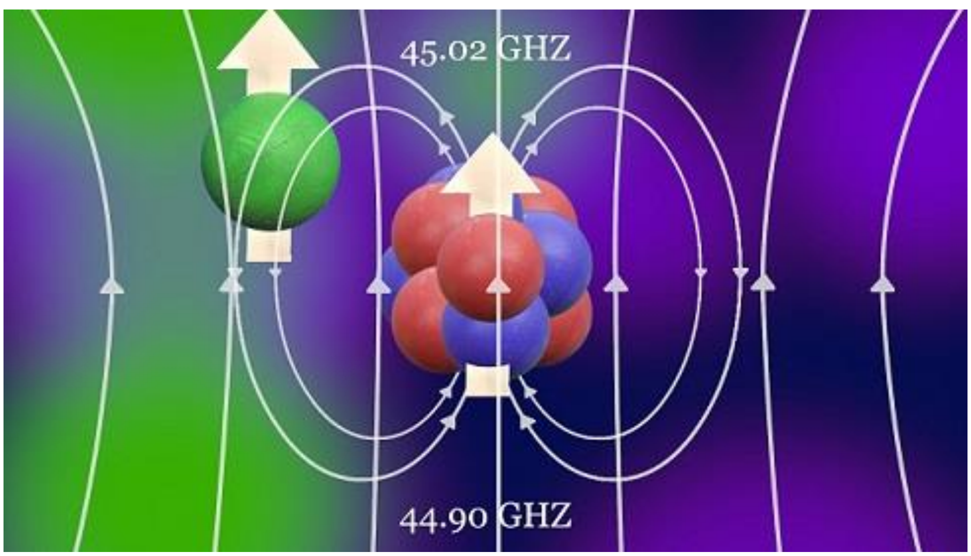
parallelism that's unique to quantum systems. In effect, a collection of n qubits can explore all of its 2n possible configurations at once; a classical system has to look at the 2n bit patterns one at a time.

One last aspect of quantum weirdness is entanglement. When two or more qubits interact, they may form a fused state that cannot be teased apart to show the contributions of individual qubits. In other words, you cannot poke around inside a quantum register and alter one qubit while leaving the rest undisturbed. Entanglement is a prerequisite for at least some of the more important quantum algorithms.

Among those algorithms, the best known is a procedure for finding the factors of integers, devised in 1994 by Peter W. Shor, now at MIT. When factoring an n-digit number, the fastest known classical algorithms take an amount of time that grows exponentially with n; Shor's algorithm works in time proportional to n3. For large enough n, the quantum algorithm is far faster.

Some of the things Quantum programmers have to deal with:

- **superposition of 0 and 1 states,**
- **the physics of waves,**
- **destructive interference,**
- **amplitudes that can be either positive or negative,**
- **and entanglement.**



Hold on – I need to run to the store for another 12-pack!

Quantum Computing Tutorial
(Presented by D-Wave)

The following material is written using very high level concepts and is designed to be accessible to both technical and non-technical audiences. This tutorial is intended to introduce the concepts and terminology used in Quantum Computing, to provide an overview of what a Quantum Computer is, and why you might want to program one.

What you will learn

By following through the material in this primer, you will learn:

- **How quantum physics gives us a new way to compute data.**
- **The similarities and differences between quantum computing and classical computing.**
- **How the fundamental units of quantum computing (qubits) are manipulated to solve hard problems.**
- **Why Quantum Computing is well suited to Artificial Intelligence (AI) and machine learning applications, and how quantum computers may be used as AI co-processors.**

(Go to next page)

SECTION 1

1.1 - Conventional computing

To understand quantum computing, it is useful to first think about conventional computing. We take modern digital computers and their ability to perform a multitude of different applications for granted. Our desktop PCs, laptops and smart phones can run spreadsheets, stream live video, allow us to chat with people on the other side of the world, and immerse us in realistic 3D environments. But at their core, all digital computers have something in common. They all perform simple arithmetic operations. Their power comes from the immense speed at which they are able to do this. Computers perform billions of operations per second. These operations are performed so quickly that they allow us to run very complex high level applications. Conventional digital computing can be summed up by the diagram shown in figure 1.

