# MA469 MATHS IN ACTION: QUANTUM COMPUTING

## Contents

## Preface

Throughout this project one of the most difficult problems I have faced is trying to understand quantum mechanics. There are many different approaches, as will be seen, and many strange "phenomena" that may seem quite simple to describe at first, but actually require some deep understanding. In some areas I have picked out interesting results, explained them further and given examples, whilst not going in to detail in others. This is because certain details will be useful in applications later on whilst others will not. Some, I just find interesting. I hope, however, that I have provided enough quality references so that the reader can consider these further.

Much of the section on quantum mechanics relies on an undergraduate level of understanding of linear algebra. There are some new results and some restatements of definitions or theorems which are particularly useful or provide a useful tool for understanding.

Some of the sources I have used to compile this report have been very recent. For example, the book by Mikio Nakahara and Tetsuo Ohmi [1] was released after I began writing this report. This shows how quickly the field of quantum computing is advancing. New texts are becoming available regularly.

I have created every diagram in this report. Most are based on existing diagrams and in these cases I have referenced them.

## 1. Introduction

1.1. **Motivation.** Throughout history we have seen advancements made through engineering and based on the work of scientists, who structure their ideas upon mathematical foundations. The Romans constructed aqueducts; a civil service which helped the Romans grow to be one of the most famous civilisations in history. These waterways were built as directed in works "De Architectura"[2], a first century BC work by Marcus Vitruvius. In the opening line of his work, Vitruvius describes his subject:

> "Architecture is a science arising out of many other sciences"

In his work he discusses the importance of theory and mathematics in creating his guide to engineering.

In the first century BC experiments were relatively simple and thus the mathematics to model these experiments was relatively simple. What Vitruvius did show however, was that these experiments conducted on a small scale, along with some thought, could be used to build giant structures and ensure they fit their purpose. It also highlighted boundaries.

Work by engineers today often comes very close to the physical boundaries of what can be done, resulting in amazing structures, smaller computer processors or faster cars; just to name a few examples. At some point however we do arrive at these boundaries. This does not mean to say that we must stop. A new technology or approach may break these boundaries. Quantum computing breaks the boundaries set by the classical model of the universe by accepting a new understanding of the way the world works at a very small level.

Many of the technological advancements of the last 200 years have been preceded by many years of speculation and theoretical interest in the area. The wright brothers may be accredited with building the worlds first successful aeroplane but this is not to say Leonardo Da Vinci had not tried around 400 years earlier.

Much of the work done in the area of quantum computing is theoretical. You can count the number of real-life quantum computers on the fingers of just one of your hands. The progress made in constructing quantum computers seems to be very slow, but this is not a valid reason to stop studying the area. Without the work of mathematicians like Peter Shor, the motivation to construct a quantum computer would be have half the strength it has.

Thanks to his algorithm for factoring large numbers, we now know that anyone who is able to build a quantum computer, with less than a millionth of the power and space of a modern computer, is able to break almost all security used to transfer data over the Internet. This means that online shopping, Internet banking and accessing a remote computer could no longer be safe. There is of course the potential for criminals to make a lot of money and is therefore why it is of great interest for national security organisations.

Lov Grover found a quantum algorithm for speeding up database searches in 1996. Databases lie behind many of the systems in place today. Even logging into your email requires a database search. This algorithm would offer a huge boost to companies who rely on large databases, saving them money in time lost searching the databases, and in shrinking power-hungry computers needed to perform the searches.

If we now consider the Internet. Without the work into graph theory, data compression and communication protocols, it would be more of a disconnected, slow speed failure than the success it is today. It is, of course, not often that we hear the names of these people. However, their work was fundamental in the rapid and successful construction of this world wide network. Is it possible that quantum computing could be this successful?

There are many hurdles to overcome. Quantum systems need to be completely independent of their surroundings. This currently means cooling them to temperatures colder than outer space and keeping them in a vacuum. They require expensive machinery and a lot of space. This was the case when modern computers came into being in the 1940s. Now, a computer with many more features and much more power fits in your pocket and runs on batteries. Our progress towards the computers we have today was generally continuous with a few notable leaps. Although it is impossible to say how the story of the quantum computer will unfold, we do know that it is guaranteed a huge market and most likely a place in everybody's home and work place.

Even if we do find that quantum computers won't quite fit in every home or that it really is impossible to cheaply produce quantum computers, we can still enjoy an exploration into the unusual world of quantum mechanics and maybe we will find some possibilities for new technology. As we shall see, there is already one product directly linked with quantum computing on the market. The probabilistic nature of quantum mechanics makes its study very challenging. As Dirac[3] and Dicke [4] explain; "Why should Newtonian mechanics hold outside the region of space and time we see", "Our brains evolved not to like quantum mechanics... [we believe] things are linear and they either exist or don't exist".

The study of quantum mechanics, however challenging it is, is also very rewarding. Even if it was instantly proven that our quantum mechanical model was incorrect tomorrow, the study is very intriguing from a mathematical point of view. Thus, even just for the rewards pure mathematics brings, it is worth building a theoretical quantum computer, and this shall be one of the main themes running through this report.

In regards to the structure of this report, we shall take a fresh look at classical mechanics in order to prepare us for an exploration into quantum mechanics. We will not consider all of quantum mechanics; just the bits we need. Then, we can take a shot at constructing a theoretical quantum computer and look at some problems we can solve with it. We shall end with a few interesting "quantum corollaries".

1.2. **A Glimpse at the History of Quantum Computing.** To begin our exploration into quantum computing we take a look back at how the subject came into being. In 1965 Gordon Moore, co-founder of Intel, wrote an article[5] discussing a trend he had noticed in integrated electronics production. The number of transistors in electronic devices had been doubling and Moore predicted this would continue into the future. The prediction has held well since then and by 2005 there were over 1,000,000,000 transistors on new processors[6]. Moore's law affects other components of a computer system. We have seen exponential increases in disk space, memory and processing amongst others. Transistors will have to continue to get smaller if Moore's law is to hold, and as Moore himself admits;

> "In terms of size [of a transistor] you can see that we're approaching the size of atoms which is a fundamental barrier."[7]

Indeed, in ten to twenty years we will need to be producing devices which work on the quantum level. To do this effectively we need to create computers that work within the framework of quantum mechanics.

It is generally accepted that the idea of the *quantum computer* was first suggested by the famous physicist Richard Feynman in 1982. This followed work by Paul Benioff which suggested how to build a machine in the quantum world that could mimic a normal computer[8]. However, this would not take advantage of some of the useful quantum phenomena that could speed our computers up "exponentially". Computers using these phenomena in their computations were given the name *Quantum Computers*, whilst computers relying on classical mechanics were coined *Classical Computers*. In 1985, David Deutsch finished the task of creating a theoretical quantum computer and showed that there are fundamental processes a quantum computer could undertake, which

are unperformable by classical computers[9]. It was not until 1994, when Peter Shor released his Quantum Factoring Algorithm[10], that the world finally saw how beneficial quantum computing could be. Shor's algorithm explained how to use a quantum computer to factor a large number into its prime factors in much less time than would be possible on a classical computer.

Since Shor's algorithm was made known to the public, there has been massive investment in research into the engineering of a quantum computer. Building a quantum computer has proven to be a difficult task, with only two notable quantum computers being built by Intel and D-Wave systems. We will look into these further near the end of this report.

## 2. Some Ideas About Classical Mechanics

In order to look into quantum mechanics it is first helpful to put forward a few ideas about classical mechanics. The progression of these ideas here will be similar to the progression of ideas when we study quantum mechanics. Thus, they should provide motivation for the route taken later and hopefully keep the reader on track.

2.1. **States.** In the world around us we see all objects three dimensions - If there are no restrictions on these objects then their positions have three degrees of freedom. All possible positions can be written in terms of three numbers. Choosing three suitable directions and an origin we can write the position of any point in space in terms of these directions - a *linear combination* of these directions, or *basis vectors*. If we fix the directions and the origin we find these numbers are unique to each point, giving us an isomorphism between the set of points in space and an ordered collection of three numbers which we call a *vector in three dimensions*. If we vary the basis vectors we originally chose, we can find new numbers to describe the same point in space.

From the evidence we have seen throughout our lives we would assume these numbers form a continuum and thus we should let the numbers be numbers along the real line. We come to the conclusion that from our experience the basic space around us should be modeled in $\mathbb{R}^3$.

We would like to be able to move across this space, compare points in it and build up more complex structures. So we introduce operators on the space to let us "add" and "subtract" points, move a point away from, or closer to, the origin (multiplication by a scalar) and move the point with respect to the origin (multiplication by a matrix). We find it useful to be able to project these vectors onto other vectors and find out how these two vectors compare. This is normally done with the scalar product (an example of an inner product). We end up with a structure we call an inner product space.

**Definition 2.1** (Inner Product Space)**.** An inner product space is a vector space with the additional structure of an inner product.

*Remark.* The definitions of terms used above can be found in any good book on linear algebra, for example [11].

To describe our points in space we only require a three dimensional space over the real numbers. Suppose we have a particle which we define, for the sake of simplicity, to be so small that it occupies only one point in space. The position of this particle can be any one of the points in our space but it can also have a number of other properties. For example, we may wish to give it a velocity; in which case we would require an extra three dimensions to fully describe the particle. We may have a particle which changes colour, in which case we should add an extra dimension to describe this (We may not need all of this extra dimension, for example if we parameterised its colour by the wavelength of light it reflects). Depending on the number of properties we wish to give this particle we will need an $n$ dimensional space, which we label $S$. In any experiment that we might wish to conduct on this space we will be required to prepare our *system* into some *state* $s_0 \in S$. We may need to place some particles in a particular position and set them moving at certain velocities. Any preparation we do can be described as a vector $s_0 \in S$. We call this a state. Over time the state may change, and we may introduce things to our experiment which change this state. However, if our space describes everything in our system, then we can always attach a vector to this state.

2.2. **Measurements.** The most important part of any experiment is having the ability to measure properties of the system. We may wish to measure position, velocity, energy, or anything else permissible in our state space. Some of these measurements may be direct measurements of part of our state vector. For example, if in $S = \mathbb{R}^3$, a particle is placed at a specific point, measuring position simply means extracting the state vector directly. Other measurements however are functions of the state vector. For example, if we wish to measure the distance of the particle from the origin, we apply the function $d : S \to \mathbb{R}$, where

$$d\left((x_1, x_2, x_3)\right) = \sqrt{x_1^2 + x_2^2 + x_3^2}$$

In the case of measuring the position directly, we are again applying a function. It is simply the identity function $i : S \to \mathbb{R}^3$, where

$$i\left((x_1, x_2, x_3)\right) = (x_1, x_2, x_3)$$

When making these measurements in classical mechanics, we always assume that we can make the measurements without disrupting the state. For example, if we wish to measure the position of a particle, we could take a ruler and measure the distance from the origin and the angles from our basis vectors that we took the measurement. We just have to be careful not to nudge the particle. We will see later that quantum mechanics says that we can not always do this - the measuring device always plays a part in the system, and its presence will affect future measurements made on the system. In fact these measurements cause an unpredictable (up to a point) change of state. This means that our quantum system is undeterministic in general - we cannot predict exactly what will happen in the future or figure out exactly what has happened in the past. Classical mechanics assumes that the world is completely *deterministic*. If we know the exact state of a system, we can predict all its future states and figure out all its previous states. This is why in the following we are allowed to consider the functions discussed for all times $t \in \mathbb{R}$.

2.3. **Time Evolution.** So far we have looked at the state of a system at a single time. However, systems in general change with progression in time. Progression in time does not introduce any new states (particles do not seem to appear or disappear as time progresses), and thus our state space does not change as time progresses. Any chance that our states could have moved outside the state space should have been taken account of in constructing the state space. In the example above with a particle in three dimensional space, we gave it enough room to move by considering its position to be valid anywhere in $\mathbb{R}^3$.

Supposing we have our state at time $t_0$ as a vector $s_o \in S$, then we can write any future state $s(t)$ as a function $f_t : S \to S$, so that $s(t) = f_t(s_0)$. We should impose different constraints on $s(t)$ depending on the system we are in. For our example of a particle in space, we should at least have $s(t)$ continuous otherwise our particle could jump from one place to another, and this does definitely not agree with evidence of the world around us! In general, it also true that $s(t_1 + t_2) = f_{t_1}(f_{t_2}(s_0))$ as our physical laws do not change as time progresses. We also find that certain conservation laws hold. We have for example the conservation of energy and the conservation of momentum. As we discussed earlier, these are both functions of state space, and thus for these functions, say $E : S \to \mathbb{R}$ and $M : S \to \mathbb{R}$ respectively, we should have

$$E(s(t)) = E(s_0)$$
$$M(s(t)) = M(s_0) \quad \forall\, t \in \mathbb{R}$$

The final thing we will discuss before studying quantum computing, is the idea of a *machine*. A machine has a purpose that it is built to fulfill. The machine may, for example, measure

temperature and tell a heating system when to turn on because it's too cold, or turn off because it's too hot. This machine is a thermostat. It takes an input, which it processes so that it can output some signal to the heating system. Our machine may be a very simple one, but it is also very useful in that it means it is possible to keep your house at a reasonable temperature without any trouble.

In general, a machine will take some inputs, process them and then output a result.

## 3. Constructing a State Space

We will find the journey from understanding classical mechanics to accepting quantum mechanics much like the journey from understanding the real numbers to accepting the complex numbers. Most overtly, our state space moves from the field of real numbers to a space occupied by complex numbers. Our use of quantum mechanics may seem trivial and awkward at first, but its true power will be seen in later sections and hopefully by then the reader will be more confident in its use. We will see a system which is quick and relatively easy to use evolve and the notational simplicity we gain will help us solve problems without too much trouble.

To help us construct the space inhabited by quantum particles we use the example of a photon and the quantum phenomena we observe with respect to its polarisation. In sections 3.1 and 3.2 I have constructed some formalism for ideas that I have seen in other places so that there is some motivation for the later work on constructing the mathematics. The polarisation section was mainly developed from the wikipedia entry on polarisation [12] and for the section on the Mach-Zehnder interferometer I found the article [13], which can be found at *http://stacks.iop.org/0031-9120/35/46*, to be a well written and relatively non-mathematical source. I have however developed section 3.2 and especially section 5.2 further than I have seen in any source.

### 3.1. **Polarisation State.**

In 1864, Maxwell postulated that light was composed of wave-like electric and magnetic fields and gave equations (the famous Maxwell equations) showing how a moving electric field generates a magnetic field and vice versa[14]. To look at what we mean by the polarisation of light, we restrict ourselves to considering the electric field. We look at how the direction of the electric field changes as we move along the direction in which the light is traveling.

If we have both components of the electric field oscillating in time as we move along the direction of travel (which we will take to be the direction $z$), we see linear polarisation as in figure 3.1. In this figure the direction of the electric field always lies in a plane which contains the direction of

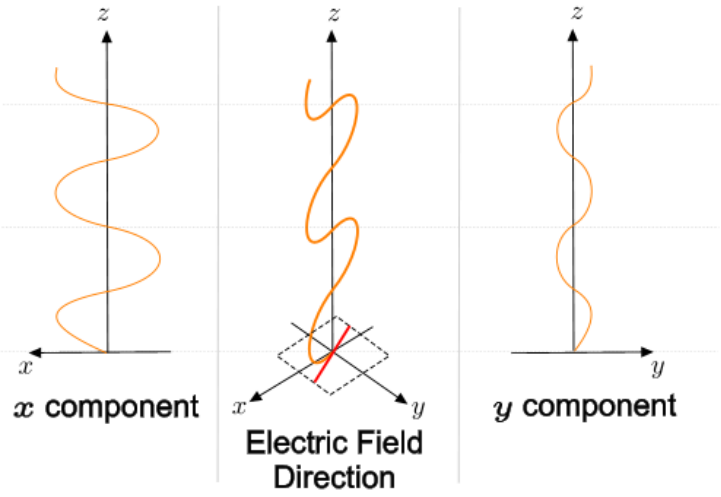


FIGURE 3.1. Linear polarisation with the electric field forming a sinusoidal wave as it moves in direction $z$. This figure is based on the wikipedia entry on polarisation.

travel. We do not necessarily have to have the amplitudes of each component of the field equal, we just require that they are *in phase*.

Now we keep the frequencies of both components of the oscillation the same but move them out of phase. We find that as we move along the direction of travel, the electric field traces out an ellipse in the $x$-$y$ plane (shown by the red line in figure 3.2). We can write the sinusoidal wave, with a fixed frequency corresponding to $\omega$, in each component as

$$\begin{aligned} x =& A_x \sin(\omega z + \theta_x) \\ y =& A_y \sin(\omega z + \theta_y) \end{aligned}$$

We note that the frequency and amplitude of the photon is invariant in most experiments (and all experiments we shall be considering in this report). Taking the frequency to be constant means that we can, without loss of generality, assume that $\omega = 1$. We can then describe each component $j = x, y$ by a *phase angle* $\theta_j$ and a real number $A_j$, or equivalently encode these both into a complex number $c_j = A_j e^{i\theta}$. Following the notation used originally by Dirac[3], we can then write our photon as $c_x |x\rangle + c_y |y\rangle$, or as is more common, $c_H |H\rangle + c_V |V\rangle$ (we have just relabeled $x$ to $H$ and $y$ to $V$). Next we impose the constancy of amplitude on the previous expression. It is easiest

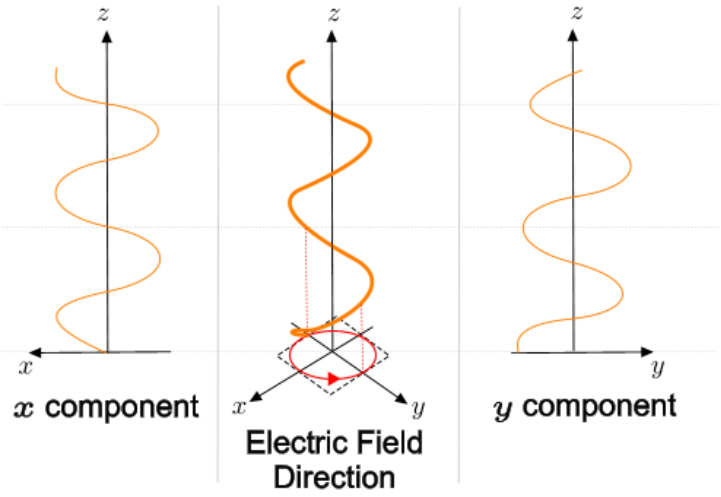


FIGURE 3.2. Circular polarisation with the electric field moving along a helix as it moves in direction $z$. This figure is based on the wikipedia entry on polarisation.

(and as we shall see later, useful) to consider the photon having unit amplitude, thus satisfying

$$|c_H|^2 + |c_V|^2 = 1$$

We can see why this should be true. We consider the sum of the squares of the amplitudes of the two components. Pythagoras tells us that this quantity must be the same as the square of the amplitude of the diagonal (i.e. the amplitude of the photon).

The notation used here is called *bra-ket notation*, with the quantities $|H\rangle, |V\rangle$ called *kets* and the quantities $c_H$,$c_V \in \mathbb{C}$ called *phase factors*. We will leave defining exactly what this means until later, but for now we can see the expression $c_H |H\rangle + c_V |V\rangle$ encodes all we need to know about a photon with varying polarisation. We call this expression the *polarisation state* of the photon. We continue by considering how the polarisation state affects an experiment.

