Quantum Computation and Quantum Information

Nishant Abhangi

June 10, 2020

Abstract

This report aims to provide an introduction to the theory and practice of quantum computation. Broadly, it covers the physics of information processing and quantum logic gates and tries to emphasize that computation is inherently physical in nature.

Contents

1 History and development of quantum computation

This section covers the historical development of quantum mechanics and classical computing and information theory, and where this fits into quantum information science.A brief discussion of models for classical and quantum computation is given, which would be further expanded in the rest of the report.

1.1 Models for computation

Some models for classical computation are the Turing machine, Finite automaton and Lambda calculus.The Turing machine and Lambda calculus are universal, i.e. any computation can be done with these models while the Finite automaton is not universal–it does not solve all computational problems. However, the model which is most used for quantum computers is the circuit model, and this is actually super-universal. In other words, there are some uncomputable problems which can be solved with a circuit model, but not by a Turing machine. This is a problem, and it would be fixed later.

The circuit model is doing computation by wires and gates like NOT, AND and OR. The input bits are fed to a circuit of gates and finally some output bits are returned. These circuits are much older than Turing machine, which was conceptualized in 1936. So, why did Turing not use these circuits for his model of computation? It's because circuits are only designed for a finite size input. For example, consider the problem of multiplying two n-bit numbers.A circuit can be made for multiplying two 10-bit numbers or two 12-bit numbers, but circuits can't do computations which are independent of the size of the input because circuits are specified for a particular input size. Interestingly, that's what allows it to do uncomputable problems.

Consider an uncomputable problem. Does the Turing machine with a program P and input I halt? This can be solved by a family of circuits. Let the program P have p bits and and input I have i bits. Consider a circuit with $2^{p \cdot i}$ inputs and the output is given by the function

$$f(P, I) = \begin{cases} 1 & \text{if the Turing machine halts} \\ 0 & \text{if the Turing machine does not halt} \end{cases}$$

There is a different circuit for different sizes of input, and hence, a family of circuits could solve the problem. This makes it possible to hide uncomputable information in the design of a circuit. This is not good, and so Turing invented the Turing machine instead of using circuits for his model of computation.

Though a quantum Turing machine is theoretically possible, but because it has to be kept in a coherent superposition of the position of the head on different places of the tape, it is almost impossible to engineer this in practice, and hence this model is not widely used. On the other hand, in the circuit model, quantum bits are kept in coherent superposition, and gates are applied to do the computation. Though this model is also difficult to make, it is much more feasible than a quantum Turing machine. In fact, quantum computers based on circuit model consisting of about 50 quantum bits have been made, while not a single quantum Turing machine has been made.

For the circuit model, the description of the circuit should be the output by a classical computer program. This prevents it from hiding uncomputable information in the description of different circuits of different sizes. This means, that there should be a classical computer program where the input is n and the output is a quantum circuit of n quantum bits, and that circuit should solve the problem of interest.

1.2 History of quantum mechanics and quantum computation

1900-1930 Quantum mechanics was developed.

1936 Einstein, Podolski, and Rosen (EPR) argued that quantum mechanics is incomplete. It is not believed anymore. Quantum mechanics allows preparation of two particles(EPR pairs) such that whenever the first particle's momentum p (position x) is measured, the second one's momentum(position) is $-p(-x)$ and vice-versa. Hence, if the position of first particle and simultaneously the momentum of second particle are measured, both the momentum and position of first particle are determined, but quantum mechanics doesn't allow the simultaneous knowledge of both position and momentum as they are conjugate variables. Hence, they argued that there must be some information hidden in these particles, which is beyond what quantum-mechanics tells. Schrodinger replied that quantum mechanics is not incomplete, but these particles are *entangled* and this is a property entangled particles have.

1964 Bell proved that there is no classical explanation for the behavior of EPR pairs. So, if quantum mechanics is incomplete, it is not incomplete in a classical way.

1982 Aspect experimentally demonstrated that Bell's predictions were correct. Since then, many experiments have been conducted which closed various loopholes, strngthening the fact that there is no classical explanation for quantum mechanics.

1982 Herbert published a paper called FLASH—First Laser-Amplified Superluminal Hookup, which suggested faster than light communication using strange properties of quantum mechanics and EPR pairs. But it was soon proved to be wrong, as the no-cloning theorem was proved, which stated that an unknown quantum state cannot be duplicated.

1982 Feynman and Manin said that it is very difficult to simulate quantum mechanics on a classical computer. It requires $O(2^n)$ computational steps for simulating n particles. They argued that quantum computers would be faster.

1985	Deutsch described the quantum Turing machine. He said it would be good for simulating quantum mechanics, although he didn't give much details. He posed the question whether it is efficient for classical computations.
1992	Deutsch and Jozsa showed that indeed quantum computers were efficient for classical computation, although it was not that satisfactory.
1993	Bernstein and Vazirani gave a quantum algorithm which was linearly faster than a classical computer.
1994	Simon suggested a problem that can be solved exponentially faster on a quantum computer than a classical computer. His algorithm used periodicity, which was a key component in future quantum algorithms.
1994	Shor gave almost exponentially faster quantum algorithms for two important problems: factoring and discrete logarithm. This created a huge interest in quantum computing.
1995	Grover came up with his search algorithm, which searches a space of size n in $O(\sqrt{n})$ steps.
1996-2020	The above algorithms are the most important algorithms. Since then, many quantum algorithms were developed for less important problems.

Table 1: Major historical events in quantum computation

1.3 Computation must be robust against noise

Immediately, after Shor presented his factoring algorithm, objections were raised that factoring cannot be done because error correction is not there in a quantum computer. If every quantum operation is not accurate to even 1 in 10^9, the errors are going to mount up, and the result will be worthless. The major obstacle to error correction is the no-cloning theorem.

It is natural to develop quantum error correction along similar lines to classical error correction. The techniques for classical fault tolerance are:

1. *Checkpointing* : The state of the system is recorded periodically, and if something goes wrong, the computation can revert to the previous checkpoint.

2. *Error correcting codes* : k bits are mappped to n bits, where $k < n$ and the true state can be recovered from some number of errors. These are excellent for memory and for communication, and hence almost all computer memories and electronic communications use this technique. But it is not used for classical computations.

3. *Massive redundancy* : A large number of copies of the computation are maintained. These are compared periodically, and those found different from the majority are discarded and are replaced with the correct computation. This technique can be applied to individual bits or whole computation.

However, quantum operations must obey the no-cloning theorem and hence the above techniques must be checked for compatibility with no-cloning theorem. Clearly massive redundancy can't be used as the computation can be started with many copies, the copies can also be compared, but the correct states cannot be copied due to no-cloning and hence the number of copies keep diminishing and the method becomes useless. Checkpointing is also not feasible as to create a checkpoint, the state of the system has to be copied and again no-cloning theorem prevents it. Hence, the only option which remains is creating error correcting codes, and in fact, it works. It can be used for computation, communication and memory in the quantum computer. The reason it it is not used for computation in classical computers is that it requires a lot more overhead compared to checkpointing and massive redundancy.

2 Classical computation and reversibility

2.1 Physical and conceptual models of classical computation

There are a wide variety of models for classical computation, some of them quite bizarre! The model can be mechanical—the abacus, the Curta calculators and even one machine which does addition with the help of a boll rolling down! Some thermostats are driven by pneumatic pressure and hence they are a kind of computer which controls temperature, but they are analog, unlike most digital computers. They can come in all shapes and forms. Perhaps, The most famous mechanical computer in history is Babbage's difference engine, and if he had been able to complete it, it would actually have become a universal computer.

There can be electrical models like the circuit model, which is also a way of buuilding a universal computer. Photons, instead of electrons can be used to make optical computers. There are also biological models of computation. In some sense, human beings are walking-talking computers.This idea of biology and biological systems as computers is currently going through a renaissance because of the notions of neural networks and deep learning and other kinds of networks of neurons that act as computation.

Note that the above discussed physical models are different from conceptual models. Physical models are the models by which computation is actually realized, while conceptual models are those we strive to realize. One of the examples of a conceptual model is The Turing machine. It has a head and a tape.The tape has slots in it which has 1s and 0s, and the tape extends infinitely in one direction. The tape is kind of a memory. The head does most of the computation, and it can read and write to the tape. The head is a finite state machine, which means that it can be in different states and it can transition to another state depending on what is read from the tape and the current state. The Turing machines come in many variations like the probabilistic Turing machine and universal Turing machine, which is capable of simulating any Turing machine. Another less known model is a Minsky machine.

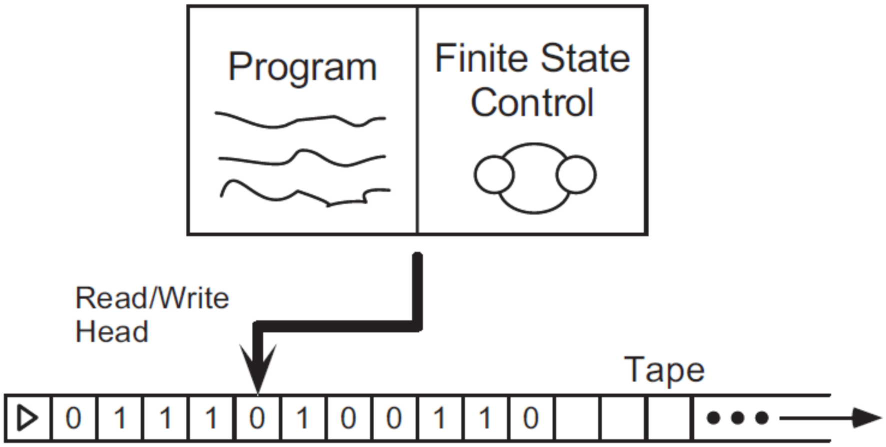


Figure 1: Main elements of a Turing machine. Blanks on the tape are denoted by a ‘b’. Note the $\triangleright$ marking the left hand end of the tape.

Another famous model is the cellular automaton. It is a grid in n dimensions, where each cell undergoes transitions based on the state of its neighbours, and the rules of these transitions and the transitions themselves give rise to computation. The best known example is Conway’s *Game of Life*. Then there is the von Neumann architecture. The key idea is to separate memory from arithmetic. So, there is an arithmetic logic unit(ALU) , the memory and some registers. Data is read from the memory and fed into the ALU and then the ALU feeds the data back into the memory after performing the computation. And this model is used in almost all processors today.

A new physical model is DNA computation. Here, there are strands of bases A, G, C, T. A associates with T and G associates with C, and this association is called ligation. When there are two different strands of DNA, they will ligate together at some locations. And this allows computation! And these biological operations have primitives very different from the electrical and optical gates. Some of these operations are :

1. *Melting* : This operation separates the ligated strands.
2. *Appending* : Adding two strands together, not by ligating, but by connecting them at the ends.
3. *Cutting* : Breaking a single strand in two smaller strands.
4. *Insertion* : Restriction enzymes are used to insert pieces of DNA at different places.
5. *Amplification* : The strands are amplified for measurement by Polymerase Chain Reaction.

It’s really wonderful that tools such as DNA can be used for computation. And in today’s world, such models of computation are to be explored, as the Silicon age is coming to an end, and Moore’s law is not reliable to scale computers exponentially in terms of their computational power and size. Different physical mechanisms should be investigated for computation. So, although this report is about quantum computation, the readers are urged to think about the broader questions about the physics of computation, and how to exploit these physical mechanisms to do computation. For example, today we have *CRISPR/Cas9* which allows us to snip out very specific locations on the DNA, which might change the complexity of DNA computation,

2.2 A brief introduction to computational complexity

Universality should be described in the language of circuit model of computation, as that is the basis for building a quantum computer. For example, all Boolean circuits(circuits that compute Boolean functions) can be composed of AND and NOT gates, which will be proved shortly. Boolean functions are those that take n bit of inputs and take it to to 1 bit.

$$f(x_0, x_1, \ldots, x_{n-1}) = 0 \text{ or } 1$$

In this notation the AND gate is $f(x_1, x_2) = x_1 x_2$ and the NOT gate is $f(x) = \bar{x}$. The question of universality also brings the question of complexity, because it is not useful to say something can be done without specifying its cost. The key idea is that some math problems are harder than others. This leads to the *strong Church-Turing thesis* :

> **Strong Church–Turing thesis**: *Any model of computation can be simulated on a probabilistic Turing machine with at most a polynomial increase in the number of elementary operations required.*

It's an extraordinary thesis. It defines an equivalence between models, whether they be electrical and optical, or electrical and DNA, or quantum and DNA, or quantum and classical. And for even that to be possible conceptually is remarkable.

