

Slide 1



The background image shows a person from behind, sitting at a desk and reading a large textbook titled 'PRECALCULUS'. The person is in a room with a whiteboard covered in various mathematical notes, diagrams, and sticky notes. One note prominently displays the formula $\sin(x)/x$. Another note has the title 'Fun Functions' and includes the expression $y = \sin(x)$. The Microsoft logo is visible in the top left corner of the slide.

 Microsoft

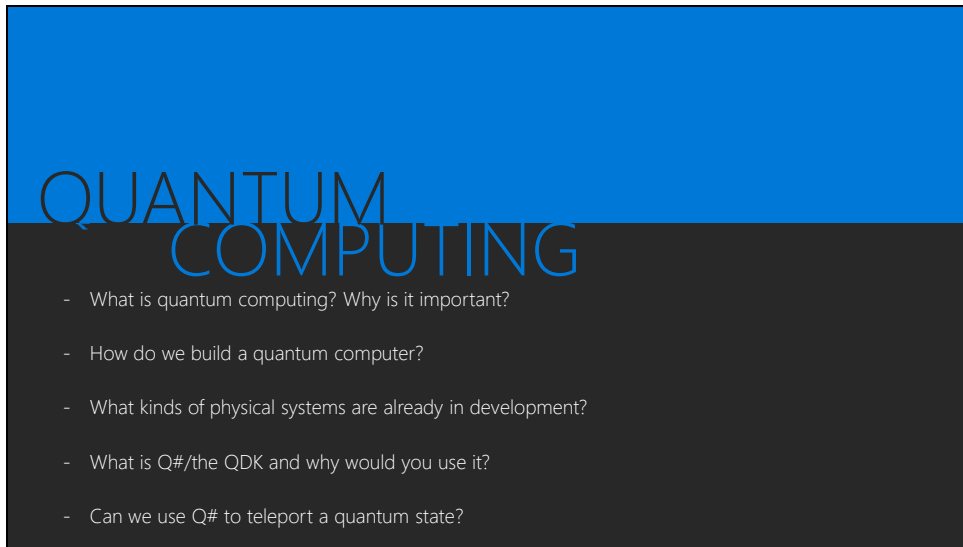
Quantum Computing: Don't Panic

Anita Ramanan | Frances Tibble
Software Development Engineers @ Microsoft

<https://aka.ms/quantumadventures>

@whywontitbuild | @frances_tibble
anraman@microsoft.com | fritibble@microsoft.com

Slide 2

A presentation slide with a blue header and a dark grey body. The title 'QUANTUM COMPUTING' is centered in the header. Below the title, a bulleted list of five topics is presented in the dark grey area.

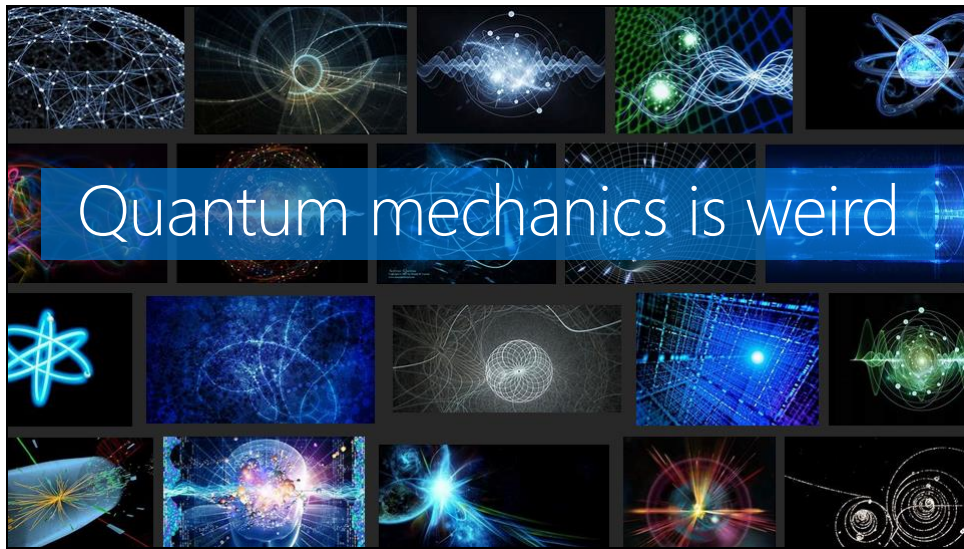
QUANTUM COMPUTING

- What is quantum computing? Why is it important?
- How do we build a quantum computer?
- What kinds of physical systems are already in development?
- What is Q#/the QDK and why would you use it?
- Can we use Q# to teleport a quantum state?

You've probably all heard of this quantum computing thing and are curious to learn more.

Good job, you're all in the right place ☺

I hope that you will all leave this session with a high-level understanding of what quantum computing is and why it's important. We will find out what conditions a physical system must satisfy in order to qualify as a quantum computer and touch upon some systems currently in development. We will end with a whirlwind introduction to Microsoft's Quantum Development Kit and Q# language, as well as some suggested learning resources so you can continue your quantum journey. All of this will be done from the ground up, no programming or significant physics/maths background required! I'll try to use analogy where possible to explain the weirder concepts – try not to take these too literally!



Quantum mechanics is weird. All right, I said it. That's why when you put quantum into Bing (yes Bing :P) Image Search, you get all these weird, abstract graphics. It's really hard to visualise!

At its most basic, quantum mechanics is just the science of really tiny things that act weirdly because they're so tiny. We as human beings exist on a macro scale and our understanding of the world is heavily influenced by this fact – actions have consequences, effect follows cause and everything tends to act quite sensibly, all things considered. When we start to look at things on a quantum level however, this predictable pattern of cause and effect breaks down and we descend into a world of uncertainty which is really hard for us to understand because it seems so at odds with our everyday experience.

Let's take, for example, these stickers I have here. Now if I were to leave them on this table here and come back ten minutes later, I'd expect them to still be there (and if not, I'd assume you guys nicked all of them!)

Now imagine we've been shrunk down to the size of an atom (so we are now small enough to notice quantum effects). Now when I put the stickers down and leave, I put them into a superposition of all the possible places they could be in the lecture theatre (more on this later). When I come back and have a look to see where they are, their wavefunction (the thing that describes their current state) collapses instantaneously down to one option (state), and I find my stickers again, but they could now be anywhere in the lecture theatre! Even weirder, they could even be found *outside* the theatre, even if I removed all the doors and windows so there was no way for them to get out!



Now the really upsetting thing (for scientists at any rate) is that *we have no idea why all this stuff happens*. We can measure and predict the effects and come up with theories and maths that explain our observations in experiment, however there's currently no single theory for why this weird stuff happens in the first place that everybody can agree. There are all sorts of explanations ranging from the sensible to the bizarre for why these things work like they do, but so far we haven't been able to prove any of them! Instead, we have many different interpretations of quantum mechanics and physicists worldwide are working to figure out which is the right one!

Examples include things like many world theory where every time a quantum state collapses, a different multiverse pops into existence for each possible outcome and we happen to exist in the one with our particular result. This was Stephen Hawking's favourite version of events actually.

Hidden variable theorem is another interpretation – this basically says that there is a way for us to predict with 100% accuracy how quantum states evolve over time, just like in our macro universe of cause and effect. The problem is (apparently), we haven't yet been able to discover the hidden variables and equations that would enable us to do these predictions and therefore are reduced to explaining things in this weird, probabilistic manner.

There are too many interpretations to go through them all here sadly, not to mention this strays dangerously close to the realms of philosophy! We'll stick with the current most widely accepted interpretation which is called the Copenhagen interpretation of quantum mechanics and basically says that things don't have definite physical properties until we measure them and collapse the wavefunction to just one of the options – reality is all in the observation. This is what we have been discussing so far.

With weird interpretations like those knocking about, it's no surprise that we end up with some very strange misconceptions about how quantum physics could affect our futures – sadly though, we aren't hurtling into a Rick-and-Morty-style future littered with portal guns and time travel and different dimensions. Star Trek style teleportation and faster-than-light travel/communication aren't going to happen (sadly, but good news for the space-time continuum!) – at least, that's what we currently think given our understanding of physics today (that's the beautiful thing about science – this could all change in the future as new discoveries are made!)

I mean, physicists are just as excited by all this stuff as you are, but sadly we can't break science to bring it to life ☹

We can still do some pretty awesome (and still mindblowing) things with quantum though, don't worry! Let's have a look at some examples.

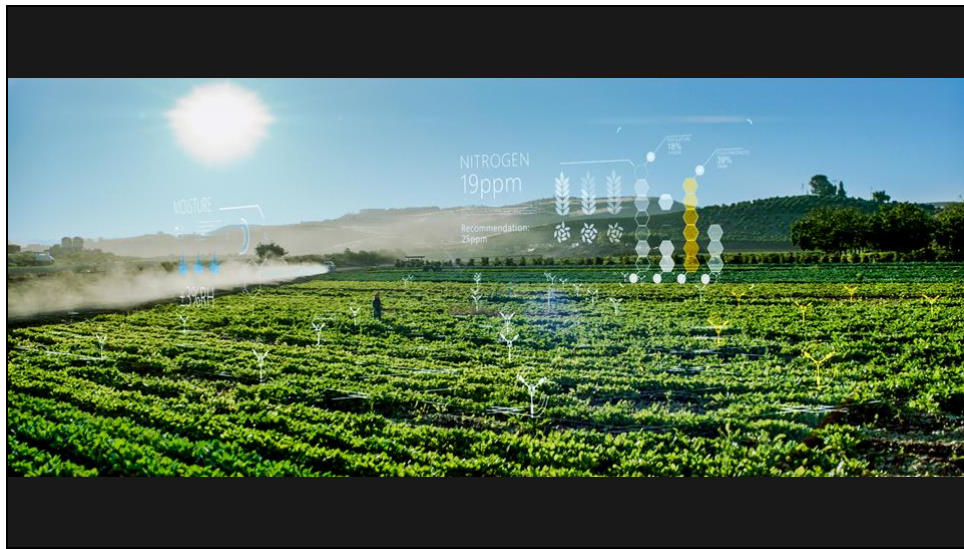


- First three on the left – achievable through accurate simulation of materials on a quantum scale.
- ML can touch every facet of our lives. Offload compute-intensive workloads to QC not graphics processor.

Here we see only a tiny selection of the things we could do with the power of quantum computers – the first three are based on using quantum computers to simulate hugely complicated quantum systems such as proteins and new materials we could use for batteries or construction!

Machine learning is a great example of a field that stands to gain hugely from the of quantum computing, providing our data scientists with vastly more processing power than they currently have to hand – instead of offloading compute-intensive workloads to graphics processors, in future they could offload them to quantum processors! Machine learning has the potential to touch every walk of life so you can see even from this limited selection of applications that the potential reach of the technology is huge! And this stuff is barely scratching the surface!

But let's take the example of nitrogen fixation and elaborate a little further, to give you a better idea of where we are coming from.



- 80% of atmosphere. Very stable
- Proteins
- Organic fertiliser (manure, decaying organic matter)
- 1910, Haber & Bosch
 - 400 degrees Celsius
 - 200 atm
- 4% of global energy output annually – must be a more efficient way!

You might not know it, but nitrogen fixation is actually pretty crucial to our continued existence as a species – nitrogen is one of the key ingredients that makes up the proteins that we (and everything else on the planet) are made up of! Nitrogen is therefore a key component in farming – plants need plenty of it in order to grow healthily.

Back in the day before industrial farming was a thing, farmers used to use organic fertilisers (like manure) to enrich the fields with ammonia. Ammonia is a good fertiliser because it contains nitrogen in a form that's easily broken down and reused by plants. However, production of organic fertiliser is slow and inefficient, no matter how many cows you put to the task!

This is where Haber and Bosch came to the rescue in 1910 with the invention of a new industrial process for producing ammonia – this came to be known as the Haber process and it unblocked the way for industrial-scale farming as we know it today. More than 100 years later, it's still the most commonly used method for nitrogen fixation. However the picture isn't all rosy – this process requires huge amounts of energy to maintain the high temperatures (400deg) and pressures (200atm) required for the reactions to take place. In fact, 4% of all the energy produced globally every year goes to this process (about all the

energy produced using nuclear sources in 2015). Think about that for a second – 4% is *huge*. Surely there must be a more efficient way of doing this!

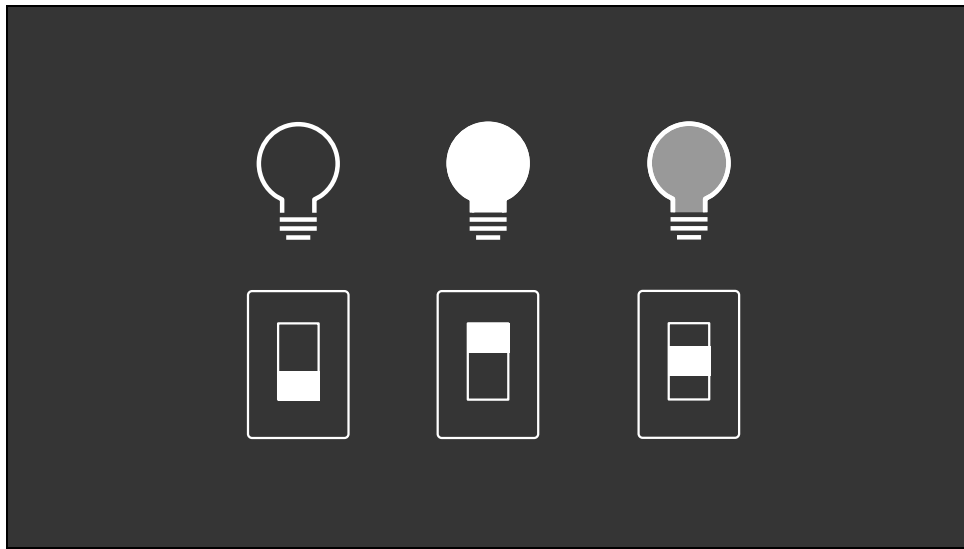
<https://www.google.co.uk/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&ved=0ahUKEwiinK7TveTZAhVEAcAKHbGIAesQFgg5MAE&url=https%3A%2F%2Fwww.worldenergy.org%2Fwp-content%2Fuploads%2F2016%2F10%2FWorld-Energy-Resources-Full-report-2016.10.03.pdf&usg=AOvVaw06YyeK-LmUrQBgE2XLgGNf>

- Turns out there is! Farmers for centuries have been planting beans to restore the nitrogen

Unfortunately for us, we have no idea how nitrogenase does its thing, other than in the most

Quantum computers, however, could do this simulation using as few as 170 qubits (more on

Quantum computers, however, could do this simulation using as few as 170 qubits (more on these later!) – to do this on a regular computer, you would require every atom on ten Earths (10^{50} atoms) acting as memory for our simulation (ain't gonna happen!)