3.2. **The Mach-Zehnder Interferometer.** Suppose we set up an experiment as in figure 3.3, with a photon source capable of emitting a photon so that it follows the paths shown. A half-silvered mirror, which is made up of block of glass and a half-silvered surface, is designed so that it lets photons through with probability $1/2$ and reflects photons with probability $1/2$. Both half silvered mirrors have blocks of glass of the same depth behind them. The figure shows the path of the photon being split by a half-silvered mirror, then the paths being reflected so that the two

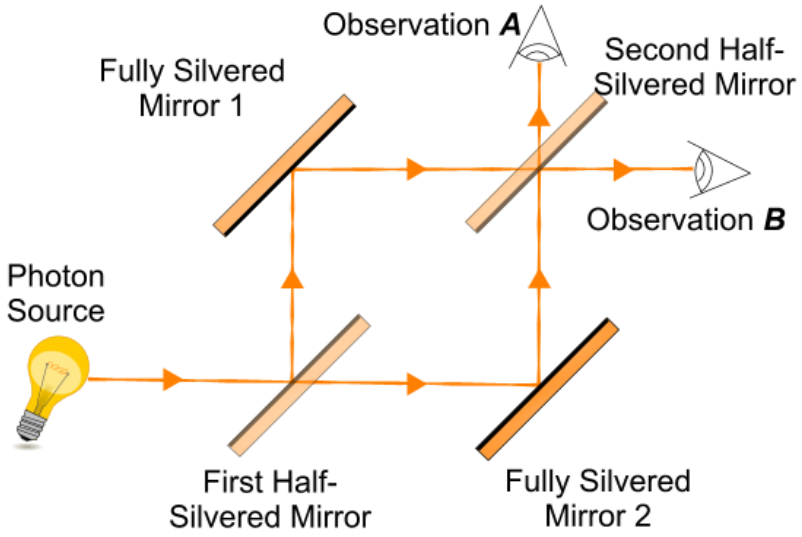


FIGURE 3.3. Deutsch's example of a quantum phenomenon[9]

paths the photon can take meet again at the second silver mirror. We place two detectors as shown so we can observe what has happened to the photon.

We understand that a photon hitting the first half-silvered mirror has probability $1/2$ of taking the path up to mirror 1 and probability $1/2$ of continuing on to mirror 2. Considering the path which touches mirror 1, we would think that any photon taking this path would then hit the second half-silvered mirror and we would see the photon at $A$ and $B$ with equal probability. Similarly, any photon which has taken the other path and has been reflected off mirror 2 would be observed at $A$ or $B$ with equal probability. So overall we would expect any photon being emitted from our source to be observed at $A$, 50% of the time and at $B$, 50% of the time. This is not the case.

Experiments show that all photons from the photon source are detected at $B$; no light makes it to an observer at $A$. It may seem counter-intuitive, but if we take into account the following facts and remove the assumption of *locality*, we can show why this is the case.

**Fact.** *Light which is reflected by a denser material than the one it is traveling through undergoes a phase angle change of half a wavelength, equivalent to $\pi$ radians.*

**Fact.** *When light travels through glass its phase is retarded by an amount linearly dependent on the depth of the glass through which it travels.*

Our assumption of locality here is that the photon must exist in one place at each moment in time. Quantum mechanics removes this assumption and allows the photon to travel both paths after the first half-silvered mirror. Both at the same time. The photon is somehow split in two. It is only when we choose to check whether the photon is traveling one of these paths (for example by putting a detector along one of these paths) that the delocalised photon suddenly becomes localised to either one path or another. It seems that the photon is split so that its spread over the two paths can be described by a probability distribution.

In the experiment above we find that we are getting constructive and destructive interference of the photon with itself. As the photon takes each path, we find that the phase is changed differently along each path according to the rules above. When the paths cross again an *interference* process occurs, "summing" the polarisations of the light from each of the paths to give light with new polarisations traveling along the new paths. We find what is called *constructive interference*: when the polarisations are such that the electric field oscillations complement each other and we get larger oscillations. In *destructive interference*, some (or possibly all) of the electric field vectors cancel each other and we end up with smaller oscillations (or no oscillations) compared with before the interference occured. From the way we have defined our polarisation state, it is easy to see that interference simply occurs from summing our phase factors for each $|H\rangle$ and each $|V\rangle$.

Let us assume the photon begins in the polarisation state $A_H e^{i\theta_H} |H\rangle + A_V e^{i\theta_V} |V\rangle$. Taking into account the two facts above, we look at what happens to the photon's polarisation state as we travel each of the paths. We label the path taken which touches mirror 1, path 1, and the path which touches mirror 2, path 2. We assume that as photon passes through the glass behind the half-silvered mirrors, its phase angle changes by some constant $c$. We get the following table giving the results of the phase changes.

| Distance along Path | Polarisation state along path 1 | Polarisation state along path 2 |
|---|---|---|
| Immediately after the first silvered mirror | $A_H e^{i(\theta_H+\pi)} \lvert H\rangle + A_V e^{i(\theta_V+\pi)} \lvert V\rangle$ | $A_H e^{i(\theta_H+c)} \lvert H\rangle + A_V e^{i(\theta_V+c)} \lvert V\rangle$ |
| Immediately after each mirror | $A_H e^{i(\theta_H+2\pi)} \lvert H\rangle + A_V e^{i(\theta_V+2\pi)} \lvert V\rangle$ | $A_H e^{i(\theta_H+\pi+c)} \lvert H\rangle + A_V e^{i(\theta_V+\pi+c)} \lvert V\rangle$ |
| Just before the semi reflective surface of the second half-silvered mirror | $A_H e^{i(\theta_H+2\pi+c)} \lvert H\rangle + A_V e^{i(\theta_V+2\pi+c)} \lvert V\rangle$ | $A_H e^{i(\theta_H+\pi+c)} \lvert H\rangle + A_V e^{i(\theta_V+\pi+c)} \lvert V\rangle$ |

Now, light from path 1 is hitting the half-silvered mirror from the glass (which is denser than air) side. Its phase remains unchanged whether it is reflected or travels through the mirror, so it has polarisation state

$$A_H e^{i(\theta_H+2\pi+c)} \lvert H\rangle + A_V e^{i(\theta_V+2\pi+c)} \lvert V\rangle = A_H e^{i(\theta_H+c)} \lvert H\rangle + A_V e^{i(\theta_V+c)} \lvert V\rangle$$

heading towards either detector. The light from path 2 is either reflected towards $B$, with polarisation state

$$A_H e^{i(\theta_H+2\pi+c)} \lvert H\rangle + A_V e^{i(\theta_V+2\pi+c)} \lvert V\rangle = A_H e^{i(\theta_H+c)} \lvert H\rangle + A_V e^{i(\theta_V+c)} \lvert V\rangle$$

Alternatively, it travels through the half-silvered mirror's surface, retaining its polarisation state

$$A_H e^{i(\theta_H+\pi+c)} \lvert H\rangle + A_V e^{i(\theta_V+\pi+c)} \lvert V\rangle = -A_H e^{i(\theta_H+c)} \lvert H\rangle - A e^{i(\theta_V+c)} \lvert V\rangle$$

Along each of these paths the two parts of the photon interfere, so that along the path to $A$, we get a wave with polarisation state

$$\left[A_H e^{i(\theta_H+c)} \lvert H\rangle + A_V e^{i(\theta_V+c)} \lvert V\rangle\right] + \left[-A_H e^{i(\theta_H+c)} \lvert H\rangle - A e^{i(\theta_V+c)} \lvert V\rangle\right] = 0$$

We get no light at all traveling towards detector $A$.

Along the path to $B$, we find constructive interference and get a wave with polarisation state

$$\left[A_H e^{i(\theta_H+c)}\left|H\right\rangle + A_V e^{i(\theta_V+c)}\left|V\right\rangle\right] + \left[A_H e^{i(\theta_H+c)}\left|H\right\rangle + A_V e^{i(\theta_V+c)}\left|V\right\rangle\right]$$
$$= 2\left[A_H e^{i(\theta_H+c)}\left|H\right\rangle + A_V e^{i(\theta_V+c)}\left|V\right\rangle\right]$$

We are almost there. We have found that at $A$ we have no light, as our experiment showed. However, at $B$ we seem to have twice the photon we did before and we have violated the law of energy conservation. We have forgotten to take into account the probability distributions after the photon is split by the half-silvered mirrors. As we found before, when we considered the photon as a particle, the photon reaches detector $A$ with probability $1/2$ and detector $B$ with probability $1/2$. In some sense, our phase factor and our probability must be linked. Multiplying our phase factors, along the path to $B$, both by the probability $1/2$ we find we end up with our original photon with a constant phase angle adjustment:

$$A_H e^{i(\theta_H+c)}\left|H\right\rangle + A_V e^{i(\theta_V+c)}\left|V\right\rangle$$

This experiment shows why we must consider the wave-like properties (sometimes called the internal state) of the photon as well as the particle-like properties (sometimes called the external state) of the photon. We have given a flavour of how quantum mechanics works and using simple probabilistic arguments we were able to bring our model into agreement with experiment by considering the photon as both a wave and a particle. However we did not, for example, consider the mathematics describing exactly how the photon was probabilistically "split" after each the first half-silvered mirror. Schrödinger's wave equation gives us these answers, telling us exactly how probability is distributed and taking into account the phase factors of polarisation states and more general states. We will not however look directly at the wave equation, but we will now formalise what we have just seen, following the work done by Dirac in [3] and following a similar route to [1].

3.3. **Bra-ket Notation.** Dirac takes us through the general process of discovering what the relationship is between phase factors and probability amplitudes. Following the notation used originally by Dirac, we write a column vector, in the $n$ dimensional vector space $\mathbb{C}^n$, with components $x_1, x_2, \ldots, x_n \in \mathbb{C}$, as a *ket vector*

$$\left|x\right\rangle = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \in \mathbb{C}^n$$

I have taken the following two assumptions almost directly from Dirac's book [3]. We assume that at a particular time each state of a dynamical system corresponds to a to a ket vector. We make the further assumption that if this state results from the superposition of other states, its ket vector is linearly dependent on the ket vectors corresponding to the other states.

Thus $\mathbb{C}^n$ is our *state space*. For now we will just allow vectors in this space to be summed and multiplied by scalars, so that $\mathbb{C}^n$ take its standard vector space form. This gives rise to the idea of linear functions, and the space they occupy.

We introduce the idea of a *dual vector space*. This is the space formed by the functions $f : \mathbb{C}^n \to \mathbb{C}$ which operate linearly on vectors in the vector space. We note that there is a one to one correspondence between linear functions $f : \mathbb{C}^n \to \mathbb{C}$ and 1 by $n$ matrices (i.e. row vectors). Thus we can write the dual space of $\mathbb{C}^n$ as the space of row vectors $(\alpha_1, \alpha_2, \ldots, \alpha_n)$ with $\alpha_1, \alpha_2, \ldots, \alpha_n \in \mathbb{C}$, which we label as $\langle\alpha| = (\alpha_1, \alpha_2, \ldots, \alpha_n)$. The vector $\langle\alpha|$ is called a *bra vector*. We come to two important definitions

**Definition 3.1** (Dual Space of $\mathbb{C}^n$)**.** The *dual space* of the vector space of ket vectors is the vector space of bra vectors over $\mathbb{C}$. We label this space $\mathbb{C}^{n*}$.

**Definition 3.2** (Inner Product of a Bra Vector and a Ket Vector)**.** The inner product of a bra vector $\langle\alpha|$ and $|x\rangle$ is

$$(\langle\alpha|)(|x\rangle) := \sum_{i=1}^{n} \alpha_i x_i$$

For brevity we shorten the inner product to $\langle\alpha|x\rangle$, so that $\langle\alpha|x\rangle \equiv (\langle\alpha|)(|x\rangle)$

It is easy to verify that the inner product is linear in the ket vector: fixing $\langle\alpha|$, we see that

$$\begin{aligned}(\langle\alpha|)(c_1|x\rangle + c_2|y\rangle) &= \sum_{i=1}^{n} \alpha_i c_1 x_i + \alpha_i c_2 y_i \\ &= c_1 \sum_{i=1}^{n} \alpha_i x_i + c_2 \sum_{i=1}^{n} \alpha_i y_i \\ &= c_1 \langle\alpha|x\rangle + c_2 \langle\alpha|y\rangle\end{aligned}$$

Now we define a linear isomorphism between our ket and bra vectors. Let $\varphi : \mathbb{C}^n \to \mathbb{C}^{n*}$ be defined as taking the linear function taking the complex conjugate transpose of the ket in the domain. i.e. For $|x\rangle = (x_1, x_2, \ldots, x_n)^t \in \mathbb{C}^n$,

$$\varphi(|x\rangle) = (\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_n) =: \langle x| \in \mathbb{C}^{n*}$$

The superscript $t$ denotes the transpose of the vector and the bar operator gives the complex conjugate of its operand ($\overline{(a+bi)} = a - bi$, $i = \sqrt{-1}$). We have also defined the bra vector with the same label to be the complex conjugate transpose of the ket vector with the same label. In short hand we write

$$\langle x| = |\bar{x}\rangle^t \quad \equiv \quad \langle x| = \varphi\left(|x\rangle\right)$$

and

$$|x\rangle = \langle\bar{x}|^t$$

With this isomorphism, we can say that $|x\rangle$ and $\langle x|$ correspond to the same quantum state. This may seem unusual, but it allows us to develop many further properties easily.

**Theorem 3.3.** *The space of ket vectors, $\mathbb{C}^n$, with the inner product $\langle\cdot|\cdot\rangle$ forms an inner product space.*

*Proof.* Above we did not prove that the operator we said was an inner product was actually an inner product. We do this here. We have already shown the inner product is linear in the second component. We now show positive-definiteness ($\langle x|x\rangle \in \mathbb{R}_{>0} \Leftrightarrow |x\rangle \neq 0$).

$$\langle x|x\rangle = \left(|\bar{x}\rangle^t\right)|x\rangle = \sum_{i=1}^{n} \bar{x}_i x_i$$

The product of a complex number and its conjugate is a real number not less than zero. So the sum is also real and greater than zero unless every component is zero, in which case $|x\rangle = 0$. The final property of an inner product is that conjugate symmetry holds. For all $|x\rangle, |y\rangle \in \mathbb{C}^n$,

$$\langle x|y\rangle = \left(|\bar{x}\rangle^t\right)|y\rangle = \sum_{i=1}^{n} \bar{x}_i y_i = \overline{\left(\sum_{i=1}^{n} x_i \bar{y}_i\right)} = \overline{\left(\sum_{i=1}^{n} \bar{y}_i x_i\right)} = \overline{\left(\left(|\bar{y}\rangle^t\right)|x\rangle\right)} = \overline{\langle y|x\rangle}$$

To finish the proof we note that by assumption our space of ket vectors is a vector space over $\mathbb{C}$. □

From defining our space in this way, we get, as is shown in many linear algebra books (for example [11]), the following properties.

**Proposition 3.4.** *The inner product is sesquilinear:*

$$\langle x|c_1 y_1 + c_2 y_2\rangle = c_1\langle x|y_1\rangle + c_2\langle x|y_2\rangle$$
$$\langle c_1 x_1 + c_2 x_2|y\rangle = \bar{c}_1\langle x_1|y\rangle + \bar{c}_2\langle x_2|y\rangle$$

*where* $|c_1y_1 + c_2y_2\rangle \equiv c_1 |y_1\rangle + c_2 |y_2\rangle$. *Note that we proved the first equation earlier.*

**Definition 3.5** (Length of a State)**.** We define a norm on the space of ket vectors by

$$\||x\rangle\| = \sqrt{\langle x|x\rangle}$$

and we call this the length of the vector $|x\rangle$.

The fact that we are in an inner product tells us this norm is well defined and that the properties of norm hold.

3.4. **The Tensor and Outer Products.** We now add further operators to our inner product space which will be useful later. To combine systems in quantum mechanics, we use the *tensor product*, which takes and $m \times n$ matrix and a $p \times q$ matrix to give a $mp \times nq$ matrix

**Definition 3.6** (Tensor Product)**.** Let $A$ be an $m \times n$ matrix and $B$ be a $p \times q$ matrix with elements $(A_{i,j})$ and $(B_{k,l})$ respectively. The matrices are then

$$A = \begin{pmatrix} A_{1,1} & A_{1,2} & \cdots & A_{1,n} \\ A_{2,1} & A_{2,2} & \cdots & A_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ A_{m,1} & A_{m,2} & \cdots & A_{m,n} \end{pmatrix}, \quad B = \begin{pmatrix} B_{1,1} & B_{1,2} & \cdots & B_{1,q} \\ B_{2,1} & B_{2,2} & \cdots & B_{2,q} \\ \vdots & \vdots & \ddots & \vdots \\ B_{p,1} & B_{p,2} & \cdots & B_{p,q} \end{pmatrix}$$

The tensor product $A \otimes B$ is the matrix formed by joining the matrices $A_{i,j}B$ so that the $((i-1)m + k, (j-1)n + l)^{\text{th}}$ element of $A \otimes B$ is $A_{i,j}B_{k,l}$. This can be seen more intuitively as

$$A \otimes B = \begin{pmatrix} A_{1,1}B & A_{1,2}B & \cdots & A_{1,n}B \\ A_{2,1}B & A_{2,2}B & \cdots & A_{2,n}B \\ \vdots & \vdots & \ddots & \vdots \\ A_{m,1}B & A_{m,2}B & \cdots & A_{m,n}B \end{pmatrix}$$

**Example 3.7** (Simple Matrix Tensor Product)**.**

$$\begin{pmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \end{pmatrix} \otimes \begin{pmatrix} B_{1,1} & B_{1,2} \\ B_{2,1} & B_{2,2} \end{pmatrix} = \begin{pmatrix} A_{1,1}\begin{pmatrix} B_{1,1} & B_{1,2} \\ B_{2,1} & B_{2,2} \end{pmatrix} & A_{1,2}\begin{pmatrix} B_{1,1} & B_{1,2} \\ B_{2,1} & B_{2,2} \end{pmatrix} \\ A_{2,1}\begin{pmatrix} B_{1,1} & B_{1,2} \\ B_{2,1} & B_{2,2} \end{pmatrix} & A_{2,2}\begin{pmatrix} B_{1,1} & B_{1,2} \\ B_{2,1} & B_{2,2} \end{pmatrix} \end{pmatrix}$$

$$= \begin{pmatrix} A_{1,1}B_{1,1} & A_{1,1}B_{1,2} & A_{1,2}B_{1,1} & A_{1,2}B_{1,2} \\ A_{1,1}B_{2,1} & A_{1,1}B_{2,2} & A_{1,2}B_{2,1} & A_{1,2}B_{2,2} \\ A_{2,1}B_{1,1} & A_{2,1}B_{1,2} & A_{2,2}B_{1,1} & A_{2,2}B_{1,2} \\ A_{2,1}B_{2,1} & A_{2,1}B_{2,2} & A_{2,2}B_{2,1} & A_{2,2}B_{2,2} \end{pmatrix}$$

The tensor product is used in quantum mechanics as it reflects the property of entanglement. In classical mechanics, the cross product is used. If we have classical systems in $\mathbb{R}^m$ and $\mathbb{R}^n$, then the two systems together act in the space $\mathbb{R}^m \times \mathbb{R}^n = \mathbb{R}^{m+n}$, giving us $m + n$ degrees of freedom. In the quantum world however, taking the tensor product of two separate systems in $\mathbb{C}^m$ and $\mathbb{C}^n$ we get the space $\mathbb{C}^{mn}$, which has $mn$ degrees of freedom. This is bigger than the number of degrees of freedom in the classical case for $m, n > 2$. This means that our quantum systems become very complex as we move into higher dimensions, but it also means that we get some unusual, but useful, effects occurring.

Before we move on, we look at the definition of the *outer product* for completeness.