One of the greatest motivating factors for quantum computing is the difference between two of the most important classes of mathematical problems. It is a fact that many problems can be expressed as decision problems.These are yes/no questions. Some examples are:

1. *Primality Testing*: Is the number m prime? Is this a hard problem or an easy problem? This actually was not known for many years. It was then realized that you could answer this question in polynomial time with some randomness. You wouldn't know it for certain. This is *Rabin's primality testing* algorithm. And then some 18 years ago, *Agrawal, Kayal and Saxena* proved that you could do it deterministically in polynomial time. So today there is a deterministic primality testing algorithm.

2. *Factoring*: Given composite integers m and l, $l < m$, does m have a non-trivial factor less than l? So we need to bound the size of the range of numbers we're going to consider as being factors.

If the time needed to answer these questions is polynomial in the size of the question(for example, in factoring, the size is number of bits of m), then the problems are polynomial in complexity. We say that it's in the class **P**. If the "yes" instances of the problem are verified in polynomial time with the aid of a witness, which is a piece of information that enables somebody else who's not terribly skilled at the arts but can be very reliable to verify your claim, then the problem is said to be in class **NP**. For the sake of completeness, we have the parallel class co-**NP** where the "no" instances are verified in polynomial time.

The reason for discussing these classes is because so much of the motivation for computation today, and much of quantum computation, comes from this question of **NP** versus **P**. We think that the **P** problems are easy, and the **NP** problems are the hard and meaningful ones to do.The figure below shows the relation between **P** and **NP**.

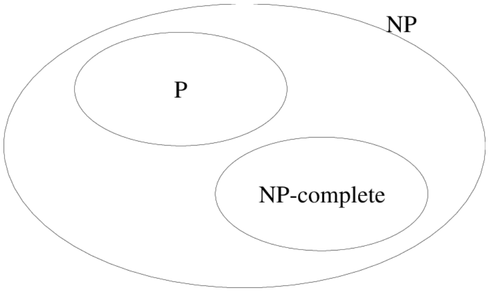


Figure 2: Relation between **P**, **NP** and **NP**-complete classes

P is a subset of something we might say is **NP**. But there's also the **NP**-complete class, which is the hardest of the **NP** problems. And they they're defined as such because if you can solve any of the problems that are **NP**-complete in polynomial time, then it gives you a polynomial time algorithm to solve any of the other problems in the **NP** regime. So where is quantum computation in all of this map? Well, I'm not going to answer that now, but I hope that after reading this report you'll start to appreciate where quantum computation is, relative to this landscape of the field of all problems and their complexities. Quantum computation sits kind of differently in this landscape. It has a complexity class, typically, of something we call **BQP**. And part of the reason for that is because the model is slightly different and doesn't fit directly into either of these classes, because sometimes the output is quantum mechanical, there are errors involved, and some things we'll come to later.

An example of an **NP**-complete problem, and now I'm going to go back to the concept of universality. is a problem called 3-SAT. And this is about the satisfiability of Boolean functions, which look something like this.

$$f(x_0, x_1, \dots, x_{n-1}) = (x_1 + x_3 + x_9)(x_4 + \bar{x_7} + x_{11}) \dots$$

Each sum term has 3 bits and and the goal is to answer does there exist an assignment of 0s and 1s to the n inputs such that the output is equal to 1. It seems very simple. It's very easy to write this problem on paper. But if you can solve this problem fast i.e. in time polynomial of n, then it turns out you can solve all the rest of the **NP** problems fast. In fact, if you can solve this problem fast, you can solve problems like the optimum way to pack boxes into a delivery truck, or the optimum way to route a packet in for information from Mumbai to Delhi. It's really remarkable how powerful such a simple problem can be. And yet it can be shown that in practice you can't really solve most of the instances of this problem as well as we would like to.

Are there problems that are neither **P** nor **NP**? The answer is yes. Another interesting class that sits outside the **NP** class is the class of counting the number of solutions to a problem. That's the class **#P**, pronounced sharp P. And so on and so forth you can create new classes, because you can count the number of things to count.

2.3 Universality of AND and NOT for boolean circuits

Now, I prove the claim about universality of AND and NOT for Boolean circuits made in the previous section. Let f be an arbitrary boolean function which maps $n+1$ bits to just 1 bit. And then we define two other functions, one called f_0 and another called f_1, which take in n bits.

$$f_0(x_1, \ldots, x_n) = f(0, x_1, \ldots, x_n)$$
$$f_1(x_1, \ldots, x_n) = f(1, x_1, \ldots, x_n)$$

Let y be an arbitrary bit. Then it is easy to see that following equation is correct.

$$f(y, x_1, \ldots, x_n) = yf_1 + \bar{y}f_0$$

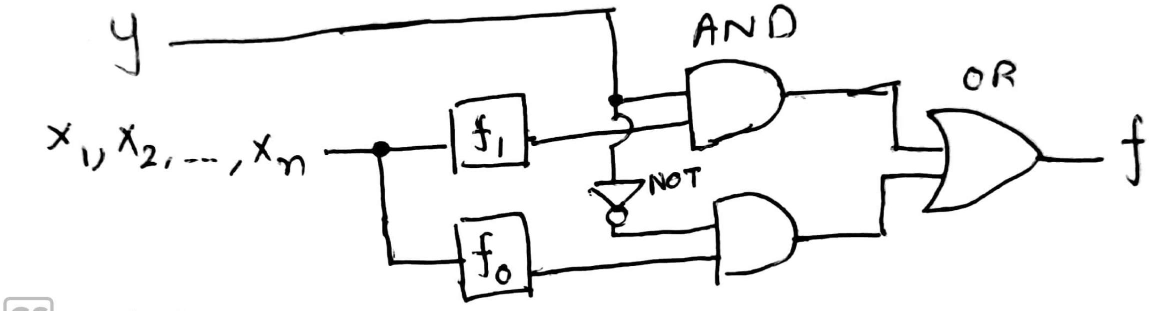


Figure 3: Circuit diagram for the function f

The "+" is an OR gate, and I leave it as an exercise to prove that the OR gate can be constructed from AND and NOT gates. It's easy, just use DeMorgan's theorem. Note that the proof of universality can be easily completed by induction now.

What are the resources required to compute f? It is very clear from the circuit diagram. We see that we need to double the number of function evaluations at each stage. We will recursively simulate f_0 and f_1 with exactly the same pattern and just keep on going. So you can see that the resources needed, grow exponentially, at least as 2 to the number of bits used in the system. Hence the complexity is approximately $O(2^n)$.

In addition, we use several other resources that are physical. I don't want this just to be a computer science argument. We have wires that are crossing over each other. We have wires that involve copying of data, something which is not allowed in quantum computation. So if we think about the physics of the circuit, we realize that there are odd resources beyond AND, OR, and NOT. We also had to use the resource of FANOUT and CROSSOVER. And of course, there are wires. So there's communication of the bits of information between all of these gates. And you want to think, if you're realizing a model of a computation, how do all of those resources scale? How many CROSSOVERs, how many FANOUTs, how many wires do I need as I am building something? And these kinds of questions are only recently becoming a barrier in implementations of large-scale integrated circuits as we reach limits where energy consumption becomes important to think about.

2.4 Thermodynamics and computation - Maxwell's Daemon

What's the physical cost of realizing such computations? It is very interesting to consider the fundamental limits upon computation imposed by the laws of physics. It's really remarkable, in many ways, how I discussed a great deal about computation so far ignoring the laws of physics. And physicists don't like this, from a standpoint of fundamental principles. Shouldn't everything be governed by the laws of physics? How come I was able to give you that strong Church-Turing thesis and it said absolutely nothing about Newton's laws or about Maxwell's equations? It seemed like it was a construct of pure math really, maybe pure audacity to be able to ignore the laws of physics. And there was a whole group of researchers around the world who thought deeply about that in the early 1900s and the late 1800s. And they came up with the relation of thermodynamics to computation.

The idea starts in 1867 with a construct known as Maxwell's Daemon. We have a cylinder with gas molecules in it. And these gas molecules are careening around at different velocities. And some of them are going to be hot, and some of them are going to be cold. Now, this box has a partition in it with a hole. And this partition has a sliding door, which can be actuated by this clever tiny person, called a daemon by Maxwell, sitting inside this cylinder. And Maxwell's Daemon looks at each one of these molecules hitting the door and tries to achieve this configuration where he or she puts all the cold molecules on the left and the hot molecules on the right-hand side by virtue of looking at them and selecting them one at a time as they can go through the door. So by doing this, Maxwell's Daemon should be able to create an imbalance of temperature, by virtue of which now Maxwell's Daemon hasn't done any work, apparently, because all the daemon had to do was open and close the door.

So it seems like this daemon is able to accomplish something very useful, create a configuration of gas which could then be used to extract energy and accomplish work and so forth, and yet expend exceedingly little energy in comparison with what the daemon could harvest from the gas molecules in a room, because any random distribution of molecules in a room, have a wide distribution, some which are cold, some which are hot. So why shouldn't the daemon be able to separate them just like this? This is a real question and it's connected to computation. Note that this is actually impossible(second law of thermodynamics)! And although I put exclamation mark here, really there should be a question mark, because at first it wasn't understood why it's

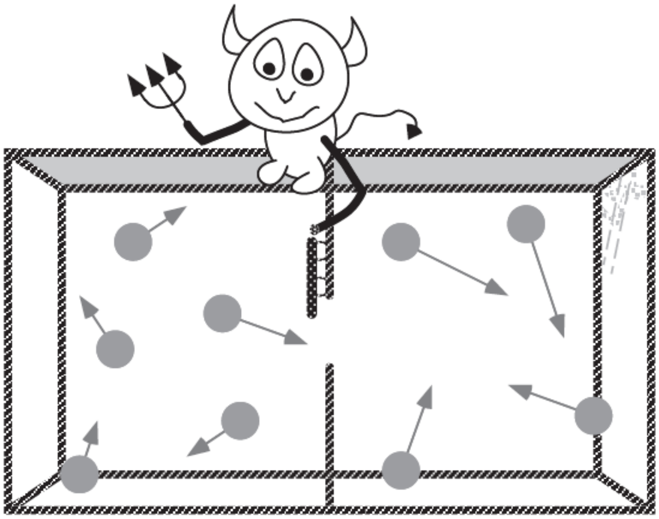


Figure 4: Maxwell's Daemon

impossible.

Is it because of the energy required to move the wall? Is it because of the measurement? Maybe it's too hard to measure single molecules. Maybe by measuring them sufficiently well, the Maxwell's Daemon would end up kicking the molecules in the wrong place. Or is it hard because there's a need for intelligence, in some way? Like, maybe no automaton could accomplish this, but an PhD could.

2.5 Szilard's engine and Landauer's exorcism of Maxwell's Daemon

Now I explain why Maxwell's Daemon cannot exist because of a fundamental reason related to computation. And this will help us a great deal in our journey through quantum computation, because we want to be aware of these limits of physics on computation, as we try to realize a new model of computation. So the way I will describe this is with a beautiful construction, called Szilard's engine from 1929. There are seven stages in a Szilard's engine. The basic idea is the core engine, which will again, be a cylinder of gas with very light pistons on its end faces which can slide in and out in the cylinder. There will be a wall that can come down through a small hole in the cylinder, dividing the cylinder into two equal parts. And imagine that there's just one gas molecule rumbling around, for sake of simplicity, inside this idealized vacuum chamber. In addition to this, there is a piece of scratch paper, which might seem a little odd. It's a little memory. Also, the cylinder is in contact with an heat reservoir.

