Unentangling Quantum Algorithms for Mathematicians and Engineers

by

Jaden K. Pieper

A thesis submitted to the

Faculty of the Graduate School of the

University of Colorado in partial fulfillment

of the requirements for the degree of

Master of Science

Department of Applied Mathematics

2017

This thesis entitled:
Unentangling Quantum Algorithms for Mathematicians and Engineers
written by Jaden K. Pieper
has been approved for the Department of Applied Mathematics

Manuel E. Lladser

Prof. James Meiss

Prof. Stephen Becker

Prof. Harvey Segur

Date

The final copy of this thesis has been examined by the signatories, and we find that both the content and the form meet acceptable presentation standards of scholarly work in the above mentioned discipline.

Pieper, Jaden K. (M.S., Applied Mathematics)

Unentangling Quantum Algorithms for Mathematicians and Engineers

Thesis directed by Prof. Manuel E. Lladser

As industry continues to inspire considerable growth in the research and development of quantum computers, it is increasingly worthwhile to familiarize oneself with the computational theory of this new and exciting field.

In this manuscript, we introduce readers to quantum computing by first developing a quantum intuition through the enlightening results of the so-called Stern-Gerlach experiment. After getting a feeling for quantum physics concepts such as superposition and measurement, and the need of linear algebra and probability to describe quantum phenomena, we move to a discussion of quantum computing. We develop a "quantum toolbox," the mathematical tools used to explore quantum computing, before walking through two quantum algorithms, Deutsch's and Grover's algorithms. We finally explore a real quantum computer that is available to the public and show readers how to implement these quantum algorithms on it. We also consider the effect of error due to quantum decoherence, and the limitations such errors pose on quantum algorithms.

Acknowledgements

First, I would like to thank Prof. Manuel Lladser for his advisement over the past year. Prof. Lladser was consistently a great resource and helped keep me going in the right direction. His excitement and encouragement for the project always inspired me and made working with him a pleasure. I am very appreciative to have had the opportunity to learn from him. Thank you!

I would also like to thank the entire Applied Mathematics Department. I have had many amazing professors who made my time at this university a true pleasure. In particular, I would like to thank my thesis committee members, Prof. Stephen Becker, Prof. James Meiss, and Prof. Harvey Segur.

I was also very fortunate to receive partial funding from the EXTREEMS-QED NSF grant #1407340, while I worked on this thesis.

Finally, I want to thank all my friends and family who supported me during this endeavor. Your many words of encouragement were essential to my completing this project.

Contents

Chapter

Tables

Table

Figures

Figure

Chapter 1

Introduction

During a lecture in the early 1980s, Richard Feynman proposed the concept of simulating physics with a quantum computer [12]. He postulated that by manipulating the properties of quantum mechanics and quantum particles one could develop an entirely new kind of computer, one that could not be described by the current theory of computation with Turing machines. Nature does not explicitly perform the calculations to determine the speed of a ball dropped from a tall building, it does so implicitly. Extending this line of thinking, Feynman wondered if we could harness the complex calculations nature performs intrinsically in quantum mechanics to design a computer with more computational power.

Soon after Feynman's lecture, new results about the advantages of quantum computers began to filter in. By 1985, Deutsch had developed the concept of the Quantum Turing machine, which formalized the theory of quantum computation [9], and introduced the first problem that a quantum computer could solve faster than any known classical algorithm. By 1992 Deutsch and Josza proposed a generalized algorithm for this problem [11], that further demonstrated the potential speed increases quantum computers could provide. Soon algorithms were being developed that provided speed up for searching [14] and integer factorization. With the exponential speed up in integer factorization shown by Shor [21], and its implications to cryptography and security, the theory of quantum computation had shown its true potential. Research into the development of physical quantum computers began in earnest.

Recently there have been many advancements in physical implementations of the Deutsch-

Josza algorithm [18, 15], quantum search [4], quantum integer factorization [23, 20], quantum Fourier Transform [5], and programmable quantum computers [8, 17]. With IBM's Quantum Experience, a programmable 5-qubit quantum computer is now available to the public [1].

The field of quantum computing is developing rapidly. Many resources for learning how and why it works are readily available. However, without a background in quantum mechanics, some of its concepts are difficult to grasp and one's ability to participate in this exciting new field is limited.

In this manuscript, we will develop for readers a quantum intuition by focusing on the illuminating Stern-Gerlach experiment. Through this experiment we will derive notions of quantum systems, states, operators, superposition, and measurement, all of which are key to quantum computing. We will then transfer these concepts to the theory of quantum computation and perform a brief study of the ground breaking algorithms developed by Deutsch and Grover. Finally, using IBM's 5-qubit computer, we will implement these algorithms and discuss current issues and limitations in physical quantum computers.