**Definition 3.8** (Outer Product)**.** For $|x\rangle, |y\rangle \in \mathbb{C}^n$ we define the outer product

$$|x\rangle \langle y| := |x\rangle \otimes \langle y| = |x\rangle \otimes |\bar{y}\rangle^t$$

$|x\rangle \langle y|$ is an $n \times n$ matrix.

**Proposition 3.9.** *To fit in with the notation we have already developed, we should have* $\forall |x\rangle, |y\rangle, |p\rangle, |q\rangle \in \mathbb{C}^n$*,*

(1) $(|x\rangle \langle y|) |p\rangle = |x\rangle (\langle y|p\rangle)$
(2) $\langle q| (|x\rangle \langle y|) = (\langle q|x\rangle) |y\rangle$

*Proof.* For part 1,

$$\begin{aligned}(|x\rangle\langle y|)|p\rangle &= \begin{pmatrix} x_1\bar{y}_1 & x_1\bar{y}_2 & \cdots & x_1\bar{y}_n \\ x_2\bar{y}_1 & x_2\bar{y}_2 & \cdots & x_2\bar{y}_n \\ \vdots & \vdots & \ddots & \vdots \\ x_n\bar{y}_1 & x_n\bar{y}_2 & \cdots & x_n\bar{y}_n \end{pmatrix}\begin{pmatrix} p_1 \\ p_2 \\ \vdots \\ p_n \end{pmatrix} \\ &= \begin{pmatrix} \sum_{i=1}^{n} x_1\bar{y}_i p_i \\ \sum_{i=1}^{n} x_2\bar{y}_i p_i \\ \vdots \\ \sum_{i=1}^{n} x_n\bar{y}_i p_i \end{pmatrix} \\ &= \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}\sum_{i=1}^{n}\bar{y}_i p_i = |x\rangle(\langle y|p\rangle)\end{aligned}$$

For part 2,

$$\begin{aligned}\langle q|(|x\rangle\langle y|) &= (\bar{q}_1, \bar{q}_2, \ldots, \bar{q}_n)\begin{pmatrix} x_1\bar{y}_1 & x_1\bar{y}_2 & \cdots & x_1\bar{y}_n \\ x_2\bar{y}_1 & x_2\bar{y}_2 & \cdots & x_2\bar{y}_n \\ \vdots & \vdots & \ddots & \vdots \\ x_n\bar{y}_1 & x_n\bar{y}_2 & \cdots & x_n\bar{y}_n \end{pmatrix} \\ &= \left(\sum_{i=1}^{n}\bar{q}_i x_i\bar{y}_1\,,\ \sum_{i=1}^{n}\bar{q}_i x_i\bar{y}_2\,,\ \ldots\,,\ \sum_{i=1}^{n}\bar{q}_i x_i\bar{y}_n\right) \\ &= \sum_{i=1}^{n}\bar{q}_i x_i\,(\bar{y}_1, \bar{y}_2, \ldots, \bar{y}_n) = (\langle q|x\rangle)|y\rangle\end{aligned}$$

□

*Remark.* This proves associativity, and thus we can remove the brackets in the expressions above.

3.5. **Basis of The Inner Product Space.** A set of $n$ linearly independent vectors $\{|v_1\rangle, |v_2\rangle, \ldots, |v_n\rangle\} \subset \mathbb{C}^n$ is called a basis and any ket $|x\rangle$can be written as a linear combination of these kets, with the coefficient of each basis vector $|v_i\rangle$ equal to $\langle v_i|x\rangle$. We say two kets $|x\rangle, |y\rangle$ are *orthogonal* if

$$\langle x|y\rangle = 0$$

Any basis in which all the vectors are orthogonal to each other and each vector has length 1 is called an *orthonormal basis*. From any set of $k$ linearly independent vectors, we can get an orthonormal basis of the subspace spanned by these vectors by performing a *Gram-Schmidt orthonormalisation*. The preceding facts are all proved in [11] and should be familiar to the reader.

**Definition 3.10** (A Projection Operator)**.** Let $\{|e_1\rangle, |e_2\rangle, \ldots, |e_n\rangle\}$ be an orthonormal basis of $\mathbb{C}^n$, then the matrix

$$P_k = |e_k\rangle \langle e_k|$$

is called a projection operator in the direction $|e_k\rangle$.

**Example 3.11** (A Projection Operator Acting on a Vector)**.** Suppose we take the standard basis $\left\{(1,0)^t, (0,1)^t\right\}$ of $\mathbb{C}^2$, then the projection operator

$$P_1 = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$$

acting on a vector $v = (v_1, v_2)^t$, gives us $P_1 v = (v_1, 0)^t$ - the projection of the vector to the line defined by $(1,0)^t$. Similarly, $P_2 v = (0, v_2)^t$, giving us the projection of $v$ on the line formed by extending $(0,1)^t$.

In general, applying the projection operator to a vector gives us a new vector in the direction of the projection operator whose length is the inner product of the appropriate basis vector, and the operand vector. We shall see that the projection operator is useful for making measurements on our system and also useful for constructing time evolution operations. Geometrically it is easy to see that following properties hold, but their proofs are short, so we shall prove them anyway.

**Proposition 3.12.** *Let $P_k$ be a projection operator on $\mathbb{C}^n$, then*

(1) $P_k^2 = P_k$,
(2) $P_k P_j = 0$,
(3) *The completeness relation,* $\sum_{i=1}^{n} P_i = I_n$, *holds.*

*Where $I_n$ is the $n \times n$ identity matrix with diagonal entries all equal to one, and zeros everywhere else.*

*Proof.* (1) $P_k^2 = |e_k\rangle \langle e_k| |e_k\rangle \langle e_k| = |e_k\rangle \langle e_k \mid e_k\rangle \langle e_k| = |e_k\rangle 1 \langle e_k| = |e_k\rangle \langle e_k| = P_k$
(2) $P_k P_j = |e_k\rangle \langle e_k| |e_j\rangle \langle e_j| = |e_k\rangle \langle e_k \mid e_j\rangle \langle e_j| = |e_k\rangle 0 \langle e_j| = 0$

(3) For any $|x\rangle \in \mathbb{C}^n$,

$$I_n |x\rangle = |x\rangle = \sum_{i=1}^{n} \langle e_i \mid x\rangle |e_i\rangle = \sum_{i=1}^{n} |e_i\rangle \langle e_i \mid x\rangle = \left(\sum_{i=1}^{n} |e_i\rangle \langle e_i|\right) |x\rangle$$

And thus $\sum_{i=1}^{n} |e_i\rangle \langle e_i| = I_n$.

□

## 4. Making Measurements

The inner product space defined gives us a region in which to model states, superpose them and perform other operations on them. We would now like to extract *measurements* from our system. We may, for example, wish to measure energy, momentum or position. As in the experiment in section 3.2, when taking measurements we lose the non-locality and the probability distribution associated with the quantum system. We shall thus need to take account of this in our model for measurement. We will describe which state the system will *collapse* to by considering the *eigenvectors* of our measurements and the probability of the system doing this will be given by the *eigenvalues* associated to these.

### 4.1. Measurements as Linear Operators.

As we saw in the classical case, measurements are simply functions on the state of our system. We look at *linear operators* to begin with, which are linear functions from $\mathbb{C}^n \to \mathbb{C}^n$. When applying a linear operator $A : \mathbb{C}^n \to \mathbb{C}^n$ to a vector $|x\rangle \in \mathbb{C}^n$, we write $A|x\rangle$. We note the one to one correspondence between linear operators and $n \times n$ matrices and construct a vector space so that for linear operators, or equivalently, matrices $A, B$;

(1) $A|x\rangle = 0 \; \forall |x\rangle \quad \Leftrightarrow \quad A = 0$
(2) $(A + B)|x\rangle = A|x\rangle + B|x\rangle$
(3) $(AB)|x\rangle = A(B|x\rangle)$
(4) $(\langle x| A)|y\rangle = \langle x|(A|y\rangle)$

Note that multiplying a ket by a scalar value $k \in \mathbb{C}$ gives the same result as multiplying the ket by the matrix $kI$, thus we can regard scalar multiplication as a linear operator.

We look at expressing our matrix in terms of an orthonormal basis $B = \{|e_1\rangle, |e_2\rangle, \ldots, |e_n\rangle\}$. For $A : \mathbb{C}^n \to \mathbb{C}^n$ , $A|e_k\rangle$ is another ket vector in $\mathbb{C}^n$ and thus we can expand it in terms of our basis,

so for some unique $A'_{i,k} \in \mathbb{C}$

$$A\left|e_k\right\rangle = \sum_{i=1}^{n} A'_{i,k}\left|e_i\right\rangle$$

Multiplying by the bra $\langle e_j| \in B^*$, we get

$$\left\langle e_j\right| A\left|e_k\right\rangle = \left\langle e_j\right| \sum_{i=1}^{n} A'_{i,k}\left|e_i\right\rangle = \sum_{i=1}^{n} A'_{i,k}\left\langle e_j \mid e_i\right\rangle = A'_{j,k}$$

From these values $A'_{j,k} = \langle e_j| A |e_k\rangle$ we can construct a new matrix $A'$ such that $(A')_{j,k} = A'_{j,k} = \langle e_j| A |e_k\rangle$. This matrix tells us how the linear operator acts on vectors written in terms of the basis $B$. Again it is linear, so we can conclude that changing orthonormal bases does not affect the linearity of our linear operators, and in fact, from the completeness relation (proposition 3.12, part 3), we can write

$$A = IAI = \sum_{j,k=1}^{n} \left|e_j\right\rangle \left\langle e_j\right| A\left|e_k\right\rangle \left\langle e_k\right| = \sum_{j,k=1}^{n} \left|e_j\right\rangle A_{j,k} \left\langle e_k\right| = \sum_{j,k=1}^{n} A_{j,k}\left|e_j\right\rangle \left\langle e_k\right|$$

This tells us that $A$ is a linear transformation of $A'$, and similarly, $A'$ is a linear transformation of $A$. This decomposition is called a *spectral decomposition* and shows that we can describe a matrix fully in terms of its action on the basis vectors.

When we conducted our experiment in section 3.2, we found that if we removed the second half-silvered mirror, we had a probability of $1/2$ of observing an emitted photon at $A$ and a probability of $1/2$ of observing it at $B$, however we also said that the photon actually traveled both paths (this allowed interference to occur when we inserted the second half-silvered mirror). Our measurement of the position of the photon could not show us that half a photon had arrived at either detector. We had to see a whole photon at either one or the other. We called this the *collapse* of our probability distribution. We now look into modeling which states a system can collapse to, and the measurements we would get if the system collapsed into that state. We model these by eigenvectors in our state space and eigenvalues respectively.

**Definition 4.1** (Eigenvalue and Eigenvector)**.** Fixing our linear operator $A$, we call the solutions $\lambda \in \mathbb{C}$, $|x\rangle \in \mathbb{C}^n$ to

$$A\left|x\right\rangle = \lambda\left|x\right\rangle$$

the eigenvalues and eigenvectors of $A$.

We may get several eigenvectors for a specific eigenvalue, and we can show these eigenvectors form a vector subspace of $\mathbb{C}^n$, which contains all vectors that are eigenvectors corresponding to the eigenvalue. We can also show the eigenvectors corresponding to different eigenvalues are orthogonal. Any eigenvector can be normalised (i.e. multiplied by a factor to give a vector of unit length) and it will still be an eigenvector. With these facts in mind, we use symbol $|\lambda\rangle$ to denote an eigenvector of an eigenvalue $\lambda$. For the method of finding eigenvectors and eigenvalues the reader should perform a search online or consult a book on linear algebra such as [11].

4.2. **Hermitian Matrices.** We take a measurement to be a calculation of the form

$$\langle x| A |y\rangle$$

Where $A$ linear operator and the states $|x\rangle, |y\rangle$ are being measured. It is normally the case that we would like to measure one state, in which case we take the measurement

$$\langle x| A |x\rangle$$

We are talking about linear operators as measurements and we therefore need them to have real number results. We could measure the imaginary and real parts of a complex number, but as we saw before, there are many problems associated with making two measurements, especially if the two are linked. How can we guarantee our results are real numbers? Well, if $A$ is our linear operator, then we want that for all states $|x\rangle, |y\rangle \in \mathbb{C}^n$,

$$\langle x| A |y\rangle = \overline{\langle x| A |y\rangle}$$

For this we will define a *self-adjoint* linear operator, or in terms of a matrix, a *Hermitian* matrix.

**Definition 4.2** (Hermitian Conjugate)**.** Given an $n \times n$ matrix $A$ over $\mathbb{C}$, We define its Hermitian conjugate $A^*$ to be the complex conjugate transpose of $A$. i.e. $A_{i,j} = A^*_{i,j}$

$$A = \begin{pmatrix} A_{1,1} & A_{1,2} & \cdots & A_{1,n} \\ A_{2,1} & A_{2,2} & \cdots & A_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ A_{n,1} & A_{n,2} & \cdots & A_{n,n} \end{pmatrix} \iff A^* = \begin{pmatrix} \overline{A_{1,1}} & \overline{A_{2,1}} & \cdots & \overline{A_{n,1}} \\ \overline{A_{1,2}} & \overline{A_{2,2}} & \cdots & \overline{A_{n,2}} \\ \vdots & \vdots & \ddots & \vdots \\ \overline{A_{1,n}} & \overline{A_{2,n}} & \cdots & \overline{A_{n,n}} \end{pmatrix}$$

**Proposition 4.3.** *For all* $|x\rangle, |y\rangle \in \mathbb{C}^n$

$$\langle x| A |y\rangle = \langle A^* x \mid y\rangle = \overline{\langle y| A^* |x\rangle}$$

*Where* $\langle A^* x| \equiv \overline{(A^* |x\rangle)^t}$

*Proof.* We prove the first equality:

$$\langle x|\, A\, |y\rangle = (\overline{x}_1, \bar{x}_2, \ldots, \overline{x}_n) \begin{pmatrix} A_{1,1} & A_{1,2} & \cdots & A_{1,n} \\ A_{2,1} & A_{2,2} & \cdots & A_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ A_{n,1} & A_{n,2} & \cdots & A_{n,n} \end{pmatrix} |y\rangle$$

$$= \left( \sum_{i=1}^{n} \overline{x}_i A_{i,1}\, ,\, \sum_{i=1}^{n} \overline{x}_i A_{i,2}\, \ldots\, \sum_{i=1}^{n} \overline{x}_i A_{i,n} \right) |y\rangle$$

$$= \overline{\begin{pmatrix} \sum_{i=1}^{n} x_i \overline{A_{i,1}} \\ \sum_{i=1}^{n} x_i \overline{A_{i,2}} \\ \vdots \\ \sum_{i=1}^{n} x_i \overline{A_{i,n}} \end{pmatrix}^t} |y\rangle$$

$$= \overline{\left[ \begin{pmatrix} \overline{A_{1,1}} & \overline{A_{2,1}} & \cdots & \overline{A_{n,1}} \\ \overline{A_{1,2}} & \overline{A_{2,2}} & \cdots & \overline{A_{n,2}} \\ \vdots & \vdots & \ddots & \vdots \\ \overline{A_{1,n}} & \overline{A_{2,n}} & \cdots & \overline{A_{n,n}} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \right]^t} |y\rangle \quad = \quad \langle A^* x \mid y \rangle$$

And now the second equality comes from the conjugacy relation $\langle a \mid b \rangle = \overline{\langle b \mid a \rangle}$ described in theorem 3.3:

$$\langle A^* x \mid y \rangle = \overline{\langle y \mid A^* x \rangle} \equiv \overline{\langle y|\, A^*\, |x\rangle}$$

□

Thus if we have $A = A^*$, our measurement is equal to its conjugate and thus real.

**Definition 4.4** (Hermitian Matrix)**.** A matrix $A$ with the property $A = A^*$ is called Hermitian.

*Remark.* Equivalently, a linear operator $A$ which has the property that its matrix $A = A^*$ is called *self-adjoint.*

**Proposition 4.5.** *A Hermitian matrix has real eigenvalues.*

*Proof.* If $A|\lambda\rangle = \lambda|\lambda\rangle$ for all eigenvectors $|\lambda\rangle$ corresponding to $\lambda$ then

$$\begin{aligned}\overline{\lambda}\langle\lambda \mid \lambda\rangle =& \overline{\lambda}\left|\overline{\lambda}\right\rangle^t|\lambda\rangle = \overline{(\lambda|\lambda\rangle)^t}|\lambda\rangle = \overline{(A|\lambda\rangle)^t}|\lambda\rangle \\ =& \langle A^*\lambda \mid \lambda\rangle = \langle\lambda| A|\lambda\rangle = \langle\lambda|\lambda|\lambda\rangle \\ =& \lambda\langle\lambda \mid \lambda\rangle\end{aligned}$$

Thus $\overline{\lambda} = \lambda$ and $\lambda \in \mathbb{R}$. □

**Proposition 4.6.** *The eigenvectors corresponding to different eigenvalues of a Hermitian matrix are orthonormal*

*Proof.* Assume we have to eigenvalues $\lambda \neq \lambda'$ of a Hermitian matrix $A$, then

$$\lambda'\langle\lambda' \mid \lambda\rangle = \langle A\lambda' \mid \lambda\rangle = \langle\lambda| A|\lambda\rangle = \langle\lambda'|\lambda|\lambda\rangle = \lambda\langle\lambda' \mid \lambda\rangle$$

Since $\lambda$ and $\lambda'$ are real and not equal, we must have $\langle\lambda' \mid \lambda\rangle = 0$ and $|\lambda\rangle$ is orthogonal to $|\lambda'\rangle$ □

Applying the Gram-Schmidt orthonormalisation procedure, we can find an orthonormal basis of the vector spaces spanned by the eigenvectors corresponding to each eigenvalue. Since the eigenvectors corresponding to each eigenvalue are all orthogonal to each other, we can construct an orthonormal basis of the space of all eigenvectors of a linear operator. We assume from now on that all our basis are orthonormal basis.

We now look at how probability factors into measurements. Suppose that we have basis states $\{|e_1\rangle, |e_2\rangle, \ldots, |e_n\rangle\}$ of $\mathbb{C}^n$. Making that measurement causes the state to jump to one of the eigenvectors. We look at applying the projection operator to make a measurement of the state

$$c_1|e_1\rangle + c_2|e_2\rangle + \ldots + c_n|e_n\rangle$$

Then, for each $|e_i\rangle$

$$\begin{aligned}(\overline{c}_1\langle e_1| + \overline{c}_2\langle e_2| + \ldots + \overline{c}_n\langle e_n|)|e_i\rangle\langle e_i|(c_1|e_1\rangle + c_2|e_2\rangle + \ldots + c_n|e_n\rangle) \\ = \overline{c}_i c_i\langle e_i| e_i\rangle\langle e_i| e_i\rangle \\ = |c_i|^2\end{aligned}$$

It takes a little work, but Dirac shows that this is the probability of the state *collapsing* to the state $|e_i\rangle$ (it is quite easy to guess this is the case). Now, if we consider this holds for each $|e_i\rangle$ and

that the system cannot collapse to a state outside its state space, we must have that the sum of these probabilities is 1. i.e.

$$|c_1|^2 + |c_2|^2 + \ldots + |c_n|^2 = 1$$

As we suspected in the experiment of section 3.2, our probabilities and phase factors are linked.

## 5. Quantum Mechanical Time Evolution and Larger Systems

While discussing time evolving quantum mechanics we ignore making measurements during the time evolution. We classify a certain type of matrix which we use to transform our state space from one time, to our state space at the next. The laws of quantum mechanics tell us that the matrices that perform time evolution are *unitary matrices*.

**Definition 5.1** (Unitary Matrix)**.** An invertible matrix $U$ which satisfies $U^* = U^{-1}$ (i.e. its inverse is its Hermitian conjugate) is called a *unitary matrix*. A unitary matrix with unit determinant is said to be a *special unitary matrix*.