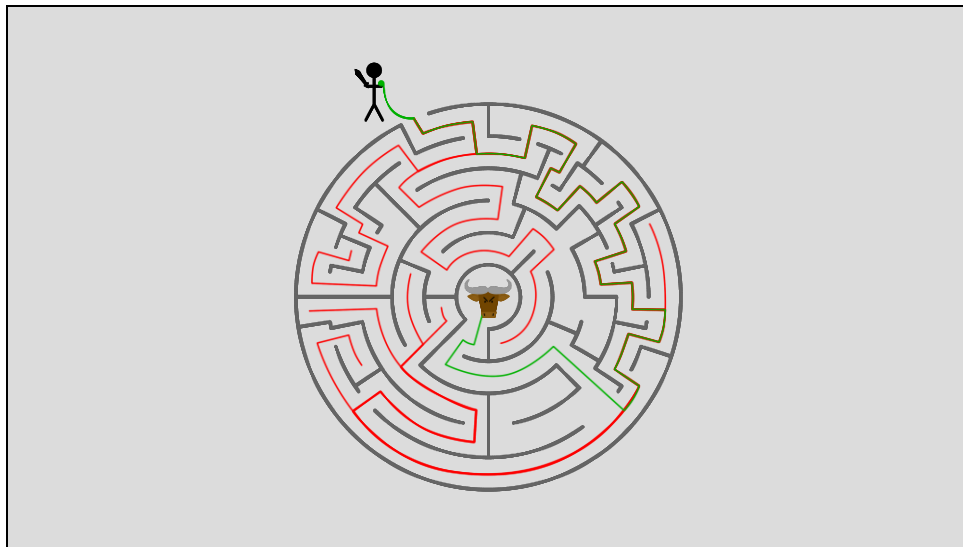


So why are quantum computers so special then? What makes them so powerful?

Two of the key differences between a quantum computer and a regular, classical one are the concepts of superposition and entanglement. We'll start with superposition.

The bit ('binary digit') is the smallest possible unit of digital data. Classically, bits can only take one of two values at any one time: 0 or 1 (off or on, like our two light switches on the left here). A quantum bit (or qubit) obeys the laws of quantum mechanics however, and therefore can exist as a combination (known as a superposition) of both states 0 and 1 simultaneously (kind of like the dimmer switch, or my cup of tea from earlier!).

It is the act of measuring the qubit that causes it to adopt either the 0 or 1 classical state at the end of our computation – this is known as 'collapsing the superposition'.



Let's illustrate this with an analogy (pinch of salt)

Hopefully you're all familiar with the story of Theseus and the Minotaur. If not, to cut a long story short, our hero Theseus must navigate a maze, find the murderous half-man, half-bull Minotaur that dwells at the centre, slay it, and then find his way back out of the maze and home to freedom. He is equipped with a sword and a ball of yarn, which he ties to the gate of the maze so that he can find his way back out again.

So our hero Theseus knows that there is a correct route out there with the Minotaur at the end and his job is to search until he finds it!

Unfortunately for him, he's a classical, macroscopic being and therefore must try each route in turn before he finds the one he is looking for. He could of course get lucky and find it on the first attempt, or conversely he could try all the other paths first before finding the right one. By law of averages, he'll have to try roughly 50% of the routes before he gets the right one. This is slow.

- Left/right

Obviously, he doesn't know the way into the centre where the monster lurks – he has to use trial and error until he can find the correct route.

Let's assume Theseus is a macroscopic being living in our regular reality (no God powers for him). His behaviour can therefore be described in terms of classical physics (not quantum) – he is forced to try route after route until he locates the monster at the centre of the maze. Each route is a sequence of choices – left or right.

This is analogous to how a classical computer works
He'll have to try roughly 50% of the routes unless he gets lucky.

In this quantum world, when he enters the maze he is transformed into a superposition of

Now normally, this would just produce a random state, be it right or wrong. However, we

CLICK

overall (superposed) state $|\psi\rangle$ possible qubit states $|0\rangle$ and $|1\rangle$

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$$

probability amplitudes α and β

Probability of seeing $|0\rangle$ when measured: $|\alpha|^2$

Probability of seeing $|1\rangle$ when measured: $|\beta|^2$

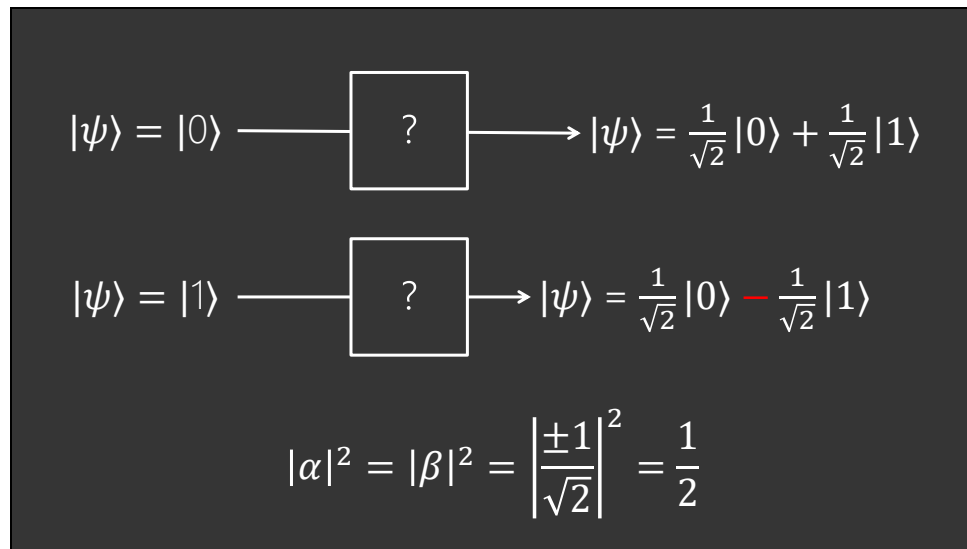
Total probability of seeing either $|0\rangle$ or $|1\rangle$: $|\alpha|^2 + |\beta|^2 = 1$

So, how do we represent this information? Where regular bits are concerned, we simply label them as 0 or 1 and get on with our lives. In the quantum case, this is not so simple! This is where Dirac's bra-ket notation comes in handy – we represent our qubit states as ket zero and ket one (kets are those lopsided brackets pointing to the right), and our superposition as a linear combination of the two, as shown here.

You'll notice that factors of alpha and beta have suddenly appeared in the superposed state. These are known as probability amplitudes (as opposed to classical probabilities). We can calculate the classical probability of finding our qubit in either state by taking the squared modulus, as shown here:

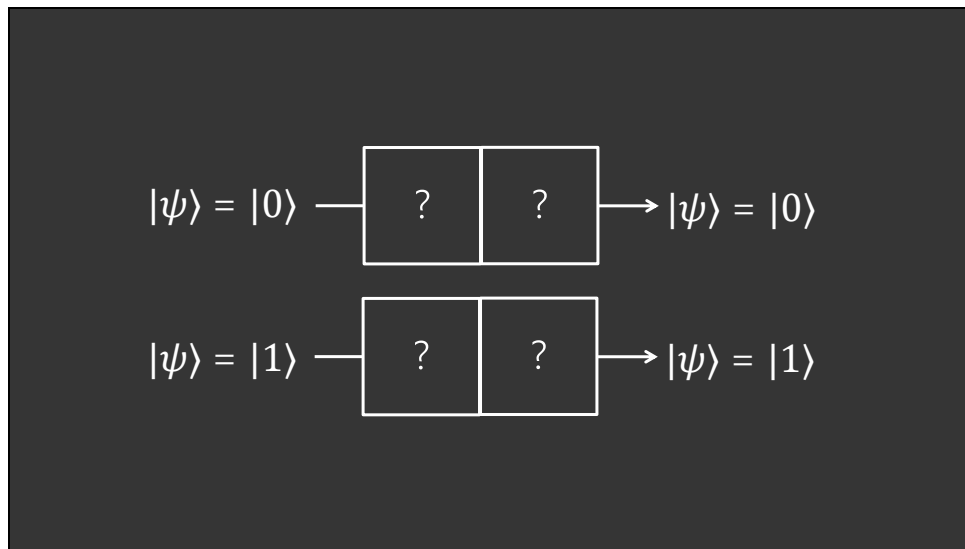
For our superposition to be a valid quantum state, the following relationship must hold true: That is to say, the probability of finding our qubit in either state 0 or 1 after measurement is 100% – i.e. it couldn't be found in some other, totally random state, which makes sense! You can kinda think of these probability amplitudes as a measure of the 'oneness' and 'zerness' of the state – the bigger the modulus of the amplitude, the more likely we are to find our qubit in that state when we measure it.

Now, here's the cool bit: when we think about classical probabilities, we think of percentages or fractions – we say, for example, that a fair coin has a 50% chance of landing heads-up. Classical probabilities are always real, positive numbers. Probability *amplitudes*, on the other hand, can be positive, negative, or even imaginary, unlike our regular classical probabilities. We can use this to our advantage by introducing the concept of constructive and destructive interference – we can arrange our superposition in such a way that all the unwanted probability amplitudes cancel out, leaving us with only our desired state.



Let's say I have a magic box that does the following: when I put in a qubit in the zero state, it creates a superposition like we see on the left, and when I put a one in we get a superposition like the one seen on the right (note the negative probability amplitude). You'll notice that the coefficients are the same for both the zero and one states in both cases illustrated here – when we apply the maths we see that we have a 50% classical probability of measuring and finding a one or a zero if we measure this state.

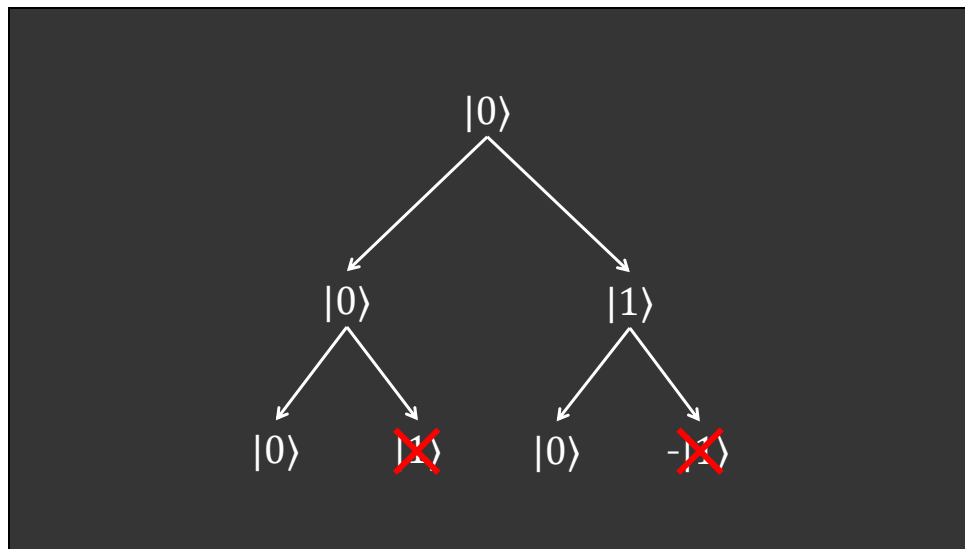
So what happens when I put my state through the box twice, without measuring in the middle?



So what happens if I put my original state through the box twice, without measuring in the middle? Weirdly, we always get the original state back!

NOTE: NOT ALWAYS THE CASE – depends on the type of box we put our state through 😊

But how can this be if all we've done is put it through the magical superposition box twice? Surely we should just get a random result back! The cause of this effect is actually an example of *quantum interference*. Let's find out how this works :O



This is normally where I'd pull out a lot of maths and get cancelling but let's start with something a bit friendlier, shall we? 😊

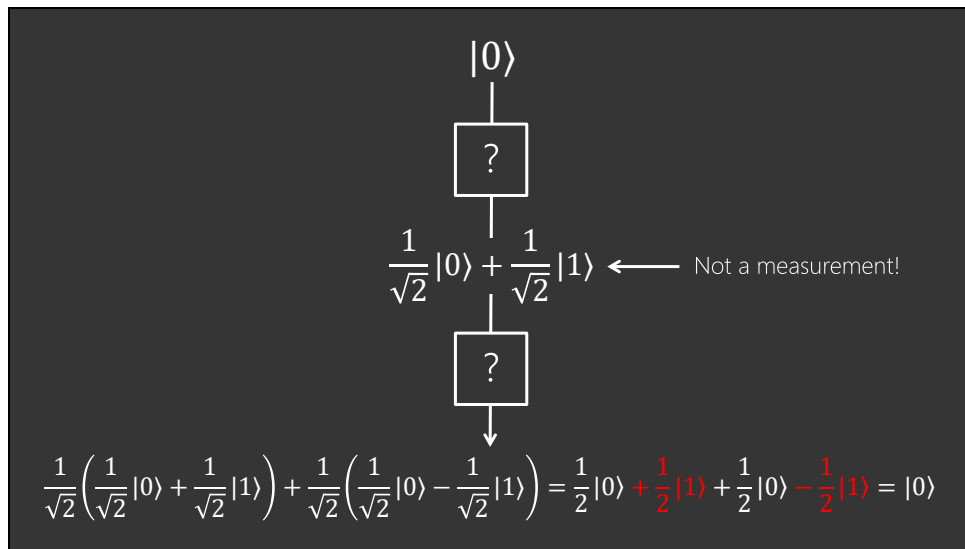
Let's stick with the case where I start with a qubit in the zero state, like we see here at the top. First, we apply our magic superposition box and get a state that's 50% likely to give us a zero or a one, as seen in the previous slide (we're not measuring and therefore collapsing the superposition though).

Everything on a particular layer has equal probability amplitude other than in sign which is noted on the diagram.

Our quantum superposition has *interfered with itself* to give us back the zero state we started with!