Chapter 2

A Quantum Intuition: The Stern-Gerlach Experiment

The Stern-Gerlach experiment demonstrates the quantization of spin, although the notion of spin had not yet been conceived when the experiment was originally conducted in 1922 [13]. It is a particularly instructive experiment to analyze as the fundamental concepts of quantum states, probabilities associated with measurement, and linear operators, all exhibit themselves in this relatively simple yet puzzling experiment. We follow a similar approach to McIntyre's [19] as we walk through this experiment and familiarize readers with examples of quantum phenomenon.

2.1 Quantization of Spin

The general set up for the Stern-Gerlach experiment is as follows: a beam of particles is shot from an oven through an "up" magnetic field (i.e. a magnetic field with a gradient pointing in the positive direction on the z-axis), and then onto a detector screen. The particles have a magnetic moment, μ, which causes them to interact with the magnetic field and be deflected, by an amount proportional to the z-component of μ. As the particles originate in the thermal environment of the oven, classical physics predicts that the z-component of μ take values over a continuous spectrum, so that the particles are distributed uniformly when observed on the detector screen.

However, when using silver atoms as particles what is observed defies the predictions of classical physics: upon exiting the magnetic field the particles split perfectly along two distinct paths rather than a continuum of them (see Figure 2.1). Physicists attribute this deflection to an interaction between the magnetic field and a sort of *intrinsic angular momentum* or *spin* of

the silver atoms—a three dimensional vector, usually denoted as S. The magnetic field is then interpreted as *measuring* the z-component of the spin, a quantity usually denoted as S_z.

The fact that the particles travel along one of two distinct paths implies that S_z is discrete, or that *spin is quantized*. Experimentally, it is found that the two values that the z-component of spin can take are always $S_z = \pm\hbar/2$, where $\hbar$ is called the reduced Planck constant. We further state that particles deflected up are *spin up particles*, and those deflected down are *spin down*. We also describe the state of the particles as they leave the oven generally as $|\psi\rangle$, which reads "ket psi". We denote spin up particles as $|+\rangle$ and spin down as $|-\rangle$.

We note that the factor of $1/2$ for the two possible values of S_z motivates the jargon of a *spin-1/2 system*, which is a specific instance of a *two-state quantum system*. Further experiments reveal that *spin is quantized in any direction*. Namely if we perform the same experiment with the magnetic field oriented about the x or y-axes the same results are observed, and we are now measuring the x or y-components of spin, S_x, S_y, respectively. We will call an experiment that measures S_z, S_x, or S_y a Z, X, or Y analyzer respectively.

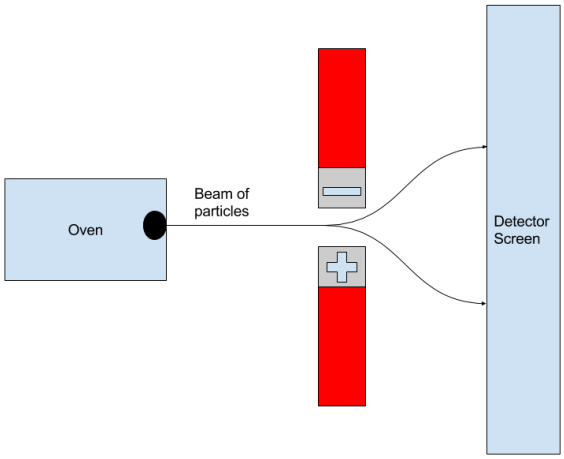

Figure 2.1: *Stern-Gerlach Experiment.* Particles from a thermal oven are shot through a magnetic field with its gradient in the $+z$-direction. The particles then hit a detector screen, where it can be seen the particles beam splits into two distinct paths after passing through the magnetic field. This experiment demonstrates the quantization of spin.

2.2 Expected Behavior of Spin

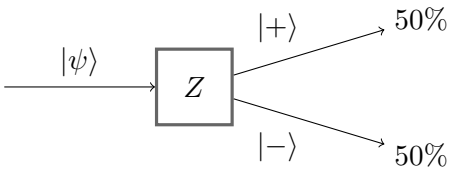

Figure 2.2: *Discrete Properties of Spin.* This diagram is equivalent to Figure 2.1. The box represents a Z analyzer i.e. a magnetic field with its gradient in the $+z$ direction. Spin up and down particles are denoted as $|+\rangle$ and $|-\rangle$, respectively.

We introduce the notation of Figure 2.2 to equivalently describe the experiment of Figure 2.1. With this new notation we will explore more advanced versions of the Stern-Gerlach experiment to understand the properties of spin. As such all figures in this section detail actual experimental results, which we will use to develop the mathematical formulation of quantum mechanics. We have already seen that spin is described in some sense by probability, as the general quantum states of a particle emerging from the oven, $|\psi\rangle$, are equally likely to be measured in state $|+\rangle$ or $|-\rangle$. If we now consider the experiment of Figure 2.3 we see that we can in fact make accurate predictions about the state of a particle's spin. Namely if we make a measurement of spin, and immediately repeat the same measurement we always get the same result. This is a natural and intuitive property, but one that future experiments will show should not be taken for granted.

