Basics of Quantum Computing

Arvind Saraswat

2022

Contents

List of Figures

Chapter 1

Introducing Quantum Computing

I make an attempt to write on the topic of **Quantum Computing**.

Quantum Computing is often in news and with complex-sounding terms and ideas, such as Quantum Superposition, Qubit, Quantum Entanglement, and Teleportation, Shor's algorithm that can break our current encryption methods, and Quantum key distribution that can fix the broken encryption issue, to name a few.

What do all these terms mean? And how to explain them without shedding many tears while reading long equations and complex mathematical symbols.

A little bit of mathematics is necessary though.

Many of the concepts in Quantum Computing beats intuition and perhaps that's the reason most of us find it difficult to understand. Let's take an example of superposition - a qubit can be in any of the infinite states, a superposition of 0 and 1, but when we measure it, it will always be in one of the two classical states, 0 or 1. The act of measurement changes the qubit state. This is difficult to understand intuitively, but a simple mathematical model describes it very well.

Another example is entanglement. When we measure one of the qubits, it affects the state of the other, no matter how far it is. The two qubits can be separated by a few meters or at the opposite ends of the galaxy! Again, difficult to describe intuitively, but easier via a mathematical model.

In the next few chapters, I will try to explain these concepts.

Let's go!

Chapter 2

On Measurements

In classical computing, the basic unit of computation is **bit**. Anything that can be in exactly one of two states can represent a bit.

A **qubit**, like a bit, can also be in one of two states, but, unlike a bit, it can also be in a combination of these two states. A qubit can be represented by the spin of an electron or the polarization of a photon.

To understand these concepts, let's start with an experiment performed by Otto Stern and Walther Gerlach on the spin of silver atoms.

2.1 Atoms, Electrons and Spin

In 1922, Niels Bohr's planetary model described the current understanding of atoms. In this model, an atom consisted of a positive nucleus orbited by negative electrons. These orbits were circular and were constrained to certain radii. The innermost orbit could contain at most two electrons. Once this was filled, electrons would start filling the next level, where at most eight electrons could be held. Silver atoms have 47 electrons. Two of these are in the innermost orbit, then eight in the next orbit, then eighteen more electrons in both the third and fourth levels. This leaves one lone electron in the outermost orbit.

It is known that electrons moving in circular orbits generate magnetic fields. The electrons in the inner orbits are paired, and from Pauli's exclusion principle, each of the pairs rotates in the opposite direction to its partner, resulting in their magnetic fields cancel. However, the single electron in the outer orbit generates a magnetic field that is not canceled by other electrons. This means that the

atom as a whole can be considered as a little magnet with both a south pole and a north pole.

Stern and Gerlach designed an experiment to test whether the north-south axes of these magnets can have any direction whatsoever or whether they were constrained to certain directions.

The experiment setup is shown in the below figure.

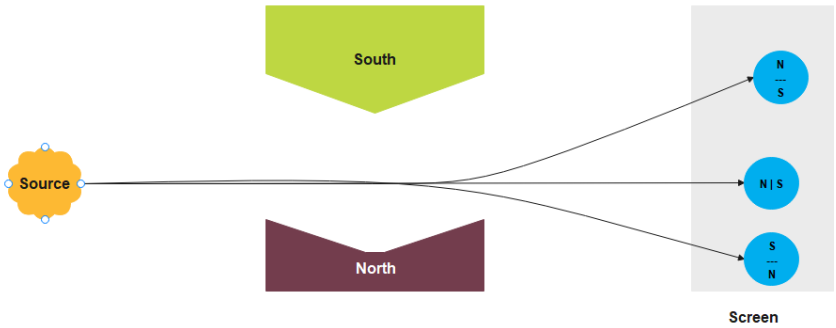

Figure 2.1: Stern Gerlach Experiment

A beam of silver atoms is shot from the source, passes through two magnets, and is collected at a screen at the other end. *The magnets are such that the **South** magnet is stronger than **North** magnet*.

Consider the following scenarios.

Considering the silver atom as a small magnet with North on top and South on the bottom, it will be attracted to both the magnets of the apparatus, but since the South magnet is stronger, it wins and the particle is deflected upward.

Alternatively, if the silver atom is considered as a magnet with South on top and North on the bottom, it will be repelled by both the magnets of the apparatus, but since the South magnet is stronger, it wins and the particle is deflected downward.

Going by the classical theory, the magnetic poles of the silver atom can be aligned in any direction. If they were aligned horizontally, there would be no deflection. In general, the size of the deflection would correspond to the amount the magnetic axis of the atom differs from the horizontal, with maximum deflections occurring when the magnetic poles of the atom are aligned vertically as explained previously.

If the classical theory is correct, and we send a large number of silver atoms through the magnets, we expect to see a continuous line on-screen running from top to bottom. However, this is not what Stern and Gerlach found. When they looked at the screen, they found just two dots: one at the extreme top and the other at the extreme bottom. All of the atoms behaved like little bar magnets that were aligned vertically. None of the atoms had any other orientation.

Not only do atoms act like little magnets, but so also do their components such as electrons. As with silver atoms, if we measure the spin in the vertical direction, we find that the electron is either deflected in the north direction or the south direction. Again, like silver atoms, we find that electrons are little magnets with their north and south poles perfectly aligned in the vertical direction. They do not seem to have any other orientation.

To summarize, consider the following depictions.

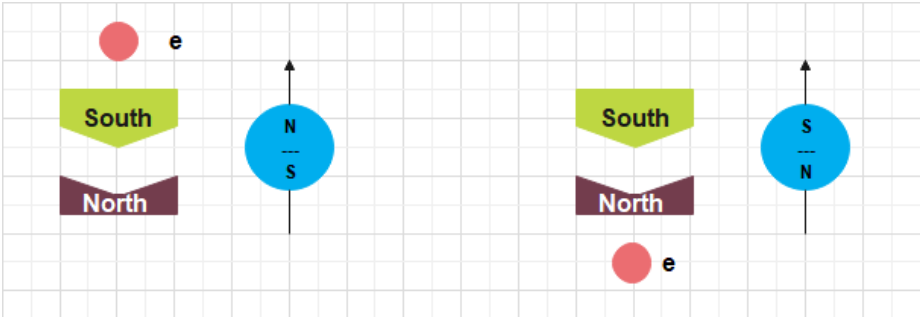

Figure 2.2: Detected Electrons And Spins

In the left figure, we will say that electron has **spin N** in the vertical direction. In the right figure, we will say that the electron has **spin S** in the vertical direction.

All the while do remember that in this experiment *the magnets are such that the **South** magnet is stronger than **North** magnet.*

There is nothing special about the vertical direction. As an example, we can rotate the magnets through 90˚ and the electrons will still be deflected in the direction given by either the north magnet or the south magnet. In this case, the electrons now behave like magnets with their north and south poles aligned in the horizontal direction, as shown in the below picture.

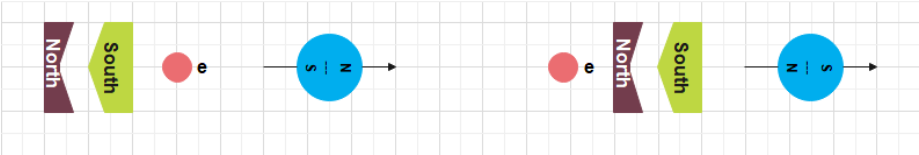

Figure 2.3: Detected Electrons And Spins Detection

We may want to rotate the magnets through various angles. We will measure angles in the clockwise direction with 0° denoting the upward vertical direction and θ measuring the angle from the upward vertical. The below picture shows an electron with spin N in the direction of a general angle θ°.

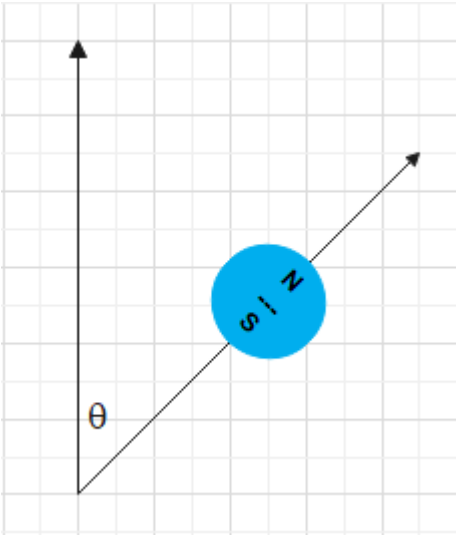

Figure 2.4: At an angle

2.2 A Strange Clock

Consider a clock with hours marked in normal positions with only an hour-hand. We are not supposed to look at it and observe the position. The only way allowed to interact with this clock is by asking a question if the hand is pointing at a particular number.

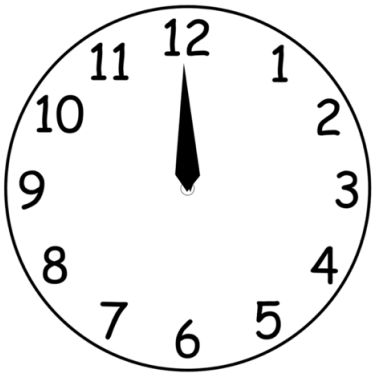

Figure 2.5: A Strange Clock

If it was a normal clock, we will rarely get a "yes" answer to our question.

But this is a strange clock - it either answers "yes" or tells that the hand is pointing in the direction exactly opposite to what you have asked about. For example, if we ask whether the hand is pointing at twelve, it either answers "yes" or tells that it is pointing at six. Similarly, if we ask whether the hand is pointing at three, it either answers "yes" or tells that it is pointing at nine.

This is similar to electron spin measurement as discussed in the previous section.

Since we want to do computations with qubits, we will be doing measurements. Let's see what happens when we measure multiple times.

Let's call this a Quantum Clock.

2.2.1 Measuring in same direction

Let's begin by measuring the electron's spin in the vertical direction. We will use the same setup as described in the previous section.

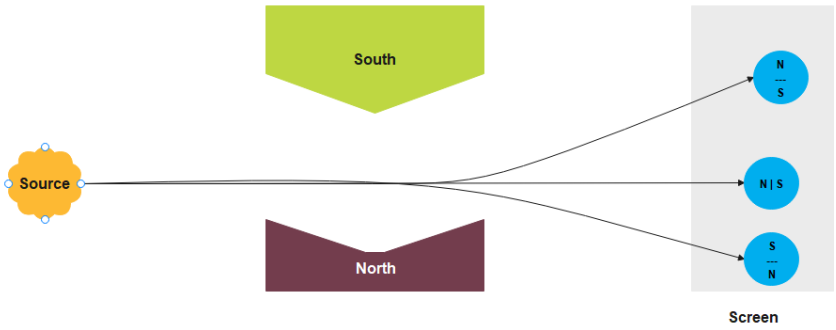

Figure 2.6: Stern Gerlach Experiment

We then repeat the same experiment by placing two more, exactly same, setup behind the first one. One is placed in exactly the right place to catch electrons that are deflected upward by the first apparatus. The other is placed to catch the electrons deflected downward. The complete test setup looks as below now.

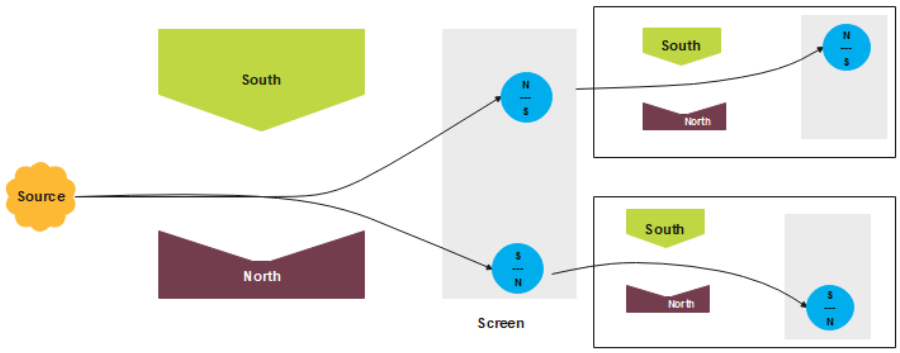

Figure 2.7: Multiple Spin Measurement

The electrons that are deflected upward by the first apparatus are deflected up by the second, and the ones deflected down by the first apparatus are deflected down by the second. This means that electrons measured to have spin N in direction 0° initially also have spin N in direction 0° when we repeat the experiment. Similarly, if an electron is initially measured to have spin S in direction 0° and we repeat exactly the same experiment, it will still have spin S in direction 0°.

Going back to the Quantum Clock scenario, if we repeatedly ask whether the hand is pointing at two, we will repeatedly get the same answer: either it is always pointing toward two or it is always pointing toward eight.

Lastly, we note that there is nothing special about the vertical direction. We can start by measuring in direction θ°, and if then we repeatedly measure in the same direction, we will obtain exactly the same result each time. We will end up with a string of letters consisting entirely of **N**s or one entirely of **S**s.

What will happen if we first measure vertically and then horizontally? Let's see.

2.2.2 Measuring in different direction

We will run the experiment again by measuring the electron's spin first in the vertical direction, and then in the horizontal direction. We place two more detectors behind the first setup to catch electrons coming from the first detector. These two new detectors are rotated by 90° and hence measure spin in the horizontal direction.

First, let's observe the stream of electrons that were deflected upward by the first detector (these have spin N in the direction of 0°).When these electrons pass through the second detector, it is observed that half of them have spin N and half have spin S in direction of 90°. The observed sequence of spins in direction of 90° is completely random. This means, there is no way to predict whether the electron that had a spin N in 0° after the first detector will have spin N or spin S when it's measured again in direction 90° by the second detector.

Similar results are observed for electrons that were observed to have spin S in 0° after the first detector - half of them have spin N and half have spin S in direction of 90°. This sequence of spin N and spin S after the second detector is also random.

If we consider the Quantum Clock introduced earlier, it's same as asking whether the hand is pointing at two and then asking if it's pointing in direction of four. If we setup a large number of such clocks, the answers to the second question will appear random. Half of these clocks will say it's pointing in direction of four and a half will say it's pointing in direction of ten. The answer to the first question will have no bearing on the answer to the second question.

Let's see what happens if we measure spin one more time. First, we measure vertically, then horizontally, and then vertically once more.

Consider the stream of electrons coming from the first detector that has spin N in the direction 0°. We have already seen that half of these will have spin N and half will have spin S in the direction 90°. Let's restrict the observation to the stream that corresponds to spin N for the first two measurements and then, for the third observation, measure spin again in the vertical direction. It

is observed that half of the electrons have spin N in direction 0° and a half have spin S. Also, the sequence of **N**s and **S**s is random. This means the fact that the electrons had spin N in the vertical direction has no bearing on whether or not they will have still have spin N when measured again in the vertical direction.

We can summarize the observations so far.

First, if we repeat the same question, we get the same answer. This means, sometimes there are definite answers and we don't get random answers to every question.

Second, randomness does occur sometimes. If we ask a sequence of questions, the final result can be random.

Third, the act of measurement seems to affect the result. We saw that if we ask the same question three times, we get exactly the same answer three times. However, when the second question is different from the first and third, the answer to the first and third questions may not the same. For example, if we ask three times in a row if the hand is pointing towards twelve, we will get the same answer every time. But if we ask first if it is pointing toward twelve, then whether it is pointing to three, and finally again whether it is pointing toward twelve, the answers to the first and third question need not be the same. The only difference between the two scenarios is the second question, so that question must be affecting the outcome of the following question.

2.3 Measurements

In the domain of classical mechanics, consider the path of a ball thrown up into the air under the influence of gravity. The path can be traced using calculus, but to do the calculation, we need to know certain quantities such as the mass of the ball and its initial velocity. How we measure these are not part of the theory. We just assume that they are known. The implicit assumption is that taking a measurement does not affect the system being modeled. For the example of a ball being thrown into the air, this makes sense. We can measure its initial velocity using a radar gun, for example. This involves bouncing photons off the ball and, though bouncing photons will affect the ball, it is negligible. This is the philosophy underlying classical mechanics: Measurements will affect the objects being studied, but experiments can be designed so the effect of measurement is negligible and so can consequently be ignored.

In quantum mechanics, we are often considering tiny particles like atoms or electrons. Here bouncing photons off them has an effect that is no longer negligible. In order to perform some measurements, we have to interact with the

system. These interactions are going to disturb our system, so we can no longer ignore them. For example, consider the case where we measure the spin of an electron first in the vertical direction and then in the horizontal one. We have seen that exactly half of the electrons that have spin N in direction 0° after passing through the first detector will have spin N in direction 90° when measured by the second detector. It might seem that the strength of the magnets might be having some effect on the outcome, perhaps they are so strong that they are causing the magnetic axes of the electrons to twist to align with the magnetic field of the measuring device, and that if we had weaker magnets the twisting would be lessened and we might get a different result. However, this is not true. It is the actual process of taking the measurement, however it is done, that affects the system. Each time a measurement is made, we will see that the system is changed in certain prescribed ways; these prescribed ways depend on the type of measurement being made but not on the strength of the measurement.

2.4 Randomness

The classical experiment for generating a random sequence of two symbols each associated with the probability of half is that of tossing a fair coin. If we toss a fair coin we might get a sequence HTTHHHTT...

Quantum mechanics also involves randomness. For example, if we first measure the spin of a stream of electrons in the vertical direction, then in the horizontal direction, and record the results from the second measuring device, we will obtain a string of **N**s and **S**s. This sequence of spins is completely random. For example, it might look something like NSSNNNSS ...

Although these two examples yield similar results, there is a big difference in how randomness is interpreted in the two theories.

Tossing a coin can be completely described by classical mechanics. To compute whether the coins land heads or tails up, you need first to carefully measure the initial conditions: the weight of the coin, the height above the ground, the force of the impact of the thumb on the coin, the exact location on the coin where the thumb hits, the position of the coin, and so on. Given all of these values exactly, the theory will tell us which way up the coin lands. There is no actual randomness involved. Tossing a coin seems random because each time we do it the initial conditions vary slightly. These slight variations can change the outcome from heads to tails and vice versa. There is no real randomness in classical mechanics, just what is often called **sensitive dependence to initial conditions**. The underlying idea concerning randomness in quantum mechanics is different. The randomness is true randomness.

In the Quantum domain, it was hypothesized that there are **hidden variables** that reflect in the randomness. There have been experiments conducted to differentiate between the hidden variable and the true randomness hypotheses. This experiment has been performed several times. The outcomes have always shown that the randomness is real and that there is no simple hidden variable theory that can eliminate it.

More on this later...

2.5 On Photons and Polarization

All electromagnetic waves (light waves, microwaves, X-rays, radio waves) are transverse. All sound waves are longitudinal. We will focus on transverse waves.

In transverse waves, the movement of the particles in the wave is perpendicular to the direction of motion of the wave. Light is the interaction of electric and magnetic fields traveling through space. The electric and magnetic vibrations of a light wave occur perpendicularly to each other. The electric field moves in one direction and magnetic in another, though always perpendicularly. So, we have one plane occupied by an electric field, the magnetic field perpendicular to it, and the direction of travel which is perpendicular to both. These electric and magnetic vibrations can occur in numerous planes. A light wave that is vibrating in more than one plane is known as unpolarized light. As you can see in the image below, the direction of propagation is constant, but the planes on which the amplitude occurs are changing.

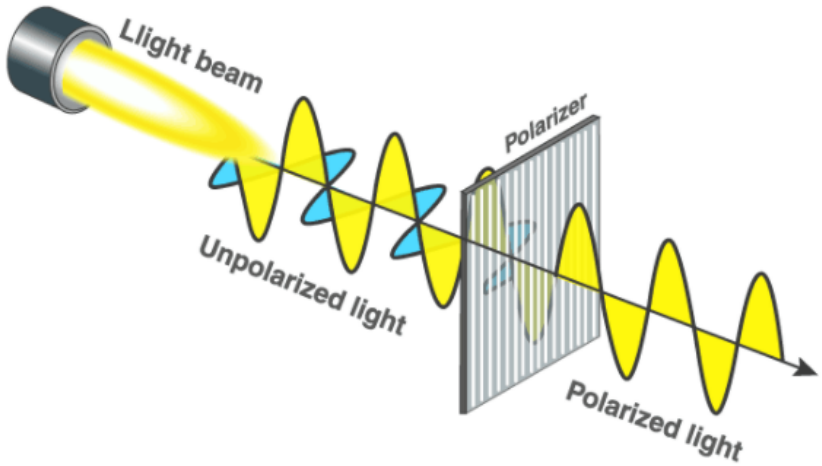

Figure 2.8: Transverse Waves

The other kind of wave is a polarized wave. Polarized waves are light waves in which the vibrations occur in a single plane. Plane polarized light consists of waves in which the direction of vibration is the same for all waves. In the image above, you can see that a Plane polarized light vibrates on only one plane. The process of transforming unpolarized light into polarized light is known as polarization.

When we measure polarization we find that photons are polarized in two perpendicular directions, both of which are perpendicular to the direction of travel of the photon. The polarized square lets through photons that are polarized in one of the two directions and absorbs the photons that are polarized in the other. The polarized squares correspond to the Stern-Gerlach apparatus. Sending light through a square can be considered making a measurement. As with spin, there are two possible outcomes: Either the direction of polarization is directly aligned with the orientation of the square, in which case the photon passes through, or the direction of polarization is perpendicular to the orientation of the square, in which case the photon is absorbed.

We start by assuming that our square has a vertical orientation so that it lets through photons with vertical polarization and absorbs the ones with horizontal polarization, and consider several experiments that correspond to the ones we described for electron spin.

First, suppose that we have two squares, both with the same orientation, so they both let through photons with vertical polarization. If we look at the squares individually they look gray, as they are both absorbing some photons — those with horizontal polarization. If we then slide one of the squares over the other, there is minimal change. The amount of light let through the two overlapping squares is about the same as the amount that comes through each square when they are not overlapping. This is depicted in the below figure.

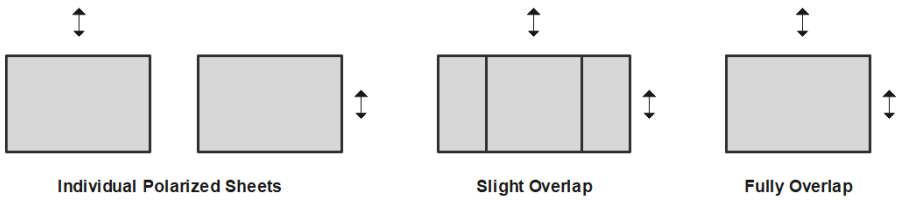

Figure 2.9: SamePolarization

Next, we will now rotate one of the squares through ninety degrees. We can safely assume that the proportion of horizontally polarized photons is equal to the proportion of vertically polarized ones, and both squares will look equally gray. We repeat the experiment of overlapping these squares. This time no light is let through the region of overlap, as depicted in the below figure.

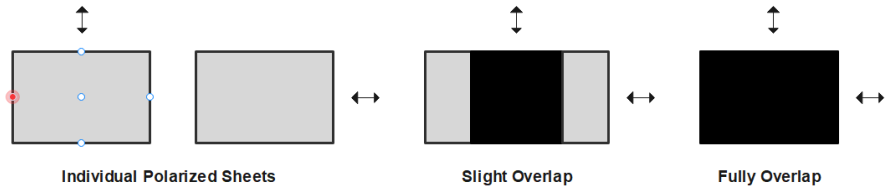

Figure 2.10: DifferentPolarization

The third experiment is to take the third sheet and rotate it through forty-five degrees. Under normal light conditions, nothing appears to happen as we rotate the square. It maintains the same shade of gray. We now slide this square between the other two squares, one of which has a vertical orientation, and the other has a horizontal orientation. The result is both surprising and unintuitive. Some light comes through the region of overlap of all three squares.

These polarized squares are sometimes called filters, but clearly, they are not acting in the conventional ways that filters work. We are seeing more light when there are three filters compared with two filters.

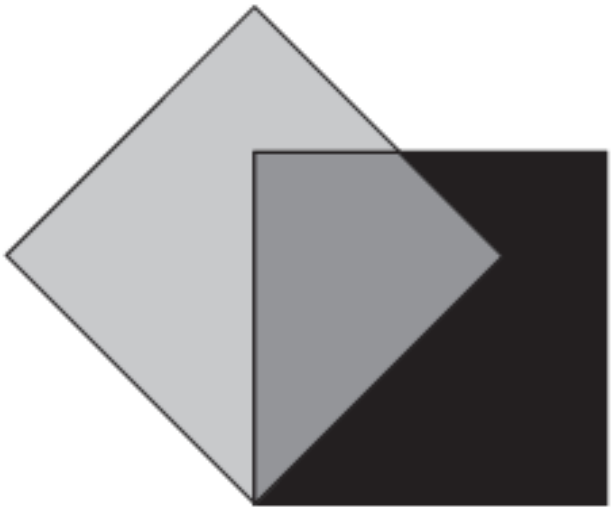

Figure 2.11: Three Polaroids

The result of the third experiment is non-intuitive. In this case, the filter that has been rotated through forty-five degrees is now measuring the polarization at

angles of 45° and 135°. We know that the photons coming through the first filter are polarized vertically. When measured by the second filter, half of the photons are found to be polarized in the 45° direction and half in the 135° directions. The ones with 45° polarization pass through the filter, and the others are absorbed. The third filter again measures the polarization in the vertical and horizontal directions. The photons entering have 45° polarization, and when measured in the vertical and horizontal directions, half will have vertical polarization and half will have horizontal polarization. The filter absorbs the vertically polarized photons and lets through those that are polarized horizontally.

There is another way to explain this behavior.

Consider the below diagram. What percentage of the unpolarized light is oriented to exactly 0° or very nearly 0°? Almost none of it. So, if a polarizer simply knocked out undesirable orientations, the strength of the remaining light would be almost entirely gone — it would be less than 1% as strong as the original light source. But this is now what we observe.

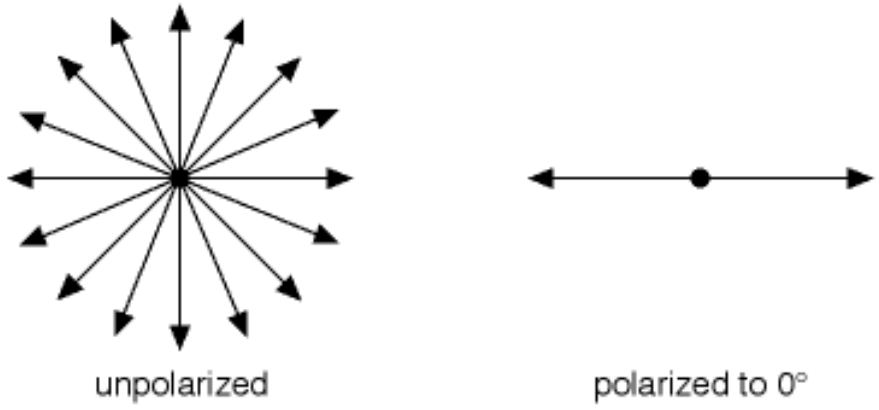

Figure 2.12: Polarization Example

Let's break this into multiple components and see what happens to each of the orientations represented in our simple diagram.

Figure 2.13: Zero Degree example

In this case, we see that light already at 0° is unchanged.

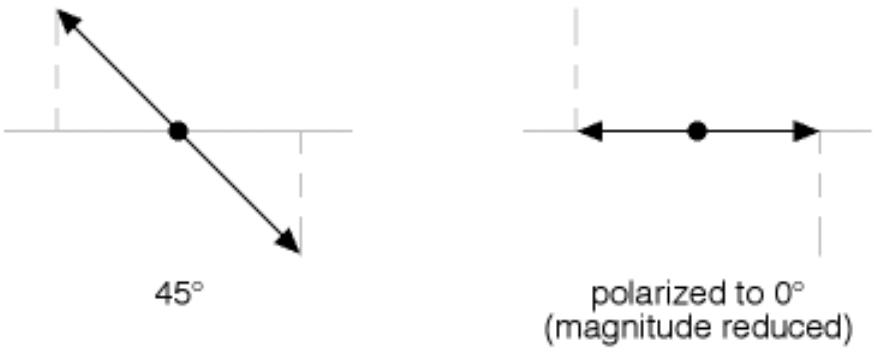

Figure 2.14: Forty Five Degree example

In this case, we see what happens to light oriented at 45° — it has its transverse (vertical) component destroyed, and becomes oriented to 0°, but with a weaker magnitude. Simple geometry tells us that it must have a magnitude about 71% of what it had before being polarized to 0° — i.e., $\frac{1}{\sqrt{2}} = .707$.

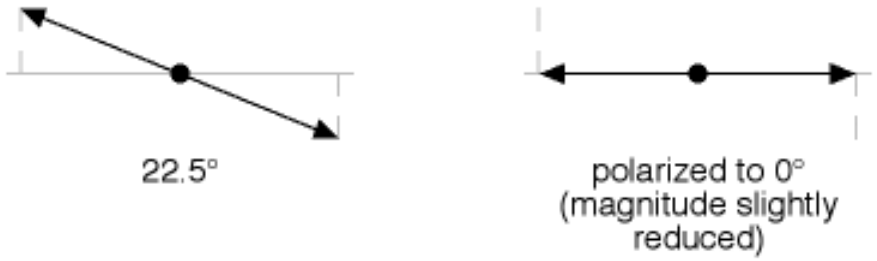

Figure 2.15: 22.5 Degree example

The above figure shows us that light that is close to 0° loses only a little of its magnitude when being crushed to 0° ...

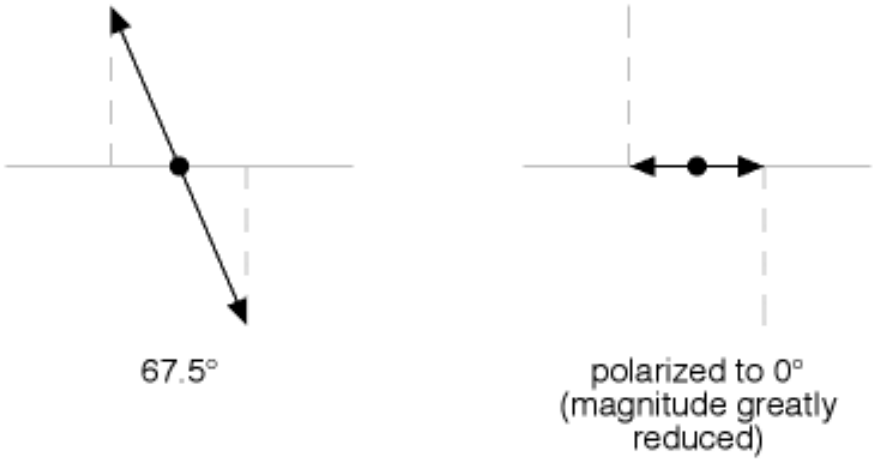

Figure 2.16: 67.5 Degree example

... while the above figure shows us that light that is close to 90° off of the polarizer loses most of its magnitude when being crushed to 0°.

Figure 2.17: Vertical example

And finally, the above figure illustrates light at 90° to the filter being crushed completely out of existence. This example illustrates what is happening in the

case where we had two linear polarized squares with different orientations in the path of our light source.

If we combine our understanding of all the above scenarios, we can see what happens between the first and second filter and then between the second and third filter.

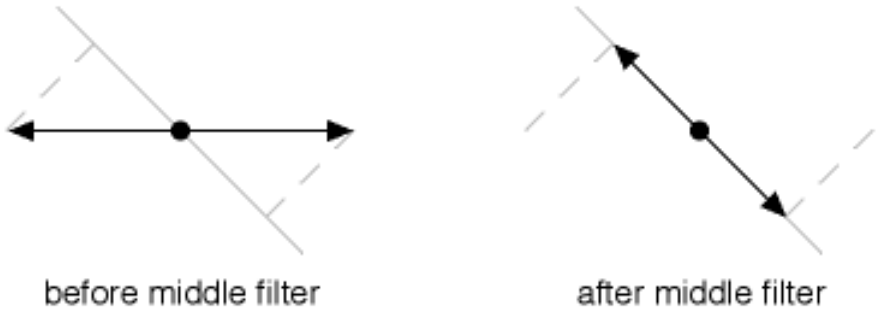

Figure 2.18: Before first and second filter

We can see that the middle filter takes 0° polarized light (from the first filter) and crushes it to a 45° orientation. This causes the light to drop to about 71% of its magnitude coming from the first filter.

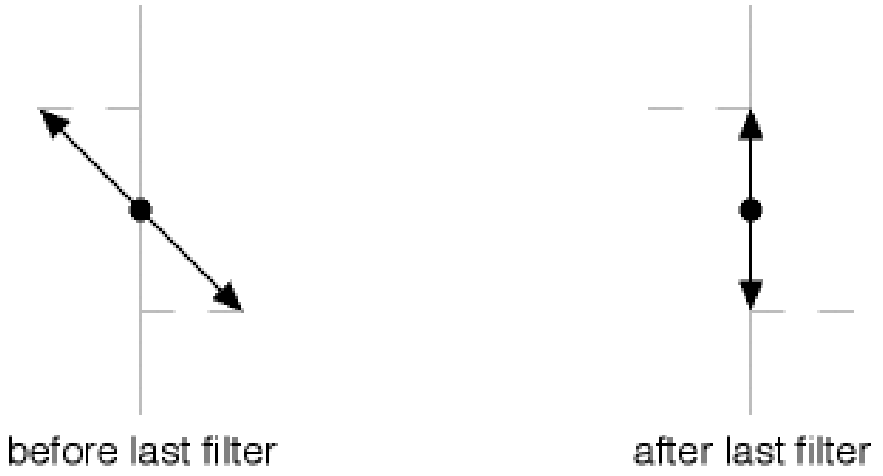

Figure 2.19: Between second and third filter

Lastly, the above figure shows the last filter taking the 45° polarized light (from the middle filter) and crushing it to a 90° orientation. This causes another 29% drop in magnitude, for an overall drop of exactly 50%.

Thus we can see why more light comes through three filters than comes through two!

2.6 Summary

Classical bits can be represented by everyday objects like switches in the on or off position, but qubits are generally represented by the spin of electrons or the polarization of photons.

To measure spin, you first have to **choose a direction** and **then measure it in that direction**. Spin is quantized: When measured, it gives just two possible answers—not a continuous range of answers. We can assign classical bits to these results. For example, if we obtain an N we can consider it to be the binary digit 0, and if we obtain an S we can consider it to be the binary digit 1. This is exactly how we get answers from the quantum computation. **The last stage of the computation is to take a measurement**. The result will be one of two things, which will be interpreted as either 0 or 1. Although the actual computation will involve qubits, the final answer will be in terms of classical bits.

Chapter 3

Mathematical Foundation

Linear Algebra is the language of Quantum Computing. Linear Algebra is a subject that can quickly become very complex and often involve analysis and computations in infinite dimensions. For our understanding of quantum computing, luckily, we need only finite dimensions and eves fewer tools.

The mathematics may seem complex but becomes one becomes comfortable with practice. The actual computation steps are no more than the addition and multiplication of numbers, and occasional square root and trigonometric functions.

I will also introduce Paul Dirac's notation. This will surely look strange at the beginning, but it's easier to follow once the basic rules are clear.

A very good resource for Linear Algebra can be found at Essence of linear algebra.

Let's start with Vectors.

3.1 Vectors

A vector can be considered as a list of numbers. The **dimension** of the vector is the number of numbers in the list. If the lists are written vertically, we call them column vectors or *kets*. If the lists are written horizontally, we call them row vectors or *bras*. Here is an example of a four-dimensional bra and a three-dimensional ket.

$$[2, 6, \pi, -10.0], \begin{bmatrix} 2 \\ 5 \\ 10 \end{bmatrix}$$

The names bra and ket come from Paul Dirac. He also introduced a notation for naming these two types of vectors: a ket with name v is denoted by $|v\rangle$; a bra with name w is denoted by $\langle w|$. So, we may write

$$|v\rangle = \begin{bmatrix} 2 \\ 5 \\ 10 \end{bmatrix} \text{ and } \langle w| = [2, 6, \pi, -10.0]$$

3.1.1 Length of Vectors

The length of a vector is defined as square root of the sum of its entries. The length of a ket $|a\rangle$ is denoted by $||a\rangle|$. As an example, for $|a\rangle = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$, $||a\rangle| = \sqrt{2^2 + 1^2} = \sqrt{5}$.

As a more general example, if $|a\rangle = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix}$, then $||a\rangle| = \sqrt{a_1^2 + a_2^2 + \cdots + a_n^2}$

The vectors with length 1 are called **unit vectors**. Qubits are often represented in terms of unit vectors.

3.2 Multiplication by a Scalar

A vector can be multiplied by a number. The effect of this multiplication is to often **scale** the vector by appropriate factor. More generally, multiplying the ket $|a\rangle = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix}$ by a scalar k yields $k\,|a\rangle = \begin{bmatrix} ka_1 \\ ka_2 \\ \vdots \\ ka_n \end{bmatrix}$

If the scalar k is positive, it gives a vector of a scaled length in the same direction as the original vector. We will often need to get a unit vector in the direction of a non-zero vector. Given any non-zero vector $|a\rangle$, its length is $||a\rangle|$. If we

multiply $|a\rangle$ by the reciprocal of its length, we obtain a unit vector. An example makes it clearer.

If $|a\rangle = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$, then $||a\rangle| = \sqrt{5}$.

Lets define a new vector such that $|u\rangle = \frac{1}{\sqrt{5}} \begin{bmatrix} 2 \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{2}{\sqrt{5}} \\ \frac{1}{\sqrt{5}} \end{bmatrix}$

then $||u\rangle| = \sqrt{(\frac{2}{\sqrt{5}})^2 + (\frac{1}{\sqrt{5}})^2} = \sqrt{\frac{4}{5} + \frac{1}{5}} = \sqrt{\frac{5}{5}} = \sqrt{1} = 1$.

Formally, we say $|u\rangle$ is a unit vector in direction of $|a\rangle$.

3.3 Vector Addition

Given two vectors that have the same type - they are both bras or both kets — and they have the same dimension, we can add them to get a new vector of the same type and dimension. The first entry of this vector just comes from adding the first entries of the two vectors, the second entry from adding the two-second entries, and so on.

Expressed mathematically, if $|a\rangle = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix}$ and $|b\rangle = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$, then $|a+b\rangle = \begin{bmatrix} a_1 + b_1 \\ a_2 + b_2 \\ \vdots \\ a_n + b_n \end{bmatrix}$.

Specifically, if $|a\rangle = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$ and $|b\rangle = \begin{bmatrix} 5 \\ 1 \end{bmatrix}$, then $|a+b\rangle = |b+a\rangle = \begin{bmatrix} 6 \\ 3 \end{bmatrix}$.

Vector addition can also be explained via Parrellolgram law of vector addition. The following figure makes it clear.

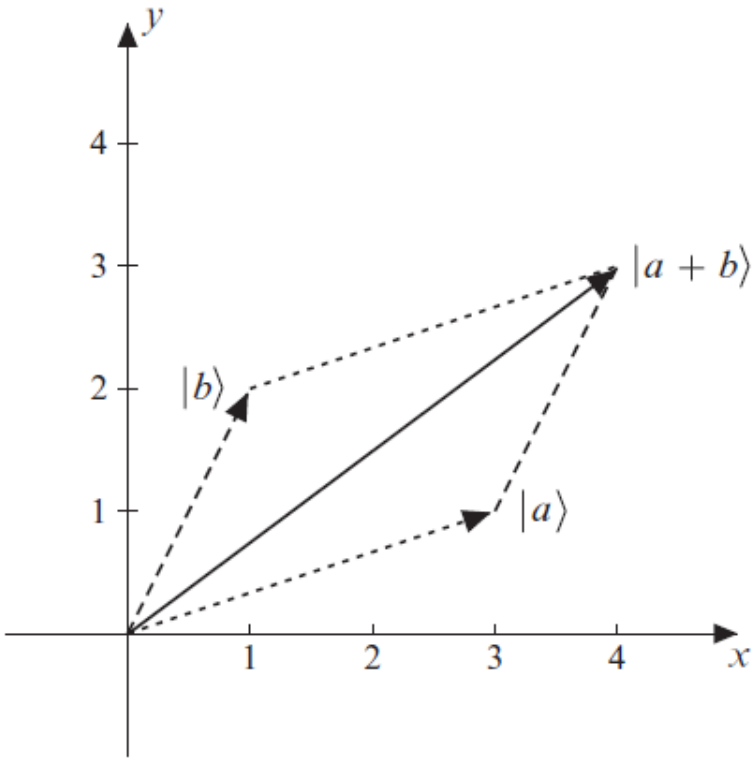

Figure 3.1: Parallelogram Law of Vector Addition

3.4 Orthogonal Vectors

The Pythagorean Theorem states that if a, b, c are lengths of three sides of a triangle, then $a^2 + b^2 = c^2$ iff the triangle is a right triangle.

The figure above then tells us that the vectors $|a\rangle$ and $|b\rangle$ are perpendicular if and only if $||a\rangle|^2 + ||b\rangle|^2 = ||a + b\rangle|^2$.

Linear algebra uses the word *orthogonal* to represent perpendicular vectors. Thus, we can state that two vectoes $|a\rangle$ and $|b\rangle$ are **orthogonal** if and only if $||a\rangle|^2 + ||b\rangle|^2 = ||a + b\rangle|^2$.

3.5 Bra - Ket Multiplication

Only bra and ket of the same dimension can be multiplied together. This means they should have the same number of elements.

Let $\langle a|$ and $|b\rangle$ be n-dimensional, that is: $\langle a| = \begin{bmatrix} a_1 & a_2 & \cdots & a_n \end{bmatrix}$ and $|b\rangle = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$, then their product is defined as:

$$\langle a|b\rangle = \begin{bmatrix} a_1 & a_2 & \cdots & a_n \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} = a_1b_1 + a_2b_2 + \cdots + a_nb_n$$

This is the same as *inner product* or *dot product* in Linear Algebra. Here we will use the bra-ket notation.

If we have a bra and a ket such that, $\langle a| = \begin{bmatrix} a_1 & a_2 & \cdots & a_n \end{bmatrix}$ and $|a\rangle = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix}$, then we have: $\langle a|a\rangle = a_1^2 + a_2^2 + \cdots + a_n^n$, or we can say $||a\rangle| = \sqrt{\langle a|a\rangle}$.

3.6 BraKets and test for Orthogonality

Recall from earlier, two vectoes $|a\rangle$ and $|b\rangle$ are **orthogonal** if and only if $||a\rangle|^2 + ||b\rangle|^2 = ||a + b\rangle|^2$.

So, to check for orthogonality, we need to calculate the number of square operations. Is there a simpler way?

Consider two kets in two-domensional setting. Let $|a\rangle = \begin{bmatrix} a_1 \\ a_2 \end{bmatrix}$ and $|b\rangle = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$, then $|a + b\rangle = \begin{bmatrix} a_1 + b_1 \\ a_2 + b_2 \end{bmatrix}$

$$
\begin{aligned}
||a+b\rangle|^2 &= \begin{bmatrix} a_1+b_1 & a_2+b_2 \end{bmatrix} \begin{bmatrix} a_1+b_1 \\ a_2+b_2 \end{bmatrix} \\
&= (a_1+b_1)^2 + (a_2+b_2)^2 \\
&= (a_1^2+b_1^2+2a_1b_1) + (a_2^2+b_2^2+2a_2b_2) \\
&= (a_1^2+a_2^2) + (b_1^2+b_2^2) + 2(a_1b_1+a_2b_2) \\
&= ||a\rangle|^2 + ||b\rangle|^2 + 2\langle a|b\rangle
\end{aligned}
$$

Thus, we can say that these two vectors are orthogonal if $2\langle a|b\rangle = 0$, or $\langle a|b\rangle = 0$.

More formally, two vectors $|a\rangle$ and $|b\rangle$ are orthogonal if $\langle a|b\rangle = 0$.

3.7 Orthonormal Bases

In linear algebra, two vectors are orthonormal if they are **orthogonal** (or perpendicular along a line) **unit** vectors. A set of vectors form an orthonormal set if all vectors in the set are mutually orthogonal and all of the unit length.

Wikipedia defines Basis Vectors as a set **B** of vectors in a vector space **V** is called a basis if every element of **V** may be written in a unique way as a finite linear combination of elements of **B**. The coefficients of this linear combination are referred to as components or coordinates of the vector with respect to **B**. The elements of a basis are called **basis vectors**.

Equivalently, a set **B** is a basis if its elements are linearly independent and every element of **V** is a linear combination of elements of **B**.

If you can write every vector in a given space as a linear combination of some vectors and these vectors are independent of each other then we call them basis vectors for that given space.

The following are three important properties of basis vectors:

1. Basis vectors must be linearly independent of each other
2. Basis vectors must span the whole space
3. Basis vectors are not unique

Let's take an example to understand better. The set of all two dimensional vectors is donated by $\mathbb{R}^2$. An orthonormal basis for $\mathbb{R}^2$ consists of a set containing two unit vectors $|b_1\rangle$ and $|b_2\rangle$ that are orthogonal. Thus, given a pair of kets, to

check if they form an orthonormal basis, we must check if they are unit vectors, and check whether they are orthogonal. We can check both of these conditions using bra-kets: $\langle b_1|b_1\rangle = 1$, $\langle b_2|b_2\rangle = 1$ and $\langle b_1|b_2\rangle = 0$.

In $\mathbb{R}^2$ space, the two **standard basis** vectors are $|b_1\rangle = \begin{bmatrix}1\\0\end{bmatrix}$ and $|b_2\rangle = \begin{bmatrix}0\\1\end{bmatrix}$. You can verify that these two vectors satisfy the properties of basis vectors.

The basis $\left\{\begin{bmatrix}1\\0\end{bmatrix}, \begin{bmatrix}0\\1\end{bmatrix}\right\}$ is not the only basis in $\mathbb{R}^2$ and there are infinitely more to choose from. Two particularly important for Quantum Mechanics and study of elctron spin are: $\left\{\begin{bmatrix}\frac{\mathbf{1}}{\sqrt{\mathbf{2}}}\\ \frac{\mathbf{-1}}{\sqrt{\mathbf{2}}}\end{bmatrix}, \begin{bmatrix}\frac{\mathbf{1}}{\sqrt{\mathbf{2}}}\\ \frac{\mathbf{1}}{\sqrt{\mathbf{2}}}\end{bmatrix}\right\}$ and $\left\{\begin{bmatrix}\frac{\mathbf{1}}{\mathbf{2}}\\ \frac{\sqrt{\mathbf{3}}}{\mathbf{2}}\end{bmatrix}, \begin{bmatrix}\frac{-\sqrt{\mathbf{3}}}{\mathbf{2}}\\ \frac{\mathbf{1}}{\mathbf{2}}\end{bmatrix}\right\}$.

Earlier, we looked at spin measured in the vertical direction and in the horizontal direction. The mathematical model for measuring spin in the vertical direction will be given using the **standard basis**. Rotating the measuring apparatus will be described mathematically by choosing a new orthonormal basis.

The three two-dimensional bases that were mentioned in the previous section have important interpretations concerning spin, so instead of naming these basis vectors with letters, we will use arrows, with the direction of the arrow related to the direction of spin.

$$|\uparrow\rangle = \begin{bmatrix}\mathbf{1}\\ \mathbf{0}\end{bmatrix} \quad |\downarrow\rangle = \begin{bmatrix}\mathbf{0}\\ \mathbf{1}\end{bmatrix}$$

$$|\rightarrow\rangle = \begin{bmatrix}\frac{\mathbf{1}}{\sqrt{\mathbf{2}}}\\ \frac{\mathbf{-1}}{\sqrt{\mathbf{2}}}\end{bmatrix} \quad |\leftarrow\rangle = \begin{bmatrix}\frac{\mathbf{1}}{\sqrt{\mathbf{2}}}\\ \frac{\mathbf{1}}{\sqrt{\mathbf{2}}}\end{bmatrix}$$

$$|\nearrow\rangle = \begin{bmatrix}\frac{\mathbf{1}}{\mathbf{2}}\\ \frac{\mathbf{-3}}{\sqrt{\mathbf{2}}}\end{bmatrix} \quad |\swarrow\rangle = \begin{bmatrix}\frac{\mathbf{3}}{\sqrt{\mathbf{2}}}\\ \frac{\mathbf{1}}{\mathbf{2}}\end{bmatrix}$$

These three basis can be written as $\{|\uparrow\rangle, |\downarrow\rangle\}$, $\{|\rightarrow\rangle, |\leftarrow\rangle\}$ and $\{|\nearrow\rangle, |\swarrow\rangle\}$.

Lastly, since they are orthonormal, we have the following:

$\langle\uparrow|\uparrow\rangle = 1$, $\langle\downarrow|\downarrow\rangle = 1$, $\langle\uparrow|\downarrow\rangle = 0$, $\langle\downarrow|\uparrow\rangle = 0$

$\langle \rightarrow | \rightarrow \rangle = 1$, $\langle \leftarrow | \leftarrow \rangle = 1$, $\langle \rightarrow | \leftarrow \rangle = 0$, $\langle \leftarrow | \rightarrow \rangle = 0$

$\langle \nearrow | \nearrow \rangle = 1$, $\langle \swarrow | \swarrow \rangle = 1$, $\langle \nearrow | \swarrow \rangle = 0$, $\langle \swarrow | \nearrow \rangle = 0$

Pretty neat!

3.8 Linear combination of Basis vectors

Any vector $|v\rangle$ in $\mathbb{R}^2$ can be written as a combination of $|\uparrow\rangle$ and $|\downarrow\rangle$. As an example, $\forall c, d$, following has a solution:

$$\begin{bmatrix} c \\ d \end{bmatrix} = x_1 \begin{bmatrix} 1 \\ 0 \end{bmatrix} + x_2 \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

The solution is trivial in this case: $x_1 = c, x_2 = d$.