**Proposition 5.2.** *A unitary matrix $U$ acting on $\mathbb{C}^n$ preserves inner products: For all $|x\rangle, |y\rangle \in \mathbb{C}^n$, $\langle Ux \mid Uy\rangle = \langle x \mid y\rangle$.*

*Proof.*

$$\langle Ux \mid Uy\rangle = \langle U^*Ux \mid y\rangle = \left\langle U^{-1}Ux \mid y\right\rangle = \langle x \mid y\rangle$$

□

Thus it is also easy to see they preserve norms, and if two vectors are orthogonal before being acted upon by a unitary matrix, they will be orthogonal afterwards. We find that unitary operators are the exact transformations of our space which occur as time progresses. So long as our system is left undisturbed, it will progress so that for a state $s(t)$ at time $t > 0$, and initial state $s_0$ at $t = 0$, there is a time dependent matrix $U(t)$, which is unitary for every $t$, and such that

$$s(t) = U(t)\, s_0$$

But then, by the fact that $U(t)$ is invertible, we find

$$s_0 = U^{-1}(t)\, s(t)$$

Thus, so long as we do not interrupt the system between times $t_0$ and $t_1$and we know the state $s_T$ at time $T \in [t_0, t_1]$ and the time dependent unitary operator $U(t)$ for $t \in [t_0, t_1]$, we can determine

the state at $s_0$ and therefore all the states $s(t)$. Thus a quantum system which is free from external influence evolves deterministically.

We finally note some properties of unitary matrices $U_1, U_2$ (these are proven in [11])

(1) $U_1U_2$ is unitary
(2) $U_1 \otimes U_2$ is unitary
(3) $U^{-1}$ is unitary

5.1. **A Quantum Coin Toss.** The quantum coin flip is a good example (taken from [15]) of how strange our departure is, not only from classical mechanics, but from classical probability ideas. We consider tossing a quantum coin. This quantum coin must have two measurable states: $|H\rangle$ and $|T\rangle$ - in the classical sense corresponding to heads or tails facing up. The act of tossing the coin from either original state must be a unitary operation $U$ giving us the probability of measuring $|H\rangle$ to be $1/2$ and the probability of measuring $|T\rangle$ to be $1/2$. Noting our discussion on probabilities at the end of section 4.2, our phase factors therefore must be $\pm 1/\sqrt{2}$ after the operation. To ensure our operation is unitary, we must have either of the following

$$\left.\begin{aligned} U|H\rangle &= \frac{1}{\sqrt{2}}|H\rangle + \frac{1}{\sqrt{2}}|T\rangle \\ U|T\rangle &= \frac{1}{\sqrt{2}}|H\rangle - \frac{1}{\sqrt{2}}|T\rangle \end{aligned}\right\} \quad (1)$$

$$\left.\begin{aligned} U|H\rangle &= -\frac{1}{\sqrt{2}}|H\rangle + \frac{1}{\sqrt{2}}|T\rangle \\ U|T\rangle &= \frac{1}{\sqrt{2}}|H\rangle + \frac{1}{\sqrt{2}}|T\rangle \end{aligned}\right\} \quad (2)$$

We will just consider the first situation, as the the second is very similar. We would expect that applying the unitary operation to either of the resulting states would give us another state where the probability of observing the state $|H\rangle$ is $1/2$ and of observing $|T\rangle$ is $1/2$. However, assuming we do not make a measurement, we see

$$\begin{aligned} U^2|H\rangle &= U\left(\frac{1}{\sqrt{2}}|H\rangle + \frac{1}{\sqrt{2}}|T\rangle\right) = |H\rangle \\ U^2|T\rangle &= U\left(\frac{1}{\sqrt{2}}|H\rangle - \frac{1}{\sqrt{2}}|T\rangle\right) = |T\rangle \end{aligned}$$

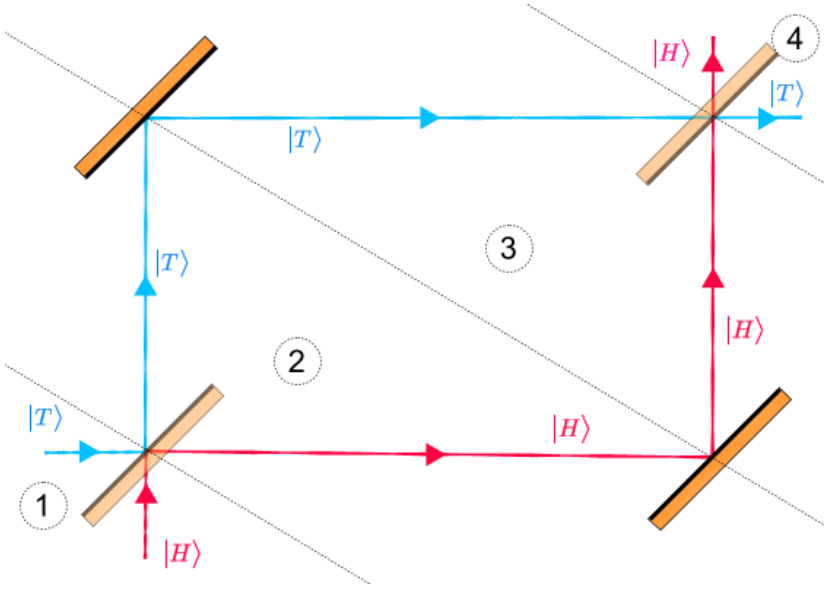


FIGURE 5.1. Mach-Zehnder Interferometer with an extra input

A measurement made on $U^{2m}|H\rangle$ $(m \in \mathbb{N})$ will give us $|H\rangle$ with probability 1 and a measurement made on $U^{2m}|T\rangle$ $(m \in \mathbb{N})$ will give us $|T\rangle$ with probability 1. Classical mechanics tells us that if we knew all the information about the coin, and the way it was tossed, we would be able to work out whether it lands as a head or a tails. This experiment does not tell us that there is no quantum equivalent of tossing a coin, it serves as a counterexample to why we could not simplify our space to a simple probabilistic one in which case the factors are real numbers. We need complex phase factors.

**Definition 5.3** (Hadamard-Walsh Matrix)**.** The unitary operator just described is called a *Hadamard-Walsh matrix*

$$H_2 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

*Remark.* The previous example showed that $H_2^2 = I$.

5.2. **Further Development of the Mach-Zehnder Interferometer.** Returning to the experiment with the Mach-Zehnder interferometer we studied in section 3.2, we see that the Hadamard matrix actually describes the action of the beam splitter if we take the limit as the width of the glass at the back of each beam splitter tends to zero. Take the state $|H\rangle$ to be the state of the photon existing somewhere, in each section, along the path labeled $|H\rangle$ (the red path) in figure 5.1, and the state $|T\rangle$ to be the state where the photon exists somewhere along the path labeled

$|T\rangle$ (the blue path), in each section, in the same figure. Now assume each section 1,2,3 and 4 correspond to the same amount of time (i.e. the mirrors are positioned on the corners of a square), then in each section, or time frame $T_i$ $(i = 1, 2, 3, 4)$, the polarisation state of the photon is constant so long as we do not make any measurements.

We stated in section 3.2 that reflection $R_O$ of a mirror from the outside (side made up of air) changes the photons phase by $\pi$, whereas reflection $R_I$ from the inside (glass side) caused no phase change. i.e. $R_O |S\rangle = -|S\rangle$ and $R_I = I$. Thus we get the following table of states, showing what happens if we initially fire a photon along either the $|H\rangle$ path or $|T\rangle$ from the bottom left of the figure 5.1.

| $T_1$ | $\|H\rangle$ | $\|T\rangle$ |
|---|---|---|
| $T_2$ | $\frac{1}{\sqrt{2}}(\|H\rangle + \|T\rangle)$ | $\frac{1}{\sqrt{2}}(\|H\rangle - \|T\rangle)$ |
| $T_3$ | $-\frac{1}{\sqrt{2}}(\|H\rangle + \|T\rangle)$ | $-\frac{1}{\sqrt{2}}(\|H\rangle - \|T\rangle)$ |
| $T_4$ | $\|H\rangle$ | $\|T\rangle$ |

Write $|H\rangle$ and $|T\rangle$ as the standard basis for $\mathbb{C}^2$:

$$|H\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad |T\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

Between the four times above we have three unitary matrices acting on our state space $\mathbb{C}^2$. The first of these $U_{1\to 2}$ we see is the Hadamard-Walsh matrix $H_2$:

$$H_2 |H\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \frac{1}{\sqrt{2}} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{2}} |H\rangle + \frac{1}{\sqrt{2}} |T\rangle$$

$$H_2 |T\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 0 \end{pmatrix} 0 \frac{1}{\sqrt{2}} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{2}} |H\rangle - \frac{1}{\sqrt{2}} |T\rangle$$

Between times $T_2$ and $T_3$, the reflection $R_0$ acts on both paths. By looking at the effect it has on our vectors, we see we can write

$$U_{2\to 3} = R_0 = \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix} = -I$$

We look at the final transform which occurs between $T_3$ and $T_4$ and see that we can write this as

$$U_{3\to 4} = \frac{1}{\sqrt{2}} \begin{pmatrix} -1 & -1 \\ -1 & 1 \end{pmatrix} = -H_2$$

In accordance with our mathematical construction, we see that

$$U_{3\to4} \cdot U_{2\to3} \cdot U_{1\to2} = (-H_2) \cdot (-I) \cdot (H_2) = H_2^2 = I$$

This agrees with the evidence we have seen. If we set our experiment to one of our basis states $|H\rangle$ or $|T\rangle$ at $T_1$ (i.e. fire a photon from either the bottom or left of the first half-silvered mirror), then by the time we get to time $T_4$ our system will be in its original state (i.e. a photon will exist along the same path somewhere after the second half-silvered mirror). However, this result also tells us that if we sent a photon spread across the two paths, then it would come out spread across the two paths in exactly the same way.

The choice of paths $|H\rangle$ and $|T\rangle$ was made to ensure we used our first unitary operator in section 5.1. We could of course have chosen different paths and used, amongst others, the second unitary operator of section 5.1. Both models of the experiment are equivalent.

Unitary transforms which act on a two dimensional quantum space are called *quantum gates*. $H_2$ is an example of a quantum gate, which we call the *Hadamard gate*. We shall find out more about these in section 6. We note that there are many different implementations of $H_2$ (and other quantum gates) into real world constructions. Another, more useful implementation of the Hadamard gate can be found in [16].

5.3. **Simultaneous Coin Flips.** For our work on quantum mechanics to be useful, we will have to extend our work on one quantum particle to look at several particles. Loosely following the development on pp10 and pp11 of [15], we begin by making a small jump and considering the gedanken experiment[1] of section 5.1 with two "quantum coins". We have four states for our quantum system; the first coin $|C_1\rangle$ taking basis states $|H_1\rangle$ or $|T_1\rangle$ and the second coin $|C_2\rangle$ having basis states $|H_2\rangle$ and $|T_2\rangle$. We can think of the basis states as pairs $(|H_1\rangle, |H_2\rangle)$, $(|H_1\rangle, |T_2\rangle)$, $(|T_1\rangle, |H_2\rangle)$ and $(|T_1\rangle, |T_2\rangle)$. Thus we can simplify the notation to represent the compound states of the system by

$$|C_1C_2\rangle = c_{H,H}\,|H_1H_2\rangle + c_{H,T}\,|H_1T_2\rangle + c_{T,H}\,|T_1H_2\rangle + c_{T,T}\,|T_1T_2\rangle$$

with $|c_{H,H}|^2 + |c_{H,T}|^2 + |c_{T,H}|^2 + |c_{T,T}|^2 = 1$.

[1]Gedanken is the German word for "thought", and thus a gedanken experiment is a thought experiment - One in which no real experiment is done, but the idea is to explore the principle being studied.

We can this in terms of two subsystems; one for each coin. We construct a new space out of the tensor product of these subspaces (definition 3.6) . We take

$$|H_1\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad |T_1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

as a basis for the system describing our first coin, and

$$|H_2\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad |T_2\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

as a basis for the system describing our second coin. The tensor products then give a natural basis

$$\begin{aligned} |H_1H_2\rangle &= |H_1\rangle \otimes |H_2\rangle = (1,\, 0,\, 0,\, 0)^t \\ |H_1T_2\rangle &= |H_1\rangle \otimes |T_2\rangle = (0,\, 1,\, 0,\, 0)^t \\ |T_1H_2\rangle &= |T_1\rangle \otimes |H_2\rangle = (0,\, 0,\, 1,\, 0)^t \\ |T_1T_2\rangle &= |T_1\rangle \otimes |T_2\rangle = (0,\, 0,\, 0,\, 1)^t \end{aligned}$$

Assume we start with two coins with the heads facing upwards, i.e. we are in the $|H_1H_2\rangle$ state.

Applying the Hadamard matrix $H_2$ to the first the first coin, we get

$$H_2\,|H_1\rangle = \frac{1}{\sqrt{2}}\left(|H_1\rangle + |T_1\rangle\right)$$

in the subsystem constructed for the first coin. Applying the Hadamard to the second coin, we see

$$H_2\,|H_2\rangle = \frac{1}{\sqrt{2}}\left(|H_2\rangle + |T_2\rangle\right)$$

What is the total state of the system? Taking the tensor product, we see it is in state

$$\begin{aligned} \left[\frac{1}{\sqrt{2}}\left(|H_1\rangle + |T_1\rangle\right)\right] \otimes \left[\frac{1}{\sqrt{2}}\left(|H_2\rangle + |T_2\rangle\right)\right] &= \left[\frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 1 \end{pmatrix}\right] \otimes \left[\frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 1 \end{pmatrix}\right] \\ &= \frac{1}{2}\begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} = \frac{1}{2}\left(|H_1H_2\rangle + |H_1T_2\rangle + |T_1H_2\rangle + |T_1T_2\rangle\right) \end{aligned}$$

The tensor product works out such that we do not need to normalise - each state has probability $1/4$ of being seen, but since any state here is the tensor product of two of the original states, the

probability of measuring each of the states $|H_1\rangle, |T_1\rangle, |H_2\rangle, |T_2\rangle$ is still $1/2$. Notice that in this system we can measure the two subsystems independently, so that measuring the value of the first coin will not affect measuring the value of the second coin. This is because we can *decompose* our total system into two subsystems.

**Definition 5.4** (Decomposable and Entangled States)**.** Suppose we have two subsystems in which the basis states are $|x_1\rangle, |x_2\rangle, \ldots, |x_m\rangle$ and $|y_1\rangle, |y_2\rangle, \ldots, |y_n\rangle$, and the compound system is represented as

$$\sum_{i=1}^{m}\sum_{j=1}^{n} c_{i,j}\left(|x_i\rangle \otimes |y_j\rangle\right) \equiv \sum_{i=1}^{m}\sum_{j=1}^{n} c_{i,j}\,|x_i y_j\rangle$$

where $c_{i,j} \in \mathbb{C}$ are the phase factors. If we can write state of the system as

$$\left(\sum_{i=1}^{m} a_i\,|x_i\rangle\right) \otimes \left(\sum_{j=1}^{n} b_j\,|y_j\rangle\right)$$

for phase factors $a_i, b_j \in \mathbb{C}$, then we say the system is in a *decomposable state*. If we cannot write the state of the system in this way, we say it is in an *entangled state*.

*Remark.* We normally write the tensor product of ket vectors $|x\rangle, |y\rangle$ as $|xy\rangle \equiv |x\rangle \otimes |y\rangle$ as we have done above. It is also often the case that the tensor product symbol is removed so $|xy\rangle \equiv |x\rangle \otimes |y\rangle \equiv |x\rangle\,|y\rangle$.

We have just seen a system in a decomposable state. We can certainly write the state before the Hadamard transform on both coins as the tensor product of two states. After the Hadamard transforms on both coins, we can see that by the way we constructed our compound state,

$$\frac{1}{2}\left(|H_1H_2\rangle + |H_1T_2\rangle + |T_1H_2\rangle + |T_1T_2\rangle\right) = \left[\frac{1}{\sqrt{2}}\left(|H_1\rangle + |T_1\rangle\right)\right] \otimes \left[\frac{1}{\sqrt{2}}\left(|H_2\rangle + |T_2\rangle\right)\right]$$

We notice that the transform that applies the Hadamard transform to both coins in the compound system can be written as

$$H_4 = H_2 \otimes H_2 = \frac{1}{2}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix}$$

$$\implies H_4\,|H_1H_2\rangle = \frac{1}{2}\left(|H_1H_2\rangle + |H_1T_2\rangle + |T_1H_2\rangle + |T_1T_2\rangle\right)$$

The compound transform is the tensor product of the transforms acting on the subsystems for each coin, and thus it can not "entangle" the two coins. $H_4$ is one of many *separable* matrices.

**Definition 5.5** (Separable Matrix)**.** A square matrix $M$ is *separable* if there is an $m \times m$ matrix $A$ and an $n \times n$ matrix $B$, with $m > 1$, $n > 1$, such that $M = A \otimes B$.

The question however still remains: can we entangle two pairs of quantum states? Einstein [17] had much trouble accepting that this could exist, as having two particles which are entangled means that we can separate them over an arbitrary distance, then measure one of them and know what the state of the unmeasured particle is. For example, suppose we have two entangled photons traveling away from each other with the polarisation state of the whole system being

$$\frac{1}{\sqrt{2}}\left(|H_1H_2\rangle + |V_1V_2\rangle\right)$$

If we measure the first one and find it to be in state $|H_1\rangle$, then there is no other choice but for the second one to be in state $|H_2\rangle$ as there are no states $|H_1V_2\rangle$ or $|V_1H_2\rangle$ which could allow one photon to be horizontally polarised and the other to be vertically polarised. Similarly if we measure states $|T_1\rangle, |H_2\rangle$ or $|T_2\rangle$, then we will also measure states $|T_2\rangle, |H_1\rangle$ or $|T_2\rangle$ respectively. Pairs of particles which share entangled properties are called *EPR pairs* after the authors of paper [17]. There are many methods of creating EPR pairs, and more methods are still be researched. There is a short article on these at [18] and some examples of how to realise EPR pairs at [19] and [20].

The construction of these pairs is very useful, as we will see in section 8.

## 6. The Components of a Quantum Computer

Constructions such as the one in section 5.2 form one of the parts of a quantum computer. Constructions which carry out operations such as the Hadamard-Walsh transform form the processors of quantum computers and we call the most basic of these components *quantum gates*. We will see how we construct other parts of our computer such as the memory and output. In this section we look at very general view of what a computer is, the idea of which came from chapter 3 of [15], and a good source for further details is [21]. We follow this by looking at a less general, but more realisable view of computers so that we can theoretically build a quantum computer. All of the diagrammatic notation for section 6.4 is taken from [1], with the extra information I have added in blue.

6.1. **What is a Computer?** Today we think of a computer as an electronic device which communicates with the user (and other computers), and stores and processes data. Looking back to when computers were in their youth, a computer would have very limited communication ability. The emphasis, at this time, was on improving storage and processing speed amongst other things (such as size and reliability). A basic computer would take a length of magnetised tape as its input, process this input using a collection of electronics and then output its data, either back to tape or to a display. In his paper [22], Turing begins to set up a theoretical definition of a computer. Turing takes an infinitely long *tape* made up of blocks on which a device, called the *head*, can act to read and write information one block at a time. The head is allowed to move left or right along the tape one step at a time (or stay where it is) and each time it reads information from a block, it sends the information to be processed before doing anything else. The processing of the information takes place by considering two things, a finite and constant *table of actions* and the state of a *state register*, which can vary as the machine runs. The state register is a finite collection $Q$ of possible states for the machine to be in. The table of actions takes, as its input, the state of the machine and the information on the tape at the head's current position. It has a finite set of rules, which can modify, in the following order,

(1) The information on the block of the tape that the head is currently located at.
(2) The state of the machine.
(3) The position of the head (move it one block to the left, one block to the right, or keep it where it is).