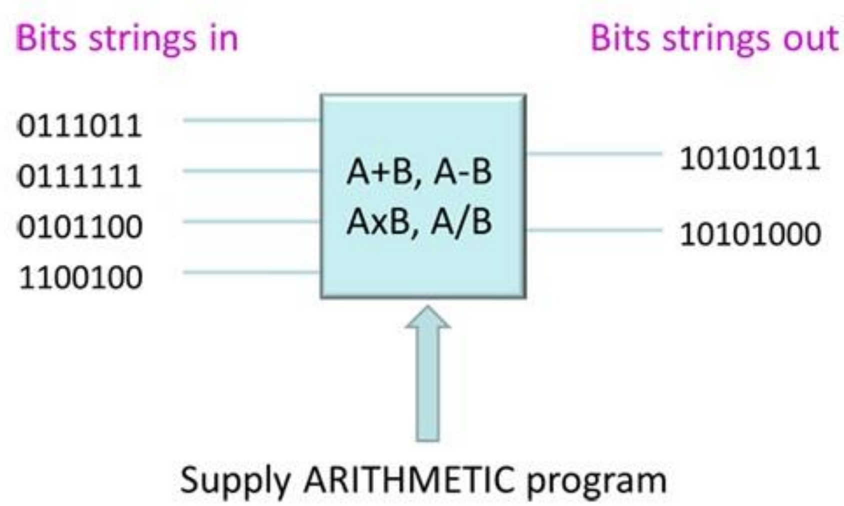


Figure 1. Dataflow in a conventional computer

Although there are many tasks that conventional computers are very good at, there are still some areas where calculations seem to be exceedingly difficult. Examples of these areas are: Image recognition, natural language (getting a computer to understand what we mean if we speak to it using our own language rather than a programming language), and tasks where a computer must learn from experience to become better at a particular task. Even though there has been much effort and research poured into this field over the past few decades, our progress in this area has been slow and the prototypes that we do have working usually require very large supercomputers to run them, consuming vast quantities of space and power.

We can ask the question: Is there a different way of designing computing systems, from the ground up? If we could start again from scratch and do something completely different, to be better at these tasks that conventional computers find hard, how would we go about building a new type of computer?

1.2 - A new kind of computing

Quantum computing is radically different from the conventional approach of transforming bits strings from one set of 0's and 1's to another. With quantum computing, everything changes. The physics that we use to understand bits of information and the devices that manipulate them are totally different. The way in which we build such devices is different, requiring new materials, new design rules and new processor architectures. Finally, the way we program these systems is entirely different. This document will explore the first of these issues, how replacing the conventional bit (0 or 1) with a new type of information - the qubit - can change the way we think about computing.

1.3 - The light switch game

To begin learning about quantum computing, it is important to understand why we can't use conventional digital computers to solve certain problems. Let's consider a mathematical problem, which we'll call the light switch game, that illustrates this point.

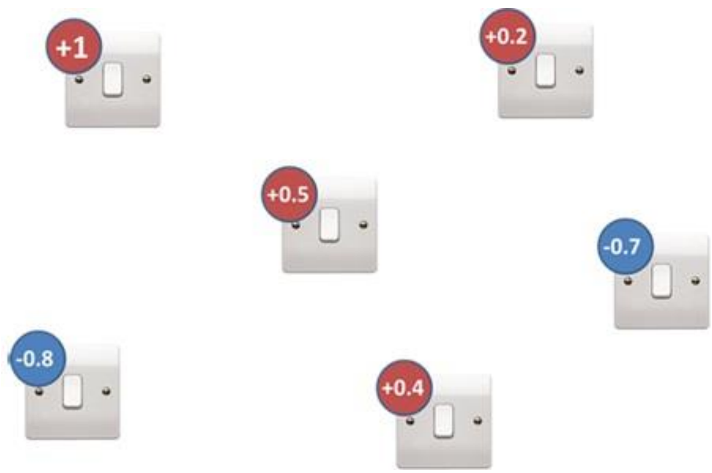


Figure 2. The light switch game

The light switch game involves trying to find the best settings for a bunch of switches. Here's a graphical example introducing this problem: Let's imagine that each light switch has a number associated with it, which is chosen for you (you don't get to change this). We call this a 'bias value'. You get to choose whether to turn each light switch ON or OFF. In our game, ON = +1 and OFF = -1. We then add up all the switches' bias values multiplied by their ON/OFF values. This gives us a number. The objective of the game is to set the switches to get the lowest number. Mathematically, we call the bias values of each switch h_i and the switch settings are called s_i.