We now consider an experiment where we send particles through a Z analyzer, and send all particles observed in state $|+\rangle$ through an X analyzer, seen in Figure 2.4. We find that knowing the particles spin about the Z axis provides no information about the particles spin about the X axis. It is in fact arbitrary that we select Z and X for this experiment, and the results would be the same for any two different analyzers, X, Y, or Z. Thus we say that measurements on a particles spin are uncorrelated about any orthogonal axis. We further develop the notation $|+\rangle_x$ to describe the state of particles deflected "up" by an X analyzer and $|-\rangle_x$ for those deflected "down". This notation can be extended to describe the states observed from an experiment with a Y analyzer

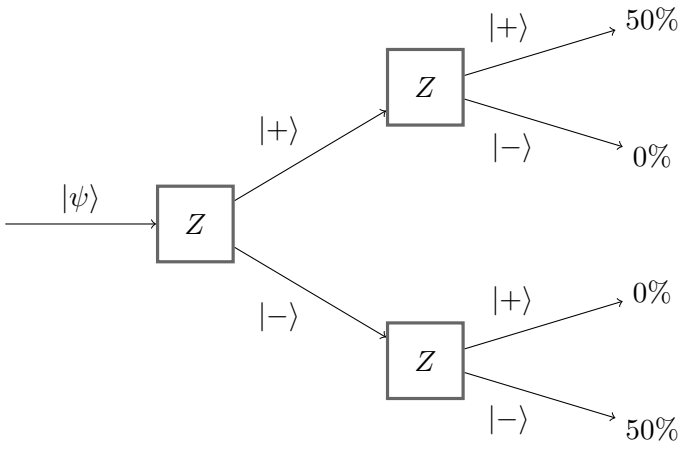

Figure 2.3: *Preservation of Spin State Knowledge.* In this experiment particles are shot through a Z analyzer, and all particles observed in state $|+\rangle$ are sent through a second Z analyzer. It is found that the knowledge we gain from the first analyzer is maintained, and all particles are again observed to be in state $|+\rangle$.

with $|+\rangle_y$ and $|-\rangle_y$.

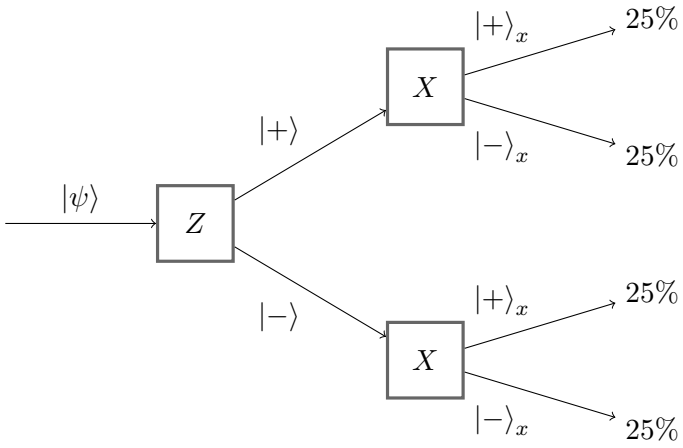

Figure 2.4: *Uncorrelated Measurements.* In this experiment particles are first shot through a Z analyzer, and all particles observed in state $|+\rangle$ are then sent through an X analyzer. Half these particles are observed to be in state $|+\rangle_x$ and half in state $|-\rangle_x$. Similarly when particles that were observed in state $|-\rangle$ are sent through an X analyzer they are equally likely to be measured as $|+\rangle_x$ or $|-\rangle_x$.

2.3 Abnormal Behavior of Spin

Before we discuss some atypical behavior of spin, consider the following thought experiment. We are given a large collection of unsorted coins, that can either be red or blue in color and round or square in shape. However the coins were made so small that we cannot visually tell the different types of coins apart! Luckily we also have two machines that separate coins based on color and shape respectively. Now consider sending our collection through a color machine and taking all the coins that are blue and sending them through a shape machine. If we isolate all the round coins that come out of this machine, our intuitive notion of reality dictates that we have successfully isolated a group of blue, round coins. However we find that when we send this group through another color machine we somehow count half the coins to be red! This seems to imply that somehow when we learned the shape of the observed coins we somehow lost information about their color, even though we had already measured it. This experiment seems to defy our entire logical notion of how objects behave and interact when we perform measurements on them, and suggests that these coins do not behave according to the same rules as objects we deal with physically. It is clear that no macroscopic object we can interact with behaves like these coins, but the particles of the Stern-Gerlach experiment do.

In particular if we extend our previous experiment, as seen in Figure 2.5, by adding one more Z analyzer we find our measurements of a particles spin about the z and x-axes behave exactly as described in the coin experiment above. Somehow when we measure the quantity S_x, we lose all the knowledge we previously had about the quantity S_z, which implies that *measurement is directly affecting quantum states*. In fact it turns out that particles in the states $|+\rangle_x, |-\rangle_x$ are in uniform *superpositions* of the states $|+\rangle$ and $|-\rangle$. What this means is that in some sense these particles are simultaneously in both states at once and only when measured with another Z analyzer do they collapse with equal probability to either $|+\rangle$ or $|-\rangle$. And as the same experimental results hold if we switch X and Z analyzers, $|+\rangle, |-\rangle$ are uniform superpositions of $|+\rangle_x$ and $|-\rangle_x$. Returning to the coin example, this would be as if knowing we had a red or blue coin meant the coin was

simultaneously a square coin and a round coin. Similarly we can also say that knowing our coin is square or round means that is in a superposition of being blue and red.