Let's try another example. Can a vector $|v\rangle$ in $\mathbb{R}^2$ can be written as a combination of $|\rightarrow\rangle$ and $|\leftarrow\rangle$? Mathematically, how do we solve for $\forall c, d$?

$$\begin{bmatrix} c \\ d \end{bmatrix} = x_1 |\rightarrow\rangle + x_2 |\leftarrow\rangle$$

One way to solve this would be to rewrite the above as:

$$\begin{bmatrix} c \\ d \end{bmatrix} = x_1 \begin{bmatrix} \frac{\mathbf{1}}{\sqrt{\mathbf{2}}} \\ \frac{\mathbf{-1}}{\sqrt{\mathbf{2}}} \end{bmatrix} + x_2 \begin{bmatrix} \frac{\mathbf{1}}{\sqrt{\mathbf{2}}} \\ \frac{\mathbf{1}}{\sqrt{\mathbf{2}}} \end{bmatrix}$$

The above can be expanded to give:

$$c = \frac{\mathbf{1}}{\sqrt{\mathbf{2}}} x_1 + \frac{\mathbf{1}}{\sqrt{\mathbf{2}}} x_2$$

$$d = \frac{\mathbf{-1}}{\sqrt{\mathbf{2}}} x_1 + \frac{\mathbf{1}}{\sqrt{\mathbf{2}}} x_2$$

These two equations can be solved algebraically to solve for x_1 and x_2. If there are n equations to solves, it quickly becomes cumbersome to solve.

Can we do better? It seems yes.

Let's start by rewriting the problem once again.

$\begin{bmatrix} c \end{bmatrix} = x_1 \left|\rightarrow\right\rangle + x_2 \left|\leftarrow\right\rangle$

Multiplying above by bra $\langle\rightarrow|$, we get:

$$\langle\rightarrow| \begin{bmatrix} c \\ d \end{bmatrix} = \langle\rightarrow| \left(x_1 \left|\rightarrow\right\rangle + x_2 \left|\leftarrow\right\rangle\right)$$

$$\implies \langle\rightarrow| \begin{bmatrix} c \\ d \end{bmatrix} = x_1 \left\langle\rightarrow \mid \rightarrow\right\rangle + x_2 \left\langle\rightarrow \mid \leftarrow\right\rangle$$

From property of otheronormal basis that we have seen earlier, we know $\langle\rightarrow \mid \rightarrow\rangle = 1$ and $\langle\rightarrow \mid \leftarrow\rangle = 0$.

$$\implies \langle\rightarrow| \begin{bmatrix} c \\ d \end{bmatrix} = x_1 \times 1 + x_2 \times 0$$

$$\implies x_1 = \langle\rightarrow| \begin{bmatrix} c \\ d \end{bmatrix}$$

$$\implies x_1 = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} c \\ d \end{bmatrix}$$

$$\implies x_1 = \frac{1}{\sqrt{2}} c - \frac{1}{\sqrt{2}} d$$

$$\implies x_1 = \frac{c - d}{\sqrt{2}}$$

We can follow the same method to solve for x_2. We start by mulltiplying the original equation $\begin{bmatrix} c \\ d \end{bmatrix} = x_1 \left|\rightarrow\right\rangle + x_2 \left|\leftarrow\right\rangle$ on left by $\langle\leftarrow|$

$$\implies \langle\leftarrow| \begin{bmatrix} c \\ d \end{bmatrix} = \langle\leftarrow| \left(x_1 \left|\rightarrow\right\rangle + x_2 \left|\leftarrow\right\rangle\right)$$

$$\implies \langle\leftarrow| \begin{bmatrix} c \\ d \end{bmatrix} = x_1 \left\langle\leftarrow \mid \rightarrow\right\rangle + x_2 \left\langle\leftarrow \mid \leftarrow\right\rangle$$

From property of otheronormal basis that we have seen earlier, we know $\langle\leftarrow \mid \rightarrow\rangle = 0$ and $\langle\leftarrow \mid \leftarrow\rangle = 1$.

$$\implies \langle\leftarrow| \begin{bmatrix} c \\ d \end{bmatrix} = x_1 \times 0 + x_2 \times 1$$

$$\implies x_2 = \langle\leftarrow| \begin{bmatrix} c \\ d \end{bmatrix}$$

$$\implies x_2 = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} c \\ d \end{bmatrix}$$

$$\implies x_2 = \frac{1}{\sqrt{2}}c + \frac{1}{\sqrt{2}}d$$

$$\implies x_2 = \frac{c+d}{\sqrt{2}}$$

Thus, we can summarize by writing

$$\begin{bmatrix} c \\ d \end{bmatrix} = \frac{c-d}{\sqrt{2}} \left|\rightarrow\right\rangle + \frac{c+d}{\sqrt{2}} \left|\leftarrow\right\rangle$$

We have shown that any vector $|v\rangle$ in $\mathbb{R}^2$ can be written as a *linear combination* of basis vector. We have seen two examples of expressing an arbitrary in terms of $\{|\uparrow\rangle, |\downarrow\rangle\}$ and $\{|\rightarrow\rangle, |\leftarrow\rangle\}$.

Moving forward, consider n ket $|v\rangle$ and an orthonormal basis $\{b_1, b_2, \cdots, b_n\}$. Can we write $|v\rangle$ as a linear combination of these basis vectors?

$$|v\rangle = x_1 |b_1\rangle + x_2 |b_2\rangle + \cdots + x_i |b_i\rangle + \cdots + x_n |b_n\rangle$$

One way is to expand the above equation into individual equations and then solve the $n - domensional$ system of equations. This very quickly becomes complex to handle efficiently.

Alternately, we can follow the method outlined above. We can left-multiply the equation above by $\langle b_i|$. Since all these basis vectors are orthonormal, we know that $\langle b_i|b_k\rangle = 0$ when $i \neq k$ and $\langle b_i|b_k\rangle = 1$ when $i = k$.

$$\langle b_i| |v\rangle = \langle b_i| \{x_1 |b_1\rangle + x_2 |b_2\rangle + \cdots + x_i |b_i\rangle + \cdots + x_n |b_n\rangle\}$$

$$\implies \langle b_i|v\rangle = x_1 \langle b_i|b_1\rangle + x_2 \langle b_i|b_2\rangle + \cdots + x_i \langle b_i|b_i\rangle + \cdots + x_n \langle b_i|b_n\rangle$$

$$\implies \langle b_i|v\rangle = x_1 \times 0 + x_2 \times 0 + \cdots + x_i \times 1 + \cdots + x_n \times 0$$

$$\implies x_i = \langle b_i|v\rangle$$

This means, $x_1 = \langle b_1|v\rangle$, $x_2 = \langle b_2|v\rangle$ and so on.

$$\implies \boldsymbol{|v\rangle = \langle b_1|v\rangle |b_1\rangle + \langle b_2|v\rangle |b_2\rangle + \cdots + \langle b_i|v\rangle |b_i\rangle + \cdots + \langle b_n|v\rangle |b_n\rangle}$$

What does all this mean? We have already seen that different orthonormal bases correspond to choosing different orientations to measure spin. **The numbers given by the brakets like $\langle b_i|v\rangle$ are called probability amplitudes. The square of $\langle b_i|v\rangle$ will give us the probability of $|v\rangle$ jumping to $|b_i\rangle$ when we measure it**.

3.9 Ordered Basis

An **ordered basis** is a basis in which the vectors have been given an order, that is, there is a first vector, a second vector, and so on.

If $\{|b_1\rangle, |b_2\rangle, \cdots, |b_n\rangle\}$ is a basis, an ordered basis is represented as $(|b_1\rangle, |b_2\rangle, \cdots, |b_n\rangle)$.

Consider the standard basis in $\mathbb{R}^2$ - $\{|\uparrow\rangle, |\downarrow\rangle\}$.

Two sets are equal if they have the same elements — the order of the elements does not matter, so $\{|\uparrow\rangle, |\downarrow\rangle\} = \{|\downarrow\rangle, |\uparrow\rangle\}$. The two sets are identical.

However, for an ordered basis the order the basis vectors are given matters. $(|\uparrow\rangle, |\downarrow\rangle) \neq (|\downarrow\rangle, |\uparrow\rangle)$. The first vector in the ordered basis on the left is not equal to the first vector in the ordered basis on the right, so the two ordered bases are distinct.

We know that the standard basis $\{|\uparrow\rangle, |\downarrow\rangle\}$ corresponds to measuring the spin of an electron in the vertical direction. The ordered basis $(|\uparrow\rangle, |\downarrow\rangle)$ will correspond to measuring the spin when the south magnet is on top of our measuring apparatus. If we flip the apparatus through 180°, we will also flip the basis elements and use the ordered basis $(|\downarrow\rangle, |\uparrow\rangle)$.

3.10 Length of vectors, once again

Assuming a ket $|v\rangle$ and an orthonormal basis $\{|b_1\rangle, |b_2\rangle, \cdots, |b_n\rangle\}$, we now know that this $|v\rangle$ can be expressed in terms of the basis vectors.

$$\boldsymbol{|v\rangle = \langle b_1|v\rangle\,|b_1\rangle + \langle b_2|v\rangle\,|b_2\rangle + \cdots + \langle b_i|v\rangle\,|b_i\rangle + \cdots + \langle b_n|v\rangle\,|b_n\rangle}$$

We can write this as:

$$\boldsymbol{|v\rangle = c_1\,|b_1\rangle + c_2\,|b_2\rangle + \cdots + c_i\,|b_i\rangle + \cdots + c_n\,|b_n\rangle} \text{ where } c_i = \langle b_i|v\rangle\,\forall i$$

We will prove that length of vector $|v\rangle$ can be expressed as:

$$||v\rangle| = c_1^2 + c_2^2 + \cdots + c_i^2 + \cdots + c_n^2$$

By definition, we know, $||v\rangle| = \langle v|v\rangle$

$$\implies ||v\rangle| = (c_1\langle b_1| + c_2\langle b_2| + \cdots + c_i\langle b_i| + \cdots + c_n\langle b_n|)(c_1|b_1\rangle + c_2|b_2\rangle + \cdots + c_i|b_i\rangle + \cdots + c_n|b_n\rangle)$$

If we expand the above expression, we notice that there will be n^2 terms. We do not have to evaluate all these terms - we can use the property of orthonormal basis to simplify it.

We know that $\langle b_i|b_k\rangle = 0$ when $i \neq k$ and $\langle b_i|b_k\rangle = 1$ when $i = k$.

$$\implies ||v\rangle| = {c_1}^2 \langle b_1|b_1\rangle + c_1c_2 \langle b_1|b_2\rangle + \cdots + c_1c_n \langle b_1|b_n\rangle + \cdots + {c_n}^2 \langle b_n|b_n\rangle$$

Using the above rules, it can be simplified to:

$$||v\rangle| = \langle v|v\rangle = c_1^2 + c_2^2 + \cdots + c_i^2 + \cdots + c_n^2$$

3.11 Matrices

Wikipedia defines a matrix (plural matrices) as a rectangular array or table of numbers, symbols, or expressions, arranged in rows and columns, which is used to represent a mathematical object or a property of such an object.

A matrix $\mathbf{M}$ with m rows and n columns is called an $m \times n$ matrix. As an example, consider:

$$\mathbf{A} = \begin{bmatrix} 1 & -4 & 2 \\ 2 & 3 & 0 \end{bmatrix} \text{ and } \mathbf{B} = \begin{bmatrix} 1 & 2 \\ 7 & 5 \\ 6 & 1 \end{bmatrix}$$

Going by above definition, $\mathbf{A}$ is a 2×3 matrix and $\mathbf{B}$ is a 3×2 matrix.

We can consider bras and kets as being special types of matrices: bras have just one row, and kets have just one column.

The transpose of a $m \times n$ matrix $\mathbf{M}$, denoted $\mathbf{M}^\top$, is the $n \times m$ matrix formed by interchanging the rows and the columns of $\mathbf{M}$. The i^{th} row of $\mathbf{M}$ becomes the i^{th} column of $\mathbf{M}^\top$, and the j^{th} column of $\mathbf{M}$ becomes the j^{th} row of $\mathbf{M}^\top$. For our matrices $\mathbf{A}$ and $\mathbf{B}$ we have:

$$\mathbf{A}^\top = \begin{bmatrix} 1 & 2 \\ -4 & 3 \\ 2 & 0 \end{bmatrix} \text{ and } \mathbf{B}^\top = \begin{bmatrix} 1 & 7 & 6 \\ 2 & 5 & 1 \end{bmatrix}$$

The addition and subtraction rules for matrices are similar to those for vectors. Matrix multiplication is different and we look at it now.

Column vectors can be considered as matrices with just one column, and row vectors can be considered as matrices with just one row. With this interpretation, the relation between bras and kets with the same name is given by $\langle a| = |a\rangle^\top$ and $|a\rangle = \langle a|^\top$

Given a general matrix that has multiple rows and columns, we think of the rows as denoting bras and the columns as denoting kets. In the above example, we can think of **A** as consisting of two bras stacked on one another or as three kets side by side. Similarly, **B** can be considered as three bras stacked on one another or as two kets side by side.

The product of the matrices **A** and **B** uses this idea. The product is denoted by **AB**. It's calculated by thinking of **A** as consisting of bras and **B** of kets. Remember that bras always come before kets.

$$\mathbf{A} = \begin{bmatrix} \langle a_1| \\ \\ \langle a_2| \end{bmatrix} \text{ where } \langle a_1| = \begin{bmatrix} 1 & -4 & 2 \end{bmatrix} \text{ and } \langle a_2| = \begin{bmatrix} 2 & 3 & 0 \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} |b_1\rangle & |b_2\rangle \end{bmatrix} \text{ where } |b_1\rangle = \begin{bmatrix} 1 \\ \\ 7 \\ \\ 6 \end{bmatrix} \text{ and } |b_2\rangle = \begin{bmatrix} 2 \\ \\ 5 \\ \\ 1 \end{bmatrix}$$

The product **AB** is defined as:

$$\mathbf{AB} = \begin{bmatrix} \langle a_1| \\ \\ \langle a_2| \end{bmatrix} \begin{bmatrix} |b_1\rangle & |b_2\rangle \end{bmatrix} = \begin{bmatrix} \langle a_1|b_1\rangle & \langle a_1|b_2\rangle \\ \langle a_2|b_1\rangle & \langle a_2|b_2\rangle \end{bmatrix}$$

The matrix products are defined only when the dimensions of participating matrices match appropriately.

For matrix multiplication, the number of columns in the first matrix must be equal to the number of rows in the second matrix. The resulting matrix, known as the matrix product, has the number of rows of the first and the number of columns of the second matrix.

In our example, **A** is a 2×3 matrix and **B** is a 3×2 matrix. The product **AB** is a 2×2 matrix and the product **BA** is a 3×3 matrix.

In general, consider an $m \times r$ matrix **A** and an $r \times n$ matrix **B**, we can write **A** in terms of $r - dimensional$ bras and **B** in terms of $r - dimensional$ kets.

$$\mathbf{A_{mr}} = \begin{bmatrix} \langle a_1| \\ \\ \langle a_2| \\ \\ \vdots \\ \\ \langle a_n| \end{bmatrix} \quad \mathbf{B_{rn}} = \begin{bmatrix} |b_1\rangle & |b_2\rangle & \cdots & |b_n\rangle \end{bmatrix}$$

Since, the matrices **A** and **B** are of right dimensions, the **AB** is defined as:

$$\mathbf{AB_{mn}} = \begin{bmatrix} \langle a_1|b_1\rangle & \langle a_1|b_2\rangle & \dots & \langle a_1|b_j\rangle & \dots & \langle a_1|b_n\rangle \\ \langle a_2|b_1\rangle & \langle a_2|b_2\rangle & \dots & \langle a_2|b_j\rangle & \dots & \langle a_2|b_n\rangle \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \langle a_i|b_1\rangle & \langle a_i|b_2\rangle & \dots & \langle a_i|b_j\rangle & \dots & \langle a_i|b_n\rangle \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \langle a_m|b_1\rangle & \langle a_m|b_2\rangle & \dots & \langle a_m|b_j\rangle & \dots & \langle a_m|b_n\rangle \end{bmatrix}$$

Reversing the order of multiplication gives **BA**, but we cannot even begin the calculation if m is not equal to n because the bras and kets would have different dimensions. Even if m is equal to n, and we can multiply them, we would end up with a matrix that has size $r \times r$. This is not equal to **AB**, which has size $n \times n$, if n is not equal to r. Even in the case when n, m and r are all equal to one another, it is usually not the case that **AB** will equal **BA**. We say that matrix **multiplication is not commutative** to indicate this fact.

Matrices with the same number of rows as columns are called **square matrices**. A square matrix that has all leading diagonal entries equal to 1 and all other entries equal to 0 is called an **identity matrix**. The $n \times n$ identity matrix is denoted by I_n.

$$\mathbf{I_2} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}; \mathbf{I_3} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Suppose that **A** is an $m \times n$ matrix. Then $\mathbf{I_m A} = \mathbf{A I_n} = \mathbf{A}$.

What do we want to use these matrices for? Let's see.

3.12 Computing using Matrices

Suppose that we are given a set of $n-$*dimensional* kets $\{|b_1\rangle, |b_2\rangle, \cdots, |b_n\rangle\}$ and we want to check to see form an orthonormal basis. First, we have to check that they are all unit vectors. Then we have to check that the vectors are mutually orthogonal to one another. We have seen how to check both of these conditions using bras and kets, but the calculation can be expressed simply using matrices.

Let $\mathbf{A} = \left[|b_1\rangle\, |b_2\rangle \cdots |b_n\rangle\right]$ and then we take its trannspse.

$$\mathbf{A}^\top = \begin{bmatrix} \langle b_1| \\ \\ \langle b_2| \\ \\ \vdots \\ \\ \langle b_n| \end{bmatrix}$$

$$\mathbf{A}^\top\mathbf{A} = \begin{bmatrix} \langle b_1| \\ \\ \langle b_2| \\ \\ \vdots \\ \\ \langle b_n| \end{bmatrix} \begin{bmatrix} |b_1\rangle & |b_2\rangle & \cdots & |b_n\rangle \end{bmatrix} = \begin{bmatrix} \langle b_1|b_1\rangle & \langle b_1|b_2\rangle & \dots & \langle b_1|b_n\rangle \\ \langle b_2|b_1\rangle & \langle b_2|b_2\rangle & \dots & \langle b_2|b_n\rangle \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ \langle b_n|b_1\rangle & \langle b_n|b_2\rangle & \dots & \langle b_n|b_n\rangle \end{bmatrix}$$

The entries down the main diagonal are exactly what we need to calculate to find if the kets are unit. And the entries off the diagonal are what we have to calculate to see if the kets are mutually orthogonal. This means that **a set of vectors is an orthonormal basis if and only if $\mathbf{A}^\top\mathbf{A} = \mathbf{I_n}$**.

Given a ket $|v\rangle$ and an orthonormal basis $\{|b_1\rangle, |b_2\rangle, \cdots, |b_n\rangle\}$, we now know that this $|v\rangle$ can be expressed in terms of the basis vectors.

$$\boldsymbol{|v\rangle = \langle b_1|v\rangle\,|b_1\rangle + \langle b_2|v\rangle\,|b_2\rangle + \cdots + \langle b_i|v\rangle\,|b_i\rangle + \cdots + \langle b_n|v\rangle\,|b_n\rangle}$$

Using the matrix representation, this can also be expressed as follow.

$$\mathbf{A}^\top|v\rangle = \begin{bmatrix} \langle b_1| \\ \\ \langle b_2| \\ \\ \vdots \\ \\ \langle b_n| \end{bmatrix} |v\rangle = \begin{bmatrix} \langle b_1|v\rangle \\ \\ \langle b_2|v\rangle \\ \\ \vdots \\ \\ \langle b_n|v\rangle \end{bmatrix}$$

A square matrix $\mathbf{A}$ that has real entries and has the property that $\mathbf{A}^\top\mathbf{A}$ is equal to the identity matrix is called an **orthogonal matrix**.

3.13 Summary

Building upon concepts in this chapter, this section will summarize the steps that are routinely required to be done.

1. Checking for Orthonormal bias

Given a set of $n-dimensional$ kets $\{|b_1\rangle, |b_2\rangle, \cdots, |b_n\rangle\}$, how to check if they form an orthonormal basis.

To do this, first construct $\mathbf{A} = \begin{bmatrix} |b_1\rangle & |b_2\rangle & \cdots & |b_n\rangle \end{bmatrix}$. Then compute $\mathbf{A}\mathbf{A}^\top$. If this is the identity matrix, we have an orthonormal basis.

2. Representing a vector as a linear combination of basis vectors

Consider n ket $|v\rangle$ and an orthonormal basis $\{b_1, b_2, \cdots, b_n\}$. The ket $|v\rangle$ can be expressed as a linear combination of basis vector:

$$|v\rangle = x_1 |b_1\rangle + x_2 |b_2\rangle + \cdots + x_i |b_i\rangle + \cdots + x_n |b_n\rangle$$

Now, these x_i are to be identified.

Let $\mathbf{A} = \begin{bmatrix} |b_1\rangle & |b_2\rangle & \cdots & |b_n\rangle \end{bmatrix}$. Then, we have:

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \mathbf{A}^\top |v\rangle = \begin{bmatrix} \langle b_1| \\ \langle b_2| \\ \vdots \\ \langle b_n| \end{bmatrix} |v\rangle = \begin{bmatrix} \langle b_1|v\rangle \\ \langle b_2|v\rangle \\ \vdots \\ \langle b_n|v\rangle \end{bmatrix}$$

3. Length of a vector Assuming a ket $|v\rangle$ and an orthonormal basis $\{|b_1\rangle, |b_2\rangle, \cdots, |b_n\rangle\}$, we know:

$$|v\rangle = \langle b_1|v\rangle |b_1\rangle + \langle b_2|v\rangle |b_2\rangle + \cdots + \langle b_i|v\rangle |b_i\rangle + \cdots + \langle b_n|v\rangle |b_n\rangle$$

$$\implies |v\rangle = c_1 |b_1\rangle + c_2 |b_2\rangle + \cdots + c_i |b_i\rangle + \cdots + c_n |b_n\rangle \text{ where } c_i = \langle b_i|v\rangle \, \forall i$$

$$||v\rangle| = c_1^2 + c_2^2 + \cdots + c_i^2 + \cdots + c_n^2$$

4. Summary of Common Basis Vectors

$$|\uparrow\rangle = \begin{bmatrix} \mathbf{1} \\ \mathbf{0} \end{bmatrix} \; |\downarrow\rangle = \begin{bmatrix} \mathbf{0} \\ \mathbf{1} \end{bmatrix}$$

$$|\rightarrow\rangle = \begin{bmatrix} \frac{\mathbf{1}}{\sqrt{\mathbf{2}}} \\ \frac{\mathbf{-1}}{\sqrt{\mathbf{2}}} \end{bmatrix} \; |\leftarrow\rangle = \begin{bmatrix} \frac{\mathbf{1}}{\sqrt{\mathbf{2}}} \\ \frac{\mathbf{1}}{\sqrt{\mathbf{2}}} \end{bmatrix}$$

$$|\nearrow\rangle = \begin{bmatrix} \frac{\mathbf{1}}{\mathbf{2}} \\ \\ \frac{\mathbf{-3}}{\sqrt{\mathbf{2}}} \end{bmatrix} \; |\swarrow\rangle = \begin{bmatrix} \frac{\mathbf{3}}{\sqrt{\mathbf{2}}} \\ \\ \frac{\mathbf{1}}{\mathbf{2}} \end{bmatrix}$$

These three basis can be written as $\{|\uparrow\rangle, |\downarrow\rangle\}$, $\{|\rightarrow\rangle, |\leftarrow\rangle\}$ and $\{|\nearrow\rangle, |\swarrow\rangle\}$.

Lastly, since they are orthonormal, we have the following:

$\langle\uparrow | \uparrow\rangle = 1$, $\langle\downarrow | \downarrow\rangle = 1$, $\langle\uparrow | \downarrow\rangle = 0$, $\langle\downarrow | \uparrow\rangle = 0$

$\langle\rightarrow | \rightarrow\rangle = 1$, $\langle\leftarrow | \leftarrow\rangle = 1$, $\langle\rightarrow | \leftarrow\rangle = 0$, $\langle\leftarrow | \rightarrow\rangle = 0$

$\langle\nearrow | \nearrow\rangle = 1$, $\langle\swarrow | \swarrow\rangle = 1$, $\langle\nearrow | \swarrow\rangle = 0$, $\langle\swarrow | \nearrow\rangle = 0$

Chapter 4

Qubits

In an earlier chapter, we saw the effects of measurement on the spin of an electron. We noticed that if we measure spin in the vertical direction, we don't get continuous values, instead just two of them (either the electron has its north-pole directed vertically upwards, or it's vertically downward). We also noticed that if we measure the spin, first in the vertical direction, and then once again in the same direction, we get the same result for both measurements. As part of another experiment, we noticed that if we first measure in the vertical direction and then in the horizontal direction, the electrons will have spin **N** and **S** in the 90° direction with equal probability of $\frac{1}{2}$. The result of the second measurement in the horizontal direction was independent of the result of the first measurement in the vertical direction.

Before we delve deeper into qubits, let's briefly discuss probabilities that we have seen to be at the center of measurements.

4.1 Probabilities

Let's take a coin and repeatedly toss it and count the number of tosses and number of times it lands "heads". Assuming that the coin is fair, if we toss it a large number of times, we will find that number of "heads" to the total number of tosses will be closed to 0.5. In other words, we can say that the probability of "heads" is 0.5.

When we do an experiment, it is assumed that it has a finite number of possible outcomes. Lets call these $E_1, E2, \ldots E_n$. It is further assumed that when an

experiment is done, the result of the experiment, or measurement, will be one and only one of these n possibilities. Also, with each outcome E_i, a number is associated, p_i, called **probability**. Probabilities must be numbers between 0 and 1 that sum to 1. In our experiment of a coin toss, the two possible outcomes are landing a "head" or a "tail". Further, since the coin is fair, the probability of each event is $\frac{1}{2}$.

Let's go back to the experiment where we measure the spin of an electron as described in an earlier chapter. We will measure the spin of the electron in direction 0°. There are two possible outcomes that we will denote as **N** and **S** and each of these outcomes have an associated probability. We denote by p_N the probability of obtaining **N**, and p_S the probability of obtaining **S**. If we know that the electron has spin **N** in direction 0°, then we know that when we measure again in the same direction we will get the same result. So, in this case, $p_N = 1$ and $p_S = 0$. On the other hand, if we know our electron has spin **N** in direction 90° and we now measure in direction 0°, then we are equally likely to obtain **N** and **S** as the outcome, so, in this case, $p_N = p_S = 0.5$.

4.2 Maths behind Spin and Probability

In an experiment, when we make a measurement, there will be several possible outcomes. This number of possible outcomes defines that underlying **vector space**.

In the case of electron spin, there are only two possible outcomes for any measurement, so the underlying vector space is 2-dimensional which can be modeled as $\mathbb{R}^2$. This works fine since we are rotating our measurement setup in a plane. Even if we chose to rotate the measurement setup in a three-dimensions, the underlying vector space would still be two-dimensional, since there are only two possible outcomes for each measurement, and instead of vectors with real coefficients, we will use vectors with complex numbers. In this case, the vector space can be modeled as $\mathbb{C}^2$

Furthermore, we will be restricting ourselves to onlyl unit vectors in $\mathbb{R}^2$. For kets, this means we limit ourselves to kets of the form $|v\rangle = \begin{bmatrix} c_1 \\ c_2 \end{bmatrix}$ where ${c_1}^2 + {c_2}^2 = 1$.

When we chose a direction to measure electron spin, it means choosing an ordered orthonormal basis $(|b_1\rangle, |b_2\rangle)$ where the two basis vectors represent the two possible outcomes of the experiment. As a practice, we associate **N** with the first basis vector and **S** with the second. Before the spin is measured, the electron will be in a state given by a linear combination of $|b_1\rangle$ and $|b_2\rangle$, that

is, of the form $c_1 |b_1\rangle + c_2 |b_2\rangle$. After the measurement, the state vector jumps to either $|b_1\rangle$ or $|b_2\rangle$. This raises one of the main ideas in Quantum Mechanics - **measurement causes the state vector to change**. The new state is one of the basis vectors associated with the measurement and the probability of getting a particular basis vector is given by the initial state. The probability of state being $|b_1\rangle$ is ${c_1}^2$ and being $|b_2\rangle$ is ${c_2}^2$. The numbers c_1 and c_2 are called probability amplitudes and the square of these amplitudes is the probability.

We have seen in previous chapter, the ordered orthonormal basis corresponding to measuring spin in the vertical direction is given by $(|\uparrow\rangle, |\downarrow\rangle)$ where $|\uparrow\rangle = \begin{bmatrix} \mathbf{1} \\ \\ \mathbf{0} \end{bmatrix}$ and $|\downarrow\rangle = \begin{bmatrix} \mathbf{0} \\ \\ \mathbf{1} \end{bmatrix}$. The first vector listed in the basis corresponds to the electron having spin **N** in direction 0° and the second vector to **S** in direction 0°.

Also, the ordered orthonormal basis corresponding to measuring spin in the horizontal direction is given by $(|\rightarrow\rangle, |\leftarrow\rangle)$ where $|\rightarrow\rangle = \begin{bmatrix} \frac{\mathbf{1}}{\sqrt{\mathbf{2}}} \\ \\ \frac{\mathbf{-1}}{\sqrt{\mathbf{2}}} \end{bmatrix}$ and $|\leftarrow\rangle = \begin{bmatrix} \frac{\mathbf{1}}{\sqrt{\mathbf{2}}} \\ \\ \frac{\mathbf{1}}{\sqrt{\mathbf{2}}} \end{bmatrix}$. The first vector listed in the basis corresponds to the electron having spin **N** in direction 90° and the second vector to **S** in direction 90°.

We start by measuring the spin of the electron in the vertical direction. We might not know the spin state of the incoming electron, but we know that it can be represented as $c_1 |\uparrow\rangle + c_2 |\downarrow\rangle$ where ${c_1}^2 + {c_2}^2 = 1$. We now make the measurement. There are only two possibilities - either the electron is diverted upward in which case the state jumps to $|\uparrow\rangle$ or the electron is diverted downward in which case the state jumps to $|\downarrow\rangle$. Lastly, the probability that it's diverted upward is ${c_1}^2$, and the probability that it's diverted downward is ${c_2}^2$.

We repeat the experiment once again and measure the spin once more in the vertical direction. Suppose that the electron was deflected upward by the first set of magnets, so we know that its spin state is $|\uparrow\rangle = 1 |\uparrow\rangle + 0 |\downarrow\rangle$. When we measure again, the state will jump to $|\uparrow\rangle$ with probability $1^2 = 1$, or to $|\downarrow\rangle$ with probability $0^2 = 0$. This means, if we repeat the measurement in the same direction, we get the same result. In this case, the electron stays in the state $|\uparrow\rangle$.

Similarly, if the electron was deflected downward initially, it will be in state $|\downarrow\rangle = 0 |\uparrow\rangle + 1 |\downarrow\rangle$. We know that no matter how many times we measure it in

the vertical direction, it will remain in this state. If we repeat exactly the same experiment, we get exactly the same answer.

Let us change the experiment a bit now. Instead of repeatedly measuring spin in the vertical direction, we will first measure in the vertical direction and then measure in the horizontal direction. Suppose that we have just performed the first measurement and found that the electron has spin **N** in direction 0° and its state vector is $|\uparrow\rangle$. Next, we are going to measure spin in the horizontal direction and we need to represent this vector in terms of orthonormal basis that corresponds to this (horizontal) direction. In other words, we must find the values x_1 and x_2 solving $|\uparrow\rangle = x_1 |\rightarrow\rangle + x_2 |\leftarrow\rangle$.

How do we do this? We have already seen this in previous chapter.

First, we construct the matrix **A** by stacking the kets that form the orthonormal basis side by side.

$$\mathbf{A} = \begin{bmatrix} |\rightarrow\rangle & |\leftarrow\rangle \end{bmatrix} = \begin{bmatrix} \frac{\mathbf{1}}{\sqrt{\mathbf{2}}} & \frac{\mathbf{-1}}{\sqrt{\mathbf{2}}} \\ \\ \frac{\mathbf{1}}{\sqrt{\mathbf{2}}} & \frac{\mathbf{1}}{\sqrt{\mathbf{2}}} \end{bmatrix}$$

Now, we calculate $\mathbf{A}^\top |\uparrow\rangle$ to get the probability amplitudes for the new basis.

$$\mathbf{A}^\top |\uparrow\rangle = \begin{bmatrix} \frac{\mathbf{1}}{\sqrt{\mathbf{2}}} & \frac{\mathbf{-1}}{\sqrt{\mathbf{2}}} \\ \\ \frac{\mathbf{1}}{\sqrt{\mathbf{2}}} & \frac{\mathbf{1}}{\sqrt{\mathbf{2}}} \end{bmatrix} \begin{bmatrix} \mathbf{1} \\ \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \frac{\mathbf{1}}{\sqrt{\mathbf{2}}} \\ \\ \frac{\mathbf{1}}{\sqrt{\mathbf{2}}} \end{bmatrix}$$

Thus, we get $|\uparrow\rangle = \frac{\mathbf{1}}{\sqrt{\mathbf{2}}} |\rightarrow\rangle + \frac{\mathbf{1}}{\sqrt{\mathbf{2}}} |\leftarrow\rangle$

This means that when we measure in horizontal direction, the state will jump to $|\rightarrow\rangle$ with probability $(\frac{\mathbf{1}}{\sqrt{\mathbf{2}}})^2 = \frac{\mathbf{1}}{\mathbf{2}}$, or it will jump to $|\leftarrow\rangle$ with probability $(\frac{\mathbf{1}}{\sqrt{\mathbf{2}}})^2 = \frac{\mathbf{1}}{\mathbf{2}}$. This means that the probability that the electron has spin **N** in the 90° direction is equal to the probability that it has spin **S** in the 90° direction; both probabilities are exactly one-half.

In the earlier experiment, the spin was measured three times. First, in the vertical direction, and then in the horizontal direction, and finally again in the vertical direction. After the second measurement, the state vector of the electron will have one of two values, either $|\rightarrow\rangle$ or $|\leftarrow\rangle$. As part of the experiment, in the third step, we will now measure the spin again in the vertical direction. This

means that should represent $|\rightarrow\rangle$ and $|\leftarrow\rangle$ in terms of basis vectors of vertical direction, $|\uparrow\rangle$ or $|\downarrow\rangle$.

Let's do it step by step.

Let $\mathbf{A} = \begin{bmatrix} |\uparrow\rangle & |\downarrow\rangle \end{bmatrix} = \begin{bmatrix} \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} \end{bmatrix}$

Now, we calculate $\mathbf{A}^\top|\rightarrow\rangle$ to get the probability amplitudes with respect to the new basis.

$$\mathbf{A}^\top|\rightarrow\rangle = \begin{bmatrix} \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} \end{bmatrix} \begin{bmatrix} \frac{\mathbf{1}}{\sqrt{\mathbf{2}}} \\ \frac{\mathbf{-1}}{\sqrt{\mathbf{2}}} \end{bmatrix} = \begin{bmatrix} \frac{\mathbf{1}}{\sqrt{\mathbf{2}}} \\ \frac{\mathbf{-1}}{\sqrt{\mathbf{2}}} \end{bmatrix}$$

Thus, we get $|\rightarrow\rangle = \frac{\mathbf{1}}{\sqrt{\mathbf{2}}}|\uparrow\rangle - \frac{\mathbf{1}}{\sqrt{\mathbf{2}}}|\downarrow\rangle$

Similarly, $\mathbf{A}^\top|\leftarrow\rangle = \begin{bmatrix} \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} \end{bmatrix} \begin{bmatrix} \frac{\mathbf{1}}{\sqrt{\mathbf{2}}} \\ \frac{\mathbf{1}}{\sqrt{\mathbf{2}}} \end{bmatrix} = \begin{bmatrix} \frac{\mathbf{1}}{\sqrt{\mathbf{2}}} \\ \frac{\mathbf{1}}{\sqrt{\mathbf{2}}} \end{bmatrix}$

Thus, we get $|\leftarrow\rangle = \frac{\mathbf{1}}{\sqrt{\mathbf{2}}}|\uparrow\rangle + \frac{\mathbf{1}}{\sqrt{\mathbf{2}}}|\downarrow\rangle$

In summary, after the third measurement, we have:

$$|\rightarrow\rangle = \frac{\mathbf{1}}{\sqrt{\mathbf{2}}}|\uparrow\rangle - \frac{\mathbf{1}}{\sqrt{\mathbf{2}}}|\downarrow\rangle$$

$$|\leftarrow\rangle = \frac{\mathbf{1}}{\sqrt{\mathbf{2}}}|\uparrow\rangle + \frac{\mathbf{1}}{\sqrt{\mathbf{2}}}|\downarrow\rangle$$

This means, in either case, when we measure spin in the vertical direction the state vector will jump to either $|\uparrow\rangle$ or to $|\downarrow\rangle$, each occurring with probability one-half.

4.3 Equivalent State Vectors

We have a collection of electrons and their spins are either $|\uparrow\rangle$ or $-|\uparrow\rangle$. Is there any experiment that we can conduct that can tell them apart? In short, NO. Let's see why is it so?

Let's pick a direction to measure the spin. This is same as selecting an ordered orthonormal basis and let it be denoted by $(|b_1\rangle, |b_2\rangle)$.

Suppose the electron has state $|\uparrow\rangle$, so we must find values x and y such that $|\uparrow\rangle = x\,|b_1\rangle + y\,|b_2\rangle$. When we do the measurement, as usual, the probability of spin being **N** is x^2 and the probability of spin being **S** is y^2.

Next, assume that the electron has state $-|\uparrow\rangle$. For the same values x, y we have $-|\uparrow\rangle = -x\,|b_1\rangle - y\,|b_2\rangle$. When we do the measurement, as usual, the probability of spin being **N** is $(-x)^2 = x^2$ and the probability of spin being **S** is $(-y)^2 = y^2$.

This shows that we get the same probabilities in both the same cases and thus no measurement can differentiate between electrons with state vectors $|\uparrow\rangle$ and $-|\uparrow\rangle$.

Building on the same argument, given electrons with state $|\nu\rangle$, there is no way to differentiate them from electrons with state $-|\nu\rangle$. Since the states are the same, they can be called equivalent. In other words, saying that an electron has a state given by $|\nu\rangle$ is the same as saying that it has spin given by $-|\nu\rangle$.

To take a concrete example, consider these four kets:

$$\frac{1}{\sqrt{2}}|\uparrow\rangle + \frac{1}{\sqrt{2}}|\downarrow\rangle,\ -\frac{1}{\sqrt{2}}|\uparrow\rangle - \frac{1}{\sqrt{2}}|\downarrow\rangle,\ \frac{1}{\sqrt{2}}|\uparrow\rangle - \frac{1}{\sqrt{2}}|\downarrow\rangle,\ -\frac{1}{\sqrt{2}}|\uparrow\rangle + \frac{1}{\sqrt{2}}|\downarrow\rangle.$$

From the arguments earlier in this section, we know that $\frac{1}{\sqrt{2}}|\uparrow\rangle + \frac{1}{\sqrt{2}}|\downarrow\rangle$, and $-\frac{1}{\sqrt{2}}|\uparrow\rangle - \frac{1}{\sqrt{2}}|\downarrow\rangle$ are equivalent and same is true for $\frac{1}{\sqrt{2}}|\uparrow\rangle - \frac{1}{\sqrt{2}}|\downarrow\rangle$ and $-\frac{1}{\sqrt{2}}|\uparrow\rangle + \frac{1}{\sqrt{2}}|\downarrow\rangle$

How about these two: $\frac{1}{\sqrt{2}}|\uparrow\rangle + \frac{1}{\sqrt{2}}|\downarrow\rangle$, $\frac{1}{\sqrt{2}}|\uparrow\rangle - \frac{1}{\sqrt{2}}|\downarrow\rangle$. If we measure the spin in the vertical direction, these two kets are not distinguishable. In both cases, we get $|\uparrow\rangle$ or $|\downarrow\rangle$ each occurring with probability of a half. But we know that $|\rightarrow\rangle = \frac{1}{\sqrt{2}}|\uparrow\rangle + \frac{1}{\sqrt{2}}|\downarrow\rangle$ and $|\leftarrow\rangle = \frac{1}{\sqrt{2}}|\uparrow\rangle - \frac{1}{\sqrt{2}}|\downarrow\rangle$. Thus, if we measure in the 90° direction, we will obtain **S** for the first ket and **N** for the second. This choice of basis does distinguish them, and so they are not equivalent.

The results of this section will come handy in the next section.

4.4 Basis Vectors in any direction

We have seen that ordered basis with 0° direction is $\left(\begin{bmatrix} \mathbf{1} \\ \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} \mathbf{0} \\ \\ \mathbf{1} \end{bmatrix} \right)$ and the one associated with 90° direction is $\left(\begin{bmatrix} \frac{\mathbf{1}}{\sqrt{\mathbf{2}}} \\ \\ \frac{\mathbf{-1}}{\sqrt{\mathbf{2}}} \end{bmatrix} \begin{bmatrix} \frac{\mathbf{1}}{\sqrt{\mathbf{2}}} \\ \\ \frac{\mathbf{1}}{\sqrt{\mathbf{2}}} \end{bmatrix} \right)$.

Is there any method that can help identify the basis for any given direction? Let's find out more in this section.

We take our measurement setup, set the vertical direction as the starting point, and start rotating in the clockwise direction. When we have rotated by 90°, we will be making measurements in the horizontal direction. When we have rotated by 180°, we will be again measuring the vertical direction. We know that if an electron has spin **N** in 0° direction, it will have spin **S** in direction 180°. Also, if an electron has spin **S** in direction 0°, it will have spin **N** in the direction 180°.

We start with the Standard Basis as shown in figure below. This basis is represented as $\left(\begin{bmatrix} \mathbf{1} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} \mathbf{0} \\ \mathbf{1} \end{bmatrix} \right)$.

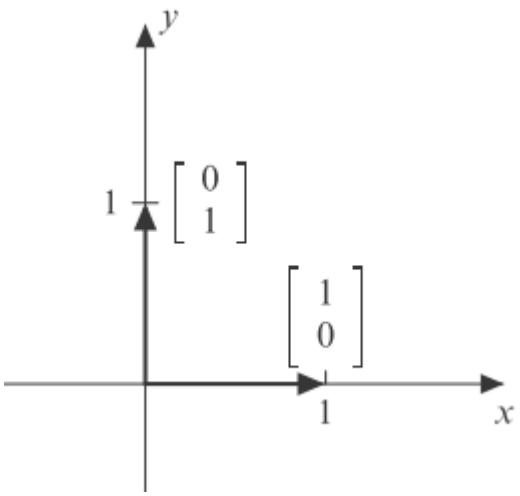

Figure 4.1: Standard Basis

Next we rotate these basis vectors. After rotation with α° it looks like as follow. Basic trigonometry, tells us that $\begin{bmatrix}\mathbf{1}\\\mathbf{0}\end{bmatrix}$ moves to $\begin{bmatrix}\mathbf{cos}(\alpha)\\-\mathbf{sin}(\alpha)\end{bmatrix}$, and $\begin{bmatrix}\mathbf{0}\\\mathbf{1}\end{bmatrix}$ moves to $\begin{bmatrix}\mathbf{sin}(\alpha)\\\mathbf{cos}(\alpha)\end{bmatrix}$.

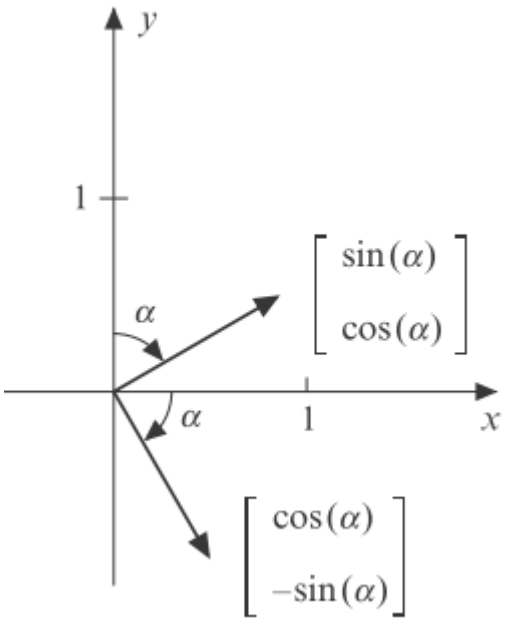

Figure 4.2: Standard Basis Rotated

In short, rotating via α° moves the initial ordered basis $\left(\begin{bmatrix}\mathbf{1}\\\mathbf{0}\end{bmatrix}, \begin{bmatrix}\mathbf{0}\\\mathbf{1}\end{bmatrix}\right)$ to $\left(\begin{bmatrix}\mathbf{cos}(\alpha)\\-\mathbf{sin}(\alpha)\end{bmatrix}, \begin{bmatrix}\mathbf{sin}(\alpha)\\\mathbf{cos}(\alpha)\end{bmatrix}\right)$.

Once the basis is rotated by 90°, the standard basis $\left(\begin{bmatrix}\mathbf{1}\\\mathbf{0}\end{bmatrix}, \begin{bmatrix}\mathbf{0}\\\mathbf{1}\end{bmatrix}\right)$ moves to $\left(\begin{bmatrix}\mathbf{cos}(\mathbf{90}^\circ)\\-\mathbf{sin}(\mathbf{90}^\circ)\end{bmatrix}, \begin{bmatrix}\mathbf{sin}(\mathbf{90}^\circ)\\\mathbf{cos}(\mathbf{90}^\circ)\end{bmatrix}\right)$ which is same as $\left(\begin{bmatrix}\mathbf{0}\\-\mathbf{1}\end{bmatrix}, \begin{bmatrix}\mathbf{1}\\\mathbf{0}\end{bmatrix}\right)$.

Recall from previous section that $\begin{bmatrix}\mathbf{0}\\-\mathbf{1}\end{bmatrix}$ is same as $\begin{bmatrix}\mathbf{0}\\\mathbf{1}\end{bmatrix}$. This means, rotation by 90° brings us back to a basis that is equivalent to original one, **except that the order of basis elements is interchanged**.

Let θ be the angle by which we rotate our measurement setup and let α be the angle by which we rotate our basis vectors. We have seen that we get a complete

set of directions as θ goes from 0° to 180°, and that we get a complete set of rotated bases as α goes from 0° to 90°. Once we reach $\theta = 180^\circ$ or equivalently $\alpha = 90^\circ$, **N** and **S** measured in direction 0° are interchanged.

Its natural to defined $\theta = 2\alpha$. Thus, the basis after the test setup is rotated by θ is given by $\left(\begin{bmatrix}\mathbf{cos(\theta/2)}\\ \mathbf{-sin(\theta/2)}\end{bmatrix}, \begin{bmatrix}\mathbf{sin(\theta/2)}\\ \mathbf{cos(\theta/2)}\end{bmatrix}\right)$.

Let us work on a concrete example and see what happens when we first measure the spin in 0° and then rotate the measurement setup by 60° and measure the spin again. If we had observed **N** as a result of the first measurement in $0\circ$ direction, what is the probability of measuring **N** when we have rotated measurement setup by 60°?

We know that once we have rotated our measurement setup by 60°, the new basis are $\left(\begin{bmatrix}\mathbf{cos(30^\circ)}\\ \mathbf{-sin(30^\circ)}\end{bmatrix}, \begin{bmatrix}\mathbf{sin(30^\circ)}\\ \mathbf{cos(30^\circ)}\end{bmatrix}\right)$ which is same as $\left(\begin{bmatrix}\frac{\sqrt{3}}{2}\\ -\frac{1}{2}\end{bmatrix}, \begin{bmatrix}\frac{1}{2}\\ \frac{\sqrt{3}}{2}\end{bmatrix}\right)$.

Since the electron was first observed to have spin **N** in direction 0°, its state vector after the initial measurement was $\begin{bmatrix}1\\0\end{bmatrix}$. We need to express this in terms of the new basis vectors that were obtained after rotating the setup and we already know how to do this.

Let $\mathbf{A} = \begin{bmatrix}\frac{\sqrt{3}}{2} & \frac{1}{2}\\ -\frac{1}{2} & \frac{\sqrt{3}}{2}\end{bmatrix}$

$$\implies \mathbf{A}^\top \begin{bmatrix}1\\0\end{bmatrix} = \begin{bmatrix}\frac{\sqrt{3}}{2} & -\frac{1}{2}\\ \frac{1}{2} & \frac{\sqrt{3}}{2}\end{bmatrix} \begin{bmatrix}1\\0\end{bmatrix} = \begin{bmatrix}\frac{\sqrt{3}}{2}\\ \frac{1}{2}\end{bmatrix}$$

This means, $\begin{bmatrix}1\\0\end{bmatrix} = \frac{\sqrt{3}}{2}\begin{bmatrix}\frac{\sqrt{3}}{2}\\ -\frac{1}{2}\end{bmatrix} + \frac{1}{2}\begin{bmatrix}\frac{1}{2}\\ \frac{\sqrt{3}}{2}\end{bmatrix}$

So, the probability of observing **N** while measuring in 60° is $(\frac{\sqrt{3}}{2})^2 = \frac{3}{4}$.

4.5 Revisiting Photon Polarization, mathematically

As convention, we associate 0° with a filters that is polarized vertically. This filter will allow photos that are polarized verticxally to pass and block horizontally polarized photons. Furthermore, we assign standard basis $\left(\begin{bmatrix}\mathbf{1}\\\mathbf{0}\end{bmatrix}, \begin{bmatrix}\mathbf{0}\\\mathbf{1}\end{bmatrix}\right)$ to the direction 0°. The vector $\begin{bmatrix}\mathbf{1}\\\mathbf{0}\end{bmatrix}$ will correspond to vertically polarized photon and the vector $\begin{bmatrix}\mathbf{0}\\\mathbf{1}\end{bmatrix}$ will correspond to horizontally polarized photon.

We now rotate the filter through an angle β° in the clockwise direction. This will allow to pass photons that are polarized in direction of β° and absorb photons that are polarized in the direction perpendicular to β°.

We now assign, for each direction of measurement, an ordered basis $(|b_1\rangle, |b_2\rangle)$ associated with making a polarization measurement in that direction. The ket $|b_1\rangle$ corresponds to a photon polarized in the given direction and it passes through the filter. The ket $|b_2\rangle$ corresponds to a photon polarized in the perpendicular direction and is absorbed by the filter.