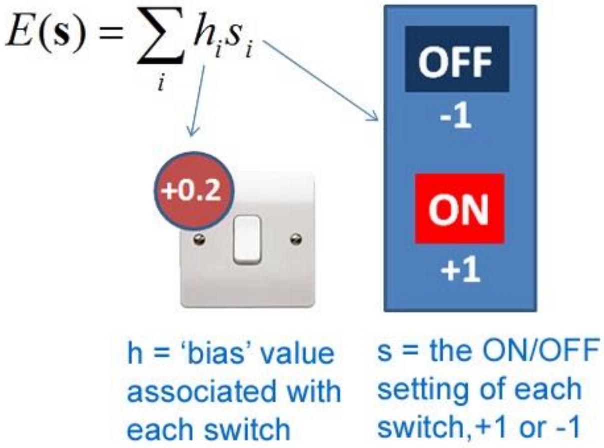


Figure 3. Playing the light switch game - add up the bias values of each switch multiplied by their settings (which you have to choose)

So, depending upon which switches we set to +1 and which we set to -1, we will get a different score overall. You can try this game. Hopefully you'll find it easy because there's a simple rule to winning:

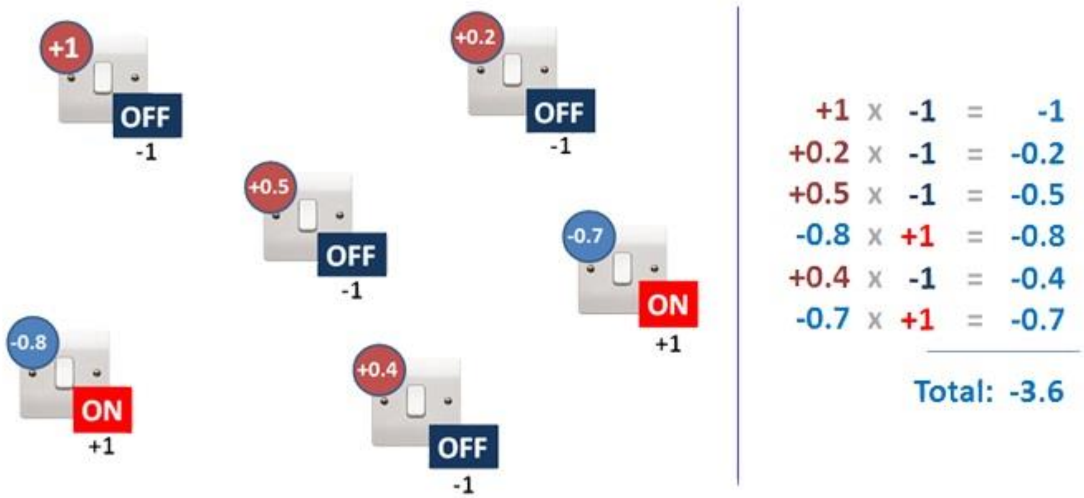


Figure 4. Working out the answer for a particular "guess" at the switch settings

We find that if we set all the switches with positive biases to OFF and all the switches with negative biases to ON and add up the result then we get the lowest overall value. Easy, right? I can give you as many switches as I want with many different bias values and you just look at each one in turn and flip it either ON or OFF accordingly.

OK, let's make it harder. So now imagine that many of the pairs of switches have an additional rule, one which involves considering PAIRS of switches in addition to just individual switches... we add a new bias value (called J) which we multiply by BOTH the switch settings that connect to it, and we add the resulting value we get from each pair of switches to our overall number too. Still, all we have to do is decide whether each switch should be ON or OFF subject to this new rule.

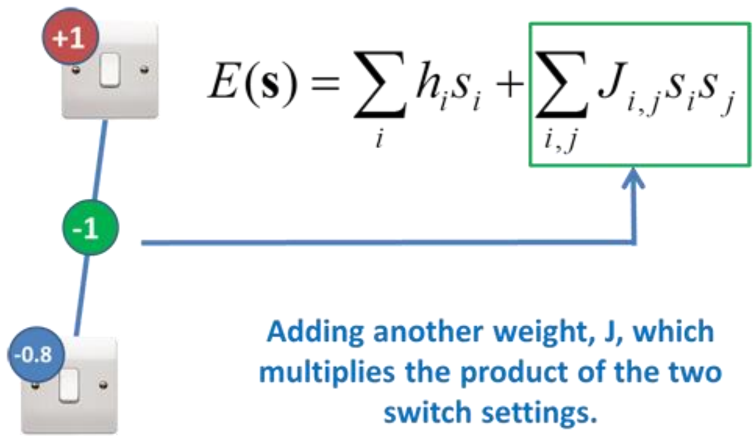


Figure 5. Making the game harder by adding additional terms that depend on the settings of pairs of switches.

But now it is much harder to decide whether a switch should be ON or OFF, because its neighbors affect it. Even with the simple example shown with 2 switches in the figure above, you can't just follow the rule of setting them to be the opposite sign to their bias value anymore (try it!). With a complex web of switches having many neighbors, it quickly becomes very frustrating to try and find the right combination to give you the lowest value overall.