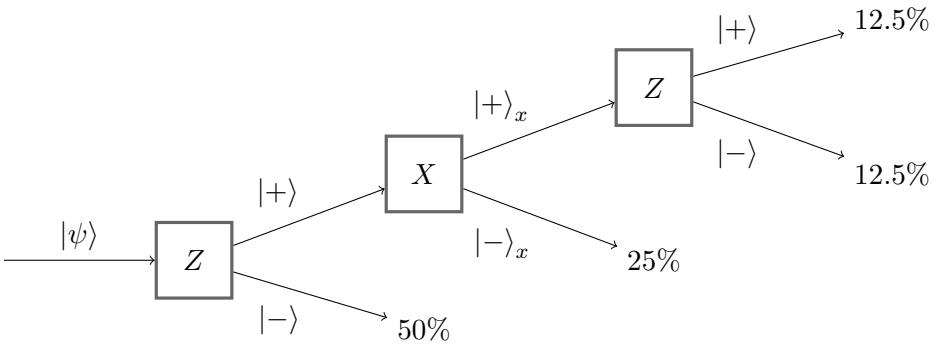

Figure 2.5: *Loss of Spin State Knowledge.* In this experiment particles in state $|+\rangle$ are sent through an X analyzer, and the particles measured in state $|+\rangle_x$ are then shot through another Z analyzer. As we have previous knowledge of z-spin, we expect that knowledge to be maintained regardless of our measurement of x-spin. However we find that the measurement of x-spin changed what we already knew about z-spin, and that particles that we had previously observed to be in state $|+\rangle$ could now be in state $|+\rangle$ or $|-\rangle$.

We can begin to understand the mechanics of how measurement can affect our knowledge of a quantum state by briefly returning our attention to the experiment of Figure 2.3. This experiment first introduces some notion of orthogonality between measurement states, in that if you have state $|+\rangle$, there is zero probability of measuring the particle to have negative spin, associated with $|-\rangle$. We can then notice we are performing a measurement on a state $|+\rangle$, and in this special case our measurement returns the exact same state $|+\rangle$, and the particle has been deflected upward, associated with some constant value of spin, $S_z = \frac{\hbar}{2}$. This is reminiscent of the relationship between operators and their eigenvectors, often called eigenstates in this context, and suggests linear algebra may be at play in this experiment. In particular it hints that measurements are associated with operators. It is important to realize that given any quantum state $|\psi\rangle$, a measurement with a Z analyzer yields particles in either state $|+\rangle$ or $|-\rangle$, deflected by an amount associated with $S_z = \pm\frac{\hbar}{2}$. This indicates that a measurement with a Z analyzer does not represent an operator acting on a state, but rather that the measurement projects the state onto the eigenvectors of its

associated operator, and that the measured value corresponds to the eigenvalue of the observed eigenvector. Thus we can say that $|+\rangle$ and $|-\rangle$ are the eigenvectors of the operator associated with a measurement of z-spin, and that $\pm\frac{\hbar}{2}$ are its eigenvalues.

When we make a measurement of the x-component of spin, S_x, in Figure 2.5, we are not preserving any prior knowledge of the particles state, we are projecting the particle to be in an entirely new state. This projection throws away all our prior knowledge about z-spin, and explains how measurement affects quantum states. These concepts will be further developed in the next section, as we derive the operators associated with measurement and their associated eigenvectors.

2.4 Spins Program for further experiments

As we move towards a more in-depth exploration of the Stern-Gerlach experiments, it useful to know that Open Source Physics (OSP) has a SPINS program that can simulate the Stern-Gerlach experiment for spin 1/2 systems. It is highly recommended to download their java app to simulate the different experiments explored here. The program can be found in reference [6].

Chapter 3

Stern-Gerlach Experiment: A Closer Look

In this section we use the Stern-Gerlach experiment to formalize key concepts of quantum computing, including the notions of states, operators and measurement. We begin by presenting a series of postulates that we will carry with us through this section as we explore these experiments with a bit more depth. We first postulate that the quantum states of the Stern-Gerlach experiment can be described by a ket, which in turn represents a two-dimensional complex vector. We can then associate the measurements performed by Stern-Gerlach analyzers with operators that act on these kets, where measurement projects the quantum state onto an eigenvector of its associated operator. Finally, the elements of a vector written in the eigen-basis associated with a measurement operator, relate to the probability of that ket being projected onto either basis state under measurement.