A photon's polarization state caan be represented by ket $|\nu\rangle$ and can be represented as linear combiantion of basis vectors as: $|\nu\rangle = d_1 |b_1\rangle + d_2 |b_2\rangle$.

If we measure polarization in the direction of ordered basis $(|b_1\rangle, |b_2\rangle)$, we will find that photon is polarized in the given direction with probability ${d_1}^2$ and is polarized perpendicular to it with probability ${d_2}^2$. In other words, the probability that photon passes through the filter is ${d_1}^2$ and the probability that photon is absorbed by the filter is ${d_2}^2$. Lastly, if the result of the measurement is that the photon is polarized in the given direction — it passes through the filter — then the state of the photon becomes $|b_1\rangle$.

Recall from previous sections, if wew start with the standard basis $\left(\begin{bmatrix}\mathbf{1}\\\mathbf{0}\end{bmatrix}, \begin{bmatrix}\mathbf{0}\\\mathbf{1}\end{bmatrix}\right)$ and rotate the basis vectors by an angle α°, we get the new orthonormal basis as $\left(\begin{bmatrix}\mathbf{cos}(\alpha)\\-\mathbf{sin}(\alpha)\end{bmatrix}, \begin{bmatrix}\mathbf{sin}(\alpha)\\\mathbf{cos}(\alpha)\end{bmatrix}\right)$. Also, rotating through an angle of 90° brings us back to the same basis as the original, except that the order of the basis elements are interchanged.

Now we will rotate the polarized filter by an angle β°. When $\beta^\circ = 0^\circ$, the measurements are in vertical and horizontal direction, the vertically polarized photons passes through the filter and the horizontally polarized photons are absorbed by the filter. When $\beta^\circ = 90^\circ$, we will still be measuring the vertical and

horizontal directions, but now we observe that verticallly polarized photons are absorbed and horizontally polarized photons pass through the filter. Thus, we find that $\beta^\circ = 90^\circ$ (rotating filter setup) corresponds to $\alpha^\circ = 90^\circ$ (rotating basis vectors) and we can assume $\beta = \alpha$. We can say that the ordered orthonormal basis associated with rotating a polarized filter through an angle β is given by $\left(\begin{bmatrix} \mathbf{cos}(\beta) \\ -\mathbf{sin}(\beta) \end{bmatrix}, \begin{bmatrix} \mathbf{sin}(\beta) \\ \mathbf{cos}(\beta) \end{bmatrix}\right)$.

We now revisit the experiments of Chapter 2.

In the first experiment, we have two polarized squares. One measures polarization in 0° direction and the other in the 90° direction. No photons pass through the overlapped region, as shown in the below figure.

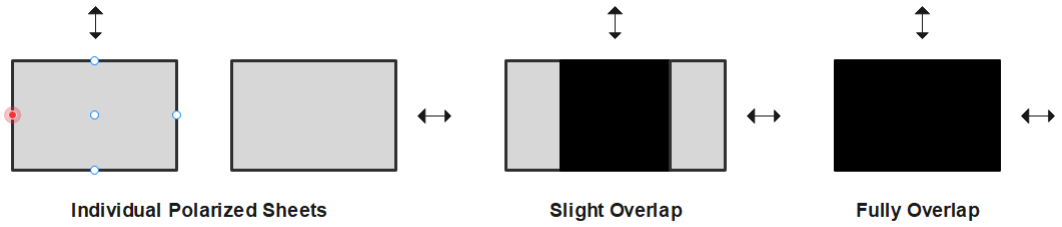

Figure 4.3: DifferentPolarization

The basis associated with 0° direction is the standard orthhonormal basis. The basis associated with 90° is same, except the order of elements is interchanged. A photon that has passed through the first filter has had a measurement made, its polarized in vertical direction, and its in state $\begin{bmatrix} \mathbf{1} \\ \mathbf{0} \end{bmatrix}$. We now measure it with second filter that allows the photons with state $\begin{bmatrix} \mathbf{0} \\ \mathbf{1} \end{bmatrix}$ to pass and absorb photons with state $\begin{bmatrix} \mathbf{1} \\ \mathbf{0} \end{bmatrix}$. Thus, we see that any photons that pass through first filter are absorbed by the second.

In the third experiment, we take the third sheet and rotate it through forty-five degrees and slide this square between the other two squares, one of which has a vertical orientation, and the other has a horizontal orientation. We note that some light comes through the region of overlap of all three squares. This is shown in the below figure.

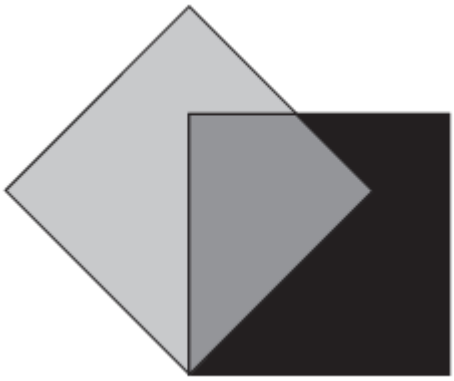

Figure 4.4: Three Polaroids

The ordered orthonormal basis for teh three filters are $\left(\begin{bmatrix}\mathbf{1}\\\mathbf{0}\end{bmatrix}, \begin{bmatrix}\mathbf{0}\\\mathbf{1}\end{bmatrix}\right)$, $\left(\begin{bmatrix}\frac{\mathbf{1}}{\sqrt{\mathbf{2}}}\\\frac{\mathbf{-1}}{\sqrt{\mathbf{2}}}\end{bmatrix} \begin{bmatrix}\frac{\mathbf{1}}{\sqrt{\mathbf{2}}}\\\frac{\mathbf{1}}{\sqrt{\mathbf{2}}}\end{bmatrix}\right)$ and $\left(\begin{bmatrix}\mathbf{0}\\\mathbf{1}\end{bmatrix}, \begin{bmatrix}\mathbf{1}\\\mathbf{0}\end{bmatrix}\right)$. A photon passing through this system will have three measurements made. Let's see one by one.

The photons that pass throughh first filter will be in $\left(\begin{bmatrix}\mathbf{1}\\\mathbf{0}\end{bmatrix}\right)$ state.

The second measurement corresponds to passing through the filter rotated by $45°$. We now write the state of the photon using the appropriate basis.

$$\begin{bmatrix}\mathbf{1}\\\mathbf{0}\end{bmatrix} = \frac{\mathbf{1}}{\sqrt{\mathbf{2}}}\begin{bmatrix}\frac{\mathbf{1}}{\sqrt{\mathbf{2}}}\\-\frac{\mathbf{1}}{\sqrt{\mathbf{2}}}\end{bmatrix} + \frac{\mathbf{1}}{\sqrt{\mathbf{2}}}\begin{bmatrix}\frac{\mathbf{1}}{\sqrt{\mathbf{2}}}\\\frac{\mathbf{1}}{\sqrt{\mathbf{2}}}\end{bmatrix}$$

Hence, the probability of a photon passing first through first filter and then through the second filter is $\left(\frac{1}{\sqrt{2}}\right)^2 = \frac{1}{2}$. This further means that half of the photons that pass through first filter will pass through second filter aand those that do will be in state $\begin{bmatrix}\frac{\mathbf{1}}{\sqrt{\mathbf{2}}}\\-\frac{\mathbf{1}}{\sqrt{\mathbf{2}}}\end{bmatrix}$.

The third filter causes measurement using the third basis. Writing the state of the photon that has just crossed the second filter in terms of the basis for the third filter, we get:

$$\begin{bmatrix} \frac{\mathbf{1}}{\sqrt{\mathbf{2}}} \\ -\frac{\mathbf{1}}{\sqrt{\mathbf{2}}} \end{bmatrix} = -\frac{\mathbf{1}}{\sqrt{\mathbf{2}}} \begin{bmatrix} \mathbf{0} \\ \mathbf{1} \end{bmatrix} + \frac{\mathbf{1}}{\sqrt{\mathbf{2}}} \begin{bmatrix} \mathbf{1} \\ \mathbf{0} \end{bmatrix}$$

The third filter allows photons in state $\begin{bmatrix} \mathbf{0} \\ \mathbf{1} \end{bmatrix}$ to pass through. The probability of this happening is $\left(-\frac{1}{\sqrt{2}}\right)^2 = \frac{1}{2}$, meaning half the photons that pass through the first two filters will pass through the third filter.

We have shown that the mathematical model of electron spin and photon measurement are identical. We move to Qubits now.

4.6 Qubits

A classical bit is either 0 or 1. It can be represented by anything that has two mutually exclusive states. In classical computer science, there is no concept of measurement of bit. A bit is a bit, and it's either 0 or 1 and that's all. But for qubits, the situation is more complicated, and measurement is a critical part of its mathematical description.

We define a qubit to be any unit ket in $\mathbb{R}^2$. Usuallly, give a qubit, we willw ant to measure it. If we are going on measure it, we are going to a need a direction of measurement. We accomplish by introducing an ordered orthonormal basis $(|b_0\rangle, |b_1\rangle)$ and now the qubit can be written as linear combination of the basis vectors, generally in the form $|\nu\rangle = d_0 |b_0\rangle + d_1 |b_1\rangle$. After the measurement, the qubit will either jump to $|b_0\rangle$ or $|b_1\rangle$. The probability of it being $|b_0\rangle$ is ${d_0}^2$ and the probability of it being $|b_1\rangle$ is ${d_1}^2$. We not connect the classical bits 0 and 1 to the basis vectors. We associate the $|b_0\rangle$ basic vecotr with bit the 0 and $|b_1\rangle$ basic vecotr with bit the 1. Thus, when we measure the qubit $d_0 |b_0\rangle + d_1 |b_1\rangle$, we will get 0 with probability ${d_0}^2$ and 1 with probability ${d_1}^2$.

Since a qubit can be any unit ket and there are infinitely many unit kets, there are infinitely many possible values for a qubit. This is quite unlike classical computation, where we just have two bits. It is important, however, to notice that to get information out of a qubit we have to measure it. When we measure it we will get either 0 or 1, so the result is a classical bit.

4.7 Introducing Alice, Bob and Eve

Alice wants to send a confidential message to Bob and Eve wants to eavesdrop on it. How should Alice encrypt the message so that Bob can read it but Eve cannot decipher it? Let us first consider the scenario of communication between Alice and Bob where Alice sends a stream of qubits to Bob.

Alice measures qubits using her orthonormal basis, which we will denote as $(|a_0\rangle, |a_1\rangle)$. Bob measures the qubits that Alice sends to him using his orthonormal basis $(|b_0\rangle, |b_1\rangle)$.

Let's consider the case where Alice wants to send 0. She can use her measurement setup to sort qubits into either staate $|a_0\rangle$ or $|a_1\rangle$. Since she has decided to send 0, send sends the qubit in state $|a_0\rangle$. We also know that Bob is measuring with respect to his ordered basis. To see what will happen, we write $|a_0\rangle$ as a linear combiantion of his basis vectors. It will be in the form $|a_0\rangle = d_0 |b_0\rangle + d_1 |b_1\rangle$. When Bob measures, either it jumps to state $|b_0\rangle$ with probability ${d_0}^2$ and he records a 0, or it jumps to state $|b_1\rangle$ with probability ${d_1}^2$ and he records a 1.

In another scenario, Alice and Bob can choose the same basis pair and this will ensure that Bob would receive 0 with certainty whenever Alice sends 0 and would receive 1 with certainty whenever Alice sends 1. So, why they don't do it? Enter the eavesdropper, Eve. If she also chooses the same basis, she will be read all the messages that Alice and Bob are exchanging with certainty.

Alice and Bob may choose to measure the qubits using the basis $\left(\begin{bmatrix}\mathbf{1}\\\mathbf{0}\end{bmatrix}, \begin{bmatrix}\mathbf{0}\\\mathbf{1}\end{bmatrix}\right)$ or $\left(\begin{bmatrix}\frac{\mathbf{1}}{\sqrt{\mathbf{2}}}\\\frac{\mathbf{-1}}{\sqrt{\mathbf{2}}}\end{bmatrix} \begin{bmatrix}\frac{\mathbf{1}}{\sqrt{\mathbf{2}}}\\\frac{\mathbf{1}}{\sqrt{\mathbf{2}}}\end{bmatrix}\right)$. The calculations are same as before where we considered spin measurement in vertical and horizontal direction except that we replace **N** with 0 and **S** with 1. If Alice and Bob use the same basis, Bob will get the exact bit that Alice sent. If they choose different basis, half the time Bob will get the right bit and half the time he will get the wrong bit. This doesn't seem very useful and intuitive, but we will see that this method can be used for setting up secure communication channel.

To consider a concrete example, let Alice measure in the 240° direction and Bob in the 120° direction. We know that orthonormal basis in direction θ is given by $\left(\begin{bmatrix}\mathbf{cos}(\theta/\mathbf{2})\\\mathbf{-sin}(\theta/\mathbf{2})\end{bmatrix}, \begin{bmatrix}\mathbf{sin}(\theta/\mathbf{2})\\\mathbf{cos}(\theta/\mathbf{2})\end{bmatrix}\right)$. This translates to Alice's basis as

$\left(\begin{bmatrix} \frac{-1}{2} \\ \frac{-\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} \frac{\sqrt{3}}{2} \\ \frac{-1}{2} \end{bmatrix}\right)$ and Bob's basis as $\left(\begin{bmatrix} \frac{1}{2} \\ \frac{-\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} \frac{\sqrt{3}}{2} \\ \frac{1}{2} \end{bmatrix}\right)$. Recall from previous section that multiplying a ket by -1 gives an equivalent ket, we can c rewrite Alice's basis as $\left(\begin{bmatrix} \frac{1}{2} \\ \frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} \frac{\sqrt{3}}{2} \\ \frac{-1}{2} \end{bmatrix}\right)$.

Since Alice wants to send 0, she sends $\begin{bmatrix} \frac{1}{2} \\ \frac{\sqrt{3}}{2} \end{bmatrix}$. To see what Bob measures we need to write this as a linear combination of his basis vectors. We can get the probability amplitudes by forming the matrix consisting of the bras of his basis vectors and then multiplying the qubit by this matrix.

$$\begin{bmatrix} \frac{1}{2} & \frac{-\sqrt{3}}{2} \\ \frac{\sqrt{3}}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} \frac{1}{2} \\ \frac{\sqrt{3}}{2} \end{bmatrix} = \begin{bmatrix} \frac{-1}{2} \\ \frac{\sqrt{3}}{2} \end{bmatrix}$$

This means, $\begin{bmatrix} \frac{1}{2} \\ \frac{\sqrt{3}}{2} \end{bmatrix} = \frac{-1}{2} \begin{bmatrix} \frac{1}{2} \\ \frac{-\sqrt{3}}{2} \end{bmatrix} + \frac{\sqrt{3}}{2} \begin{bmatrix} \frac{\sqrt{3}}{2} \\ \frac{1}{2} \end{bmatrix}$

This means that when Bob measures the qubit, he gets 0 with probability $\frac{1}{4}$ and 1 with probability $\frac{3}{4}$. Similarly, it can be checked that if Alice sends 1, Bob will get 1 with probability $\frac{1}{4}$ and 0 with probability $\frac{3}{4}$.

4.8 Interference

A qubit has the form $d_0 |b_0\rangle + d_1 |b_1\rangle$ where d_0 and d_1 are probability amplitudes. The square of these numbers gives the probabilities that the qubit jumps to the corresponding basis vector. Probabilities are not allowed to be negative, but the probability amplitudes can be negative. This is the fact that allows for constructive and destructive interference.

To take an example, consider the qubits denoted by $|\leftarrow\rangle$ and $|\rightarrow\rangle$. When these are measured in the standard basis, thhese will jump either to $|\uparrow\rangle$ or $|\downarrow\rangle$, each with probability $\frac{1}{2}$. We take a superposition now of original qubits: $\nu = \frac{1}{\sqrt{2}}|\leftarrow\rangle + \frac{1}{\sqrt{2}}|\rightarrow\rangle$. If we measure ν in the horizontal direction, we will either get$|\leftarrow\rangle$ or $|\rightarrow\rangle$ with equal probability. However, if we measure in vertical direction, we get 0 with certainity because we know:

$$\nu = \frac{1}{\sqrt{2}}|\leftarrow\rangle + \frac{1}{\sqrt{2}}|\rightarrow\rangle$$

$$\implies \nu = \frac{1}{\sqrt{2}}\begin{bmatrix}\mathbf{\frac{1}{\sqrt{2}}}\\ \mathbf{\frac{1}{\sqrt{2}}}\end{bmatrix} + \frac{1}{\sqrt{2}}\begin{bmatrix}\mathbf{\frac{1}{\sqrt{2}}}\\ \mathbf{\frac{-1}{\sqrt{2}}}\end{bmatrix}$$

$$\implies \nu = \mathbf{1}\begin{bmatrix}\mathbf{1}\\ \mathbf{0}\end{bmatrix} + \mathbf{0}\begin{bmatrix}\mathbf{0}\\ \mathbf{1}\end{bmatrix}$$

The terms in $|\leftarrow\rangle$ and $|\rightarrow\rangle$ that gives 0 have interfered constructively, and the terms that give 1 have interfered destructively.

Thus **it is important to choose linear combinations carefully**.

There are a very limited number of things that we can do with one qubit, but one thing we can do is to enable Alice and Bob to communicate securely.

4.9 The BB84 Protocol

We often want to send and receive secure messages. This is often accomplished by agreeing on a secret key among the participants and the participants use this key to encode/decode the messages. Alice and Bob also want to communicate securely and Eve wants to eavesdrop. Alice and Bob want to agree on a secret key, and they need to be sure that Eve does not know it.

The BB84 protocol derives its name from its inventors, Charles Bennett and Gilles Brassard, and the year that it was invented, 1984. It uses two sets of ordered orthonormal bases: the standard basis used to measure spin in the vertical direction, called **V**, and the standard basis used to measure spin in the horizontal direction, called **H**. In both cases, the classical bit 0 will correspond to the first vector in the ordered basis and 1 to the second.

$$\mathbf{V} = \left(\begin{bmatrix} \mathbf{1} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} \mathbf{0} \\ \mathbf{1} \end{bmatrix} \right)$$

$$\mathbf{H} = \begin{bmatrix} \frac{\mathbf{1}}{\sqrt{\mathbf{2}}} \\ \frac{\mathbf{-1}}{\sqrt{\mathbf{2}}} \end{bmatrix} \begin{bmatrix} \frac{\mathbf{1}}{\sqrt{\mathbf{2}}} \\ \frac{\mathbf{1}}{\sqrt{\mathbf{2}}} \end{bmatrix}$$

Alice chooses the key, a string of classical bits, that she wants to send to Bob. For each bit, Alice chooses one of the two bases **V** and **H** at random and with equal probability. She then sends Bob the qubit consisting of the appropriate basis vector. For example, if she wants to send 0 and chose **V**, she will send $\begin{bmatrix} \mathbf{1} \\ \mathbf{0} \end{bmatrix}$, if she chooses **H**, she will send $\begin{bmatrix} \frac{\mathbf{1}}{\sqrt{\mathbf{2}}} \\ \frac{\mathbf{-1}}{\sqrt{\mathbf{2}}} \end{bmatrix}$. Alice follows the same method for each bit, keeping a record of bases she has used for each bit. If the string is $4n$ binary digits long, she will end up with a string of length $4n$ consisting of **V**s and **H**s. The reason for length $4n$ will become clear by end of the section and it makes a few calculations easier.

On the receiving side, Bob also chooses between the two bases at random with equal probability and then measures the qubit in the chosen basis. Bob repeats this for each bit and keeps a record of which basis was used. At the end of transmission, Bob will have two strings to length $4n$ - one a string of 0 and 1 corresponding for measurement, the other consisting of corresponding **V**s and **H**s for the bases he chose.

Both Alice and Bob are choosing their basis for each bit at random. Half the time they will end up using the same basis and half the time they will be using a different basis. If they both have selected the same basis, then Bob will receive the bit that Alice has sent with certainty. If they have selected a different basis, half the time Bob will receive the right bit, while half the other time Bob will receive the wrong bit.

Alice and Bob now compare their strings of **V**s and **H**s over an open channel. They keep the bits corresponding to the times when they both used the same basis and erase the bits that correspond to times when they used different bases. If Eve is not intercepting the message, they both end up with the same string of binary digits that has a length about $2n$. They now check to see if Eve was listening in.

If Eve intercepts the qubit on the way from Alice to Bob, she would really like to clone it, sending one copy on to Bob and measuring the other qubit. Unfortunately for Eve, this is impossible. To obtain any information, she has

to measure the qubit that Alice has sent, and this could change the qubit — it will end up as one of the basis vectors in the basis with which she chooses to measure. The best she can do is to choose one of the two bases at random, measure the qubit, and then send the qubit on to Bob. Let us see what happens now.

Alice and Bob are interested only in the measurements where they chose the same basis and we will restrict our attention to these times. When Alice and Bob agree on the basis, half the time Eve will also agree, and half the time she will choose the other basis. If all three agree on the basis, then they will all get the same bit as the measurement. If Eve chooses the wrong basis, then she will send a qubit that is in a superposition of Bob's basis states. When Bob measures this qubit he will get 0 and 1 with equal probability; he will get the right bit one half of the time.

We now go back to Alice and Bobb and their strings of bits of length, currently, $2n$. They know that if Eve is not interpreting the qubits, these two strings will be identical. In case Eve is intercepting qubits, she will be choosing the wrong basis half the time and in these cases, Bob will end with wrong bits half the time. In, summary, if Eve is intercepting qubits, a quarter of Bob's bits will disagree with Alice's bits. They now compare half of $2n$ bit over the open channel. If they agree on all the bits, they are sure that Eve is not listening and they can use the other n bits as the key. If they disagree on a quarter of the bits, they know that Eve is intercepting their qubits and they must find a new method to secure their communication.

This is a nice example of sending one qubit at a time. There are, however, very few things that we can do with qubits that don't interact with one another. In the next chapter, we look at what happens when we have two or more qubits. In particular, we look at yet another phenomenon that is not part of our classical worldview but that plays an essential part of the quantum world: **entanglement**.

Chapter 5

Entanglement

In this chapter, we introduce the concept of **Tensor Product** from Linear Algebra which will play a key role in understanding mathematics of **entanglement**. We will start by studying two systems that do not have any interaction between them. Since there is no interaction, we can study both systems independently without referring to one another. We will also see how to combine the two systems using the tensor product. Once we introduce the tensor product of two vector spaces, we will see that most of the vectors in this product will represent what are called entangled states.

Throughout this chapter, we will consider two qubits. Alice will have one, and Bob will have the other. We will start by studying the case where there is no interaction between Alice's and Bob's systems. You may initially find that the current description makes something very simple overly complicated, but once everything is described in terms of tensor product it will become clearer.

5.1 When Qubits Are Not Entangled

We assume that Alice is measuring using the orthonormal basis $(|a_0\rangle, |a_1\rangle)$ and Bob is measuring using the orthonormal basis $(|b_0\rangle, |b_1\rangle)$. A typical qubit for Alice is $|\nu\rangle = c_0 |a_0\rangle + c_1 |a_1\rangle$, and for Bob is $|w\rangle = d_0 |b_0\rangle + d_1 |b_1\rangle$. We now introduce the tensor product of these two state vectors, represented as $|\nu\rangle \otimes |w\rangle$.

$$|\nu\rangle \otimes |w\rangle = (c_0 |a_0\rangle + c_1 |a_1\rangle) \otimes (d_0 |b_0\rangle + d_1 |b_1\rangle)$$

We can expand this product as a normal product of form $(a + b)(c + d)$. We then get:

$$|\nu\rangle \otimes |w\rangle = c_0 d_0 |a_0\rangle \otimes |b_0\rangle + c_0 d_1 |a_0\rangle \otimes |b_1\rangle + c_1 d_0 |a_1\rangle \otimes |b_0\rangle + c_1 d_1 |a_1\rangle \otimes |b_1\rangle$$

If we further shorten $|\nu\rangle \otimes |w\rangle$ to $|\nu\rangle |w\rangle$, we will get:

$$|\nu\rangle |w\rangle = (c_0 |a_0\rangle + c_1 |a_1\rangle)(d_0 |b_0\rangle + d_1 |b_1\rangle)$$

$$\implies |\nu\rangle |w\rangle = c_0 d_0 |a_0\rangle |b_0\rangle + c_0 d_1 |a_0\rangle |b_1\rangle + c_1 d_0 |a_1\rangle |b_0\rangle + c_1 d_1 |a_1\rangle |b_1\rangle$$

Though this is a standard way to multiply two expressions, we should note one aspect carefully: **the first ket in the tensor product belongs to Alice, and the second ket belongs to Bob**. For example, the product $|\nu\rangle |w\rangle$ means that $|\nu\rangle$ belongs to Alice and $|w\rangle$ belongs to Bob. Similarly, the product $|w\rangle |\nu\rangle$ means that $|w\rangle$ belongs to Alice and $|\nu\rangle$ belongs to Bob. In general, the tensor product is not commutative, that is $|\nu\rangle |w\rangle \neq |w\rangle |\nu\rangle$.

We know Alice is measuring using her orthonormal basis $(|a_0\rangle, |a_1\rangle)$ and Bob is measuring using the orthonormal basis $(|b_0\rangle, |b_1\rangle)$ and we describe both Alice's and Bob's qubits using the tensor notation. This description involves the four tensor products that come from the basis vectors: $(|a_0\rangle |b_0\rangle, |a_0\rangle |b_1\rangle, |a_1\rangle |b_0\rangle, |a_1\rangle |b_1\rangle)$. We now note three things: these four products form an orthonormal basis for tensor product of Alice's and Bob's systems, each of these products is a unit vecotr, and tehya re orthogonal to each other.

The number $c_0 d_0$ is a probability amplitude. Its square, $(c_0 d_0)^2$ gives the probability that when both Alice and Bob measure their qubits, Alice's qubit jumps to $|a_0\rangle$, that is, she reads 0, and Bob's qubit jumps to $|b_0\rangle$, that is, he reads 1. We also know that the probability that Alice's qubit would jump to $|a_0\rangle$ is ${c_0}^2$ and the probability that Bob's qubit would jump to $|b_0\rangle$ is ${d_0}^2$. Thus, the probability that both would occur is ${c_0}^2 {d_0}^2$, which is same as $(c_0 d_0)^2$. On similar lines, note that the numbers ${c_0}^2 {d_1}^2, {c_1}^2 {d_0}^2, {c_1}^2 {d_1}^2$ gives the probability that Alice and Bob read $01, 10, 11$ respectively. Lastly, always remember that Alice's bit is listed before Bob's bit.

We further define $r = c_0 d_0, s = c_0 d_1, t = c_1 d_0, u = c_1 d_1$.

This will give $|\nu\rangle |w\rangle = r |a_0\rangle |b_0\rangle + s |a_0\rangle |b_1\rangle + t |a_1\rangle |b_0\rangle + u |a_1\rangle |b_1\rangle$.

We further know that $r^2 + s^2 + t^2 + u^2 = 1$ since these are probability amplitudes and, lastly we note that $ru = st$ since both of thse equal to $c_0 c_1 d_0 d_1$.

We now introduce a new concept. We will describe the state of Alice's and Bob's qubits by tensors of the form $r |a_0\rangle |b_0\rangle + s |a_0\rangle |b_1\rangle + t |a_1\rangle |b_0\rangle + u |a_1\rangle |b_1\rangle$. We willl further resctrict $r^2 + s^2 + t^2 + u^2 = 1$ so that the numbers r, s, t, u can be considered as probability amplitudes. We no longer insist that $ru = st$ and we allow all values of r, s, t, u as long as $r^2 + s^2 + t^2 + u^2 = 1$.

Given a tensor of the form $r\,|a_0\rangle\,|b_0\rangle + s\,|a_0\rangle\,|b_1\rangle + t\,|a_1\rangle\,|b_0\rangle + u\,|a_1\rangle\,|b_1\rangle$ with condition $r^2 + s^2 + t^2 + u^2 = 1$, we have two cases to consider. In the first case, if $ru = st$, we say that Alice's and Bob's qubits are **not entangled**. In the second case, if $ru \neq st$, we say that Alice's and Bob's qubits are **entangled**.

This rule is easy to remember if the terms are written out with the subscripts in the order we have presented them: $00, 01, 10, 11$. In this order, ru are the outer terms, and st are the inner, so the qubits are not entangled if the product of the outer terms is equal to the product of the inner ones, and they are entangled if the products are not equal. We will look at examples that illustrate both of these cases.

5.2 Example - Unentangled Qubits

Assume that Alice's and Bob's qubits are given by:

$$\frac{1}{2\sqrt{2}}\,|a_0\rangle\,|b_0\rangle + \frac{\sqrt{3}}{2\sqrt{2}}\,|a_0\rangle\,|b_1\rangle + \frac{1}{2\sqrt{2}}\,|a_1\rangle\,|b_0\rangle + \frac{\sqrt{3}}{2\sqrt{2}}\,|a_1\rangle\,|b_1\rangle$$

If we compute and compare the product of outer and inner probability amplitudes, we find that they are equal (both being $\frac{\sqrt{3}}{8}$) and we will say that qubits are unentangled. The probability amplitudes also tell us what happens when Alice and Bob both make measurements. They will get 00 with probability $\frac{1}{8}$, 01 with probability $\frac{3}{8}$, 10 with probability $\frac{1}{8}$ and 11 with probability $\frac{3}{8}$.

So far as we have seen what happens when both Alice and Bob make the measurements. We will further check what will happen if only one of them will make the measurement. We assume that Alice will make the measurement, and Bob does not. We now rewrite the tensor product from Alice's perspective:

$$|a_0\rangle\left(\frac{1}{2\sqrt{2}}\,|b_0\rangle + \frac{\sqrt{3}}{2\sqrt{2}}\,|b_1\rangle\right) + |a_1\rangle\left(\frac{1}{2\sqrt{2}}\,|b_0\rangle + \frac{\sqrt{3}}{2\sqrt{2}}\,|b_1\rangle\right)$$

We would prefer the expressions in the parenthesis to be unit vectors, so we divide by their lengths inside the parenthesis and multiply by their lengths outside. Thus, we rewrite it as:

$$\frac{1}{\sqrt{2}}\,|a_0\rangle\left(\frac{1}{2}\,|b_0\rangle + \frac{\sqrt{3}}{2}\,|b_1\rangle\right) + \frac{1}{\sqrt{2}}\,|a_1\rangle\left(\frac{1}{2}\,|b_0\rangle + \frac{\sqrt{3}}{2}\,|b_1\rangle\right)$$

We can rearrange the above expression by pulling the common factor in parenthesis, however, since it belongs to Bob, it must be kept on right. This yields:

$$\left(\frac{1}{\sqrt{2}}|a_0\rangle + \frac{1}{\sqrt{2}}|a_1\rangle\right)\left(\frac{1}{2}|b_0\rangle + \frac{\sqrt{3}}{2}|b_1\rangle\right)$$

Written in this way, it becomes clear that the states are not-entangled. We have a tensor product of Alice's qubit with that of Bob's. We can further deduce that if Alice measures first, she will get 0 and 1 with equal probability and this measurement has no impact on the state of Bob's qubit. Similarly, if Bob measures first, he will get 0 with probability $\frac{1}{4}$ and 1 with probability $\frac{3}{4}$ and this measurement has no effect on Alice's qubit.

To summarize this section, when the qubits are unentangled, a measurement of one of the qubits has no impact on the other qubit. In the next section, we will see that when qubits are entangled, the measurement of one will have an effect on the other one.

5.3 Entangled Qubits

Assume that Alice's and Bob's qubits are given by:

$$\frac{1}{2}|a_0\rangle|b_0\rangle + \frac{1}{2}|a_0\rangle|b_1\rangle + \frac{1}{\sqrt{2}}|a_1\rangle|b_0\rangle + 0\,|a_1\rangle|b_1\rangle$$

We start by calculating the products of inner and outer probabilities. The outer product of outer terms is 0 and the product of inner terms is $\frac{1}{2\sqrt{2}}$. Since these two products are not equal, we will say that the qubits are entangled.

As in the previous section, we use the probability amplitudes to see what will happen when both Alice and Bob measure their qubits. We can see that they will get 00 with probability $\frac{1}{4}$, 01 with probability $\frac{1}{4}$, 10 with probability $\frac{1}{2}$, and 11 with probability 0. So far, everything is the same as in the case of unentangled qubits.

So far as we have seen what happens when both Alice and Bob make the measurements. We will further check what will happen if only one of them will make the measurement. We assume that Alice will make the measurement, and Bob does not. We now rewrite the tensor product from Alice's perspective:

$$|a_0\rangle\left(\frac{1}{2}|b_0\rangle + \frac{1}{2}|b_1\rangle\right) + |a_1\rangle\left(\frac{1}{\sqrt{2}}|b_0\rangle + 0\,|b_1\rangle\right)$$

We would prefer the expressions in the parenthesis to be unit vectors, so we divide by their lengths inside the parenthesis and multiply by their lengths outside. Thus, we rewrite it as:

$$\frac{1}{\sqrt{2}}|a_0\rangle\left(\frac{1}{\sqrt{2}}|b_0\rangle+\frac{1}{\sqrt{2}}|b_1\rangle\right)+\frac{1}{\sqrt{2}}|a_1\rangle\left(1|b_0\rangle+0|b_1\rangle\right)$$

In the previous section we noticed that the terms in parenthesis were the same, and thus could be pulled out as a common factor. But, in this case, the terms in parenthesis aren't the same and this is what is meant to be entangled.

The probability amplitudes prefixing Alice's kets tell us that when she measures, she will 0 and 1 with equal probability. We further see that when she gets 0, her qubit will jump to $|a_0\rangle$, the entire system jumps to the unentangled state $|a_0\rangle\left(\frac{1}{\sqrt{2}}|b_0\rangle+\frac{1}{\sqrt{2}}|b_1\rangle\right)$, and Bob's qubit is no longer entangled with Alice's. When Alice gets 1, again Bob's qubit is no longer entangled with Alice's and becomes $|b_0\rangle$.

The result of Alice's measurement affects Bob's qubit. If she gets 0, Bob's qubit becomes $\left(\frac{1}{\sqrt{2}}|b_0\rangle+\frac{1}{\sqrt{2}}|b_1\rangle\right)$. If she gets 1, Bob's qubit becomes $|b_0\rangle$. This is strange, Alice and Bob can be far apart, and as soon as she makes a measurement, Bob's qubit becomes unentangled, but exactly what it is depends on Alice's outcome.

We now see what happens when Bob measures first. We start with the initial tensor product:

$$\frac{1}{2}|a_0\rangle|b_0\rangle+\frac{1}{2}|a_0\rangle|b_1\rangle+\frac{1}{\sqrt{2}}|a_1\rangle|b_0\rangle+0|a_1\rangle|b_1\rangle$$

We now rewrite the tensor product from Bob's perspective:

$$\left(\frac{1}{2}|a_0\rangle+\frac{1}{\sqrt{2}}|a_1\rangle\right)|b_0\rangle+\left(\frac{1}{2}|a_0\rangle+0|a_1\rangle\right)|b_1\rangle$$

We would prefer the expressions in the parenthesis to be unit vectors, so we divide by their lengths inside the parenthesis and multiply by their lengths outside. Thus, we rewrite it as:

$$\left(\frac{1}{\sqrt{3}}|a_0\rangle+\frac{\sqrt{2}}{\sqrt{3}}|a_1\rangle\right)\frac{\sqrt{3}}{2}|b_0\rangle+\left(1|a_0\rangle+0|a_1\rangle\right)\frac{1}{2}|b_1\rangle$$

When Bob measures his qubit, he gets 0 with probability $\frac{3}{4}$, and 1 with probability $\frac{1}{4}$. When Bob gets 0, Alice's qubut jumps to state $\left(\frac{1}{\sqrt{3}}|a_0\rangle + \frac{\sqrt{2}}{\sqrt{3}}|a_1\rangle\right)$. When Bob gets 1, Alice's qubut becomes $|a_0\rangle$.

To summarize, in the case of entangled quits, when the first person measures the qubit, the second person's qubit immediately jumps to one of the two states. These states depend on the result of the first person's measurement. In later sections, we will see how to exploit entangled bits, but first, we consider the case of superluminal communication.

5.4 Superluminal Communication

Superluminal communication is a hypothetical process in which information is sent at faster-than-light (FTL) speeds. The current scientific consensus is that faster-than-light communication is not possible, and to date, it has not been achieved in any experiment.

On the other hand, suppose Alice and Bob are on the opposite sides of the universe and have a number of entangled qubits. These are electrons whose spin states are entangled. Alice has one of each entangled pair of electrons with her, and Bob has the other one. Even though we describe in terms of entangled electrons, it should be clear that the actual electrons are totally separate and it's their spin states that are entangled.

When Alice makes a measurement on one of her electrons, the spin state of the corresponding electron in Bob's possession instantaneously jumps into one of two distinct states. Instantaneous is clearly faster than the speed of light! Can't entanglement be used for instantaneous communication?

Let's suppose that each pair of the entangled electrons is in the entangled spin state that we have seen in the previous section:

$$\frac{1}{2}|a_0\rangle|b_0\rangle + \frac{1}{2}|a_0\rangle|b_1\rangle + \frac{1}{\sqrt{2}}|a_1\rangle|b_0\rangle + 0\,|a_1\rangle|b_1\rangle$$

Suppose Alice measures the spins of her electrons before Bob measures the spin of their partners. We know that she gets a random string of 0s and 1s, with each symbol occurring with equal probability.

In the case where Bob measures his spins before Alice, and then Alice measures the spins, what will she observe? Once Alice has made the measurements, they

both will have made the measurements, so we can use the probability amplitudes of the initial tensor expression.

We know that they will get 00 with probability $\frac{1}{4}$, 01 with probability $\frac{1}{4}$, 10 with probability $\frac{1}{2}$, and 11 with probability 0. This means, Alice will get 0 with probability of $\frac{1}{4} + \frac{1}{4} = \frac{1}{2}$, and 1 with probability $\frac{1}{2} + 0 = \frac{1}{2}$. This means Alice will get a random sequence of 0s and 1s, with each symbol occurring with equal probability. But, this is exactly the same as the case when Alice measured first. So, Alice cannot tell from her measurements whether they were made before or after Bob's measurements. All entangled states behave this way. If there is no way of Alice and Bob being able to tell from their measurements who went first, there certainly can be no way of sending any information from one to another.

We have shown that Alice and Bob cannot send information when their qubits have a particular entangled state, but the argument generalizes to any entangled state. No matter what states Alice's and Bob's qubits have, it is impossible for them to send information by solely measuring their qubits.

We have now shown that superluminal communication is not possible. In the next section, we see how to write tensor products using the standard basis vectors.

5.5 The Standard Basis for Tensor Products

The standard basis for $\mathbb{R}^2$ is $\left(\begin{bmatrix}\mathbf{1}\\\mathbf{0}\end{bmatrix}, \begin{bmatrix}\mathbf{0}\\\mathbf{1}\end{bmatrix}\right)$. When both Alice and Bob use the standard basis, the tensor product is of the form:

$$r\begin{bmatrix}\mathbf{1}\\\mathbf{0}\end{bmatrix} \otimes \begin{bmatrix}\mathbf{1}\\\mathbf{0}\end{bmatrix} + s\begin{bmatrix}\mathbf{1}\\\mathbf{0}\end{bmatrix} \otimes \begin{bmatrix}\mathbf{0}\\\mathbf{1}\end{bmatrix} + t\begin{bmatrix}\mathbf{0}\\\mathbf{1}\end{bmatrix} \otimes \begin{bmatrix}\mathbf{1}\\\mathbf{0}\end{bmatrix} + u\begin{bmatrix}\mathbf{0}\\\mathbf{1}\end{bmatrix} \otimes \begin{bmatrix}\mathbf{0}\\\mathbf{1}\end{bmatrix}$$

Therefore, the standard ordered basis for $\mathbb{R}^2 \otimes \mathbb{R}^2$ is:

$$\left(\begin{bmatrix}\mathbf{1}\\\mathbf{0}\end{bmatrix} \otimes \begin{bmatrix}\mathbf{1}\\\mathbf{0}\end{bmatrix}, \begin{bmatrix}\mathbf{1}\\\mathbf{0}\end{bmatrix} \otimes \begin{bmatrix}\mathbf{0}\\\mathbf{1}\end{bmatrix}, \begin{bmatrix}\mathbf{0}\\\mathbf{1}\end{bmatrix} \otimes \begin{bmatrix}\mathbf{1}\\\mathbf{0}\end{bmatrix}, \begin{bmatrix}\mathbf{0}\\\mathbf{1}\end{bmatrix} \otimes \begin{bmatrix}\mathbf{0}\\\mathbf{1}\end{bmatrix}\right)$$

Since it has four vectors in the basis, it is a four-dimensional space. The standard four-dimensional space is $\mathbb{R}^4$ with ordered basis is given by $\left(\begin{bmatrix}\mathbf{1}\\\mathbf{0}\\\mathbf{0}\\\mathbf{0}\end{bmatrix}, \begin{bmatrix}\mathbf{0}\\\mathbf{1}\\\mathbf{0}\\\mathbf{0}\end{bmatrix}, \begin{bmatrix}\mathbf{0}\\\mathbf{0}\\\mathbf{1}\\\mathbf{0}\end{bmatrix}, \begin{bmatrix}\mathbf{0}\\\mathbf{0}\\\mathbf{0}\\\mathbf{1}\end{bmatrix}\right)$

We identify basis vectors in $\mathbb{R}^2 \otimes \mathbb{R}^2$ with those in $\mathbb{R}^4$, all the while making sure of ordering.

$$\begin{bmatrix}\mathbf{1}\\\mathbf{0}\\\mathbf{0}\\\mathbf{0}\end{bmatrix} = \begin{bmatrix}\mathbf{1}\\\mathbf{0}\end{bmatrix} \otimes \begin{bmatrix}\mathbf{1}\\\mathbf{0}\end{bmatrix}, \begin{bmatrix}\mathbf{0}\\\mathbf{1}\\\mathbf{0}\\\mathbf{0}\end{bmatrix} = \begin{bmatrix}\mathbf{1}\\\mathbf{0}\end{bmatrix} \otimes \begin{bmatrix}\mathbf{0}\\\mathbf{1}\end{bmatrix}, \begin{bmatrix}\mathbf{0}\\\mathbf{0}\\\mathbf{1}\\\mathbf{0}\end{bmatrix} = \begin{bmatrix}\mathbf{0}\\\mathbf{1}\end{bmatrix} \otimes \begin{bmatrix}\mathbf{1}\\\mathbf{0}\end{bmatrix}, \begin{bmatrix}\mathbf{0}\\\mathbf{0}\\\mathbf{0}\\\mathbf{1}\end{bmatrix} = \begin{bmatrix}\mathbf{0}\\\mathbf{1}\end{bmatrix} \otimes \begin{bmatrix}\mathbf{0}\\\mathbf{1}\end{bmatrix}$$

The easiest way to remember this is by the following construction.

$$\begin{bmatrix}\mathbf{a_0}\\\mathbf{a_1}\end{bmatrix} \otimes \begin{bmatrix}\mathbf{b_0}\\\mathbf{b_1}\end{bmatrix} = \begin{bmatrix}\mathbf{a_0}\begin{bmatrix}\mathbf{b_0}\\\mathbf{b_1}\end{bmatrix}\\\\\mathbf{a_1}\begin{bmatrix}\mathbf{b_0}\\\mathbf{b_1}\end{bmatrix}\end{bmatrix} = \begin{bmatrix}\mathbf{a_0b_0}\\\\\mathbf{a_0b_1}\\\\\mathbf{a_1b_0}\\\\\mathbf{a_1b_1}\end{bmatrix}$$

5.6 (Optional) How Do We Entangle Qubits?

We represent entangled qubits by either entangled electrons or photons. Though we often say that particles are entangled, in reality, we mean that the vector describing their states, a tensor in $\mathbb{R}^2 \otimes \mathbb{R}^2$, is entangled. The actual particles are separate, and as we saw earlier, can be very far apart. We see two methods of creating a pair of particles whose state vector is entangled - first, a physical experiment that can create entangled particles, and second, how quantum gates can be used to create entangled qubits.

One of the most widely used methods is called **Spontaneous Parametric Down Conversion**. A laser beam sends photons through a special crystal. Most of the photons just pass through but some are split into two. Energy and momentum must be conserved — the total energy and momentum of the two resulting photons must equal the energy and momentum of the initial photon. The conservation laws guarantee that the state describing the polarization of the two photons is entangled.

In the universe, electrons are often entangled. At the start of the book, we described Stern and Gerlach's experiment on silver atoms. Recall that the electron spins in the inner orbits canceled, leaving the lone electron in the outer orbit to give its spin to the atom. The innermost orbit has two electrons. These are entangled so that their spins cancel. We can think of the state vector describing the spin of these electrons as:

$$\frac{1}{\sqrt{2}}\begin{bmatrix}\mathbf{1}\\\mathbf{0}\end{bmatrix} \otimes \begin{bmatrix}\mathbf{0}\\\mathbf{1}\end{bmatrix} - \frac{1}{\sqrt{2}}\begin{bmatrix}\mathbf{0}\\\mathbf{1}\end{bmatrix} \otimes \begin{bmatrix}\mathbf{1}\\\mathbf{0}\end{bmatrix}$$

Entangled electrons also occur in superconductors, and these electrons have been used in experiments. However, often we want to have entangled particles that are far apart, as we will see when we discuss Bell Test.

The main problem with using entangled electrons that are near one another and then separating them is that they have a tendency to interact with the environment. It is difficult to separate them without this happening. On the other hand, entangled photons are much easier to separate, though more difficult to measure. It is possible, however, to get the best of both worlds. This has been done by an international team based at the Delft University of Technology in what they describe as a **A loophole-free Bell test**. They used two diamonds separated by 1.3 kilometers. Each diamond had slight imperfections—nitrogen atoms altered the carbon atom lattice structure in places. Electrons become trapped in the defects. A laser excited an electron in each of the diamonds in such a way that both electrons emitted photons. The emitted photons were entangled with the spins of the electrons that they were emitted from. The photons then traveled toward one another through a fiber optic cable and met in a beam splitter — a standard piece of equipment that is usually used to split a beam of photons in two, but here it is used to entangle the two photons. The photons were then measured. The result was that the two electrons were now entangled with one another.

In quantum computing, we will usually input unentangled qubits and entangle them using the **CNOT gate**. Later we will explain exactly what gates are, but the actual computations involve just matrix multiplication. We next look at this.

5.7 CNOT Gate and Entangled Qubits

We will have an actual definition and description of quantum gates in a later chapter, and in this section, we will see how they correspond to orthonormal bases or, equivalently, to orthogonal matrices.

The standard basis for four-dimensional space in $\mathbb{R}^4$ is given by:

$$\left(\begin{bmatrix} \mathbf{1} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} \mathbf{0} \\ \mathbf{1} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{1} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{1} \end{bmatrix} \right)$$

The CNOT gate comes from interchanging the order of the last two elements. This results in the matrix for the CNOT gate.

$$\left(\begin{bmatrix} \mathbf{1} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} \mathbf{0} \\ \mathbf{1} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{1} \end{bmatrix}, \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{1} \\ \mathbf{0} \end{bmatrix} \right)$$

This gate acts on pairs of qubits. To use the matrix, everything must be written using four-dimensional vectors. Let us look at an example.

We start by considering the unentangled tensor product given by

$$\frac{1}{\sqrt{2}} \begin{bmatrix} \mathbf{1} \\ \mathbf{1} \end{bmatrix} \otimes \begin{bmatrix} \mathbf{1} \\ \mathbf{0} \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} \mathbf{1} \\ \mathbf{0} \\ \mathbf{1} \\ \mathbf{0} \end{bmatrix}$$

When we send qubits through the gate, they are changed. The resulting qubits are obtained by multiplying by the matrix.

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ 0 \\ \frac{1}{\sqrt{2}} \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

The final vector represents a pair of entangled qubits. This can be verified since the product of outer amplitudes does not match the product of inner amplitudes. This can be further written as:

$$\frac{1}{\sqrt{2}} \begin{bmatrix} \mathbf{1} \\ \mathbf{0} \end{bmatrix} \otimes \begin{bmatrix} \mathbf{1} \\ \mathbf{0} \end{bmatrix} + \frac{1}{\sqrt{2}} \begin{bmatrix} \mathbf{0} \\ \mathbf{1} \end{bmatrix} \otimes \begin{bmatrix} \mathbf{0} \\ \mathbf{1} \end{bmatrix}$$

We will often use entangled qubits in this state. It has the very nice property that if Alice and Bob measure in the standard basis, they will both get $\begin{bmatrix} \mathbf{1} \\ \mathbf{0} \end{bmatrix}$, corresponding to 0, or they will both get $\begin{bmatrix} \mathbf{0} \\ \mathbf{1} \end{bmatrix}$, corresponding to 1.

Entangled Quantum Clocks Remember the Quantum Clocks that were introduced earlier? These are the clocks to which we can only ask if the hands are pointing in a certain direction, and the clock with either answer "yes", or tell that it's pointing in opposite direction. Refer to Chapter-2 for a recap.

We start by letting vector $\begin{bmatrix} \mathbf{1} \\ \mathbf{0} \end{bmatrix}$ correspond to pointing to twelve and $\begin{bmatrix} \mathbf{0} \\ \mathbf{1} \end{bmatrix}$ corresponding to point to six. Also, consider a pair of locks in the entangled state:

$$\frac{1}{\sqrt{2}}\begin{bmatrix}\mathbf{1}\\\mathbf{0}\end{bmatrix}\otimes\begin{bmatrix}\mathbf{1}\\\mathbf{0}\end{bmatrix}+\frac{1}{\sqrt{2}}\begin{bmatrix}\mathbf{0}\\\mathbf{1}\end{bmatrix}\otimes\begin{bmatrix}\mathbf{0}\\\mathbf{1}\end{bmatrix}$$

We further consider one hundred such pairs of clocks, each pair of which in this state. We divide these hundred pairs such that you have one hundred of these clocks, and I have their other hundred partners. We both will be asking the same question repeatedly: is the hand pointing to twelve? We have two scenarios to consider, let's understand them one by one.