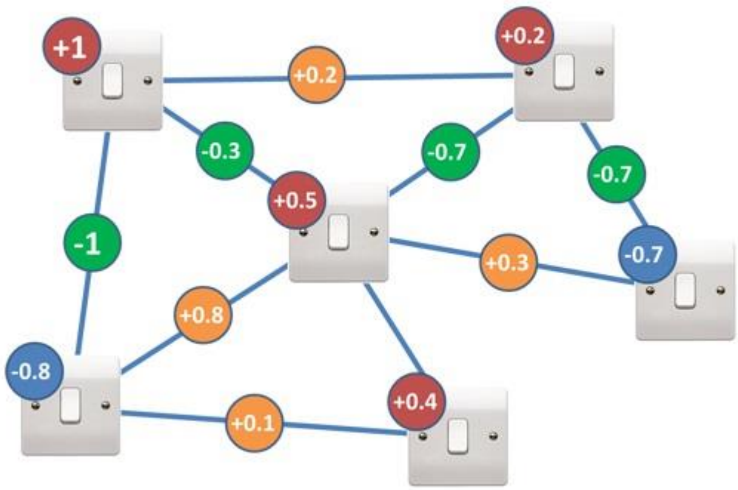


Figure 6. The light switch game with connecting terms added, generating an 'interacting' web of light switches

1.4 - How does quantum mechanics help?

With a couple of switches, you can just try every combination of ON's and OFF's, there are only four possibilities: [ON ON], [ON OFF], [OFF ON] or [OFF OFF]. But as you add more and more switches, the number of possible ways that the switches can be set grows exponentially:

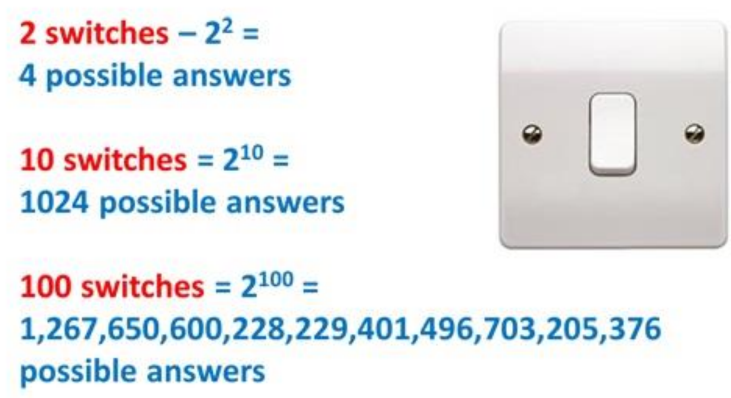


Figure 7. The exponential problem with the light switch game.

You can start to see why the game isn't much fun anymore. In fact, it is even difficult for our most powerful supercomputers. Being able to store all those possible configurations in memory, and moving them around inside conventional processors to calculate if our guess is right takes a very, very long time. With only 500 switches, there isn't enough time in the Universe to check all the configurations.

Quantum mechanics can give us a helping hand with this problem. The fundamental power of a quantum computer comes from the idea that you can put bits of information into a superposition of states. You can think of this as being a situation where the qubit has not yet decided which state it wants to be in. Some people like to refer to the qubit in superposition as 'being in both states at the same time'. You can alternatively think of the qubit's state as being undecided as to whether it is +1 or -1. Which means that using a quantum computer, our light switches can be ON and OFF at the same time:

Figure 8: A quantum mechanical bit of information (qubit) can reside in what is known as a superposition state, where it has not yet decided whether to be in the +1 or the -1 state (alternatively, you can think of it as being 'in both states').

Now let’s consider the same bunch of switches as before, but now held in a quantum computer's memory (notice that the bias values haven't been added yet):

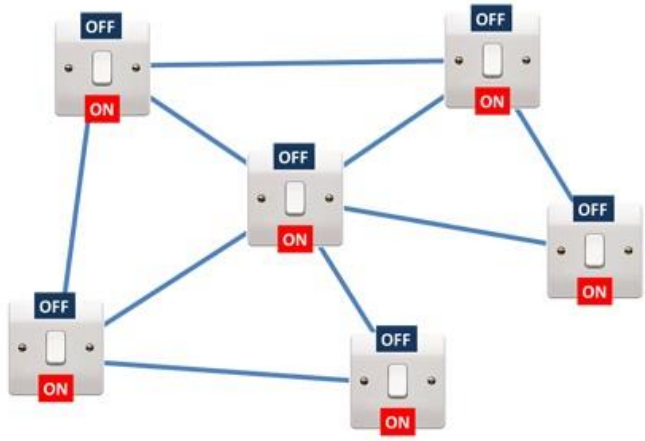


Figure 9. A network of connected quantum bits in superposition. The answer is in there somewhere!