In the first stage, the wall is taken out, the pistons are drawn to the end faces of the cylinder and the scratchpad is empty(Fig. 5a). In the second stage, the wall is lowered in the cylinder(Fig 5b). For the sake of concreteness, assume that the molecule is on the right of the wall. It's equally probable that it might have been on the left-hand side. In the third stage, we're going to measure where the molecule is. So imagine this is just like Maxwell's daemon looking with his or her

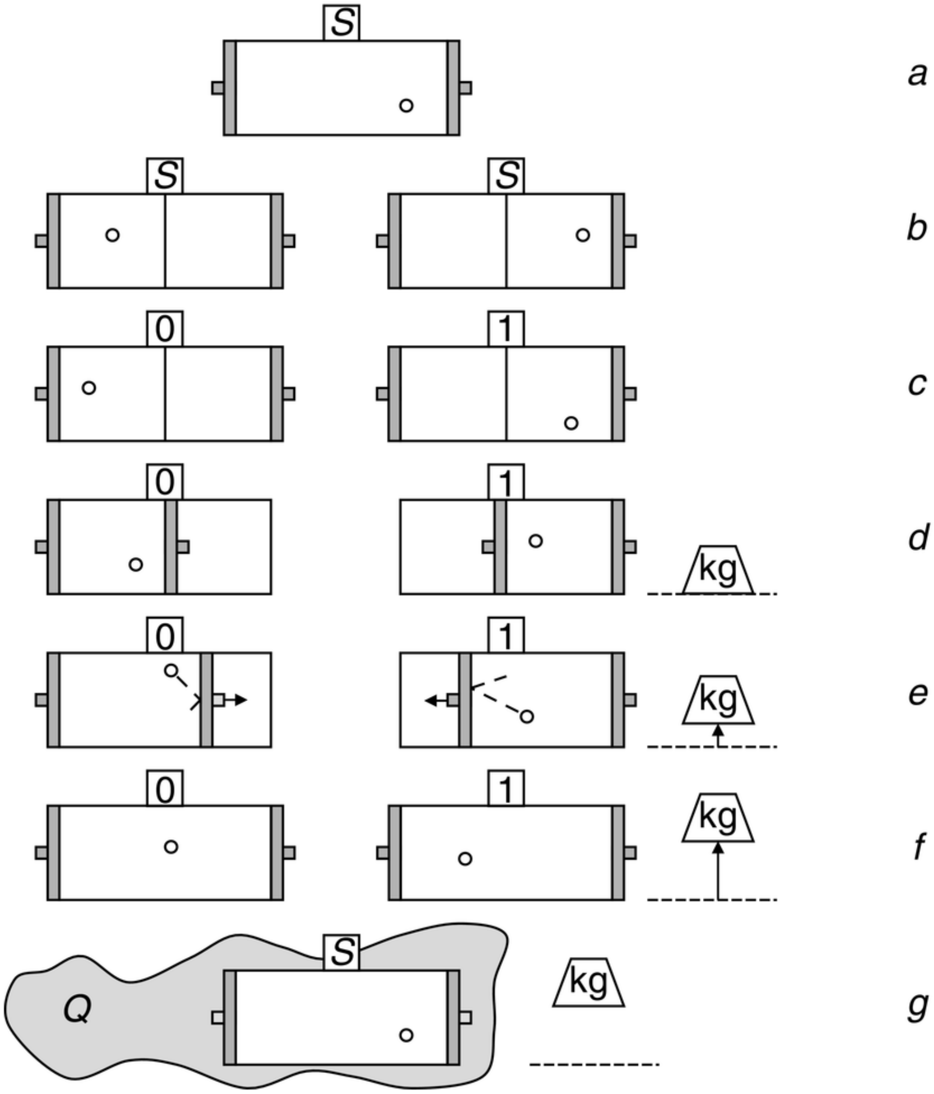


Figure 5: Szilard's engine. S denotes an empty scratchpad. Both possibilities where the molecule can be found are shown. The work on the piston is symbolised by lifting a weight.

eyes at a gas molecule to know whether the molecule is on the right-hand side or on the left-hand side of this cylinder.We don't change anything about the cylinder or its two pistons. We write down 1 on the scratchpad if it's on the right and 0 if it's on the left(Fig. 5c). In our example, we write 1. The fourth stage is the one which makes a difference with respect to whether you knew it was on the left or on the right. In our example, we push the piston on the left to the wall. The key idea is it costs us no energy and effort to move the left piston in, because there's no molecule over there(Fig 5d). And we can do that because we know the molecule is on the right-hand side. In the fifth stage, the wall is raised. And in this step, we have the molecule running around at some random point on the right-hand side of this cylinder. The molecule at some point will randomly have a trajectory, which hits the left piston and bounces off. And when it hits this movable piston, the piston moves backward. Remember that these are very light pistons. And this is a momentous molecule that's moving around. So after each collision, the piston moves out further(Fig 5e). In the sixth stage, the left piston again reaches the end of the cylinder(Fig. 5f). The process may be very slow, but eventually the piston reaches the left end. In the seventh stage, we clear the scratchpad, allow the molecule to regain the initial temperature and regain our initial setup(Fig 5g). This allows 100% conversion of heat into work, again violating second law of thermodynamics!

But there is a catch here. What's so hard about clearing the memory? What could be so hard about forgetting a single bit of information? It takes work, because we had to put some information by saying it's blank. We get rid of the randomness(0 and 1 were equally likely) by clearing the scratchpad. Getting rid of randomness(entropy) is hard(takes work), while getting rid of something you know is in some state is not so hard. And this is an important principle we'll come back to in reversible circuits. And it is exactly that cost that had been neglected in Maxwell's daemon's scenario. He didn't actually think about the cost of the daemon having to forget the information about the last molecule he or she had seen.

Does the movement of the wall require energy? It can be made very small because we are really good about building my wall on rollers. And so compared with the cost of everything else, it's a very small amount of energy. It's also reversible. Imagine that the wall was heavy. So I just let it drop, harvest that energy from the dropping, and bring it back up. So it can be made very, very small.

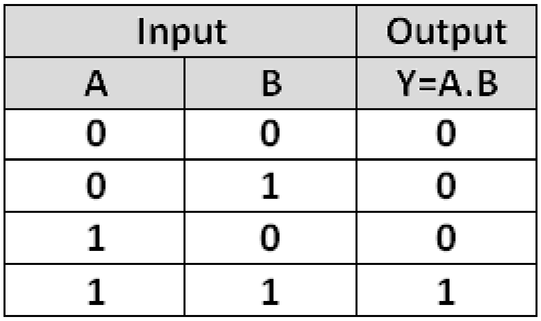
The connection between thermodynamics and computation was really made by Rolf Landauer in 1961. He argued that clearing a n-bit register is like compressing a gas in the sense that all the configurations, no matter what value we have in that n bit register, must go to some definite final state, whether it be all zeros or ones or whatever else we decide. We lose the randomness. This causes a flow of entropy. The entropy in the register goes to the environment due to the second law of thermodynamics. Since entropy cannot decrease, the decrease in entropy of the system must correspond to atleast an equal increase in the entropy of the environment. This means that the erasure costs some energy and it is nicely captured by *Landauer's principle*:

> **Landauer's principle**: *Suppose a computer erases a single bit of information. The amount of energy dissipated into the environment is at least $k_BT\,ln\,2$, where k_B is a universal constant known as Boltzmann's constant, and T is the temperature of the environment of the computer.*

This is due to the well-known result : $\Delta Q = T\Delta S$. Here $k_B\,ln\,2$ is the change in entropy as a single bit has 2 microstates. This result is known as the "Exorcism of Maxwell's Daemon". But, how much does this principle actually impact computation that we use around ourselves today in our daily work? Modern computers actually consume something like 500 $k_BT\,ln\,2$ joules per bit erased. It might not seem like very much. But we erase a lot of bits all the time in a modern processor. A DNA computer uses about 2 ATPs per bit erasure, and that is approximately 120 $k_BT\,ln\,2$. So, our computers are not that far away from biological systems. There's only a factor of 4. Can we do better? Why do we need to expend energy like this? Could we build a computer that doesn't need to erase? If we could build a computer that did not need to erase information, maybe it would not need to dissipate energy to do the computation. And in fact, we could actually achieve it in principle by reversible computation.

2.6 Reversible computation and the billiard ball model

Reversible computing was born historically as a topic motivated by the connections between physics and computation. The key question here is ”Is $k_BT \, ln \, 2$ fundamental?”. Consider the AND gate. If we look at the output, we see that the AND gate is irreversible. And that means if we had random inputs coming in, we have less randomness in the output. This loss of entropy means that the AND gate always costs us energy to operate. But can we build a computation without using AND gates, i.e. without this kind of a cost? The answer is yes.



Input		Output
A	B	Y=A.B
0	0	0
0	1	0
1	0	0
1	1	1

Figure 6: Truth Table of AND Gate

And that is by showing that we can build a computer out of just billiard balls knocking together and hitting each other. This is a mechanical model of computation. And yet it is so inspiring to look at this model because all it requires is the laws of Newtonian mechanics. The system is assumed to be frictionless. We can build logic gates out of balls moving around and colliding. Consider the following figure.

Let’s call the absence or presence of the top ball A and the absence or presence of bottom ball B. Now the two balls will collide(if both are present), because they have a finite radius and subject to the initial positions and velocities. And then they will bounce off and go to certain final positions as shown in the figure. And there is a logical assignment to the final positions and it is easy to see that they correspond to what is shown in the figure. So, we have a gate here. The AND gate can be made by just looking at AB. The NOT gate can be made if we let $B = 1$ (always have the bottom ball) and checking $\bar{A}B$. Since AND and NOT gates are universal, we can have all computation by billiard balls, and note that it is reversible! For each distinct start configuration, there is a unique end configuration. Also, note that other things like swap operations, fanouts and crossovers can be implemented easily and I leave that as an exercise. Reflectors(hard walls) can be used too in this model.

By the way, nobody has actually built a physical billiard ball computer. Why doesn’t Intel build its computers out of balls banging around inside processors ? If we have a small error, a small misalignment, a small mistiming of a ball that went a little too fast or started a little too late, then we have to correct for that error. Because otherwise the error will just cascade and

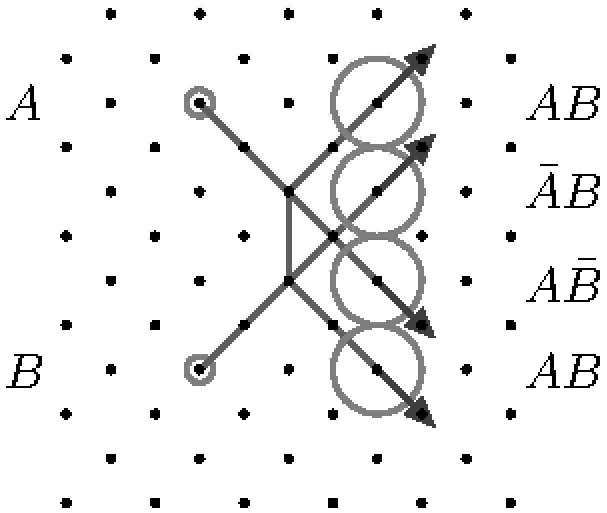


Figure 7: Billiard ball model

grow into bigger and bigger errors. And pretty soon the balls will all be moving chaotically all over the place. It turns out the only reason we need energy for our computers is because we need our computer to be reliable, to correct errors. Because otherwise, at least from a fundamental principle standpoint, we do not need to consume energy for computation.

2.7 Reversible gates and circuits

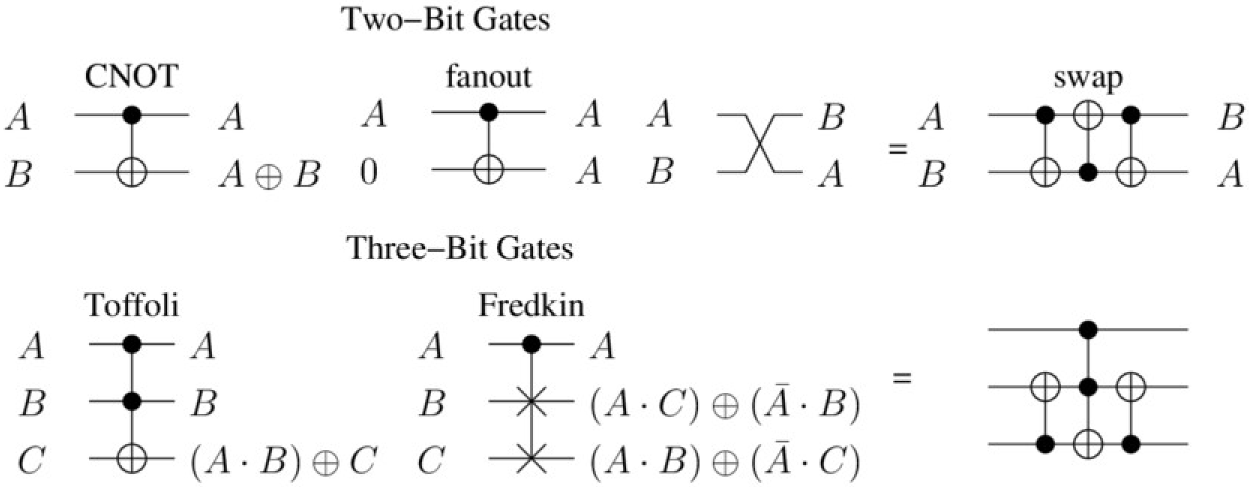


Figure 8: Reversible Gates

The CNOT stands for controlled-NOT. Note that the $\oplus$ is an XOR(exclusive OR), or addition modulo 2. The Fredkin gate can be thought of as swapping the bottom bits if the top bit is 1, else leaving them unchanged. We discuss some important theorems about these gates.