In the first case, we don't contact each other. We go through all the clocks one by one and ask the question. Each time the clock will answer yes or no. We write 1 if it's yes, and 0 if it's a no. After we have run through all the hundred clocks, we have a string of 0s and 1s. We analyze our strings and see a random sequence of 0s and 1s, with each occurring about the same number of times. We now contact each other and compare the strings. We know that both the strings will match, in all the hundred places.

In the second case, we again start with the same setup of hundred pairs of clocks. This time, we have an agreement that you will measure first. You ask your questions on the hour, and I will ask my question half hour later. During these half-hour intervals, you will call and inform me what my clock's answer will be. At the end of the experiment, after we have gone through all hundred pairs of clocks, we both have a string of 0s and 1s agreeing in all positions. Every time you called and told me what my result was going to be, you were exactly right. Does this mean that your measurement affected mine?

Actually, I was cheating and I didn't follow the rules. I was actually asking the questions to the clock half an hour before you asked yours. I knew your answer before you did, and your calls just confirmed what I already knew. There is no way from the data that you can tell whether or not I was following the rules or I was cheating. There is no way you can tell whether I am asking my questions before or after you asked yours.

There is no causation here, just correlation. We learned in the earlier section that we cannot use these clocks to send messages between us, but the observations are still mired in mystery. Albert Einstein called entanglement as "spooky action at distance".

Suppose that you and I have a pair of entangled quantum clocks, and we are talking on the phone to one another. Neither of us has asked our clock a question, so they are still entangled. In this state, if you were to ask your clock the question, you would have an equal chance of getting an answer that the hand was pointing to twelve or six. But as soon as I ask my clock a question, you no longer have an equal chance of getting one of the two answers. You will get exactly the same answer as mine.

This correlation would not be spooky if when our clocks were entangled it was decided, but unknown to us, whether both our hands were pointing at either twelve or six. We had to wait until one of us asked the question, and as soon as one of us knows the answer so does the other.

But this is not what our model describes. Our model says that the decision on which direction our hands are pointing is not made beforehand. It's made only when the first of us ask our question. This is what makes it spooky.

In the next chapter, we will look at this in detail. We will look at a model that incorporates correlation in an intuitive and non-spooky way. Unfortunately, it is wrong. John Stewart Bell came up with an ingenious test that shows that the simple explanation is not correct and that the mysterious spookiness has to remain.

Chapter 6

Bell's Inequality

The model that we have seen so far describing qubits and spin of electrons or polarization of photons is often called the Copenhagen Interpretation, named after the city where Niels Bohr was living and working. Some of the other leading physicists of the time, including Albert Einstein and Erwin Schrödinger, objected to the model's interpretation of states jumping to basis states based on probabilities, and to the concept of *action at a distance*. They hoped for a better model using "hidden variables" and "local realism". Their objection wasn't to using Copenhagen Interpretation for purpose of calculations, but they expected a deeper theory that would explain why the Copenhagen Interpretation was producing correct answers.

Bohr and Einstein had a series of debates about the meaning and philosophy of quantum mechanics. In this chapter, we will look at their viewpoints. At present, we know that Einstein and Schrödinger's view was wrong, and Copenhagen Interpretation is considered the standard model to understand quantum computing. However, both Einstein and Schrödinger were brilliant physicists, and we should study their arguments for several reasons.

One of the main areas where Bohr and Einstein debated was around local realism. In simple terms, local realism means that a particle can only be influenced by something changing in its near environment. This seems intuitive, but quantum mechanics tells us that reality is different. Einstein's model seems to be natural and correct, and when quantum entanglement is introduced firstly, it's natural to assume a model similar to that of Einstein.

John Stewart Bell, an Irish physicist, devised an innovative test that could distinguish between the two models. All the mathematics that we have seen so far in previous chapters is good enough to understand Bell's result. Bell's

test has been conducted several times, and the results have always confirmed Copenhagen Interpretation. Even though it is difficult to eliminate all the biases in the setup of the experiments, over the years more and more loopholes have been excluded. We will look into this in the current chapter.

We will look at the end of the chapter how the result of Bell's inequality can be used for secure exchange of messages between Alice and Bob. The entangled qubits that were used by Bell will reappear when we discuss quantum algorithms in later chapters. We start this chapter by looking at entangled qubits that were introduced in the last chapter, and we explore what happens when we measure them using different bases. We start the analysis using the Copenhagen Interpretation.

6.1 Entangled Qubits and Different Basis

We start by looking at two entangled clocks in the state

$$\frac{1}{\sqrt{2}}\begin{bmatrix}\mathbf{1}\\\mathbf{0}\end{bmatrix}\otimes\begin{bmatrix}\mathbf{1}\\\mathbf{0}\end{bmatrix}+\frac{1}{\sqrt{2}}\begin{bmatrix}\mathbf{0}\\\mathbf{1}\end{bmatrix}\otimes\begin{bmatrix}\mathbf{0}\\\mathbf{1}\end{bmatrix}$$

We learned in the previous chapter that if Alice and Bob both had one of these clocks, and both asked whether the hand was pointing towards twelve, both would either get the answer that it was or that it was pointing in direction of six. Both of them will always get the same answer. What would happen if Alice and Bob change the direction of measurement? As an example, what would be the answer if they both ask whether the hands are pointing in direction of four? We know that clocks will answer that the hands are either pointing toward four, or towards ten. We have two questions to answer - will Alice and Bob get exactly the same answer? Are both the answers equally likely?

We start with the intuitive argument about the qubits in the entangled state

$$\frac{1}{\sqrt{2}}\begin{bmatrix}\mathbf{1}\\\mathbf{0}\end{bmatrix}\otimes\begin{bmatrix}\mathbf{1}\\\mathbf{0}\end{bmatrix}+\frac{1}{\sqrt{2}}\begin{bmatrix}\mathbf{0}\\\mathbf{1}\end{bmatrix}\otimes\begin{bmatrix}\mathbf{0}\\\mathbf{1}\end{bmatrix}$$

Suppose that both Alice and Bob measure the spin of electrons in the 0° direction. If Alice gets **N**, Bob gets **S**. If Alice gets **S**, Bob gets **N**. These entangled states may be representing two electrons in an orbit where the spins cancel. Intuitively, we would expect the spins to cancel in every direction, so we assume that if Alice and Bob chose a new basis direction of measurement, they would get spins in the opposite direction. Furthermore, from symmetry, we can further assume that both directions will be equally likely.

These intuitive arguments lead us to conjecture that if we have entangled qubits in the state

$$\frac{1}{\sqrt{2}}\begin{bmatrix}\mathbf{1}\\\mathbf{0}\end{bmatrix}\otimes\begin{bmatrix}\mathbf{1}\\\mathbf{0}\end{bmatrix}+\frac{1}{\sqrt{2}}\begin{bmatrix}\mathbf{0}\\\mathbf{1}\end{bmatrix}\otimes\begin{bmatrix}\mathbf{0}\\\mathbf{1}\end{bmatrix}$$

and then rewrite this state using a new orthonormal basis $(|b_0\rangle, |b_1\rangle)$, we should get

$$\frac{1}{\sqrt{2}}|b_0\rangle\otimes|b_1\rangle+\frac{1}{\sqrt{2}}|b_1\rangle\otimes|b_1\rangle$$

We have observed on several occasions that intuitive arguments are wrong when it comes to the quantum realm. However, in this case, our intuitive argument is correct and we will next prove it.

To Prove:

$$\frac{1}{\sqrt{2}}\begin{bmatrix}\mathbf{1}\\\mathbf{0}\end{bmatrix}\otimes\begin{bmatrix}\mathbf{1}\\\mathbf{0}\end{bmatrix}+\frac{1}{\sqrt{2}}\begin{bmatrix}\mathbf{0}\\\mathbf{1}\end{bmatrix}\otimes\begin{bmatrix}\mathbf{0}\\\mathbf{1}\end{bmatrix}=\frac{1}{\sqrt{2}}|b_0\rangle\otimes|b_1\rangle+\frac{1}{\sqrt{2}}|b_1\rangle\otimes|b_1\rangle$$

We start by expressing kets $|b_0\rangle$ and $|b_1\rangle$ as column vectors. Let $|\mathbf{b_0}\rangle = \begin{bmatrix}\mathbf{a}\\\mathbf{b}\end{bmatrix}$ and $|\mathbf{b_1}\rangle = \begin{bmatrix}\mathbf{c}\\\mathbf{d}\end{bmatrix}$. Now we express our standard basis vectors in terms of the new basis (refer to Chapter-2 for the steps). We start with $\begin{bmatrix}\mathbf{1}\\\mathbf{0}\end{bmatrix}$.

The equation

$$\begin{bmatrix}\mathbf{a} & \mathbf{b}\\\mathbf{c} & \mathbf{d}\end{bmatrix}\begin{bmatrix}\mathbf{1}\\\mathbf{0}\end{bmatrix}=\begin{bmatrix}\mathbf{a}\\\mathbf{c}\end{bmatrix}$$

tells us that

$$\begin{bmatrix}\mathbf{1}\\\mathbf{0}\end{bmatrix}=\mathbf{a}\begin{bmatrix}\mathbf{a}\\\mathbf{b}\end{bmatrix}+\mathbf{c}\begin{bmatrix}\mathbf{c}\\\mathbf{d}\end{bmatrix}$$

Thus,

$$\begin{bmatrix}\mathbf{1}\\\mathbf{0}\end{bmatrix}\otimes\begin{bmatrix}\mathbf{1}\\\mathbf{0}\end{bmatrix}=\left(\mathbf{a}\begin{bmatrix}\mathbf{a}\\\mathbf{b}\end{bmatrix}+\mathbf{c}\begin{bmatrix}\mathbf{c}\\\mathbf{d}\end{bmatrix}\right)\otimes\begin{bmatrix}\mathbf{1}\\\mathbf{0}\end{bmatrix}$$

$$\implies \begin{bmatrix}\mathbf{1}\\\mathbf{0}\end{bmatrix}\otimes\begin{bmatrix}\mathbf{1}\\\mathbf{0}\end{bmatrix}=\mathbf{a}\begin{bmatrix}\mathbf{a}\\\mathbf{b}\end{bmatrix}\otimes\begin{bmatrix}\mathbf{1}\\\mathbf{0}\end{bmatrix}+\mathbf{c}\begin{bmatrix}\mathbf{c}\\\mathbf{d}\end{bmatrix}\otimes\begin{bmatrix}\mathbf{1}\\\mathbf{0}\end{bmatrix}$$

$$\implies \begin{bmatrix}\mathbf{1}\\\mathbf{0}\end{bmatrix}\otimes\begin{bmatrix}\mathbf{1}\\\mathbf{0}\end{bmatrix}=\begin{bmatrix}\mathbf{a}\\\mathbf{b}\end{bmatrix}\otimes\begin{bmatrix}\mathbf{a}\\\mathbf{0}\end{bmatrix}+\begin{bmatrix}\mathbf{c}\\\mathbf{d}\end{bmatrix}\otimes\begin{bmatrix}\mathbf{c}\\\mathbf{0}\end{bmatrix}$$

A similar line of argument will show that

$$\begin{bmatrix}\mathbf{0}\\\mathbf{1}\end{bmatrix}\otimes\begin{bmatrix}\mathbf{0}\\\mathbf{1}\end{bmatrix}=\begin{bmatrix}\mathbf{a}\\\mathbf{b}\end{bmatrix}\otimes\begin{bmatrix}\mathbf{0}\\\mathbf{b}\end{bmatrix}+\begin{bmatrix}\mathbf{c}\\\mathbf{d}\end{bmatrix}\otimes\begin{bmatrix}\mathbf{0}\\\mathbf{d}\end{bmatrix}$$

Putting it all together, we get

$$\begin{bmatrix}\mathbf{1}\\\mathbf{0}\end{bmatrix}\otimes\begin{bmatrix}\mathbf{1}\\\mathbf{0}\end{bmatrix}+\begin{bmatrix}\mathbf{0}\\\mathbf{1}\end{bmatrix}\otimes\begin{bmatrix}\mathbf{0}\\\mathbf{1}\end{bmatrix}=\left(\begin{bmatrix}\mathbf{a}\\\mathbf{b}\end{bmatrix}\otimes\begin{bmatrix}\mathbf{a}\\\mathbf{0}\end{bmatrix}+\begin{bmatrix}\mathbf{c}\\\mathbf{d}\end{bmatrix}\otimes\begin{bmatrix}\mathbf{c}\\\mathbf{0}\end{bmatrix}\right)+\left(\begin{bmatrix}\mathbf{a}\\\mathbf{b}\end{bmatrix}\otimes\begin{bmatrix}\mathbf{0}\\\mathbf{b}\end{bmatrix}+\begin{bmatrix}\mathbf{c}\\\mathbf{d}\end{bmatrix}\otimes\begin{bmatrix}\mathbf{0}\\\mathbf{d}\end{bmatrix}\right)$$

$$\Longrightarrow$$

$$\begin{bmatrix}\mathbf{1}\\\mathbf{0}\end{bmatrix}\otimes\begin{bmatrix}\mathbf{1}\\\mathbf{0}\end{bmatrix}+\begin{bmatrix}\mathbf{0}\\\mathbf{1}\end{bmatrix}\otimes\begin{bmatrix}\mathbf{0}\\\mathbf{1}\end{bmatrix}=\begin{bmatrix}\mathbf{a}\\\mathbf{b}\end{bmatrix}\otimes\left(\begin{bmatrix}\mathbf{a}\\\mathbf{0}\end{bmatrix}+\begin{bmatrix}\mathbf{0}\\\mathbf{b}\end{bmatrix}\right)+\begin{bmatrix}\mathbf{c}\\\mathbf{d}\end{bmatrix}\otimes\left(\begin{bmatrix}\mathbf{c}\\\mathbf{0}\end{bmatrix}+\begin{bmatrix}\mathbf{0}\\\mathbf{d}\end{bmatrix}\right)$$

$$\Longrightarrow$$

$$\begin{bmatrix}\mathbf{1}\\\mathbf{0}\end{bmatrix}\otimes\begin{bmatrix}\mathbf{1}\\\mathbf{0}\end{bmatrix}+\begin{bmatrix}\mathbf{0}\\\mathbf{1}\end{bmatrix}\otimes\begin{bmatrix}\mathbf{0}\\\mathbf{1}\end{bmatrix}=\begin{bmatrix}\mathbf{a}\\\mathbf{b}\end{bmatrix}\otimes\begin{bmatrix}\mathbf{a}\\\mathbf{b}\end{bmatrix}+\begin{bmatrix}\mathbf{c}\\\mathbf{d}\end{bmatrix}\otimes\begin{bmatrix}\mathbf{c}\\\mathbf{d}\end{bmatrix}$$

$$\Longrightarrow$$

$$\begin{bmatrix}\mathbf{1}\\\mathbf{0}\end{bmatrix}\otimes\begin{bmatrix}\mathbf{1}\\\mathbf{0}\end{bmatrix}+\begin{bmatrix}\mathbf{0}\\\mathbf{1}\end{bmatrix}\otimes\begin{bmatrix}\mathbf{0}\\\mathbf{1}\end{bmatrix}=|\mathbf{b_0}\rangle\otimes|\mathbf{b_0}\rangle+|\mathbf{b_1}\rangle\otimes|\mathbf{b_1}\rangle$$

$$\Longrightarrow$$

$$\frac{1}{\sqrt{2}}\begin{bmatrix}\mathbf{1}\\\mathbf{0}\end{bmatrix}\otimes\begin{bmatrix}\mathbf{1}\\\mathbf{0}\end{bmatrix}+\frac{1}{\sqrt{2}}\begin{bmatrix}\mathbf{0}\\\mathbf{1}\end{bmatrix}\otimes\begin{bmatrix}\mathbf{0}\\\mathbf{1}\end{bmatrix}=\frac{1}{\sqrt{2}}|b_0\rangle\otimes|b_1\rangle+\frac{1}{\sqrt{2}}|b_1\rangle\otimes|b_1\rangle$$

Q.E.D.

This tells us that if Alice and Bob's qubits are entangled with the state

$$\frac{1}{\sqrt{2}}\begin{bmatrix}\mathbf{1}\\\mathbf{0}\end{bmatrix}\otimes\begin{bmatrix}\mathbf{1}\\\mathbf{0}\end{bmatrix}+\frac{1}{\sqrt{2}}\begin{bmatrix}\mathbf{0}\\\mathbf{1}\end{bmatrix}\otimes\begin{bmatrix}\mathbf{0}\\\mathbf{1}\end{bmatrix},$$

and if they both chose to measure their qubits in an orthonormal basis ($|\mathbf{b_0}\rangle, |\mathbf{b_1}\rangle$), the entangled state can be written as $\frac{1}{\sqrt{2}}|\mathbf{b_0}\rangle\otimes|\mathbf{b_0}\rangle+\frac{1}{\sqrt{2}}|\mathbf{b_1}\rangle\otimes|\mathbf{b_1}\rangle$. When the first measurement is made, the state jumps to either of the unentangled states $|\mathbf{b_0}\rangle|\mathbf{b_0}\rangle$ or $|\mathbf{b_1}\rangle|\mathbf{b_1}\rangle$ with equal probability. It further means that when Alice and Bob have measured their qubits, they will both get 0 or 1 with equal probability.

To understand Bell's result, we will measure the entangled qubits using three different basis directions. These are the basis corresponding to rotating our measurement device by 0°, 120°, and 240°. If interpreting in terms of quantum clocks, we will ask one of the three questions - whether the hand is pointing

towards twelve, four or eight. If we denote these basis by $(|\uparrow\rangle, |\downarrow\rangle)$, $(|\searrow\rangle, |\nwarrow\rangle)$, and $(|\swarrow\rangle, |\nearrow\rangle)$, then the following three are descriptions of same entangled state:

$$\frac{1}{\sqrt{2}}|\uparrow\rangle|\uparrow\rangle + \frac{1}{\sqrt{2}}|\downarrow\rangle|\downarrow\rangle, \frac{1}{\sqrt{2}}|\searrow\rangle|\searrow\rangle + \frac{1}{\sqrt{2}}|\nwarrow\rangle|\nwarrow\rangle, \frac{1}{\sqrt{2}}|\swarrow\rangle|\swarrow\rangle + \frac{1}{\sqrt{2}}|\nearrow\rangle|\nearrow\rangle$$

We now look at Einstein and see his views on these entangled states.

6.2 Local Realism

We take the example of Gravity to understand the concept of local realism. Newton's law of Universal Gravitation gave the formula to calculate the strength of the force between two masses. Once we put in the two masses, the distance separating them, and the Gravitational constant, the formula will give the magnitude of the attractive force between them. The law can be further used to prove that planets orbit around a star in elliptical orbits. Further, though the law tells us the value of the force, it doesn't give any information about the mechanism that connects the planet and the star. The law of gravitation is very useful for calculations, but it doesn't explain how gravity works. Everyone had a belief that there must a deeper theory that can explain how gravity works. There were many proposals, on which though there was no consensus, everyone had a belief that **gravity is not spooky action at a distance**, and that an explanation would be found one day. There was a belief in something called local realism.

Einstein gave the General Theory Relativity that not only improved the accuracy of calculations over Newton's laws, but it also explained how gravity worked. It described working of working in terms of space-time. All objects, including planets, move according to the shape of space-time where its located. it eliminated the need for "spooky action at a distance". Einstein's theory explained how gravity worked, and this description was local. The planets move according to the shape of space in their neighborhood. This is what is called **local realism**.

We have seen that when we measure a pair of entangled qubits, the state changes immediately, even when qubits are far apart. This Copenhagen Interpretation introduced the idea of spooky action at a distance. Einstein had just eliminated spooky action from the theory of Gravity, and now he was staring at it once again. Bohr had a belief that no deeper theory will explain the mechanism behind this action, Einstein disagreed and believed that he can prove Bohr wrong.

Einstein, along with Boris Podolsky and Nathan Rosen, wrote a paper describing how the Special Theory of Relativity implied that no information can travel

faster than the speed of light, but the Copenhagen Interpretation implied information can be sent from Alice to Bob instantaneously. This problem came to be known as the **EPR Paradox**.

The EPR Paradox was originally presented in terms of the position and momentum of two entangled particles. David Bohm reformulated this in terms of spin, and this was further used by Bell to calculate his important inequality. In the previous chapter, we saw that Copenhagen Interpretation doesn't allow information to be exchanged at speeds faster than the speed of light, and so even though EPR Paradox isn't really a paradox, there is still the question of explaining spooky action at a distance.

6.3 Hidden Variables

The classical view of physics is deterministic. This means, that once all the initial conditions are known to infinite precision, the outcome can also be predicted with certainty. In reality, the initial conditions can only be measured to finite precision, meaning that there will always be a difference between the measured value and the true value. Over a long period, this error will grow and we will not be able to make any sensible prediction. This phenomenon is called "sensitive dependence on initial conditions". This is one of the reasons why predicting the weather for more than a fortnight is highly unreliable. We should remember that the underlying theory is deterministic. The weather prediction is unpredictable, not because of any inherent randomness, but due to imperfections in accurate measurements.

The laws of thermodynamics is another area where probability and classical physics cross path once again. The underlying theory is deterministic in the case of gas laws too. If we know the exact mass and velocity of all molecules in the gas volume, in theory, we can predict, with absolute accuracy, what happens to each molecule in the future. In reality, the number of molecules is too large to be considered individually, and we choose to look at gas from a statistical point of view.

Einstein had a feeling that the use of probability in quantum physics means that the theory is incomplete. He longed for a deeper theory, involving new variables, which is deterministic and looks probabilistic when we don't consider all these, yet, unknown variables. These, yet, unknown variables came to be known as **hidden variables**.

6.4 Explaining Entanglement - Classical Way

Consider the quantum clocks in the entangled state given by $\frac{1}{\sqrt{2}}|\uparrow\rangle|\uparrow\rangle+\frac{1}{\sqrt{2}}|\downarrow\rangle|\downarrow\rangle$. Alice and Bob both ask if the clock hand is pointing towards the direction of twelve. The quantum models tell us that both will get the same answer, with equal probability. When the actual experiment is performed using the spin of entangled electrons, the results align perfectly with what the quantum model predicts. How can we explain this with the classical model?

The classical interpretation goes like this. Electrons have a definite spin in any direction and they become entangled through some local interaction. We do not know how, or what, but there is some local process that puts the electrons in exactly the same spin configuration state. When they are entangled, a spin direction is chosen for both the electrons.

To consider further, let us take a deck of cards that we can shuffle. We now take a card out, without looking at it, cut it into two parts, and put the two parts in two envelopes. We note that we do not have any knowledge of which card has been selected. We then send the cards to Bob and Alice who lives at opposite ends of the universe. Alice and Bob do not have any information about what card they have, and it can be any of the fifty-two cards. However, as soon as Alice opens her envelope and sees a Queen of Hearts, she knows that Bob's card is also the same. We can see that there is no spooky action at a distance here, all seems logical.

As mentioned earlier, for Bell's result we will measure the entangled qubits in three different directions. Going back to our clock example, we will be asking one of the three questions - whether the hand is pointing to twelve, to four, or to eight. The Copenhagen Interpretation model predicts that for each question the answer will be either that the hand is pointing in the direction asked or that it is pointing in the opposite direction. For each question, both the answers are equally likely. When Alice and Bob ask exactly the same question, they will both get exactly the same answer. How do we explain this from classical theory's point of view?

We start by stating that there is some local process that entangles the clocks. We do not describe how it is done, and we rely on hidden variables or a deeper theory that explains it. Once the clocks are entangled, definite answers to the three questions are chosen. This is similar to us having three decks of cards, each with a different colored overleaf. We take a card each from red, green, and blue decks. We then cut these cards in half and mail one set of halves to Alice and the other set of halves to Bob. Alice and Bob can be located at opposite ends of the universe. If Alice sees her red card as Queen of Hearts, she knows that Bob's green card is also a Queen of Hearts.

In the case of entangled clocks, the classical theory says that there is a definite answer to each question, and that answer is already determined before its asked. The Quantum theory, on the other hand, says that the answer to these questions is not determined until we ask it.

6.5 Bell's Test

Assume we generate a stream of entangled qubits in state $\frac{1}{\sqrt{2}}|\uparrow\rangle|\uparrow\rangle + \frac{1}{\sqrt{2}}|\downarrow\rangle|\downarrow\rangle$ and send to Alice and Bob. Let us see how they measure the qubits now.

Alice randomly chooses to measure her qubits in the direction of 0°, 120°, and 240° each with probability $\frac{1}{3}$. Alice doesn't record the direction she has chosen, but she records whether she has measured 0 or 1. We remember once again that 0 corresponds to the first basis vector and 1 corresponds to the second basis vector. Once Alice is done with the measurements, Bob starts to measure. He also randomly selects one of the same directions, each with probability $\frac{1}{3}$ and measures his qubit. Bob also doesn't record the direction in which he has done the measurement, but he does record whether he has observed 0 or 1.

At the end of the measurements, both Alice and Bob would have written a long string of 0s and 1s. Now, they start to compare these symbols, one at a time. They each take the first symbol, and if they agree, they write **A** and **D** if they disagree. They then do the same process for all the bits.

Once a new string consisting of **A**s and **D**s is generated, Bell wondered what proportion of the string is made up of **A**s? Bell realized that the quantum mechanics model and the classical model gave different numbers for the answer.

6.6 Answer to Bell's question from Quantum Mechanics Point of View

We start by taking qubits in the entangled state $\frac{1}{\sqrt{2}}|\uparrow\rangle|\uparrow\rangle + \frac{1}{\sqrt{2}}|\downarrow\rangle|\downarrow\rangle$. We have seen earlier that if both Alice and Bob measure in same direction they both will observe the same answer. We now see what is observed when they measure in different directions.

We further assume that Alice chooses $(|\searrow\rangle, |\nwarrow\rangle)$, and Bob chooses $(|\swarrow\rangle, |\nearrow\rangle)$. The entangled state $\frac{1}{\sqrt{2}}|\uparrow\rangle|\uparrow\rangle + \frac{1}{\sqrt{2}}|\downarrow\rangle|\downarrow\rangle$ can be written in Alice's basis as

$\frac{1}{\sqrt{2}}|\searrow\rangle|\searrow\rangle + \frac{1}{\sqrt{2}}|\nwarrow\rangle|\nwarrow\rangle$. Upon measurement by Alice, the state jumps to either $|\searrow\rangle\,|\searrow\rangle$ or $|\nwarrow\rangle\,|\nwarrow\rangle$, both being equally likely. If the state jumps to $|\searrow\rangle\,|\searrow\rangle$, she records a 0, and if the state jumps to $|\nwarrow\rangle\,|\nwarrow\rangle$, she records a 1.

Now it's Bob's turn to make the measurement. We assume that after Alice's masurement the qubits are in state $|\searrow\rangle\,|\searrow\rangle$, meaning Bob's qubit is in state $|\searrow\rangle$. To see the result of Bob's measurement, we should express his state in terms of his basis vectors yielding $|\searrow\rangle = \frac{1}{2}\,|\swarrow\rangle + \frac{\sqrt{3}}{2}\,|\nearrow\rangle$. This tells us that when Bob makes the measurement, he will get a 0 with probability $\frac{1}{4}$ and 1 with probability $\frac{3}{4}$. On similar lines, if Alice gets a 1, Bob's probability of getting a 1 is also $\frac{1}{4}$.

The other cases are all similar: If Bob and Alice measure in different directions, they will agree $\frac{1}{4}$ of the time and disagree $\frac{3}{4}$ of the time.

To tie everything together, one-third of the time they measure in the same direction and agree each time; two-thirds of the time they measure in different directions and agree on just one-quarter of these measurements. This means, that the proportion of **A**'s in the string of **A**s and **D**s is $\frac{1}{3} \times 1 + \frac{2}{3} \times \frac{1}{4} = \frac{1}{2}$.

This means that from the quantum mechanical point of view, in the long run, the proportion of **A**s should be $\frac{1}{2}$.

We now look at the classical model.

6.7 Answer to Bell's question from a Classical Point of View

As we have seen earlier, the classical view stipulates that the measurements in all directions are known from the start. There are three directions of measurements and each measurement can give either a 0 or a 1. This generates eight possible configurations: 000, 001, 010, 011, 100, 101, 110, 111 where the leftmost digit gives us the answer if we were to measure in the basis $(|\uparrow\rangle, |\downarrow\rangle)$, the middle digit gives us the answer if we were to measure in the basis $(|\searrow\rangle, |\nwarrow\rangle)$, and the right most digit gives us the answer if we were to measure in the basis $(|\swarrow\rangle, |\nearrow\rangle)$.

Entanglement means that the configuration for Alice's and Bob's qubits are identical. This means, that if Alice has a qubit in configuration 001, the same

holds true for Bob. What happens when Alice and Bob choose a direction to measure. In the example scenario of qubit configuration in the state 001m if Alice measures using basis $(|\uparrow\rangle, |\downarrow\rangle)$ and Bob measures using $(|\swarrow\rangle, |\nearrow\rangle)$, then Alice would have observed 0 and Bob would have measured 1 resulting in an disagreement.

For our sample configurtation space, the below table gives all the possibvle outcomes. The left column gives the configurations, and the top row gives the possibilities for Alice and Bob's measurement bases. We further use letters a, b, c to represent the bases. We let a represent $(|\uparrow\rangle, |\downarrow\rangle)$, b represent $(|\searrow\rangle, |\nwarrow\rangle)$, and c represent $(|\swarrow\rangle, |\nearrow\rangle)$. We will also list Alice's bases first and then Bob's basis. For example, (b, c) means Aliice is choosing $(|\searrow\rangle, |\nwarrow\rangle)$, and Bob is choosing $(|\swarrow\rangle, |\nearrow\rangle)$ for measurement. The **A**s and **D**s in the table represent if their measurements agree of disagree.

	Measurement directions								
Config.	(a,a)	(a,b)	(a,c)	(b,a)	(b,b)	(b,c)	(c,a)	(c,b)	(c,c)
000	A	A	A	A	A	A	A	A	A
001	A	A	D	A	A	D	D	D	A
010	A	D	A	D	A	D	A	D	A
011	A	D	D	D	A	A	D	A	A
100	A	D	D	D	A	A	D	A	A
101	A	D	A	D	A	D	A	D	A
110	A	A	D	A	A	D	D	D	A
111	A	A	A	A	A	A	A	A	A

Figure 6.1: Possible State Configurations

We may not know the exact probabilities for each of the configurations. There are eight possible configurations and we might assume that each of them is equally likely, or they may not be. We will not make any assumptions about this. However, we can easily allocate probabilities for the measurement directions. Both Alice and Bob are choosing each of the three bases with equal probability, and hence each of the nine possible direction pairs occurs with probability $\frac{1}{9}$.

We can see that each row contains at least five **A**s. This means that for a given pair of qubits with any configuration, the probability of getting an **A** is at least $\frac{5}{9}$. Further, since the probability of getting an **A** is at least $\frac{5}{9}$ for each of the spin configurations, we can safely assume that the overall probability must be at least $\frac{5}{9}$, irrespective of the proportion of time we get any of the configurations.

The quantum theory model told us that Alice's and Bob's sequence will agree exactly half the time. The classical model tells us that Alice's and Bob's sequence will agree at least $(\frac{5}{9})^{th}$ of the time. **This is how we can distinguish between the two theories**.

John Clauser and Stuart Freedman first performed the experiment to verify Bell's result in 1972. The results showed that the quantum mechanical predictions were correct. The experimenters have made some assumptions that could not be verified, leaving a doubt that the classical view may still be correct. Since 1972, the experiment has been repeated many times, each time with increasing sophistication, and each time the results have confirmed quantum mechanical observations.

The early experiments were riddled with three challenges. First, Alice and Bob were too close to each other. Secondly, the measurements were missing too many entangled particles. Thirdly, the direction selected by Alice and Bob were not truly random.

If the experimenters are too close to each other, it is possible that the measurement is influenced by some other mechanism. For example, it might happen that as soon as the first measurement is made, a photon travels to influence the second measurement. To counter this, the experimenters are placed far enough apart to ensure that the time interval between their measurements is less than the time it takes for a photon to travel between them. To further address the first challenge, entangled photons are used since, unlike entangled electrons, they can travel long distances without interacting with the environment. Unfortunately, this property of not interacting readily with the outside world makes it difficult to measure them. In experiments involving photons, many of the entangled photons escape measurement, so it is theoretically possible that there is some selection bias going on — the results are reflecting the properties of a nonrepresentative sample. To counter the selection bias loophole, electrons have been used. But if electrons are used, how do you get the entangled electrons far enough apart before you measure them? This is exactly the problem that the team from Delft, which we mentioned in the previous chapter, solved using electrons trapped in diamonds that are entangled with photons. Their experiment seems to have closed both loopholes simultaneously.

The problem of randomness is difficult to handle. If the Copenhagen interpretation is accepted, producing a stream of random numbers is easy. Since we are questioning this interpretation in relation to randomness, we need to test the strings of numbers of **A**s and **D**s to check for randomness. Mathematics and Statistics have given many tests to look for underlying patterns among numbers to check for randomness. These tests however can prove only a negative result. This means, that if a string fails the test, we know that the string is not random. The passing of the test is a good sign, but it's not a proof that

the string is random. Based on experiments done so far, we can say that no quantum mechanical generated string has failed a test for randomness.

6.8 Measurements

We have seen that the state vector jumps to one of the basis vectors when we make a measurement. Everything is deterministic until a measurement is made, and then it jumps to one of the basis vectors. The probabilities for jumping to each of the basis vectors are known exactly, but they are nevertheless probabilities.

We have been talking about "measurements" all along. What exactly do we mean by "measurement"? One explanation is that measurement generally means interacting with a macroscopic device. The measuring device is considered large enough to be modeled using classical mechanics. This means that whenever we make a measurement, we need to physically interact with the object being measured and this causes the jump. This explanation looks OK but lacks mathematical rigor.

Various interpretations of quantum mechanics have been proposed, each trying to eliminate something that seems problematic in the Copenhagen interpretation.

The *many worlds* interpretation deals with the measurement problem by stating that it only appears that the state vector is jumping to one of the many possibilities, but in fact, there are multiple universes and each of the possibilities is an actual occurrence in one of those many universes. The version of you in this universe sees one outcome, but there are other versions of you in other universes that see the other outcomes.

Bohmian mechanics tackle the introduction of probabilities. It is a deterministic theory in which particles behave like classical particles, but there is also a new entity called the pilot wave that gives the nonlocality properties.

There are many followers of each of these theories. However, at this moment, there are no tests that have shown conclusively to give preference to one of the theories over the other. Perhaps, at some point, there will be an insightful genius like Bell who can show that the different interpretations lead to different conclusions that can be experimentally differentiated and that experiments will then give us some reason for choosing one interpretation over another. But at this point, most physicists subscribe to the Copenhagen interpretation.

Before we end this chapter, we will briefly look at one of the applications of Bell's theorem for securely sharing keys used in cryptography.

6.9 The Ekert Protocol and Quantum Key Distribution

In 1991, Artur Ekert proposed a method based on entangled qubits used in Bell's test. The qubits are in entangeled state $\frac{1}{\sqrt{2}}|\uparrow\rangle|\uparrow\rangle + \frac{1}{\sqrt{2}}|\downarrow\rangle|\downarrow\rangle$. Both Alice and Bob receive a stream of qubits, Alice receives one qubit and Bob receives the other.

We know that if Alice and Bob measure their qubits using the same orthonormal basis, they will get either 0 or 1 with equal probability, and both will get the same answer. We devise a protocol where both Alice and Bob decide to measure the qubits using the standard basis every time. They both will end with exactly the same string of bits, a random string of 0s or 1s. The problem is that this method is not at all secure. If Eve wants to intercept Bob's qubits, she can measure them in the standard basis and then send the observed unentangled qubit to Bob. The end result is that Alice, Bob, and Eve all have identical strings in the end.

The solution is to measure the qubits using a random choice of three bases — exactly as we did with the Bell test. As in the BB84 protocol, for each measurement, Alice and Bob write down both the result and the basis that they chose. After they have made $3n$ measurements, they compare the sequences of bases that they chose. This can be done on an insecure channel — they are only revealing the basis, not the result. They will agree on approximately n of them. In each place they have chosen the same basis they will have made the same measurement. They will either both have 0, or both have 1. This gives them a string of n 0s and 1s. This will be their key if Eve is not listening in.

They now test for Eve. If Eve is eavesdropping, she will have to make measurements. Whenever she does, the entangled states become unentangled. Alice and Bob look at the strings of 0s and 1s that come from the times when they chose different bases. This gives two strings of 0s and 1s with a length of about $2n$. From the Bell inequality calculation, they know that if their states are entangled, in each place they should only agree $\frac{1}{4}$ of the time. However, if Eve is measuring one of the qubits the proportion of times they agree on changes. For example, if Eve measures a qubit before Alice and Bob have made their measurements, it is fairly straightforward to check all the possibilities to show that the proportion of times that Alice and Bob will agree on increases to $\frac{3}{8}$. This gives them a test for the presence of Eve. They calculate the proportion of agreement. If it is $\frac{1}{4}$, they can conclude that nobody has interfered and can use the key.

The Ekert protocol has the useful feature that the process generates the key. No digits need to be generated and stored beforehand, thus eliminating one of the main security threats to encryption. This protocol has been successfully carried out in the lab using entangled photons.

Chapter 7

Classical Logic, Gates and Circuits

We start this chapter by looking at the ideas of classical computation beginning with boolean functions and logic, developed by George Boole in the late nineteenth century. We then move on to cover boolean algebra and Claude Shannon's realization that boolean functions can be described using electrical switches. The electrical components that correspond to boolean functions are called logic gates and composing boolean functions becomes the study of circuits involving these gates. The initial few sections are standard and can be found in any introductory computer science book.

In the 1970s, Richard Feynman developed an interest in computation partly due to his interactions with Edward Fredkin and his ideas on computation. Fredkin believed that the universe is a computer and that since the laws of physics are reversible we should study reversible computation and reversible gates. For a few years in the early 1980s, Feynman taught a course on computation at the California Institute of Technology, and these were later written up as textitFeynman Lectures on Computation. Feynman also collaborated with Edward Fredkin who believed that since the laws of physics are reversible, we should have reversible computation and reversible gates. Fredkin further proposed something called a billiard ball computer that led Feynman to think of particles, instead of billiard balls.

Earlier in the late nineteenth century, George Boole realized that some laws of logic could be expressed in algebraic terms. Let's look at the basics of boolean logic, starting with the truth tables of the basic operations.

7.1 NOT Logic

In logic, negation, also called the logical complement, is an operation that takes a proposition P to another proposition "not **P**", written $\neg\mathbf{P}$, $\sim\mathbf{P}$ or $\overline{\mathbf{P}}$. It is interpreted intuitively as being true when **P** is false, and false when **P** is true. We can further say, if **P** is true, then $\neg\mathbf{P}$ is false. If **P** is false, then $\neg\mathbf{P}$ is true.

If we use the symbols **T** and **F** to represent true and false, we can summarize the properties using a table.

P	$\neg P$
T	F
F	T

Figure 7.1: NOT Truth Table

7.2 AND Logic

The symbol for textitand is $\wedge$. If we have two statements **P** and **Q**, we can combine them to form $\mathbf{P} \wedge \mathbf{Q}$. The statement $\mathbf{P} \wedge \mathbf{Q}$ is true iff both the compoising statements **P** and **Q** are true. The truth table for textitand logic is given below - the first two columns give the possible values of **P** and **Q** and the last column gives the corresponding truth value for $\mathbf{P} \wedge \mathbf{Q}$.

P	Q	$P \wedge Q$
T	T	T
T	F	F
F	T	F
F	F	F

Figure 7.2: AND Truth Table

7.3 OR Logic

The symbol for textitor is $\vee$. If we have two statements **P** and **Q**, we can combine them to form $\mathbf{P} \vee \mathbf{Q}$. The statement $\mathbf{P} \vee \mathbf{Q}$ is true if either or both of the compoising statements **P** and **Q** are true. The truth table for textitor logic is given below - the first two columns give the possible values of **P** and **Q** and the last column gives the corresponding truth value for $\mathbf{P} \vee \mathbf{Q}$. This is sometimes called textitinclusive or. There is an textitexclusive or as well and is described in next section.

P	Q	$P \vee Q$
T	T	T
T	F	T
F	T	T
F	F	F

Figure 7.3: OR Truth Table

7.4 Exclusive-OR (XOR) Logic

The symbol for textitexclusive or is $\oplus$. If we have two statements **P** and **Q**, we can combine them to form $\mathbf{P} \oplus \mathbf{Q}$. The statement $\mathbf{P} \oplus \mathbf{Q}$ is true only if either of the compoising statements **P** and **Q** are true. The statement $\mathbf{P} \oplus \mathbf{Q}$ is false when either both **P** and **Q** are false or both **P** and **Q** are true. The truth table for textitexclusive or logic is given below - the first two columns give the possible values of **P** and **Q** and the last column gives the corresponding truth value for $\mathbf{P} \oplus \mathbf{Q}$.

P	Q	$P \oplus Q$
T	T	F
T	F	T
F	T	T
F	F	F

Figure 7.4: Exclusive OR (XOR) Truth Table

7.5 Truth Table for an Arbitrary Boolean Expression

We take an arbitrary boolean expression, $\neg(\neg\mathbf{P} \wedge \neg\mathbf{Q})$, and see how to build a truth table for it in a step by step manner.

As a first step, we list the possible values of **P** and **Q**.

P	Q
T	T
T	F
F	T
F	F

Figure 7.5: Step 1 - Building Arbitrary Expression Truth Table

We then add columns for $\neg\mathbf{P}$ and $\neg\mathbf{Q}$ yeilding following table.

P	Q	$\neg P$	$\neg Q$
T	T	F	F
T	F	F	T
F	T	T	F
F	F	T	T

Figure 7.6: Step 2 - Building Arbitrary Expression Truth Table

Next we add the column for $\neg\mathbf{P} \wedge \neg\mathbf{Q}$. This will have true values only when both $\neg\mathbf{P}$ and $\neg\mathbf{Q}$ are true.

P	Q	$\neg P$	$\neg Q$	$\neg P \wedge \neg Q$
T	T	F	F	F
T	F	F	T	F
F	T	T	F	F
F	F	T	T	T

Figure 7.7: Step 3 - Building Arbitrary Expression Truth Table

Lastly, we add the column for $\neg(\neg \mathbf{P} \wedge \neg \mathbf{Q})$. It will have true values where $\neg \mathbf{P} \wedge \neg \mathbf{Q}$ is false and conversely.

P	Q	$\neg P$	$\neg Q$	$\neg P \wedge \neg Q$	$\neg(\neg P \wedge \neg Q)$
T	T	F	F	F	T
T	F	F	T	F	T
F	T	T	F	F	T
F	F	T	T	T	F

Figure 7.8: Step 4 - Building Arbitrary Expression Truth Table

Following this step-by-step process, we can create a truth table for any arbitrary boolean expression.

7.6 Logical Equivalence

From the last section, we can see that the truth table for $\neg(\neg\mathbf{P} \wedge \neg\mathbf{Q})$ is same as truth table for $\mathbf{P} \vee \mathbf{Q}$. We will say that the statements $\mathbf{P} \vee \mathbf{Q}$ and $\neg(\neg\mathbf{P} \wedge \neg\mathbf{Q})$ are logically equivalent, and write as $\mathbf{P} \vee \mathbf{Q} \equiv \neg(\neg\mathbf{P} \wedge \neg\mathbf{Q})$.

This means that we need never use or. Every case where textitor occurs can be replaced using expressions involving $\neg$ and $\wedge$.

Can we do the same for textitexclusive or? In other words, can we replace $\oplus$ with an expression involving only the use of $\neg$ and $\wedge$? The answer is YES.

We start with the truth table for $\oplus$, textitexclusive or.

P	Q	$P \oplus Q$
T	T	F
T	F	T
F	T	T
F	F	F

Figure 7.9: Exclusive OR (XOR) Truth Table

We look for **T** in the third column. The first occurs when **P** has value **T** and **Q** has value **F**. An expression that gives us a value of **T** only for these particular truth-values of **P** and **Q** is $\mathbf{P} \wedge \neg\mathbf{Q}$.

The next value of **T** in the third column occurs when **P** has value **F** and **Q** has value **T**. An expression that gives us a value of **T** only for those particular truth-values of **P** and **Q** is $\neg\mathbf{P} \wedge \mathbf{Q}$.

These are the only places where **T** occurs in the third column. To get an expression equivalent to the one that we want, we now join all the expressions we have generated so far using $\wedge$, giving us: $\mathbf{P} \oplus \mathbf{Q} \equiv (\mathbf{P} \wedge \neg\mathbf{Q}) \wedge (\neg\mathbf{P} \wedge \mathbf{Q})$.

Using the earlier derived fact that $\mathbf{P} \vee \mathbf{Q} \equiv \neg(\neg\mathbf{P} \wedge \neg\mathbf{Q})$, we get:

$$\mathbf{P} \oplus \mathbf{Q} \equiv \neg\left(\neg\left(\mathbf{P} \wedge \neg\mathbf{Q}\right) \wedge \left(\neg\left(\neg\mathbf{P} \wedge \mathbf{Q}\right)\right)\right)$$

This means that we need never use $\oplus$. Every case where $\oplus$ occurs can be replaced using expressions involving $\neg$ and $\wedge$. The method we have just used for replacing $\oplus$ using $\neg$ and $\wedge$ works quite generally.

7.7 Functional Completeness

The logical operators can also be considered mathematical functions. The operator $\neg$ can be visualized as a function of one input and one output; the operator $\vee$ can be considered as a function of two inputs and one output. We can also construct an arbitrary function that takes several inputs, each being a **T** or an **F**, and produces an output, again a **T** or an **F**. We call it a textitboolean function. As an example, let us construct a custom boolean function taking three inputs, $\mathbf{P}, \mathbf{Q}, \mathbf{R}$ and producing an output represented by $\mathbf{f}(\mathbf{P}, \mathbf{Q}, \mathbf{R})$. For a complete definition, we will have to complete this truth table.

P	Q	R	$f(P,Q,R)$
T	T	T	
T	T	F	
T	F	T	
T	F	F	
F	T	T	
F	T	F	
F	F	T	
F	F	F	

Figure 7.10: Boolean Function Outline

In the last column, we need to fill 8 values. Since each can be either **T** or **F**, we have a total of $\mathbf{2^8}$ possibilities. We will see that no matter how we fill the last column, we can find an equivalent expression involving only $\wedge$ and $\neg$.

To put a concrete example, let us fill some combination of **T** and **F** in the last column and follow the reasoning of the previous section to find the equivalent expression.

P	Q	R	$f(P,Q,R)$
T	T	T	F
T	T	F	F
T	F	T	T
T	F	F	F
F	T	T	F
F	T	F	T
F	F	T	F
F	F	F	T

Figure 7.11: Boolean Function Example

The first **T** occurs when **P** and **R** have values **T**, and **Q** has value **F**. A function that gives us a value of **T** for only this set of truth-values is $\mathbf{P} \wedge \neg\mathbf{Q} \wedge \mathbf{R}$. The next **T** occurs when **P** and **R** have values **F** and **Q** has value **T**. A function that gives us a value of **T** for only this set of truth-values is $\neg\mathbf{P} \wedge \mathbf{Q} \wedge \neg\mathbf{R}$. The final **T** occurs when **P**, **Q**, and **R** all have value **F**. A function that gives us a value of **T** for only this set of truth-values is $\neg\mathbf{P} \wedge \neg\mathbf{Q} \wedge \neg\mathbf{R}$.

An expression that takes on value T in just these three cases is given by: $(\mathbf{P} \wedge \neg\mathbf{Q} \wedge \mathbf{R}) \vee (\neg\mathbf{P} \wedge \mathbf{Q} \wedge \neg\mathbf{R}) \vee (\neg\mathbf{P} \wedge \neg\mathbf{Q} \wedge \neg\mathbf{R})$.

We now use the fact that $\mathbf{P} \vee \mathbf{Q} \equiv \neg(\neg\mathbf{P} \wedge \neg\mathbf{Q})$ step by step and simplify the above expression.

Replacing the first expression yields:

$$\mathbf{f}(\mathbf{P}, \mathbf{Q}, \mathbf{R}) \equiv \neg(\neg(\mathbf{P} \wedge \neg\mathbf{Q} \wedge \mathbf{R}) \wedge (\neg\mathbf{P} \wedge \mathbf{Q} \wedge \neg\mathbf{R})) \vee (\neg\mathbf{P} \wedge \neg\mathbf{Q} \wedge \neg\mathbf{R})$$

Finally replacing the second instance, we get:

$$\mathbf{f}(\mathbf{P}, \mathbf{Q}, \mathbf{R}) \equiv \neg(\neg[\neg(\neg(\mathbf{P} \wedge \neg\mathbf{Q} \wedge \mathbf{R}) \wedge \neg(\neg\mathbf{P} \wedge \mathbf{Q} \wedge \neg\mathbf{R}))] \wedge \neg[\neg\mathbf{P} \wedge \neg\mathbf{Q} \wedge \neg\mathbf{R}])$$

This method works in general. If $\mathbf{f}$ is a function that is defined by a truth table, then $\mathbf{f}$ is logically equivalent to some expression that involves only the functions $\wedge$ and $\neg$. Since we can generate any boolean function whatsoever using just these two functions, we say that $\{\wedge, \neg\}$ is a textitfunctionally complete set of boolean operators.

We have so far seen that any logical function can be expressed in terms of only $\{\wedge, \neg\}$. However, we can do even better. Using the binary operator **NAND** alone, we can express any expression solely in terms of it. We will study it next.

7.8 NAND Logic

Nand is formed by combining textitnot and textitand logic, and is represented by $\uparrow$ symbol. More formally, $\mathbf{P} \uparrow \mathbf{Q} = \neg(\mathbf{P} \wedge \mathbf{Q})$. The truth table is given below.