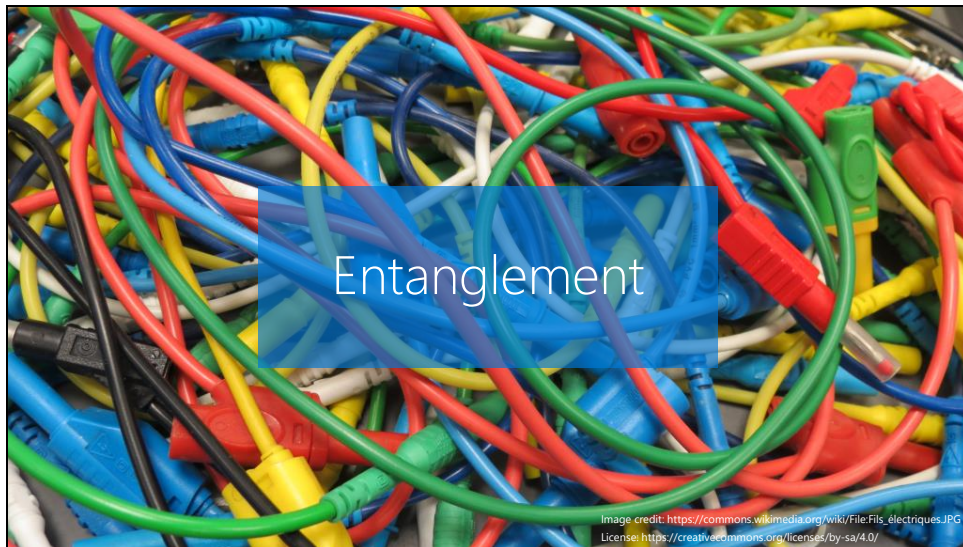
Really simple example of how we use quantum interference, but you can see how this might come in useful when we start adding more qubits!

You can see from this diagram that each time we add a new qubit, we double the degrees of freedom we have to play with – this is exponentially more information processing power than the classical case where each new bit simply adds one new



Just in case you were wondering, this is what the maths looked like!

Hopefully that made sense – Frances will be coming back to our magic superposition box (also known as the Hadamard gate) a bit later...



Now, onto entanglement! Sounds weird, and it is!

This is the phenomenon that Einstein once famously referred to as ‘spooky action at a distance’.

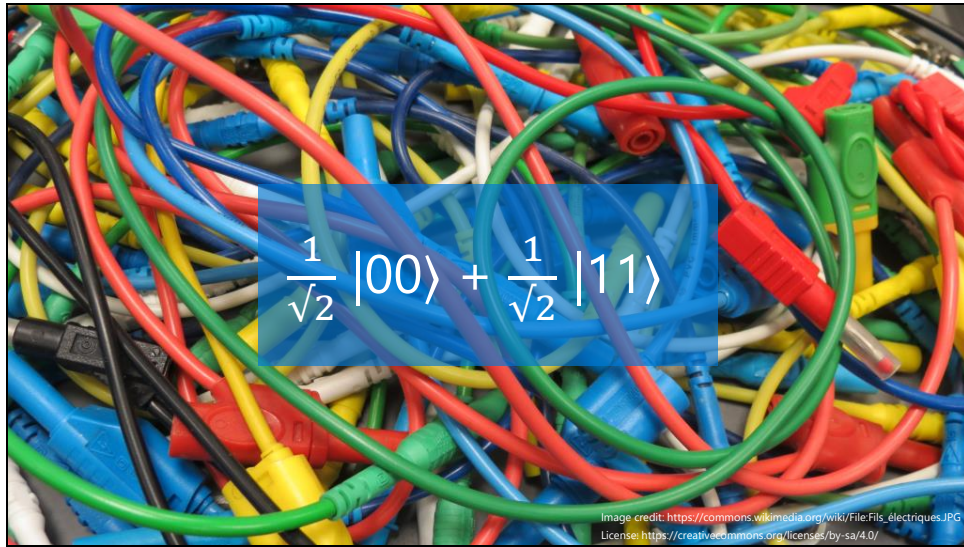
Let me try to explain. Say a friend and I get together for a science party one day and decide to create a pair of entangled qubits. We do this and go back to our respective lives the day after, keeping hold of one qubit each. A while later, I get a job as an astronaut and fly to Mars! I still have my qubit and one day decide to measure it and see what state it’s in. This collapses the superposition, leaving me with a qubit in a single state (let’s say 0). Assuming my friend hasn’t done anything weird with her qubit to break the entanglement, this causes the state of her qubit to collapse to the state corresponding to mine (1) *at the exact same time that I measure my qubit*. I now know exactly what state my friend’s qubit is in, despite not communicating with her.

Let’s hear that again – it’s pretty mindblowing. (use fists) This is my friend and this is me. We meet up and create an entangled pair of qubits. I then travel to a different planet. When I measure my qubit, I find it is in state 0. *Simultaneously*, back on Earth, I know that my friend’s qubit has collapsed to state 1.

Now you wouldn’t be blamed for thinking that this means we can suddenly start communicating across infinite distance with zero latency (would make Skype calls smoother, for sure!), or travelling faster than the speed of light! Our qubits *have not communicated*. Their states were *correlated* when we entangled them which means a measurement on one will also cause the other to collapse to the corresponding state. We couldn’t, for instance, force my qubit into a certain state (say 1 means yes), then measure it to cause my friend to

see 0 and know my answer from the other end of the universe. Let me illustrate with an analogy:

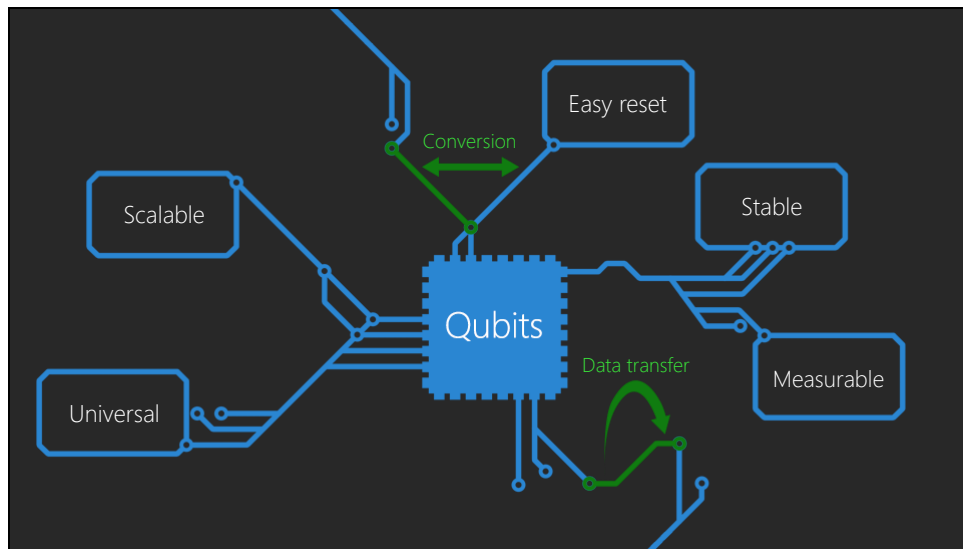
Imagine I have two ping pong balls. I paint one of them red and one of them blue. I then mix them up behind my back and put each into a separate box – I don't know which ball is in which box. I then mail this box to my friend, who lives in Australia. Now, if I open my box (perform a measurement) and see a blue ball, I know that my friend has the red, and vice versa – the results are correlated. The balls have (obviously) not communicated with one another. This holds true regardless of whether my friend lives in Australia or on Pluto. What I *can't* do however is paint my ball red and turn my friend's one blue. You can think of entanglement in the same way (to a point, it's only an analogy after all!)



As far as our Dirac notation goes, it's fairly straightforward to tell when a state is entangled because there's no way to factorise the state so that each qubit is isolated.

Take, for example, the state we see here – we aren't able to factorise this so that we get two brackets, one describing the first qubit's state only and one describing the second. It is therefore impossible to describe the state of one of the qubits individually without also describing the state of the other one – this is the more mathematical definition of entanglement and it's why our measurements are correlated.

FYI – not limited to just two qubits!



So now we've learned what makes quantum computers so special (and weird), let's find out what we need to do to build one. There are five different things a system must do before we can count it as a proper quantum computer, plus two extra ones for communication between qubits to happen.

- **Easy reset:** we can always reset our system to a simple, well-known starting state, i.e. we know that we have a red and blue ping pong ball only at the start, no green ones lurking!
- **Stable:** Because qubits are incredibly tiny, they can be incredibly sensitive to perturbations in the local environment. This can come in the form of errant cosmic rays, temperamental equipment or just bad luck, but the long and short of it is that quantum states tend to be short-lived. We basically need to make sure that these states exist for longer than it takes us to actually use them for computation (kinda makes sense when you think about it really!) This process of losing the quantum properties of the system is also known as decoherence.
- **Measurable:** Fairly obvious this one – basically we need to be able to actually measure our qubits at the end to see the result of our computation.
- **Scalable:** In just the same way as our regular computers make use of more than one bit (a one-bit PC would be a bit depressing really), we must also be able to add more qubits to our system so we can do more complex computations.
- **Universality:** When we do complex computations, we don't just input the maths directly into our system and press play! Instead, we must break these complex operations down

into sequences of simple steps that our computer is able to perform. We call each of these steps an operation, and each operation is performed by a gate.

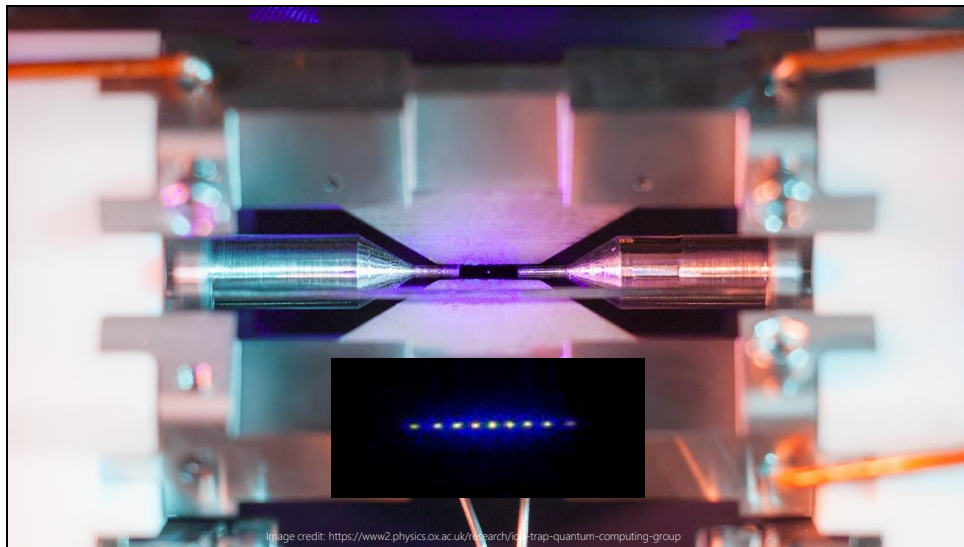
- A universal set of gates is just a collection of gates that can be combined to perform any computation we want on the system.
- The idea here is that we should be able to physically implement at least one of these sets, otherwise we have a computer limited to only doing specific things, which seems silly!

Now we come to the two communication requirements. These two are pretty straightforward:

- We have to be able to convert between stationary qubits (used for computation) and moving ones (used for communication).
- We have to be able to reliably transfer information between parts of our quantum computer (i.e. reliably move qubits between different parts of the computer)

Sweet, now we know what we need to do to build a quantum computer! Not so complicated after all 😊

Let's find out what systems people are already building!



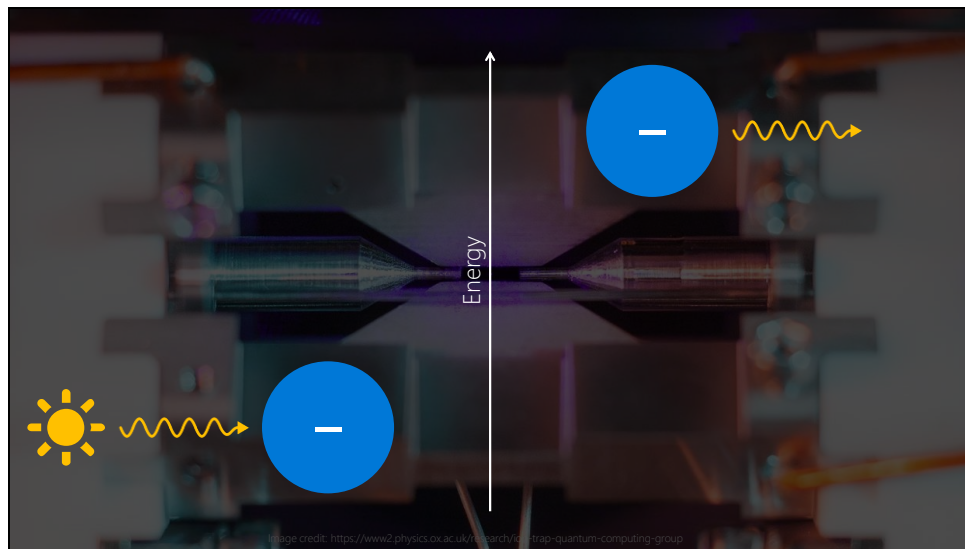
One type of qubit that various groups are currently working on is based on ions. Ions are basically regular atoms that have lost or gained an electron/electrons and therefore have a charge (positive or negative). Because they have charges, we can use magnets and electrodes to trap and control them, like we see in this rather dull schematic. Let's see what a real ion trap looks like!

CLICK

Much shinier! What we are seeing here is the trap itself – if you look closely, you can actually see a single strontium ion trapped between the two needle-like electrodes we see coming from the left and right! The photo was taken by using a really long exposure shot so that we can pick up the very faint fluorescence of the strontium ion – this particular ion trap is currently being developed by the Ion Trap Quantum Computing Group in Oxford.