The information that can be stored on each block of the tape is constricted to a finite set of values called the tape alphabet $\Gamma$. We assume there is a blank symbol $b \in \Gamma$, which all blocks of the tape are initialised to before the machine starts to operate, and an initial state $q_0 \in Q$ that the state begins in. The machine then stops operating, or *halts*, when the machine reaches a state $q_a$ or $q_r \in Q$ . In the *accepting state* $q_a$ the machine should have completed the task given. The machine can also halt at the *rejecting state* $q_r$ or never halt.

Each time the machine reads from the tape (and thus processes, writes and moves the head) we say we have taken another *computational step*. Suppose we set up a Turing machine with a fixed table of actions (along with a fixed states and tape alphabet), then unless the Turing machine never halts, we can say it has taken a certain number of *computational steps* to halt.

The behavior of the Turing machine can all be encoded into one function, the *transition function* $\delta : \Gamma \times Q \to \Gamma \times Q \times D$, which tells us our table of actions. $D = \{L, 0, R\}$ is the set of movements the head can make with $L$ corresponding to a movement one block to the left, $R$ corresponding to

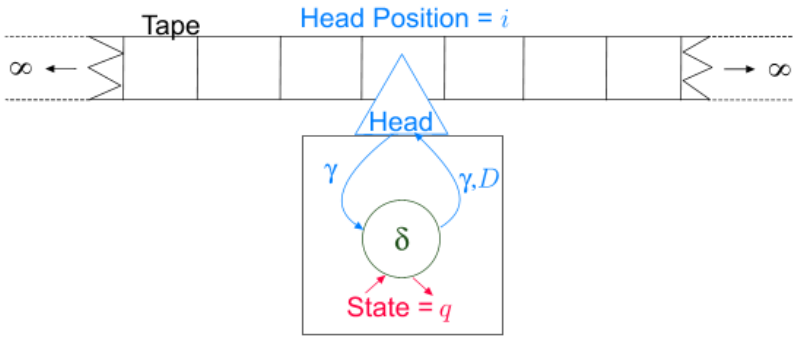


FIGURE 6.1. Turing Machine

one block to the right, and 0 being no movement at all. The machine itself stores its state. The tape is an ordered set $T$ where each element of $T$ is also an element of $\Gamma$. The head provides us with functions $r : \mathbb{Z} \times T \to \Gamma$[2] to read from, and $w : \mathbb{Z} \times T \times \Gamma \to T$ to write to, the current block and $m : \mathbb{Z} \times D \to \mathbb{Z}$ to move the head to the left, right or not move at all. Suppose $i \in \mathbb{Z}$ is the position of the head on the tape then $r(i, T)$ gives us the information at $i^{\text{th}}$ position on the tape, and $w(i, T, \gamma)$ results in a tape $T'$, identical to $T$ except with $\gamma \in \Gamma$ written to the the $i^{\text{th}}$ place on the tape. $m$ changes our value of $i$ depending on the direction $d \in D$ it is told to move.

The machine can thus be in a state

$$s \in S = \{T, q, i \,:\, q \in Q,\, i \in \mathbb{Z}\}$$

The function $p : S \to S$, which describes the change of state making a computational step causes, can be written as

$$p(T, q, i) = (w \circ \delta_\Gamma(r(i, T), q),\ \delta_Q(r(i, T), q),\ m(i, \delta_D(r(i, T), q)))$$

where $\delta_\Gamma, \delta_Q$ and $\delta_D$ are the respective components of the function $\delta$.

We have full mathematical formulation of a machine we call a *deterministic Turing machine*.

**Definition 6.1** (Deterministic Turing Machine)**.** A deterministic Turing machine $M$ over an alphabet $\Gamma$ is a sixtuple $(Q, A, \delta, q_0, q_a, q_r)$, where $q_0, q_a, q_r \in Q$ are the initial, accepting and rejecting

[2]Here I have used the notation $T$ to mean the set of tapes with elements in $\Gamma$ as well as to signify a specific tape $T$

states respectively, Q is a finite set of control states and $\delta : Q \times \Gamma \to Q \times \Gamma \times \{L, 0, R\}$ is the transition function.[3]

We see that a computer which runs in accordance with classical mechanics is modeled very well by the Turing machine and many useful results on complexity have come from this. We note that when constructing a quantum computer, we would like to introduce the idea that the $\delta$ function above acts probabilistically over the complex numbers, and thus we rewrite it as

$$\delta : Q \times \Gamma \times Q \times \Gamma \times \{L, 0, R\} \to \mathbb{C}$$

This transition function can give us the probability of moving from the state given in the first two arguments to the state in the final three arguments. We obtain this probability by squaring the modulus of $\delta$. We now have to consider the state of the system by considering $\delta$ for every state at each stage of the computation. It is thus necessary to simplify what we mean by a quantum computer. We will do this by setting up our tape, for which we use *qubits*, then creating a *quantum circuit* (analogous to an electronic circuit) to process these qubits. Throughout much of the discussion we will ignore the idea of measurement, assuming that we leave it until the end of the task at hand and that it is possible.

6.2. **Storing Information in Bits.** In a classical computer we store information in binary format. Storage space is an ordered set of components which can be set to a 1 state or a 0 state, and the state can be read. Each individual component is called a *bit*. We take a quick look into binary format. There are standard ways of encoding, amongst others, whole decimal numbers, certain real numbers and characters into binary. The easiest objects to encode out of these are whole decimal numbers. Binary is simply a number system working in *base* 2 instead of *base* 10 as in the decimal number system. We count upwards from zero, writing $0, 1, 10, 11, 100, 101, \ldots$. Suppose we have a decimal number with $a_1$ as the units ($10^0$s) digit, $a_2$ as the tens ($10^1$s) digit, $a_3$ as the hundreds ($10^2$s) digit, ..., $a_n$ as the $10^n$s digit. We use the notation $a_n a_{n-1} \ldots a_1 a_0$ to represent the number[4], so that we can write

$$\begin{aligned} a_n a_{n-1} \ldots a_1 a_0 &\equiv (a_n \cdot 10^n) + \left(a_{n-1} \cdot 10^{n-1}\right) + \ldots + (a_1 \cdot 10) + a_0 \\ &= \sum_{i=0}^{n} 10^i a_i \end{aligned}$$

[3]There are many different definitions of a Turing machine. This one is similar to the one given at the start of chapter 3 in [15]

[4]Note we do not mean that the numbers are multiplied by each other, we just use this notation as it shows how the number would actually be written.

How would we find the digits $a_n a_{n-1} \dots a_1 a_0$, given a number $A$? We would use the previous equation and write that $a_i$ is the remainder when $A - \sum_{j=0}^{i-1} 10^j a_j$ is divided by $10^{i+1}$. This gives us a recursive method for calculating $a_0$, then $a_1$,... For example to find the digits of $a = 5386$ we get the following table

| $i$ | $10^{i+1}$ | $5386 - \sum_{j=0}^{i-1} 10^j a_j$ | Remainder $= a_i$ |
|---|---|---|---|
| 0 | 10 | 5386 | 6 |
| 1 | 100 | 5380 | 8 |
| 2 | 1000 | 5300 | 3 |
| 3 | 10000 | 5000 | 5 |

We stop once $a - \sum_{j=0}^{i} 10^j a_j$ gets to zero. Equivalently, we could let

$$\begin{aligned} A_i &= \frac{A_{i-1} - a_{i-1}}{10} \in \mathbb{N} \\ A_0 &= A \end{aligned}$$

Then $a_i$ is the remainder when $A_i$ is divided by 10. This removes the large powers of 10 we get, and makes the derivation of the binary section straightforward. Taking the same example above;

| $i$ | Divisor | $A_i$ | Remainder $= a_i$ |
|---|---|---|---|
| 0 | 10 | 5386 | 6 |
| 1 | 10 | 538 | 8 |
| 2 | 10 | 53 | 3 |
| 3 | 10 | 5 | 5 |

We stop when our $A_i$ gets to zero.

Now suppose we are in base 2, then our $a_i \in \{0, 1\}$ and we write

$$\begin{aligned} a_n a_{n-1} \dots a_1 a_0 \equiv a &= (a_n \cdot 2^n) + \left(a_{n-1} \cdot 2^{n-1}\right) + \dots + (a_1 \cdot 2) + a_0 \\ &= \sum_{i=0}^{n} 2^i a_i \end{aligned} \tag{6.1}$$

Then if we are given a decimal number $a$, we can write its binary representation as $a_n a_{n-1} \dots a_1 a_0$, where $a_i$ is the remainder when $A_i$ is divided by and

$$\begin{aligned} A_i &= \frac{A_{i-1} - a_{i-1}}{2} \in \mathbb{N} \\ A_0 &= A \end{aligned} \tag{6.2}$$

The following table shows an example $a = 133$.

| $i$ | $A_i$ | Remainder $= a_i$ |
|---|---|---|
| 0 | 133 | 1 |
| 1 | 66 | 0 |
| 2 | 33 | 1 |
| 3 | 16 | 0 |
| 4 | 8 | 0 |
| 5 | 4 | 0 |
| 6 | 2 | 0 |
| 7 | 1 | 1 |

We stop when our $A_i$ gets to zero. So the decimal number 133 is written as 1000101 in binary. This is normally written as $133_B \equiv (133)_B \equiv 1000101$. To decode the decimal number from binary we can use equation 6.1. We can use a simple proof by contradiction to show that a binary representation of a number is unique. We give a classification of the space requirements to store a decimal number in the following proposition.

**Proposition 6.2.** *The binary representation of a decimal integer $A \geq 0$ , with $2^{m-1} \leq A < 2^m$, has exactly $m$ significant digits ($m$ digits excluding leading zeros).*

*Proof.* We construct the sequence $A_i$ as above and analyse upper and lower bounds for it to find $m$ such that $A_m = 0$. As each $A_i \neq 0$ corresponds to a significant digit in the binary representation and $A_i$ is strictly decreasing, $m$ is the number of significant digits in the binary representation. We begin with the assumption $2^{m-1} \leq A < 2^m$. As $a_i \in \{0, 1\}$, using equation 6.2;

$$\begin{array}{rccc} & A_0 & < & 2^m \\ \Rightarrow & A_1 & < & 2^{m-1} \\ \vdots & \vdots & \vdots & \vdots \\ \Rightarrow & A_m & < & 2^0 \end{array}$$

Thus $A_m = 0$ and we have at most $m$ significant digits. Also using the assumed lower bound, and applying equation 6.2 iteratively;

$$\begin{aligned}
&\Rightarrow 2^{m-1} \leq A_0 \\
&\Rightarrow \frac{2^{m-1}-1}{2} = 2^{m-2} - \frac{1}{2} \leq A_1 \\
&\Rightarrow \frac{2^{m-2}-1-\frac{1}{2}}{2} = 2^{m-3} - \frac{1}{2} - \frac{1}{2^2} \leq A_2 \\
&\qquad\qquad\qquad\qquad\qquad\qquad \vdots \\
&\Rightarrow 2^{m-m} - \sum_{i=1}^{m-1} 2^{-i} = \frac{1}{2^{m-1}} \leq A_{m-1}
\end{aligned}$$

So $A_{m-1} \neq 0$ and so we need more than $m-1$ binary digits to represent $A$. For the final part of the previous sequence of inequalities we use the sum of a geometric series formula.

Putting together the two parts of the proof, we conclude that we require exactly $m$ binary digits. □

Suppose we know we have exactly $n$ bits, then the previous proposition allows us to work out the size of the numbers we can store, namely all the integers from 0 to $2^n - 1$.

It is possible to add in the ability to encode negative integers into binary also so we can represent any given integer providing we have enough *bits*. There are algorithms which perform operations such as addition, subtraction, multiplication and division. If the reader wishes to find out more about this, there is more information in the first two chapters of [23].

This book also describes the standard method for encoding a large (but finite) set of real numbers into binary using the "IEEE754" standard.

To store characters (such as letters and punctuation), we create a map from the binary (or equivalently decimal) numbers to a set of characters. The first widespread standard for this was the "ASCII" standard, later followed by the "Unicode" standard. More information about these can be found out on their respective wikipedia pages.

6.3. **Qubits.** Throughout this report we have seen many quantum two state systems. For example, $|H\rangle, |V\rangle$ in section 3.1 and $|H\rangle, |T\rangle$ in section 5.1. Relabeling each of these two state systems to $|0\rangle, |1\rangle$, we get a system which can be in a $|1\rangle$ state or a $|0\rangle$ state - a *quantum bit*, or, for short, *qubit*. Note that in contrast to a classical bit, a qubit can also be in an infinity of states in between

0 and 1, but when measured will collapse to be 0 or 1. We call the ordered combination of $m$ of these systems a *quantum register* of length $m$.

**Definition 6.3** (Qubit)**.** A *qubit* is a two-level quantum system $\mathbb{C}^2$ equipped with a fixed basis $B = \{|0\rangle, |1\rangle\}$, called a *computational basis*.

**Definition 6.4** (Quantum Register)**.** A *quantum register* of length $m$ is an ordered system of $m$ qubits. Its state space is $\mathbb{C}^{2^m}$, with basis states

$$\{|x\rangle : x \in \{0,1\}^m\}$$

We can therefore represent any of the objects of the previous section as basis states of a quantum register. We can "write" to the quantum register by setting it up in that state and "read" by setting up the system so that it will collapse to the "written" basis states with probability 1 when measured. One property that our quantum register has, that is not shared by its classical counterpart, is the property that we can entangle the states of our quantum register and then perform operations simultaneously on all our register at once. In classical computing, each bit is completely independent of every other bit and it is impossible to act on an arbitrarily large number of these in one computational step. We should then expect that for certain algorithms, quantum computing gives an exponential speed up. This property is well reflected in the degrees of freedom we get from combining systems of bits in the classical world and the quantum world. The state of a classical register can be described by a vector in the space formed by the cross product of $n$ bits

$$\{0,1\} \times \{0,1\} \times \ldots \times \{0,1\}$$

This gives us an $n$ dimensional space on which we can perform operations.

The state of the quantum register is described by a vector in the space formed by the tensor product of $n$ qubits:

$$\{|0\rangle, |1\rangle\} \otimes \{|0\rangle, |1\rangle\} \otimes \ldots \otimes \{|0\rangle, |1\rangle\}$$

This, however, gives us a $2^n = e^{n \ln 2}$ dimensional space (note the dimensions given in definition 3.6) on which to perform operations. This does not mean we can store more information; we still have the same number of possible 0's and 1's upon measurement. However, our system, in between setup and measurement, contains exponentially more information. As well as causing our system to be much more operable on (which in turn has the possibility to produce faster algorithms), it also means any operation performed on this system is much more complex. The complexity of creating

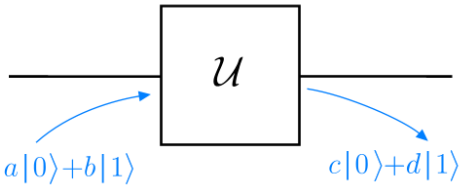


FIGURE 6.2. Representation of a unary gate in a quantum circuit

algorithms left a huge gap between realising the possible advantages of quantum computing that Deutsch discovered in 1985[9] and the discovery of the first quantum algorithm, in 1994[10], which took advantage of this exponential complexity.

The information storage component of our quantum computer is now theoretically complete. We now enroll in the difficult task of operating on this information. We take a slow approach and construct a processing device for our quantum computer from *quantum gates*.

6.4. **Quantum Gates.** We have already seen one device which was labeled a *quantum gate*: the Hadamard-Walsh gate $H_2$. This device acted on one qubit to change its state to a state, which if we measured, would give us a 1 half of the time, and a 0 half of the time. We note here the action of a $H_2$ gate on a basis state $|x\rangle$ where $x \in \{0, 1\}$

$$H_2 |x\rangle = \frac{1}{\sqrt{2}} (|0\rangle + (-1)^x |1\rangle)$$

This will be useful later when we come to constructing *quantum circuits* for our algorithms.

The Hadamard-Walsh gate $H_2$ is an example of a *unary quantum gate*.

**Definition 6.5** (Unary Quantum Gate)**.** A *unary quantum gate* is a an operation on a qubit. It is a unitary mapping $U : \mathbb{C}^2 \to \mathbb{C}^2$.

A unary gate can be represented in a *quantum circuit* as shown in figure 6.2. These will be discussed later after further development of quantum gates.

In classical computing, the only unary gate is the NOT gate. This takes a bit in a state one and transforms it to a state 0, and takes a bit in state 0 and transforms it to the state 1.

**Definition 6.6** (Quantum NOT Gate)**.** The quantum NOT gate is the unitary gate defined by the matrix

$$U_{NOT} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

It acts on the basis states $|0\rangle$ and $|1\rangle$ as follows:

$$|0\rangle \mapsto |1\rangle$$
$$|1\rangle \mapsto |0\rangle$$

and in general, $c_1 |0\rangle + c_2 |1\rangle \mapsto c_2 |0\rangle + c_1 |1\rangle$.

We can expand upon unary gates by considering quantum gates which act upon two qubits.

**Definition 6.7** (Binary Quantum Gate)**.** A *binary quantum gate* is a an operation on two qubits. It is a unitary mapping $U : \mathbb{C}^4 \to \mathbb{C}^4$.

In general;

**Definition 6.8** (Quantum Gate)**.** A *quantum gate* is an operation on $n$ qubits. It is a unitary mapping $U : \mathbb{C}^{2n} \to \mathbb{C}^{2n}$

We shall first consider an extension of our unary NOT gate. It would be useful to be able to control whether the NOT gate is in operation by another bit, called a *control bit*.

**Definition 6.9** (Quantum CNOT Gate)**.** A binary quantum gate acting on two qubits such that the first bit (the *control bit*) controls whether a NOT gate is applied to the second bit (the *target bit*) is called a *controlled NOT gate*, or *CNOT* gate for short. As a matrix acting on a two qubit system it can be written as

$$U_{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Its action on the basis states is as follows:

$$U_{CNOT} |00\rangle = |00\rangle$$
$$U_{CNOT} |01\rangle = |01\rangle$$
$$U_{CNOT} |10\rangle = |11\rangle$$
$$U_{CNOT} |11\rangle = |10\rangle$$

We represent a CNOT gate in a *quantum circuit* as in figure 6.3.