Because all the light switches are on and off at the same time, we know that the correct answer (correct ON/OFF settings for each switch) is represented in there somewhere - it is just currently hidden from us. But that is OK, because quantum mechanics is going to find it for us. The quantum computer allows you to take a 'quantum representation' like this, and extract the configuration of ONs and OFFs with the lowest value. Here's is how it works:

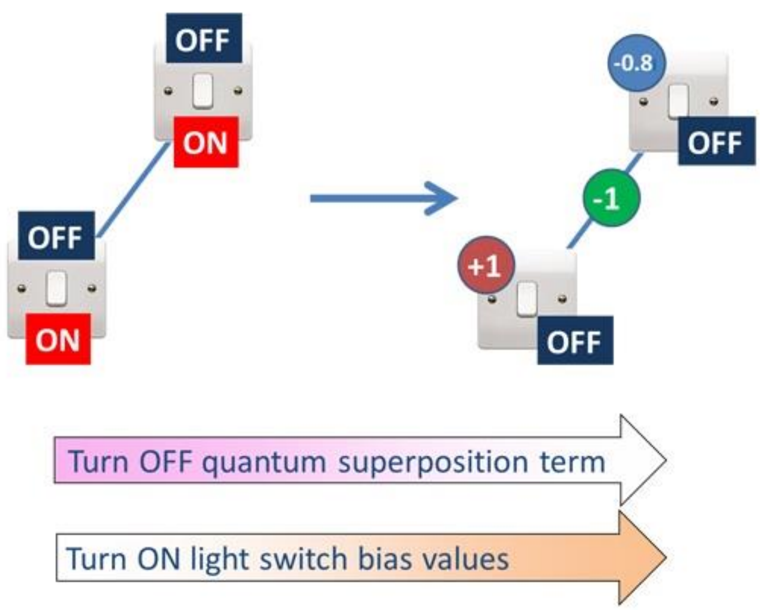


Figure 10. The computer begins with the bits in superposition, ends with them behaving as regular classical bits, and finds the answer along the way.

You start with the system in its quantum superposition as described above, and you slowly adjust the quantum computer to turn off the quantum superposition effect. At the same time, you slowly turn up all those bias values (the h and J's from earlier). As this operation is performed, the switches slowly drop out of their superposition state and choose a classical state, either ON or OFF. At the end, each switch MUST have chosen to be either ON or OFF. The quantum mechanics working inside the computer helps the light switches settle into the right states to give the lowest overall value when you add them all up at the end. Even though with N switches there are 2^N possible configurations it could have ended up in, it finds the lowest one, winning the light switch game. So, we see that the quantum computer allows us to minimize expressions such as the one considered here:

$$E(s) = \sum_i h_i s_i + J_{ij}s_i s_j$$

which can be difficult (if not impossible) for classical computers.

(Go to next page)

SECTION 2

2.1 - It's a math expression - who cares?

We didn't build a machine to play a strange masochistic light switch game. The concept of finding a good configuration of binary variables (switches) in this way lies at the heart of many problems that are encountered in everyday applications. A few are shown in figure below. Even the concept of scientific discovery itself is an optimization problem (you are trying to find the best 'configuration' of terms contributing to a scientific equation which match real world observations).