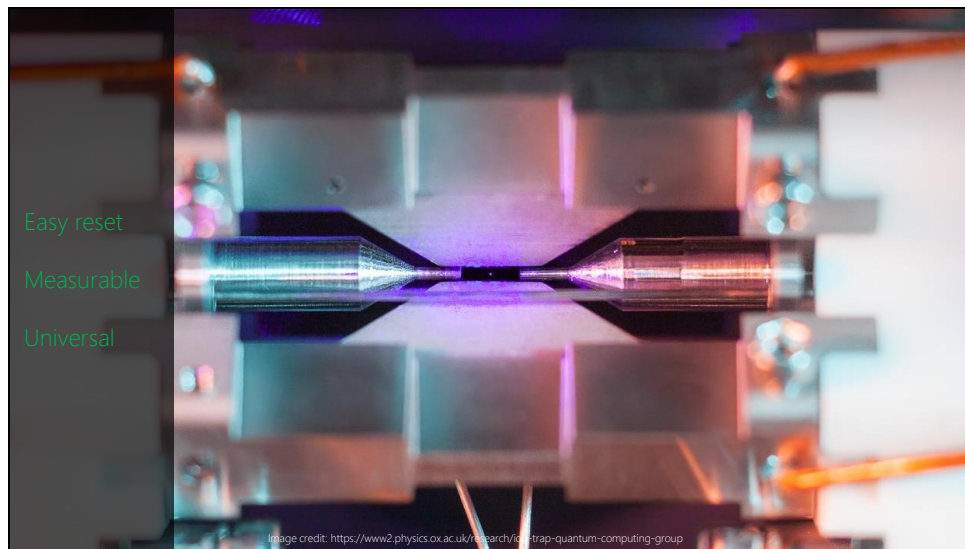
But one ion isn't really going to cut it, is it? Fortunately, they don't stop there 😊 At the top, we can see a string of such ions held in a trap – now we can start to do some actual computation!

<https://www2.physics.ox.ac.uk/research/ion-trap-quantum-computing-group>



Ok so how does this all actually work then?

In an ion trap system, we use the energy levels of the ions as our qubit 0 and 1 states – our ions are set up to have two accessible levels (of lower and higher energy). When we shoot light at it, the ion absorbs energy and becomes excited – this is our '1' state. It can then relax, releasing light (which we can detect) and settling back into the lower energy state (this is our '0' state).



So how does this system stand up to our requirements from earlier then?

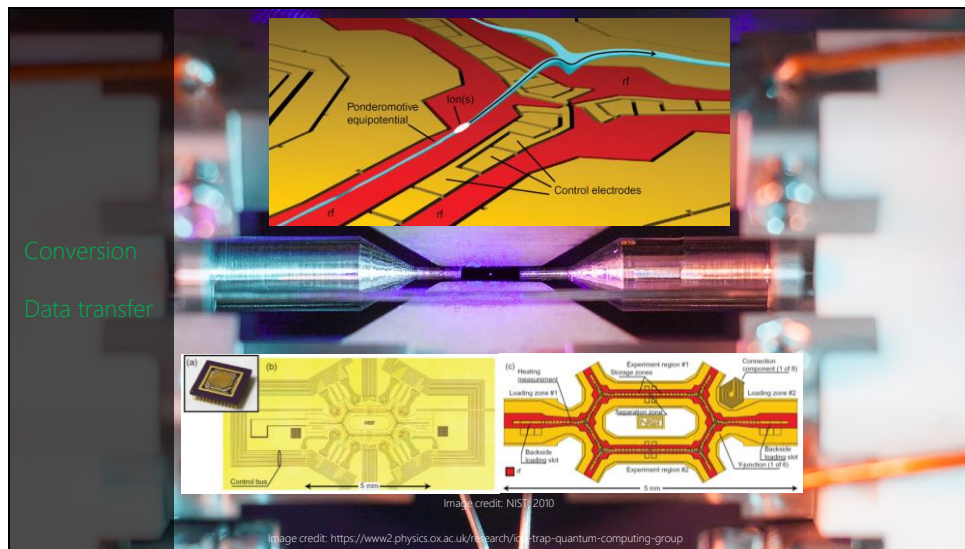
Pretty well actually!

Ions are pretty easy to get hold of, manipulate and measure, meaning they satisfy our requirements for measurement, easy prep & reset. It has also been proved (theoretically and experimentally) that we can make all the gates we need for universality. Great!

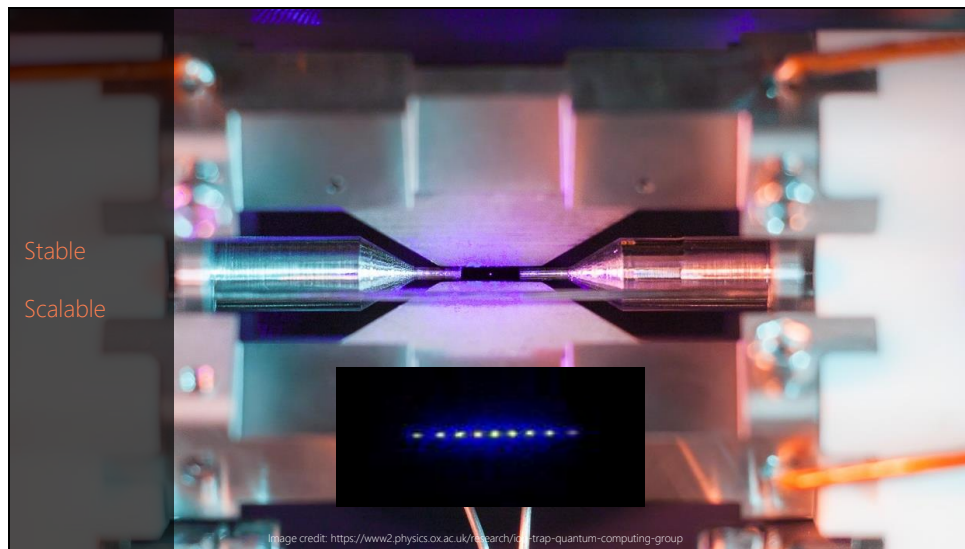
We can use further magnets and electrodes to move ions about the system, fulfilling our communication requirements (conversion and data transfer).

Unfortunately for us however, ions are still pretty sensitive to perturbations from outside and isolating them properly from these effects is quite hard (but not impossible!) Furthermore, whilst great leaps have been made in realising the dream of the ion-trap-on-a-chip, there is currently no truly large-scale implementation of the ion trap system available for computation, so we aren't quite there with scalability yet either! Very promising area of research though, watch this space! 😊

This is only one example of a physical system for quantum computing though – and there are many, believe me! Some of them make use of neutral atoms in a similar way to the ion trap (using light to trap the atoms instead of magnets and electrodes). Some (such as NMR-based methods and quantum dots) use the spin of electrons or atomic nuclei (spin can be either up or down, mapping rather conveniently to our zero and one states). Each of these different systems has its own advantages and disadvantages, but we won't go into the details here today (we don't have the time, sadly!) I'd just like to cover off one more example which is quite different to the ones I've spoken about thus far – topological quantum computing.



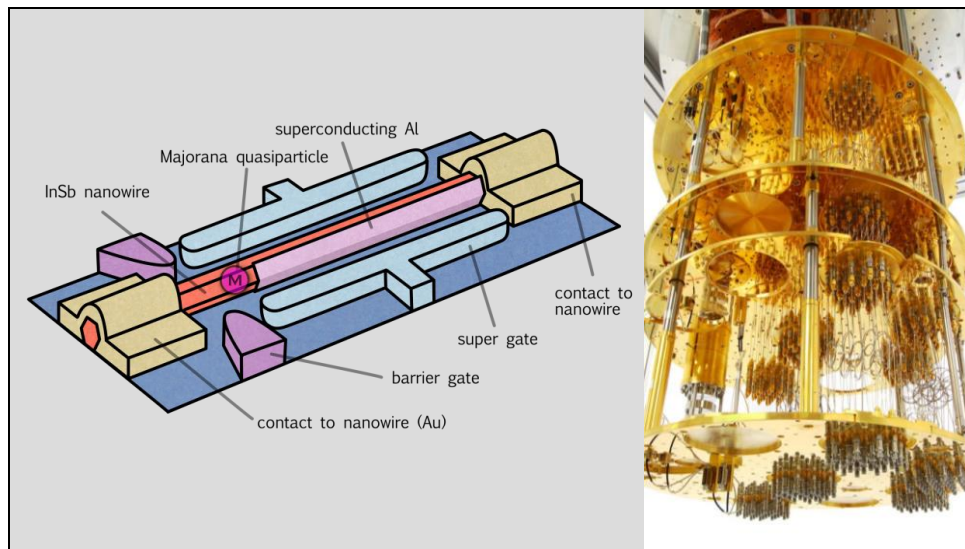
We can use further magnets and electrodes to move ions about the system, fulfilling our communication requirements (conversion and data transfer).



Unfortunately for us however, ions are still pretty sensitive to perturbations from outside and isolating them properly from these effects is quite hard (but not impossible!) Furthermore, whilst great leaps have been made in realising the dream of the ion-trap-on-a-chip, there is currently no truly large-scale implementation of the ion trap system available for computation, so we aren't quite there with scalability yet either! Very promising area of research though, watch this space! 😊

This is only one example of a physical system for quantum computing though – and there are many, believe me! Some of them make use of neutral atoms in a similar way to the ion trap (using light to trap the atoms instead of magnets and electrodes). Some (such as NMR-based methods and quantum dots) use the spin of electrons or atomic nuclei (spin can be either up or down, mapping rather conveniently to our zero and one states).

Each of these different systems has its own advantages and disadvantages, but we won't go into the details here today (we don't have the time, sadly!) I'd just like to cover off one more example which is quite different to the ones I've spoken about thus far – topological quantum computing.



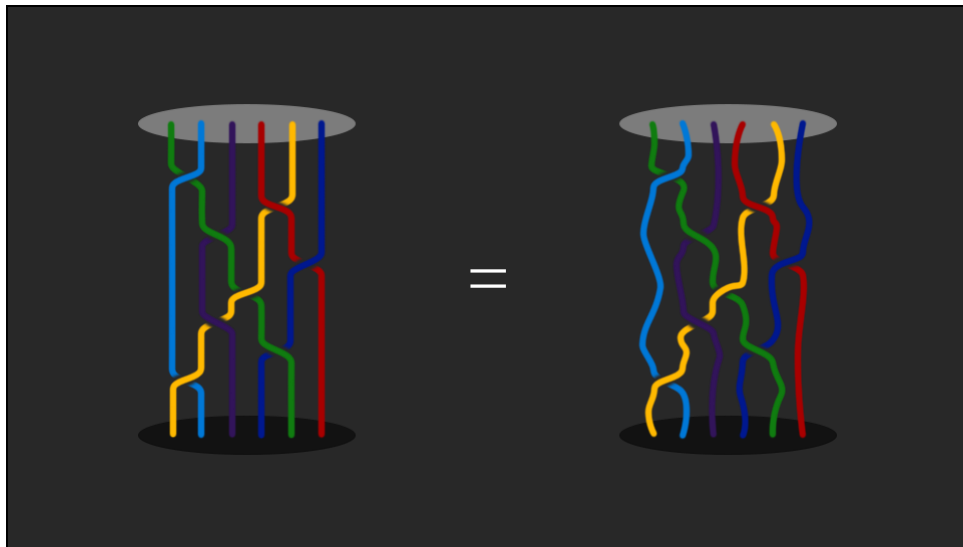
Soo topological QC – what on Earth does that mean?!

Basically what we do is take a nanowire and cool it down to just a few millikelvin above absolute zero (nearly -300 degrees Celsius) – our lab is quite possibly the coldest place in the universe right now!! At these super low temperatures, the electrons in the wire start to behave strangely and exhibit properties consistent with Majorana Fermions. The image on the right is the ‘quantum fridge’ used to lower our system to such a cold temperature.

Majorana Fermions are exotic quasiparticles whose antiparticles are the same as their particles. What happens here is that we end up with two of these Majorana Fermions, one at either end of the wire. We are then able to manipulate each half of the pair independently in order to perform computations (more on this in a sec).

When we bring the particles together, they collapse down to either an electron or the absence of an electron – this gives us our two-state qubit system!

So how do we actually use these Majorana Fermions to perform computation and why would we want to use this topological approach instead of a more traditional method such as ion trap QC? The answer lies in the way we encode information in the system.



- Now, imagine we have several of these pairs of Majorana particles together in our system. Because we can move each of the two particles independently around the wire, we are able to effectively 'braid' these particles into a pattern like we see here
- Can use this braiding scheme to represent quantum information and perform computations
- More braids = higher accuracy
- Because we're not using properties of the particles themselves and rather use the braid pattern for computation, this is much more robust than other schemes – need far fewer error correction qubits
- As you can see on the right, topological qubits can undergo a lot of interference before they are rendered useless
- Up until quite recently, has been quite theoretical, however we've very recently published a paper in Nature offering significant experimental evidence for the existence of these Majorana Fermions, which is hugely exciting.
- Our research team is focused on designing a system that is robust, universal, and highly scalable – now we've proved the particles actually exist, scale-up work can commence! Watch this space 😊
- See future blog post for more details
- Now, over to Frances who will pick up where I left off with the software side of things! 😊

Soooo, topological quantum computing. What on Earth does that mean?! Good question!

To cut a very long (and complicated!) story short, topological qubits are made of exotic quasiparticles called Majorana Fermions that we can braid together to form qubits. The more anyons we add to the braid, the more accurate our computations get. We do computations by changing the way the anyons are braided together and creating knots.

At this point, you'd be forgiven for asking why on Earth we are spending time and energy looking into this when there are already so many other options out there that use particles that we actually know exist. There is, of course, an excellent reason to do this, and it's stability.