3.1 States of a Quantum System and Quantum Measurement

The Stern-Gerlach experiment is fundamentally an experiment of measurement. All the experiments we study involve the relationship in probability between an input state and output state for a measurement. By studying the experiments in detail and assigning some structure to quantum states we can develop a mathematical formalism that perfectly describes the observations from experiments. Further this formulation will allow us to make predictions of future experiments and later extend directly into quantum computing. We begin by defining a very generalized structure for a quantum state in a ket, and use that structure to develop a definition of measurement. With the definition of measurement in hand we go on to derive the quantum states of the Stern-Gerlach

expirement directly from their results.

3.1.1 Formalizing Quantum States and Measurement

In quantum mechanics a ket describes the state of a quantum system and all the information that can be known about it. A ket is generally an element of a Hilbert space but, in the case of quantum computing, it is sufficient to describe it as a finite dimensional column vector with complex entries. The main implication of kets being elements of a Hilbert space is the existence of an inner product. The inner product between two quantum states $|\psi\rangle$ and $|\varphi\rangle$ is denoted by the braket $\langle\varphi|\psi\rangle$, where $\langle\varphi|$ reads "bra phi". In the context of the Stern-Gerlach experiment, kets describe the state of a particle's spin, and are represented as a two-dimensional complex vector, $|\psi\rangle = (\alpha_0, \alpha_1)$, where $(,)$ represents a column vector. A bra is thus a row vector, and is defined to be the conjugate transpose of its associated ket.

Back in the setup of the experiment in Figure 2.2, a fundamental question is: given we send some state $|\psi\rangle$ through the Z analyzer, *what is the probability that we measure* $|+\rangle$ *and* $|-\rangle$ *as outputs, respectively?* We denote the probability of the state $|\psi\rangle$ being measured in state $|+\rangle$ as $P(|\psi\rangle, |+\rangle)$. Quantum Physics postulates that these probabilities are given by

$$\begin{aligned} P(|\psi\rangle, |+\rangle) &= |\langle +|\psi\rangle|^2 \\ P(|\psi\rangle, |-\rangle) &= |\langle -|\psi\rangle|^2 \end{aligned}$$

More generally, given an experiment where the observed states after measurement are $|\varphi_0\rangle$ and $|\varphi_1\rangle$, the probability of measuring state $|\varphi_i\rangle$ as the output is

$$P(|\psi\rangle, |\varphi_i\rangle) = |\langle \varphi_i|\psi\rangle|^2 \tag{3.1}$$

Here it is important to recognize some subtleties of our notation. In particular when we discuss the probability of a state, $|\psi\rangle$, being in state $|\varphi_0\rangle$ under measurement, it is implied that we are sending a particle in state $|\psi\rangle$ through an analyzer that yields states in either state $|\varphi_0\rangle$ or $|\varphi_1\rangle$. For instance the notation $P(|\psi\rangle, |+\rangle_x) = |{}_x\langle +|\psi\rangle|^2$ implies that we are making a measurement using an X analyzer.

Immediately it is apparent the definition of equation (3.1) guarantees our probabilities are real numbers, a necessary condition. It also provides a further constraint on our quantum states, in that they must be normalized. To see this observe the experiment of Figure 2.3 and note that the probability of measuring a state to be in its current state is 1. Formally

$$P(|\psi\rangle, |\psi\rangle) = |\langle\psi|\psi\rangle|^2 = 1 \tag{3.2}$$

With the above constraint we can now examine the specific states observed in the Stern-Gerlach experiment and use the probability of measurements to define a concrete mathematical structure for quantum states.

3.1.2 Deriving the z States

We now consider the fundamental question *How can we describe the states encountered in the Stern-Gerlach experiment?* Let us first explore the states $|+\rangle$ and $|-\rangle$.

We need to be sure the mathematical description of $|+\rangle$ and $|-\rangle$ match observations from experiments. Looking at the experiment in Figure 2.3, where we send the particles deflected up from a Z-analyzer in state $|+\rangle$ through a second Z-analyzer, we find that all particles are measured to be in state $|+\rangle$ by the second analyzer. Recalling $P(|\psi\rangle, |\varphi\rangle)$ denotes the probability of a state $|\psi\rangle$ being measured in state $|\varphi\rangle$ then we see

$$\begin{aligned} P(|+\rangle, |+\rangle) &= |\langle+|+\rangle|^2 = 1 \\ P(|+\rangle, |-\rangle) &= |\langle-|+\rangle|^2 = 0 \end{aligned} \tag{3.3}$$

It was in fact arbitrary that we choose to send particles in the state $|+\rangle$ through the second Z analyzer, and we would get similar results if we instead selected particles in the state $|-\rangle$. In particular

$$\begin{aligned} P(|-\rangle, |+\rangle) &= |\langle+|-\rangle|^2 = 0 \\ P(|-\rangle, |-\rangle) &= |\langle-|-\rangle|^2 = 1 \end{aligned} \tag{3.4}$$

We thus can see some notion of orthogonality between the two observable states in this measurement, as there is zero probability of a state $|+\rangle$ being measured in state $|-\rangle$.