P	Q	$P \uparrow Q$
T	T	F
T	F	T
F	T	T
F	F	T

Figure 7.12: NAND truth table

The last section showed that $\{\wedge, \neg\}$ is a functionally complete set and any arbitrary logical expression can be expressed in terms of these. We now show that textitNAND is functionally complete. To this, we need to show equivalent expressions of $\wedge$ and $\neg$ operators in terms of $\uparrow$.

Consider the following truth table consisting of statement $\mathbf{P}$, $\mathbf{P} \wedge \mathbf{P}$, and then $\mathbf{P} \uparrow \mathbf{P}$.

P	$P \wedge P$	$\neg(P \wedge P)$
T	T	F
F	F	T

Figure 7.13: NOT truth table

From above truth table, its clear that $\neg\mathbf{P} \equiv \neg(\mathbf{P} \wedge \mathbf{P})$. However by definition, $\mathbf{P} \uparrow \mathbf{P} = \neg(\mathbf{P} \wedge \mathbf{P})$, so we can say that $\neg\mathbf{P} \equiv \mathbf{P} \uparrow \mathbf{P}$.

Let's focus on *and* now. We know:

$\mathbf{P} \wedge \mathbf{Q} \equiv \neg\neg(\mathbf{P} \wedge \mathbf{Q})$

Since $\mathbf{P} \uparrow \mathbf{Q} = \neg(\mathbf{P} \wedge \mathbf{Q})$, we get

$\mathbf{P} \wedge \mathbf{Q} \equiv \neg(\mathbf{P} \uparrow \mathbf{Q})$

Using the identity $\neg\mathbf{P} \equiv \mathbf{P} \uparrow \mathbf{P}$, we finally get

$\mathbf{P} \wedge \mathbf{Q} \equiv (\mathbf{P} \uparrow \mathbf{Q}) \uparrow (\mathbf{P} \uparrow \mathbf{Q})$

Boolean variables take on one of two values. We have been using **T** and **F** for these, but we can use any two symbols. In particular, we can use **0** and **1**. We will use **0** to represent **F** and **1** to represent **T**.

7.9 Logic Gates and Circuits

It was soon realized that if logic can be expressed in terms of algebra, then we can design machines that can perform logical operations. Claude Shannon showed that entire boolean algebra can be performed using electrical switches. At discrete time intervals, either a pulse of the electrical signal is transmitted or not. If at an appropriate time interval we receive a pulse, we can this of it to represent truth value **T** or binary value **1**. If we do not, we can use it to represent truth value **T** or binary value **0**. The combination of switches that correspond to the binary operators is called textitgates. We now look at some of these gates.

We start with the **NOT Gate**. If we input **1**, we get **0**. If we input **0**, we get **1**. We represent the **NOT Gate** as in following picture.

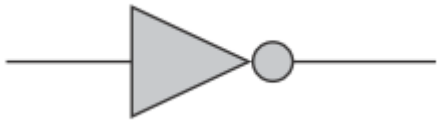

Figure 7.14: NOT Gate

Next, we see the **AND, OR, and NAND Gates** with all possible inputs and output.

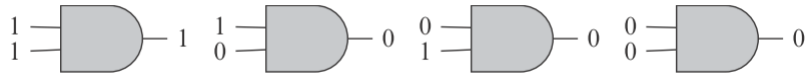

Figure 7.15: AND Gate

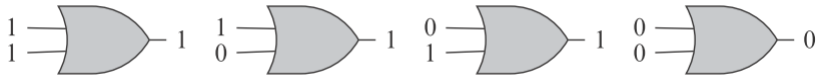

Figure 7.16: OR Gate

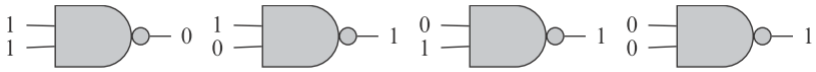

Figure 7.17: NAND Gate

These logic gates can be connected to form logic circuits. These are read from left to right, the inputs are fed into left leads and output is read from right leads. We look at some of the examples corresponding to boolean functions that we looked at earlier.

We start with boolean expression $\neg(\neg\mathbf{P} \wedge \neg\mathbf{Q})$. The corresponding circuit is shown in below figure with input and output clearly defined. If we recall that $\neg(\neg\mathbf{P} \wedge \neg\mathbf{Q}) \equiv \mathbf{P} \vee \mathbf{Q}$, this circuit is equivalent to **OR Gate**.

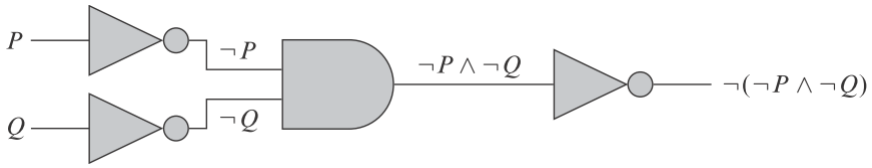

Figure 7.18: OR Circuit

Next we look at $\mathbf{P} \uparrow \mathbf{P}$ and we know that $\mathbf{P} \uparrow \mathbf{P} \equiv \neg\mathbf{P}$. So, we can say that this circuit is equivalent to a **NOT Gate**.

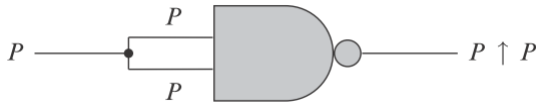

Figure 7.19: NOT Circuit

We next look at expression $(\mathbf{P} \uparrow \mathbf{Q}) \uparrow (\mathbf{P} \uparrow \mathbf{Q})$ and since $\mathbf{P} \wedge \mathbf{Q} \equiv (\mathbf{P} \uparrow \mathbf{Q}) \uparrow (\mathbf{P} \uparrow \mathbf{Q})$, the following circuit is equivalent to **AND Gate**.

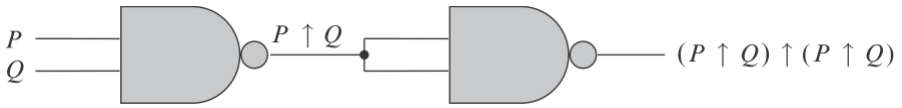

Figure 7.20: AND Circuit

7.10 Gates and Computing

We have already seen how Gates can be used to perform a logical operation. In addition, these can be used for computing operations too. Let's take an example to understand better.

We start by revisiting XOR operation defined as:

$\mathbf{0} \oplus \mathbf{0} = \mathbf{0}, \quad \mathbf{0} \oplus \mathbf{1} = \mathbf{1}, \quad \mathbf{1} \oplus \mathbf{0} = \mathbf{1}, \quad \mathbf{1} \oplus \mathbf{1} = \mathbf{0}$

This is similar to adding odd and even numbers. We know:

$\mathbf{even} + \mathbf{even} = \mathbf{even}$

$\mathbf{even} + \mathbf{odd} = \mathbf{odd}$

$\mathbf{odd} + \mathbf{even} = \mathbf{odd}$

$\mathbf{odd} + \mathbf{odd} = \mathbf{even}$

This addition of "odd" and "even" is generally called "addition modulo 2". If wee let **0** stand for "even" and **1** stand for "odd", we can see that "addition modulo 2" is same as XOR, $\oplus$, operation. Symbolically, XOR is represented as below.

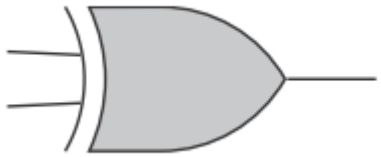

Figure 7.21: XOR Gate

Next, we consider a construct called **half-adder** which can be used to add two binary digits. We will see how to construct this using XOR and AND gates. If we have two digits that add up to less than **10**, we just add them and there is no carry digit. For example, $\mathbf{1} + \mathbf{5} = \mathbf{6}$. If the digits add up to more than **10**, we must remember the carry digit. For example, $\mathbf{3} + \mathbf{9} = \mathbf{2}$, and we have a carry of **1**.

A binary **half-adder** does a similar computation and can be constructed using XOR and AND gates. The XOR gate computer the digit sum and AND gate computes the carry part. More specifically, we have:

$\mathbf{0} + \mathbf{0} = \mathbf{0}, \mathbf{with\ carry} = \mathbf{0}$

$\mathbf{0} + \mathbf{1} = \mathbf{1}, \mathbf{with\ carry} = \mathbf{0}$

$\mathbf{1} + \mathbf{0} = \mathbf{1}, \mathbf{with\ carry} = \mathbf{0}$

$\mathbf{1} + \mathbf{1} = \mathbf{0}, \mathbf{with\ carry} = \mathbf{1}$

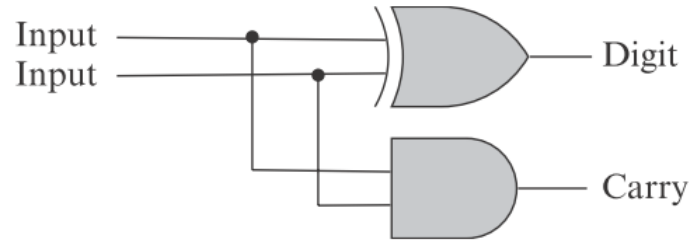

Figure 7.22: Half Adder

The reason that this is called a half-adder, and not just an adder, is that it doesn't take into account that we might have a carry coming in from the step before. Since all of our gates can be replaced with NAND gates, we can build an adder just using NAND gates.

7.11 Gates and Memory

We have already seen how to use logic gates to build computing components. To build a computer, we also need a memory unit to store data. Logic gates can be used to build **flip-flop**, a circuit that has two stable states and can be used to store state information. A brief introduction to workings of flip-flop can be found here. The following figure shows a schematic diagram of a flip-flop realized using NAND Gates.

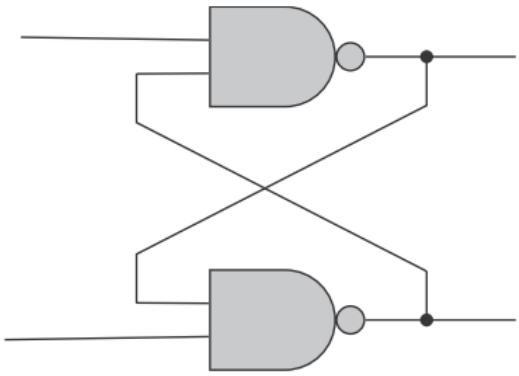

Figure 7.23: Flip Flop Using NAND Gates

It's built using a feedback loop - the output of the gates is fed back into the inputs. Once we start using feedback it is important to get the timing of inputs and outputs exactly right. This is where the clock comes in, sending pulses of electricity at constant time intervals.

7.12 Reversible Gates and Computation

We have already seen that gates can be considered as boolean functions. For example, the AND gate takes two boolean inputs and gives one boolean output. It is represented by the following truth table.

AND

Input		Output
0	0	0
0	1	0
1	0	0
1	1	1

Figure 7.24: AND Truth Table

Similarly, a half-adder can also be represented using a truth table which has two boolean inputs and two boolean outputs.

Half-adder

Input		Output	
		digit	carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Figure 7.25: Half Adder Truth Table

In this section, we will look at reversible gates. These correspond to invertible functions. Given an output, can we determine what the input was? If we can in every case, the function is invertible — the gate is reversible.

Let's look at AND gate first. We know that if the output is **1**, the input values must have been both **1**. However, if we get an output of **0**, there are three sets of input values that can give this output. If we are not given any other information, we have no way of knowing which of the three input values gave the output **0**. Thus, we will say that AND is not a reversible gate.

When we look at the half-adder truth table, it's easy to see that half-adder is also not reversible. There are two pairs of input values that give a digit of **1** and a carry of **0**. In both of these cases, we have two bits of input but are not getting two bits of output. We have lost some information doing the computation.

The study of reversible gates and reversible computation began by looking at the thermodynamics of computation and Entropy is a defined property in thermodynamics. Shannon then defined entropy for information and wondered if these two entropies are related to each other. He wondered if there is any minimum energy required to perform computation? John von Neumann conjectured that when information was lost energy is expended — it dissipates as heat. Rolf Landauer proved the result and gave the minimum possible amount of energy

to erase one bit of information. This amount of energy is called the **Landauer limit**.

If the computation is reversible, however. no information is lost and theoretically, it can be performed with no energy loss. We next look at three reversible gates: the CNOT, Toffoli, and Fredkin gates.

7.13 Controlled NOT (CNOT) Gate

The CNOT gate is binary, takes two inputs, and gives two outputs. The first input bit, represented by $\mathbf{x}$, is called the "control bit". If this bit is **0**, then there is no effect on the second bit. If the control bit is **1**, the gate acts as NOT gate on the first bit. The control bit is the first input bit, is not changed by the gate, and becomes the first output bit. The second output is the same as the second input if the control bit is **0**, but it's flipped when the control bit is **1**. We can express this as $\mathbf{f}(\mathbf{x}, \mathbf{y}) = (\mathbf{x}, \mathbf{x} \oplus \mathbf{y})$, or equivalently with the below truth table.

CNOT

Input		Output	
x	y	x	$x \oplus y$
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0

Figure 7.26: CNOT Truth Table

It's easy to see that this is an invertible gate. For any pair of output values,

there is exactly one pair of input values that correspond to it. The CNOT circuit and its usual representation are shown next.

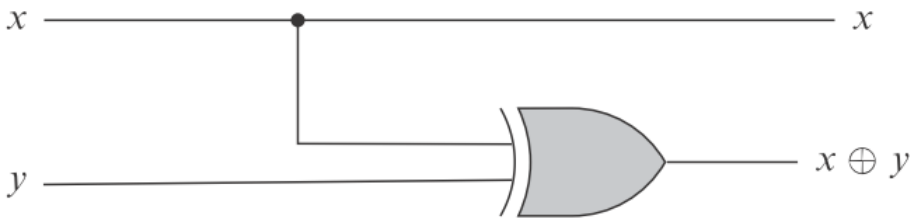

Figure 7.27: CNOT Circuit

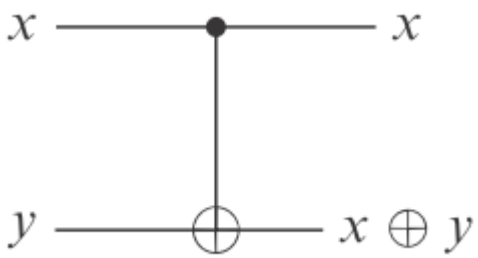

Figure 7.28: CNOT Usual Reprsentation

We cover one more important property of the CNOT gate before moving on to other invertible gates. The CNOT gate is not only invertible but also its own inverse. This means that if we put two CNOT gates in a series, where the output of the first gate becomes the input of the second gate, the output from the second gate is identical to the input to the first gate. The second gate undoes what the first gate does. We can prove it easily.

We know that applying the CNOT gate once is given by:

$\mathbf{f}(\mathbf{x}, \mathbf{y}) = (\mathbf{x}, \mathbf{x} \oplus \mathbf{y})$

Using this output as the input of another CNOT gate gives:

$\mathbf{f}(\mathbf{x}, \mathbf{x} \oplus \mathbf{y}) = (\mathbf{x}, \mathbf{x} \oplus \mathbf{x} \oplus \mathbf{y})$

Since $\mathbf{x} \oplus \mathbf{x} = 0$ and $\mathbf{0} \oplus \mathbf{y} = y$, we get:

$\mathbf{f}(\mathbf{x}, \mathbf{x} \oplus \mathbf{y}) = (\mathbf{x}, \mathbf{x} \oplus \mathbf{x} \oplus \mathbf{y}) = (x, y)$

Q.E.D.

7.14 The Toffoli Gate

The Toffoli gate, invented by Tommaso Toffoli, has three inputs and three outputs. The first two inputs are control bits. They flip the third bit if they are both 1, otherwise, the third bit remains the same. Mathematically, its described as: $\mathbf{T}(\mathbf{x}, \mathbf{y}, \mathbf{z}) = (\mathbf{x}, \mathbf{y}, (\mathbf{x} \wedge \mathbf{y}) \oplus \mathbf{z})$. Its truth table and typical representation are given below.

Toffoli gate

Input			Output		
x	y	z	x	y	$(x \wedge y) \oplus z)$
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	0	0
1	0	1	1	0	1
1	1	0	1	1	1
1	1	1	1	1	0

Figure 7.29: Toffoli Truth Table

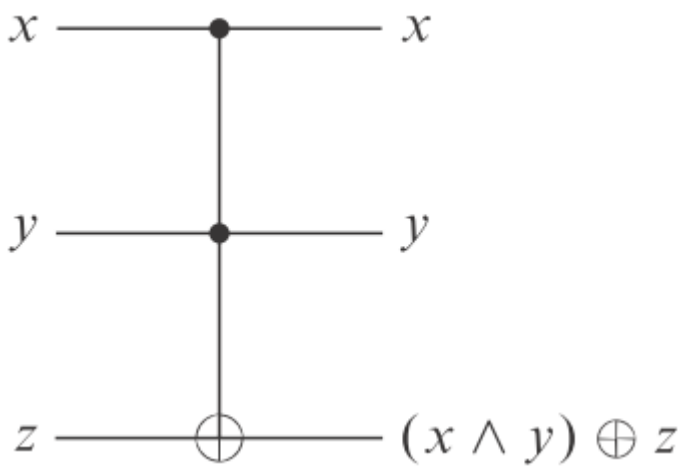

Figure 7.30: Toffoli Usual Representation

We can see from the truth table that Toffoli is invertible as each triplet of output corresponds to exactly one triplet of input values. We can also prove that the Toffoli gate, like the CNOT gate, is its own inverse.

We know that: $\mathbf{T}(\mathbf{x}, \mathbf{y}, \mathbf{z}) = (\mathbf{x}, \mathbf{y}, (\mathbf{x} \wedge \mathbf{y}) \oplus \mathbf{z})$

Now, using the output as the next input and applying the Toffoli gate definition again, we get:

$\mathbf{T}(\mathbf{x}, \mathbf{y}, (\mathbf{x} \wedge \mathbf{y}) \oplus \mathbf{z}) = (\mathbf{x}, \mathbf{y}, (\mathbf{x} \wedge \mathbf{y}) \oplus (\mathbf{x} \wedge \mathbf{y}) \oplus \mathbf{z})$

using the fact that $(\mathbf{x} \wedge \mathbf{y}) \oplus (\mathbf{x} \wedge \mathbf{y}) = \mathbf{0}$ and $\mathbf{0} \oplus \mathbf{z} = \mathbf{z}$, we get:

$\mathbf{T}(\mathbf{x}, \mathbf{y}, (\mathbf{x} \wedge \mathbf{y}) \oplus \mathbf{z}) = (\mathbf{x}, \mathbf{y}, \mathbf{z})$, which is same as input.

Q.E.D.

We can show that the **Toffoli gate is a universal gate**. We know that any boolean logic can be written using NAND and fan-outs. To prove that the Toffoli gate is universal, it is sufficient to prove that both NAND and fan-outs can be computed solely from the Toffoli gate.

We first prove for NAND logic. The NAND gate is defined as $(f(x, y) = \neg(x \wedge y))$. So we want to use the Toffoli gate such that when inputs are $\mathbf{x}, \mathbf{y}$, we get an output of $\neg(\mathbf{x} \wedge \mathbf{y})$. However, the Toffoli gate uses three inputs and gives three outputs. We start by noticing $\neg(x \wedge y) \equiv (x \wedge y) \oplus 1$, and hence we can choose

the third input to Toffoli gate to always be **1** and ignore the first two output values. Mathematically, we can write it as $\mathbf{T}(\mathbf{x}, \mathbf{y}, \mathbf{1}) = (\mathbf{x}, \mathbf{y}, (\mathbf{x} \wedge \mathbf{y}) \oplus \mathbf{1}) = (\mathbf{x}, \mathbf{y}, \neg(\mathbf{x} \wedge \mathbf{y}))$ to show that we emulate NAND gate by inputting $\mathbf{x}, \mathbf{y}$ and reading off only the third value from the output of Toffoli gate.

We follow the same logic for showing how to construct a fan-out from a Toffoli gate. A fan-out means that we input one value, $\mathbf{x}$ and receive two output values that are both the same, again $\mathbf{x}$. Since a Toffoli gate has three inputs and three outputs, we chose the two other inputs apart from $\mathbf{x}$ such that we get two $\mathbf{x}$ as output and ignore the third one. We do achieve it in this configuration: $\mathbf{T}(\mathbf{x}, \mathbf{1}, \mathbf{0}) = (\mathbf{x}, \mathbf{1}, \mathbf{x})$.

We can thus prove that any logical circuit can be constructed solely using Toffoli gates.

While using reversible gates, it commonly happens that even though the number of inputs and outputs must be equal, we want to compute with an unequal number of inputs and outputs. This is typically done by adding additional bits, often called **ancilla bits**, to the inputs, or by ignoring bits that are output. The Output bits that are ignored are sometimes called **garbage bits**. In the example above where we realized fan-out using Toffoli gate, it was showed that $\mathbf{T}(\mathbf{x}, \mathbf{1}, \mathbf{0}) = (\mathbf{x}, \mathbf{1}, \mathbf{x})$; the bits $\mathbf{1}, \mathbf{0}$ in the input are the ancilla bits and the **1** in the output is the garbage bit.

7.15 The Fredkin Gate

The Fredkin gate also has three inputs and three outputs. The first input is a control bit. If it is **0**, the second and third inputs are unchanged. If the control bit is **1**, it swaps the second and third inputs - the second output is the third input and the third output is the second input. More formally, it is defined as:

$\mathbf{F}(\mathbf{0}, \mathbf{y}, \mathbf{z}) = (\mathbf{0}, \mathbf{y}, \mathbf{z})$ $\qquad$ $\mathbf{F}(\mathbf{1}, \mathbf{y}, \mathbf{z}) = (\mathbf{1}, \mathbf{z}, \mathbf{y})$

The truth table corresponding to Fredkin Gate is given below.

Fredkin gate

Input			Output		
x	y	z	x		
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	0	0
1	0	1	1	1	0
1	1	0	1	0	1
1	1	1	1	1	1

Figure 7.31: Fredkin Gate Truth Table

The graphical representation of a Fredkin gate is given below.

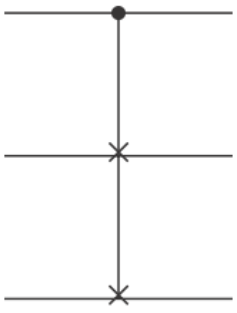

Figure 7.32: Fredkin Usual Representation

We can see from the truth table that Fredkin is invertible as each triplet of output corresponds to exactly one triplet of input values. We can also prove that the Fredkin gate, like CNOT and Toffoli gates, is its own inverse. Another

interesting property of Fredkin gate, as can be seen from its truth table, is that the number of **1**s for each input is equal to the number of **1**s in the corresponding output. We will make use of this property in the next section.

From the definition of Fredkin gate, we know $\mathbf{F}(\mathbf{0},\mathbf{0},\mathbf{1}) = (\mathbf{0},\mathbf{0},\mathbf{1})$ and $\mathbf{F}(\mathbf{1},\mathbf{0},\mathbf{1}) = (\mathbf{1},\mathbf{1},\mathbf{0})$. Thus, for both possible values of $\mathbf{x}$, we have $\mathbf{F}(\mathbf{x},\mathbf{0},\mathbf{1}) = (\mathbf{x},\mathbf{x},\neg\mathbf{x})$. This means that Fredkin gate can be used for both negation and fan-out logic. For nagtion, we consider both the $\mathbf{x}$s as garbage bit and for fan-out, we consider $\neg\mathbf{x}$ as garbage bit.

Moreover, if we consider $\mathbf{z} = 0$ in Fredkin gate, we get:

$\mathbf{F}(\mathbf{0},\mathbf{0},\mathbf{0}) = (\mathbf{0},\mathbf{0},\mathbf{0})$

$\mathbf{F}(\mathbf{0},\mathbf{1},\mathbf{0}) = (\mathbf{0},\mathbf{1},\mathbf{0})$

$\mathbf{F}(\mathbf{1},\mathbf{0},\mathbf{0}) = (\mathbf{1},\mathbf{0},\mathbf{0})$

$\mathbf{F}(\mathbf{1},\mathbf{1},\mathbf{0}) = (\mathbf{1},\mathbf{0},\mathbf{1})$

We can write this as: $\mathbf{F}(\mathbf{x},\mathbf{y},\mathbf{0}) = (\mathbf{x},\neg\mathbf{x}\wedge\mathbf{y},\mathbf{x}\wedge\mathbf{y})$ which tells us that we can use the Fredkin gate to construct the AND gate ($\mathbf{0}$ is an ancilla bit, and both $\mathbf{x}$ and $\neg\mathbf{x}\wedge\mathbf{y}$ are garbage bits). Since any boolean circuit can be constructed using just NOT and AND gates along with fan-out, we can construct any boolean circuit using just Fredkin gates. This proves that the Fredkin gate is universal.

We started with this definition of Fredkin gate. Let us look at another expression for the same which will come in handy in the next section.

$\mathbf{F}(\mathbf{0},\mathbf{y},\mathbf{z}) = (\mathbf{0},\mathbf{y},\mathbf{z})$ $\qquad$ $\mathbf{F}(\mathbf{1},\mathbf{y},\mathbf{z}) = (\mathbf{1},\mathbf{z},\mathbf{y})$

This gate outputs three numbers. The first number output is always equal to the first input $\mathbf{x}$. The second output will be $\mathbf{1}$ if either $\mathbf{x} = \mathbf{0}$ and $\mathbf{y} = \mathbf{1}$ or if $\mathbf{x} = \mathbf{1}$ and $\mathbf{z} = \mathbf{1}$. This can be expressed as $(\neg\mathbf{x}\wedge\mathbf{y})\vee(\mathbf{x}\wedge\mathbf{z})$. The third output will be $\mathbf{1}$ if either $\mathbf{x} = \mathbf{0}$ and $\mathbf{z} = \mathbf{1}$ or if $\mathbf{x} = \mathbf{1}$ and $\mathbf{y} = \mathbf{1}$ which can be expressed as $(\neg\mathbf{x}\wedge\mathbf{z})\vee(\mathbf{x}\wedge\mathbf{y})$. To summarize, this gives a new definition of Fredkin gate as: $\mathbf{F}(\mathbf{x},\mathbf{y},\mathbf{z}) = (\mathbf{x},(\neg\mathbf{x}\wedge\mathbf{y})\vee(\mathbf{x}\wedge\mathbf{z}),(\neg\mathbf{x}\wedge\mathbf{z})\vee(\mathbf{x}\wedge\mathbf{y}))$.

7.16 Billiard Ball Computing

We have been talking about gates in this chapter. How do we build them? One way to build gates is via electrical switches where the presence or absence of signal can represent logical $\mathbf{1}$ and $\mathbf{0}$ respectively. Intuitively, Fredkin demonstrated that one can build logical gates using billiard balls and mirrors. A mirror

is considered as a solid wall that the billiard ball bounces off and they are called mirrors because they ensure the angle of incidence is equal to the angle of reflection. Billiard ball gates are theoretical devices where the balls have the same size, mass, speed and all collisions are elastic and no energy is lost. One such gate, called switch-gate, is shown in the next figure where solid lines represent mirrors and the grid lines are used to keep track of the center of billiard balls as they move.

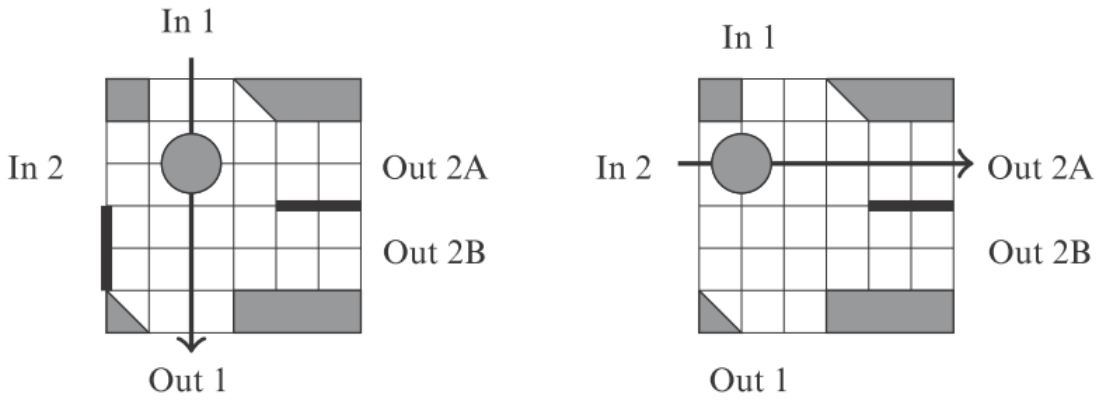

Figure 7.33: Billiard Ball Gate

We have four scenarios to consider. In the picture on the left, we see a ball that has entered via input "In 1" and since there is no other ball to interact with it, it moves unimpeded and exits via output "Out 1". On the similar lines, in the figure on right, the ball enters via input "In 2" and since there is no other ball to interact with it, it moves unhindered and exits via output "Out 2A".

In the third scenario, if no balls enter through the two input slots, then no balls exit. In the last scenario, we consider what happens when balls are sent through both the inputs simultaneously. In this case, they collide with each other in the upper-left-hand corner of the gate, then bounce off the mirror walls and redirect each other to collide again in the lower-right-hand corner of the gate. Finally, these leave via exits marked "Out 1" and "Out 2".

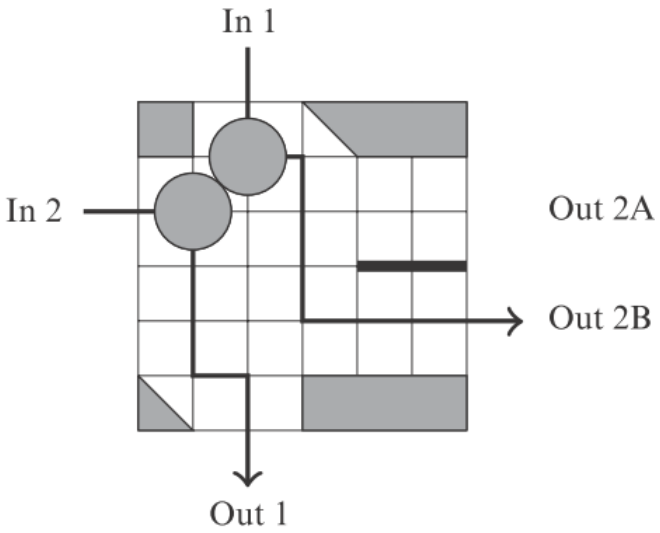

Figure 7.34: Switch Gate Collisions

If **1** represents the presence of the ball and **0** represents its absence, we get the following truth table.

Switch gate

Input		Output		
1	2	1	2A	2B
0	0	0	0	0
0	1	0	1	0
1	0	1	0	0
1	1	1	0	1

Figure 7.35: Switch Gate Truth Table

The same table can be constructed using generic $\mathbf{x}, \mathbf{y}, \neg\mathbf{x} \wedge \mathbf{y}$ and $\mathbf{x} \wedge \mathbf{y}$.

x	y	x	$\neg x \wedge y$	$x \wedge y$
0	0	0	0	0
0	1	0	1	0
1	0	1	0	0
1	1	1	0	1

Figure 7.36: Equivalent Switch Gate Truth Table

This equivalent truth table helps us to represent Switch Gate as a Black Box with inputs and outputs appropriately marked. If a ball enters via $\mathbf{x}$, it must leave via $\mathbf{x}$. If a ball enters via $\mathbf{y}$, it will leave via $(\neg\mathbf{x} \wedge \mathbf{y})$ if there is no input ball via $\mathbf{x}$ input, and it will leave via $(\mathbf{x} \wedge \mathbf{y})$ if there is an input ball entering via $\mathbf{x}$.

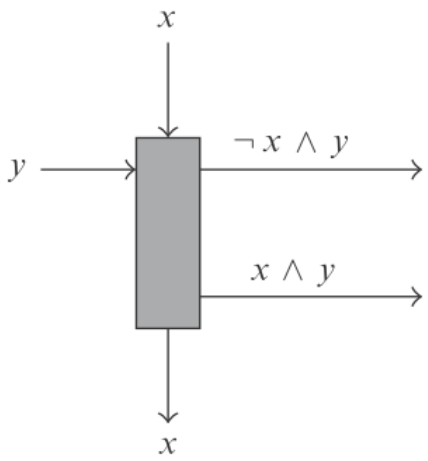

Figure 7.37: Switch Gate As a Black Box

The gate can be reversed as well as shown in the next figure, however, it requires

careful interpretation.

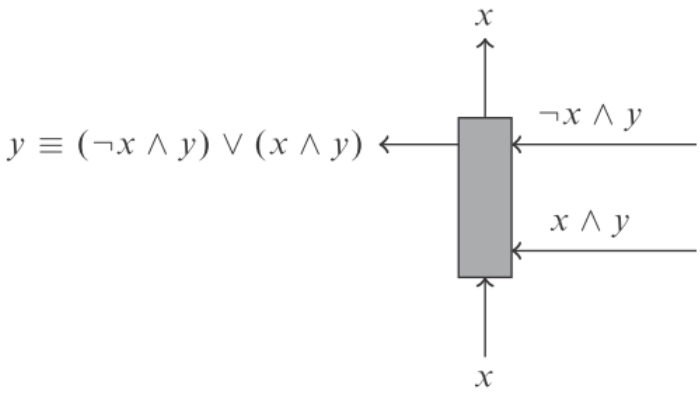

Figure 7.38: Switch Gate Reversed

If a ball enters via $(\neg \mathbf{x} \wedge \mathbf{y})$, then there won't be a ball entering via $\mathbf{x}$, and so the ball sails directly across. If a ball enters via $(\mathbf{x} \wedge \mathbf{y})$, then there will be a ball entering via $\mathbf{x}$, and consequently, they will collide. One ball exits through the top of the gate and one exits via the output on the left. This means that a ball will exit through the left output if either $(\neg \mathbf{x} \wedge \mathbf{y})$ or $(\mathbf{x} \wedge \mathbf{y})$, so this exit can be labeled $(\neg \mathbf{x} \wedge \mathbf{y}) \vee (\mathbf{x} \wedge \mathbf{y})$. However, $(\neg \mathbf{x} \wedge \mathbf{y}) \vee (\mathbf{x} \wedge \mathbf{y})$ is the same as $\mathbf{y}$, which means that reversing the gate just reverses the arrows but leaves all the labels the same.

We have covered enough ground to cover Fredkin gate now. We recall that Fredkin gate is defined as:

$$\mathbf{F}(\mathbf{x}, \mathbf{y}, \mathbf{z}) = (\mathbf{x}, (\neg \mathbf{x} \wedge \mathbf{y}) \vee (\mathbf{x} \wedge \mathbf{z}), (\neg \mathbf{x} \wedge \mathbf{z}) \vee (\mathbf{x} \wedge \mathbf{y}))$$

We basically need a construction that takes $\mathbf{x}, \mathbf{y}, \mathbf{z}$ as input and outputs $\mathbf{x}$, $(\neg \mathbf{x} \wedge \mathbf{y}) \vee (\mathbf{x} \wedge \mathbf{z})$, and $(\neg \mathbf{x} \wedge \mathbf{z}) \vee (\mathbf{x} \wedge \mathbf{y})$. This can be done using for switch gates as shown below.

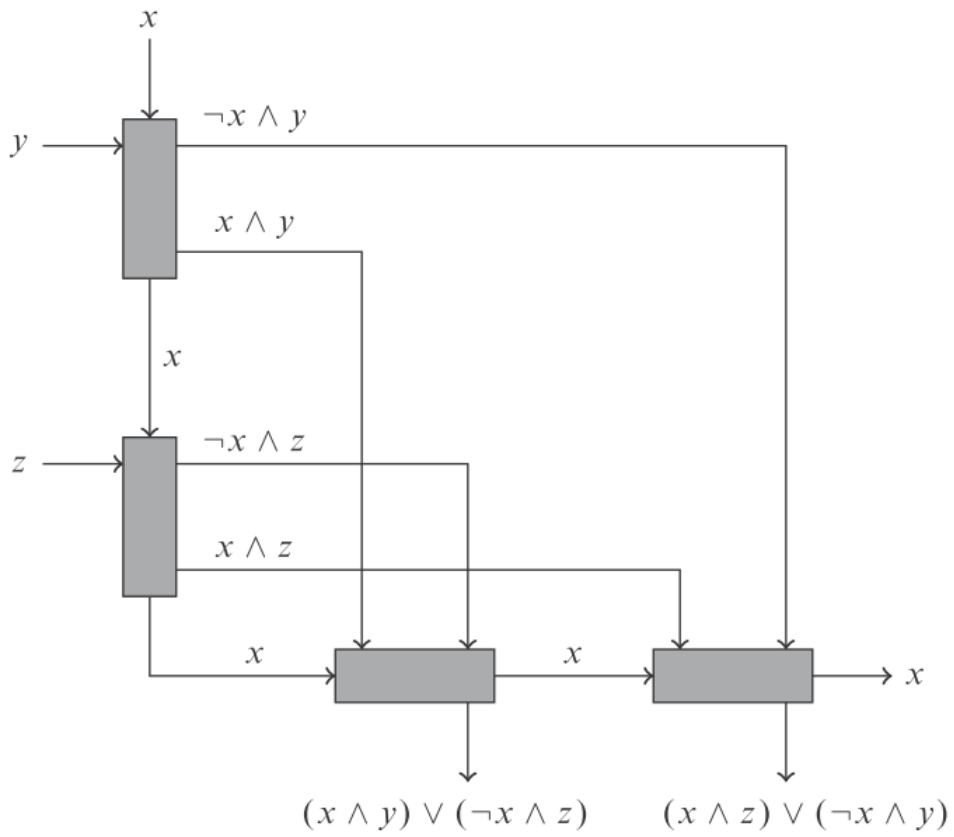

Figure 7.39: Fredkin Gate via Switch Gates

In this picture, the right angles in the paths are obtained by bouncing off diagonally placed mirrors. The only other interactions occur in the switch gates. Paths crossing don't indicate collisions; the balls pass through the intersection points at different times. To make sure that balls don't collide where they shouldn't and do collide where they should, we can always add delays to paths by adding little detours to paths using mirrors.

By putting mirrors in the appropriate places and adding delays, we can construct the gate so that the outputs are lined up with the inputs and when balls enter at the same time they leave at the same time as shown in the next figure.

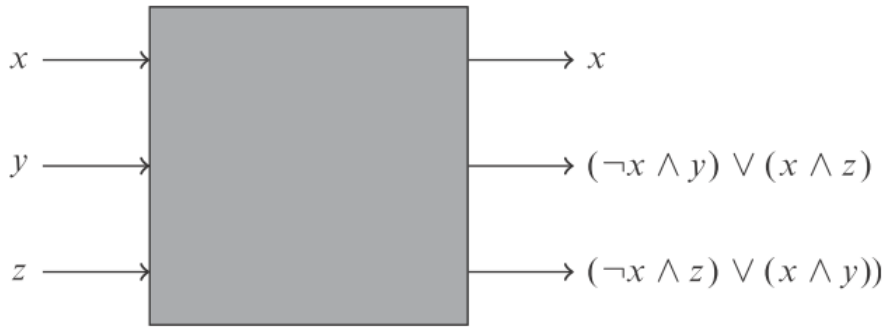

Figure 7.40: Billiard Ball Fredkin Gate

We can now form circuits that contain more than one Fredkin gate. Since the Fredkin gate is universal, it can be used to construct any boolean circuit. **Consequently, any boolean circuit can be constructed using just billiard balls and mirrors**.

Fredkin introduced the idea of the Billiard Ball computer and Feynman was impressed by it. Both also realized that any slightest of error in position or velocity of a ball would result in an error that would propagate and amplify over time. Also, collisions are never perfectly elastic as there is friction and heat is lost in the system. Even though the Billiard Ball computer is a purely theoretical device, this machine does conjure images of atoms bouncing off one another, and it led Feynman to consider gates based on quantum mechanics rather than classical mechanics. This will be the subject of the next chapter.

Chapter 8

Quantum Gates and Circuits

Quantum gates and circuits are the logical extension of classical gates and circuits. They can also be visualized as a mathematical framework used to send qubits from Alice to Bob.

Imagine yourself sitting near a window in a stationary train on a platform, along with another stationary train, just a few feet away. When one of the trains moves slowly, sometimes it is impossible to tell which train is moving without looking out the window on the opposite side. It might be the case that your train is moving forward, or the other train moving forward in opposite direction. Both the scenarios are equally possible. The same logic can be applied to Bob's measurements. We can either think that Bob is rotating his measuring apparatus, or we can think that Bob is keeping his apparatus in the same direction as Alice, but somehow, the qubits get rotated while traveling from Alice to Bob. It is more natural to think of the apparatus as fixed and the qubit being rotated between the time it is sent and the time it is measured. The process of sending the qubits through the quantum gate does this rotation. Earlier we noticed that choosing directions to measure the qubits correspond to choosing an orthogonal matrix. Now, we can consider the directions of measurement as fixed and the orthogonal matrix correspond to a quantum gate through which the qubit passes through.

As we are thinking the measureing device as fixed, we can use only one ordered basis for both sending and receiving qubits. We can choose the standard basis $\left(\begin{bmatrix}\mathbf{1}\\\mathbf{0}\end{bmatrix}, \begin{bmatrix}\mathbf{0}\\\mathbf{1}\end{bmatrix}\right)$ which is represented by $(|\uparrow\rangle, |\downarrow\rangle)$. We further let $|\mathbf{0}\rangle$ denote $\begin{bmatrix}\mathbf{1}\\\mathbf{0}\end{bmatrix}$ and $|\mathbf{1}\rangle$ denote $\begin{bmatrix}\mathbf{0}\\\mathbf{1}\end{bmatrix}$ to reflect how these kets relate to bits.

In general, a qubit will be of the form $\mathbf{a_0}\left|\mathbf{0}\right\rangle + \mathbf{a_1}\left|\mathbf{1}\right\rangle$ where $\mathbf{a_0}^\mathbf{2} + \mathbf{a_0}^\mathbf{2} = \mathbf{1}$. When we measure this qubit, either it's state will jump to $\left|\mathbf{0}\right\rangle$ and we read $\mathbf{0}$ with probability $\mathbf{a_0}^\mathbf{2}$, or it's state will jump to $\left|\mathbf{1}\right\rangle$ and we read $\mathbf{1}$ with probability $\mathbf{a_1}^\mathbf{2}$.

We will typically have a sytem with more than one qubit. This means that we will be using tensor products and, as an example, for a system two two qubits, the ordered basis is given by $\left(\begin{bmatrix}\mathbf{1}\\\mathbf{0}\end{bmatrix}\otimes\begin{bmatrix}\mathbf{1}\\\mathbf{0}\end{bmatrix},\begin{bmatrix}\mathbf{1}\\\mathbf{0}\end{bmatrix}\otimes\begin{bmatrix}\mathbf{0}\\\mathbf{1}\end{bmatrix},\begin{bmatrix}\mathbf{0}\\\mathbf{1}\end{bmatrix}\otimes\begin{bmatrix}\mathbf{1}\\\mathbf{0}\end{bmatrix},\begin{bmatrix}\mathbf{0}\\\mathbf{1}\end{bmatrix}\otimes\begin{bmatrix}\mathbf{0}\\\mathbf{1}\end{bmatrix}\right)$. This can be succintly written as $(\left|\mathbf{0}\right\rangle\otimes\left|\mathbf{0}\right\rangle,\left|\mathbf{0}\right\rangle\otimes\left|\mathbf{1}\right\rangle,\left|\mathbf{1}\right\rangle\otimes\left|\mathbf{0}\right\rangle,\left|\mathbf{1}\right\rangle\otimes\left|\mathbf{1}\right\rangle)$ which can be further reduced to $(\left|\mathbf{0}\right\rangle\left|\mathbf{0}\right\rangle,\left|\mathbf{0}\right\rangle\left|\mathbf{1}\right\rangle,\left|\mathbf{1}\right\rangle\left|\mathbf{0}\right\rangle,\left|\mathbf{1}\right\rangle\left|\mathbf{1}\right\rangle)$. Lastly, if we let $\left|\mathbf{ab}\right\rangle$ represent $\left|\mathbf{a}\right\rangle\left|\mathbf{b}\right\rangle$, the ordered basis pair can be further reduced to $(\left|\mathbf{00}\right\rangle,\left|\mathbf{01}\right\rangle,\left|\mathbf{10}\right\rangle,\left|\mathbf{11}\right\rangle)$.

We next revisit the CNOT gate and see how to build quantum gates.

8.1 The CNOT Gate

The CNOT gate is binary, takes two inputs, and gives two outputs. The first input bit, represented by $\mathbf{x}$, is called the "control bit". If this bit is $\mathbf{0}$, then there is no effect on the second bit. If the control bit is $\mathbf{1}$, the gate acts as NOT gate on the first bit. The control bit is the first input bit, is not changed by the gate, and becomes the first output bit. The second output is the same as the second input if the control bit is $\mathbf{0}$, but it's flipped when the control bit is $\mathbf{1}$. We can express this as $\mathbf{f}(\mathbf{x},\mathbf{y}) = (\mathbf{x},\mathbf{x}\oplus\mathbf{y})$, or equivalently with the below truth table.

CNOT

Input		Output	
x	y	x	$x \oplus y$
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0

Figure 8.1: CNOT Truth Table

We can extend this to qubits by replacing **0** by $|\mathbf{0}\rangle$, and **1** by $|\mathbf{1}\rangle$. Doing this, we get the following.

CNOT

Input		Output	
x	y	x	$x \oplus y$
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0

Figure 8.2: CNOT Qubits Truth Table

This can be further reduced by using the tensor notation.

CNOT

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

Figure 8.3: CNOT Qubits Truth Table (Tensor)

We can see the effect of the CNOT gate on basis vectors. We can apply the CNOT gate to a linear combination of basis vectors and observe that it just swaps the probability amplitudes of $|\mathbf{10}\rangle$ and $|\mathbf{11}\rangle$.

$$\mathbf{CNOT}(\mathbf{r}\,|\mathbf{00}\rangle + \mathbf{s}\,|\mathbf{01}\rangle + \mathbf{t}\,|\mathbf{10}\rangle + \mathbf{u}\,|\mathbf{11}\rangle) = \mathbf{r}\,|\mathbf{00}\rangle + \mathbf{s}\,|\mathbf{01}\rangle + \mathbf{u}\,|\mathbf{10}\rangle + \mathbf{t}\,|\mathbf{11}\rangle$$

The following figure shows the usual representation of the CNOT Gate. In the case of classical bits, the bit entering the top wire from the left, leave the top wire on the right unchanged. The same is true when the top qubit entering the gate is either $|\mathbf{0}\rangle$ or $|\mathbf{1}\rangle$, but it's not true for other qubits. We see it with an example.

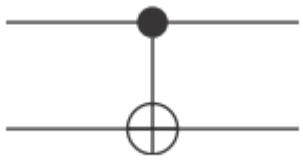

Figure 8.4: CNOT Usual Representation

We start by taking $\frac{\mathbf{1}}{\sqrt{\mathbf{2}}}\,|\mathbf{0}\rangle + \frac{\mathbf{1}}{\sqrt{\mathbf{2}}}\,|\mathbf{1}\rangle$ as the top qubit and $|\mathbf{0}\rangle$ for the bottom one.

The input to the CNOT gate will be $\left(\frac{\mathbf{1}}{\sqrt{\mathbf{2}}}|\mathbf{0}\rangle + \frac{\mathbf{1}}{\sqrt{\mathbf{2}}}|\mathbf{1}\rangle\right) \otimes |\mathbf{0}\rangle = \frac{\mathbf{1}}{\sqrt{\mathbf{2}}}|\mathbf{00}\rangle + \frac{\mathbf{1}}{\sqrt{\mathbf{2}}}|\mathbf{10}\rangle$. The CNOT gate will translate this to $\frac{\mathbf{1}}{\sqrt{\mathbf{2}}}|\mathbf{00}\rangle + \frac{\mathbf{1}}{\sqrt{\mathbf{2}}}|\mathbf{11}\rangle$. We have already seen, from EPR experiment, that this is an entangled state. Thus, we cannot assign individual states to the top and bottom wires on the right side. We represent it as follows.

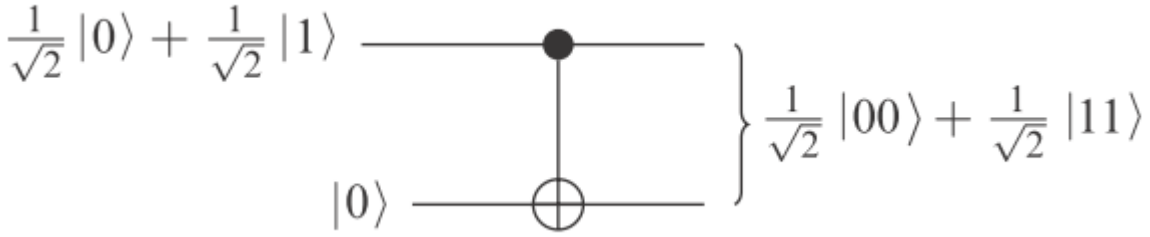

Figure 8.5: Entangled State and CNOT Gate

One last note before we move on to quantum gates. The wires in the above picture represent electrons or photons. These are separate objects that can be far apart from each other. We will routinely discuss about the top and bottom qubits, and we should always remember that they can be far apart. We know that if the qubits are entangled, a measurement on one will affect the other. We can input two unentangled qubits and use the CNOT gate to entangle them. We will often use the CNOT gate the same way.

8.2 Quantum Gates

We have seen that the CNOT gate performs permutation on basis vectors. When basis vectors in an ordered basis are permuted, we get another set of ordered orthonormal basis, and each of these basis is associated with an orthogonal matrix. Thus, the matrix corresponding to the CNOT gate is also orthogonal. Further, all the reversible gates that we discussed in the previous chapter also permute basis vectors and they all correspond to orthogonal matrices.

We can thus give a definition of quantum gates as operations that can be described by orthogonal matrices. As we saw with classical computation, we want to assemble a small set of simple gates that can be used together to form circuits. We start by looking at the simplest of gates, those that act on a single qubit.

In classical computing, there are only two possible boolean operators that can act on a bit: the **identity** operator that leaves the bit unchanged, and the **NOT** operator that flips the bit value. In the case of qubits, there are infinitely many possible gates.