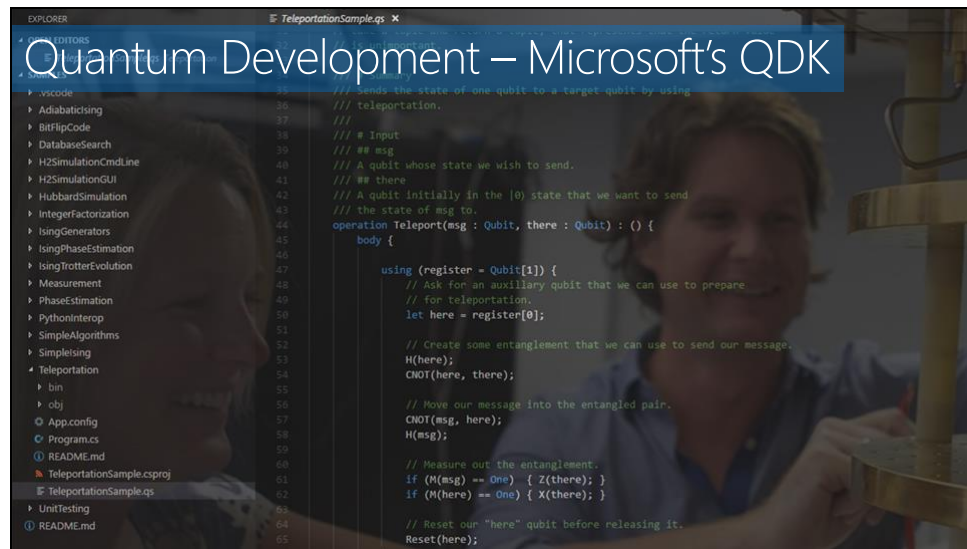
As we've come to learn, our qubits need to survive long enough for us to actually do computations with them (the stability requirement). In regular systems, our qubits are represented by the properties of relatively easily perturbed particles such as ions, electrons, atoms and molecules. This means that we need a whole bunch of backup qubits sitting around in order for us to do error correction when a qubit gets disturbed by the environment, dramatically increasing the number of qubits needed in total to do our computations.

Anyons are much more resilient to perturbations than regular particles however, because we don't use the particles themselves to encode information, but rather the order in which we swap them (known as 'braiding' because of the way the paths look, see the picture on the slide). As you can see from the above, even when these particles become disturbed, the braid order doesn't change and therefore our qubit state is preserved. Long story short, this means that they are much less likely to be disturbed by the environment and therefore we need far fewer qubits to produce a system with low enough error rates to be useful.

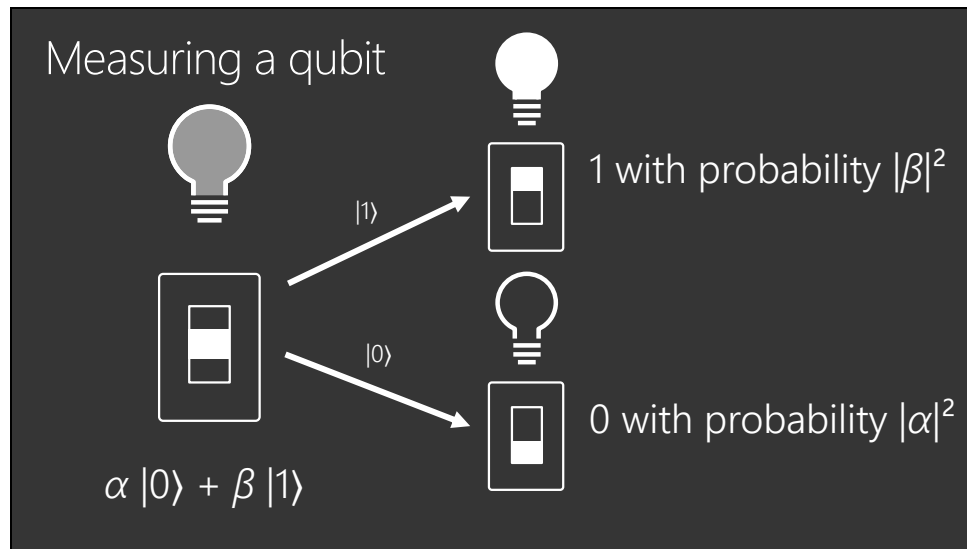
There is a much deeper discussion to be had here but it's a little out of scope for today. I'll be doing a blog post on this sometime in the nearish future so keep an eye out for that if your interest has been piqued!

<https://www.nature.com/news/inside-microsoft-s-quest-for-a-topological-quantum-computer-1.20774>

<http://www-thphys.physics.ox.ac.uk/people/SteveSimon/overview.html>



- Windows, Mac, Linux. 30 qubits locally, more than 40 in the cloud simulator.
- Libraries are already written, basic operations and not-so-basic

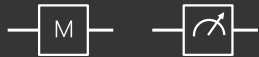


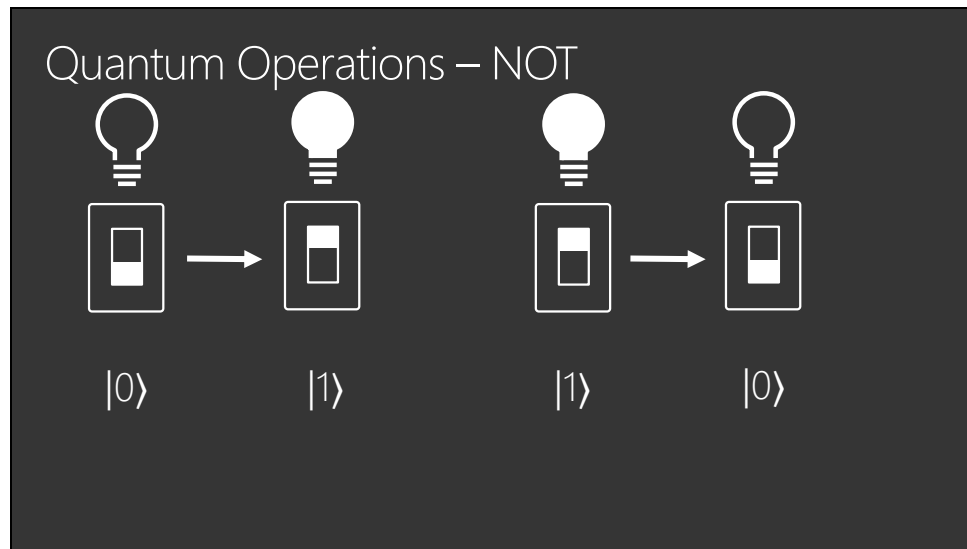
So we've got this qubit state, which we've said can be continuous. So now we want to access that information, we measure the state. The trouble with quantum states, is that when we measure them, they collapse. So that continuous quantum state on measurement will collapse to either one or nought. Which is odd, I know, but we're not done yet. We can say with some probability which state it will collapse to. For the general state α nought plus β one that we've got here. It will collapse to 0 with probability modulus α squared, or to 1 with probability modulus β squared.

But this state isn't general, that I've got here, so I'll show you this example with numbers.

Measuring a qubit

```
operation M (qubit : Qubit) : Result
```





Just as in classical computing we can apply NOT gates, AND gates, OR gates to bits, in quantum computing we can apply quantum gates to qubits. We're going to look at some of these gates, and some of them will look familiar.

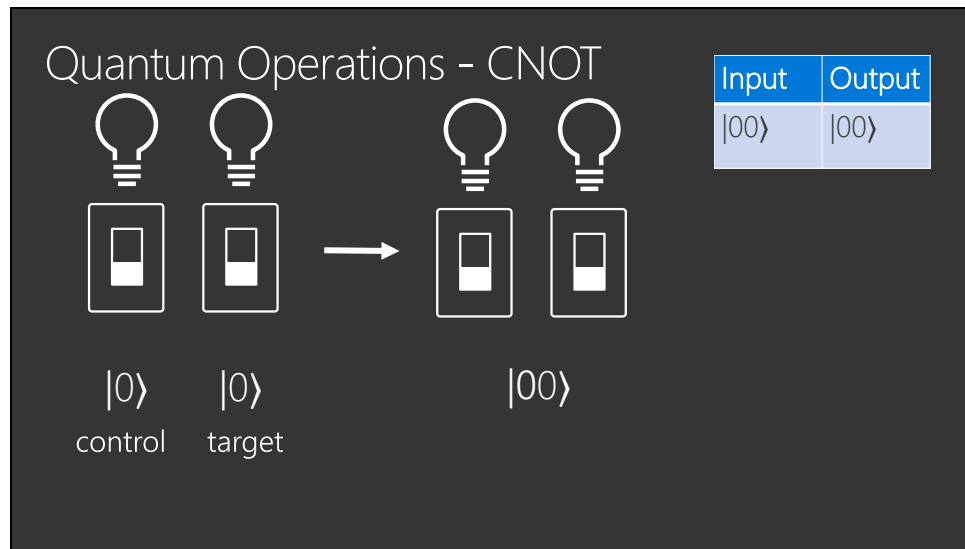
First let's look at quantum NOT. Just with classical NOT it takes 0 to 1, and 1 to 0.

Quantum Operations – NOT

```
operation X (qubit : Qubit) : ()
```

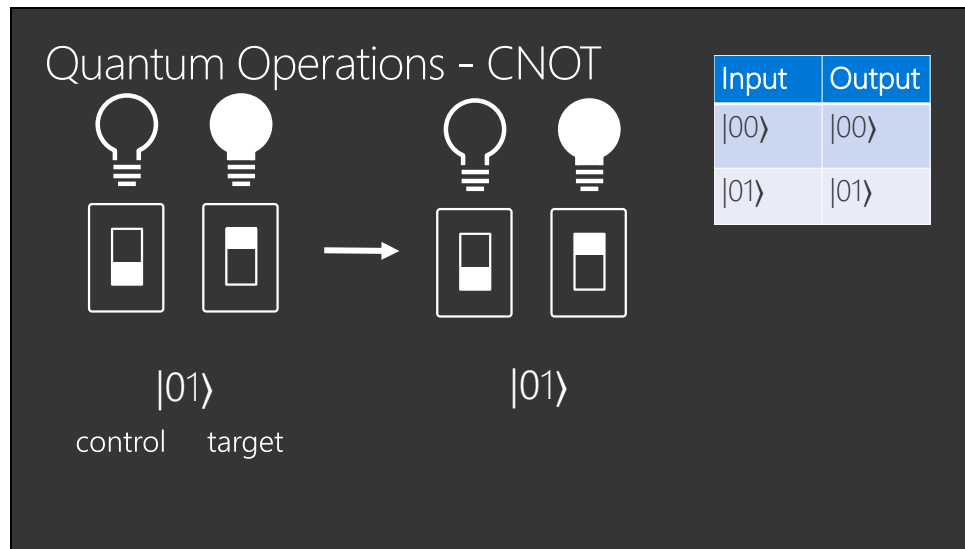


First lets look at quantum NOT. Just with classical NOT it takes 0 to 1, and 1 to 0.

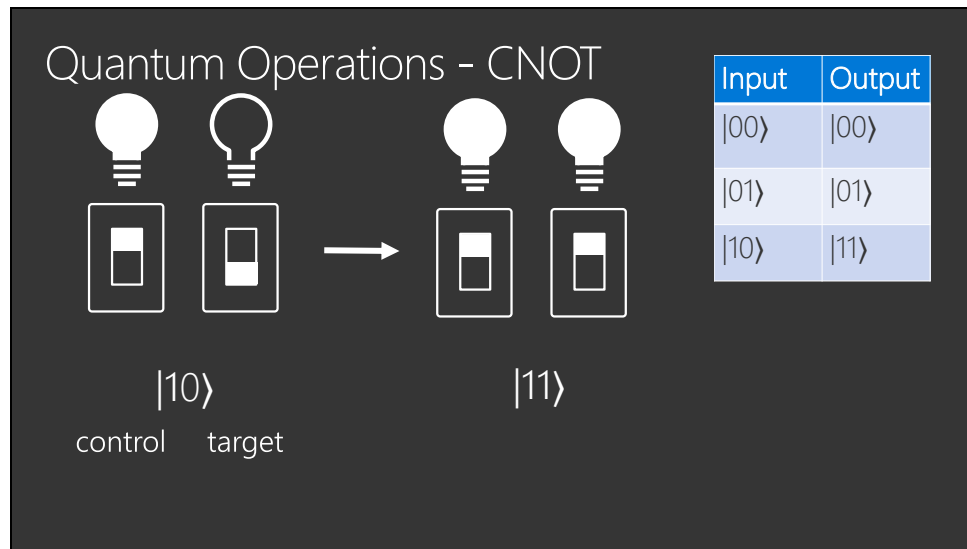


Now we're going to look at the CNOT gate, which means controlled NOT. And it is controlled because we introduce an additional control bit. So say we've got a two qubit state like this. The first is the control bit. And the second qubit is the target. With a two qubit state we can write this more simply like this.

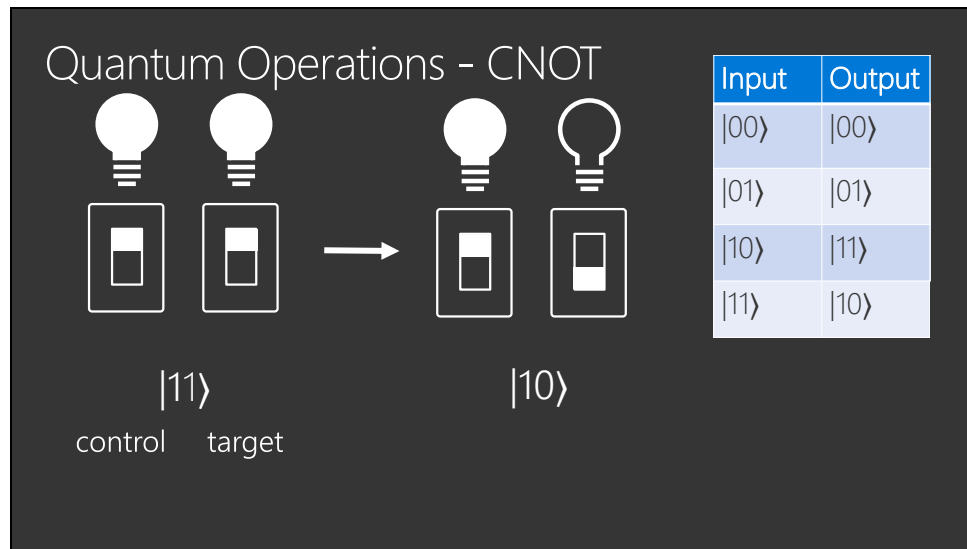
When the control is 0, we do nothing to the target. So the CNOT gate does nothing here.



Let's look at CNOT applied to another state. This time 01 . Again, because the control is 0 , we do nothing to the target.



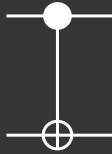
Let's look at CNOT applied to another state. This time 10 . Now here the control is 1 , so we apply a NOT to the target qubit, which takes it from 0 to 1 .



Now we can take the final two qubit state 11, and you hopefully you have the idea by now. The control is 1, so that makes the target qubit 0 like this.