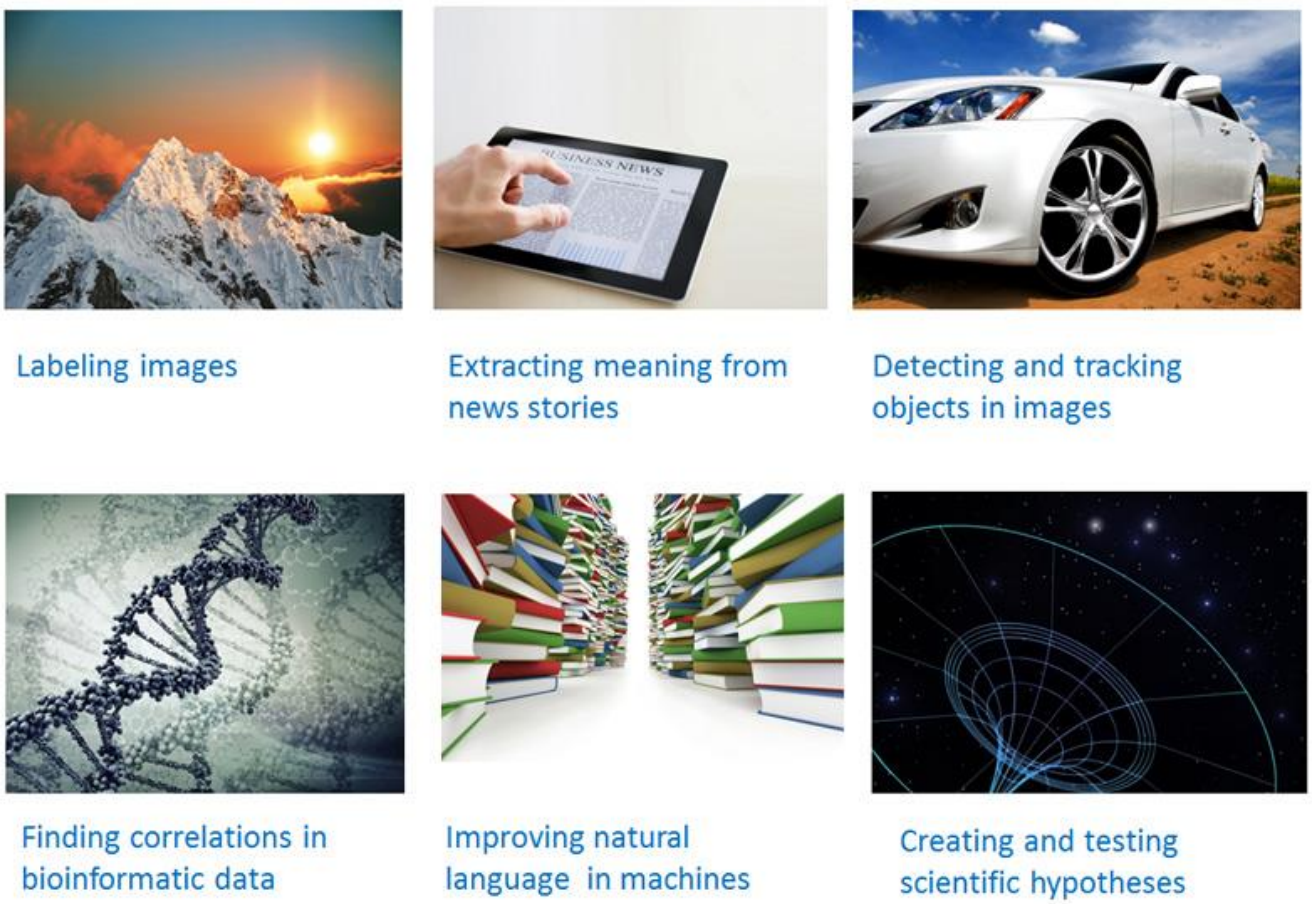


Figure 11. Examples of applications which under the hood all involve finding good 'switch settings' and can be tackled more efficiently with quantum computers.

2.2 - The energy program

To understand how these problems might be cast as finding settings of switches, let's consider how the quantum computer is programmed. Recall Figure 1, where bit strings in were transformed into other bits strings via the application of a logic program. Instead of that, we now have a resource where bits can be undecided, so the computation is performed in a fundamentally different way, as shown in Figure 12. In this case, a group of qubits are initialized into their superposition states, and this time an ENERGY PROGRAM (instead of a logic program) is applied to the group. The qubits go from being undecided at the beginning of the computation, to all having chosen either -1 or +1 states at the end of the computation. What is an Energy Program? It is just those h and J numbers - the bias settings - that we introduced earlier. In the light switch game, we said that the h and J's were given to you. Well, now we see where they come from - they are the definition of the problem you are trying to solve.

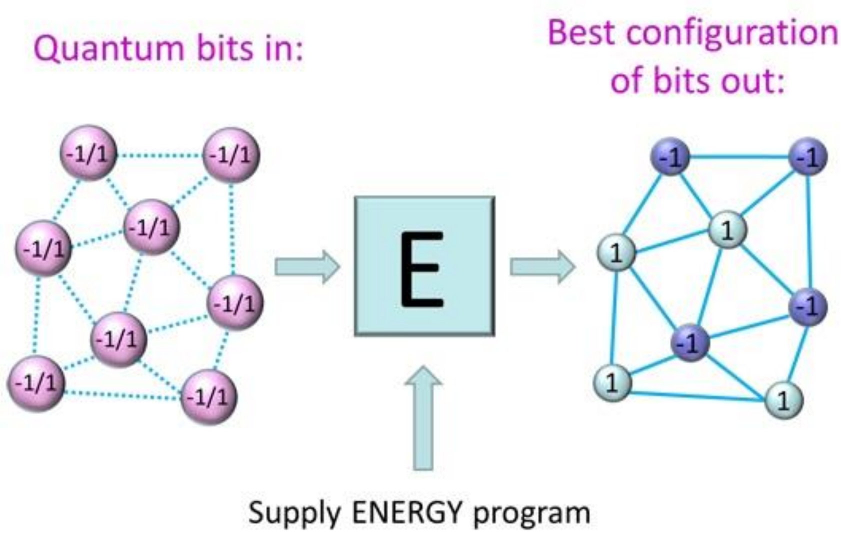


Figure 12. The basic operation of a quantum computer is to supply an energy program (a series of h and J numbers) and let the computer find the switch settings (+1 and -1).