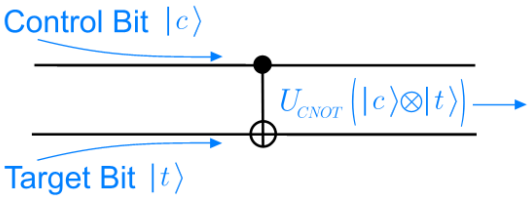


FIGURE 6.3. Representation of a CNOT gate (in black)

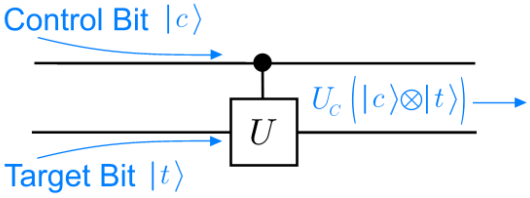


FIGURE 6.4. The representation of a controlled $U$ gate given in black

Using the projection operator given in definition 3.10, write

$$U_{CNOT} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \otimes I + \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \otimes U_{NOT} = |0\rangle\langle 0| \otimes I_2 + |1\rangle\langle 1| \otimes U_{NOT} \tag{6.3}$$

This gives our controlled not gate in terms of the action on the control bit and action on the target bit. We move on to look at other controlled operations. If we have a unary gate $U$, then

$$U_C := |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes U$$

is the gate for which the target bit is acted on by $U$ when the control bit is $|1\rangle$. It is represented in a quantum circuit by figure 6.4

**Proposition 6.10.** *For any $2 \times 2$ unitary matrix $U$, $U_C$ is separable if and only if $U = cI_2$ for some $c \in \mathbb{C}$.*

*Proof.* Suppose we can write the matrix $U_C \neq cI_2$ as the tensor product of two $2 \times 2$ matrices $A$ and $B$. Then,

$$\begin{pmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \end{pmatrix} \otimes \begin{pmatrix} B_{1,1} & B_{1,2} \\ B_{2,1} & B_{2,2} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & U_{1,1} & U_{1,2} \\ 0 & 0 & U_{2,1} & U_{2,2} \end{pmatrix}$$

Amongst others, we get the following equations

$$\begin{aligned} A_{1,1}B &= I_2 \\ A_{2,2}B &= U_C \end{aligned}$$

The first one tells us

$$\begin{aligned} B &= \begin{pmatrix} A_{1,1} & 0 \\ 0 & A_{1,1} \end{pmatrix} \\ \Rightarrow A_{2,2}B &= \begin{pmatrix} A_{2,2}A_{1,1} & 0 \\ 0 & A_{2,2}A_{1,1} \end{pmatrix} \end{aligned}$$

Using the second one, we get

$$U_C = A_{2,2}B = \begin{pmatrix} A_{2,2}A_{1,1} & 0 \\ 0 & A_{2,2}A_{1,1} \end{pmatrix}$$

But then $U_C$ is a complex multiple of the identity, contradicting our original assumption. Thus $U_C \neq cI_2$ is not separable. Now suppose unitary $U_C = cI_2$, then taking

$$B = \begin{pmatrix} c & 0 \\ 0 & c \end{pmatrix}, \quad A = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

we get $U_C = A \otimes B$. □

**Example** (The Hadamard Gate)**.** We can write the controlled Hadamard gate as

$$\begin{aligned} H_{C2} &= |0\rangle\langle 0| \otimes I_2 + |1\rangle\langle 1| \otimes H_2 \\ &= \frac{1}{\sqrt{2}} \begin{pmatrix} \sqrt{2} & 0 & 0 & 0 \\ 0 & \sqrt{2} & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 \end{pmatrix} \end{aligned}$$

This is non-separable and refers to the quantum circuit shown in figure 6.4 with $U = H_2$**.** Now if we simply write $U_{I,H} = I_2 \otimes H_2$, then

$$U_{H,I} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 \end{pmatrix}$$

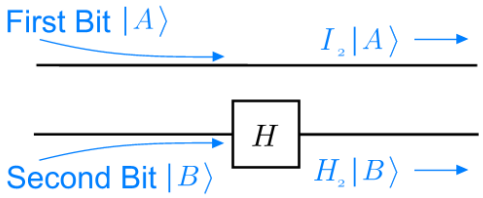


FIGURE 6.5. Hadamard transform acting on second qubit only

This corresponds to the quantum circuit in figure 6.5

The most useful gate in terms of modeling a classical computer is the *CCNOT gate*. We shall see why very soon.

**Definition 6.11** (Quantum CCNOT Gate)**.** The quantum *CCNOT gate* (or *Toffoli gate*) is a controlled CNOT gate. It takes two control bits as its input and one target bit. The target bit is flipped when the two control bits are in the state $|1\rangle$. As a matrix we write it as

$$U_{CCNOT} = (|00\rangle\langle 00| + |01\rangle\langle 01| + |10\rangle\langle 10|) \otimes I_2 + |11\rangle\langle 11| \otimes U_{NOT}$$

Throughout this section we have talked about *quantum circuits*. We finally define one here.

**Definition 6.12** (Quantum Circuit)**.** A *quantum circuit* is a directed graph in which information flows from left to right. Each vertex is a quantum gate, connected by edges, which we call *wires*. If we can construct a circuit with only unary and binary quantum gates, we assume each gate represents a *computational step*[5]. It is common to have the most significant, or control bit being at the top of the diagram.

A quantum circuit is a useful representation of the operations we perform on qubits. It is a diagram allowing an engineer to construct an experiment possible of performing the unitary operation it describes.

There is much theory about the use of gates in classical computers. All processors are built of gates (which are in turn built from transistors). They are so useful in classical computing due to a result by Post in 1941 [24]. This result tells us that any function $f : \{0,1\}^m \to \{0,1\}^n$ can be computed by some circuit using only the gates AND, OR and NOT. If we can construct the quantum counterparts of these gates, we can show that anything possible on a classical computer is possible on a quantum computer.

[5]Note that this is not actually the case, but for our purposes, the estimates it gives are accurate enough.

6.5. **Quantum AND, OR and NOT gates.** We have already seen how to construct a quantum NOT gate, but it will be helpful to construct a more complicated one in terms of a three qubit CCNOT gate.

**Definition 6.13** (Classical AND and OR Gates)**.** A classical AND gate is a function $f_{AND} : \{0,1\}^2 \to \{0,1\}$ such that

$$f_{AND}(x,y) = \begin{cases} 1 & \text{if } x = y = 1 \\ 0 & \text{otherwise} \end{cases}$$

A classical OR gate is a function $f_{OR} : \{0,1\}^2 \to \{0,1\}$ such that

$$f_{OR}(x,y) = \begin{cases} 0 & \text{if } x = y = 0 \\ 1 & \text{otherwise} \end{cases}$$

Both of these output one bit, and thus it is necessary to introduce at least one more *ancilla* bit or qubit in the output which we can discard after its use in the process. Even if we extend $f_{AND}$ and $f_{OR}$ to the domain $\{0,1\}^2$, neither of them are unitary operations as their inverses are undefined (in the sense that they $f_{AND}^{-1}(0)$ has more than one value and $f_{OR}^{-1}(1)$ has more than one value). It is possible, however, to perform a unitary operation on a three qubit system, where two of these qubits are the input qubits and one is the output.

In the case of the AND gate, this gate is simply the CCNOT gate with the target bit set to state $|0\rangle$. We can see this by looking at the action of the CCNOT gate on the two input qubits with the output qubit set to state $|0\rangle$.

$$\begin{aligned} U_{CCNOT}|000\rangle &= |000\rangle \\ U_{CCNOT}|010\rangle &= |010\rangle \\ U_{CCNOT}|100\rangle &= |100\rangle \\ U_{CCNOT}|110\rangle &= |111\rangle \end{aligned}$$

We can then denote our AND gate graphically as in figure 6.6.

Our OR gate is a little more complicated. Notice that if we relabel our 1s as 0s and our 0s as 1s in the definition of $f_{OR}$, we get $f_{AND}$. Transformation wise, this corresponds to putting a NOT gate on each of the inputs before an AND transformation, and a NOT gate on the output after the AND transformation. So our quantum OR gate looks like figure 6.7. We can then write the

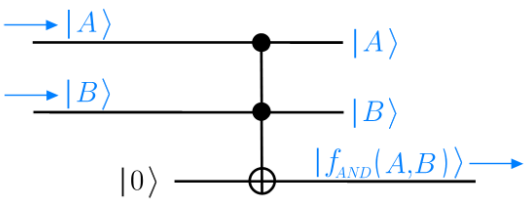


FIGURE 6.6. Quantum AND gate represented in black

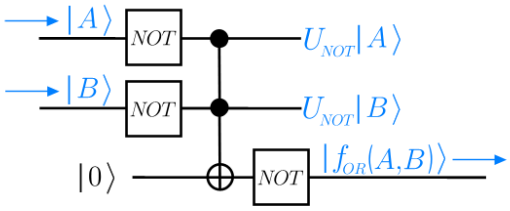


FIGURE 6.7. Quantum OR gate represented in black

quantum OR gate as

$$\begin{aligned} U_{OR} &= (|00\rangle\langle 00| + |01\rangle\langle 01| + |10\rangle\langle 10|) \otimes (U_{NOT} I_2) + |11\rangle\langle 11| \otimes (U_{NOT} U_{NOT}) \\ &= (|00\rangle\langle 00| + |01\rangle\langle 01| + |10\rangle\langle 10|) \otimes U_{NOT} + |11\rangle\langle 11| \otimes I_2 \end{aligned}$$

Finally, we reconstruct our NOT gate in terms of a CCNOT gate. We simply set both the control bits of the CCNOT gate to the state $|1\rangle$, then we get the following:

$$\begin{aligned} U_{CCNOT}\,|110\rangle &= |111\rangle \\ U_{CCNOT}\,|111\rangle &= |110\rangle \end{aligned}$$

Thus we are able to construct the classical AND, OR and NOT gates using only the quantum CCNOT gate. Post [24] tells us that from these gates we can construct any classical gate, and thus for any classical circuit we can construct a quantum circuit which produces equivalent results for equivalent inputs. Our quantum computer can model any classical computer!

There is a similar result concerning the construction of any quantum gate from a set of quantum gates which is given in [25]. We restate this here

**Theorem 6.14.** *All quantum circuits can be constructed using only CCNOT gates and unary gates.*

The proof is long enough so that it is unfeasible to provide it here, but is very accessible and easy to read (it can be found on Google scholar as a PDF).

We conclude this section with one final gate.

**Definition 6.15** (XOR gate)**.** The exclusive OR gate, or XOR gate, is the gate which can be represented by the function $f_{XOR} : \{0,1\}^m \to \{0,1\}$ with

$$f_{XOR}(x,y) = \begin{cases} 0 & \text{if } x = y \\ 1 & \text{otherwise} \end{cases}$$

The XOR gate performs the action of addition modulo 2. From now on we will use the symbol $\oplus$ to denote addition modulo 2 (even though in general it may mean addition modulo $n$). So we have

$$\begin{aligned} 0 \oplus 0 &= f_{XOR}(0,0) = 0 \\ 0 \oplus 1 &= f_{XOR}(0,1) = 1 \\ 1 \oplus 0 &= f_{XOR}(1,0) = 1 \\ 1 \oplus 1 &= f_{XOR}(0,0) = 0 \end{aligned}$$

As before, we need at least one ancilla qubit to allow our transform to be reversible. In this case we only need one and in fact the XOR gate can be given by a CNOT gate, with its output being the target bit of the CNOT gate. To confirm this we look again at the operation of a CNOT gate on a two qubit system.

$$\begin{aligned} U_{CNOT}|00\rangle &= |00\rangle \\ U_{CNOT}|01\rangle &= |01\rangle \\ U_{CNOT}|10\rangle &= |11\rangle \\ U_{CNOT}|11\rangle &= |10\rangle \end{aligned}$$

We see that the second qubit in the result of each of these is the sum, modulo 2, of the two input qubits. Thus we can compute $x \oplus y$ using a CNOT gate.

## 7. Quantum Algorithms

The previous section has already given us a multitude of algorithms we can run on our quantum computer. Any classical algorithm can be written in terms of the quantum circuits we set up and

thus we can at least do all the things a conventional computer can do with our quantum computer. In this section we look at a small number of algorithms which are unique to a quantum computer and seem to provide a massive speed increase compared to their classical counterparts. I would wish to be more original, but with the proven complexity of finding quantum algorithms, I have had to settle for using other peoples algorithms. This section mostly follows [15] and [1].

7.1. **The Deutsch Algorithm.** The *Deutsch algorithm* takes its name from a character we have met many times already and was one of the first quantum algorithms which showed that quantum algorithms could be more efficient than classical algorithms. Let $f : \{0,1\} \to \{0,1\}$ be a binary function. We have four possibilities for the function, two of which are *constant* (mapping all of the co-domain to either 0 or 1), and the other two we call *balanced*. To figure out whether a given function $f(x)$ is constant or balanced we need to put both $x = 0$ and $x = 1$ into $f(x)$. If both map to the same value then our function is constant, otherwise our function is balanced. This would be the classical approach. Deutsch, however devised a quantum approach only requiring one function call, with a superposition of both $x = 0$ and $x = 1$.

Construct a unitary operator $U_f : |x, y\rangle \mapsto |x, y \oplus f(x)\rangle$. We can then see from its matrix representation

$$U_f = \begin{pmatrix} f(0)\oplus 1 & f(0) & 0 & 0 \\ f(0) & f(0)\oplus 1 & 0 & 0 \\ 0 & 0 & f(1)\oplus 1 & f(1) \\ 0 & 0 & f(1) & f(1)\oplus 1 \end{pmatrix}$$

that it is unitary. For brevity we write $f(x) \oplus 1 = f^{\neg}(x)$, so that we can write

$$U_f = |0\rangle\langle 0| \otimes \begin{pmatrix} f^{\neg}(0) & f(0) \\ f(0) & f^{\neg}(0) \end{pmatrix} + |1\rangle\langle 1| \otimes \begin{pmatrix} f^{\neg}(1) & f(1) \\ f(1) & f^{\neg}(1) \end{pmatrix} \tag{7.1}$$

The problem then becomes finding out what $U_f$ is, and our four possibilities are

$$\begin{aligned} U_f &= |0\rangle\langle 0| \otimes I_2 + |1\rangle\langle 1| \otimes U_{NOT} && \text{if } f = id \\ U_f &= |0\rangle\langle 0| \otimes U_{NOT} + |1\rangle\langle 1| \otimes I_2 && \text{if } f^{\neg} = id \\ U_f &= |0\rangle\langle 0| \otimes I_2 + |1\rangle\langle 1| \otimes I_2 = I_4 && \text{if } f = 0 \\ U_f &= |0\rangle\langle 0| \otimes U_{NOT} + |1\rangle\langle 1| \otimes U_{NOT} && \text{if } f = 1 \end{aligned}$$

We set up a state

$$|s_0\rangle = \frac{1}{2}\left(|00\rangle - |01\rangle + |10\rangle - |11\rangle\right)$$

in a two qubit system. This could be done by applying a transform $H_2 \otimes H_2$ to an initial state $|01\rangle$ for example. After applying our unknown $U_f$ we get

$$\begin{aligned} |s_1\rangle =& U_f\,|s_0\rangle \\ =& \frac{1}{2}\left(|0, f\,(0)\rangle - |0, f^{\neg}\,(0)\rangle + |1, f\,(1)\rangle - |1, f^{\neg}\,(1)\rangle\right) \end{aligned}$$

Now, if we apply a transform $H_2 \otimes I_2$, then we get the state

$$\begin{aligned} |s_2\rangle &= (H_2 \otimes I_2)\,|s_1\rangle \\ &= \frac{1}{2\sqrt{2}}\left((|0\rangle + |1\rangle) \otimes (|f\,(0)\rangle - |f^{\neg}\,(0)\rangle) + (|0\rangle - |1\rangle) \otimes (|f\,(1)\rangle - |f^{\neg}\,(1)\rangle)\right) \end{aligned}$$

Suppose $f$ is constant, in which case $|f\,(0)\rangle = |f\,(1)\rangle$, then our state will be

$$\begin{aligned} |s_2\rangle &= \frac{1}{2\sqrt{2}}\left(2\,|0, f\,(0)\rangle - |0, f^{\neg}\,(0)\rangle\right) \\ &= \frac{1}{\sqrt{2}}\,|0\rangle \otimes (|f\,(0)\rangle - |f^{\neg}\,(0)\rangle) \end{aligned}$$

If $f$ is balanced (the case $|f^{\neg}\,(0)\rangle = |f\,(1)\rangle$), then

$$\begin{aligned} |s_2\rangle &= \frac{1}{2\sqrt{2}}\left(2\,|1, f\,(0)\rangle - |1, f^{\neg}\,(0)\rangle\right) \\ &= \frac{1}{\sqrt{2}}\,|1\rangle \otimes (|f\,(0)\rangle - |f^{\neg}\,(0)\rangle) \end{aligned}$$

Thus by measuring the first bit we can determine whether $f$ is constant or balanced. Measuring a $|0\rangle$ means $f$ must be constant, whilst measuring a $|1\rangle$ means $f$ must be balanced. So long as we set up our states perfectly, perform operations perfectly and measure perfectly, we will get the same measurement every time.

Note the similarity between equation 7.1 and the equation given for a CNOT gate (equation 6.3). The NOT gate operates on the second bit if $f\,(1) = 1$ and on the first bit if $f\,(0) = 1$. We represent *Deutsch's black box function* $U_f$ as in figure 7.1.

7.2. **The Quantum Fourier Transform.** Fourier transforms are useful for many applications, from interpreting physical data (for example studying crystal structure) to data compression to

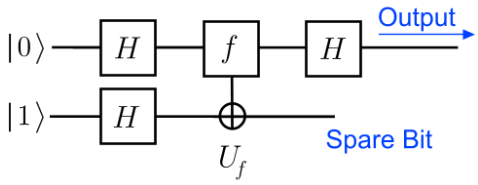


FIGURE 7.1. The quantum circuit for running Deutsch's algorithm

number theory (we shall see an example of this in section 7.3). The use of *fast Fourier transforms* (efficient algorithms for calculating discreet Fourier transforms) is extensive in computer science. We shall now find an algorithm for computing the discrete Fourier transform of a function $f : G \to \mathbb{C}$, where $G$ is a set $\{0, 1, \ldots, 2^n - 1\}$ with $2^n$ elements.

Suppose we encode each element of $G$ into binary (as we did in section 6.2) so that we can describe our set $G$ by multi-qubit system, with basis states $|0_B\rangle, |1_B\rangle, \ldots, |(2^n-1)_B\rangle$, then by proposition 6.2, we only need $n$ qubits. We can then represent our function $f$ as the phase factors of these basis states so that the state

$$c_0 |0_B\rangle + c_1 |1_B\rangle + \ldots + c_{n-1} |(2^n-1)_B\rangle \quad \text{where } f(k) = c_k,\ k = 0, 1, \ldots, 2^n - 1$$

can be seen as equivalent to the function $f$. We can now define the quantum Fourier transform of $f$.

**Definition 7.1** (Quantum Fourier Transform (QFT))**.** The quantum Fourier transform is the operation transforming the basis states as follows:

$$f(k) |k_B\rangle \mapsto \hat{f}(k) |k_B\rangle$$

where

$$\hat{f}(k) := \frac{1}{\sqrt{2^n}} \sum_{j=0}^{2^n-1} f(j)\, e^{-\frac{2\pi kji}{2^n}} \tag{7.2}$$

We call the $\hat{f}(k)$ the *Fourier coefficients*.

Our first inclination should be to enquire whether this is a unitary operation. It is certainly linear as the definition of $\hat{f}$ shows. The question as to whether it is unitary is slightly more difficult. It is possible to prove that operators that preserve norms are unitary (see pp120 of [15]), and then *Parseval's identity* (see [26]) tells us that $\left\|\hat{f}(k)\right\| = \|f(k)\|$ for $k = 0, 1, \ldots, n$. Thus we should

be able to find a quantum circuit implementing this transform, which we could write down as the matrix

$$U_{QFT,n} = \frac{1}{\sqrt{2^n}} \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & e^{-\frac{2\pi i}{2^n}} & e^{-\frac{4\pi i}{2^n}} & \cdots & e^{-\frac{2^n \pi i}{2^n}} \\ 1 & e^{-\frac{4\pi i}{2^n}} & e^{-\frac{8\pi i}{2^n}} & \cdots & e^{-\frac{2^{n+1} \pi i}{2^n}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & e^{-\frac{2(n-1)\pi i}{2^n}} & e^{-\frac{4(n-1)\pi i}{2^n}} & \cdots & e^{-\frac{2^{(n-1)^2+1} \pi i}{2^n}} \end{pmatrix}$$

For $n = 1$, we have

$$U_{QFT,1} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} = H_2$$

For $n = 2$, we have

$$U_{QFT,2} = \frac{1}{2} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -i & -1 & i \\ 1 & -1 & 1 & -1 \\ 1 & i & -1 & -i \end{pmatrix}$$

In this case it would be useful to decompose this into smaller transforms so that we can implement it with binary and unary quantum gates. For this, we will need to introduce a new unary gate.