Quantum Operations – CNOT

```
operation CNOT (control : Qubit, target : Qubit) : ()
```



Quantum Operations - Z

$$|0\rangle \longrightarrow |0\rangle$$

$$|1\rangle \longrightarrow -|1\rangle$$

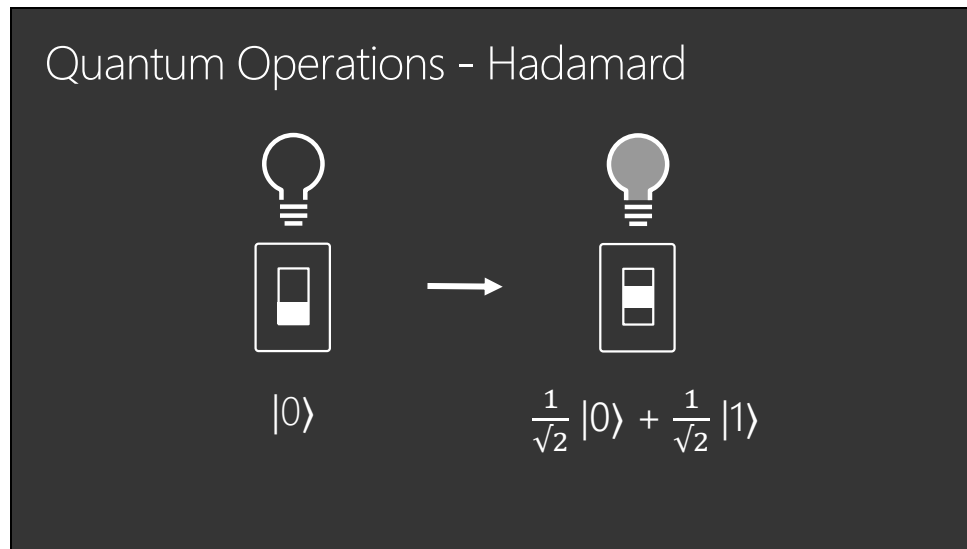
Input	Output
$ 0\rangle$	$ 0\rangle$
$ 1\rangle$	$- 1\rangle$

There's some transformation that I can't demonstrate with lightbulbs. Maybe that's a relief because you're a bit tired of them now. This next one is Z. It's simple enough – it does nothing to 0, and makes 1 negative.

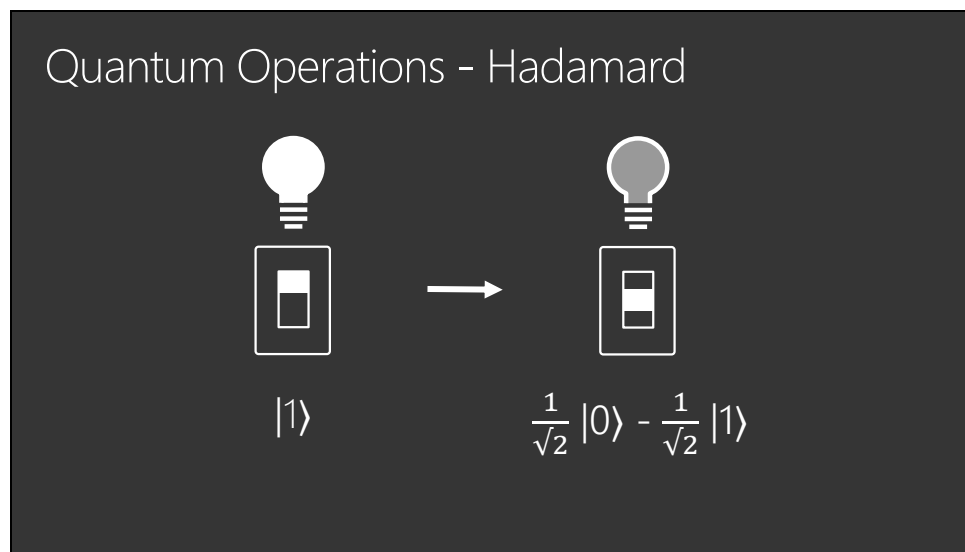
Quantum Operations – Z

```
operation Z (qubit : Qubit) : ()
```





What a Hadamard gate does it that it takes 0 to this state.



And it takes a 1 to a negative half state

Quantum Operations – Hadamard

```
operation H (qubit : Qubit) : ()
```



Our Toolbox:

NOT

Input	Output
$ 0\rangle$	$ 1\rangle$
$ 1\rangle$	$ 0\rangle$

CNOT

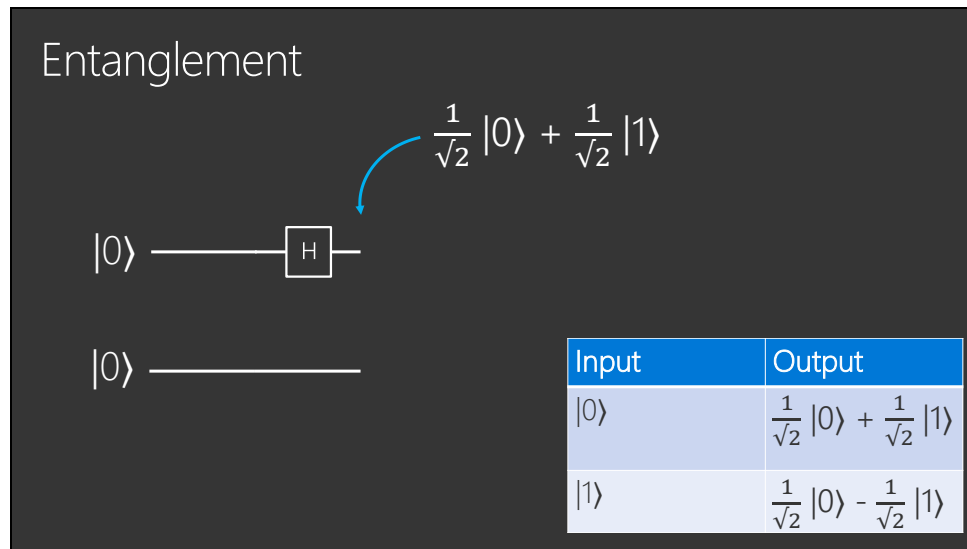
Input	Output
$ 00\rangle$	$ 00\rangle$
$ 01\rangle$	$ 01\rangle$
$ 10\rangle$	$ 11\rangle$
$ 11\rangle$	$ 10\rangle$

Z

Input	Output
$ 0\rangle$	$ 0\rangle$
$ 1\rangle$	$- 1\rangle$

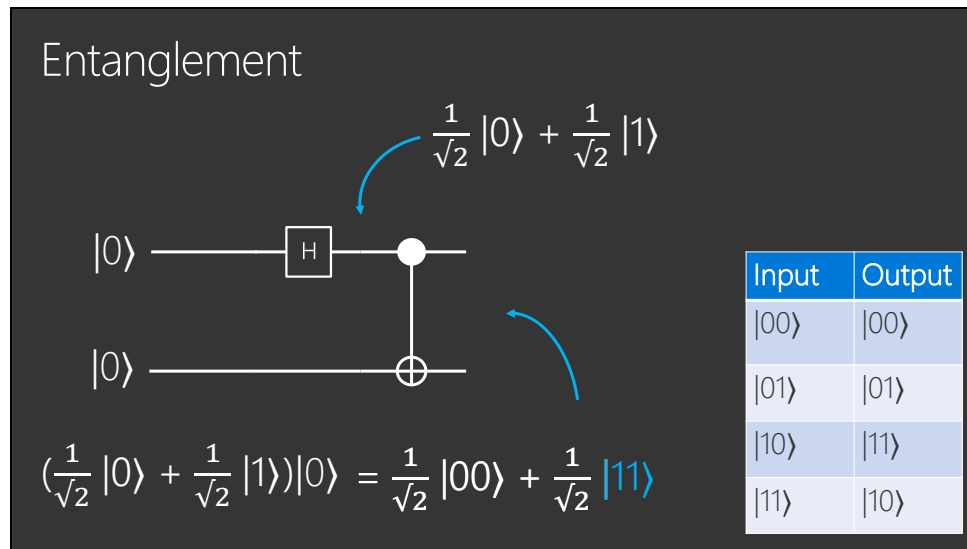
Hadamard

Input	Output
$ 0\rangle$	$\frac{1}{\sqrt{2}} 0\rangle + \frac{1}{\sqrt{2}} 1\rangle$
$ 1\rangle$	$\frac{1}{\sqrt{2}} 0\rangle - \frac{1}{\sqrt{2}} 1\rangle$



Earlier Anita discussed entanglement, how we can have two qubits in such a way that they are correlated. I'm going to show you how we can do that using the gates we just learned about.

First we're gonna need two qubits, one is mine, and one is going to be Anita's, and we're going to entangle them. There's two steps to this process, the first is to apply a Hadamard gate, and then a CNOT gate. So let's apply the Hadamard gate first. We apply that to my qubit.



Then we apply the CNOT gate.

Entanglement - Summary

$$|00\rangle \longrightarrow \frac{1}{\sqrt{2}} |00\rangle + \frac{1}{\sqrt{2}} |11\rangle$$

So this takes us from 00 to this state, which has an interesting property. When I measure my qubit, the first one, the state collapses and gets rid of the one part, so we are left with this. That means that Anita's qubit, this one, is also in the same state – due to a measurement that *I* made. The same can be said, if I were to measure my qubit, and the result was a one, Anita would know exactly what state her qubit was in without having to measure it. We can use this in some interesting ways, such as with quantum teleportation which I'm going to explain now.

Quantum Teleportation - Motivation

 $|\psi\rangle$

Physically?

 $|0\rangle$

Classically?

 $|0\rangle$

Quantum teleportation!

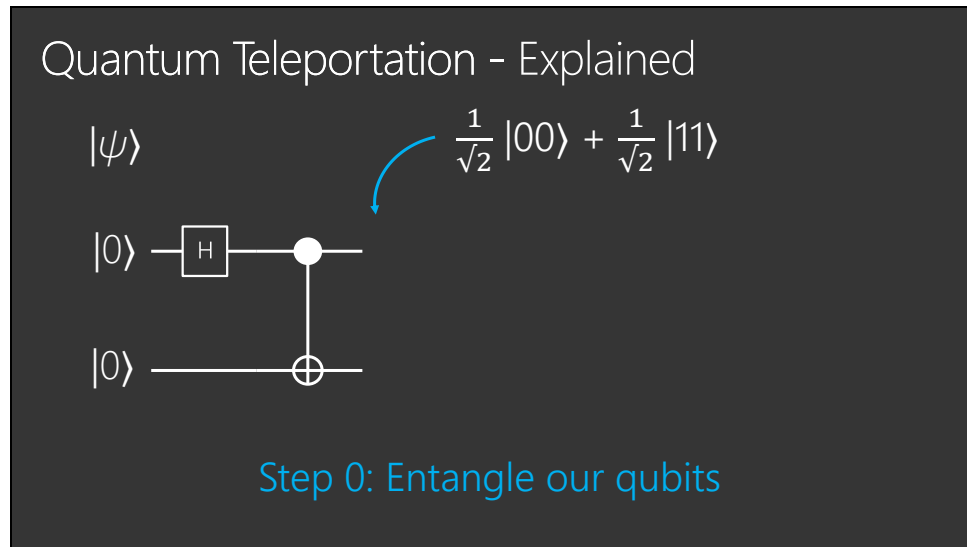
So now I'm going to talk about quantum teleportation. If at this point you're still struggling to get to grips with the previous stuff, don't worry. I'm going to explain the motivation of **why** you'd want to do this, so you can come back and look at the implementation later. The key thing is understanding why this is useful. Now just like before, I have my qubit, and Anita has hers. Now I'm going to introduce a third quantum state. And it's got this funny sign here because that's how people tend to notate a general qubit. It's general in the sense that I don't know what those values of alpha and beta are. However I want to send this state to Anita.

Now there's a few problems we have to think about first. Firstly – physically how can we send it? Currently the networks we have, fibre if you're lucky, send classical information. 0s and 1s. You need different networks to send quantum states and we don't have those linking me and Anita.

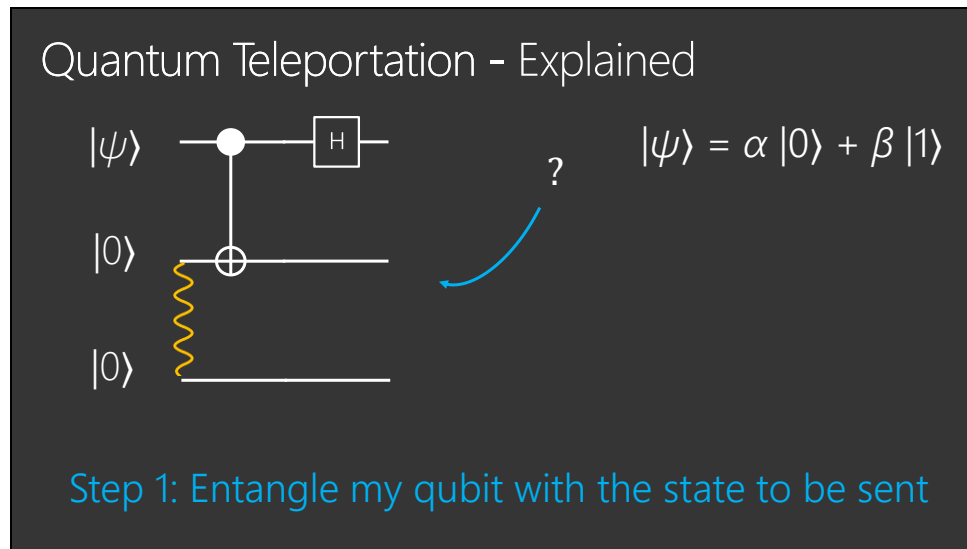
So if I want to send this state to Anita, I need some classical representation of a quantum state. Now we know quantum states are continuous, so even if I did have complete knowledge of the state, it would be described by an infinite number of bits... and you know that's going to take a long time to send.

And from what we said about measurement, the state collapses, so I couldn't even know that information anyway.

The answer is quantum teleportation, which is a very smart way of using what we've just learned about entanglement in order to communicate this unknown state to Anita, using only two bits of information. So let's see how we do that.



I say step 0, because we needed to entangle our qubits **beforehand**, that requires us to be physically in the same location. So let's say we did that, and then Anita went home to Bristol, and I went back to Reading. Our qubits are now in the state ...