We look at four gates, called Pauli Transformations. Two of these quantum gates correspond to the classical **identity** gate that leaves the qubits $|\mathbf{0}\rangle$ and $|\mathbf{1}\rangle$ unchanged. The next two quantum gates flip the qubits $|\mathbf{0}\rangle$ and $|\mathbf{1}\rangle$.

The Gates *I* and *Z*

The gate $\boldsymbol{I}$ is the identity matrix: $\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}$

Let us see how $\boldsymbol{I}$ acts on a qubit $\mathbf{a_0}|\mathbf{0}\rangle + \mathbf{a_1}|\mathbf{1}\rangle$.

$$\mathbf{I}\left(\mathbf{a_0}|\mathbf{0}\rangle + \mathbf{a_1}|\mathbf{1}\rangle\right) = \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}a_0\\ a_1\end{bmatrix} = \begin{bmatrix}a_0\\ a_1\end{bmatrix} = \mathbf{a_0}|\mathbf{0}\rangle + \mathbf{a_1}|\mathbf{1}\rangle$$

We can see that $\boldsymbol{I}$ acts as identity and leaves the qubit unchanged.

The gate $\boldsymbol{Z}$ is defined by the matrix: $\begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix}$

Let us see how $\boldsymbol{Z}$ acts on a qubit $\mathbf{a_0}|\mathbf{0}\rangle + \mathbf{a_1}|\mathbf{1}\rangle$.

$$\mathbf{Z}\left(\mathbf{a_0}|\mathbf{0}\rangle + \mathbf{a_1}|\mathbf{1}\rangle\right) = \begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix}\begin{bmatrix}a_0\\ a_1\end{bmatrix} = \begin{bmatrix}a_0\\ -a_1\end{bmatrix} = \mathbf{a_0}|\mathbf{0}\rangle - \mathbf{a_1}|\mathbf{1}\rangle$$

At the first look, $\boldsymbol{Z}$ leaves the probability amplitude of $|\mathbf{0}\rangle$ unchanged, and it changes the sign of probability amplitude of $|\mathbf{1}\rangle$. We will look at it more carefully as the gate $\boldsymbol{Z}$ isn't same as gate $\boldsymbol{I}$.

We will start by looking how gate $\boldsymbol{Z}$ acts on basis vectors. We see that $\mathbf{Z}(|\mathbf{0}\rangle) = |0\rangle$ and $\mathbf{Z}(|\mathbf{1}\rangle) = -|1\rangle$. We learnt in previous chapter that a state vector is equivalent to same state vector multiplied by $-\mathbf{1}$, that is, $-|\mathbf{1}\rangle$ is equivalent to $|\mathbf{1}\rangle$. This implies that $\boldsymbol{Z}$ preserves the basis vecotrs.

Next we apply $\boldsymbol{Z}$ to qubit $\frac{\mathbf{1}}{\sqrt{\mathbf{2}}}|\mathbf{0}\rangle + \frac{\mathbf{1}}{\sqrt{\mathbf{2}}}|\mathbf{1}\rangle$. We get:

$$\mathbf{Z}\left(\frac{\mathbf{1}}{\sqrt{\mathbf{2}}}|\mathbf{0}\rangle + \frac{\mathbf{1}}{\sqrt{\mathbf{2}}}|\mathbf{1}\rangle\right) = \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle$$

We have already seen that $\frac{\mathbf{1}}{\sqrt{\mathbf{2}}}|\mathbf{0}\rangle + \frac{\mathbf{1}}{\sqrt{\mathbf{2}}}|\mathbf{1}\rangle$ is different from $\frac{\mathbf{1}}{\sqrt{\mathbf{2}}}|\mathbf{0}\rangle - \frac{\mathbf{1}}{\sqrt{\mathbf{2}}}|\mathbf{1}\rangle$. We can sumamrize by saying that even though $\boldsymbol{Z}$ transformation preserves both

the basis vecotr, it changes all other qubits. The operation of changing the sign of probability amplitude is called changing the **relative phase** of the qubit.

The Gates *X* and *Y*

The gate $\boldsymbol{X}$ and $\boldsymbol{Y}$ are defined as follows: $\mathbf{X} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$, $\mathbf{Y} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$

Both these gates are similar to NOT gate in the sense that they interchange $|\mathbf{0}\rangle$ and $|\mathbf{1}\rangle$. The gate $\boldsymbol{X}$ just flips, and gate $\boldsymbol{Y}$ flips and changes the relative phase as well.

The Hadamard Gate

This gate is defined as: $\mathbf{H} = \frac{\mathbf{1}}{\sqrt{\mathbf{2}}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$

The most often use of the Hadamard gate is to place standard basis vectors into superpositions.

$$\mathbf{H}(|\mathbf{0}\rangle) = \frac{\mathbf{1}}{\sqrt{\mathbf{2}}}(|\mathbf{0}\rangle + |\mathbf{1}\rangle), \qquad \mathbf{H}(|\mathbf{1}\rangle) = \frac{\mathbf{1}}{\sqrt{\mathbf{2}}}(|\mathbf{0}\rangle - |\mathbf{1}\rangle).$$

So far we have seen five quantum gates that act on one qubit. Of course, there are infinitely more. Any rotation will give us an orthogonal matrix, and there are infinitely many of these, all of which can be considered as gates.

Pictorially, we will show a gate acting on a qubit by a square with the appropriate letter drawn in the center. For example, the Hadamard gate acting on one bit is denoted by the following.

Figure 8.6: Hadamard Gate

8.3 What about Universal Quantum Gates?

In the case of classical computing, we saw that Fredkin and NAND gates are considered universal since every boolean function can be realized by a logical

circuit consisting solely of these gates. Logically, we wonder if there are any universal quantum gates.

In the classical case, there are two boolean functions of one variable. There are four for the case of two variables. In general, there are $\mathbf{2^n}$ possible functions with $\mathbf{n}$ variables. We can see that there are only a finite number of boolean functions for a given number of variables. The situation is completely different with quantum gates. In the previous section, we saw there are infinite possibilities for gates that can act on a single qubit. If we have a finite number of gates and those are connected in a finite number of ways, we will end with a finite number of circuits. So, it is not possible for a finite number of gates to generate an infinite number of circuits.

In short, there is no finite set of quantum gates that is universal. However, even though it is impossible to have a finite number of quantum gates that will generate every other possible quantum circuit, it is proven that a there is a finite collection of gates can be used to approximate every possible circuit. Specifically for our scenario, all the circuits that we need can be constructed from the gates that were introduced in the previous section; five that act on one qubit and one, the CNOT gate, that acts on two qubits.

8.4 No Cloning Theorem

In the case of classical circuits, we came across the fan-out operation which can be used to split the input signal into two identical copies. The fan-out operation could also be achieved with reversible gates using the ancilla bit by taking the second input always to be $\mathbf{0}$.

We already saw one way to achieve this using the CNOT gate. By definition, we know that $\mathbf{CNOT}\left(|\mathbf{0}\rangle|\mathbf{0}\rangle\right) = |\mathbf{0}\rangle|\mathbf{0}\rangle$, and $\mathbf{CNOT}\left(|\mathbf{1}\rangle|\mathbf{0}\rangle\right) = |\mathbf{1}\rangle|\mathbf{1}\rangle$. This means, $\mathbf{CNOT}\left(|\mathbf{x}\rangle|\mathbf{0}\rangle\right) = |\mathbf{x}\rangle|\mathbf{x}\rangle$ if $|\mathbf{x}\rangle$ is either $|\mathbf{0}\rangle$ or $|\mathbf{1}\rangle$. However, if $|\mathbf{x}\rangle$ is not $|\mathbf{0}\rangle$ or $|\mathbf{1}\rangle$, we won't end up with two copies. We saw that if we input $\left(\frac{\mathbf{1}}{\sqrt{(\mathbf{2})}}|\mathbf{0}\rangle + \frac{\mathbf{1}}{\sqrt{(\mathbf{2})}}|\mathbf{1}\rangle\right)|\mathbf{0}\rangle$ to the CNOT gate, we ended up with entangled qubits, not two copies of the left qubit. In sumamry, we can use CNOT gate to copy classical bits, but not general qubits.

In the quantum computing domain, the term cloning is used in place of fan-out. We want a gate that takes as input a general qubit $|\mathbf{x}\rangle$ and ancilla bit, a fixed second input, as $|\mathbf{0}\rangle$ and output two copies of $|\mathbf{x}\rangle$. The desired gate looks like this.

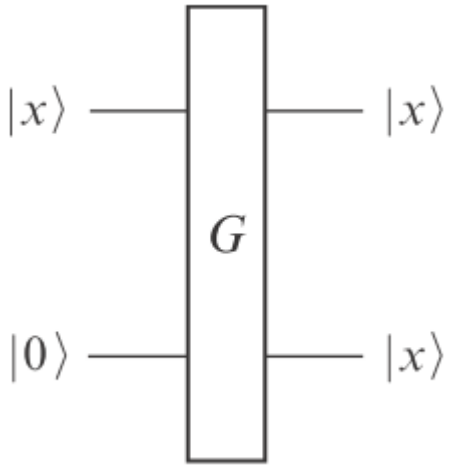

Figure 8.7: Cloning qubits

We will see that it is impossible to clone general qubits. The proof is as follows.

If **G** exists, the cloning properties give:

1. $\mathbf{G}\left(|\mathbf{0}\rangle|\mathbf{0}\rangle\right) = |\mathbf{0}\rangle|\mathbf{0}\rangle$

2. $\mathbf{G}\left(|\mathbf{1}\rangle|\mathbf{0}\rangle\right) = |\mathbf{1}\rangle|\mathbf{1}\rangle$

3. $\mathbf{G}\left(\left(\frac{\mathbf{1}}{\sqrt{\mathbf{2}}}|\mathbf{0}\rangle + \frac{\mathbf{1}}{\sqrt{\mathbf{2}}}|\mathbf{1}\rangle\right)|\mathbf{0}\rangle\right) = \left(\frac{\mathbf{1}}{\sqrt{\mathbf{2}}}|\mathbf{0}\rangle + \frac{\mathbf{1}}{\sqrt{\mathbf{2}}}|\mathbf{1}\rangle\right)\left(\frac{\mathbf{1}}{\sqrt{\mathbf{2}}}|\mathbf{0}\rangle + \frac{\mathbf{1}}{\sqrt{\mathbf{2}}}|\mathbf{1}\rangle\right)$

These three statements can be restated as follows:

1. $\mathbf{G}\left(|\mathbf{00}\rangle\right) = |\mathbf{00}\rangle$

2. $\mathbf{G}\left(|\mathbf{10}\rangle\right) = |\mathbf{11}\rangle$

3. $\mathbf{G}\left(\frac{\mathbf{1}}{\sqrt{\mathbf{2}}}|\mathbf{00}\rangle + \frac{\mathbf{1}}{\sqrt{\mathbf{2}}}|\mathbf{10}\rangle\right) = \frac{\mathbf{1}}{\mathbf{2}}\left(|\mathbf{00}\rangle + |\mathbf{01}\rangle + |\mathbf{10}\rangle + |\mathbf{11}\rangle\right)$

The gate **G**, like other matrix operators, must be linear. This means:

$$\mathbf{G}\left(\frac{\mathbf{1}}{\sqrt{\mathbf{2}}}|\mathbf{00}\rangle + \frac{\mathbf{1}}{\sqrt{\mathbf{2}}}|\mathbf{10}\rangle\right) = \frac{\mathbf{1}}{\sqrt{\mathbf{2}}}\mathbf{G}\left(|\mathbf{00}\rangle\right) + \frac{\mathbf{1}}{\sqrt{\mathbf{2}}}\mathbf{G}\left(|\mathbf{10}\rangle\right)$$

Using the equations (1) and (2) for definitions of $\mathbf{G}\left(|\mathbf{00}\rangle\right)$ and $\mathbf{G}\left(|\mathbf{10}\rangle\right)$, we get:

$$\mathbf{G}\left(\frac{\mathbf{1}}{\sqrt{\mathbf{2}}}|\mathbf{00}\rangle + \frac{\mathbf{1}}{\sqrt{\mathbf{2}}}|\mathbf{10}\rangle\right) = \frac{\mathbf{1}}{\sqrt{\mathbf{2}}}|\mathbf{00}\rangle + \frac{\mathbf{1}}{\sqrt{\mathbf{2}}}|\mathbf{10}\rangle$$

As per equation (3)we have:

$$\mathbf{G}\left(\frac{\mathbf{1}}{\sqrt{\mathbf{2}}}|\mathbf{00}\rangle + \frac{\mathbf{1}}{\sqrt{\mathbf{2}}}|\mathbf{10}\rangle\right) = \frac{\mathbf{1}}{\mathbf{2}}(|\mathbf{00}\rangle + |\mathbf{01}\rangle + |\mathbf{10}\rangle + |\mathbf{11}\rangle)$$

However, we know that $\frac{\mathbf{1}}{\sqrt{\mathbf{2}}}|\mathbf{00}\rangle + \frac{\mathbf{1}}{\sqrt{\mathbf{2}}}|\mathbf{10}\rangle \neq \frac{\mathbf{1}}{\mathbf{2}}(|\mathbf{00}\rangle + |\mathbf{01}\rangle + |\mathbf{10}\rangle + |\mathbf{11}\rangle)$

We have arrived at a contradiction now and this proves that $\mathbf{G}$ cannot exist.

Q.E.D.

We see that it is impossible to construct a gate that clones general qubits. We used $|\mathbf{0}\rangle$ as an ancilla qubit. There is nothing special about this and the exact same argument can be used for whatever value is chosen for this bit.

The inability to clone a qubit has important consequences. We often take backup of files and send copies of files to other users. Copying is ubiquitous. Every day computers are based on von Neumann architecture, which is heavily based on the ability to copy. When we run a program we are always copying bits from one place to another. In quantum computing, copying is not possible for general qubits. So, if programmable quantum computers are designed they will not be based on our current architecture.

On the other hand, often we want to prevent copying. We want to secure our data, we don't want our communications to be intercepted. Here, as we saw with Eve, the fact that we cannot clone qubits can be used to our advantage, preventing unwanted copies from being made.

We know that the qubits $|\mathbf{0}\rangle$ and $|\mathbf{1}\rangle$ correspond to classical bits **0** and **1**. If we run the quantum CNOT gate with qubits $|\mathbf{0}\rangle$ and $|\mathbf{1}\rangle$, and do not use any superposition qubits, then we find that the result is the same as running the classical CNOT gate with bits **0** and **1**. The same result applies to the quantum Fredkin gate. Since the classical Fredkin gate is universal and the quantum Fredkin gate using only qubits $|\mathbf{0}\rangle$ and $|\mathbf{1}\rangle$ is equivalent to the classical gate, we can conclude that a quantum circuit can compute anything that can be calculated by the classical circuit. The inability to clone general qubits sounds worrisome, but it doesn't restrict us from doing classical computations in any way. This further means that Quantum computation includes all of the classical computation and is a more general form of computation.

Next, let us see how we can make quantum circuits using the gates that we have seen so far.

8.5 The Bell Circuit

We define the quantum Bell Circuit as follows.

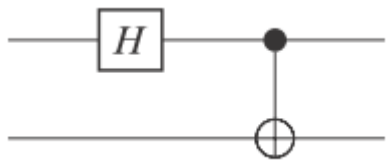

Figure 8.8: Bell Circuit

To understand the workings of the Bell circuit, we will input the four pair of qubits that are part of standard basis. We start with qubit $|\mathbf{00}\rangle = |\mathbf{0}\rangle\,|\mathbf{0}\rangle$. The first qubit is acted upon by Hadamard gate and is changed to $\frac{\mathbf{1}}{\sqrt{\mathbf{2}}}\,|\mathbf{0}\rangle + \frac{\mathbf{1}}{\sqrt{\mathbf{2}}}\,|\mathbf{1}\rangle$. At this moment, the system of two qubits has the state $\left(\frac{\mathbf{1}}{\sqrt{\mathbf{2}}}\,|\mathbf{0}\rangle + \frac{\mathbf{1}}{\sqrt{\mathbf{2}}}\,|\mathbf{1}\rangle\right)|\mathbf{0}\rangle = \frac{\mathbf{1}}{\sqrt{\mathbf{2}}}\,|\mathbf{00}\rangle + \frac{\mathbf{1}}{\sqrt{\mathbf{2}}}\,|\mathbf{10}\rangle$. Next we apply the CNOT gate which changes $|\mathbf{10}\rangle$ to $|\mathbf{11}\rangle$ giving the final state as $\frac{\mathbf{1}}{\sqrt{\mathbf{2}}}\,|\mathbf{00}\rangle + \frac{\mathbf{1}}{\sqrt{\mathbf{2}}}\,|\mathbf{11}\rangle$. This is represented as follows:

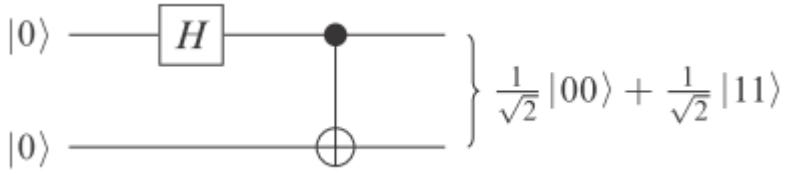

Figure 8.9: Bell Circuit Acting on Qubits

We can summarize the Bell circuit output as follows.

$$\mathbf{B}\left(|\mathbf{00}\rangle\right) = \frac{\mathbf{1}}{\sqrt{\mathbf{2}}}\,|\mathbf{00}\rangle + \frac{\mathbf{1}}{\sqrt{\mathbf{2}}}\,|\mathbf{11}\rangle$$

$$\mathbf{B}\left(|\mathbf{01}\rangle\right) = \frac{\mathbf{1}}{\sqrt{\mathbf{2}}}\,|\mathbf{01}\rangle + \frac{\mathbf{1}}{\sqrt{\mathbf{2}}}\,|\mathbf{10}\rangle$$

$$\mathbf{B}\left(|\mathbf{10}\rangle\right) = \frac{\mathbf{1}}{\sqrt{\mathbf{2}}}\,|\mathbf{00}\rangle - \frac{\mathbf{1}}{\sqrt{\mathbf{2}}}\,|\mathbf{11}\rangle$$

$$\mathbf{B}\left(|\mathbf{11}\rangle\right) = \frac{\mathbf{1}}{\sqrt{\mathbf{2}}}|\mathbf{01}\rangle - \frac{\mathbf{1}}{\sqrt{\mathbf{2}}}|\mathbf{10}\rangle$$

Each of these outputs is entangled. Since the inputs form an orthonormal basis in $\mathbb{R}^4$, the output must also form an orthonormal basis. This output basis consisting of the four entangled kets is called the **Bell Basis**.

We know that to check for orthogonality of a square matrix $\mathbf{A}$, one way is to calculate $\mathbf{A}\mathbf{A}^\top$ where $\mathbf{A}^\top$ is the transpose matrix. If the result is the identity matrix $\mathbf{I}$, then we say that the matrix $\mathbf{A}$ is orthogonal and the columns of the matrix give us an orthonormal basis. We have defined our gates to be orthogonal, and they all have this property. All the gates introduced earlier, except the Pauli matrix $\mathbf{Y}$, have the property that when we take the transpose matrix, we end with the same matrix that we started with. This means, for these gates $\mathbf{A}\mathbf{A}^\top = \mathbf{I}$, and further, if we apply the fate twice in a row we end up with output that is unchanged from the input.

We also know that the Hadamard gate and CNOT gate are their inverses. Let us consider the following circuit.

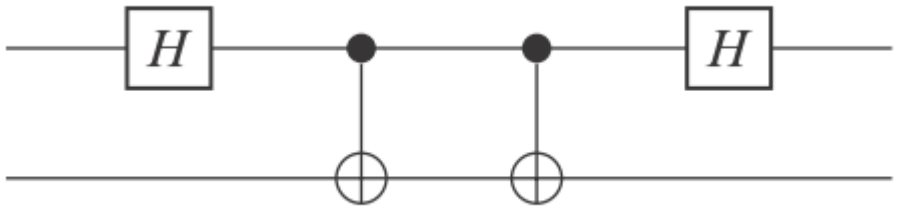

Figure 8.10: Hadamard and CNOT gate

When a paid or qubits is sent through this circuit, first the Hadamard gate is applied and then the CNOT gate is applied to them. The action of the first CNOT gate is undone by the second CNOT gate, and the second Hadamard gate undoes the action of the first Hadamard gate. This means that this circuit doesn't change anything, and then output qubits are the same as input qubits.

This means that the following circuit, called Reverse Bell Circuit, denoted by **RB** undoes the action of the Bell Circuit.

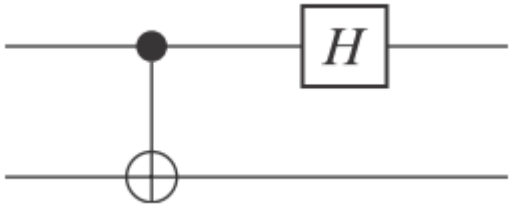

Figure 8.11: Reverse bell Circuit

If we input vectors from Bell basis to this circuit, it will output the vectors in the standard basis. More formally, it is summarized as follows.

$$\mathbf{RB}\left(\frac{\mathbf{1}}{\sqrt{\mathbf{2}}}|\mathbf{00}\rangle + \frac{\mathbf{1}}{\sqrt{\mathbf{2}}}|\mathbf{11}\rangle\right) = |\mathbf{00}\rangle$$

$$\mathbf{RB}\left(\frac{\mathbf{1}}{\sqrt{\mathbf{2}}}|\mathbf{01}\rangle + \frac{\mathbf{1}}{\sqrt{\mathbf{2}}}|\mathbf{10}\rangle\right) = |\mathbf{01}\rangle$$

$$\mathbf{RB}\left(\frac{\mathbf{1}}{\sqrt{\mathbf{2}}}|\mathbf{00}\rangle - \frac{\mathbf{1}}{\sqrt{\mathbf{2}}}|\mathbf{11}\rangle\right) = |\mathbf{10}\rangle$$

$$\mathbf{RB}\left(\frac{\mathbf{1}}{\sqrt{\mathbf{2}}}|\mathbf{01}\rangle - \frac{\mathbf{1}}{\sqrt{\mathbf{2}}}|\mathbf{10}\rangle\right) = |\mathbf{11}\rangle$$

We next look at two interesting applications, superdense coding, and quantum teleportation, of ideas introduced so far.

8.6 Superdense Coding

Superdense coding is a quantum communication protocol that allows a sender to send two classical bits of information to another user by only utilizing one qubit. Superdense coding was first proposed by Charles Bennett and Stephen Wiesner in 1992 and experimentally actualized four years later, in 1996.

Superdense Coding and Quantum Teleportation are closely related. Quantum teleportation is a process by which a user can transmit one qubit using two classical bits whereas Superdense Coding is a process by which a user can transmit two classical bits using one qubit. In short, Superdense Coding can be thought of as the flipped version of Quantum Teleportation.

The initial conditions for both superdense coding and quantum teleportation are the same. There are two electrons in entangled state $\frac{\mathbf{1}}{\sqrt{\mathbf{2}}}|\mathbf{00}\rangle + \frac{\mathbf{1}}{\sqrt{\mathbf{2}}}|\mathbf{11}\rangle$, one of which is given to Alice and other to Bob. Both of them then travel far apart, being careful not to make any measurements of their respective electrons, thereby preserving the entangled state.

In the case of superdense coding, Alice wants to send Bob two classical bits of information. She wants to send one of the four from $\mathbf{00}, \mathbf{01}, \mathbf{10}, \mathbf{11}$. She will achieve this by sending Bob one qubit, her electron. The solution seems easy and straightforward. Alice will send Bob the qubit $\mathbf{a_0}|\mathbf{0}\rangle + \mathbf{a_1}|\mathbf{1}\rangle$. There are infinitely many ways to select this qubit, since any qubit satisfying the condition $\mathbf{a_0}^{\mathbf{2}} + \mathbf{a_1}^{\mathbf{2}} = \mathbf{1}$ will be sufficient. Surely, it must be easy to construct a way of transmitting two bits of information — one out of four possibilities — if you are allowed to send something that can be one of an infinite number of things.

The problem with his solution is that Bob can never know what the qubit is. He can get information only by measurement. Once Bob measures the spin in the standard basis, he will either get $|\mathbf{0}\rangle$ or $|\mathbf{1}\rangle$. If Alice has sent $\mathbf{a_0}|\mathbf{0}\rangle + \mathbf{a_1}|\mathbf{1}\rangle$, he will get $|\mathbf{0}\rangle$ with probability $\mathbf{a_0}^{\mathbf{2}}$ and $|\mathbf{1}\rangle$ with probability $\mathbf{a_1}^{\mathbf{2}}$. If Bob gets $|\mathbf{0}\rangle$, he has no information about $\mathbf{a_0}$ except that it is non-zero. Bob can get at most one bit of information from each qubit. To get two bits of information he will have to extract one bit from the particle that Alice is sending him, but he must also extract one bit from the particle in his possession.

At the start of the experiment, Alice and Bob had one electron each. In the end, Bob will have both the electrons and will measure their spins. We image Bob having a quantum circuit with two wires exiting on the right. If Alice wants to send $\mathbf{00}$, the circuit should be arranged such that just before Bob starts to measure, the top electron is in state $|\mathbf{0}\rangle$, and the bottom electron is in state $|\mathbf{0}\rangle$. This means that we want the pair of electrons to be in unentangled state $|\mathbf{00}\rangle$ just before Bob measures their spins. Continuing with similar logic, we want the pair of electrons to be in state $|\mathbf{01}\rangle$ if Alice is sending $\mathbf{01}$, in state $|\mathbf{10}\rangle$ if she is sending $\mathbf{10}$, and in state $|\mathbf{11}\rangle$ if Alice wants to send $\mathbf{11}$. Lastly, Bob must do the same thing to every pair of electrons that he receives. He cannot do different things depending on what Alice is trying to send, because he doesn't know what she is trying to send.

We start by letting Alice act on her electron in one of the four ways. Each of the actions will result in qubits in one of the Bell basis vectors. Bob will then simply run the pair of qubits through the reverse Bell circuit and get the unentangled state.

Alice uses four quantum circuits, one for each of the two-bit choices. Each circuit uses a Pauli gate and they are shown below.

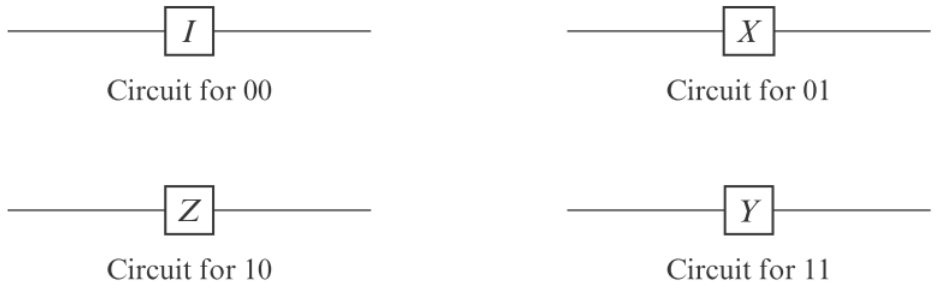

Figure 8.12: Gates for Superdense Coding

Let us see what happens to the qubits in each of these cases. Initially, Alice and Bob's qubits are entangled and are in the state $\frac{\mathbf{1}}{\sqrt{\mathbf{2}}}|\mathbf{00}\rangle + \frac{\mathbf{1}}{\sqrt{\mathbf{2}}}|\mathbf{11}\rangle$. This can be rewritten as $\frac{\mathbf{1}}{\sqrt{\mathbf{2}}}|\mathbf{0}\rangle \otimes |\mathbf{0}\rangle + \frac{\mathbf{1}}{\sqrt{\mathbf{2}}}|\mathbf{1}\rangle \otimes |\mathbf{1}\rangle$.

Alice will send her electron through her circuits which may change her kets. It is important to remember that Alice's circuits do not affect Bob's electron in any way. Let us see all four cases one by one.

If Alice wants to send **00**, she applies **I** which doesnt change anything. The qubits remain in the state $\frac{\mathbf{1}}{\sqrt{\mathbf{2}}}|\mathbf{00}\rangle + \frac{\mathbf{1}}{\sqrt{\mathbf{2}}}|\mathbf{11}\rangle$.

If Alice wants to send **01**, she applies **X** which interchanges her $|\mathbf{0}\rangle$ and $|\mathbf{1}\rangle$. The new resultant state will be $\frac{\mathbf{1}}{\sqrt{\mathbf{2}}}|\mathbf{1}\rangle \otimes |\mathbf{0}\rangle + \frac{\mathbf{1}}{\sqrt{\mathbf{2}}}|\mathbf{0}\rangle \otimes |\mathbf{1}\rangle$ which can be rewritten as $\frac{\mathbf{1}}{\sqrt{\mathbf{2}}}|\mathbf{10}\rangle + \frac{\mathbf{1}}{\sqrt{\mathbf{2}}}|\mathbf{01}\rangle$

If Alice wants to send **10**, she applies **Z** which leaves $|\mathbf{0}\rangle$ alone, but interchanges her $|\mathbf{1}\rangle$ to $-|\mathbf{1}\rangle$. The new resultant state will be $\frac{\mathbf{1}}{\sqrt{\mathbf{2}}}|\mathbf{0}\rangle \otimes |\mathbf{0}\rangle + \frac{\mathbf{1}}{\sqrt{\mathbf{2}}}(|-\mathbf{1}\rangle) \otimes |\mathbf{1}\rangle$ which can be rewritten as $\frac{\mathbf{1}}{\sqrt{\mathbf{2}}}|\mathbf{00}\rangle - \frac{\mathbf{1}}{\sqrt{\mathbf{2}}}|\mathbf{11}\rangle$.

If Alice wants to send **11**, she applies **Y** and the qubits ends up in the $\frac{\mathbf{1}}{\sqrt{\mathbf{2}}}|\mathbf{10}\rangle - \frac{\mathbf{1}}{\sqrt{\mathbf{2}}}|\mathbf{01}\rangle$ entangled state.

We see that each of the output state vectors is a distinct Bell basis vector. Alice now sends her electron to Bob, and once Bob receives it, he uses a Reverse Bell

Circuit that inputs both the qubit that Alice has sent and the one that was always in his possession.

If Alice was sending **00**, Bob receives the qubits in the state $\frac{\mathbf{1}}{\sqrt{\mathbf{2}}}|\mathbf{00}\rangle + \frac{\mathbf{1}}{\sqrt{\mathbf{2}}}|\mathbf{11}\rangle$. He sends this through the reverse Bell circuit which changes the state to $|\mathbf{00}\rangle$ and unentangles them. The top and bottom bits are both $|\mathbf{0}\rangle$. Bob measures the qubits and get reads **00**.

If Alice was sending **01**, Bob receives the qubits in the state $\frac{\mathbf{1}}{\sqrt{\mathbf{2}}}|\mathbf{10}\rangle + \frac{\mathbf{1}}{\sqrt{\mathbf{2}}}|\mathbf{01}\rangle$. He sends this through the reverse Bell circuit which changes the state to $|\mathbf{01}\rangle$ and unentangles them. The top bit is $|\mathbf{0}\rangle$ and bottom bit is $|\mathbf{1}\rangle$. Bob measures the qubits and get reads **01**.

The same logic applies to the case where Alice sends **10** and **11**. In all the cases, Bob ends up with two bits that Alice wants to send to him.

8.7 Quantum Teleportation

Quantum teleportation is a process by which a user can transmit one qubit using two classical bits. It was first proposed by C. H. Bennett, G. Brassard, C. Crépeau, R. Jozsa, A. Peres, and W. K. Wootters in 1993 and experimentally realized in 1997 by two research groups, led by Sandu Popescu and Anton Zeilinger.

As in the case of superdense coding, Alice and Bob each have one electron and are far apart. Thir electrons share the entangled state $\frac{\mathbf{1}}{\sqrt{\mathbf{2}}}|\mathbf{00}\rangle + \frac{\mathbf{1}}{\sqrt{\mathbf{2}}}|\mathbf{11}\rangle$. Apart from this, Alice has another electron in the state $\mathbf{a}|\mathbf{0}\rangle + \mathbf{b}|\mathbf{1}\rangle$. Alice doesn't know the values of probability amplitudes **a** and **b**, but she and Bob want to change Bob's electron such that it has state $\mathbf{a}|\mathbf{0}\rangle + \mathbf{b}|\mathbf{1}\rangle$. In short, they want to teleport the state of Alice's electron to Bob's. We will see that it can be achieved by Alice sending two classical bits to Bob. There are infinitely many possibilities for the initial state of Alice's electron and we will see that we can send one of these infinite numbers of possibilities using only two classical bits. We should also remember that Alice starts with a qubit and Bob ends up with it, and neither of them can ever know what it is. If they want to learn about the qubit, they should make a measurement. Once they do, they will either get $|\mathbf{0}\rangle$ or $|\mathbf{1}\rangle$.

Let us see how it can work. At the end of quantum teleportation, we know that Bob ends with an electron the unentangled state $\mathbf{a}|\mathbf{0}\rangle + \mathbf{b}|\mathbf{1}\rangle$. At the start, both Alice and Bob share the entangled state and to dis-entangle, one of them

should make a measurement. Its obvious that it cannot be Bob since if he makes a measurement, he will end with an electron in either $|\mathbf{0}\rangle$ or $|\mathbf{1}\rangle$, not the desired $\mathbf{a}\,|\mathbf{0}\rangle + \mathbf{b}\,|\mathbf{1}\rangle$. So, Alice will be making the measurement and we will get the third electron's state somehow involved. Alice will somehow entangle the state of this electron with the state of her other electron that is already entangled with Bob's. Alice can achieve this is by sending two qubits that she controls through a CNOT gate, and then she applies the Hadamard gate to the top qubit. In other words, she passes the two qubits that she controls through a reverse Bell Circut. The next figure shows the setup where Alice's qubits are shown above Bob's qubits, and the second and third rows depict the entangled qubits.

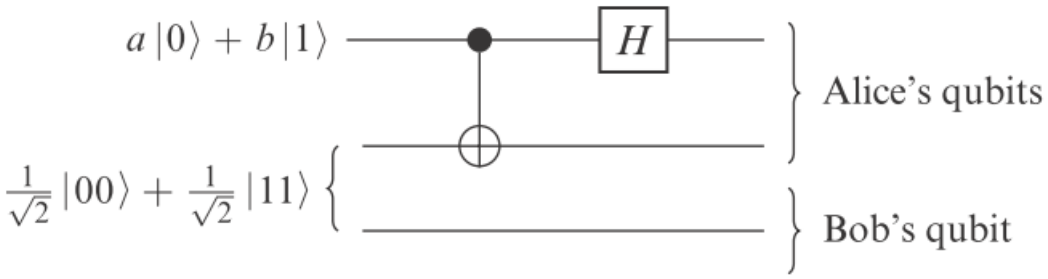

Figure 8.13: Setup for Quantum Teleportation

We have three qubits, the initial state that describes the three electrons is $(\mathbf{a}\,|\mathbf{0}\rangle + \mathbf{b}\,|\mathbf{1}\rangle) \otimes \left(\frac{\mathbf{1}}{\sqrt{\mathbf{2}}}\,|\mathbf{00}\rangle + \frac{\mathbf{1}}{\sqrt{\mathbf{2}}}\,|\mathbf{11}\rangle\right)$, which we can write as:

$\frac{\mathbf{a}}{\sqrt{\mathbf{2}}}\,|\mathbf{000}\rangle + \frac{\mathbf{a}}{\sqrt{\mathbf{2}}}\,|\mathbf{011}\rangle + \frac{\mathbf{b}}{\sqrt{\mathbf{2}}}\,|\mathbf{100}\rangle + \frac{\mathbf{b}}{\sqrt{\mathbf{2}}}\,|\mathbf{111}\rangle$. Since it is Alice that is acting on the qubits, we can rewrite it to emphasize the fact as:

$$\frac{\mathbf{a}}{\sqrt{\mathbf{2}}}\,|\mathbf{00}\rangle \otimes |\mathbf{0}\rangle + \frac{\mathbf{a}}{\sqrt{\mathbf{2}}}\,|\mathbf{01}\rangle \otimes |\mathbf{1}\rangle + \frac{\mathbf{b}}{\sqrt{\mathbf{2}}}\,|\mathbf{10}\rangle \otimes |\mathbf{0}\rangle + \frac{\mathbf{b}}{\sqrt{\mathbf{2}}}\,|\mathbf{11}\rangle \otimes |\mathbf{1}\rangle.$$

Alice will apply the reverse Bell circuit which we can consider as applying the CNOT gate to the first two qubits and then the Hadamard gate to the top bit. Applying the CNOT gate gives:

$$\frac{\mathbf{a}}{\sqrt{\mathbf{2}}}\,|\mathbf{00}\rangle \otimes |\mathbf{0}\rangle + \frac{\mathbf{a}}{\sqrt{\mathbf{2}}}\,|\mathbf{01}\rangle \otimes |\mathbf{1}\rangle + \frac{\mathbf{b}}{\sqrt{\mathbf{2}}}\,|\mathbf{11}\rangle \otimes |\mathbf{0}\rangle + \frac{\mathbf{b}}{\sqrt{\mathbf{2}}}\,|\mathbf{10}\rangle \otimes |\mathbf{1}\rangle$$

Alice will now act on the first qubit, so we rewrite it to emphasize this:

$$\frac{\mathbf{a}}{\sqrt{\mathbf{2}}}|\mathbf{0}\rangle \otimes |\mathbf{0}\rangle \otimes |\mathbf{0}\rangle + \frac{\mathbf{a}}{\sqrt{\mathbf{2}}}|\mathbf{0}\rangle \otimes |\mathbf{1}\rangle \otimes |\mathbf{1}\rangle + \frac{\mathbf{b}}{\sqrt{\mathbf{2}}}|\mathbf{1}\rangle \otimes |\mathbf{1}\rangle \otimes |\mathbf{0}\rangle + \frac{\mathbf{b}}{\sqrt{\mathbf{2}}}|\mathbf{1}\rangle \otimes |\mathbf{0}\rangle \otimes |\mathbf{1}\rangle$$

The Hadamard gate changes $|\mathbf{0}\rangle$ to $\frac{\mathbf{1}}{\sqrt{\mathbf{2}}}|\mathbf{0}\rangle + \frac{\mathbf{1}}{\sqrt{\mathbf{2}}}|\mathbf{1}\rangle$ and $|\mathbf{1}\rangle$ to $\frac{\mathbf{1}}{\sqrt{\mathbf{2}}}|\mathbf{0}\rangle - \frac{\mathbf{1}}{\sqrt{\mathbf{2}}}|\mathbf{1}\rangle$ and it results in state:

$$\begin{aligned}\frac{\mathbf{a}}{\mathbf{2}}|\mathbf{0}\rangle \otimes |\mathbf{0}\rangle \otimes |\mathbf{0}\rangle + \frac{\mathbf{a}}{\mathbf{2}}|\mathbf{1}\rangle \otimes |\mathbf{0}\rangle \otimes |\mathbf{0}\rangle + \\ \frac{\mathbf{a}}{\mathbf{2}}|\mathbf{0}\rangle \otimes |\mathbf{1}\rangle \otimes |\mathbf{1}\rangle + \frac{\mathbf{a}}{\mathbf{2}}|\mathbf{1}\rangle \otimes |\mathbf{1}\rangle \otimes |\mathbf{1}\rangle + \\ \frac{\mathbf{b}}{\mathbf{2}}|\mathbf{0}\rangle \otimes |\mathbf{1}\rangle \otimes |\mathbf{0}\rangle - \frac{\mathbf{b}}{\mathbf{2}}|\mathbf{1}\rangle \otimes |\mathbf{1}\rangle \otimes |\mathbf{0}\rangle + \\ \frac{\mathbf{b}}{\mathbf{2}}|\mathbf{0}\rangle \otimes |\mathbf{0}\rangle \otimes |\mathbf{1}\rangle - \frac{\mathbf{b}}{\mathbf{2}}|\mathbf{1}\rangle \otimes |\mathbf{0}\rangle \otimes |\mathbf{1}\rangle\end{aligned}$$

This can be further simplified to:

$$\begin{aligned}\frac{\mathbf{1}}{\mathbf{2}}|\mathbf{00}\rangle \otimes (\mathbf{a}|\mathbf{0}\rangle + \mathbf{b}|\mathbf{1}\rangle) + \\ \frac{\mathbf{1}}{\mathbf{2}}|\mathbf{01}\rangle \otimes (\mathbf{a}|\mathbf{1}\rangle + \mathbf{b}|\mathbf{0}\rangle) + \\ \frac{\mathbf{1}}{\mathbf{2}}|\mathbf{10}\rangle \otimes (\mathbf{a}|\mathbf{0}\rangle - \mathbf{b}|\mathbf{1}\rangle) + \\ \frac{\mathbf{1}}{\mathbf{2}}|\mathbf{11}\rangle \otimes (\mathbf{a}|\mathbf{1}\rangle - \mathbf{b}|\mathbf{0}\rangle)\end{aligned}$$

Alice now measures her two electrons in the standard basis. She will get one of $|\mathbf{00}\rangle, |\mathbf{01}\rangle, |\mathbf{10}\rangle, |\mathbf{11}\rangle$ each with a probability of $\frac{1}{4}$. We note:

If Alice gets $|\mathbf{00}\rangle$, Bob's qubit will jump to state $\mathbf{a}|\mathbf{0}\rangle + \mathbf{b}|\mathbf{1}\rangle$.

If Alice gets $|\mathbf{01}\rangle$, Bob's qubit will jump to state $\mathbf{a}|\mathbf{1}\rangle + \mathbf{b}|\mathbf{0}\rangle$.

If Alice gets $|\mathbf{10}\rangle$, Bob's qubit will jump to state $\mathbf{a}|\mathbf{0}\rangle - \mathbf{b}|\mathbf{1}\rangle$.

If Alice gets $|\mathbf{11}\rangle$, Bob's qubit will jump to state $\mathbf{a}|\mathbf{1}\rangle - \mathbf{b}|\mathbf{0}\rangle$.

Alice and Bob want Bob's qubit to be in the state $\mathbf{a}|\mathbf{0}\rangle + \mathbf{b}|\mathbf{1}\rangle$. We are almost there. To bridge the gap, Alice will let Bob know which of the four possible situations he is in. She sends Bob two classical bits of information, $\mathbf{00}, \mathbf{01}, \mathbf{10}, \mathbf{11}$, corresponding to the results of her measurements, to let him know. This information can be sent in any way, by plain text, for example. We now have:

If Bob receives **00**, he knows that his qubit is in the correct state and he does nothing.

If Bob receives **01**, he knows that his qubit is in $\mathbf{a}\,|\mathbf{1}\rangle + \mathbf{b}\,|\mathbf{0}\rangle$ state. He applies gate **X**.

If Bob receives **10**, he knows that his qubit is in $\mathbf{a}\,|\mathbf{0}\rangle - \mathbf{b}\,|\mathbf{1}\rangle$ state. He applies gate **Z**.

If Bob receives **11**, he knows that his qubit is in $\mathbf{a}\,|\mathbf{1}\rangle - \mathbf{b}\,|\mathbf{0}\rangle$ state. He applies gate **Y**.

We can see that now in every case, Bob's qubit will end up in the state $\mathbf{a}\,|\mathbf{0}\rangle + \mathbf{b}\,|\mathbf{1}\rangle$, the original state of the qubit that Alice wanted to teleport!

During this entire process, there is only one qubit in the state $\mathbf{a}\,|\mathbf{0}\rangle + \mathbf{b}\,|\mathbf{1}\rangle$. Initially, Alice has it. In the end, Bob has it, but as the no-cloning theorem tells us, we can't copy, so only one of them can have it at any time. When Alice sends her qubit through her circuit, Bob's qubit instantaneously jumps to one of the four states. Bob still has to wait for Alice to send the two classical bits before he can decide which of the four qubits correspond to Alice's original qubits. The fact that two bits have to be sent by some conventional transport mechanism, clearly shows that instantaneous transmission of information is not possible.

Superdense Coding can be thought of as the flipped version of Quantum Teleportation. Quantum teleportation is a process by which a user can transmit one qubit using two classical bits whereas Superdense Coding is a process by which a user can transmit two classical bits using one qubit. In the case of superdense coding, Alice encodes using the Pauli transformations, and Bob decodes using the reverse Bell circuit. In the case of quantum teleportation, Alice encodes using the reverse Bell circuit, and Bob decodes using the Pauli transformations.

Quantum teleportation is actually being performed, usually using entangled photons rather than entangled electrons, where it can be done over substantial distances. The longest distance of successful teleportation is about 1400 km by the group of JianWei Pan using the Micius satellite for space-based quantum teleportation.

Qubits tend to interact with the environment and get corrupted and quantum teleportation can be used to correct such errors. We will next look at a simple example of this.

8.8 Classical Error Correction (1 bit only)

The two basic tenets of encoding digital information are compression and error correction. First, We want to eliminate redundancy and compress the information as much as possible to make the message as short as possible (think zipping a file). Secondly, we want to add some redundancy that will help us correct errors. There are so many ways that a message is slightly corrupted, and given a slightly corrupted message, we should be able to correct it.

Error correction is essential for transmissions involving qubits. We are using photons and electrons to encode them. These particles can interact with the rest of the universe and unwanted interactions may change the states of some qubits. We next look at the most basic classical error correcting code and modify it to support qubits.

A simple error correcting code is to repeat three times the symbol that we want to send. If Alice wants to send **0**, she will send **000**. If she wants to send **1**, she will send **111**. If Bob keeps on receiving a sequence of three **0**'s and three **1**'s, he knows that all is well. On the other hand, if he receives something else, say **101**, he knows that an error has occurred. If Alice has sent **000**, then two errors have occurred. If Alice has sent **111**, then only one error has occurred. If errors are fairly unlikely, it is more probable that one error, rather than two errors, has occurred. In this case, Bob assumes that the least number of errors have occurred and replaces **101** with **111**.

There are eight three-bit strings that Bob could receive. Four of these, $(\mathbf{000}, \mathbf{001}, \mathbf{010}, \mathbf{100})$, will all be decoded as **000** by Bob. The other four, $(\mathbf{111}, \mathbf{110}, \mathbf{101}, \mathbf{011})$, will all be decoded as **111** by Bob. If the chances of errors are very small, this scheme corrects many errors and reduces the overall error rate. Can we use the same method for qubits? We should remember that to read the qubits, we need to measure them, and once read, they move to a new state. We will next see how Bob can use the parity test to overcome this challenge with qubits.

Let Bob receive the qubits as $\mathbf{b_0 b_1 b_2}$. He then computes $\mathbf{b_0} \oplus \mathbf{b_1}$ and $\mathbf{b_0} \oplus \mathbf{b_2}$. The first sum checks the parity of the first two bits —that is, it checks whether they are the same digit or not. The second sum performs a parity check on the first and third digits.

If all the three bits are the same, either all **0** or all **1**, Bob will get **0** for both the sums that he has computed. If all the bits are not the same, then two will be equal and the third one will be different. It will be this third symbol that needs to be flipped from **0** to **1**, or from **1** to **0**. The algorithm flows as follows.

START

If $\mathbf{b_0} = \mathbf{b_1} \neq \mathbf{b_2}$, then $\mathbf{b_0} \oplus \mathbf{b_1} = \mathbf{0}$ and $\mathbf{b_0} \oplus \mathbf{b_2} = \mathbf{1}$.

If $\mathbf{b_0} = \mathbf{b_2} \neq \mathbf{b_1}$, then $\mathbf{b_0} \oplus \mathbf{b_1} = \mathbf{1}$ and $\mathbf{b_0} \oplus \mathbf{b_2} = \mathbf{0}$.

If $\mathbf{b_0} \neq \mathbf{b_1} = \mathbf{b_2}$, then $\mathbf{b_0} \oplus \mathbf{b_1} = \mathbf{1}$ and $\mathbf{b_0} \oplus \mathbf{b_2} = \mathbf{1}$.

Bob looks at pair of bits $\mathbf{b_0} \oplus \mathbf{b_1}$ and $\mathbf{b_0} \oplus \mathbf{b_2}$.

If he observes **00**, he does nothing as there is nothing to correct.

If he observes **01**, he flips $\mathbf{b_2}$.

If he observes **10**, he flips $\mathbf{b_1}$.

If he observes **11**, he flips $\mathbf{b_0}$.

END

There is one more small point to observe. Suppose Bob receives a string and there is an error in the first bit. This means he has received either **100** or **011**. After Bob does the parity tests, he will get **11** for both the strings and will know that there is an error in the first bit. The important thing to note here is that the parity test tells us where the error is, they do not tell us whether it is a **0** that needs to be flipped to **1**, or a **1** that needs to be flipped to **0**.

We will now see how this error-correcting idea can be modified to support quits.

8.9 Quantum Error Correction (1 bit only)

We assume that Alice wants to send the qubit $\mathbf{a}|\mathbf{0}\rangle + \mathbf{b}|\mathbf{1}\rangle$ to Bob. We will restrict ourselves to the cases where the error causes the bits to flip, that is $\mathbf{a}|\mathbf{0}\rangle + \mathbf{b}|\mathbf{1}\rangle$ gets changed to $\mathbf{a}|\mathbf{1}\rangle + \mathbf{b}|\mathbf{0}\rangle$. Alice now wants to send three copies of her qubit to Bob, but the no-cloning theorem tells us that she cannot make copies of her qubit. However, she can use the following circuit consisting of two CNOT gates to replace her $|0\rangle$ with $|000\rangle$ and $|1\rangle$ with $|111\rangle$.