The Toffoli gate is universal. That's easy to see by construction. The AND gate can be constructed by Toffoli($A, B, 0$) = (A, B, AB) and the NOT gate can be constructed by Toffoli($A, 1, 1$) = ($A, 1, \bar{A}$). Since AND and NOT are universal, the Toffoli gate is universal. Similarly, the Fredkin gate is universal. Again, the AND gate can be constructed by Fredkin($A, B, 0$) = ($A, \bar{A}B, AB$) and the NOT gate can be constructed by Fredkin($A, 1, 0$) = ($A, \bar{A}, A$).

Note that in the above implementations of AND and NOT, there are extra output bits. These are called garbage. However, there's a theorem that the amount of garbage needed to reversibly simulate any Boolean circuit is of the order of the width of the circuit and not the depth. So the amount of garbage can be a controlled amount. And hand-in-hand with this theorem is another theorem that says that any Boolean circuit of n gates can be simulated by a reversible circuit of order of polynomial in n reversible gates, approximately $O(n^2)$ gates. We will not prove these theorems.

An important consequence of universality of reversible gates is that any computation can be done reversibly. But we don't do that. And largely, we do not do it today because of the need to tolerate errors and become reliable. And so this is also important to quantum computing, because quantum computers fundamentally need to be reversible due to the laws of quantum mechanics, as we will see next. We'll see how reversible computation subsumes normal classical computation, and because quantum mechanics subsumes reversible computation, it also subsumes all of computation today. But the laws of quantum mechanics are the laws of physics, and therefore, the laws of physics govern computation. Isn't that a great thing?

3 Quantum Mechanics I – Qubits

3.1 Quantum states

The quantum state space of a system is a complex vector space. A quantum state is a unit vector in quantum state space. A qubit is a system whose state space is 2-dimensional. Some examples are:

1. The polarization of a photon has two basis states : horizontal ($|\leftrightarrow\rangle$) and vertical ($|\updownarrow\rangle$). And they can be distinguished by using polarizers.

2. Spin $\frac{1}{2}$ half particles have two basis states: up ($|\uparrow\rangle$) and down ($|\downarrow\rangle$).

3. The harmonic oscillator has the basis states :$|0\rangle, |1\rangle, |2\rangle, \ldots$ It's quantum space is infinite dimensional vector space, whose basis states are denoted by whole numbers.

Note that we use the Dirac notation in which $|v\rangle$ denotes a column vector v, and is called a "ket", while $\langle v|$ denotes the adjoint(conjugate transpose) of v, and is called a "bra".

For example consider a 4-dimensional state space. Then a quantum state $|v\rangle$ of this system is $\begin{bmatrix} 0.5 \\ 0.5 \\ -0.7i \\ 0.1 \end{bmatrix}$ and $\langle v|$ of this state is $\begin{bmatrix} 0.5 & 0.5 & 0.7i & 0.1 \end{bmatrix}$.

Also note that $\langle w|v\rangle$ denotes the inner product between states w and v. And, by definition $\langle v|v\rangle$ is 1

Let's consider the example of polarization of a photon. We can have many polarization states:

1. Diagonal polarization : $\frac{1}{\sqrt{2}}(|\leftrightarrow\rangle + |\updownarrow\rangle)$
2. Other-Diagonal polarization: $\frac{1}{\sqrt{2}}(|\leftrightarrow\rangle - |\updownarrow\rangle)$
3. Right Circular polarization: $\frac{1}{\sqrt{2}}(|\leftrightarrow\rangle - i\,|\updownarrow\rangle)$
4. Plane Polarization at angle θ to horizonatal : $\cos\theta\,|\leftrightarrow\rangle + \sin\theta\,|\updownarrow\rangle$

The quantum state space we will be using most of the time is the qubit and we will use $|0\rangle$ and $|1\rangle$ as the basis states for a qubit.

Let's consider a spin $\frac{1}{2}$ particle. Some other spin states defined by convention are

1. Spin right: $|\rightarrow\rangle = \frac{1}{\sqrt{2}}(|\uparrow\rangle + |\downarrow\rangle)$
2. Spin left: $|\leftarrow\rangle = \frac{1}{\sqrt{2}}(|\uparrow\rangle - |\downarrow\rangle)$
3. Spin inwards: $|\otimes\rangle = \frac{1}{\sqrt{2}}(|\uparrow\rangle + i\,|\downarrow\rangle)$
4. Spin outwards: $|\odot\rangle = \frac{1}{\sqrt{2}}(|\uparrow\rangle - i\,|\downarrow\rangle)$

Multiplying a quantum state by a global phase does not change the essential nature of a quantum state. But it needs to be a global phase. For example, $|\downarrow\rangle$ is same as $i\,|\downarrow\rangle$, but

$$|\rightarrow\rangle = \frac{1}{\sqrt{2}}(|\uparrow\rangle + |\downarrow\rangle) \neq \frac{1}{\sqrt{2}}(|\uparrow\rangle + i\,|\downarrow\rangle) = |\otimes\rangle$$

3.2 Bloch sphere and Pauli matrices

We'll be working a lot with qubits. So it really helps to understand the geometry of qubits and how they transform, which can be represented by the Bloch sphere. A unit vector in two dimensional complex vector space, whose global phase does not matter is equivalent to a two dimensional real vector space, because we have removed two degrees of freedom from the complex vector space. The surface of a sphere can be used to represent two dimensional real vector space, and hence we can assign a unique point on the sphere corresponding to each qubit state.

We can see that the points on the Bloch sphere along the axes are exactly analogous to spin states, considering $|\uparrow\rangle = |0\rangle$ and $|\downarrow\rangle = |1\rangle$. It is easy to see that a general qubit state can be written as

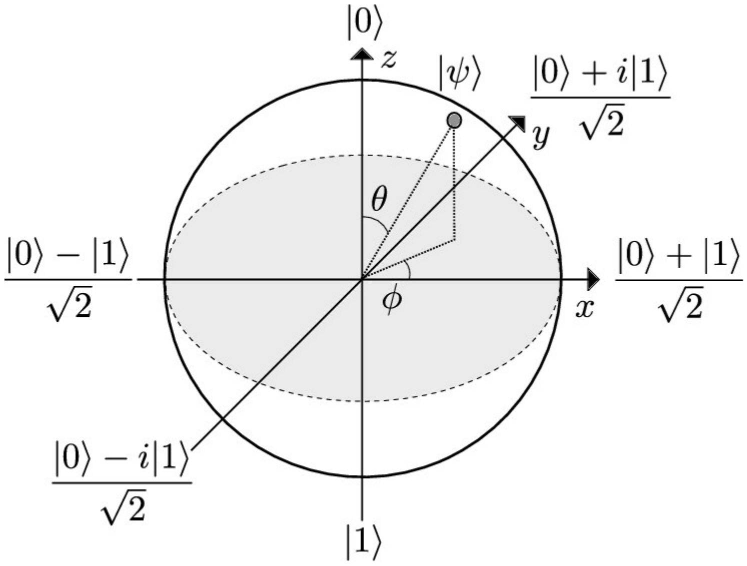

$|0\rangle$
z
$|\psi\rangle$
$\frac{|0\rangle + i|1\rangle}{\sqrt{2}}$
y
θ
$\frac{|0\rangle - |1\rangle}{\sqrt{2}}$
$\frac{|0\rangle + |1\rangle}{\sqrt{2}}$
ϕ
x
$\frac{|0\rangle - i|1\rangle}{\sqrt{2}}$
$|1\rangle$

Figure 9: Bloch Sphere

$$|\psi\rangle = \cos\frac{\theta}{2}|0\rangle + e^{i\phi}\sin\frac{\theta}{2}|1\rangle$$

The corresponding point on the Bloch sphere is represented in Fig. 9. θ is the angle from z-axis and ϕ is the angle of the projection on xy plane from x-axis.

The Pauli matrices would be very useful in our further discussion. They are given in the figure below.

$$\sigma_0 \equiv I \equiv \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad \sigma_1 \equiv \sigma_x \equiv X \equiv \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

$$\sigma_2 \equiv \sigma_y \equiv Y \equiv \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \qquad \sigma_3 \equiv \sigma_z \equiv Z \equiv \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

Figure 10: The Pauli matrices.I is included for the sake of completeness, but we will not consider it as a Pauli matrix, unless stated otherwise.

So these are all 2×2 matrices, which means they perform transformations on a two-dimensional state space. Let's try to understand what transformations they perform. Consider X.

$$X\begin{bmatrix}0\\1\end{bmatrix} = \begin{bmatrix}1\\0\end{bmatrix} \quad X\left(\frac{1}{\sqrt{2}}\begin{bmatrix}1\\1\end{bmatrix}\right) = \frac{1}{\sqrt{2}}\begin{bmatrix}1\\1\end{bmatrix} \quad X\left(\frac{1}{\sqrt{2}}\begin{bmatrix}1\\i\end{bmatrix}\right) = \frac{1}{\sqrt{2}}\begin{bmatrix}i\\1\end{bmatrix} = \frac{1}{\sqrt{2}}\begin{bmatrix}1\\-i\end{bmatrix}$$

Note that in the last step we have multiplied by a unit scalar $-i$, as global phases do not change a state.What does X do to a qubit state? It leaves the states on the x-axis unchanged and rotates the states on y and z axis by 180° about x-axis. Hence, X is equivalent to 180° rotation about x-axis. Similarly, Y and Z are equivalent to 180° rotations about y and z axis respectively.In fact, all single qubit gates are rotations on the Bloch sphere

Let's consider another matrix $H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$.

$$H \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad H \left(\frac{1}{2} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right) = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad H \left(\frac{1}{2} \begin{bmatrix} 1 \\ i \end{bmatrix} \right) = \frac{1}{\sqrt{2}} \begin{bmatrix} 1+i \\ 1-i \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -i \end{bmatrix}$$

Note that in the last step we have multiplied by a unit scalar $\frac{1}{\sqrt{2}}(1-i)$, as global phases do not change a state. From the above results we conclude that H is equivalent to 180° rotation around an axis making an angle of 45° with z-axis in xz plane.

We have already seen the following gates : X, Y, Z, H. There are two more important gates :

$$S = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix} \qquad T = \begin{bmatrix} 1 & 0 \\ 0 & e^{\frac{i\pi}{4}} \end{bmatrix}$$

Note that $S^2 = Z$ and $T^2 = S$

Some important properties of the Pauli matrices are given below, proofs of which are left as an exercise.

1. All Pauli matrices have two eigenvalues, +1 and -1. The corresponding eigenvectors are:

$$\begin{aligned} \psi_{x+} &= \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix}, & \psi_{x-} &= \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix}, \\ \psi_{y+} &= \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ i \end{pmatrix}, & \psi_{y-} &= \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -i \end{pmatrix}, \\ \psi_{z+} &= \begin{pmatrix} 1 \\ 0 \end{pmatrix}, & \psi_{z-} &= \begin{pmatrix} 0 \\ 1 \end{pmatrix}. \end{aligned}$$

2. ϵ_{jkl}, the antisymmetric tensor on three indices, for which $\epsilon_{jkl} = 0$ except for $\epsilon_{123} = \epsilon_{231} = \epsilon_{312} = 1$, and $\epsilon_{321} = \epsilon_{213} = \epsilon_{132} = -1$. The commutator of A and B is $[A, B] = AB - BA$. The commutation relations between Pauli matrices for $j, k = 1, 2, 3$ can be written as :

$$[\sigma_j, \sigma_k] = 2i \sum_{l=1}^{3} \epsilon_{jkl} \sigma_l$$

3. The Pauli matrices anti-commute, i.e. $\{\sigma_j, \sigma_k\} = 0$ where $j \neq k$ and $j, k = 1, 2, 3$. The anti-commutator of A and B is $\{A, B\} = AB + BA$.

4. $\sigma_i^2 = I$ for $i = 1, 2, 3$

5. $\sigma_j \sigma_k = \delta_{jk} I + i \sum_{l=1}^{3} \epsilon_{jkl} \sigma_l$ for $j, k = 1, 2, 3$. Here δ_{jk} is the Kronecker delta, where $\delta_{jk} = 0$ except when $j = k$, where $\delta_{jk} = 1$.