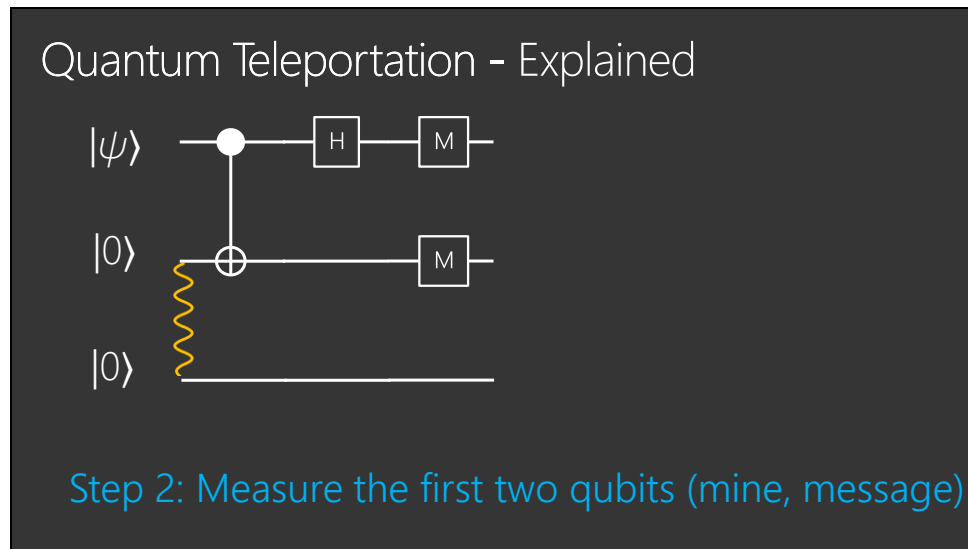
The first step we're going to do is entangle my qubit with this unknown one, and we can reuse that same circuit that we used before, entangling them by using the Hadamard gate, followed by the CNOT gate. Now the unknown general state we write as this. So what does our output look like?

Quantum Teleportation - Explained

$$\frac{1}{2} [|00\rangle(\alpha|0\rangle + \beta|1\rangle) + |01\rangle(\alpha|1\rangle + \beta|0\rangle) + |10\rangle(\alpha|0\rangle - \beta|1\rangle) + |11\rangle(\alpha|1\rangle - \beta|0\rangle)]$$

So what does our output look like? I'm not going to explain the intermediate steps because it's late and I don't want you to fall asleep, so I'm just going to show you the output. If you're interested in the maths in between, I've written a blog post on it which I'll link to on the events page.

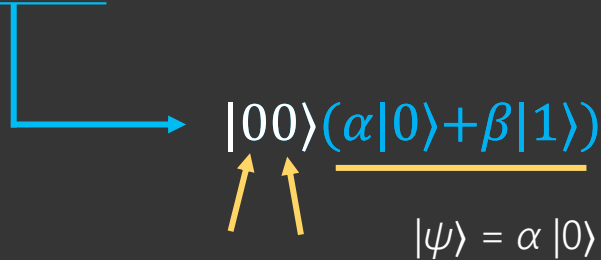
Back to our circuit though. This is the output. It does look terrifying but I'm going to explain what each part means. But first we're going to take two measurements.



This is what our circuit looks like, you see we've measure the first two qubits. So how does that relate to our result?

Quantum Teleportation - Explained

$$\frac{1}{2} [|00\rangle(\alpha|0\rangle + \beta|1\rangle) + |01\rangle(\alpha|1\rangle + \beta|0\rangle) + |10\rangle(\alpha|0\rangle - \beta|1\rangle) + |11\rangle(\alpha|1\rangle - \beta|0\rangle)]$$



$$|00\rangle(\alpha|0\rangle + \beta|1\rangle)$$

$$|\psi\rangle = \alpha |0\rangle + \beta |1\rangle$$

Step 3: Interpret the result

We can break this result down based on our measurement. Let's look at this first term. I'm going to highlight the first and second qubit. That's these two. If we measure the first qubit and get a result of nought, and measure the second qubit and get a result of nought, this is the only term that matches this result – where both the first and second term are nought. We therefore know that Anita's qubit is in this state here. If we compare that to the unknown state – they match! We know that Anita's qubit is now in the state we wished to send. What about the others?

Quantum Teleportation - Explained

$$\frac{1}{2} [|00\rangle(\alpha|0\rangle + \beta|1\rangle) + \underline{|01\rangle(\alpha|1\rangle + \beta|0\rangle)} + |10\rangle(\alpha|0\rangle - \beta|1\rangle) + |11\rangle(\alpha|1\rangle - \beta|0\rangle)]$$

Apply a NOT gate

$$|01\rangle(\underline{\alpha|1\rangle + \beta|0\rangle})$$

$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$

Step 3: Interpret the result

Say we measure our two qubits and get a result of 01, then the only term that matches it is this one. We can see Anita's qubit is in the state $\alpha|1\rangle + \beta|0\rangle$. This doesn't match the state we were trying to send, so we apply a NOT gate. We know that flips 0 and 1.

Quantum Teleportation - Explained

$$\frac{1}{2} [|00\rangle(\alpha|0\rangle + \beta|1\rangle) + |01\rangle(\alpha|1\rangle + \beta|0\rangle) + \underline{|10\rangle(\alpha|0\rangle - \beta|1\rangle)} + |11\rangle(\alpha|1\rangle - \beta|0\rangle)]$$

Apply a Z gate

$$|10\rangle(\alpha|0\rangle - \beta|1\rangle)$$

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$$

Step 3: Interpret the result

Say we measure our two qubits and get a result of 10, then the only term that matches it is this one. We can see Anita's qubit is in the state $\alpha|0\rangle + \beta|1\rangle$. This doesn't match the state we were trying to send, so we apply a Z gate. We know that flips the sign.

Quantum Teleportation - Explained

$$\frac{1}{2} [|00\rangle(\alpha|0\rangle + \beta|1\rangle) + |01\rangle(\alpha|1\rangle + \beta|0\rangle) + |10\rangle(\alpha|0\rangle - \beta|1\rangle) + |11\rangle(\alpha|1\rangle - \beta|0\rangle)]$$

Apply a Z gate and NOT gate

$$|11\rangle(\alpha|1\rangle - \beta|0\rangle)$$

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$$

Step 3: Interpret the result

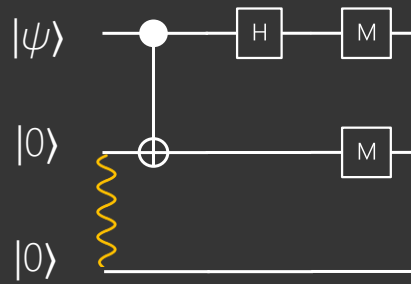
Say we measure our two qubits and get a result of 11, then the only term that matches it is this one. We can see Anita's qubit is in the state $\alpha|1\rangle - \beta|0\rangle$. This doesn't match the state we were trying to send, so we apply a Z gate, which flips the sign. Then we apply a NOT gate, which flips the 0 and 1.

Quantum Teleportation - Summary

Measurement	Operation
$ 00\rangle$	Do nothing
$ 01\rangle$	Apply NOT
$ 10\rangle$	Apply Z
$ 11\rangle$	Apply NOT, Z

Step 4: Apply the gates

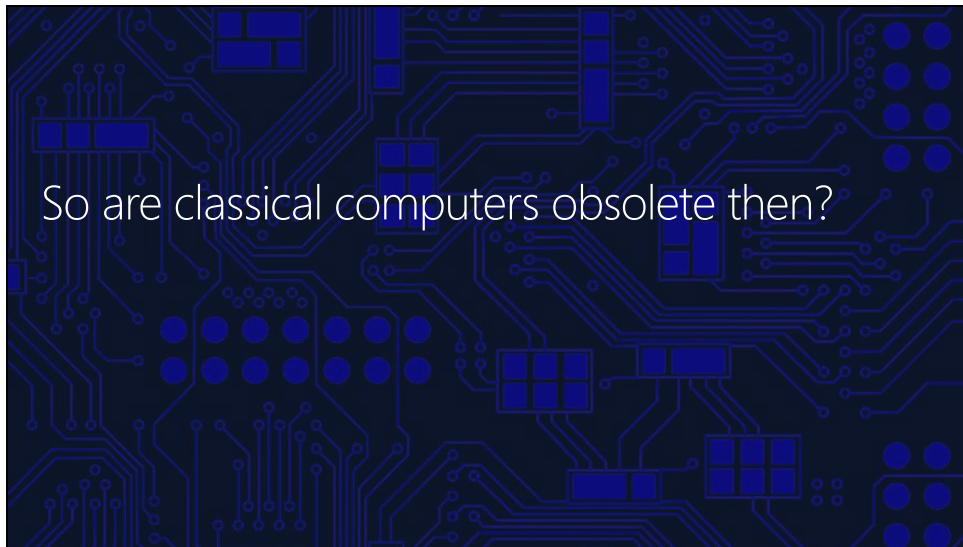
Quantum Teleportation – Code!





aka.ms/QuantumComp

Try It Out (Q#) without installing the QDK!



That kinda made it seem like quantum computers are the only way forward for computing in the future, didn't it?! So will we all be throwing away our laptops soon? The short answer is no!

Technically, because classical computers are a subset of quantum computers and you are able to simulate any classical computer using a quantum one, quantum computers will always win in terms of efficiency (overall).

However, quantum computers are difficult and expensive to make! There is also a large set of problems for which quantum methods do not provide any speedup – they will work just as well as our classical methods. So while in certain cases quantum computers can offer a much more efficient solution to a problem, this is not always true and because it's cheap to build and run a classical model, they are definitely still useful to keep around!

Classical computers are also useful in that they are easy to code and interact with – the computers of the future will likely make use of this for data preparation, control of the quantum circuits and cleanup/presentation of the results. We will build hybrid systems!

Moreover, recent advances in quantum algorithm design are actually feeding back into the design of classical algorithms, making them more efficient by utilising quantum 'tricks' – these are known as quantum-inspired algorithms and could power some really cool new tech! One example of this is our current partnership with Case Western Reserve University over in the States, where we are planning to use quantum-inspired algorithms to improve tissue characterisation for MRI scanning (this process is known as Magnetic Resonance Fingerprinting).

Because of this kind of feedback loop, the threshold that we need to pass before people can declare 'quantum supremacy' (i.e. when we have built useful quantum machines that can compute more efficiently than classical ones) keeps shifting! Looks like our good old classical computers are not out of the race just yet 😊

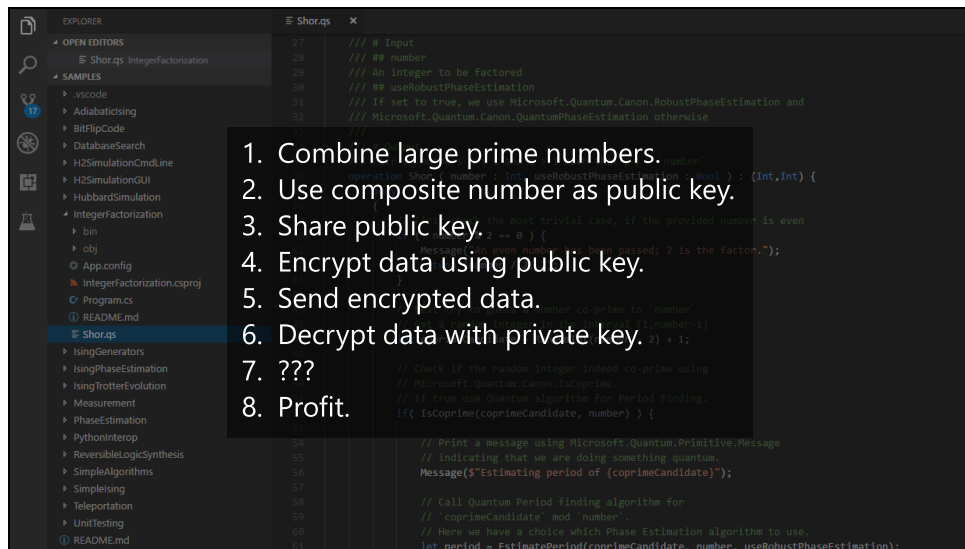
Now that we've got that out of the way, let's look into some examples of how quantum computers *could* change our lives!

Slide 58



Given that we're both developers and we work for a tech company, let's start here!

Quantum computing is set to disrupt the technology industry in a big way. The most famous example of this is probably Shor's factorisation algorithm so we'll start there!



So as Julie mentioned earlier...

Given that we're both developers and we work for a tech company, let's start here!

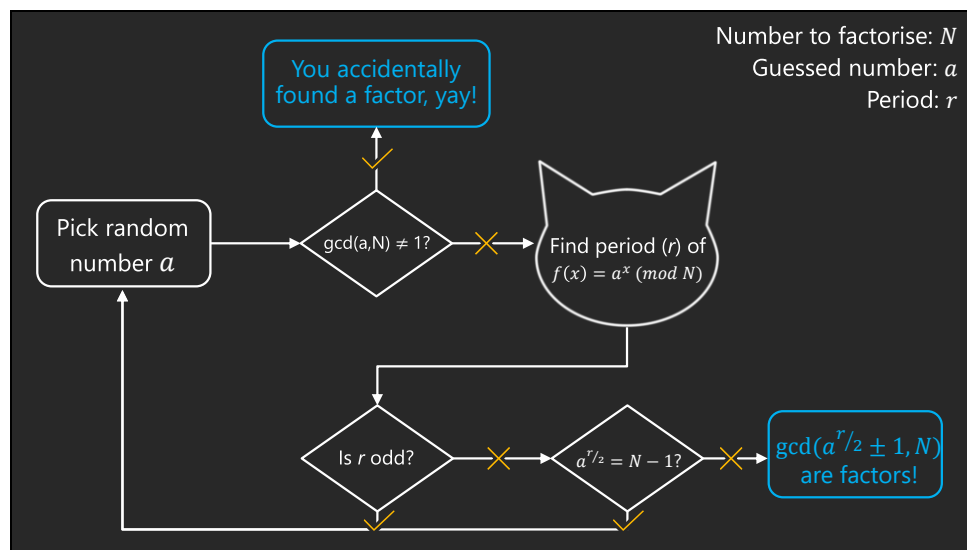
Quantum computing is set to disrupt the technology industry in a big way. The most famous example of this is probably Shor's factorisation algorithm so we'll start there!