Crafting an energy program as a series of h and J values - to encode a real-world problem that you care about - is exceedingly hard and time-consuming. It would be the equivalent of programming your desktop PC by sending in machine code to the microprocessors inside! Luckily, there is a better way to program quantum computers by using a quantum compiler. This process is explained in more detail in the Programming with D-Wave white paper. URL is:

http://www.dwavesys.com/sites/default/files/Map%20Coloring%20WP2.pdf

2.3 - Quantum computers can LEARN

The discipline of teaching computers to reason about the world and learn from experience is known as machine learning. It is a sub-branch of the field of artificial intelligence. Most of the code we write is fairly static - that is, given new data it will perform the same computation over and over again and make the same errors. Using machine learning we can design programs which modify their own code and therefore learn new ways to handle pieces of data that they have never seen before.

The type of applications that run very well on D-Wave's hardware are applications where learning and decision making under uncertain conditions are required. For example, imagine if a computer was asked to classify an object based on several images of similar objects you had shown it in the past. This task is very difficult for conventional computing architectures, which are designed to follow very strict logical reasoning. If the system is shown a new image, it is hard to get it to make a general statement about the image, such as 'it looks similar to an apple'. D-Wave's processors are designed to support applications that require high level reasoning and decision making.

How can we use a quantum computer to implement learning, for example, if we want the system to recognize objects? Writing an energy program for this task would be very difficult, even using a quantum compiler, as we do not know in detail how to capture the essence of objects that the system must recognize. Luckily there is a way around this problem, as there is a mode in which the quantum computer can tweak its own energy program in response to new pieces of incoming data. This allows the machine to make a good guess at what an object might be, even if it has never seen a particular instance of it before. The following section gives an overview of this process.

2.4 - A computer that programs itself

In order to get the system to tweak its own energy program, you start by showing the system lots and lots of instances of the concept that you want it to learn about. An example in shown in Figure 13. Here the idea is to try to get the computer to learn the difference between images of different types of fruit. In order to do this, we present images (or rather, a numeric representation of those images) to the system illustrating many different examples of apples, raspberries and melons. We also give the system the 'right' answer each time by telling it what switch settings (labels) it should be ending up selecting in each case. The system must find an energy program (shown as a question mark as we do not know it at the beginning) so that when an image is shown to the system, it gets the labels correct each time. If it gets many examples wrong, the algorithm knows that it must change its energy program.

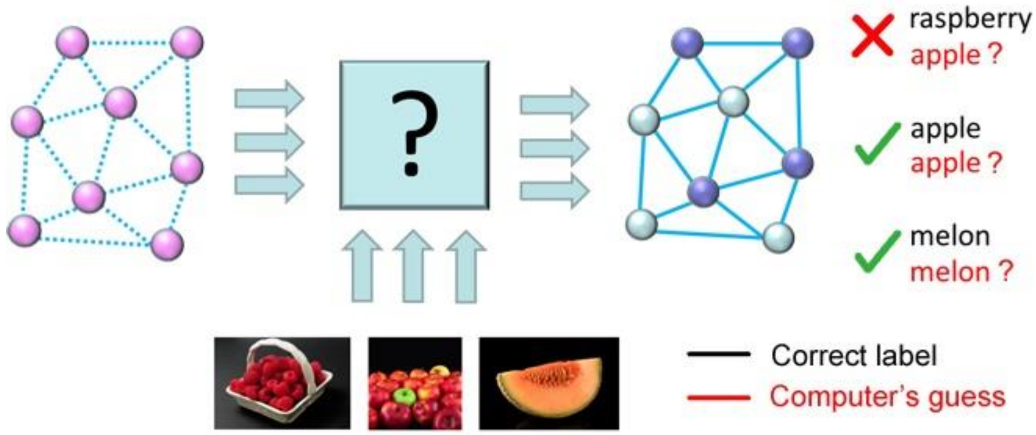


Figure 13. Teaching the quantum chip by allowing it to write its own energy program. The system tweaks the energy program until it labels all the examples that you show it correctly. This is also known as the 'training' or 'learning' phase.