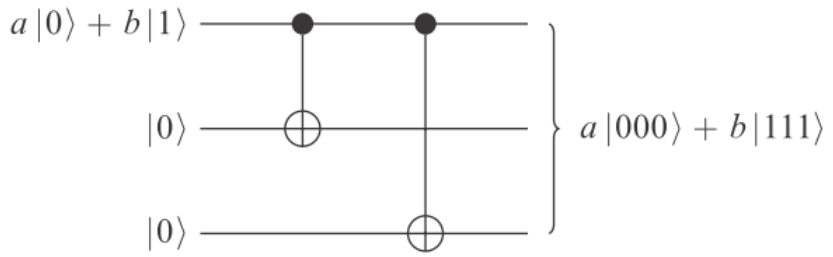

Figure 8.14: Copying Qubits

Alice starts with three qubits consisting of the qubit that she wants to encode and two ancilla bits both of which are $|0\rangle$. The initial state is then $(\mathbf{a}\,|\mathbf{0}\rangle + \mathbf{b}\,|\mathbf{1}\rangle)\,|\mathbf{0}\rangle\,|\mathbf{0}\rangle = \mathbf{a}\,|\mathbf{0}\rangle\,|\mathbf{0}\rangle\,|\mathbf{0}\rangle + \mathbf{b}\,|\mathbf{1}\rangle\,|\mathbf{0}\rangle\,|\mathbf{0}\rangle$.

The application of first CNOT gate changes it to $\mathbf{a}\,|\mathbf{0}\rangle\,|\mathbf{0}\rangle\,|\mathbf{0}\rangle + \mathbf{b}\,|\mathbf{1}\rangle\,|\mathbf{1}\rangle\,|\mathbf{0}\rangle$ and the second CNOT gate changes it to $\mathbf{a}\,|\mathbf{0}\rangle\,|\mathbf{0}\rangle\,|\mathbf{0}\rangle + \mathbf{b}\,|\mathbf{1}\rangle\,|\mathbf{1}\rangle\,|\mathbf{1}\rangle$, the desired state.

Alice now sends the qubits to Bob. Since the channel is noisy, there is a chance at one of the qubits will get flipped. This means, Bob might receive the correct qubits $\mathbf{a}\,|\mathbf{000}\rangle + \mathbf{b}\,|\mathbf{111}\rangle$, or he might receive one of the following incorrect versions $\mathbf{a}\,|\mathbf{100}\rangle + \mathbf{b}\,|\mathbf{011}\rangle$, $\mathbf{a}\,|\mathbf{010}\rangle + \mathbf{b}\,|\mathbf{101}\rangle$, or $\mathbf{a}\,|\mathbf{001}\rangle + \mathbf{b}\,|\mathbf{110}\rangle$, corresponding to error in first, second, and third qubit respectively. We should always remember that Bob cannot measure the entangled state at any time. If he does, the state becomes unentangled and he will get three qubits that are some combination of $|\mathbf{0}\rangle$s and $|\mathbf{1}\rangle$s, and the values of $\mathbf{a}$ and $\mathbf{b}$ are lost forever.

Let us continue to explore how Bob can determine which bit is flipped, and correct it without measuring any of the three qubits that were sent to him by Alice. He will utilize the parity check idea, the same idea that we saw in action with classical qubits.

Bob will use the following circuit for doing parity checks.

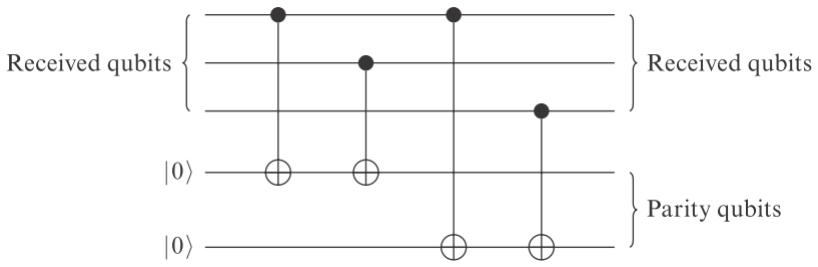

Figure 8.15: Bob's Partity Check Circuit

The circuit uses four CNOT gates. The two on the fourth wire are used to do $\mathbf{b_0} \oplus \mathbf{b_1}$ parity calculation, and the two on fifth wire are used to do $\mathbf{b_0} \oplus \mathbf{b_2}$ parity calculation. Let us assume that Bob receives $\mathbf{a}\,|\mathbf{c_0c_1c_2}\rangle + \mathbf{b}\,|\mathbf{d_0d_1d_2}\rangle$. We know that if there is an error, there will be an error in both $\mathbf{c_0c_1c_2}$ and $\mathbf{c_0c_1c_2}$, and it will occur in exactly the same place. When we apply the parity checks, both strings give the same results.

In Bob's circuit, ignoring the fifth wire temporarily, the input for the first four qubits is:

$(\mathbf{a}\,|\mathbf{c_0c_1c_2}\rangle + \mathbf{b}\,|\mathbf{d_0d_1d_2}\rangle)\,|\mathbf{0}\rangle = \mathbf{a}\,|\mathbf{c_0c_1c_2}\rangle\,|\mathbf{0}\rangle + \mathbf{b}\,|\mathbf{d_0d_1d_2}\rangle\,|\mathbf{0}\rangle$

The two CNOT gates attached to the fourth wire perform the parity check on the first two digits. Since, $\mathbf{c_0} \oplus \mathbf{c_1} = \mathbf{d_0} \oplus \mathbf{d_1}$, we have two possibilities:

If $\mathbf{c_0} \oplus \mathbf{c_1} = \mathbf{d_0} \oplus \mathbf{d_1} = \mathbf{0}$, the four qubits at the right end of the circuit will be in the state $\mathbf{a}\,|\mathbf{c_0c_1c_2}\rangle\,|\mathbf{0}\rangle + \mathbf{b}\,|\mathbf{d_0d_1d_2}\rangle\,|\mathbf{0}\rangle = (\mathbf{a}\,|\mathbf{c_0c_1c_2}\rangle + \mathbf{b}\,|\mathbf{d_0d_1d_2}\rangle)\,|\mathbf{0}\rangle$.

If $\mathbf{c_0} \oplus \mathbf{c_1} = \mathbf{d_0} \oplus \mathbf{d_1} = \mathbf{1}$, the four qubits at the right end of the circuit will be in the state $\mathbf{a}\,|\mathbf{c_0c_1c_2}\rangle\,|\mathbf{1}\rangle + \mathbf{b}\,|\mathbf{d_0d_1d_2}\rangle\,|\mathbf{1}\rangle = (\mathbf{a}\,|\mathbf{c_0c_1c_2}\rangle + \mathbf{b}\,|\mathbf{d_0d_1d_2}\rangle)\,|\mathbf{1}\rangle$.

In both cases, the fourth qubit is not entangled with the top three.

A similar argument applies to the fifth qubit and it is not entangled with the others. The fifth qubit is $|\mathbf{0}\rangle$ if $\mathbf{c_0} \oplus \mathbf{c_2} = \mathbf{d_0} \oplus \mathbf{d_2} = \mathbf{0}$, and it is $|\mathbf{1}\rangle$ if $\mathbf{c_0} \oplus \mathbf{c_1} = \mathbf{d_0} \oplus \mathbf{d_1} = \mathbf{0}$.

Since the bottom two qubits are not entangled with the top three, Bob can make measurements on the bottom two qubits, and it will leave the top three unchanged. He proceeds as follows:

If he gets **00**, he does nothing as there is nothing to correct.

If he gets **01**, he flips the third qubit using by installing an **X** gate on the third wire.

If he gets **10**, he flips the second qubit using by installing an **X** gate on the second wire.

If he gets **11**, he flips the first qubit using by installing an **X** gate on the first wire.

The result is that the bit-flip error is corrected and then qubits are back in the state that Alice sent.

This chapter introduced the idea of quantum gates and circuits. We also saw that quantum computation is a more fundamental type of computation and includes classical computation. In the next chapter, we will see how quantum circuits can be used to perform calculations faster than classical circuits.

Chapter 9

Basic Quantum Algorithms

The quantum algorithms are generally perceived to be much faster than their classical counterparts. The speedup is typically explained in terms of being able to put the input into a superposition of all possible inputs and then running the algorithm on this superposition. This means, that instead of running the algorithm on one input, as would happen in the classical case, we can run the algorithm using "parallelism" on possible inputs simultaneously. This explanation raises an important question - we seem to end up with many possible answers which are all superimposed on each other, and if we now make a measurement, won't we just get one of these answers at random? Also, there are far more likely to be wrong answers than right answers, so aren't we more likely to end up with a wrong answer than with the right answer?

There must be more to quantum computing than just placing everything into a superposition of states. The real trick lies in the ability to manipulate these superpositions so that when measured, we get a useful answer. In this chapter, we will look at three quantum algorithms and will learn how to address this problem. We will see that not every problem is suitable for quantum speedup. In fact, **it is safe to say that quantum algorithms are not classical algorithms that have been sped up**. The quantum algorithms work much faster not because of brute force, but due to ingenious ways in which underlying patterns are exploited, which can be seen only from the quantum point of view.

We will end this chapter by looking at properties that problems must-have for a quantum algorithm to solve them faster than the classical approach. First, we should understand how the speed of algorithms can be measured.

9.1 The Complexity Classes and a Million Dollar Prize

Imagine we are given the following four problems. We are further told that we can only use pen and paper to solve each of them.

- Find two natural numbers bigger than 1 whose product is equal to 65.
- Find two natural numbers bigger than 1 whose product is equal to 209.
- Find two natural numbers bigger than 1 whose product is equal to 1943.
- Find two natural numbers bigger than 1 whose product is equal to 40633.

The first one looks trivial, but each subsequent problem is more difficult and will take more time and steps to solve. Before we analyze further, consider the next four problems.

- Multiply 5 and 13 and check if the product is equal to 65.
- Multiply 11 and 19 and check if the product is equal to 209.
- Multiply 29 and 67 and check if the product is equal to 1943.
- Multiply 179 and 227 and check if the product is equal to 40633.

The second set of problems seems relatively easier than the first one. Even though the time taken for each subsequent problem increases, it grows slowly. Let n represent the number of digits in the input number, so in the first set of problems, we see that n goes from $n = 2$ to $n = 5$.

Let $T(n)$ denote the time, or the number of steps, taken to solve a problem of input length n. Complexity looks at how $T(n)$ grows with n. We ask if we can find some positive numbers k and p such that $T(n) \leq kn^p$ for all values of n. If we can, we say that the underlying problem can be solved in *polynomial time*. On the other hand, if we can find a positive number k and $c > 1$, such that $T(n) > kc^n$ for every value of n, we say that problem needs *exponential time*.

We know that given enough time, exponential growth will rise much faster than polynomial growth. The problems that can be solved in polynomial time are considered easy, and those that require exponential time are hard. Let us revisit the two sets of problems that we discussed earlier. The second set involves multiplying two numbers, and as n increases, it takes more time, but it can be proved to be a polynomial time problem. The first set of problems at first looks

to take exponential time, but is it so? It is widely believed to be, but no one has found a proof of it.

The RSA Factoring Challenge was a challenge put forward by RSA Laboratories on March 18, 1991, to encourage research into computational number theory and the practical difficulty of factoring large integers and cracking RSA keys used in cryptography. They published a list of semiprimes (numbers with exactly two prime factors) known as the RSA numbers, with a cash prize for the successful factorization of some of them. The numbers ranged from 100 to 600 decimal digits in length. The 100 digit numbers were quickly factored, however, the numbers with 300 digits are still haven't been factored. The largest number as part of the challenge is 617 decimal digits long and carries a prize of US $200,000. More can be read here: RSA Factoring Challenge.

If a problem can be solved in polynomial time, it is said to belong to the complexity class **P**. Thus, the problem of multiplying two numbers belongs to **P**. Now, consider the case where instead of solving a problem, we are given the answer and we just have to check and verify if the answer is correct. If the process of verification can be done in polynomial time, we say the problem belongs to the complexity class **NP**. The problem of factoring a large number into the product of two primes belongs to **NP**.

It is obvious that checking an answer for correctness is generally easier than actually finding the answer. This means every problem that is in **P** is also in **NP**. How about the converse question - does every **NP** problem belong to **P**? Is it true that every problem whose answer can be verified in polynomial time can also be solved in polynomial time? Most mathematicians and computer scientists expect that $\mathbf{P} \neq \mathbf{NP}$; however, it remains unproven. The problem of whether **NP** is equal to **P** is one of the most important in computer science. In 2000, the Clay Mathematics Institute listed seven Millennium Prize Problems, each with a prize of a Million dollars. The **P** versus **NP** problem is one of the seven.

9.2 Query Complexity

Most computer scientists believe that $\mathbf{P} \neq \mathbf{NP}$, and further they think that there are problems that are in **NP** but not in **P** which a quantum computer can solve in polynomial time. This means, there are problems that a quantum computer can solve in polynomial time but a classical computer cannot. The proof of this requires, as a first step, showing that some problems belong to **NP** but not to **P**, and this no one knows how to do it. So, how do we compare the speed of quantum algorithms with classical algorithms? We have two approaches to this - one theoretical, another practical. The theoretical way is to invent a

new way of measuring complexity that makes it easier to construct proofs. The practical way is to construct quantum algorithms for solving important real-world problems in polynomial time that we believe but have been unable to prove, do not belong to **P**.

An example of the second approach is Shor's algorithm for factoring the product of two primes. Peter Shor constructed a quantum algorithm that works in polynomial time. It is believed but has not been yet proven, that a classical algorithm cannot do this in polynomial time. We spend the rest of this chapter understanding the first approach to define a way of measuring complexity.

We will look at three algorithms in this chapter, all of which are related to *evaluating functions*. The Deutsch and Deutsch-Jozsa algorithms deal with functions belonging to two separate classes where we are given a function at random, and we have to identify which class the function belongs to. Simon's algorithm concerns periodic functions of a special type. Again we are given one of these functions at random, and we have to determine the period. These algorithms involve evaluating functions, and the *query complexity* counts the number of times we have evaluated the function to arrive at the answer. We should note that we do not count how long the function takes to evaluate the input, and we focus on how many times we have invoked the function. The examples will make it clearer, let's start with the simplest example.

9.3 Deutsch's Algorithm

David Deutsch is one of the founding fathers of Quantum Computing. In 1985, he published a landmark paper describing Quantum Turing Machines and Quantum Computation (Quantum theory, the Church–Turing principle, and the universal quantum computer). Deutsch's algorithm is the simplest quantum computing algorithm invented to solve a slightly contrived problem. This is the first algorithm to show that a quantum algorithm can be faster than a classical algorithm.

The algorithm deals with the functions of one variable. The input and output of these functions are either 0 or 1. There are four such functions denoted by f_0, f_1, f_2, and f_3 such that:

1. The function f_0 sends both inputs to 0; that is, $f_0(0) = 0$, $f_0(1) = 0$.
2. The function f_1 sends 0 to 0, and 1 to 1; that is, $f_1(0) = 0$, $f_1(1) = 1$.
3. The function f_2 sends 0 to 1, and 1 to 0; that is, $f_2(0) = 1$, $f_2(1) = 0$.
4. The function f_3 sends both inputs to 1; that is, $f_3(0) = 1$, $f_3(1) = 1$.

A function is called **constant** if its output is the same for all the input values. A function is called **balanced** if it sends half its inputs to 0 and the other half to 1. In our case, functions f_0 and f_3 are constant functions. The functions f_1 and f_2 are examples of balanced functions.

The problem statement is this: Given one of these four functions at random, how many function evaluations do we have to make to determine whether the function is constant or balanced? We should note that we do not want to know which of the functions (f_0, f_1, f_2, f_3) we have, but only in whether it is a balanced or a constant function.

Let us first see how can we solve this problem in the classical domain. We start by evaluating the given function at either 0 or 1. Let us evaluate it at 0. From the definitions, we see that we can either get a 0 or a 1. If we get a 0, we know that $f(0) = 0$. This means the function can either be f_0, or f_1. One of these is a balanced function and the other is a constant function, hence we need to evaluate our function one more time to decide between them. This means, in the classical sense, we need to make two function evaluations to arrive at an answer.

Let us now solve the problem in the quantum domain. We start by constructing a gate that corresponds to the four functions. The following picture represents the gates, where i takes the values $0, 1, 2, 3$ corresponding to the functions.

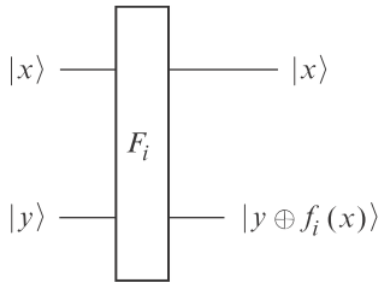

Figure 9.1: Deutsche Gate

This gate works as follows:

1. If we input $|0\rangle \otimes |0\rangle$, it's output is $|0\rangle \otimes |0 \oplus f_i(0)\rangle = |0\rangle \otimes |f_i(0)\rangle$
2. If we input $|0\rangle \otimes |1\rangle$, it's output is $|0\rangle \otimes |1 \oplus f_i(0)\rangle = |0\rangle \otimes |f_i(0) \oplus 1\rangle$
3. If we input $|1\rangle \otimes |0\rangle$, it's output is $|1\rangle \otimes |0 \oplus f_i(0)\rangle = |1\rangle \otimes |f_i(1)\rangle$

4. If we input $|1\rangle \otimes |1\rangle$, it's output is $|1\rangle \otimes |1 \oplus f_i(1)\rangle = |1\rangle \otimes |f_i(1) \oplus 1\rangle$

From the definitions, note that for each i, one of $f_i(0)$ and $f_i(0) \oplus 1$ is equal to 0 and the other is equal to 1. Similarly, one of $f_i(1)$ and $f_i(1) \oplus 1$ is equal to 0 and the other is equal to 1. This means that the four outputs always give us the standard basis elements, telling us the matrix representing our gate is orthogonal, and so we do have a gate.

We can now frame the quantum version of the problem: Given one of these four functions at random, how many times do we need to use the gate to determine whether the underlying function is constant or balanced?

If we are restricted to just entering $|0\rangle$ and $|1\rangle$ to the gate, the analysis is the same as the classical case. We need to use the gate twice to arrive at an answer. Deutsch proved that if we input qubits in a superposition of $|0\rangle$ and $|1\rangle$, the gate needs to be used only once. He proved it using the following circuit.

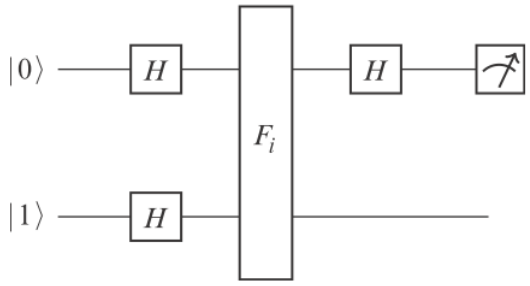

Figure 9.2: Proving via Deutsche Gate

The little meter symbol at the right end of the top wire means that we are going to measure this qubit. The lack of the meter symbol on the second wire tells us that we won't be measuring the second output qubit. Let us see how this circuit works.

The qubits $|0\rangle \otimes |1\rangle$ are the input that passes through the Hadamard gates. This places the input to the state:

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

After passing through the F_i gate, the state changes to:

$$\frac{1}{2}(|0\rangle \otimes |f_i(0)\rangle - |0\rangle \otimes |f_i(0) \oplus 1\rangle + |1\rangle \otimes |f_i(1)\rangle - |1\rangle \otimes |f_i(1) \oplus 1\rangle)$$

This is simplified to:

$$\frac{1}{2}\left(|0\rangle \otimes \left(|f_i(0) - |f_i(0) \oplus 1\rangle\right) + |1\rangle \otimes \left(|f_i(1) - |f_i(1) \oplus 1\rangle\right)\right)$$

Note that $|f_i(0)\rangle - |f_i(0) \oplus 1\rangle$ is either $|0\rangle - |1\rangle$ or $|1\rangle - |0\rangle$, depending on whether $|f_i(0)\rangle$ is 0 or 1. This means we can write:

$$|f_i(0)\rangle - |f_i(0) \oplus 1\rangle = (-1)^{f_i(0)}\left(|0\rangle - |1\rangle\right)$$

Similarly, we have:

$$|f_i(1)\rangle - |f_i(1) \oplus 1\rangle = (-1)^{f_i(1)}\left(|0\rangle - |1\rangle\right)$$

Thus, the state of qubits after passing the F_i gate can be written as:

$$\frac{1}{2}\left(|0\rangle \otimes \left((-1)^{f_i(0)}\left(|0\rangle - |1\rangle\right)\right) + |1\rangle \otimes \left((-1)^{f_i(1)}\left(|0\rangle - |1\rangle\right)\right)\right)$$

This can be rearranged as:

$$\frac{1}{2}\left(\left((-1)^{f_i(0)}|0\rangle + (-1)^{f_i(1)}|1\rangle\right) \otimes \left(|0\rangle - |1\rangle\right)\right)$$

This can be rearranged to:

$$\frac{1}{\sqrt{2}}\left((-1)^{f_i(0)}|0\rangle + (-1)^{f_i(1)}|1\rangle\right) \otimes \frac{1}{\sqrt{2}}\left(|0\rangle - |1\rangle\right)$$

This shows that the two qubits are **not entangled**, and the top qubit is in the state $\frac{1}{\sqrt{2}}\left((-1)^{f_i(0)}|0\rangle + (-1)^{f_i(1)}|1\rangle\right)$

We now analyze this state for each of the four possibilities of f_i.

1. For f_0, we have $f_0(0) = 0$, $f_0(1) = 0$, so the top qubit is in state $\frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right)$.
2. For f_1, we have $f_1(0) = 0$, $f_1(1) = 1$, so the top qubit is in state $\frac{1}{\sqrt{2}}\left(|0\rangle - |1\rangle\right)$.
3. For f_2, we have $f_2(0) = 1$, $f_2(1) = 0$, so the top qubit is in state $\frac{-1}{\sqrt{2}}\left(|0\rangle - |1\rangle\right)$.
4. For f_3, we have $f_3(0) = 1$, $f_3(1) = 1$, so the top qubit is in state $\frac{-1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right)$.

The next step in the circuit sends this qubit through a Hadamard gate. The Hadamard gate sends $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ to $|0\rangle$ and $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$ to $|1\rangle$. Therefore, we have:

1. If $i = 0$ (f_0), the qubit is $|0\rangle$.
2. If $i = 1$ (f_1), the qubit is $|1\rangle$.
3. If $i = 2$ (f_2), the qubit is $-|1\rangle$.
4. If $i = 3$ (f_3), the qubit is $-|0\rangle$.

As the last step, we now measure the top qubit in the standard basis. We will get 0 if $i = 0$, or $i = 3$, and we will get 1 if $i = 1$, or $i = 2$. We also know that the functions f_0 and f_3 are constant functions, and the functions f_1 and f_2 are balanced. Therefore, if after measuring we get a 0, we can conclude that the original function was constant. If we get 1, we know that the original function was balanced.

This shows that in the quantum case, we need to query the function only once to know the answer with certainty. We can see that Deutsch's problem will have a speedup using a quantum algorithm. This algorithm however has no real application and was designed to prove that there are quantum algorithms that are faster than classical ones. We will further look into two more algorithms that input a number of qubits and then sends them via a Hadamard gate. We now introduce some more notations and mathematics to help us keep superposition from becoming too messy.

9.4 The Kronecker Product

The Hadamard gate is defined by the following matrix.

$$\mathbf{H} = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \end{bmatrix} = \frac{\mathbf{1}}{\sqrt{\mathbf{2}}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

The means:

$$\mathbf{H}|\mathbf{0}\rangle = \frac{\mathbf{1}}{\sqrt{\mathbf{2}}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \frac{\mathbf{1}}{\sqrt{\mathbf{2}}} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \frac{\mathbf{1}}{\sqrt{\mathbf{2}}} \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \frac{\mathbf{1}}{\sqrt{\mathbf{2}}} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \frac{\mathbf{1}}{\sqrt{\mathbf{2}}}|\mathbf{0}\rangle + \frac{\mathbf{1}}{\sqrt{\mathbf{2}}}|\mathbf{1}\rangle$$

$$\mathbf{H}\left|\mathbf{1}\right\rangle = \frac{\mathbf{1}}{\sqrt{\mathbf{2}}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}\begin{bmatrix}0\\ 1\end{bmatrix} = \frac{\mathbf{1}}{\sqrt{\mathbf{2}}}\begin{bmatrix}1\\ -1\end{bmatrix} = \frac{\mathbf{1}}{\sqrt{\mathbf{2}}}\begin{bmatrix}1\\ 0\end{bmatrix} - \frac{\mathbf{1}}{\sqrt{\mathbf{2}}}\begin{bmatrix}0\\ 1\end{bmatrix} = \frac{\mathbf{1}}{\sqrt{\mathbf{2}}}\left|\mathbf{0}\right\rangle - \frac{\mathbf{1}}{\sqrt{\mathbf{2}}}\left|\mathbf{1}\right\rangle$$

If we have two input qubits and both are sent through the Hadamard gate, the four basis vectors will be sent as follows:

$$\left|\mathbf{0}\right\rangle \otimes \left|\mathbf{0}\right\rangle : \left(\frac{\mathbf{1}}{\sqrt{\mathbf{2}}}\left|\mathbf{0}\right\rangle + \frac{\mathbf{1}}{\sqrt{\mathbf{2}}}\left|\mathbf{1}\right\rangle\right) \otimes \left(\frac{\mathbf{1}}{\sqrt{\mathbf{2}}}\left|\mathbf{0}\right\rangle + \frac{\mathbf{1}}{\sqrt{\mathbf{2}}}\left|\mathbf{1}\right\rangle\right) = \frac{\mathbf{1}}{\mathbf{2}}\left(\left|\mathbf{00}\right\rangle + \left|\mathbf{01}\right\rangle + \left|\mathbf{10}\right\rangle + \left|\mathbf{11}\right\rangle\right)$$

$$\left|\mathbf{0}\right\rangle \otimes \left|\mathbf{1}\right\rangle : \left(\frac{\mathbf{1}}{\sqrt{\mathbf{2}}}\left|\mathbf{0}\right\rangle + \frac{\mathbf{1}}{\sqrt{\mathbf{2}}}\left|\mathbf{1}\right\rangle\right) \otimes \left(\frac{\mathbf{1}}{\sqrt{\mathbf{2}}}\left|\mathbf{0}\right\rangle - \frac{\mathbf{1}}{\sqrt{\mathbf{2}}}\left|\mathbf{1}\right\rangle\right) = \frac{\mathbf{1}}{\mathbf{2}}\left(\left|\mathbf{00}\right\rangle - \left|\mathbf{01}\right\rangle + \left|\mathbf{10}\right\rangle - \left|\mathbf{11}\right\rangle\right)$$

$$\left|\mathbf{1}\right\rangle \otimes \left|\mathbf{0}\right\rangle : \left(\frac{\mathbf{1}}{\sqrt{\mathbf{2}}}\left|\mathbf{0}\right\rangle - \frac{\mathbf{1}}{\sqrt{\mathbf{2}}}\left|\mathbf{1}\right\rangle\right) \otimes \left(\frac{\mathbf{1}}{\sqrt{\mathbf{2}}}\left|\mathbf{0}\right\rangle + \frac{\mathbf{1}}{\sqrt{\mathbf{2}}}\left|\mathbf{1}\right\rangle\right) = \frac{\mathbf{1}}{\mathbf{2}}\left(\left|\mathbf{00}\right\rangle + \left|\mathbf{01}\right\rangle - \left|\mathbf{10}\right\rangle - \left|\mathbf{11}\right\rangle\right)$$

$$\left|\mathbf{1}\right\rangle \otimes \left|\mathbf{1}\right\rangle : \left(\frac{\mathbf{1}}{\sqrt{\mathbf{2}}}\left|\mathbf{0}\right\rangle - \frac{\mathbf{1}}{\sqrt{\mathbf{2}}}\left|\mathbf{1}\right\rangle\right) \otimes \left(\frac{\mathbf{1}}{\sqrt{\mathbf{2}}}\left|\mathbf{0}\right\rangle - \frac{\mathbf{1}}{\sqrt{\mathbf{2}}}\left|\mathbf{1}\right\rangle\right) = \frac{\mathbf{1}}{\mathbf{2}}\left(\left|\mathbf{00}\right\rangle - \left|\mathbf{01}\right\rangle - \left|\mathbf{10}\right\rangle + \left|\mathbf{11}\right\rangle\right)$$

Rewriting the above in terms of four-dimensional kets, we will say:

$$\begin{bmatrix}1\\0\\0\\0\end{bmatrix} \text{ goes to } \frac{1}{2}\begin{bmatrix}1\\1\\1\\1\end{bmatrix}$$

$$\begin{bmatrix}0\\1\\0\\0\end{bmatrix} \text{ goes to } \frac{1}{2}\begin{bmatrix}1\\-1\\1\\-1\end{bmatrix}$$

$$\begin{bmatrix}0\\0\\1\\0\end{bmatrix} \text{ goes to } \frac{1}{2}\begin{bmatrix}1\\1\\-1\\-1\end{bmatrix}$$

$$\begin{bmatrix}0\\0\\0\\1\end{bmatrix} \text{ goes to } \frac{1}{2}\begin{bmatrix}1\\-1\\-1\\1\end{bmatrix}$$

The above describes one orthogonal basis being sent to another orthogonal basis. We can write the following matrix corresponding to this transformation.

$$\mathbf{H}^{\otimes\mathbf{2}} = \frac{\mathbf{1}}{\mathbf{2}}\begin{bmatrix}1 & 1 & 1 & 1\\ 1 & -1 & 1 & -1\\ 1 & 1 & -1 & -1\\ 1 & -1 & -1 & 1\end{bmatrix}$$

Further, we see that:

$$\begin{aligned}
\mathbf{H}^{\otimes \mathbf{2}} &= \frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \\
&= \frac{1}{2}\begin{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \end{bmatrix} & \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \end{bmatrix} \\ \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \end{bmatrix} & -\begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \end{bmatrix} \end{bmatrix} \\
&= \frac{1}{2}\begin{bmatrix} \mathbf{H} & \mathbf{H} \\ \mathbf{H} & -\mathbf{H} \end{bmatrix}
\end{aligned}$$

The pattern continues, and the matrix corresponding to inputting three qubits and sending them via the Hadamard gate can be written as:

$$\mathbf{H}^{\otimes \mathbf{3}} = \frac{1}{\sqrt{2}} \begin{bmatrix} \mathbf{H}^{\otimes \mathbf{2}} & \mathbf{H}^{\otimes \mathbf{2}} \\ \mathbf{H}^{\otimes \mathbf{2}} & -\mathbf{H}^{\otimes \mathbf{2}} \end{bmatrix}$$

$$= \frac{1}{2\sqrt{2}} \begin{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} & \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \\ \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} & - \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \end{bmatrix}$$

$$= \frac{1}{2\sqrt{2}} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix}$$

As n increases, these matrices quickly become large, but the following relation always holds:

$$\mathbf{H}^{\otimes \mathbf{n}} = \frac{1}{2\sqrt{2}} \begin{bmatrix} \mathbf{H}^{\otimes (\mathbf{n-1})} & \mathbf{H}^{\otimes (\mathbf{n-1})} \\ \mathbf{H}^{\otimes (\mathbf{n-1})} & -\mathbf{H}^{\otimes (\mathbf{n-1})} \end{bmatrix}$$

This is a recursive formula to quickly calculate for any value of n. These are called *Kronecker Products*. We will heavily use these matrices when we discuss

Simon's problem. However, for the next algorithm, we note that the top row of each of these matrices is equal to one another; for $\mathbf{H}^{\otimes \mathbf{n}}$, they are all equal to $\left(\frac{1}{\sqrt{2}}\right)^n$.

9.5 The Deutsch-Jozsa Algorithm

The Deutsch-Jozsa problem is a generalized version of Deutsch's one variable problem. We now consider functions of n variables. The input to each of these variables can be either 0 or 1, the output is also 0 or 1. We are also told that the function is either constant, sending all inputs to either 0 or all to 1, or it is balanced, sending half of the inputs to 0 and the other half to 1. We are given one of these functions at random. The problem statement then is: how many function evaluations do we need to make to determine whether the function is balanced or constant?

To understand it better, let us consider a concrete example and take $n = 3$. The function takes three inputs, each of which can take two values. Thus, there are $2^3 = 8$ possible inputs. These are:

$(0,0,0), (0,0,1), (0,1,0), (0,1,1), (1,0,0), (1,0,1), (1,1,0), (1,1,1)$

Classically, suppose we evaluate $f(0,0,0)$ and get the answer that $f(0,0,0) = 1$. Since this information alone is insufficient to deduce the answer, we do another evaluation, say $f(0,0,1)$. If we get $f(0,0,1) = 0$, we are done and we know that the function is balanced. On the other hand, if we get $f(0,0,1) = 1$, we still cannot determine the final answer from these two pieces of information. In the worst possible scenario, we may get the same answer for the first four evaluations, that is, $f(0,0,0) = 1$, $f(0,0,1) = 1$, $f(0,1,0) = 1$, and $f(0,1,1) = 1$ and we still cannot determine if the function is balanced or constant. We need to ask one more question. If the answer to the next question is also 1, we know that function is constant. If the answer to the next question is 0, we know that the function is balanced.

In general, for a function of n variables, there will be 2^n input strings. In the best-case scenario, we can get the answer in just two queries. In the worst case, it will take 2^{n-1} queries. This is an exponential growth function and quickly becomes very large even for moderate values of n. The Deutsch-Jozsa algorithm is a quantum algorithm that just requires one question to get the answer, so the speedup is substantial!

Fiven any function $f(x_0, x_1, \cdots, x_{n-1})$ with n inputs and one boolean output, we construct the gate F as follows where the slashes with n on top lines indicate that there are n lines in parallel.

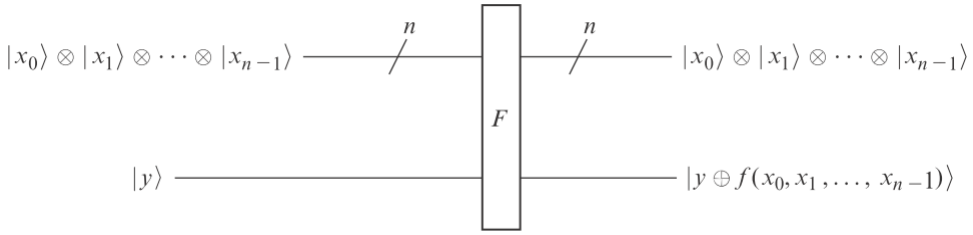

Figure 9.3: Deutsche Jozsa Gate

This circuit tells us what happens when each of the kets, $|x_i\rangle$ is either $|0\rangle$ or $|1\rangle$. The input consists of $n+1$ kets, $|x_0\rangle \otimes |x_1\rangle \otimes \cdots \otimes |x_{n_1}\rangle$ and $|y\rangle$ where the first n correspond to the function variables. The output also consists of $n+1$ kets, the first n of which are the same as input kets. The last output is the ket $|f(x_0, x_1, \cdots, x_{n-1})\rangle$ if $y = 0$, and the ket of other boolean value if $y = 1$.

We next look at the quantum circuit for this function. It is a generalization of the circuit used in Deutsch's algorithm - all the top qubits pass through the Hadamard gates on either side of the black box.

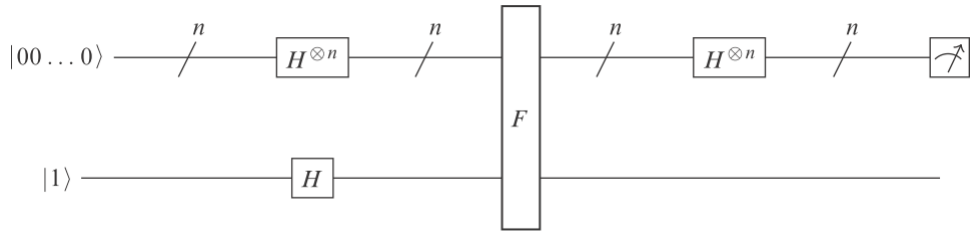

Figure 9.4: Deutsche Jozsa Algorithm

We will analyze the system step by step for $n = 2$ case. Be assured, that the algorithm will work for any value of n as well.

Step 1. The Qubits passing through the Hadamard Gates The top n inputs are all $|0\rangle$. In the case of $n = 2$, this will be $|00\rangle$. After passing through the top Hadamard gate, we get:

$$\mathbf{H}^{\otimes \mathbf{2}}\left(|\mathbf{00}\rangle\right) = \frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$
$$= \frac{1}{2}\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$
$$= \frac{1}{2}\left(|00\rangle + |01\rangle + |10\rangle + |11\rangle\right)$$

We see a superposition of all possible states, and each of the basis kets has the same probability amplitude equal to $\frac{1}{2}$. The above logic works for all values of n. After the n qubits have passed through $\mathbf{H}^{\otimes \mathbf{n}}$, all of them will be in superposition of all possible states, and each will have the same probability amplitude of $\left(\frac{1}{\sqrt{2}}\right)^n$.

The bottom qubit is $|1\rangle$. After passing through the Hadamard gate, it becomes $\left(\frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle\right)$. At this stage, the three input qubits are in this state:

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

Step 2. Qubits passing through the **F** Gate

After passing through the **F** gate, the qubit moves to state:

$$\frac{1}{2\sqrt{2}}|00\rangle \otimes \left(|f(0,0)\rangle - |f(0,0) \oplus 1\rangle\right)$$
$$+ \frac{1}{2\sqrt{2}}|01\rangle \otimes \left(|f(0,1)\rangle - |f(0,1) \oplus 1\rangle\right)$$
$$+ \frac{1}{2\sqrt{2}}|10\rangle \otimes \left(|f(1,0)\rangle - |f(1,0) \oplus 1\rangle\right)$$
$$+ \frac{1}{2\sqrt{2}}|11\rangle \otimes \left(|f(1,1)\rangle - |f(1,1) \oplus 1\rangle\right)$$

We know that of x is either 0 or 1, $|x\rangle - |x \oplus 1\rangle = (-1)^x (|0\rangle - |1\rangle)$. With this, the above can be written as:

$$
\begin{aligned}
&(-1)^{f(0,0)} \frac{1}{2} |00\rangle \otimes \frac{1}{\sqrt{2}} (|0\rangle - |1\rangle) \\
&+ (-1)^{f(0,1)} \frac{1}{2} |01\rangle \otimes \frac{1}{\sqrt{2}} (|0\rangle - |1\rangle) \\
&+ (-1)^{f(1,0)} \frac{1}{2} |10\rangle \otimes \frac{1}{\sqrt{2}} (|0\rangle - |1\rangle) \\
&+ (-1)^{f(1,1)} \frac{1}{2} |11\rangle \otimes \frac{1}{\sqrt{2}} (|0\rangle - |1\rangle)
\end{aligned}
$$

This shows that the top and bottom qubits are not entangled. We just look at the top two qubits which are in the state:

$$\frac{1}{2} \left((-1)^{f(0,0)} |00\rangle + (-1)^{f(0,1)} |01\rangle + (-1)^{f(1,0)} |10\rangle + (-1)^{f(1,1)} |11\rangle \right)$$

This is a general argument and would work similarly for any value of n. At this stage, we have a state that is a superposition of all basis kets. Each ket $|x_0 x_1 \cdots x_{n-1}\rangle$ is multiplied by $\left(\frac{1}{\sqrt{2}}\right)^n (-1)^{f(x_0, x_1, \cdots, x_{n-1})}$.

Step 3. The TOP Qubits passing through the Hadamard Gate

The standard way is to convert the state to a column vector and then multiply with the appropriate Kronecker product of the Hadamard matrix. This will give:

$$\frac{1}{4} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \begin{bmatrix} (-1)^{f(0,0)} \\ (-1)^{f(0,1)} \\ (-1)^{f(1,0)} \\ (-1)^{f(1,1)} \end{bmatrix}$$

We do not have to calculate all the entries in this resultant column vector. We just need to calculate the top entry which comes from multiplying the bra corresponding to the top row of the matrix with the ket given by the column vector. This yields the following as the probability amplitude for the ket $|00\rangle$:

$$\frac{1}{4} \left((-1)^{f(0,0)} + (-1)^{f(0,1)} + (-1)^{f(1,0)} + (-1)^{f(1,1)} \right)$$

We can now calculate this amplitude for the possible function types.

If f is constant and sends everything to 0, the probability amplitude is 1.
If f is constant and sends everything to 1, the probability amplitude is -1.
If f is balanced, the probability amplitude is 0.

Step 4. Measuring the Top Qubits

When we measure the top qubits, we will get one of $00, 01, 10, or 11$. The question we ask is "do we observe 00"? If the function is constant, we will get it with a probability of 1. If the function is balanced, we get it with probability 0. This means, that if the measurement result gives 00, we conclude that the function is constant. If the result is not 00, we conclude that the function is balanced.

The above argument can be shown to work for any value of n. Just before we measure the qubits, the probability amplitude for $|0\cdots 0\rangle$ is:

$$\frac{1}{2^n}\left((-1)^{f(0,0,\cdots,0)} + (-1)^{f(0,0,\cdots,1)} + \cdots + (-1)^{f(1,1,\cdots,1)}\right)$$

In the case of $n = 2$, this will be ± 1 if f is constant and 0 if f is balanced. Therefore, if every measurement gives 0, the function is constant. If at least one of the measurements is 1, the function is balanced.

This shows that we can solve the Deutsch-Jozsa problem for any value of n with just one use of the circuit and by asking only one question. In the classical case, in the worst case, we would have required $2^{n-1} + 1$ questions and hence the improvement is significant!

9.6 Simon's Algorithm

The algorithms seen so far were interesting in the sense that both gave the answer with only one question. In general, a quantum algorithm will be more involved than this and will often be a combination of quantum circuits and a little probability as well. The next algorithm that we consider is one such example. However, before we describe the algorithm, we look at a different way of adding binary strings.

We defined $\oplus$ as exclusive OR, or XOR, operation. The binary XOR operation was defined as follows:

$0 \oplus 0 = 0$
$0 \oplus 1 = 1$
$1 \oplus 0 = 1$
$1 \oplus 1 = 0$

We now extend the same logic to the addition of binary strings of the same lengths. This means:

$a_0a_1\cdots a_n \oplus b_0b_1\cdots b_n = c_0c_1\cdots c_n$ where

$c_0 = a_0 \oplus b_0, c_1 = a_1 \oplus b_1, \cdots, c_n = a_n \oplus b_n$

As an concrete example, we have: $1101 \oplus 0111 = 1010$. We do a bitwise XOR and ignore any carries that may be there.

Problem Statement

We have a function f that sends binary strings of length n to binary strings of length n. It further has a property that there is one secret string s, such that $f(x) = f(y)$ if and only if $y = x$ or $y = x \oplus s$. The string s doesn't consist entirely of 0s, this means that different input strings can map to the same output string. The problem is to find the secret string s by querying f (as few times as possible).

An explicit example will make things clear. We consider the case where $n = 3$, that is, the function will take binary strings of length 3 as input and output binary strings of length 3. Let us further assume that the secret string, $s = 110$. We know the following:

$000 \oplus 110 = 110 \qquad 001 \oplus 110 = 111$
$010 \oplus 110 = 100 \qquad 011 \oplus 110 = 101$
$100 \oplus 110 = 010 \qquad 101 \oplus 110 = 011$
$110 \oplus 110 = 000 \qquad 111 \oplus 110 = 001$

This mean, for $s = 110$, we have the following:

$f(000) = f(110) \qquad f(001) = f(111) \qquad f(010) = f(100) \qquad f(011) = f(101)$

A function satisfying these is:

$f(000) = f(110) = 110$
$f(001) = f(111) = 010$
$f(010) = f(100) = 111$
$f(011) = f101) = 000$

We of course do not know the function f, or the secret string s. We want to find s, and the question is how many function evaluations need to be made to determine it?

We can start by evaluating function f on strings, and we stop as soon as we get a repeated answer. Once we have found two strings that give the same answer, we can easily find s. Let's see how.

Assume that we find $f(011) = f(101)$. This means:

$011 \oplus s_0 s_1 s_2 = 101$

We also know that $011 \oplus 011 = 000$, so we bitwise add 011 to both sides and see:

$$\begin{aligned} & 011 \oplus s_0 s_1 s_2 = 101 \\ \implies & 011 \oplus 011 \oplus s_0 s_1 s_2 = 011 \oplus 101 \\ \implies & 0 \oplus s_0 s_1 s_2 = 000 \\ \implies & s_0 s_1 s_2 = 000 \end{aligned}$$

How many evaluations do we need to make using the classical algorithm? In the case of $n = 3$, we have 8 binary strings. It might happen that we get a different answer each time we evaluate four of these. However, we are guaranteed to get a repeated answer on the fifth evaluation and we can then deduce the secret string. In general, for strings of length n, there are 2_n binary strings, and in the worst case, we will need $2_{n-1} + 1$ evaluations to deduce the secret string. This is exponential growth with n. Can we do better with a quantum algorithm?

Before we look into the details of the quantum algorithm, we revisit the Kronecker Product of Hadamard matrices once again.

The Hadamard Matrix and the Dot Product Given two binary strings $a_0 a_1 \cdots a_{n-1}$ and $b_0 b_1 \cdots b_{n-1}$ of length n, we define the *dot product* as:

$a \cdot b = a_0 \times b_0 \oplus a_1 \times b_1 \oplus \cdots \oplus a_{n-1} \times b_{n-1}$ where $\times$ represents the usual multiplication.

As an example, if $a = 101$ and $b = 111$, then:

$$\begin{aligned} a \cdot b &= (1 \times 1) \oplus (0 \times 1) \oplus (1 \times 1) \\ &= 0 \oplus 0 \oplus 0 \\ &= 0 \end{aligned}$$

This operation can be thought of as multiplying corresponding terms of the sequences, then adding and finally determining whether the sum is odd or even.

We further start matrix indices from 0 and will represent them in binary form. So, for a 4×4 matrix, we show:

$$\begin{array}{c} \\ 00 \\ 01 \\ 10 \\ 11 \end{array} \begin{array}{c} \begin{array}{cccc} 00 & 01 & 10 & 11 \end{array} \\ \begin{bmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{bmatrix} \end{array}$$

The position of an entry in this matrix is given by listing both its row and column. If we further make the entry in i^{th} row and j^{th} column as $i \cdot j$, we get the following matrix:

$$\begin{array}{c} \\ 00 \\ 01 \\ 10 \\ 11 \end{array} \begin{array}{c} \begin{array}{cccc} 00 & 01 & 10 & 11 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \end{array}$$

Further recall that:

$$\mathbf{H^{\otimes 2}} = \frac{\mathbf{1}}{\mathbf{2}} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

We can see that the entries that are 1 in the dot-product matrix are in exactly the same positions as -1 in $\mathbf{H^{\otimes 2}}$. Further, using the facts that $(-1)^0 = 1$ and $(-1)^1 = -1$, we can write:

$$\mathbf{H^{\otimes 2}} = \frac{1}{2} \begin{bmatrix} (-1)^{00 \cdot 00} & (-1)^{00 \cdot 01} & (-1)^{00 \cdot 10} & (-1)^{00 \cdot 11} \\ (-1)^{01 \cdot 00} & (-1)^{01 \cdot 01} & (-1)^{01 \cdot 10} & (-1)^{01 \cdot 11} \\ (-1)^{10 \cdot 00} & (-1)^{10 \cdot 01} & (-1)^{10 \cdot 10} & (-1)^{10 \cdot 11} \\ (-1)^{11 \cdot 00} & (-1)^{11 \cdot 01} & (-1)^{11 \cdot 10} & (-1)^{11 \cdot 11} \end{bmatrix}$$

This method for finding positive and negative values holds in general as well. For example, if we want to know the entry of $\mathbf{H^{\otimes 3}}$ in row 101 and column 111, we find the dot product and get 0. This means the entry will be positive.

We next see what happens when we add two columns of these matrices. We will add two columns that are paired by the secret string s of Simon's Problem. If one of the columns is labeled x, the other will be $x \oplus s$. As an example, consider the case of a string of lengths 2. Further, assume that secret string $s = 10$. This means we will be adding columns labeled 00 and 10, or columns labeled 01 and 11. We start with $\mathbf{H^{\otimes 2}}$ definition.

$$\mathbf{H^{\otimes 2}} = \frac{\mathbf{1}}{\mathbf{2}} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

Adding columns 00 and 10 gives: $\frac{1}{2} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} + \frac{1}{2} \begin{bmatrix} 1 \\ 1 \\ -1 \\ -1 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 2 \\ 2 \\ 0 \\ 1 \end{bmatrix}$

Adding the columns 01 and 11 gives: $\frac{1}{2}\begin{bmatrix}1\\-1\\1\\-1\end{bmatrix}+\frac{1}{2}\begin{bmatrix}1\\-1\\-1\\1\end{bmatrix}=\frac{1}{2}\begin{bmatrix}2\\-2\\0\\0\end{bmatrix}$

We can see that some probability amplitudes are amplified, and some are reduced. Let us see why.

Note that, $(-1)^{a\cdot(x\oplus s)}=(-1)^{a\cdot x}\cdot(-1)^{a\cdot s}$

This tells us that $(-1)^{a\cdot(x\oplus s)}$ and $(-1)^{a\cdot x}$ will be equal if $a\cdot s=0$, and will have opposite sign if $a\cdot s=1$. This can be summarized as:

$(-1)^{a\cdot(x\oplus s)}+(-1)^{a\cdot x}=\pm 2$, if $a\cdot s=0$
$(-1)^{a\cdot(x\oplus s)}+(-1)^{a\cdot x}=0$, if $a\cdot s=1$

This means that when we add two columns given by x and $x\oplus s$, the entries in ow a will be 0 if $a\cdot s=1$, and will be ± 2 if $a\cdot s=0$. In general, the entries cancel in the rows labeled with strings that have a dot product of 1 with the secret string s.

Looking at the example from earlier, the last entries are 0 because the rows have labels 10 and 11, and both of them have a dot product of 1 with the secret string $s=10$. The non-zero entries occur in rows with labels 00 and 01 and both of them have a dot product of 0 with the secret string $s=10$.

We now have all the necessary background to understand the quantum circuit solving Simon's Problem. The quantum circuit will give us a string whose dot product with the secret string s is 0. It will achieve this by adding two columns of the Hadamard matrix. Let's see in action next.

The Quantum Circuit Solving Simon's Problem

We start by constructing the gate that acts like the function f.