In a nutshell, secure communications today work like this:

- We use combinations of large prime numbers to generate a public key and a matching private key. A message encrypted with our public key can only be decrypted using our private one.
- We share our public key with the party who wants to send us encrypted data.
- They use the public key to encrypt their message then send the encrypted message across to us.
- We then use our private key to decrypt the message.

Technically, anybody with a powerful enough computer and enough time to spare would be able to break this encryption scheme by deriving the prime factors of the public key.

However, this is really hard to do in a reasonable amount of time with a classical computer – in computer science terms, the complexity of the most efficient algorithms developed to date is sub-exponential (meaning the complexity grows faster than a polynomial function but not quite as fast as an exponential one) – this means we would be long dead before we actually managed to crack a 4096-bit key.



Unfortunately for us, this is where quantum computers come in and ruin the party. Back in 1994, a chap called Peter Shor proposed a quantum algorithm that could be used to find the prime factors of any arbitrarily long number. By leveraging quantum effects, this algorithm is able to calculate these factors in *polynomial time* (polynomial in $\log N$ in case you are interested).

How does this work?!

Shor's algorithm is a hybrid algorithm that makes use of a classical computer in the pre- and post-processing stages. It's power lies in that the preprocessing stage reduces the problem of factorisation to the problem of period finding, which quantum computers are much better at doing than classical ones. Period finding just means finding how often your function repeats e.g. if we took a sine wave it would have a period of π because that's how long it takes the function to return to its starting point

It basically works like this:

- Pick a number at random
- If it shares a common denominator with the number we are trying to factor that is not just 1, huzzah, we accidentally found a factor!
- If not, find the period of a special mathematical function – this is the bit done on a quantum computer! They are really good at finding orders/periods in polynomial time. This part makes use of the Quantum Fourier Transform and some Hadamard gates (among other things).
- If period is odd, bad luck try again!
- If our guessed number raised to the power of the period/2 equals $N-1$, bad luck try again!

- Otherwise, calculate the greatest common denominator of our number and our guess raised to $\text{period}/2 \pm 1$ to get our factors!

Let's have a look at this in code!

- Basically point out that the majority of stuff is classical

This has spun off a whole new field of research called post-quantum cryptography! This is focused on the development of (classical) cryptographic methods designed to be secure in a world where RSA encryption no longer has any integrity.



People are, of course, also working on quantum cryptographic methods! The most well known of these methods is known as quantum key distribution (or QKD), in which two parties share a key using quantum states. But how does this work? Let's illustrate with an example 😊

So let's say Frances and I are given the job of organising Satya Nadella's birthday party (he's our CEO, in case you didn't know!) It is, of course, vitally important that he does not know what we have planned! Only the best encryption will therefore do – and the best encryption is quantum encryption (as far as we know anyway)!

https://en.wikipedia.org/wiki/Quantum_key_distribution#Quantum_key_exchange



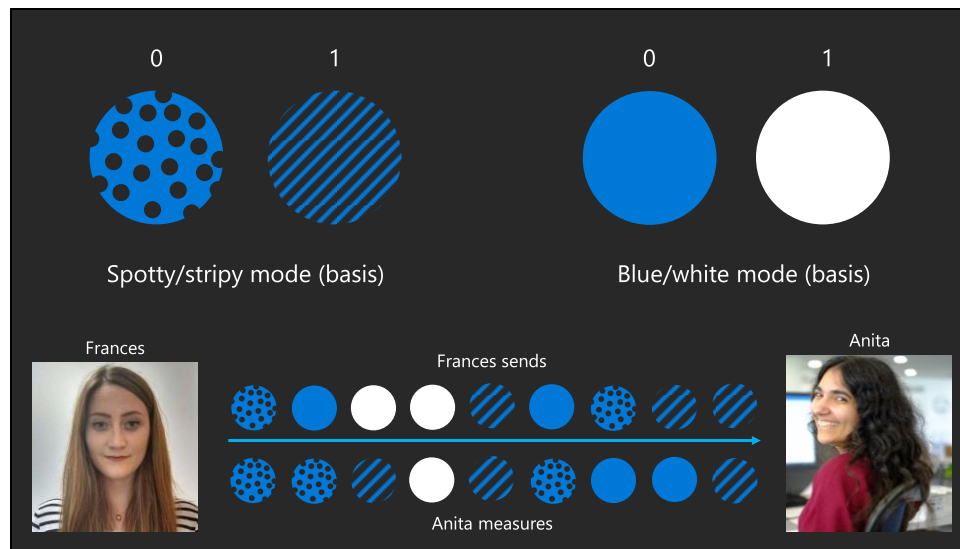
First things first, we need to share a key that we can then use to encrypt our messages. Because we need to be able to easily communicate our qubit states, we choose light particles (or photons) as our qubits – we can then send the information via fibreoptic cable or just through free space!

Now, photons have a property called polarisation. It basically means the direction in which the light wave is oscillating (hand gestures). In this instance however to avoid getting bogged down in wave mechanics, we are going to pretend our light source has two ‘modes’ (orthogonal bases, in physics-speak). One of them sends photons that are stripy or spotty, like we see on the left. The other sends ones that are either white or blue, like we see on the right.

These modes are totally independent of one another, i.e. no measurement in spotty/stripy mode can return any information about a photon in blue/white mode and vice versa. This is known as *orthogonality* in physics. If we measure our photons using the wrong mode we will just get a random result. So for example if Frances sends a stripy photon but I measure it in the white/blue mode I will get white or blue with 50:50 probability.

So beforehand what we do is agree on a code for each mode – spotty means 0 and stripy means 1, blue means 0 and white means 1 (like shown). We can then use these binary figures to give us our key 🔑

https://en.wikipedia.org/wiki/Quantum_key_distribution#Quantum_key_exchange



Next, Frances starts sending me the key! She sends a random sequence of photons in a mixture of both modes. Once I translate this to a binary sequence using our scheme from before, I can then use this to encrypt my data. This is what Frances sends: CLICK

When I start receiving photons, I begin measuring. Each time I measure, I pick a mode at random by e.g. flipping a coin. When the mode Frances has sent and the mode I measure match, I get the correct state out, as you can see. When I pick the wrong mode, I get a value (from that mode) at random. You can see this demonstrated on the slide. CLICK

Frances sends me lots and lots of these photons. When she is finished, we have a phone call (not over a secure line) and we share the modes the photons were sent/measured in. When her sending mode and my measurement mode match, we keep the photon. When they don't, we discard it. We don't tell each other over the phone which modes matched.

Interestingly, none of the communications between our two parties need be secure – any interference or eavesdropping from a third party will affect the quantum states being sent and therefore we will know we are being listened to.

Another cool computer-sciencey thing we can do with quantum states is use them as perfect random number generators – we can leverage the probabilistic nature of quantum mechanics in order to generate truly random numbers :O e.g. if you were to prepare N qubits in equal superposition then measure each one separately, you would get a random result out each time with 50:50 probability. Combine these together and you have yourself a nice, properly random number! (assuming your measurement and environs aren't biased).

https://en.wikipedia.org/wiki/Quantum_key_distribution#Quantum_key_exchange



Now all we need to do is convert the photon states to binary and we can get on with our party planning! CLICK

BUT WAIT

CLICK

Disaster! Satya's got wind of our birthday party planning mission and wants to know what we are planning! Dun dun dunnnnnn :O

Turns out, this isn't actually a problem! Because we are sending quantum states and Satya doesn't know what mode Frances is sending her photons in, he can only do the same as me and measure in each mode at random. When he gets the mode wrong he sees a random result just like I do!

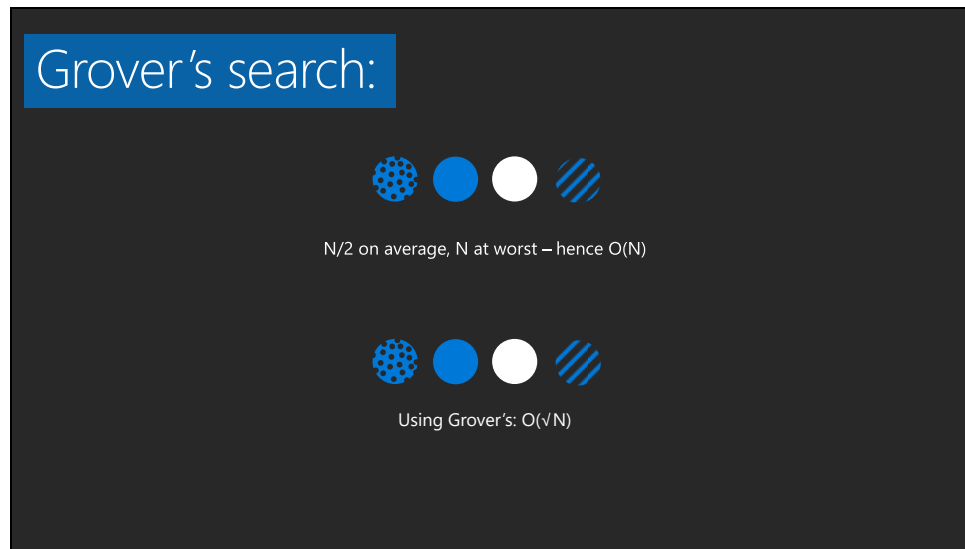
Now, because he's sitting in the middle of our communications network and it's not possible to clone the photon stream (because of something called quantum no-cloning theorem), when I measure the photons received his measurements will affect mine! This means that we can use the results of my measurement to tell if he has been eavesdropping.

Basically what we do is take a random sample of the key (i.e. photons which we sent/measured in the same mode) and compare the 0/1 values I measured with what Frances sent (again, over an insecure channel) and our readings differ (assuming we have a perfect system with no loss or corruption – there are ways to work around this but no need to go into detail here), then we know that somebody has intercepted our key! We can therefore discard the key, change our communications channels and start again 😊

Interestingly, none of the communications between our two parties need be secure – any interference or eavesdropping from a third party will affect the quantum states being sent and therefore we will know we are being listened to.

Another cool computer-sciencey thing we can do with quantum states is use them as perfect random number generators – we can leverage the probabilistic nature of quantum mechanics in order to generate truly random numbers :O e.g. if you were to prepare N qubits in equal superposition then measure each one separately, you would get a random result out each time with 50:50 probability. Combine these together and you have yourself a nice, properly random number! (assuming your measurement and environs aren't biased).

https://en.wikipedia.org/wiki/Quantum_key_distribution#Quantum_key_exchange

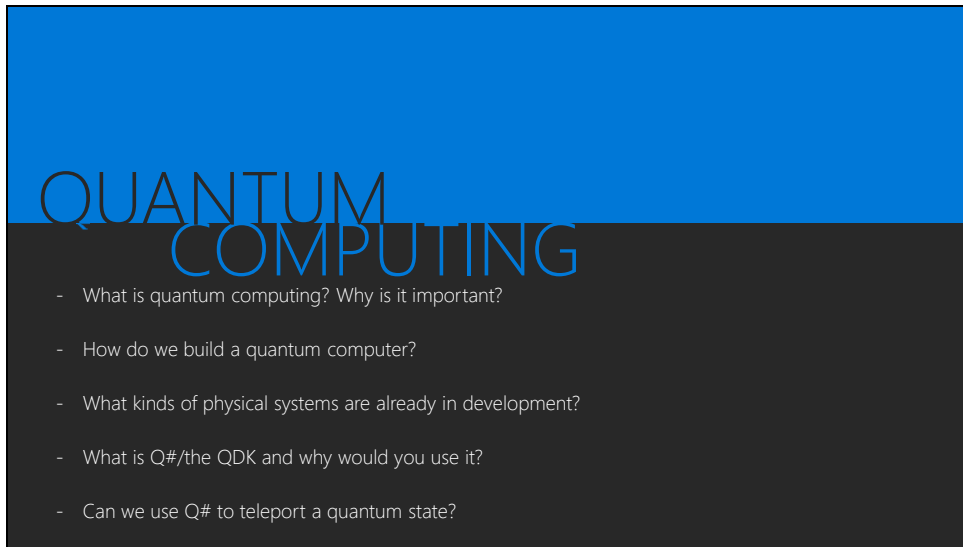


Suppose we have an unstructured, unsorted database with N entries. We wish to find a particular element of this database. With classical computation models, we wouldn't be able to find our solution in less than linear time. We would have to search through every element until we find what we're looking for. This takes an average of $N/2$ attempts and N in the worst case thus a time complexity $O(N)$ (we don't want that!).

In 1996, Lov Kumar Grover proposed an algorithm which would do this search in time complexity $O(\sqrt{N})$ (a lot faster!) by utilising the nature of quantum systems. Compared to other quantum algorithms which speed up their classical counterparts exponentially, this quadratic speed-up may not seem very impressive but it is the best known method for this problem to date.

Grover's algorithm is typically a searching algorithm which improves on classical algorithms that can already solve the same problem. For grover's algorithm to work, a function must be constructed such that when applied to a representation of items in a database, it will give a particular value for the required data entry and a different one for all others. This function is key and varies depending on the data and the output required.

It works by increasing the amplitude of the search result we require, then when we measure, the probability of it being observed has increased.



QUANTUM COMPUTING

- What is quantum computing? Why is it important?
- How do we build a quantum computer?
- What kinds of physical systems are already in development?
- What is Q#/the QDK and why would you use it?
- Can we use Q# to teleport a quantum state?

So here we are again back where we started! I hope that during this session I've answered these questions for you (at least at a high level!) – unfortunately we haven't had time to do a deep dive today but if your interest has been piqued, please don't hesitate to get in touch or check out our blog (details on the next slide).



aka.ms/QuantumComp

Try It Out (Q#) without installing the QDK!

tio.run/#qs-core

Q&A?

Q. Do you follow Microsoft Quantum on Twitter?

A. No? Go to aka.ms/QuantumTwitter

Q. Do you receive the Microsoft Quantum newsletter?

A. No? Go to aka.ms/QuantumNewsletter

Q. Interested in learning more about quantum computing from the ground up?

A. Yes? Go to aka.ms/QuantumAdventures

Bonus! github.com/frtibble/QuantumWorkshop

Before you ask us any questions, we've got some for you



Thanks for listening! Our contact details are on the slide, as well as a link to the blog we are writing as an introduction to quantum computing.

Any questions?