So we have some simple restrictions on the states $|+\rangle$ and $|-\rangle$: they must be unit and orthogonal vectors. It is conventional to describe quantum states with respect to the z basis in Stern-Gerlach, so we can define $|+\rangle$ and $|-\rangle$ as

$$|+\rangle = \begin{bmatrix} 1 \\ \\ 0 \end{bmatrix}, \quad |-\rangle = \begin{bmatrix} 0 \\ \\ 1 \end{bmatrix} \tag{3.5}$$

It is easy to see all the restrictions imposed by the experimental observations, i.e. the identities in equations (3.3) and (3.4) are satisfied by the above states. Moreover, a general state $|\psi\rangle$ can be represented as

$$|\psi\rangle = \begin{bmatrix} \alpha_0 \\ \\ \alpha_1 \end{bmatrix} = \alpha_0 |+\rangle + \alpha_1 |-\rangle \tag{3.6}$$

With these new definitions in tow we can in fact find relationships between all the other states observed in Stern-Gerlach and these z states and relate them to their vector forms.

3.1.3 Deriving the x and y States

We can now use results directly from experiment to derive the states associated with spin about the x-axis, $|+\rangle_x$ and $|-\rangle_x$. We first will consider the experiment from Figure 2.4 to find the relations between the z basis states, $|+\rangle$ and $|-\rangle$, to the x states $|+\rangle_x$ and $|-\rangle_x$.

$$\begin{aligned} P(|+\rangle, |+\rangle_x) &= |\,{}_x\langle +|+\rangle\,|^2 = 1/2 \\ P(|+\rangle, |-\rangle_x) &= |\,{}_x\langle -|+\rangle\,|^2 = 1/2 \\ P(|-\rangle, |+\rangle_x) &= |\,{}_x\langle +|-\rangle\,|^2 = 1/2 \\ P(|-\rangle, |-\rangle_x) &= |\,{}_x\langle -|-\rangle\,|^2 = 1/2 \end{aligned} \tag{3.7}$$

To derive the coordinates of $|+\rangle_x$ and $|-\rangle_x$, we write

$$\begin{aligned} |+\rangle_x &= a_0 |+\rangle + a_1 |-\rangle \\ |-\rangle_x &= b_0 |+\rangle + b_1 |-\rangle \end{aligned} \tag{3.8}$$

for complex numbers a_0, a_1, b_0 and b_1 to be determined. Since, according to the experiment, the probability a particle that is state up about the z-axis is measured to be spin up about the x-axis is $1/2$, we find that

$$\begin{aligned} P(|+\rangle, |+_x\rangle) &= |\,{}_x\langle +|+\rangle\,|^2 \\ &= |(a_0^* \langle +| + a_1^* \langle -|)\,|+\rangle\,|^2 \\ &= |a_0^* \langle +|+\rangle + a_1^* \langle -|+\rangle\,|^2 \\ &= |a_0|^2 \\ &= \frac{1}{2} \end{aligned} \tag{3.9}$$

We can follow the same procedure for the other relations given by equation (3.7) to find that

$$|a_0|^2 = |a_1|^2 = |b_0|^2 = |b_1|^2 = \frac{1}{2} \tag{3.10}$$

As such we must define

$$\begin{aligned} |+\rangle_x &= \frac{e^{ia}\,|+\rangle + e^{ib}\,|-\rangle}{\sqrt{2}} \\ |-\rangle_x &= \frac{e^{ic}\,|+\rangle + e^{id}\,|-\rangle}{\sqrt{2}} \end{aligned} \tag{3.11}$$

for suitable real constants a, b, c and d to be determined.

We will now reconsider the experiment from Figure 2.3 but with X instead of Z analyzers. As expected this experiment demonstrates a notion of orthonormality between the states $|+\rangle_x$ and $|-\rangle_x$, i.e.

$$\begin{aligned} P(|+\rangle_x, |+\rangle_x) &= |\,{}_x\langle +|+\rangle_x\,|^2 = 1 \\ P(|-\rangle_x, |-\rangle_x) &= |\,{}_x\langle -|-\rangle_x\,|^2 = 1 \\ P(|+\rangle_x, |-\rangle_x) &= |\,{}_x\langle -|+\rangle_x\,|^2 = 0 \\ P(|-\rangle_x, |+\rangle_x) &= |\,{}_x\langle +|-\rangle_x\,|^2 = 0 \end{aligned} \tag{3.12}$$

We can easily verify that the states derived in equation (3.11) satisfy the first two constraints. For

instance:

$$\begin{aligned} {}_x\langle +|+\rangle_x &= \frac{(e^{-ia}\langle +| + e^{-ib}\langle -|)(e^{ia}|+\rangle + e^{ib}|-\rangle)}{2} \\ &= \frac{\langle +|+\rangle + \langle -|-\rangle}{2} \\ &= 1 \end{aligned} \tag{3.13}$$