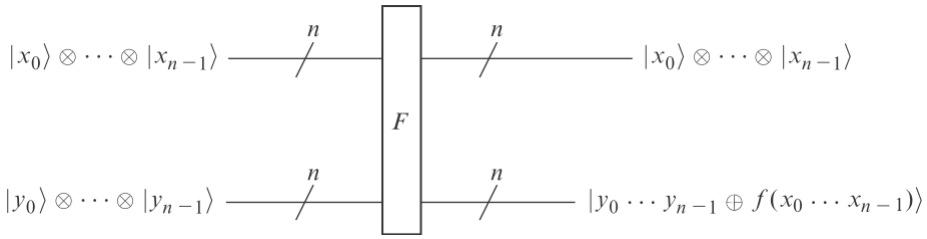

Figure 9.5: Gate for Simon's Problem

We input two equal length strings consisting of $|0\rangle$s and $|1\rangle$s. The top output string is unchanged and the bottom one is the function evaluated on the top string added bitwise to the bottom string.

We next see the quantum circuit for the problem.

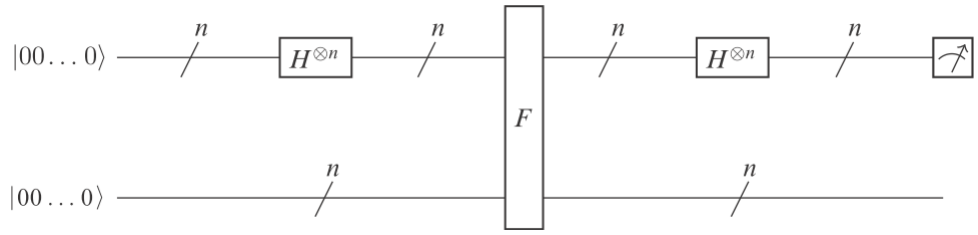

Figure 9.6: Circuit for Simon's Problem

We will see how the circuit works for the case where $n = 2$. The logic will extend generally for higher values of n.

In the first step, the top two qubits pass through the Hadamard Gate. The top two qubits are in the state $|00\rangle$. After passing the Hadamard gate, they will be in the state $\frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle)$. The bottom qubits remain in state $|00\rangle$. At this stage, the four qubits are in the state:

$$\frac{1}{2}(|00\rangle \otimes |00\rangle + |01\rangle \otimes |00\rangle + |10\rangle \otimes |00\rangle + |11\rangle \otimes |00\rangle)$$

The qubits now pass through the F gate which changes the state to:

$$\frac{1}{2}(|00\rangle \otimes |f(00)\rangle + |01\rangle \otimes |f(01)\rangle + |10\rangle \otimes |f(10)\rangle + |11\rangle \otimes |f(11)\rangle)$$

The top qubits now pass through the Hadamard Gate, which changes the state to:

$$\begin{aligned}
&\frac{1}{4}(|00\rangle + |01\rangle + |10\rangle + |11\rangle) \otimes |f(00)\rangle \\
&+ \frac{1}{4}(|00\rangle - |01\rangle + |10\rangle - |11\rangle) \otimes |f(01)\rangle \\
&+ \frac{1}{4}(|00\rangle + |01\rangle - |10\rangle - |11\rangle) \otimes |f(10)\rangle \\
&+ \frac{1}{4}(|00\rangle - |01\rangle - |10\rangle + |11\rangle) \otimes |f(111)\rangle
\end{aligned}$$

The pattern of + and − signs comes from the matrix for $\mathbf{H}^{\otimes \mathbf{2}}$. We now rearrange the terms, solving for the first two qubits, which results in the following:

$$\frac{1}{4}|00\rangle \otimes (|f(00)\rangle + |f(01)\rangle + |f(10)\rangle + |f(11)\rangle)$$
$$+ \frac{1}{4}|01\rangle \otimes (|f(00)\rangle - |f(01)\rangle + |f(10)\rangle - |f(11)\rangle)$$
$$+ \frac{1}{4}|10\rangle \otimes (|f(00)\rangle + |f(01)\rangle - |f(10)\rangle - |f(11)\rangle)$$
$$+ \frac{1}{4}|11\rangle \otimes (|f(00)\rangle - |f(01)\rangle - |f(10)\rangle + |f(11)\rangle)$$

We notice two things now. First, the pattern of + and − signs comes from the matrix for $\mathbf{H}^{\otimes \mathbf{2}}$. Second, the pairs of qubits to the left of the tensor product correspond to the row numbers. We know that $f(b) = f(b \oplus s)$, and hence $|\langle\rangle f(b)) = |\langle\rangle f(b \oplus s))$. We can further simplify things, by combining these terms, by adding their probability amplitudes. This corresponds to the column addition we just looked at.

Specifically, let $s = 10$, then we know that $f(00) = f(10)$, and $f(01) = f(11)$. Using these, we get the state as:

$$\frac{1}{4}|00\rangle \otimes (|f(00)\rangle + |f(01)\rangle + |f00\rangle + |f(01)\rangle)$$
$$+ \frac{1}{4}|01\rangle \otimes (|f(00)\rangle - |f(01)\rangle + |f(00)\rangle - |f(01)\rangle)$$
$$+ \frac{1}{4}|10\rangle \otimes (|f(00)\rangle + |f(01)\rangle - |f(00)\rangle - |f(01)\rangle)$$
$$+ \frac{1}{4}|11\rangle \otimes (|f(00)\rangle - |f(01)\rangle - |f(00)\rangle + |f(01)\rangle)$$
$$=$$
$$\frac{1}{4}|00\rangle \otimes (2|f(00)\rangle + 2|f(01)\rangle)$$
$$+ \frac{1}{4}|01\rangle \otimes (2|f(00)\rangle - 2|f(01)\rangle)$$
$$+ \frac{1}{4}|10\rangle \otimes (0)$$
$$+ \frac{1}{4}|11\rangle \otimes (0)$$

The kets to the left of the tensor products are labeled with the row numbers of the matrix. The 0s on the right of the tensor products occur in the rows whose dot product with s is 1.

This can be further simplified to:

$$\frac{1}{\sqrt{2}}|00\rangle \otimes \frac{1}{\sqrt{2}}(|f(00)\rangle + |f(01)\rangle) + \frac{1}{\sqrt{2}}|0\rangle \otimes \frac{1}{\sqrt{2}}(|f(00)\rangle - |f(01)\rangle)$$

When we measure the top two qubits we will get either 00 or 01, each with probability $\frac{1}{2}$.

We have taken the example of $n = 2$, however, the same reasoning will extend to all other values of n as well. At the end of the process, we will end up with one of the strings whose dot product with the secret string is 0. Each of these strings is equally likely.

Even after so much mathematics and calculations, we still haven't found the secret string s. We now head to **classical part of Simon's Algorithm**. Let us see how it works.

We start with an example of $n = 5$. We know that there is some secret string $s = s_0s_1s_2s_3s_4$. We are not allowed 00000, hence there are another $2^5 - 1 = 31$ choices for s. We are going to try to find it using Simon's quantum circuit.

We run the circuit and, say, get 10100 as an answer. We know that the dot product of 10100 with s will give 0. This means:

$$
\begin{aligned}
&10100 \cdot s = 0 \\
&\implies 10100 \cdot s_0s_1s_2s_3s_4 = 0 \\
&\implies 1 \times s_0 \oplus 0 \times s_1 \oplus 1 \times s_2 \oplus 0 \times s_3 \oplus 0 \times s_4 = 0 \\
&\implies s_0 \oplus s_2 = 0
\end{aligned}
$$

Since all digits are either 0 or 1, we deduce that $s_0 = s_2$.

We now run the circuit again hoping that we don't get 10100 again. The probability of this happening is $\frac{1}{16}$. We further hope that we don't get 00000 since this will not give any new information. Assume that we now get 00100. This means:

$$
\begin{aligned}
&00100 \cdot s = 0 \\
&\implies 00100 \cdot s_0s_1s_2s_3s_4 = 0 \\
&\implies 0 \times s_0 \oplus 0 \times s_1 \oplus 1 \times s_2 \oplus 0 \times s_3 \oplus 0 \times s_4 = 0 \\
&\implies s_2 = 0
\end{aligned}
$$

The previous run gave us $s_0 = s_2$, and this run tells us $s_2 = 0$. This means, $s_0 = 0$. The secret string so far looks like $s = 0s_10s_3s_4$.

We now run the circuit again and assume that we now get 11110. This means:

$$
\begin{aligned}
&11110 \cdot s = 0 \\
&\implies 11110 \cdot 0 s_1 0 s_3 s_4 = 0 \\
&\implies 1 \times s_0 \oplus 1 \times s_1 \oplus 1 \times s_2 \oplus 1 \times s_3 \oplus 0 \times s_4 = 0 \\
&\implies 1 \times 0 \oplus 1 \times s_1 \oplus 1 \times 0 \oplus 1 \times s_3 \oplus 0 \times s_4 = 0 \\
&\implies s_1 \oplus 1 \times s_3 = 0 \\
&\implies s_1 \oplus s_3 = 0
\end{aligned}
$$

Since all digits are either 0 or 1, we deduce that $s_1 = s_3$. We run the circuit once again and, say, get 00111. This means:

$$
\begin{aligned}
&00111 \cdot s = 0 \\
&\implies 00111 \cdot 0 s_1 0 s_3 s_4 = 0 \\
&\implies 0 \times s_0 \oplus 0 \times s_1 \oplus 1 \times s_2 \oplus 1 \times s_3 \oplus 1 \times s_4 = 0 \\
&\implies 0 \times 0 \oplus 0 \times s_1 \oplus 1 \times 0 \oplus 1 \times s_3 \oplus 1 \times s_4 = 0 \\
&\implies s_3 \oplus 1 \times s_4 = 0 \\
&\implies s_3 \oplus s_4 = 0
\end{aligned}
$$

Since all digits are either 0 or 1, we deduce that $s_3 = s_4$. Earlier, we saw that $s_1 = s_3$, and this means $s_1 = s_3 = s_4$. The secret string now looks like $s = 0 s_1 0 s_3 s_4$. Since $s_1 = s_3 = s_4$, and 00000 is not allowed for secret string, we must have $s_1 = s_3 = s_4 = 1$. This means, the secret string $s = 01011$. We can see that we made four calls to our function in deducing this secret string.

There are two questions that we need to further consider.

1. Q1. We used an algorithm to find the secret string s using the output of the quantum circuit, and it seems to work in a specific case. Is there an algorithm that tells us what to do in every case?

2. Q2. The second question concerns how many times we are calling the function f? We saw that in the classical case, in the worst case, we will definitely get an answer after $2^{n-1}+1$ invocations to function f. However, in the case of the quantum algorithm, the worst case is much worse! We saw that we get an answer at random. The answer does have a dot product of 0 with s, but we could get the same answer more than once. It may happen that we run our quantum circuit $2^{n-1}+1$ times, and we get a string of 0s every time. We know that a string of 0s gives no useful information in deducing the secret string, so it is possible that even after $2^{n-1}+1$ invocations to the function, we wouldn't have deduced anything useful about the secret string.

Each time we run the circuit, we get a random string whose dot product with the secret string s is 0. This gives us a linear equation with n unknowns. Running the circuit several times yields several, a system of, equations. In the previous example, every time we got a new equation, we got some useful information as well. These equations are called linearly independent. In order to solve for secret string s, we need a system of $n-1$ *linearly independent* equations. (We actually need n equations to solve a system of n unknowns. This is true when the coefficients are real numbers, but in our case, they are only 0 or 1. With this restriction and the fact that a string of all 0s is now allowed for s, we can reduce the number of equations by 1.) It is known that the number of steps needed to solve a system of n equations is bounded above by a quadratic expression involving n. We say that the system can be solved in quadratic time.

The other question that we need to address is this: How many times do we need to run the quantum circuit? As we pointed out, in the worst-case scenario, we can keep running our qubits through the circuit and never get any useful information. However, this is extremely unlikely. We examine this in the next section.

9.7 Complexity Classes, revisited

Recall that complexity class P denotes the problems that can be solved in polynomial time by classical algorithms. Further, let QP denote the problems that quantum algorithms can solve in polynomial time. These terms are typically used when we refer to the number of steps an algorithm takes. We also defined *query complexity* by counting the number of times the function f is invoked to solve the problem. We saw that the Deutsch-Jozsa problem was not in class P, but belonged to QP for query complexity. We refer to this as saying that the Deutsch-Jozsa problem separates P and QP — it is a problem that belongs to QP but not to P for query complexity.

Let us study a concrete example of a classical algorithm. Consider the case of $n = 10$, meaning, we have a function that takes 10 inputs and is either balanced or constant. We now keep evaluating our function on different inputs until we find the answer. For $n = 10$, we have $2^{10} = 1024$ input choices. The worst-case scenario is when the function is balanced, but we see the same answer for the first 512 evaluations, and then on 513^{th} invocation we get a different answer. What is the probability of this happening? this can be compared to tossing a fair coin 512 times, and getting heads every single time. The probability of this happening is $(\frac{1}{2})^{512}$, which is a minuscule number, even less than $\frac{1}{10^{100}}$.

Assume that we are given a coin and are asked if it is a fair coin or biased. If we toss it once and get a head, it is difficult to answer conclusively. If we toss it, say

10 times, and see heads turn up every time, then we would be fairly certain that the coin is a biased one. We may still be wrong, but we are willing to accept being wrong if the Probability of this happening is very small. We apply the same reasoning for bounded-error complexity classes. We pick a bound on the probability of getting an error that is acceptable, and then we look at algorithms that can answer the question within the set bound of errors.

We now revist the Deutsch-Jozsa problem. Suppose that we want at least a 99.9 percent success rate, or equivalently an error rate of less than 0.1 percent. If a function is balanced the probability of evaluating the function 11 times and getting 0 every time is 0.00049 to five decimal places. Similarly, the probability of obtaining 1 every time is 0.00049. Consequently, the probability of obtaining the same answer 11 times in a row when the function is balanced is just less than 0.001. So if we are willing to accept a bound on the probability of error of 0.1 percent, we can choose to make at most 11 function evaluations. If during the process we get both a 0 and a 1, we can stop and know with certainty that our function is balanced. If all 11 evaluations are the same, we will say the function is constant. We could be wrong, but our error rate is less than our chosen bound. Notice that this argument works for any n. In every case, we need 11 function evaluations at most. Problems that classical algorithms can solve in polynomial time with the probability of error within some bound are denoted BPP (for bounded-error probabilistic polynomial time). The Deutsch-Jozsa problem is in the class BPP.

We now return to Simon's problem. We continue to send qubits through the circuit till we get $n-1$ linearly independent equations. We have seen that in the worst case it can go on forever. Thus, we will say that Simon's algorithm is not in class QP. However, we chose a bound of error that is acceptable, and then we calculate N such that $(\frac{1}{2})^N$ is less than our chosen bound. It can be proven that if we run the circuit $n+N$ times, the probability of the $n+N$ equations containing a system of $n-1$ linearly independent equations is greater than $1-(\frac{1}{2})^N$.

The Simon's Algorithm can be defined as follows. We start by choosing an error bound and then calculate the value of N (this N is independent of n, and we can use the same value of N in all cases). We now run the circuit $n+N$ times. The number of queries is $n+N$ which, and since N is fixed, is a linear function of n. We now assume that our system of $n+N$ equations contains $n-1$ independent vectors. This assumption may be wrong, but the probability of it being wrong is less than the error bound that we have chosen. We now solve this $n+N$ system to equations using classical methods. We know that this will be quadratic in $n+N$, but since N is fixed, this can be said to be quadratic in n.

The algorithm as a whole contains the quantum part that takes linear time added to the classical part that takes quadratic time, giving quadratic time

overall. Problems that quantum algorithms can solve in polynomial time with the probability of error within some bound are denoted BQP (for bounded error quantum polynomial time). Simon's algorithm shows the problem belongs to BQP for query complexity.

We showed that the classical algorithm, in the worst case, took $2^{n-1}+1$ function evaluations — this is exponential in n, not polynomial, so the problem definitely does not belong to P. It can also be shown that even if we allow a bound on the probability of error the algorithm is still exponential, so the problem does not belong to BPP. We say that Simon's problem separates BPP and BQP for query complexity.

9.8 Quantum Algorithms

It is often said that quantum algorithms are faster because of putting input into superposition involving all the basis states. We have looked at three algorithms, and have seen that though we need to use superposition, we need to do much more as well. We now look at what is needed, and more importantly, why it is hard.

The three algorithms we've looked at are the most fundamental and are regarded as standards, although as you've probably noticed, they're far from simple. The times they were published reveal a significant narrative. His algorithm was made public by David Deutsch in his seminal work in 1985. It demonstrated that a quantum algorithm may be quicker than a classical one as the first quantum algorithm. Seven years later, in 1992, Deutsch and Jozsa published their generalization of Deutsch's method. It may seem odd that a generalization that looks to be quite simple took so long to be discovered, but it's vital to understand that the generalization is only apparent because of current notation and presentation. The problem is not described in Deutsch's work exactly as it is above, and it does not make use of the now-usual diagrams for quantum circuits. Nevertheless, the years 1993 to 1995 were a very fruitful time, and this is when many of the most significant algorithms were found. The algorithms by Peter Shor and Lov Grover, which we will examine in the following chapter, as well as Daniel Simon's algorithm, were all published in this window.

Quantum gates are represented by orthogonal matrices. Quantum circuits are made up of gate combinations. Since the product of orthogonal matrices also produces an orthogonal matrix, any quantum circuit can be described by a single orthogonal matrix. These are equivalent to multiplying orthogonal matrices. As we've seen, an orthogonal matrix relates to a change in basis—a new perspective—on the issue. This is the main concept. We have more perspectives on an issue thanks to quantum computing than we do with traditional computing.

But for it to work, there needs to be a vision that clearly distinguishes between the right response and other potential incorrect replies. The only way to see the structure of a problem that quantum computers can answer more quickly than classical computers is to transform it using an orthogonal matrix.

The issues we've looked at are the result of reverse engineering. They are not significant issues that have been debated for years, but we have only just learned that by approaching them from the proper quantum computing perspective, they become easier to resolve. Instead, they are problems that have been specifically designed using the Kronecker product structure of Hadamard matrices. Of course, what we want is to create a quantum algorithm that is quicker than any known classical method for a problem that is important rather than trying to reverse-engineer a problem. This was accomplished by Peter Shor in his seminal article from 1994, which demonstrated, among other things, how quantum computing may be used to crack the codes now in use for Internet security. The implications of quantum computing will be covered in the following chapter when we will briefly explore Shor's algorithm.

Chapter 10

Applications of Quantum Computing

Of course, it is impossible to accurately forecast how quantum computing will affect society in the long run. Nobody could have foreseen how much computers would influence society and how reliant we would become on them if we think back to the 1950s when the first modern computers were invented. There are well-known quotes attributed to early pioneers suggesting that there would only ever be a small number of computers required worldwide and that no one would ever require one in their home. Even though the statements were out of context, and the authors were referring to very specific kinds of computers, the impression that was given was largely accurate. The initial computers were bulky, required a large amount of air conditioning, and were often unreliable. All our current devices like laptops, smartphones, tablets, and even smart-watches are significantly more potent than the original computers. Even futurists like Alan Turing would be astounded by how deeply computers have invaded all spheres of society. Turing did talk about chess playing and Artificial Intelligence, but no one anticipated that the rise of e-commerce and social media would come to control such a large portion of our life.

Quantum computing is in the nearly same state as were computers in the early 1950s. The current quantum computers are large, bulky, not very powerful, and often require superconductors working at extremely low temperatures. There are skeptics already who feel that there is no need of building quantum computers and that their impact on society will be minimal. Though it is difficult to predict the technological landscape fifty years in the future, we can look at developments in quantum computing over the last few years and gauge the directions in which it is heading. In this chapter, we will look at some of these developments. Unlike the last chapter, when we examined three algorithms in great detail, we will now examine a wide range of issues at a less in-depth level.

10.1 Shor's Algorithm and Cryptanalysis

Shor's algorithm is the main quantum computing achievement in cryptanalysis. A solid understanding of mathematics is necessary to completely comprehend this algorithm. It makes use of number theory's continued fraction expansions as well as Euler's theorem. The discrete Fourier transform and complex analysis are also necessary. It indicates the point at which a more comprehensive foundation is needed for the theory of quantum computation as opposed to only basic mathematics. As a result, we won't go into great length on the method, but given its significance, we should at least briefly discuss it.

It is an algorithm with a quantum component and a classical component, similar to Simon's algorithm. The quantum component is comparable to Simon's algorithm. We'll examine the issue that Shor wished to address before providing a succinct statement.

RSA Encryption

RSA (Rivest-Shamir-Adleman) is a public-key cryptosystem that is widely used for secure data transmission. It is also one of the oldest. The acronym "RSA" comes from the surnames of Ron Rivest, Adi Shamir, and Leonard Adleman, who publicly described the algorithm in 1977. An equivalent system was developed secretly in 1973 at GCHQ (the British signals intelligence agency) by the English mathematician Clifford Cocks. That system was declassified in 1997. The security of RSA relies on the practical difficulty of factoring the product of two large prime numbers, the "factoring problem". Breaking RSA encryption is known as the RSA problem. Whether it is as difficult as the factoring problem is an open question. There are no published methods to defeat the system if a large enough key is used. These days, it is frequently used on the Internet to encrypt data exchanged between computers. It is utilized for online banking and credit card purchases for purchases made electronically.

We'll use an example where we want to share some sensitive information with our bank while also wanting to keep it safe from potential listeners to demonstrate how the encryption method functions. Encrypting the data will prevent it from being read if it is intercepted during communication with the bank. A symmetric key, which both you and the bank will need to keep secret, will be used to actually encrypt the data. Symmetric keys can be used to both encrypt and decode data. Although the key is created on your computer and given to the bank, it must first be encrypted before being sent. The key that we'll use to encrypt our communication with the bank needs to be encrypted. Here's when RSA encryption comes into play. It is a method of transferring the key to the bank securely. Let us see how it all works.

Your computer creates the key that will subsequently be used for encryption and decryption for both you and the bank to begin communication with the bank. Let's call this key K.

The bank's computer generates two large prime numbers, which will be referred to as p and q. The size of the primes should be nearly equal, and the modulus, $N = pq$, should have at least 300 digits using normal decimal numbers (1024 binary digits), which is currently thought to be large enough to ensure security. These primes can be produced efficiently, and it is simple to multiply the two primes to obtain the modulus N.

In the second step, the bank identifies a relatively small number e that has neither $p - 1$ nor $q - 1$ as common factors. This is also relatively easy to do. The bank sends the numbers N and e but keeps the prime numbers p and q a secret.

In the next step, your computer takes the key K and raises it to the power e, and takes the remainder after dividing by N. This is also relatively easy to do. The result is denoted by $K^e(modN)$, and is sent back to the bank. Since the bank knows how to factor N into p and q, this allows it to quickly calculate K.

Now consider that there is an eavesdropper who is listening to the communication channel. The eavesdropper will know N and e, as these are sent by the bank. The eavesdropper will also know the number $K^e(modN)$ which was sent by you. To calculate K, the eavesdropper should now find the factors p and q, but these are kept secret by the bank. The security of the whole system is dependent on the fact that the eavesdropper will not be able to brute-force factor the number N into factors p and q.

How hard is it to fact or a number that is a product of two large prime numbers? It seems it is quite hard. There are classical algorithms that can complete all of the other phases in RSA encryption in polynomial time, but no one has yet found one that can factor the product of two huge primes. However, nobody has shown evidence that such an algorithm doesn't exist.

At this point, Shor steps into the picture. He created a quantum algorithm that factors a huge prime number product. The algorithm is of class BQP, which means that it operates in polynomial time with bounded error. This algorithm does not talk about query complexity, but, the total number of steps needed to complete the factorization from start to finish, step by step. Shor provides a detailed algorithm for each step. Because the technique is part of the BQP, if it is used, it will be possible to factor big numbers, and more importantly, if the quantum circuit can be built, RSA encryption will no longer be secure.

Shor's Algorithm

Shor's algorithm is heavily dependent on advanced mathematics, and here we will just skim through it at a high level.

Quantum Fourier transform gate is a key component of Shor's algorithm. This can be loosely considered as a generalized version of the Hadamard gate. In

fact, for one qubit the quantum Fourier transform gate is the same as $\mathbf{H}$. We earlier saw a recursive formula to get the matrix $\mathbf{H^{\otimes n}}$ from $\mathbf{H^{\otimes n-1}}$. We can find a similar relation for the quantum Fourier transform matrix also. The main difference between $\mathbf{H^{\otimes n}}$ and quantum Fourier transform matrix is that the entries in the latter are complex numbers, specifically the complex roots of unity whereas the entries in $\mathbf{H^{\otimes n}}$ are either 1 or -1 which are the square roots of 1. If we look at, say, 4^{th} roots of unity, we again get ± 1 if we restrict ourselves to real numbers. However, there are two more roots if we consider complex numbers. In general, 1 as n complex n^{th} roots of units, and the Quantum Fourier transform matrix on n qubits involves all the $2^n th$ complex roots of unity.

In Simon's algorithm, we used $\mathbf{H^{\otimes n}}$ and all entries were either 1 or -1. When we added the terms, some of the kets were amplified and some were canceled. Shor realized this and applied a similar idea to the quantum Fourier matrix. The only difference this time was that instead of being limited to 1 or -1, the amplitudes covered all the $2^n th$ complex roots of unity. This means the algorithm could detect more types of periods than Simon's algorithm. Let us now see how Shor's algorithm works.

We have the number N and want to factor it into a product of two primes, p, and q. The algorithm starts by choosing a number a such that $1 < a < N$. We now check if a has any factors in common with N. If yes, it is obvious that a is a multiple of either p or q, and it is easy to complete the factorization. If not, we calculate $a(modN), a^2(modN), a^3(modN), \cdots$. Since these numbers are all remainders after dividing by N, they all will be less than N. Further, this sequence of numbers will eventually repeat, and there will be a number r such that $a^r(modN) = a(modN)$. This number r can be considered as a period, and this is the number that the quantum part of Shor's algorithm will compute. Once r is found, classical algorithms can be used to find the factors of N.

Even though the description is at a very high level, it gives an idea of how the quantum part of Shor's algorithm works. The key point is that Simon's algorithm for finding the secret string s can be generalized to find the value of r. Shor's algorithm has been implemented, however only for small numbers. It successfully factored 15 in 2001, and 21 in 2012, and an attempt to factor 35 failed in 2019 due to an accumulation of errors. It is a long way from factoring 300 digit numbers.

Shor's algorithm is compatible with many of the other encryption techniques that have been developed over time, but not all of them. It is now obvious that new cryptographic techniques must be created, and these techniques must be resilient to both classical and quantum computer attacks. There is now a lot of activity in the field of post-quantum cryptography, where new encryption techniques are being researched. Naturally, there is no requirement that these use quantum computing. All we need is for a quantum computer to be unable

to decrypt the encrypted communication. But quantum theories do provide us with strategies for creating secure codes.

The BB84 and Ekert's protocols are two safe quantum key distribution (QKD) techniques that we discussed earlier. Several labs have been successful in launching QKD systems. There are a few businesses that sell QKD systems as well. In 2007, ID Quantique set up a system to secure the transmission of votes between a counting station and Geneva's main voting place during a Swiss parliamentary election, making it one of the first times that QKD was utilized in a real-world situation. Small quantum networks employing optical fiber are now being tested in many nations. It may be possible to connect these via satellite and create a global quantum network in this way. Financial firms are quite interested in this research.

The most remarkable findings to date concern a Chinese satellite that is focused on quantum experiments. It was given the name Micius in honor of a Chinese philosopher who worked in optics. The quantum teleportation we discussed in an earlier chapter took place using this satellite. Additionally, QKD has made use of it. Intercontinental QKD was achieved for the first time when a team in China connected to a team in Austria. The teams began exchanging images as the connection was established. The Chinese side submitted a picture of Micius to the Austrians, and the Austrians replied with a picture of Schrödinger.

10.2 Grover's Algorithm

You have likely heard that one of the many advantages a quantum computer has over a classical computer is its superior speed in searching databases. Grover's algorithm demonstrates this capability. This algorithm can speed up an unstructured search problem quadratically, but its uses extend beyond that; it can serve as a general trick or subroutine to obtain quadratic run time improvements for a variety of other algorithms. This is called the amplitude amplification trick.

The algorithm was invented by Lov Grover in 1996, and like Deutsch's and Simon's algorithms, its speedup over classical algorithms is given in terms of query complexity. For the real-world data, we do have the oracle, and we do need to construct it. Let us first look at the what and how of the algorithm.

Imagine four cards in front of you, all of which are placed face down. We know that one of them is the ace of hearts, and you want to find this card. How many cards should you turn over before you can find the ace of hearts? You might be lucky and turn it over on the first attempt, or you might be unlucky and turn over three cards, none of which is the ace of hearts. This means we know where is the ace card after turning over between one and three cards. On average, we have to turn over 2.25 cards.

Before we explain Grover's algorithm, let us reword the problem. We have four binary strings $00, 01, 10, and 11$. We also have a function f that sends three of these to 0, and the remaining one to 1. We want to find the binary string that is sent to 1. As an example, we might have $f(00) = 0, f(01) = 0, f(10) = 1$, and $f(11) = 0$. The problem now asks how many function evaluations we need to make before we find that $f(10) = 1$. This is the same problem as described earlier in terms of cards, so we know that the answer is 2.25 on average.

As in the case of all query complexity algorithms, we construct an oracle - a gate that encapsulates the function. For the example at hand with four binary strings, the oracle is given below.

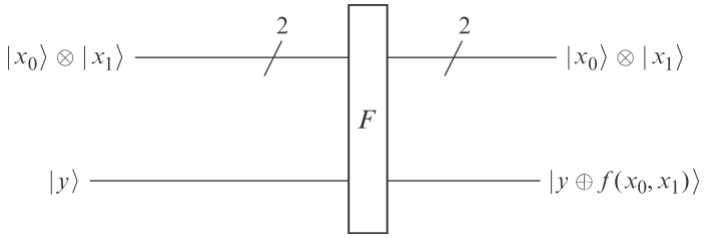

Figure 10.1: Oracle for Grover's Algorithm

The circuit for Grover's algorithm is this.

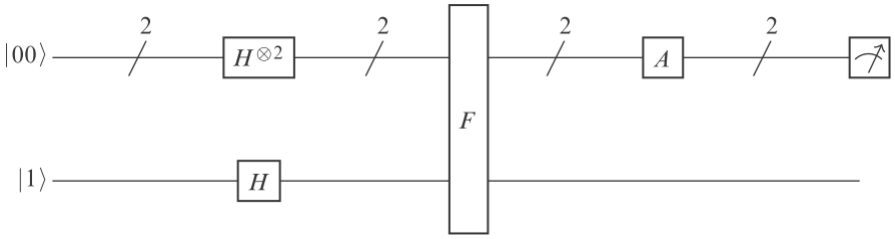

Figure 10.2: Circuit for Grover's Algorithm

The algorithm involves two steps.

1. Flip the sign of the probability amplitude corresponding to the location we are trying to find.
2. Amplify this probability amplitude.

After passing through the Hadamard gate, the top two qubits will be the state $\frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle)$.

The bottom qubit will be in the state $\frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle$. The combined state can be written as:

$$\begin{aligned}
&\frac{1}{2}\left(|00\rangle \otimes \left(\frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle\right)\right. \\
&+|01\rangle \otimes \left(\frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle\right) \\
&+|10\rangle \otimes \left(\frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle\right) \\
&\left.+|11\rangle \otimes \left(\frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle\right)\right)
\end{aligned}$$

The qubits then pass through the F gate that flips $|0\rangle$ and $|1\rangle$ of the third qubit in the location we are trying to find. For our example where $f(10) = 1$, we get:

$$\begin{aligned}
&\frac{1}{2}\left(|00\rangle \otimes \left(\frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle\right)\right. \\
&+|01\rangle \otimes \left(\frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle\right) \\
&+|10\rangle \otimes \left(\frac{1}{\sqrt{2}}|1\rangle - \frac{1}{\sqrt{2}}|0\rangle\right) \\
&\left.+|11\rangle \otimes \left(\frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle\right)\right)
\end{aligned}$$

This can be simplified to:

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

The result is that the top two qubits are not entangled with the bottom qubit, but we have flipped the sign of the probability amplitude of $|10\rangle$, which corresponds to the location we are trying to find. At this point, measuring the top two qubits would yield one of the four locations, with each response having an equal likelihood. We need another trick called *amplitude amplification.*

Amplitude amplification involves flipping a sequence of numbers about their mean. If a number is above the mean, it is flipped below the mean. If a number is below the mean, it is flipped above the mean. In each case, it ensures that the distance to the mean is preserved. An example should make it clear. Consider

four numbers: $1, 1, 1, -1$. Their sum is 2 and mean is $\frac{2}{4} = \frac{1}{2}$. We now go through the numbers one by one. The first is 1 which is $\frac{1}{2}$ above the mean. We flip it about the mean and it becomes $\frac{1}{2}$ below the mean, that is 0. The number -1 is $\frac{3}{2}$ below the mean, and when flipped about the mean, it becomes $\frac{3}{2}$ above the mean, that is 2.

The top two qubits are in the state:

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

If we now flip the probability amplitudes about the mean, we get:

$$0|00\rangle + 0|01\rangle - \frac{2}{2}|10\rangle + 0|11\rangle$$
$$= |10\rangle$$

This means, that when we measure, we will get 10 with certainty. Thus, flipping about the mean has given us exactly what we needed. We need to make sure that there is a gate, or equivalently, an orthogonal matrix that performs the flip about the mean. The following matrix does the same:

$$A = \frac{1}{2}\begin{bmatrix} -1 & 1 & 1 & 1 \\ 1 & -1 & 1 & 1 \\ 1 & 1 & -1 & 1 \\ 1 & 1 & 1 & -1 \end{bmatrix}$$

When this qubit acts on the top two qubits, we get:

$$A\left(\frac{1}{2}|00\rangle + \frac{1}{2}|01\rangle - \frac{1}{2}|10\rangle + \frac{1}{2}|11\rangle\right)$$
$$= \frac{1}{4}\begin{bmatrix} -1 & 1 & 1 & 1 \\ 1 & -1 & 1 & 1 \\ 1 & 1 & -1 & 1 \\ 1 & 1 & 1 & -1 \end{bmatrix}\begin{bmatrix} 1 \\ 1 \\ -1 \\ 1 \end{bmatrix}$$
$$= \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$
$$= |10\rangle$$

For the case of two qubits, we need to invoke the oracle only once. In this case, Grover's algorithm answered with just one question, whereas the classical case takes 2.25 questions on average. The same idea works for higher values of n also. We start by flipping the sign of probability amplitude corresponding to the location we want to find, and then we flip about the mean.

However, for larger values of n, the amplitude amplification is not as significant as in the case of two qubits. Consider the case of eight numbers, seven of which are 1 and one is -1. These will add up to 6, and thus the mean is $\frac{6}{8} = \frac{3}{4} = 0.75$. When we flip about the mean, the 1s will change to $\frac{1}{2} = 0.5$, and -1 changes to $\frac{10}{4} = 2.5$. This means when we have three qubits, and after performing the amplitude amplification, if we measure the qubits, we will get the location that we are trying to find with a higher probability than the other locations. However, there is still a small chance that we might get the wrong answer. We want a higher probability of getting the right answer — we want to magnify the amplitude even more before we measure. The solution is that we send everything back through the circuit. We flip the sign of the probability amplitude associated with the location we are trying to find again and then perform the flip about the mean again.

To summarize, we want to find something that can be in one of the m possible locations. In the classical case, we need to ask $m - 1$ questions in the worst-case scenario. The number of questions thus grows linearly with the size m. Grover calculated a formula for the number of times you should use his circuit to maximize the chance of getting the correct answer. The number given by this formula grows at the rate $\sqrt{m}$. This is a quadratic speedup.

Challenges in using Grover's Algorithm

There are many challenges in implementing this algorithm. First, the quadratic speedup observed is only for query complexity, and we must be careful in constructing the oracle such that the steps involved in the computation of the oracle are not offset by the algorithm thereby resulting in an oveall slower algorithm when compared to the classical case. Second, it is assumed that there is no structure in the underlying data. In case the data is structured, we can generally find classical algorithms that often exploit the underlying structure and result in improved performance. Lastly, the quadratic speedup is not as impressive as the exponential improvement that was seen with other algorithms, and can we do any better?

The challenge of efficient oracle implementation and structure of the data is both real and valid and indicates that Grover's algorithm may not be useful for most real-world database searches. However, in some cases, the structure of data makes it possible to construct an efficient oracle and the algorithm is faster than its classical counterparts. As to the question of doing better than

quadratic speedup, it has been proved that Grover's algorithm is optimal. This means, that no other quantum algorithm can solve the problem any better. Quadratic speedup, though not impressive as exponential speedup, it is useful when the data is at a very large scale.

The main uses of Grover's algorithm will likely not be for the algorithm as it has been presented, but rather for modifications of it. The concept of amplitude amplification is particularly helpful. Shor's and Grover's algorithms are regarded as being the most significant of the few (a detailed list can be found here) that we have described. The concepts in these two have been expanded upon by numerous additional algorithms. We now focus on other quantum computing applications rather than algorithms.

10.3 Chemistry

In 1929, Paul Dirac wrote about quantum mechanics, saying, "The fundamental laws necessary for the mathematical treatment of a large part of physics and the whole of chemistry are thus completely known, and the difficulty lies only in the fact that the application of these laws leads to equations that are too complex to be solved."

Theoretically, every aspect of chemistry involves atom-atom interactions and different electron configurations. We are aware of the underlying mathematics — quantum mechanics — but even though we can write out the equations, we are unable to precisely solve them. In reality, chemists use approximation methods as opposed to attempting to obtain precise solutions. These estimates overlook minute particulars. This strategy has been used in computational chemistry, and it has generally been successful. In many instances, classical computers can provide us with accurate results, yet there are some situations in which they cannot be used. The approximation is sometimes insufficient, and we need the specifics.

Feynman believed that simulating quantum systems would be one of the principal uses of quantum computers. It has enormous potential to research chemistry using quantum computers since it is a subject that belongs to the quantum universe. It is anticipated that quantum computing would have a significant impact in many fields. One of them is to comprehend how *nitrogenase*, an enzyme utilized to produce fertilizers, genuinely functions. The existing process for making fertilizers generates a sizable amount of greenhouse emissions and uses a sizable amount of energy. This and other catalytic reactions may be better understood with the help of quantum computers.

A team at the University of Chicago is researching photosynthesis where sunlight is converted to chemical energy in a quick and effective process. It involves

quantum mechanics. Understanding this procedure and applying it to solar cells are the long-term objectives.

Magnetism and superconductivity are examples of quantum mechanical phenomena. We might learn more about them with the aid of quantum computers. One objective is to create superconductors that don't require cooling to very low temperatures.

Even though genuine quantum computer building is still in its early stages, it is possible to start learning about chemistry with just a few qubits. On a seven-qubit quantum processor, IBM recently simulated the chemical beryllium hydride (BeH_2). With only three atoms, this molecule is quite tiny. The approximations employed in the traditional computing approach are not used in the simulation. However, because IBM's processor only uses a small number of qubits, a conventional computer can imitate the quantum processor. All operations that may be performed on this quantum processor can therefore be performed traditionally. However, when processors add more qubits, we reach a point where a classical simulation is no longer an option.

We now examine some of the techniques being employed to build quantum computers.

10.4 Building Quantum Computers

Decoherence, or the issue of your qubit interacting with something from the environment that is not part of the computation, is the most critical issue that needs to be resolved to build practical quantum computers. A qubit must be placed in an initial state and maintained there until it is required for use. Additionally, you must be able to build gates and circuits. What are the properties of a good qubit?

Photons are advantageous because they are simple to entangle and initiate, interact little with their surroundings, and maintain their coherence for extended periods. However, it is challenging to store photons and have them available when needed. Photons are perfect for communication because of their characteristics, however, creating quantum circuits can be more difficult.

Can electron spin be used instead? Earlier we saw the apparatus used in the loophole-free Bell test that used electrons trapped in synthetic diamonds. These were then manipulated by shining lasers on them. The problem here is related to scaling and generating them in large numbers. You can construct one or two qubits but, at the moment, it is not possible to generate large numbers. Instead of using electrons, spins of the nucleus have also been tried, but scalability is again the problem.

Ion energy levels are another example. Electromagnetic fields are used to hold ions in place for ion-trap computing. Vibrations must be kept to a minimum to keep the ions confined and this is achieved by freezing everything to almost absolute zero. The qubits are encoded by the energy levels of the ions, which are controllable by lasers. David Wineland built the first CNOT gate in 1995 using ion traps, for which he was awarded the Nobel Prize, and in 2016, researchers at NIST successfully entangled more than 200 beryllium ions. Future quantum computers could utilize ion traps, although other methods are also being used to build some of these machines.

Quantum computers are always shielded from light and heat to reduce their environmental impact. They are also cooled and protected from electromagnetic radiation. In these cold environments, some materials can turn into superconductors, losing all electrical resistance. Superconductors have quantum features that can be used in various applications. These involve concepts known as Josephson junctions and Cooper pairs. Cooper pairs are formed when the electrons in a superconductor team up. These electron pairs behave as separate objects. A Josephson junction is created by sandwiching tiny layers of a superconductor between thin layers of an insulator. The work of Brian David Josephson on how Cooper pairs can pass across a Josephson circuit via quantum tunneling won him the physics Nobel Prize. Today, sensitive tools for sensing magnetic fields are made using these junctions in physics and engineering. The crucial point for our purposes is that the discrete energy levels of the Cooper pairs in a superconducting loop with a Josephson junction can be used to encode qubits.

IBM's quantum computers utilize superconducting qubits. In 2016, IBM unveiled a five-qubit processor, which is now freely accessible to everyone on the cloud. As long as it employs five qubits or fewer, anyone can create their own quantum circuit and execute it on this machine. On this machine, circuits for superdense coding, Bell's inequality, and a model of the hydrogen atom have all been tested. Additionally, a crude version of Battleships was played, allowing the programmer the claim of creating the first multiplayer quantum computer game. IBM connected a 20-qubit machine to the cloud at the end of 2017, though this is a commercial venture where companies can buy access.

Google is developing a quantum computing system using superconducting qubits. In 2019, Google announced that it has a 72-qubit computer. What makes this number unique? Classical computers can simulate quantum computers if the quantum computer doesn't have too many qubits, but as the number of qubits increases, we reach the point where that is no longer possible. Google, in 2019, announced that it has reached this number, giving them the right to claim *quantum supremacy* — the first time an algorithm has been run on a quantum computer that is impossible to run or simulate, on a classical computer. IBM, however, is not giving up without a fight. Its team, using some innovative ideas,

has recently found a way to simulate a 56-qubit system classically, increasing the lower bound on the number of qubits needed for quantum supremacy.

We are likely to see spinoffs into other fields as work on creating quantum computers proceeds. No matter how we encode them, qubits are responsive to interactions with their environment. As we gain a better understanding of these interactions, we will be able to devise means for our qubits to measure their surroundings in addition to stronger shields to protect them. Electrons trapped in synthetic diamonds serve as one example. These are extremely magnetic field sensitive. This concept is being used by a startup company, NVision Imaging Technologies, to create NMR equipment that they hope will be superior to present models in terms of performance, speed, and cost.

10.5 Quantum Annealing

D-Wave has quantum computers for sale. Their latest, the D-Wave 2000Q has 2000 qubits. However, these computers are not general purpose, and they are designed for solving certain optimization problems using quantum annealing. Let us briefly look into this now.

Blacksmiths frequently have to bend and hammer metal. It can harden during this process, causing a variety of stresses and abnormalities in the crystal structure that makes it challenging to work with. The homogenous crystal structure can be recovered through traditional annealing, which also makes the metal malleable once more. The process involves heating the metal to a high temperature and allowing it to slowly cool.

Simulated annealing is a standard technique, based on annealing, that can be used for solving certain optimization problems. For example, suppose we have the graph given below and want to find the lowest point — the absolute or global minimum. Think of the graph as being the bottom of a two-dimensional bucket. We drop a ball bearing into the bucket. It will settle at the bottom of one of the valleys labeled A, B, and C.

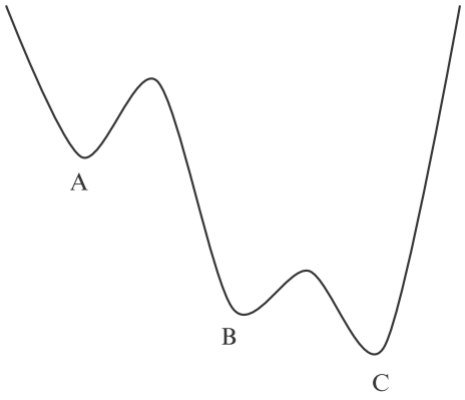

Figure 10.3: Graph of function — bottom of bucket

The ball bearing may not land at the bottom of C, but instead, it might end up at the bottom of the valley at A. The important observation in annealing is that the energy required to push the ball bearing up the hill and let it drop into valley B is much less than the energy needed to push the ball bearing up from B and let it drop into A. So, we shake the bucket with an energy level between these two values. The ball can move from A to B, but it cannot move back. After a while of shaking at this level, it will end up either at the bottom of A or B. But shaking at this level can send the ball from C to B. The next step is to shake it again, but less energetically, with enough energy to get it up the peak from B to C, but not enough to let it get back from C to B. In practice, you start shaking and gradually reduce the energy. This corresponds to gradually cooling your piece of metal in traditional annealing. The result is that the ball bearing ends up at the lowest point. You have found the absolute minimum of the function.

Quantum annealing adds *quantum tunneling*. This is a quantum effect where the ball bearing can just appear on the other side of a hill. Instead of going over, it can go through. Instead of reducing the heights of hills, the ball can climb, you reduce the length of the tunnels it can tunnel through.

D-Wave has produced several commercially available computers that use quantum annealing for optimization problems. Initially, they were met with some skepticism about whether the computers actually used quantum tunneling, but now it is generally agreed that they do. There is still some question of whether the computers are faster than classical ones, but people are buying. Volkswagen, Google, and Lockheed Martin, among others, have all bought D-Wave machines.

We now turn to deeper questions. What does quantum computation tell us about us, the universe, and what computation is at its most fundamental level?

10.6 Quantum Supremacy and Parallel Universes

There are 8 possible three-bit combinations: $000, 001, 010, 011, 100, 101, 110, 111$. If instead of bits we switch to qubits, each of these 8 three-bit strings is associated with a basis vector, so the vector space is 8-dimensional. The same analysis tells us that if we have n qubits, then we will have 2^n basis vectors, and the space will be $2n$ dimensional. As the number of qubits grows, the number of basis vectors grows exponentially, and things quickly get big.

If we have 72 qubits, the number of basis elements is 2^{72}. This is about 4,000,000,000,000,000,000,000. It is a large number and is considered to be around the point at which classical computers cannot simulate quantum computers. Once quantum computers have more than 72 or so qubits we will enter the age of *quantum supremacy* — when quantum computers can do computations that are beyond the ability of any classical computer. Google announced this in 2019. (D-Wave has 2000 qubits in its latest computer. However, this specialized machine has not been able to do anything that cannot be done by a conventional computer, so it hasn't broken the quantum supremacy barrier.)

Let's consider a machine with 300 qubits. This doesn't seem an unreasonable number for the not-too-distant future. But 2^{300} is an enormous number. It's more than the number of elementary particles in the known universe! A computation using 300 qubits would be working with 2^{300} basis elements. David Deutsch asks where computations like this, which involve more basis elements than there are particles in the universe, are done. He believes that we need to introduce parallel universes, each collaborating with one another.

This view of quantum mechanics and parallel universes goes back to Hugh Everett. Everett's idea is that, whenever we make a measurement, the universe splits into several copies, each containing a different outcome. Though this is distinctly a minority view, Deutsch is a firm believer. His paper in 1985 is one of the foundational papers in quantum computing, and one of Deutsch's goals with this work was to make a case for parallel universes. He hopes that one day that there will be a test, analogous to Bell's test, that will confirm this interpretation.

10.7 Quantum Computing

Alan Turing is one of the fathers of the theory of computation. In his landmark paper of 1936, he carefully thought about computation. He considered what humans did as they performed computations and broke it down to its most elemental level. He showed that a simple theoretical machine, which we now

call a Turing machine, could carry out any algorithm. Turing's theoretical machines evolved into our modern-day computers. They are universal computers. Turing's analysis showed us the most elemental operations. These involve the manipulation of bits. But remember, Turing was analyzing computation based on what humans do.

According to Fredkin, Feynman, and Deutsch, calculations are an integral element of physics and are performed by the cosmos. Quantum computation shifts the focus from human computation to universal computation. The 1985 paper by Deutsch ought to be regarded as a seminal work in the theory of computation. In it, he demonstrated that the qubit, rather than the bit, is the essential object.

We've seen that quantum dominance will soon be the norm and that no classical computer will be able to simulate our quantum computers. But what about the opposite? Are quantum computers capable of simulating traditional computers? They certainly can, is the answer. A quantum computer can perform any classical calculation. As a result, quantum computation is more versatile than traditional computation. Instead of being an odd technique to perform a few unique calculations, quantum computations represent a new way to conceptualize computation. The concepts of quantum computation and classical computation are not mutually exclusive. Actually, quantum computation is just computation, and classical computations are simply specific examples of quantum computations. Quantum computing represents a true paradigm shift.

In the future, it will become accepted that there is a more fundamental level of computing, and the most elemental level of computing involves qubits, entanglement, and superpositions. At the moment, the focus is on showing that certain quantum algorithms are faster than classical ones, but this will change. Quantum physics has been around longer than quantum computation. It's now accepted as its own subject. Physicists don't try to compare quantum physics with classical physics and hope to show that it is in some way better. They study quantum physics in its own right. The same shift will happen with quantum computation. We have been given new tools that change the way we study computation. We will use them to experiment and see what new things we can construct. This has started with teleportation and superdense coding, and it will continue.

We are entering a new era, with a new way of thinking about what computation really is. What we are going to discover is impossible to say, but now is the time for exploration and innovation. The greatest years for quantum computation are ahead of us.