3.3 Unitary transformations

A unitary transformation is defined as one which maps complex unit vectors to complex unit vectors. Consider an orthonormal basis $|v_1\rangle, |v_2\rangle, \dots, |v_n\rangle$ of an n-dimensional vector space. Let U be a unitary operator. Let $U|v_i\rangle = |w_i\rangle$ for all i. If we write U in matrix form, its i^{th} column is clearly $|w_i\rangle$. Hence, an elegant representation of $U = \sum_i |w_i\rangle\langle v_i|$.

Applying U to $\frac{1}{\sqrt{2}}(|v_i\rangle + |v_j\rangle)$ and $i \neq j$, we get $\frac{1}{\sqrt{2}}(|w_i\rangle + |w_j\rangle)$. Since U maps unit vectors to unit vectors, the norm of the obtained vector must be 1.

$$\begin{aligned}\frac{1}{\sqrt{2}}(\langle w_i| + \langle w_j|) \cdot \frac{1}{\sqrt{2}}(|w_i\rangle + |w_j\rangle) &= \frac{1}{2}(\langle w_i|w_i\rangle + \langle w_j|w_j\rangle + \langle w_i|w_j\rangle + \langle w_j|w_i\rangle) \\ 1 &= 1 + \frac{1}{2}\langle w_i|w_j\rangle + \langle w_i|w_j\rangle^*)\end{aligned}$$

Clearly, this implies real part of $\langle w_i|w_j\rangle$ is 0. Following the same procedure, with $\frac{1}{\sqrt{2}}(|v_i\rangle + i|v_j\rangle)$, we get imaginary part of $\langle w_i|w_j\rangle$ is 0. Hence $\langle w_i|w_j\rangle = 0$. We conclude that $|w_i\rangle$ are orthonormal, and hence the matrix representation of U has orthonormal columns. Hence, U maps one basis to another basis, and conversely any two bases are related by a unitary transformation.

For every operator A, there exists a unique operator $A^\dagger$, such that for all$|v\rangle, |w\rangle, \langle v|Aw\rangle = \langle A \dagger v|w\rangle$. This operator is called the adjoint of A. It is easy to see that if A is a matrix, then $A^\dagger$ is the complex -conjugate of A^T. Unitary transforms have the property that $U^\dagger = U^{-1}$, and the proof is simple. It follows from the fact that $|v_i\rangle$ and $|w_i\rangle$ form orthonormal basis and $(|w\rangle\langle v|)^\dagger = |v\rangle\langle w|$

$$\begin{aligned}U^\dagger U &= (\sum_i |v_i\rangle\langle w_i|)(\sum_j |w_j\rangle\langle v_j| \\ &= \sum_i\sum_j |v_i\rangle\langle w_i|w_j\rangle\langle v_j| \\ &= \sum_i\sum_j \delta_{ij}|v_i\rangle\langle v_j| \\ &= \sum_i |v_i\rangle\langle v_i| \\ &= I\end{aligned}$$

A corollary of this result is that unitary transformations preserve inner products. Because $(U|v\rangle, U|w\rangle) = \langle v|U^\dagger U|w\rangle = \langle v|I|w\rangle = \langle v|w\rangle$.

To summarise, we have the following three equivalent definations of unitary transformation:

1. A transformation which maps any unit vector to another unit vector.
2. A transformation whose matrix form has all columns/rows orthonormal.
3. A transformation whose adjoint is equal to its inverse.

Note that the Pauli matrices are hermitian and unitary.

3.4 The Copenhagen interpretation of quantum mechanics

The Copenhagen interpretation of quantum mechanics is due to Niels Bohr. The Copenhagen interpretation is really good at explaining how to calculate with quantum mechanics. It says that quantum mechanical systems undergo two kinds of evolutions. Isolated systems undergo unitary evolution. The other kind is when the experimenter, or the universe, ”looks” or interacts at the system and that is called measurement.

3.5 Rotations on the Bloch sphere

For arbitrary rotations, we need to know how to compute exponentials of matrices. We define $e^A = \sum_{i=0}^{\infty} \frac{A^i}{i!}$. For example, let us compute $e^{i\theta\vec{v}\cdot\vec{\sigma}}$. Note that here $\vec{v}$ is a three dimensional real unit vector and $\vec{v}\cdot\vec{\sigma} = \sum_{i=1}^{3} v_i\sigma_i$. We will use the fact that $(\vec{v}\cdot\vec{\sigma})^2 = I$, whose verification is left as an exercise.

$$\begin{aligned}
e^{i\theta\vec{v}\cdot\vec{\sigma}} &= \sum_{j=0}^{\infty} \frac{(i\theta\vec{v}\cdot\vec{\sigma})^j}{j!} \\
&= \sum_{j=0}^{\infty} (-1)^j \frac{\theta^{2j}}{(2j)!} I + i \sum_{j=0}^{\infty} (-1)^j \frac{\theta^{2j+1}}{(2j+1)!} \vec{v}\cdot\vec{\sigma} \\
&= \cos(\theta) I + i\sin(\theta)\vec{v}\cdot\vec{\sigma} \qquad (1)
\end{aligned}$$

This identity is used for computing the rotation operators about the x, y, z axes on the Bloch sphere by angle θ, which are defined below:

$$\begin{aligned}
R_x(\theta) &\equiv e^{-i\theta X/2} = \cos\frac{\theta}{2} I - i\sin\frac{\theta}{2} X = \begin{bmatrix} \cos\frac{\theta}{2} & -i\sin\frac{\theta}{2} \\ -i\sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{bmatrix} \\
R_y(\theta) &\equiv e^{-i\theta Y/2} = \cos\frac{\theta}{2} I - i\sin\frac{\theta}{2} Y = \begin{bmatrix} \cos\frac{\theta}{2} & -\sin\frac{\theta}{2} \\ \sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{bmatrix} \\
R_z(\theta) &\equiv e^{-i\theta Z/2} = \cos\frac{\theta}{2} I - i\sin\frac{\theta}{2} Z = \begin{bmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{bmatrix}.
\end{aligned}$$

We can check this. For example

$$R_z(\alpha)\left|\psi\right\rangle = \begin{bmatrix} e^{\frac{-i\alpha}{2}} & 0 \\ 0 & e^{\frac{i\alpha}{2}} \end{bmatrix} \begin{bmatrix} \cos\frac{\theta}{2} \\ e^{i\phi}\sin\frac{\theta}{2} \end{bmatrix} = \begin{bmatrix} e^{\frac{-i\alpha}{2}}\cos\frac{\theta}{2} \\ e^{i(\phi+\frac{\alpha}{2})}\sin\frac{\theta}{2} \end{bmatrix} = \begin{bmatrix} \cos\frac{\theta}{2} \\ e^{i(\phi+\alpha)}\sin\frac{\theta}{2} \end{bmatrix}$$

In the last step, we have multiplied by the global phase $e^{i\frac{\alpha}{2}}$. Note that, on the Bloch sphere the azimuthal angle ϕ has increased by α, verifying the correctness of $R_z(\theta)$. Similarly, we can verify the operators for x and y axes.

In general, if $\hat{n}$ is a real unit vector in three dimensions then we generalize the previous definitions by defining a rotation by θ about the $\hat{n}$ axis by the equation:

$$R_{\hat{n}}(\theta) = e^{-i\theta\hat{n}\cdot\vec{\sigma}/2} = \cos(\frac{\theta}{2})I - i\sin(\frac{\theta}{2})\hat{n}\cdot\vec{\sigma}$$

The proof is simple, but involves some algebra, which can be done easily by the properties of the Pauli matrices. Let $\hat{n}$ correspond to the point (θ, ϕ) on the Bloch sphere. Rotation of α about $\hat{n}$ can be achieved in three steps:

1. Rotate about the z-axis by $-\phi$ and then about the y-axis by $-\theta$. By this $\hat{n}$ reaches the position $(0, 0)$, i.e. on the z-axis.

2. Rotate about the z-axis by α. This is the desired rotation about $\hat{n}$.

3. To restore $\hat{n}$ to its original position, rotate about the y-axis by θ and then about the y-axis by ϕ

Hence, $R_{\hat{n}}(\alpha) = R_z(\phi)R_y(\theta)R_z(\alpha)R_y(-\theta)R_z(-\phi)$. Simplifying this is left as an exercise.

All single qubit gates must be unitary, as they take a unit vector to a unit vector. I want to claim that any unitary 2×2 matrix, upto a global phase, can be constructed out of three rotations , first about z-axis, then about x-axis and z-axis. To prove this, first focus on the north pole. Let's move it to the right position. To do that, rotate about the x-axis to get the "latitude" right, then rotate about the z-axis to get the longitude right. Finally, there can be some rotation about the north pole too. To do that, in the beginning rotate the Bloch sphere to acheive the desired rotation. I have provided a very intuitive argument to make it digestible. It's proof is not too hard, and is left as an exercise. The general representation of a 2×2 unitary matrix will be useful for the proof.

$$U = e^{i\alpha}\begin{bmatrix} \cos\theta & e^{i\phi_2}\sin\theta \\ e^{i\phi_1}\sin\theta & -e^{i(\phi_1+\phi 2)}\sin\theta \end{bmatrix}$$

3.6 Towards multi-qubit states and gates

We have discussed single qubit gates. But, of course using only single qubit gates does not allow communication and interaction between qubits and hence multi-qubit gates are necessary. To actually get any kind of quantum computation, we need to have some gate that links atleast two qubits, so that they "talk" to each other. So, what is the state space of 2 qubits? It is a 4 dimensional state space, and similarly for n-qubits it is a 2^n dimensional space. And these are built by "tensor-products" between single qubit states, which is the topic of next section. Also, we need to do measurements on the system. Unitary operations allow us to change the quantum system, but we need to get the information out of the system, and measurements allow us to do this thing. And measuring it isn't the same as writing down it's state, because by Heisenberg's uncertainty principle, the complete state of a quantum system can never be found. We can only measure some properties of it.

4 Quantum Mechanics II – Measurement and Tensor products

4.1 Brief introduction to quantum measurements

Orthogonal states are distinguishable. So for example, $|0\rangle$ and $|1\rangle$ are distinguishable. What does it mean for two things to be distinguishable? It means there's some measurement that distinguishes between them. so, for example if we apply the measurement which distinguishes $|0\rangle$ and $|1\rangle$ on a state $|\psi\rangle = a\,|0\rangle + b\,|1\rangle$, we get $|0\rangle$ with probability $|a|^2$ and $|1\rangle$ with probability $|b|^2$. In the general case, we will use a special case of the measurement axiom of quantum mechanics.

Let us have a n-dimensional state space, with orthonormal basis $|v_1\rangle, |v_2\rangle, \ldots, |v_n\rangle$. Let $|\psi\rangle = \sum_{i=1}^{n} c_i\,|v_i\rangle$. Then a measurement in that basis, yields $|v_i\rangle$ with probability $|c_i|^2$. And since $|\psi\rangle$ is a unit vector, $\sum_{i=1}^{n} |c_i|^2 = 1$, which should be since we always get one of the basis states, and hence sum of probabilities should be 1. What happens if we measure in an arbitrary normal basis, $|w_1\rangle, |w_2\rangle, \ldots, |w_n\rangle$? The probability of obtaining $|w_i\rangle$ is $\left|\langle w_i|\psi\rangle\right|^2$. Note that the system's state changes to the measured state after the measurement.

4.2 Tensor products

Suppose we have two qubits. Then it's basis is $|00\rangle, |01\rangle, |10\rangle, |11\rangle$. Note that, $|00\rangle$ is a shorthand for $|0\rangle\,|0\rangle$ or $|0\rangle \otimes |0\rangle$, where $\otimes$ denotes the Tensor product between first qubit state and second qubit state. We know that a single qubit has two degrees of freedom. Similarly, two qubits have $2 \times 4 - 2 = 6$ degrees of freedom. Because for each basis vector, we have a complex coefficient giving two degrees of freedom each, and the constraint of normality and irrelevancy of global phase takes away 2 degrees of freedom. In general, for n qubits, we have $2^{n+1} - 2$ degrees of freedom.