**Definition 7.2** ($B_{k,l}$ gate)**.** The $B_{k,l}$ gate is given by the matrix

$$\begin{pmatrix} 1 & 0 \\ 0 & e^{-i\theta_{k,l}} \end{pmatrix} \quad \text{where } \theta_{k,l} = \frac{\pi}{2^{l-k}}$$

We can then write the controlled $B_{k,l}$ gate as $\phi_{k,l} = |0\rangle \langle 0| I_2 + |0\rangle \langle 0| B_{k.l}$

*Remark.* $B_{k,l}$ is a generalisation of the Hadamard gate with $H_2 = B_{0,0} = B_{1,1} = \ldots$.

We quickly look at the action of $B_{k,l}$ on a qubit.

$$B_{k,l} |x\rangle = \begin{cases} |x\rangle & \text{if } x = 0 \\ e^{-i\theta_{k,l}} |x\rangle & \text{if } x = 1 \end{cases}$$

Notice that we can write the action of $U_{QFT,2}$ on a state $|x_B\rangle$ as

$$U_{QFT,2} |x_B\rangle = \frac{1}{2} \left( |0\rangle + e^{-i\theta_{1,1}x} |1\rangle \right) \otimes \left( |0\rangle + e^{-i\theta_{1,0}x} |1\rangle \right)$$

Now from equation 6.1, we can write $x = 2x_1 + x_0$, where $x_0, x_1 \in \{0, 1\}$ are the binary digits of $x$, so that $x_B = x_1 x_0$. Then

$$\begin{aligned}(7.3)\quad U_{QFT,2}\left|x_B\right\rangle &= U_{QFT,2}\left|x_1x_0\right\rangle = \frac{1}{2}\left(\left|0\right\rangle + e^{-\pi i(2x_1+x_0)}\left|1\right\rangle\right) \otimes \left(\left|0\right\rangle + e^{-\frac{\pi}{2}i(2x_1+x_0)}\left|1\right\rangle\right) \\ &= \frac{1}{2}\left(\left|0\right\rangle + e^{-\pi i x_0}\left|1\right\rangle\right) \otimes \left(\left|0\right\rangle + e^{-\pi i x_1}e^{-\frac{\pi}{2}ix_0}\left|1\right\rangle\right) \\ &= \frac{1}{2}\left(\left|0\right\rangle + (-1)^{x_0}\left|1\right\rangle\right) \otimes \left(B_{0,1}\right)^{x_0}\left(\left|0\right\rangle + (-1)^{x_1}\left|1\right\rangle\right)\end{aligned}$$

How does the transform $(B_{0,1})^{x_0}$ act? The answer is exactly as a controlled $B_{0,1}$ gate with $|x_0\rangle$ as the control bit and $|x_1\rangle$ as the target bit. The second part we need to worry about is the $(-1)^{x_1}$. We note that

$$\left(I_2 \otimes B_{1,1}\right)\left|x_1x_0\right\rangle = \left(I_2 \otimes H_2\right)\left|x_1x_0\right\rangle = \left|x_1\right\rangle \otimes \frac{1}{\sqrt{2}}\left(\left|0\right\rangle + (-1)^{x_0}\left|1\right\rangle\right)$$

Applying the operation $\phi_{0,1}$, we get

$$\phi_{0,1}\left(I_2 \otimes B_{1,1}\right)\left|x_1x_0\right\rangle = \left|x_1\right\rangle \otimes \frac{1}{\sqrt{2}}\left(B_{0,1}\right)^{x_1}\left(\left|0\right\rangle + (-1)^{x_0}\left|1\right\rangle\right)$$

Perform the operation $(H_2 \otimes I_2) = (B_{0,0} \otimes I_2)$ to get

$$\left(B_{0,0} \otimes I_2\right)\phi_{0,1}\left(I_2 \otimes B_{1,1}\right)\left|x_1x_0\right\rangle = \frac{1}{2}\left(\left|0\right\rangle + (-1)^{x_1}\left|1\right\rangle\right) \otimes \left(B_{0,1}\right)^{x_1}\left(\left|0\right\rangle + (-1)^{x_0}\left|1\right\rangle\right)$$

Now, we are almost back to equation 7.3. There is one problem however, our $x_0$ and $x_1$ are in the wrong places on the right hand side, but in the correct places on the left hand side. The final piece of the jigsaw is therefore adding a gate which can swap our two qubits ($|x_0\rangle \mapsto |x_1\rangle$, $|x_1\rangle \mapsto |x_0\rangle$) before we perform any of the other operations.

**Definition 7.3** (SWAP Gate)**.** The swap gate $U_{SWAP}$ is the matrix

$$U_{SWAP} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Its action takes a state $|x_0x_1\rangle$ to the state $|x_1x_0\rangle$

We have finally made it. Comparing

$$\left(B_{0,0} \otimes I_2\right)\phi_{0,1}\left(I_2 \otimes B_{1,1}\right)U_{SWAP}\left|x_1x_0\right\rangle = \frac{1}{2}\left(\left|0\right\rangle + (-1)^{x_0}\left|1\right\rangle\right) \otimes \left(B_{0,1}\right)^{x_0}\left(\left|0\right\rangle + (-1)^{x_1}\left|1\right\rangle\right)$$

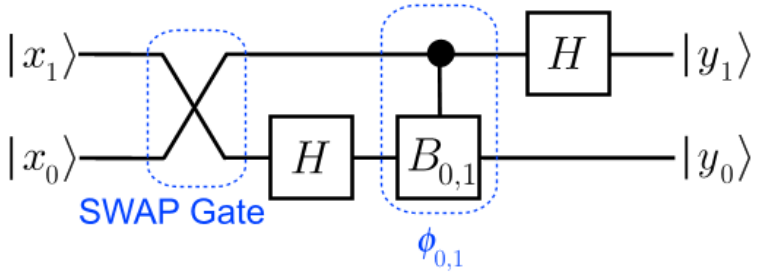


FIGURE 7.2. The quantum circuit representing a quantum Fourier transform on two bits

to equation 7.3, we see

$$U_{QFT,2} = (B_{0,0} \otimes I_2)\, \phi_{0,1}\, (I_2 \otimes B_{1,1})\, U_{SWAP}$$

In figure 7.2 we can see the swap gate labeled and the quantum circuit corresponding to $U_{QFT,2}$.

The construction of quantum circuit for a QFT on $2^n$ qubits follows a similar method to the one above. We give an outline of the method (similar to pp54 to pp57 of [15]) here:

(1) We prove the following

$$\sum_{y=0}^{2^{n-1}} e^{-\frac{2\pi i x y}{2^m}} |y_B\rangle = \left(|0\rangle + e^{-\frac{\pi i x}{2^0}} |1\rangle\right) \otimes \left(|0\rangle + e^{-\frac{\pi i x}{2^1}} |1\rangle\right) \otimes \ldots \otimes \left(|0\rangle + e^{-\frac{\pi i x}{2^{m-1}}} |1\rangle\right)$$

(2) We consider the phase factor of the state $|1\rangle$ in the $l^{\text{th}}$ bracket above, after being multiplied out. This is the $l^{\text{th}}$ bit and thus we write it as $x_l$, so that

$$x = \sum_{i=0}^{n-1} 2^i x_i \quad x_i \in \{0, 1\}$$

We prove that the phase factor of the $l^{\text{th}}|1\rangle$ can be written as

$$(-1)^{x_{l-1}} \cdot \exp\left(-\frac{\pi i x_{l-2}}{2^1}\right) \cdots \exp\left(-\frac{\pi i x_1}{2^{l-2}}\right) \cdot \exp\left(-\frac{\pi i x_0}{2^{l-1}}\right)$$

(3) We construct the quantum circuit which applies the transform above using the transform $\phi_{k,l}$, along with Hadamard gates.

Hirvensalo argues that we can ignore the swapping gates and either measure the result with a measurement device set upside down, or construct any circuit which follows so that it takes the most significant bit as its input at the bottom of the circuit. We can see the quantum circuit implementing $U_{QFT,n}$ in figure 7.3.

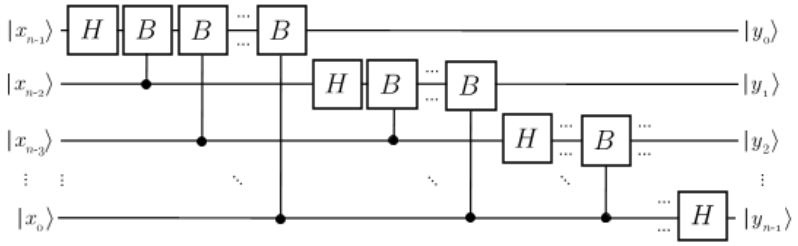


FIGURE 7.3. Quantum Circuit for $U_{QFT,n}$ with the swap not included (adapted from [15]). The coefficients subscript for each $B$ is not written in, but they form the sequence taking the subscript of the $(x_i, x_j)$ they connect (from left to right), $(n-1, n-2), (n-1, n-3), \ldots, (n-1, 0), (n-2, n-3), \ldots, (n-2, 0), \ldots,$ $(n-3, 0)$

We see that we have $n + (n-1) + (n-2) + \ldots + 2 + 1$ components in the quantum circuit for $U_{QFT,n}$. Summing this arithmetic series, we find that our quantum circuit takes

$$\frac{n(n+1)}{2}$$

computational steps to complete. Originally we would have expected over $2^n = e^{n \ln 2}$ computational steps, as if we look at equation 7.3, we have a sum from 0 to $2^n - 1$, in which we perform $2^n - 1$ and $2^n - 1$ multiplications, plus many more operations. Even fast Fourier transforms done on classical computers require around $2^n$ computational steps. This is one of the examples where using a quantum computer can "exponentially speed up" solving a problem.

**Definition 7.4** (Polynomial Time and Exponential Time Algorithms)**.** We describe an algorithm as a *polynomial time algorithm* if for an input of length $n$, we can write the number maximum number of computational steps it take to complete in the form of a polynomial of $n$. If we can write this as an exponential in $n$ then we call our algorithm a *exponential time algorithm*. We do however use the two terms exclusively. A polynomial time algorithm is also an exponential time algorithm, but we use the former term to describe it.

We should also note a fundamental difference between the quantum Fourier transform and the classical Fourier transform. The quantum Fourier transform stores its information in the phase factors of the bits, meaning we can be as accurate as we like, but also meaning we can not read off their values straight away. The classical Fourier transform however stores its results in bits, and thus we need many more. In fact, if each complex number requires $m$ bits, then we need $nm$

bits just to store the results. In the quantum case, we only need $n$ qubits, reducing our space requirements significantly.

7.3. **An Overview of Shor's Algorithm.** In this section we do not look in detail at Shor's algorithm, however we do give an example showing the factorisation of 15; quite a trivial example but the one which was used by IBM in the demonstration of their quantum computer. Most of the section is a shortening of what I have seen in [15] and [1], along with my own understanding of Fourier series.

We know that it is relatively easy to multiply together two prime numbers. A computer in fact can calculate products of large prime numbers very quickly. It is, however, very difficult to factorise this product back into its two primes without knowing its factors in the first place. RSA encryption, which is used for sending information securely, is based upon this difficulty. The idea is based on the two parties knowing some properties of these factors, so that they can encrypt information, send it, and decrypt it again. The task for anyone who wants to intercept the information but does not know the properties of these factors, or the factors themselves, is to find these factors. The classical algorithm to do this takes an exponential (in the number of digits in the product) number of computational steps to do this. So that for a large number, it is very difficult to find its prime factors.

We look at a quantum algorithm which can find these factors much faster than any known classical algorithm. This algorithm is *Shor's algorithm* [10]. It follows much the same steps as a classical factorisation algorithm, but with one step being performed on a quantum computer.

The algorithm takes an integer $N = pq$, where $p$ and $q$ are prime. The algorithm follows the following steps

(1) Choose a random positive integer $m < N$ . Calculate the greatest common factor $\gcd(m, N)$ using the Euclidean algorithm. If $\gcd(m, N) \neq 1$, then we have found a factor, thus $m$ is either $p$ or $q$. For large $N$ this is very unlikely, so we assume $\gcd(m, N) = 1$.
(2) Let $f_N : \mathbb{N} \to \mathbb{N}$ be the function taking

$$a \mapsto m^a \bmod N$$

Find the period $P \in \mathbb{Z}_{>0}$. i.e. the number $P$ such that $m^P = 1 \bmod N$.

(3) Assume $P$ is even and $m^{\frac{P}{2}}+1 \neq 0 \bmod N$. If either of these do not hold, we need to repeat 1. and 2. until we find an even $P$ satisfying $m^{\frac{P}{2}}+1 \neq 0 \bmod N$. If $P$ is even, we can write

$$\left(m^{\frac{P}{2}}-1\right)\left(m^{\frac{P}{2}}+1\right) = m^P - 1 = 0 \bmod N$$

and thus $m^{\frac{P}{2}}+1$ has a prime factor $p$ or $q$ and is less than $N$. The greatest common factor

$$d = \gcd\left(m^{\frac{P}{2}}-1, N\right)$$

is either $p$ or $q$. It is then the trivial process of division to find the other factor.

It can be shown that 1. and 3. take polynomial time on a classical computer, but step 2 is an exponential time algorithm. It is this step we look at improving by using our quantum computer. We call this algorithm the *period finding algorithm*. The question is how to implement it on a quantum computer. It is well known that Fourier transforms are useful in period finding. First note that our number $N < 2^n$ (for some $n$) can be stored in $n$ qubits. We set up two quantum registers $|A\rangle, |B\rangle$ of $n$ qubits each. Each of these we set to be all zeros.

$$|A\rangle = |B\rangle = |\{0\}^n\rangle$$

Let us apply the quantum Fourier transform to the first register, $|A\rangle$, only. The state of our system becomes

$$\left(U_{QFT,n} \otimes I_n\right)|AB\rangle = \frac{1}{\sqrt{2^n}} \sum_{a=0}^{2^n-1} |a\rangle\,|B\rangle$$

Notice that because the state $|A\rangle$ was initialised to zero, the quantum Fourier transform gives all the Fourier coefficients as $\frac{1}{\sqrt{2^n}}$.

We have our number $m$ which we wish to calculate the period $P$ of. Define a function $f : \{0,1,\ldots,2^n\} \to \{0,1,\ldots,N-1\}$, $f : k \mapsto m^k \bmod N$. As we did in section 7.1, we define an operation $U_f : |a,b\rangle \mapsto |a, b \oplus f(a)\rangle$. Of course, in this case $b = 0$, and the transformation becomes $U_f : |a,b\rangle \mapsto |a, f(a)\rangle$. Now we apply this to the previous state to get

$$U_f\left(U_{QFT,n} \otimes I_n\right)|AB\rangle = \frac{1}{\sqrt{2^n}} \sum_{a=0}^{2^n-1} |a\rangle\,|f(a)\rangle$$

Our final operation is the quantum Fourier transform. We reach the state

$$\left(U_{QFT,n} \otimes I_n\right) U_f \left(U_{QFT,n} \otimes I_n\right) |AB\rangle = \frac{1}{2^n} \sum_{x=0}^{2^n-1} \sum_{a=0}^{2^n-1} e^{-\frac{\pi i a x}{2^n}} |x\rangle |f(a)\rangle$$

Although we will not go into the details here, the Fourier series of a function creates an approximation (before the limit is taken $n \to \infty$) of that function by writing it as a linear combination of sine (in the imaginary numbers) and cosine (in the real numbers) functions. If our periodic function $f$ has period $k$, then the Fourier coefficients corresponding to cosine functions with period $k$ will be large. However, so will coefficients corresponding to $0, k, 2k, 3k, \ldots$. Thus the probabilities of observing states $|0_B\rangle$, $|P_B\rangle$, $|(2P)_B\rangle$, ... will be high when making a measurement on our first quantum register. The solution to this problem is a number-theoretical algorithm which is given in [15].

To show how this works, we take an example of factoring 15 (from [15]).

**Example.** Let $N = 15$ and take $m = 7$. We thus have $n = 5$.

$$\left(U_{QFT,n} \otimes I_n\right) |AB\rangle = \frac{1}{4} \sum_{a=0}^{15} |a_B\rangle |0\rangle$$

Applying $U_f$, we get

$$\begin{aligned} U_f \left(U_{QFT,n} \otimes I_n\right) |AB\rangle &= \frac{1}{4} \left( \left( |0_B\rangle + |4_B\rangle + |8_B\rangle + |12_B\rangle \right) \otimes |1_B\rangle \right. \\ &\quad + \left( |1_B\rangle + |5_B\rangle + |9_B\rangle + |13_B\rangle \right) \otimes |7_B\rangle \\ &\quad + \left( |2_B\rangle + |6_B\rangle + |10_B\rangle + |14_B\rangle \right) \otimes |4_B\rangle \\ &\quad \left. + \left( |3_B\rangle + |7_B\rangle + |11_B\rangle + |15_B\rangle \right) \otimes |13_B\rangle \right) \end{aligned}$$

Hence showing us how we can computer $f$ in one operation. The problem is now removing the information from the system. We do get part way towards this by taking a Fourier transform

$$\begin{aligned} \left(U_{QFT,n} \otimes I_n\right) U_f \left(U_{QFT,n} \otimes I_n\right) |AB\rangle &= \frac{1}{4} \left( \left( |0_B\rangle + |4_B\rangle + |8_B\rangle + |12_B\rangle \right) \otimes |1_B\rangle \right. \\ &\quad + \left( |0_B\rangle + i\,|4_B\rangle - |8_B\rangle - i\,|12_B\rangle \right) \otimes |7_B\rangle \\ &\quad + \left( |0_B\rangle - |4_B\rangle + |8_B\rangle - |12_B\rangle \right) \otimes |4_B\rangle \\ &\quad \left. + \left( |0_B\rangle - i\,|4_B\rangle - |8_B\rangle + i\,|12_B\rangle \right) \otimes |13_B\rangle \right) \end{aligned}$$

Now, measuring the first register, we see that we get each of $|0_B\rangle, |4_B\rangle, |8_B\rangle, |12_B\rangle$ with probability $\frac{1}{4}$. The result $|0_B\rangle$ gives us no information, however the other three states all have greatest common divisor 4. We check the result. $7^4 \bmod 15 = 2401 \bmod 15 = 1$. Is this true for any other lower powers of 7?

$$\begin{aligned} 7^1 \bmod 15 &= 1 \\ 7^2 \bmod 15 &= 49 \bmod 15 = 4 \\ 7^3 \bmod 15 &= 343 \bmod 15 = 13 \end{aligned}$$

Thus we can conclude that the period of $f$ is 4.

This example is an important example and was used by IBM in their first quantum computer, which we will look at briefly later.

The final thing we note about this algorithm, is the number of computational steps it would take to complete. Much of the algorithm can be run on a classical computer in polynomial time. The part we were interested in improving was step 2. This involved us running a quantum algorithm with two Fourier transforms and an application of $U_f$. We know that each of these Fourier transforms is a polynomial-time algorithm, and with reference to equation 7.1, we can see that $U_f$ should be made up of $N-1$ binary gates. The final step in finding the period from our measurement is run on a classical computer in polynomial time also. We should think, therefore, that our algorithm, which is composed of many polynomial time sub-algorithms, should be polynomial time itself. We are correct in our thoughts, and once again it is down to the idea of *quantum parallelism*, that we can complete the algorithm exponentially faster on a quantum computer than on a classic computer.