Before addressing the orthogonality constraints in equation (3.12), it is important to note the following. If λ is an arbitrary real number and $|\psi\rangle$ an arbitrary state then $P(|\psi\rangle, e^{i\lambda}|\psi\rangle) = 1$; in particular and as expected, the rotation of a state does not affect its magnitude. In quantum physics terms, this means that states $|\psi\rangle$ and $e^{i\lambda}|\psi\rangle$ are *equivalent upon measurement.* Consequently, we may as well define

$$\begin{aligned} |+\rangle_x &= \frac{|+\rangle + e^{i\varphi}|-\rangle}{\sqrt{2}} \\ |-\rangle_x &= \frac{|+\rangle + e^{i\theta}|-\rangle}{\sqrt{2}} \end{aligned} \tag{3.14}$$

We are getting very close to having explicit expressions for $|+\rangle_x$ and $|-\rangle_x$. We will now examine the orthogonality between the two states for one last constraint. If we consider

$$\begin{aligned} {}_x\langle -|+\rangle_x &= (\langle +| + e^{-i\theta}\langle -|)(|+\rangle + e^{i\varphi}|-\rangle) \\ &= \langle +|+\rangle + e^{i(\varphi-\theta)}\langle -|-\rangle \\ &= 1 + e^{i(\varphi-\theta)} \end{aligned} \tag{3.15}$$

then we see that $|+\rangle_x$ and $|-\rangle_x$ are orthogonal if and only if $(\varphi - \theta)$ is an odd multiple of π. In particular, since rotations do not affect measurement, it is customary to select $\varphi = 0$ and $\theta = -\pi$. Thus, we define the x basis states as

$$\begin{aligned} |+\rangle_x &= \frac{|+\rangle + |-\rangle}{\sqrt{2}} = \frac{1}{\sqrt{2}}\begin{bmatrix} +1 \\ +1 \end{bmatrix} \\ |-\rangle_x &= \frac{|+\rangle - |-\rangle}{\sqrt{2}} = \frac{1}{\sqrt{2}}\begin{bmatrix} +1 \\ -1 \end{bmatrix} \end{aligned} \tag{3.16}$$

One can follow very similar steps as above to find the y states, $|+\rangle_y$ and $|-\rangle_y$ such that

$$\begin{aligned} |+\rangle_y &= \frac{|+\rangle + i\,|-\rangle}{\sqrt{2}} = \frac{1}{\sqrt{2}} \begin{bmatrix} +1 \\ +i \end{bmatrix} \\ |-\rangle_y &= \frac{|+\rangle - i\,|-\rangle}{\sqrt{2}} = \frac{1}{\sqrt{2}} \begin{bmatrix} +1 \\ -i \end{bmatrix} \end{aligned} \tag{3.17}$$

The identities in equations (3.16) and (3.17) now represent the x and y states observed in the Stern-Gerlach experiment, written in terms of the z states.

3.2 Observables in Quantum Systems: Operators and Eigenvalues

Recall again the Stern-Gerlach Experiments in Figures 2.2 and 2.3. These experiments show us that measuring the z-spin of a particle in any general state, $|\psi\rangle$, yields particles that are then in state $|+\rangle$ or $|-\rangle$, deflected according to a spin component $S_z = \pm\frac{\hbar}{2}$ respectively. They further show that measuring the z-spin of a particle in state $|+\rangle$ yields only particles in state $|+\rangle$, deflected according to $S_z = \frac{\hbar}{2}$. In other words, when one sends the input state $|+\rangle$ through the Z-analyzer, one recovers the same state and measures the z-component of the spin, as if the Z-analyzer acted as a linear operator having $|+\rangle$ as an eigenvector associated with the eigenvalue $\hbar/2$. (Similar considerations apply to the state $|-\rangle$ and its associated measurement $-\hbar/2$.)

Quantum physics postulates that measurements of physical observables, such as the z-component of spin, are eigenvalues of related linear operators. Fundamentally, a measurement is a projection of a state onto the eigenvectors associated with that measurement. For instance we can define some operator, $\hat{Z}$, associated with a measurement using a Z analyzer, where the output states $|+\rangle, |-\rangle$ are the eigenvectors of $\hat{Z}$, and the values $\pm\frac{\hbar}{2}$ are their associated eigenvalues. Thus any state $|\psi\rangle$ that is passed through a Z analyzer will be projected onto either $|+\rangle$ or $|-\rangle$ and we will make a measurement of z-spin such that $S_z = \pm\frac{\hbar}{2}$.

Since the probabilistic interpretation of quantum states forces kets to be normalized, quantum operators must preserve norm. Unitary matrices satisfy this condition and it turns out *all*

quantum operators are unitary. As we move into a discussion where quantum operators represent gates, similar to logic gates in classical computing, we now know that we are guaranteed to have reversibility of a quantum gate. This turns out to be a very important feature of quantum computing that we will focus on in later sections.

3.3 Deriving the Unnormalized Pauli Matrices

We have postulated that measurements are associated with operators, and that upon measurement the state of a particle collapses to an eigenstate of the associated measurement operator. Using the states derived in Sections 3.1.2 and 3.1.3, which represent the output states of x, y, and z measurements, we can construct these operators.