What are the implications of these degrees of freedom? There are 4 degrees of freedom if we take the product of 2 qubits, each of which is in its own state, and there are 6 degrees of freedom if we look at the general state space of two qubits. So that means there are some states of two qubits which cannot be factored into 2 separate qubit states. We define a separable state as one in which each qubit is in its own definite state. For 2 qubits, the state can be written as

$$(a_1\,|0\rangle + b_1\,|1\rangle) \otimes (a_2\,|0\rangle + b_2\,|1\rangle) = a_1a_2\,|00\rangle + a_1b_2\,|01\rangle + b_1a_2\,|10\rangle + b_1b_2\,|11\rangle$$

A state which is not separable is called an entangled state. For example, $\frac{1}{\sqrt{2}}(|01\rangle - |10\rangle)$ is entangled, whose proof can be done by using the above representation of separable state, and is left as an exercise.

How is the inner product defined for tensors? For arbitrary tensors $|\psi\rangle = |a\rangle \otimes |v\rangle$ and $|\phi\rangle = |b\rangle \otimes |w\rangle$, $\langle\psi|\phi\rangle = \langle a|b\rangle\,\langle v|w\rangle$. The usual linear properties of the inner products remain the same. For example for two qubits, let $|\psi\rangle = \sum_{i=00}^{11} a_i\,|i\rangle$, and $|\phi\rangle = \sum_{i=00}^{11} b_i\,|i\rangle$, then $\langle\psi|\phi\rangle = \sum_{i=00}^{11} a_i^* b_i$.

We now prove the no-cloning theorem. It states that we cannot convert an unknown quantum state into two copies of this unknown quantum state. The proof is by contradiction. Suppose, we can achieve the cloning. We start with an unknown state $|\psi\rangle$ and an empty register, whose state we call $|0\rangle$, in which we would clone our second copy. This cloning is a unitary evolution, according to copenhagen interpretation. We can do a similar cloning for $|\phi\rangle$. We get the following equations:

$$\begin{aligned} U(|\psi\rangle|0\rangle) &= |\psi\rangle|\psi\rangle \\ U(|\phi\rangle|0\rangle) &= |\phi\rangle|\phi\rangle \end{aligned}$$

Now, unitary transformations preserve inner products. Hence , $\langle\psi|\phi\rangle\langle 0|0\rangle = \langle\psi|\phi\rangle\langle\psi|\phi\rangle$. Denoting, $\langle\psi|\phi\rangle$, by x, we get $x = x^2$. This is satisfied only if $x = 0$ or $x = 1$. Phsyically, this means that such a cloner could clone two distinguishable(orthogonal) states, and hence a general cloning device is impossible. But if we have two distinguishable states, we can measure the distinguishing observable. And then we can make as many copies as we want.

Tensor products are also defined on operators and matrices. If A and B are operators acting on $|v\rangle$ and $|w\rangle$. Then the operator $A \otimes B$ is defined as $(A \otimes B)(|v\rangle \otimes |w\rangle) = A|v\rangle \otimes B|w\rangle$. In fact $A \otimes B$ has a matrix representation, the proof of which is left as an exercise. Suppose A has a m by n matrix form, and B has a p by q matrix form, then $A \otimes B$ is an mp bynq matrix given by:

$$\begin{bmatrix} A_{11}B & A_{12}B & \dots & A_{1n}B \\ A_{21}B & A_{22}B & \dots & A_{2n}B \\ \vdots & \vdots & \vdots & \vdots \\ A_{m1}B & A_{m2}B & \dots & A_{mn}B \end{bmatrix}$$

Let's see some examples.

$$H \otimes I = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix} \quad I \otimes H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 \end{bmatrix} \quad H \otimes H = \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

Note that the H stands for Hadamard. What H does is it maps $|0\rangle, |1\rangle$ basis to the $|+\rangle, |-\rangle$ basis(called the Hadamard basis) and vice-versa, where $|\pm\rangle = \frac{1}{\sqrt{2}}(|0\rangle \pm |1\rangle)$. So for example, to write the entangled state $\frac{1}{\sqrt{2}}(|01\rangle - |10\rangle)$ in the $|\pm\rangle$ basis, we have to apply $H \otimes H$ to the state $\frac{1}{\sqrt{2}}(|+-\rangle - |-+\rangle)$ as $|0\rangle = H|+\rangle$ and $|1\rangle = H|-\rangle$.

$$(H \otimes H)\left(\frac{1}{\sqrt{2}} \begin{bmatrix} 0 \\ 1 \\ -1 \\ 0 \end{bmatrix}\right) = \frac{1}{2\sqrt{2}} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ -1 \\ 0 \end{bmatrix} = -\frac{1}{\sqrt{2}} \begin{bmatrix} 0 \\ 1 \\ -1 \\ 0 \end{bmatrix}$$

That means, $\frac{1}{\sqrt{2}}(|01\rangle - |10\rangle) = -\frac{1}{\sqrt{2}}(|+-\rangle - |-+\rangle)$. Hence, the state is the same entangled state in both bases, upto the global phase of -1.

Note that this method of applying single qubit gates to multiple qubits with tensor products is more algebraically involved, and instead individual gates should be applied to the qubits. For example,

$$(H \otimes I)(\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) = \frac{1}{\sqrt{2}}(H|0\rangle \otimes I|0\rangle + H|1\rangle \otimes I|1\rangle) = \frac{1}{2}(|00\rangle + |10\rangle + |01\rangle - |11\rangle)$$

Is every transformation we apply to multiple qubits a tensor product? No, any transformation need to be only unitary for it to be valid. And, in fact, if every operator was a tensor product, then we couldn't really do quantum computation. Because if we start with a tensor product of qubits and apply tensor product of unitary operators to each of them,we will never get two qubits to talk to each other.

4.3 Gates for quantum circuits

Recall that AND and NOT gates are universal for classical computation. But, this is not quite true, because there is an implicit FANOUT gate, which is used for copying the bits. For, quantum circuits, the equivalent of FANOUT is cloning. But cloning is not allowed,a nd hence FANOUT cannot be used in quantum circuits.

Let us see the CNOT gate. Recall that CNOT(x,y) = CNOT(x,x $\oplus$ y), and hence its matrix representation is:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

This is because, if x is 0, I is applied to the second qubit,as we can see in the top-left corner and if x is 1, X which performs the NOT operation is applied to y, as seen in the bottom right corner. Also, this is also a classical reversible gate, because it's just a permutation matrix and no phases or superposition is applied.

The notation for any such arbitrary controlled unitary U is:

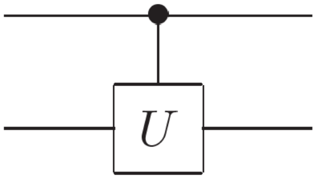


Figure 11: Controlled-U operation. The top line is the control qubit, and the bottom line is the target qubit. If the control qubit is 1 then U is applied to the target, otherwise it is left alone.

The matrix representation of controlled-U is $\begin{bmatrix} I & O \\ O & U \end{bmatrix}$, where I and O are 2 by 2 identity and zero matrices. The matrix can be asily checked to be unitary, so it is a valid gate.

Note that though CNOT can't copy qubits, it can copy classical bits because CNOT(x,0) = CNOT(x,x). So, to copy classical information in qauntum circuit, CNOT is used.

Two important gate identites are:

1. $HXH = Z, HZH = X, HYH = -Y$. The proof can be done by straightforward multiplication, but is easier if you recall that H just changes standard basis to Hadamard basis and $H^{-1} = H$. Thus, the expressions on the left are just changing the basis of the Pauli matrices to Hadamard basis. For example, X takes $|+\rangle$ to $|+\rangle$ and $|-\rangle$ to $-|-\rangle$, thus behaving like the Z gate.

2. The CNOT gate is flipped in the Hadamard basis:

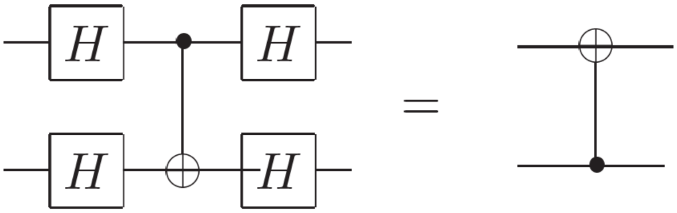


Figure 12: CNOT in Hadamard basis. The target and control qubits are exchanged.

Again this can be seen as a change of basis, and can be proved by seeing the effects of the CNOT gate in Hadamard basis. For example, $|+-\rangle$ goes to $|--\rangle$ as CNOT(+,-) = $\frac{1}{\sqrt{2}}$((0,-) -(1,-) = (-,-). Similarly, it can be checked for other input states. Physically, this identity means that the control and target qubits are arbitrary, and depend on the basis of measurement,

5 Quantum Mechanics III – Multi-qubit measurements

5.1 Combining measurement and tensor products

Suppose we have the state $\frac{1}{\sqrt{3}}(|00\rangle + |01\rangle + |11\rangle)$. If we measure both qubits in the standard basis, we get $|00\rangle, |01\rangle, |11\rangle$ with probability $\frac{1}{3}$ each and $|10\rangle$ with probability 0. But, now suppose we measure only the first qubit in the standard basis. Note that the state can be written as $\sqrt{\frac{2}{3}}|0\rangle(\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)) + \frac{1}{\sqrt{3}}|1\rangle|1\rangle$. After measurement of the first qubit, we get $|0\rangle$ with probability $\frac{2}{3}$, and the second qubit would be in state $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$. Also, we get $|1\rangle$ with probability $\frac{1}{3}$, and the second qubit would be in state $|1\rangle$.

In general, consider an orthonormal basis $|v_1\rangle, |v_2\rangle, \ldots, |v_n\rangle$ of the first qubit and $|\psi\rangle$ be a two qubit state. Then, $|\psi\rangle = \sum_{i=1}^{n} c_i |v_i |w_i\rangle\rangle$, where $|w_i\rangle$ are quantum states of the second qubit, then we get $|v_i\rangle$ with probability$|c_i|^2$, and the state of the second qubit after measurement is $|w_i\rangle$. Also, note that it does not measure whether we measure both qubits simultaneously, or one qubit after the another as the probability distributions of each qubit are same.

There is another method of measurement using ”partial inner products”. The partial inner product between $|a\rangle$ and $|b\rangle|v\rangle$ is defined as $\langle a|b\rangle|v\rangle$. Suppose, we measure the first qubit in the $|\pm\rangle$ basis. Let's compute the probability of obtaining $|+\rangle$. First take the partial inner product of $|+\rangle$ with the state, i.e . $\frac{1}{\sqrt{6}}(\langle 0| + \langle 1|)(|00\rangle + |01\rangle + |11\rangle) = \frac{1}{\sqrt{6}}(|0\rangle + 2|1\rangle) = \sqrt{\frac{5}{6}}\left[\frac{1}{\sqrt{5}}(|0\rangle + 2|1\rangle)\right]$. Hence, the probability of obtaining $|+\rangle$ is $\frac{5}{6}$ and the state of the second qubit if $|+\rangle$ is obtained is $\frac{1}{\sqrt{5}}(|0\rangle + 2|1\rangle)$.

5.2 Properties of Unitary and Hermitian matrices

Recall that a unitary matrix satisfies $U^\dagger = U^{-1}$ and a hermitian matrix satisfies $M^\dagger = M$. Note that the Pauli matrices are both unitary and hermitian. An important property of both hermitian and unitary matrices of order$n$ is that they both have n orthonormal eigenvectors. Such a decomposition of the matrix in its eigenvector basis is called *spectral decomposition*, and its proof is left as an (hard) exercise. So, a Hermitian matrix M can be written as $M = \sum_i \lambda_i |i\rangle\langle i|$, where λ_i are its eigenvalues and $|i\rangle$ are the corresponding eigenvectors. Note that the eigenvalues are real, because $\langle i|M|i\rangle = \lambda_i$ and taking the complex conjugate of the whole equation, we get $\langle i|M^\dagger|i\rangle = \lambda_i^*$. But since $M^\dagger = M$, $\lambda_i = \lambda_i^*$ and thus, the eigenvalues are real. Similarly a unitary matrix U can be written as $U = \sum_j e^{i\theta_j}|j\rangle\langle j|$, because a unitary matrix can have only unit eigenvalues, as it maps unit vectors to unit vectors. Interestingly, since the Pauli matrices are both hermitian and unitary, they can only have real unit eigenvalues, i.e. ± 1.