## 8. Quantum Corollaries

Developing the structure of a quantum computer has lead us through many interesting ideas and much of our progress through section 6 seemed guided towards constructing a computer quite similar to a classical computer. We did however see some a number of fundamental differences. In this section we look at the implications of these differences and their applications.

8.1. **The No-Cloning Theorem.** Suppose we were able to copy the state of a system to another system without destroying the original system, then we would have the ability to measure the

copied system and measure it without destroying the first. This would prove very useful in applications where we would like to make intermediate measurements or more than one measurement on entangled states. Thanks to a result by Wootters [27], we can now prove that this is impossible.

**Theorem 8.1** (The No-Cloning Theorem)**.** *There is no unitary operation* $U : \mathbb{C}^{2n} \to \mathbb{C}^{2n}$, $n > 1$ *for which* $U(|x\rangle |y\rangle) = U(|x\rangle |x\rangle)$ *for all* $|x\rangle \in \mathbb{C}^n$.

*Proof.* We prove this by contradiction. Suppose that the operation exists for $n > 1$. As $n > 1$, we can find an orthogonal state$|x'\rangle$ to $|x\rangle$, then we have

$$U(|x\rangle |x\rangle) = |x\rangle |x\rangle, \quad U(|x\rangle |x'\rangle) = |x\rangle |x\rangle$$

Then the action of $U$ on a superposition $\frac{1}{\sqrt{2}}(|x\rangle + |x'\rangle)$, is given as

$$\begin{aligned} U\left(\frac{1}{\sqrt{2}}(|x\rangle + |x'\rangle) \otimes |x\rangle\right) &= \left(\frac{1}{\sqrt{2}}(|x\rangle + |x'\rangle)\right) \otimes \left(\frac{1}{\sqrt{2}}(|x\rangle + |x'\rangle)\right) \\ &= \frac{1}{2}(|xx\rangle + |xx'\rangle + |x'x\rangle + |x'x'\rangle) \end{aligned}$$

Now since $U$ is linear, we have

$$\begin{aligned} U\left(\frac{1}{\sqrt{2}}(|x\rangle + |x'\rangle) \otimes |x\rangle\right) &= \frac{1}{\sqrt{2}}U(|x\rangle |x\rangle) + \frac{1}{\sqrt{2}}U(|x'\rangle |x\rangle) \\ &= \frac{1}{\sqrt{2}}|xx\rangle + \frac{1}{\sqrt{2}}|x'x\rangle \end{aligned}$$

These two equations form a contradiction and we therefore conclude that $U$ does not exist. □

It is however possible to perform a copy $U : \mathbb{C}^4 \to \mathbb{C}^4$ of just the basis states. Assume that the second qubit is in state $|0\rangle$, then $U$ could be any unitary transform satisfying

$$U|00\rangle \mapsto |00\rangle, \quad U|10\rangle \mapsto |11\rangle$$

One example of which is the quantum OR gate given in section 6.5. This allows us to copy basis states only, but not a superposition of $|0\rangle$ and $|1\rangle$. This can be quite easily extended to copying any number of bits and thus allows our quantum computer to copy data.

The idea that states cannot be copied is equivalent to the deterministic nature of quantum time evolution. From a thermodynamical point of view, this corresponds to the system maintaining the same energy throughout. If time evolution was not deterministic, we would have lost some information along the way, meaning that the energy that went with this information would have to

be changed into another form. In classical computers, this is generally heat. Every time we delete a piece of data, we are losing the determinism which would be able to tell us where that piece of data came from, so for example, applying AND and OR gates in a classical computer creates heat, and thus using classical methods, there is a lower limit on how many of these gates we can put in a processor.

Quantum computing has the advantage that there is theoretically no lower limit (in terms of this energy release) as to how many gates we can put on processor. Throughout the history of the classical computer, engineers have battled to lower the temperature of processors, but in quantum computing, there seems to be much research into warming our computers up. This is of course directly related to the fact that if we introduce energy into a quantum system, it will act in unpredictable ways. If our system does become subject to this external influence, then the system is said to *decohere*.

We can actually use the idea that we can copy basis states to help us overcome the effects of decoherence. If we start with a qubit of state $|0\rangle$, we may find that due to interference it flips to a state $|1\rangle$. To overcome this, we could copy the initial state, so that we have, say, three zero qubits; $|000\rangle$. Now, any interference which causes a bit flip, say of the first bit, will create a state $|100\rangle$, but because two of the bits are still $|0\rangle$, we can guess that this should have probably been a $|000\rangle$. This is where the work into *quantum error-correcting codes* begins, for more information see chapter 10 of [1].

8.2. **EPR and Bell's Inequality.** In [17], Einstein raises the issue that entanglement seemingly breaks his idea that information can, at most, travel at the speed of light. Quantum mechanics declares that measuring one state at one point in space and time instantly determines another state at another point in space in time. We are given two photons entangled in a state $1/\sqrt{2}\,(|H_1H_2\rangle + |V_1V_2\rangle)$, or in quantum computational terms in state $1/\sqrt{2}\,(|00\rangle + |11\rangle)$ as we saw in section 5.3. Suppose both are traveling away from each other. Measuring one of these photons and seeing it is in the $|0\rangle$ state necessarily means the system is in the $|00\rangle$ state, and so the other photon must also being in the $|0\rangle$ state.

As any sensible person might argue, the states of the two photons must have been decided when they were entangled so that they carry the information regarding the state of the system with them. For this Einstein proposed a hidden variables argument; there are some properties of the photons we do not know how to (or can not) measure, which carry information when telling them how to react when confronted by a measuring device. In his paper [28], Bell constructs a theory of how the quantum world works, starting from the assumptions of locality and hidden variables. He came

to a conclusion called *Bell's theorem* in which he provides an inequality, called *Bell's inequality*, which can be experimentally tested. If the inequality holds, then the theory of quantum mechanics is disputed. If it does not hold, then the hidden variables argument is deemed to be false. Much work has been done in testing this inequality, including [20]. The evidence leans strongly in the direction of quantum mechanics.

8.3. **Quantum Key Distribution.** Suppose we wish to send some information securely from one person, called *Alice*, to another person, called *Bob*. We store this information in binary as $\psi = (\psi_1, \psi_2, \ldots, \psi_n)$, and also set up a binary key $\chi = (\chi_1, \chi_2, \ldots, \chi_n)$ (with the same number of bits as $\psi$) that only Alice and Bob know. Now to encrypt this message, Alice performs addition modulo two between the corresponding bits in $\psi$ and $\chi$, to get a new key

$$\Psi = (\psi_1 \oplus \chi_1, \psi_2 \oplus \chi_2, \ldots, \psi_n \oplus \chi_n)$$

This is the data she sends to Bob. Bob receives this data, and performs addition modulo two between the corresponding bits in $\psi$ and $\chi$, to get

$$\begin{aligned}(\Psi_1 \oplus \chi_1, \Psi_2 \otimes \chi_2, \ldots, \Psi_n \oplus \chi_n) &= (\psi_1 \oplus \chi_1 \oplus \chi_1, \psi_2 \oplus \chi_2 \oplus \chi_2, \ldots, \psi_n \oplus \chi_n \oplus \chi_n) \\ &= (\psi_1, \psi_2, \ldots, \psi_n) = \psi\end{aligned}$$

So that knowing the key allows Bob to retrieve the message originally sent.

The problem here is that we need to communicate the key between the two senders. If there is a third party, *Eve*, who wishes to read the message from Alice to Bob, then Eve will wish to acquire the key. Any classical communications channel is prone to eavesdropping, whether it be by post, over the Internet, or any other way. However, with a quantum channel from Alice to Bob we can construct a secure key for them both to use and also tell whether Eve is listening in.

We construct a system whereby Alice can send linearly polarised photons to Bob. Alice has a random set of qubits $|R\rangle = |R_1\rangle \otimes |R_2\rangle \otimes \ldots \otimes |R_n\rangle$ and a key made from qubits $|\chi\rangle = |\chi_1\rangle \otimes |\chi_2\rangle \otimes \ldots \otimes |\chi_n\rangle$. Each of the $|R_i\rangle$ tell Alice which of the following coding systems to use:

(1) $|0\rangle \mapsto |\updownarrow\rangle\,, \quad |1\rangle \mapsto |\leftrightarrow\rangle$
(2) $|0\rangle \mapsto \frac{1}{\sqrt{2}}|\updownarrow\rangle + \frac{1}{\sqrt{2}}|\leftrightarrow\rangle\,, \quad |1\rangle \mapsto \frac{1}{\sqrt{2}}|\updownarrow\rangle - \frac{1}{\sqrt{2}}|\leftrightarrow\rangle$

Which we see, for each $i = 1, 2, \ldots, n$, is essentially a controlled Hadamard gate on the system $|R_i\rangle \otimes |\chi_i\rangle$ followed by a machine which fires photons polarised as by the following rule $|0\rangle \mapsto |\updownarrow\rangle\,, \quad |1\rangle \mapsto |\leftrightarrow\rangle$.

Bob also knows about these coding systems, but does not know what $|R\rangle$ is. Suppose Bob has his own set of $n$ randomly generated qubits $|R'\rangle = |R'_1\rangle \otimes |R'_2\rangle \otimes \ldots \otimes |R'_n\rangle$, and $n$ qubits $|\chi'\rangle = |\chi'_1\rangle \otimes |\chi'_2\rangle \otimes \ldots \otimes |\chi'_n\rangle$ to store the information he is sent. Bob starts at $i = 1$, and each time he receives a photon, he adds 1 to $i$, so that his measurements and Alice's transmissions correspond. He chooses to measure in the polarisation of the photons either by measuring whether they are horizontally or vertically polarised if $|R'_i\rangle = |0\rangle$, or by measuring whether they are diagonally polarised at $45°$ or $-45°$ to the horizontal if $|R'_i\rangle = |1\rangle$. He then records his data as $|\chi'_i\rangle$ as follows;

(1) If $|R'_i\rangle = |0\rangle$, use $|\updownarrow\rangle \mapsto |0\rangle\,, \quad |\leftrightarrow\rangle \mapsto |1\rangle$
(2) If $|R'_i\rangle = |1\rangle$, use $\frac{1}{\sqrt{2}}|\updownarrow\rangle + \frac{1}{\sqrt{2}}|\leftrightarrow\rangle \mapsto |0\rangle\,, \quad \frac{1}{\sqrt{2}}|\updownarrow\rangle - \frac{1}{\sqrt{2}}|\leftrightarrow\rangle \mapsto |1\rangle$

We suppose that the number of photons Alice sends is a large multiple of four.

After the whole process of sending photons is complete, both Alice and Bob have collection of control qubits and data qubits. They share their control qubits over a classical channel, keeping the data for which $i$ they used the same coding system (i.e. $|R'_i\rangle = |R_i\rangle$). They should now have approximately half the data left.

Now suppose that Eve has managed to find a way to intercept the photons travelling from Alice to Bob. To gain any result she must measure the photons, and to ensure Bob does not notice her presence she must send on to Bob the states she measures. However, she must choose between the two coding systems that Bob and Alice are using, which half of the time will be different to the one that Alice is using to send the photons. Suppose that Bob and Alice are using the same coding system (i.e. $|R'_i\rangle = |R_i\rangle$), and Eve is using the other coding system, a photon from Alice will go through a $\pm 45°$ phase change as Eve sends it on. When Bob measures it with the same coding system Alice employs, he will get a $|0\rangle$ or $|1\rangle$ disagreeing with Alice's data with probability $1/2$ (i.e. $|\chi'_i\rangle \neq |\chi_i\rangle$ even though $|R'_i\rangle = |R_i\rangle$). Eve will choose a different coding system with probability $1/2$, and when this happens value Bob measures will disagree with the data Alice sent $1/2$ of the time. Thus we find that if there is any eavesdropping, approximately one quarter of the data Bob has will disagree with Alice's data.

Alice and Bob can thus compare half of the useful data (where they used the same coding system) to check whether anyone was eavesdropping. If there is any disagreement in this, then they know someone is eavesdropping and decide not to share the information. If there is no one eavesdropping, they use the other half of the useful data as a key with which to send information over the classical channel.

This idea has very good commercial prospects. There is already a company, Magiq Technologies (`http://www.magiqtech.com/`) which is producing communications security devices using ideas similar to the one we have just looked at.

## 9. Real Life Quantum Computers and A few Final Words

There have been many advances in the technology which can be used for quantum computing, however it still seems to be difficult to fit all the pieces together. Most of the trouble with constructing a successful quantum computer is minimising interaction with the outside world to avoid decoherence. Using quantum error correction and by taking account of decoherence, IBM and D-Wave Systems have managed to construct quantum computers.

9.1. **IBM's 7 Qubit Quantum Computer[29].** In a collaboration between Stanford University and IBM, a 7 qubit computer was successfully created in 2001. The computer used the nuclei of seven atoms in a molecule as its qubits. Each of the nuclei had a quantum property called spin, which the scientists "manipulated with ... nuclear magnetic resonance techniques". They demonstrated Shor's factoring algorithm in its most simple case - factoring 15 into 5 and 3. Most of the work however seems to have been done working with error correction codes, and they present a predictive model of decoherence effects.

Based on his research at IBM, David DiVencenzo set out the following requirements for a practical computer[30].

(1) It must be physically scalable to increase the number of qubits
(2) It must be possible to initialise qubits to arbitrary values
(3) Its processing must take place faster than the decoherence time
(4) It must have a set of gates, from which it is possible to build all other gates
(5) The qubits can be read easily.

9.2. **D-Wave Systems (www.dwavesys.com).** The latest announcement from D-Wave systems is the production of a "128 qubit" chip. They are most famous, however, for a demonstration in 2007 of a 16 qubit system. They we able to demonstrate database searching, a seating arrangement algorithm and finally a Sudoku puzzle. There has been much criticism however for scientists working the area of quantum computing. Many believe that D-Wave has not produced enough evidence, and there have been arguments that the information they have provided is inconclusive or erroneous[31].

9.3. **Final Words.** We have seen many other examples of quantum computation put to use or being researched. There is normally much media hype around announcements in quantum computing, just showing how important its study is. It is not too difficult an area to explore as we have shown in this report. We tackled quantum mechanics, first by taking a fresh approach to classical mechanics and discovered where our ideas about the quantum state space came from. Using the example of a Mach-Zehnder interferometer, we set up a space on which we could make measurements and progress through time using unitary operations. This gave us the mathematical tools to build a quantum computer. We looked at how we would construct its processor by building quantum circuits from quantum gates. From the results we found, we were able to conclude that our quantum computer could model a classical computer and do so much more! The next section was devoted to showing off our quantum computer. We showed that one of the most popular complex algorithms, the fast Fourier transform, could be completed on a quantum computer even faster. Finally, we found a way to break down the encryption that is used all over the world today. Then later, how to fix this problem with the idea of quantum key distribution.

There is no doubt that quantum computing has the ability to change the world and this has come about by changing our perceptions of the world. Even Einstein was confused by the idea that the world worked in a probabilistic way. Taking this to our advantage, which is what quantum computing has done, has lead us to create "impossible" algorithms and brought us the ability to transfer information in the knowledge that it is secure. As it was in the 1940s, we may be on the edge of a technological avalanche, or it may just come to be that it is physically infeasible to create a quantum computer. Either way, it is a very interesting subject to study.

## References


[1] G. Breyta C. S. Yannoni M. H. Sherwood I. L. Chuang L.M.K. Vandersypen, M. Steffen. Experimental realization of shor's quantum factoring algorithm using nuclear magnetic resonance. *Nature*, 414:883–887, 2001.

[2] Tetsuo Ohmi Mikio Nakahara. *Quantum Computing: From Linear Algebra to Physical Realizations*. CRC Press, 2008.

[3] M. Vitruvius Pollio. *De Architectura (Translation compiled by Bill Thayer, University of Chicago)*. http://penelope.uchicago.edu/Thayer/E/Roman/Texts/Vitruvius/.

[4] Paul A. Dirac. *The Principles of Quantum Mechanics*. Oxford University Press, 4 edition, 1989.

[5] James P. Wittke Robert H. Dicke. *Introduction to Quantum Mechanics*. Addison-Wesley Pub. Co., 1960.

[6] G. E. Moore. Cramming more components onto integrated circuits. *Electronics*, 38(8), April 1965.

[7] Intel press release: Moore's law 40th anniversary, http://www.intel.com/.

[8] Manek Dubash. Moore's law is dead, says gordon moore. *Techworld*, April 2005.

[9] P. Benioff. Quantum mechanical hamiltonian models of turing machines. *Journal of Statistical Physics*, 29(3), 1982.

[10] D. Deutsch. *Proceedings of the Royal Society of London*, 400:97–117, 1985.

[11] P. W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *Proceedings of the 35th Annual Symposium on Foundations of Computer Science, Santa Fe, NM*, 1994.

[12] Steven Roman. *Advanced Linear Algebra*. Springer, 3 edition, 2007.

[13] Wikipedia article on polarisation. *http://en.wikipedia.org/wiki/Polarization*.

[14] R. M. Tocknell K. P. Zetie, S. F. Adams. How does a mach-zehnder interferometer work? *Physics Education*, 35(1):46–48, 2000.

[15] J. Clerk Maxwell. A dynamical theory of the electromagnetic field. *Philosophical Transactions of the Royal Society of London*, 155:459–512, 1865.

[16] Mika Hirvensalo. *Quantum Computing*. Springer, 2 edition, 2004.

[17] Shahram Mohammednejad Shamsolah Salemian. Quantum hadamard gate implementation using planar lightwave circuit and photonic crystal structures. *American Journal of Applied Sciences*, 5(9):1144–1148, 2008.

[18] N. Rosen A. Einstein, B. Podolsky. Can quantum-mechanical description of physical reality be considered complete? *Physical Review*, 47(10):777–780, May 1935.

[19] http://www.scientificblogging.com/news_releases/entangled_photon_pairs_experiment_tests_spooky_action_at_a_distance/. Entangled photon pairs experiment tests 'spooky action at a distance'.

[20] S. Haroche J. M. Raimond, M. Brune. Colloquium: Manipulating quantum entanglement with atoms and photons in a cavity. *Reviews of Modern Physics*, 73:565–582, 2001.

[21] Christoph Simon and William T.M. Irvine. Robust long-distance entanglement and a loophole-free bell test with ions and photons. *Physical Review Letters*, 91:110–405, 2003.

[22] Tereza Tusarova. Quantum complexity classes. Master's thesis, http://arxiv.org/abs/cs/0409051v1, 2004.

[23] Alan M. Turing. On computable numbers, with an application to the entscheidungsproblem. *Proceedings of the London Mathematical Society*, 2(42):230–265, 1937.

[24] Mi Lu. *Arithmetic and Logic in Computer Systems*. John Wiley and Sons, 2004.

[25] Emil Leon Post. *The two-valued iterative systems of mathematical logic*. Princeton University Press, 1941.

[26] Richard Cleve David P. DiVincenzo Norman Margolus Peter Shor Tycho Sleator John Smolin Harald Weinfurter Adriano Barenco, Charles H. Bennett. Elementary gates for quantum computation. *Physical Review A*, 52(5):34–57, March 1995.

[27] Thomas W. Körner. *Fourier Analysis*. Cambridge University Press, 1989.

[28] Wojciech H. Zurek William K. Wootters. A single quantum cannot be cloned. *Nature*, 299:802–803, 1982.

[29] J. S. Bell. On the einstein podolsky rosen paradox. *Physics*, 1:195–200, 1964.

[30] D. P. DiVincenzo. The physical implementation of quantum computation. 2005.

[31] MIT Scott Aaronson. http://scottaaronson.com/blog/.