At first the system chooses an energy program (remember that it is just a bunch of h and J values) at random. It will get many of the labels wrong, but that doesn't matter, as we can keep showing it the examples and each time allow it to tweak the energy program so that it gets more and more labels (switch settings) correct. Once it can't do any better on the data that it has been given, we then keep the final energy program and use that as our 'learned' program to classify a new, unseen example (figure 14).

In machine learning terminology, this is known as a supervised learning algorithm because we are showing the computer examples of images and telling it what the correct labels should be in order to help it learn. There are other types of learning algorithms supported by the system, even ones that can be used if labeled data is not available.

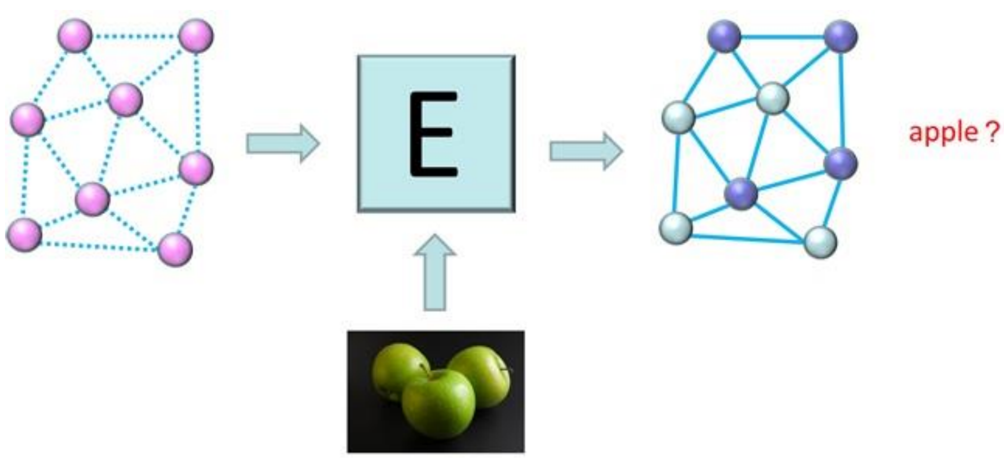
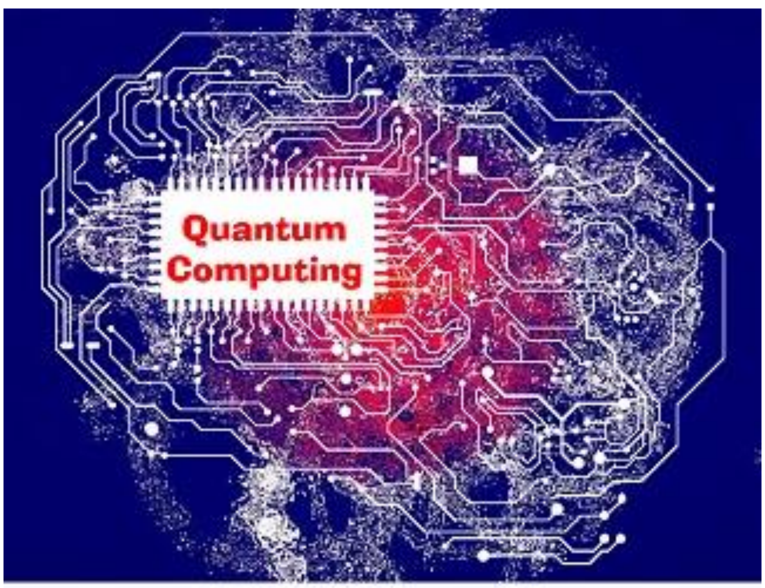


Figure 14. After the system has found a good energy program during the training phase, it can now label unseen examples to solve a real world problem. This is known as the 'testing' phase.

2.5 - Uncertainty is a feature

Another interesting point to note about the quantum computer is that it is probabilistic, meaning that it returns multiple answers. Some of these might be the answer that you are looking for, and some might not. At first this sounds like a bad thing, as a computer that returns a different answer when you ask it the same question sounds like a bug! However, in the quantum computer, this returning of multiple answers can give us important information about the confidence level of the computer. Using the fruit example above, if we showed the computer an image and asked it to label the same image 100 times, and it gave the answer 'apple' 100 times, then the computer is pretty confident that the image is an apple. However, if it returns the answer apple 50 times and raspberry 50 times, what this means is that the computer is uncertain about the image you are showing it. And if you had shown it an image with apples AND raspberries in, it would be perfectly correct! This uncertainty can be very powerful when you are designing systems which are able to make complex decisions and learn about the world.

End of Tutorial



Okay, that is it! If you made it to here, you know more about Quantum Processing than 99% of the people in the world. Congratulations!!