5.3 Measurements and Hermitian operators

There is a standard way of describing measurements in quantum mechanics called projective measurements. Any observable M is a Hermitian matrix, and hence $M = \sum_i \lambda_i P_i$, where P_i is just the projector onto the eigenspace of λ_i. Then, by measuring the observable M on a state $|\psi\rangle$, we get λ_i with probability $p_i = \langle\psi|P_i|\psi\rangle$ and the resulting state would be $\frac{P_i|\psi\rangle}{\sqrt{p_i}}$. Also, the expectation value of M on $|\psi\rangle$ is $\langle\psi|M|\psi\rangle$. The proof is:

$$E(M) = \sum_i \lambda_i p_i = \sum_i \lambda_i \langle\psi|P_i|\psi\rangle = \langle\psi|\sum_i \lambda_i P_i|\psi\rangle = \langle\psi|M|\psi\rangle$$

An example of an observable is the spin along x-axis called $S_x = \frac{1}{2}X$. The eigenvalues are $+\frac{1}{2}$ and $-\frac{1}{2}$, with the corresponding projectors $|+\rangle\langle +|$ and $|-\rangle\langle -|$ respectively.

We have only talked about projective measurements, but there are other classes of measurements like the POVM measurements, which will be used much later, and will be discussed then.

We saw the spin for a single qubit, but what about two qubits. The spin of two qubits is defined as the spin of first qubit + the spin of second qubit. Consider the example of spin along

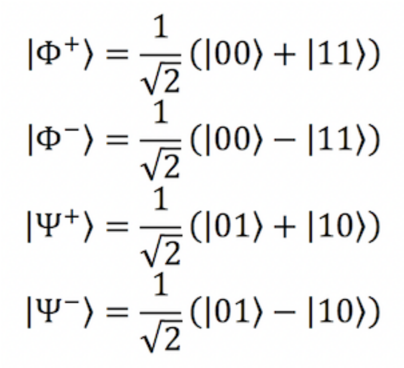
z-axis for two qubits :

$$S_z = \frac{1}{2}(Z \otimes I + I \otimes Z) = \frac{1}{2}\left(\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}\right) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$$

Clearly, the eigenvalues of S_z are 1, 0 and -1 (which was expected due independent addition of spins), and the corresponding eigenspaces are those spanned by $|00\rangle$, $(|01\rangle, |10\rangle)$ and $|11\rangle$.

For an arbitrary axis along an unit vector $\hat{n}$, the spin of a single qubit is is defined as $\frac{1}{2}\hat{n} \cdot \vec{\sigma}$. The operator $\hat{n} \cdot \vec{\sigma}$ has several interesting properties. Clearly it is Hermitian. Also, $(\hat{n} \cdot \vec{\sigma})^2 = I$, which is easy to see by breaking the operator into Pauli matrices and noting that the squares of Pauli matrices are I and they anti-commute. This property along with hermiticity imply that its eigenvalues are ± 1. But, its trace is 0 as traces of all Pauli matrices are 0 and hence the eigenvalues must be +1 and -1, because the sum of eigenvalues must be equal to the trace.

The Bell states would be quite ubiquitous in the following sections. They are all entangled and form a basis for the state space of two qubits.



$$|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$
$$|\Phi^-\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle)$$
$$|\Psi^+\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle)$$
$$|\Psi^-\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle)$$

Figure 13: The four Bell states

In particular, the last state is remarkable because the expectation value of spin along any axis on it is 0. First we show that its expectation value is 0 along the x, y, z axes. In fact, the state lies in the zero eigenspace of all three operators, For example, we can clearly see that since $|01\rangle$ and $|10\rangle$ form the basis for the 0 eigenspace of S_z, it is easily verifed for z axis. Verification for other axes is left as an exercise. Finally, note that $S_{\hat{n}} = \frac{1}{2}((\hat{n} \cdot \vec{\sigma}) \otimes I + I \otimes (\hat{n} \cdot \vec{\sigma})) = n_x S_x + n_y S_y + n_z S_z$. Hence, since the state is in 0 eigenspace of S_x, S_y, S_z, it is in the zero eigenspace of its linear combination, which completes the proof.

5.4 Cloning implies faster-than-light communication

Recall from the first section that if we could clone, then you could communicate faster than light. This was due to Herbert, who called this FLASH, for ”first laser amplified superluminal hookup". And the reason he thought we could clone is if we look at a high level description of how a laser works, it seems like it should be able to clone photons. It can't, but that apparently wasn't clear to any of the referees, because they couldn't figure out the flaw in his paper.

How does cloning allow faster than light communication? We take the state $\frac{1}{\sqrt{2}}(|01\rangle + |10\rangle)$ and distribute it between Alice and Bob. Alice measures her qubit in either the standard basis or Hadamard basis, while Bob always measures in the standard basis. The signal, sent by Alice is the measurement basis she measured her qubit.

Case 1: Alice measures in the $|0\rangle, |1\rangle$ basis. Then, Bob would have $|0\rangle$ and $|1\rangle$ each with probability $\frac{1}{2}$. He clones many copies of his qubit. He takes pairs of two qubits, and measures it in the standard basis and hence gets $|00\rangle$ and $|11\rangle$, both with probability $\frac{1}{2}$.

Case 2: Alice measures in the $|+\rangle, |-\rangle$ basis. Then, Bob would have $|+\rangle$ and $|-\rangle$ each with probability $\frac{1}{2}$. He clones many copies of his qubit. He takes pairs of two qubits and would have $|{+}{+}\rangle$ and $|{-}{-}\rangle$ with probability $\frac{1}{2}$ each. If he measures it in the standard basis, he would get $|00\rangle$, $|01\rangle$, $|10\rangle$ and $|11\rangle$, each with probability $\frac{1}{4}$.

Hence, Bob can know which basis Alice measured in, even if they are separated by any arbitrary large distance, and hence faster than light communication is acheived! In fact, a good exercise is to check that the two ensembles$|0\rangle$ and $|1\rangle$ each with probability $\frac{1}{2}$, and $|+\rangle$ and $|-\rangle$ each with probability $\frac{1}{2}$ are indistinguishable.

We know from relativity that such a communication should not be possible, and Herbert's paper motivated two groups to come up with the no-cloning theorem. And the real reason Herbert's paper didn't work is because lasers didn't work the way he thought he did. But that was actually a rather subtle point, because none of the referees, obviously, knew how lasers worked, or they would have rejected this paper. And then, the no cloning theorem would have taken another five years to be discovered.

6 Quantum Weirdness

6.1 Einstein, Podolsky, and Rosen's issues with quantum entanglement

Einstein, Podolsky, and Rosen argued that quantum mechanics is incomplete. What they used in this paper was position and momentum, but we're working with discrete quantum mechanics, so we'll just use $|0\rangle$ and $|1\rangle$. Suppose, Alice and Bob share the state $\frac{1}{\sqrt{2}}(|01\rangle - |10\rangle)$. If Alice measures in the $|0\rangle$, $|1\rangle$ basis, Bob will have the opposite particle no matter what Alice measures and if she measures in the $|+\rangle$, $|-\rangle$ basis, again Bob will have the opposite particle.

Einstein argued that because Alice and Bob can be very far apart and can measure at the same time, Bob's particle cannot know whether Alice measured a $|0\rangle$ or $|1\rangle$ when Bob measures it. So if Alice can always predict with certainty what Bob will get in his measurement after her measurement, then there must be something in the quantum state, that tells what Bob will get when he measures it. But according to quantum mechanics, there is nothing in the state that tells it to be $|0\rangle$ or $|1\rangle$. So what that means is that there must be a more complete theory of reality than quantum mechanics.

So this is what Einstein argued. And Schrodinger wrote a paper that answered that. It said that qubits are entangled, and that's why this happens, which really is not that great of an explanation. Nobody paid much attention to Einstein's argument for the next 25 or 30 years or so. And Einstein was never satisfied with quantum mechanics completely, but quantum mechanics worked so people ignored this objection to it. And then in 1964, Bell published a paper that said, Einstein was both right in that there's something very strange going on here, but wrong in that there's no way to complete this theory to a real theory because the arguments of quantum mechanics conflict with classical notions of realism and locality.

6.2 Classical CHSH game

I'm going to present Bell's argument in this section. Originally, it's a statement about marginal probabilities of a joint probability distribution. But I find it makes much more intuitive sense if we think of it as a game that two players are cooperatively trying to play. And this is called the CHSH game after Clauser, Horne, Shimony and Holt. The two players are Alice and Bob. Alice and Bob have migrated from cryptography to quantum information theory, if you want to know where they came from. Alice and Bob are two of the prototypical participants in cryptosystems. They cooperate to win the game with as high probability as possible. We have a referee who doesn't make any decisions, but sends bit a to Alice and b to Bob. The bits are chosen completely at random. And then Alice and Bob, who are in separate rooms and cannot communicate, get these two bits and output two more bits x and y respectively.

Alice and Bob win if $x \oplus y = ab$. Now, let's pretend that Alice and Bob have to do this deterministically. And there is a general result in game theory that Alice and Bob cannot do any better with a probabilistic strategy than a deterministic strategy, which we will get to later. Let x_i and y_i denote the bit Alice and Bob respectively output if input is i. Then Alice and Bob try to achieve the following conditions:

$$\begin{aligned} x_0 \oplus y_0 &= 0 \\ x_0 \oplus y_1 &= 0 \\ x_1 \oplus y_0 &= 0 \\ x_1 \oplus y_1 &= 1 \end{aligned}$$

Now, we prove that they can't win with probability 1. Because, if we add all the four equations modulo 2, we get $0 = 1$, and this is of course, impossible. Furthermore, I claim that they can at best only win with probability 0.75. This is because, when the referee sends the bits to Alice and Bob, they have to satisfy one of the above equations, chosen at random. We know that they can't satisfy all equations, so at least one equation must disagree. Since the equations are chosen at random, Alice and Bob must lose at least with probability 0.25, and hence win with maximum 0.75 probability. In fact, if Alice and Bob output 0 regardless of what they receive, they will be able to satisfy the first three equations and can win with 0.75 probability.

Now, I want to argue why a probabilistic strategy can't do any better for all such games. Let U denote the set of all deterministic strategies. Then a random strategy can be seen as selecting $u \in U$ with probability p_u, whose winning probability is w_u Then the expectation value of probability of winning is $\sum_u p_u w_u \leq p_{u^*}$, where p_{u^*} is the maximum probability of winning for a deterministic strategy. Hence the optima of random strategies are located at a deterministic strategy or a mixture of deterministic strategies, when there are more than one optimum deterministic strategies. This result is true in general for all games.

In the next section, we see that if Alice and Bob share an EPR pair of qubits, they can win with probability larger than 0.75.

6.3 Quantum protocol for CHSH game

Alice and Bob share $|\psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle$. We use the fact that if Alice makes the same measurement of the form $\alpha X + \beta Z$, such that $\alpha^2 + \beta^2 = 1$ and α, β are real, Bob's qubit is left in the same state as what Alice measured. It is trivial for Z, as the eigenvectors are $|0\rangle$ and $|1\rangle$ themselves. For X, the eigenvectors are $|+\rangle$ and $|-\rangle$, and we can see that $\langle +|\psi\rangle = \frac{1}{\sqrt{2}}|+\rangle$ and similarly for $|-\rangle$, where we have taken partial inner products. The proof for linear combination is left as an exercise. Our quantum strategy would be as follows: Alice measures Z if she gets 0 and X if she gets 1, while Bob measures $\frac{1}{\sqrt{2}}(X + Z)$ if he gets 0 and $\frac{1}{\sqrt{2}}(X - Z)$ if he gets 1. Finally, they output 0 if the measured eigenvalue is +1 and output 1 if the measured eigenvalue is -1. The xz plane on the Bloch sphere for visualizing these observables would be immensely helpful.

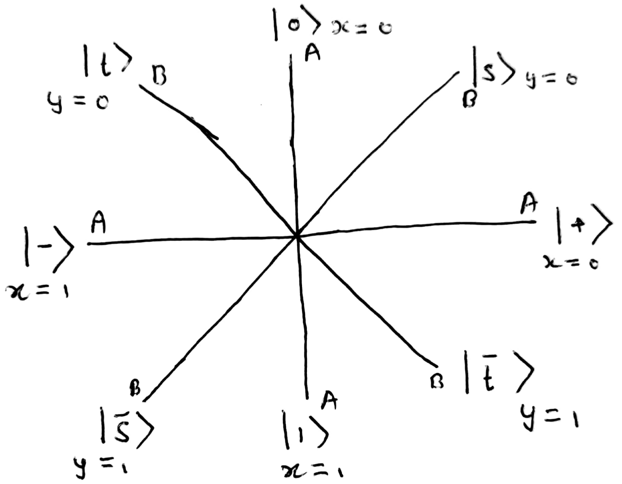


Figure 14: The xz plane on the Bloch sphere. The angles between adjacent lines is $\frac{\pi}{4}$. A and B denote Alice and Bob respectively. $|s\rangle$ and $|\bar{s}\rangle$ denote the eigenvectors of $\frac{1}{\sqrt{2}}(X + Z)$, and $|t\rangle$ and $|\bar{t}\rangle$ denote the eigenvectors of $\frac{1}{\sqrt{2}}(X - Z)$

Recall that corresponding to the state (θ, ϕ) on the Bloch sphere, the corresponding state is $|\psi\rangle = \cos\frac{\theta}{2}|0\rangle + e^{i\phi}\sin\frac{\theta}{2}|1\rangle$. Hence, an angle of θ between two states on the Bloch sphere, correspond to the fact that their inner product is $\cos\frac{\theta}{2}$. Now, suppose Alice and Bob both get 0 from the referee. So Alice measures Z and Bob measures $\frac{1}{\sqrt{2}}(X + Z)$. Suppose Alice measures

$|0\rangle$, then Bob measures $|0\rangle$, which is a winning state with probability $\cos^2(\frac{\pi}{8}) = 0.854$. Convince yourself that for all possible outcomes the winning states are separated by $\frac{\pi}{4}$ and hence the probability of winning the game with an EPR pair has increased to 0.854!

In fact, we can prove that this particular strategy is the optimal one. They cannot do better than 0.854 at this game. We are not going to prove this.

6.4 Mermin's magic square game - classical limit

Can you color the squares, with red and green as the only option, in a 3 by 3 grid so that every column has an odd number of red squares, and every row has an even number of red squares? The answer is no, because if every column has an odd number of red squares, then the total number of red squares is odd.While, if every row has an even number of red squares, then in total there is an even number of red squares.

Now we describe Mermin's game. The referee sends random numbers a and b from 1,2,3 to Alice and Bob respectively. Alice gives the colouring of row a such that it has even number of red squares and Bob gives the colouring of column b such that it has odd number of red squares. They lose if the row and the column don't agree in the square that they match, otherwise they win. Do Alice and Bob have a deterministic strategy that wins all the time? They can't, because if they had such a strategy then they could fill the 3 by 3 square such that every column has an odd number of red squares and every row has an even number of red squares. But we just proved that this can't happen. So they can't win with probability 1. Hence, Bob's and Alice's colouring must conflict in atleast one square, and in fact such a colouring can be constructed as shown below.

$$\begin{bmatrix} R & G & R \\ G & R & R \\ G & G & G/R \end{bmatrix}$$

Alice and Bob's colourings confilct in the (3,3) square and match for all other squares. So the maximum probability of winning this game is $8/9$, since the referee gives random numbers to Alice and Bob.

6.5 Quantum protocol for Mermin's game

Alice and Bob share two EPR pairs $\frac{1}{2}\left(|00\rangle + |11\rangle\right)\left(|00\rangle + |11\rangle\right)$, and then Alice and Bob measure observables depending on a and b. Note that two Hermitian matrices commute if and only if they are simultaneously diagonalizable, whose proof is left as an exercise. The strategy is to have a matrix of observables :

$$\begin{bmatrix} X \otimes I & I \otimes X & X \otimes X \\ I \otimes Z & Z \otimes I & Z \otimes Z \\ -(X \otimes Z) & -(Z \otimes X) & Y \otimes Y \end{bmatrix}$$

This matrix is a very interesting matrix. First, any two entries in the same row or column commute. Second, the product of any row is $I \otimes I$ and product of any column is $-(I \otimes I)$. Third, all entries have eigenvalues +1 and -1. All three properties are easy to verify using properties of Pauli matrices and definition of tensor products of operators.

Alice measures the observables in row a and Bob measures the observables in column b. It is possible to observe three observables simultaneously because there exists a basis in which all three are simultaneously diagonalisable because of their commuting nature. Finally, they put red if the measured eigenvalue is -1 and put green if it is +1. Since, all the observables multiply to $I \otimes I$ in a row, the values measured must multiply to 1, and hence there are an even number of -1 values or even number of red squares in a row. Similarly, since all the observables in a column multiply to $-(I \otimes I)$, the values measured multiply to -1, and hence there are odd number of red squares in the column. And the last thing is that Alice and Bob are sharing the entangled state so when they measure the same observable, they get the same result.

For example, if $a = 2$ and $b = 3$, the eigenstates for Alice are $|00\rangle, |01\rangle, |10\rangle, |11\rangle$ with eigenvalues +1,-1,-1,+1 respectively. And the eigenstates for Bob are $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$, $\frac{1}{\sqrt{2}}(|00\rangle - |11\rangle)$, $\frac{1}{\sqrt{2}}(|01\rangle + |10\rangle)$, $\frac{1}{\sqrt{2}}(|01\rangle - |10\rangle)$ with eigenvalues +1, +1, -1, -1 respectively. So, if Alice measured +1, then Bob's qubits too collapses to $|00\rangle, |11\rangle$ eigenspace and he too gets +1. The verification for other cases is left as an exercise.

What we see from these two games is that quantum mechanics(specially entanglement is very weird(and useful) because it gives Alice and Bob an advantage over the best classical strategy.

7 Teleportation and superdense coding

7.1 Overview

Asher Peres and Bill Wootters were considering the following scenario. You had a pair of identical qubits that were in three state 120 degrees apart on the Bloch sphere. And they were asking how well can you predict or test which of these three states they're in. They're not completely distinguishable. And if they're both in the same lab, the answer is very easy. But if they're in two different labs, the best that Peres and Wootters could do was come up with a very complicated protocol, which didn't get very close to the answer if they were in one lab. So they were more distinguishable in one lab than two. So we have Alice with one of our qubits and Bob with the other. So they're not allowed to bring these two qubits together. So what additional resources could they use to distinguish the possible states? They discovered an EPR pair helps. And then they started figuring out why an EPR here helps. And an EPR pair helps, because it lets you teleport Alice's qubit to Bob's lab. And then Bob can perform the measurement with both particles in the same lab. So that was a much more surprising discovery than just the fact that an EPR helps distinguish this probability distribution of correlated qubits in two different labs.

What is teleportation? Well, so some time from the past, Alice and Bob have acquired an entangled pair of qubits $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ And Alice has an unknown qubit $|\zeta\rangle$. So what Alice does is she performs a measurement on her half of the entangled pair and this unknown qubit. And this leaves Bob with a qubit in some state, $|\zeta^{'}\rangle$. And Alice gets two bits out of this measurement, because she's measuring two qubits, which is a four-dimensional quantum state, which is equivalent to two bits $b_1 b_2$ and sends it to Bob, who performs a unitary transform on $|\zeta^{'}\rangle$ based on the bits received, and miraculously he gets back the unknown state $|\zeta\rangle$. So we can send one qubit with two bits if the sender and receiver have an EPR pair. Remember that we could not send qubits over a classical channel because if we could, we could clone $|\zeta\rangle$ as a classical channel can be duplicated. But here, Alice and Bob also have this EPR pair and any eavesdropper can listen in on this conversation and get the two bits. But without the other half of the EPR pair, they can't recreate $|\zeta\rangle$.

In superdense coding Alice has 2 bits and she wants to send them to Bob. And what Alice does is she encodes the 2 qubits in her half of the EPR pair and sends her qubit to Bob, Now Bob does the joint measurement and he recovers the two bits. And this doesn't seem anywhere near as miraculous as teleportation. But if you go back to 1973, Alexander Holevo in Russia proved that 1 qubit can only transmit 1 bit of information. But, we are transmitting twice as much information as theoretically possible without the EPR pair with 1 qubit. That's why it's called superdense coding.

7.2 Quantum Teleportation - protocol and circuit

In teleportation, Alice measures in the Bell basis, the basis consisting of the four bell states $|\phi^+\rangle, |\phi^-\rangle, |\psi^+\rangle, |\psi^-\rangle$. Note that all four states are orthogonal and hence a measurement can distinguish them. So what Alice is going to do is she takes her unknown qubit and $|\phi^+\rangle$ and measure the first two qubits of this three qubit state in the Bell basis. Let's say $|\zeta\rangle = a|0\rangle + b|1\rangle$. Alice measures the first two qubits in the bell basis. She gets one of the four Bell states and Bob's qubit collapses into a certain state. To obtain that state, apply partial inner products of each Bell state with the three qubit state. The results are summarised below, which the reader should check.

Alice's measurement	Bob's state	Bob's Unitary
$\|\phi^+\rangle$	$a\|0\rangle + b\|1\rangle$	I
$\|\phi^-\rangle$	$a\|0\rangle - b\|1\rangle$	Z
$\|\psi^+\rangle$	$a\|1\rangle + b\|0\rangle$	X
$\|\psi^-\rangle$	$a\|1\rangle - b\|0\rangle$	ZX

Also, Bob needs to correct his qubit to match $|\zeta\rangle$. He applies some unitary transformation for it. In the $|\phi^+\rangle$ case, he does not need to apply any transform. For $|\phi^-\rangle$ he needs to apply Z to flip the sign of $|1\rangle$. For $|\psi^+\rangle$, he needs to apply X which behaves as NOT gate. For the last case, we could apply Y but that would add a global phase of $-i$ to $|\zeta\rangle$ so we multiply by $iY = ZX$.

Now, we build a circuit to achieve teleportation. In the circuit we denote our unknown qubit by $|\psi\rangle$ and $|\phi^+\rangle$ by $|\beta_{00}\rangle$. The first thing we want to do is to measure the first two qubits in the Bell basis, but our measuring devices are only in the standard basis.So, we map each Bell state to a state in the standard basis.A CNOT followed by an $H \otimes I$ exactly acheives this as shown in the table below:

Bell state	CNOT	$H \otimes I$
$\|\phi^+\rangle$	$\|+\rangle\|0\rangle$	$\|00\rangle$
$\|\phi^-\rangle$	$\|-\rangle\|0\rangle$	$\|10\rangle$
$\|\psi^+\rangle$	$\|+\rangle\|1\rangle$	$\|01\rangle$
$\|\psi^-\rangle$	$\|-\rangle\|1\rangle$	$\|11\rangle$

That explains the first part of the circuit below. Now we measure in the standard basis, and then apply the required unitary. We note that if Alice's first bit is 1, we want to apply X and if second bit is 1, we want to apply Z. Also, we want to apply X before Z. This explains the second part of the circuit. The powers on the Pauli matrices succinctly capture the need to apply the gates when the required bits are 1.

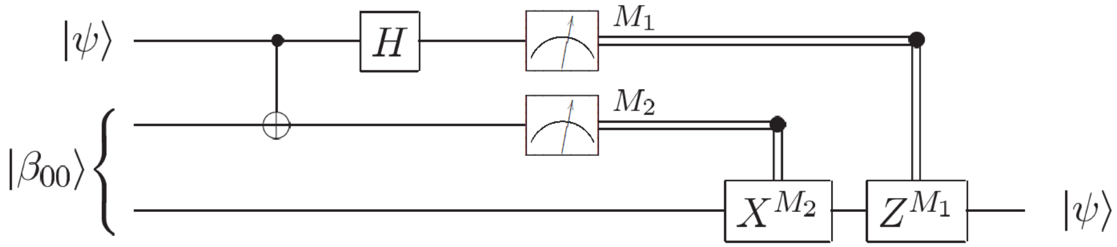


Figure 15: Quantum circuit for teleporting a qubit. The two top lines represent Alice's system, while the bottom line is Bob's system. The meters represent measurement, and the double lines coming out of them carry classical bits (recall that single lines denote qubits).

7.3 Quantum superdense coding - protocol

Alice and Bob share $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$. The following table summarises the unitary Alice applies to her qubit, corresponding to different bits she wants to send.

Bits	Alice's Unitary	Final state
00	I	$\frac{1}{\sqrt{2}}(\|00\rangle + \|11\rangle)$
01	X	$\frac{1}{\sqrt{2}}(\|10\rangle + \|01\rangle)$
10	Z	$\frac{1}{\sqrt{2}}(\|00\rangle - \|11\rangle)$
11	ZX	$\frac{1}{\sqrt{2}}(-\|10\rangle + \|01\rangle)$

Notice that all the obtained states are Bell states and hence can be distinguished by a measurement in the Bell basis. Alice sends her qubit to Bob and he makes the measurement in Bell basis to determine the bits sent.

8 Epilogue

I hope I got you hooked to quantum computing through this report. Unfortunately, I was not able to include many exciting things like quantum algorithms including Shor's factoring algorithm and Grover's search algorithm, quantum error correction, and quantum communication and key distribution. I will be updating this report on *https://github.com/the-nishant/Summer-of-Science*.