We know that measurements made with a Z analyzer yield the eigenstates $|+\rangle$ and $|-\rangle$ associated with eigenvalues $S_z = \pm\frac{\hbar}{2}$ respectively. Thus we can define some operator, $\hat{Z}$, that satisfies

$$\begin{aligned} \hat{Z} : |+\rangle &\rightarrow \frac{\hbar}{2} |+\rangle \\ |-\rangle &\rightarrow -\frac{\hbar}{2} |-\rangle \end{aligned} \tag{3.18}$$

In particular, since $|+\rangle$ and $|-\rangle$ form a standard basis for $\mathbb{C}^2$, a straightforward calculation shows that the matrix representation of $\hat{Z}$ is

$$\hat{Z} = \frac{\hbar}{2} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \tag{3.19}$$

Similarly, one finds that x and y measurements can be described by the operators

$$\begin{aligned} \hat{X} &= \frac{\hbar}{2} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \\ \hat{Y} &= \frac{\hbar}{2} \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \end{aligned} \tag{3.20}$$

Using results strictly from the experiments we have developed the mathematical formulation of quantum mechanics. We have well-defined quantum states seen in the experiment as well as

operators associated with different measurements. In fact the operators we have derived belong to an important subset of quantum operators called the Pauli operators, which will play a fundamental role in quantum computing. With these definitions in hand, we are ready to make the transition from the Stern-Gerlach experiments to quantum computing.

3.4 From Stern-Gerlach to Quantum Computing

As we continue our discussion of quantum computing, we will leave the Stern-Gerlach experiment behind. The quantum states and operators we described in this experiment are still valid in the context of a generalized quantum computing theory, however, there are a few changes that need to be addressed.

The first change is an update to our vocabulary. In what follows, when we describe a quantum state, we are equivalently describing the state of a qubit. However, the latter does not necessarily have a physical interpretation, such as the spin of a particle. Because of this, we no longer describe the state of a qubit as $|+\rangle$ or $|-\rangle$, or a linear combination of these. Rather, our new notation will be directly linked with the notions of bits from classical computing. In particular, we define:

$$|0\rangle \equiv |+\rangle \equiv \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad |1\rangle \equiv |-\rangle \equiv \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{3.21}$$

We refer from now on to $|0\rangle$ and $|1\rangle$ as our *computational basis*.

We will describe operators as quantum gates from now on. Quantum gates are to quantum computing like logic gates (i.e. conjunction, disjunction, negation, etc) are to classical computing, and are the building blocks of quantum algorithms. They allow us to manipulate quantum states over time by acting on them in predefined ways. In particular, from now on the Pauli matrices (which we derived from the Stern-Gerlach experiment as measurement operators) will be the normalized operators

$$I \equiv \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad X \equiv \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad Y \equiv \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \quad Z \equiv \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \tag{3.22}$$

We originally derived the Pauli Gates X, Y, Z as measurement operators. In particular the operator X is associated with a measurement of the x-component of a particle's spin. We achieve such a measurement by placing a particle in a magnetic field with its gradient pointing in the x direction.

We now pose the question: *what is the role of the X operator in quantum computing?* Consider a state $|\psi\rangle = a\,|0\rangle + b\,|1\rangle$. After applying the X operator we get $X\,|\psi\rangle = b\,|0\rangle + a\,|1\rangle$; in particular, $X\,|0\rangle = |1\rangle$ and $X\,|1\rangle = |0\rangle$. In other words, the normalized X operator flips the amplitudes associated with our basis states. This is exactly analogous to the logical NOT gate (also called a bit flip) from classical computing. However, whereas the classical NOT gate flips the state of a bit (sending 0 to 1 and vice versa), the quantum NOT gate flips the amplitudes.

The next natural question is: *how can one relate the notion of X as a measurement operator to this NOT operation?* Consider placing a particle in an x-magnetic field, but not making a measurement on the particle's state. Namely one does not observe the direction the particle is deflected, i.e. does not collapse its state to one of the eigenstates of X. Then we can use Schrodinger's equation [19] to describe the state of this particle as a function of time:

$$i\hbar \frac{d}{dt} |\psi(t)\rangle = H\,|\psi(t)\rangle \tag{3.23}$$

Here H is the Hamiltonian of the system, which in this case is proportional to the operator X. If we briefly return to the notation of Stern-Gerlach to describe the physical problem where the particle starts in the initial state $|+\rangle$ then:

$$|\psi(t)\rangle = e^{-iHt/\hbar}\,|\psi(0)\rangle = \frac{1}{\sqrt{2}}\left(e^{-i\omega t/2}\,|+\rangle_x + e^{i\omega t/2}\,|-\rangle_x\right), \tag{3.24}$$

where ω is a constant that contains the information describing the interaction between the particle and the magnetic field. As a result:

$$P(\psi(t), |+\rangle_x) = |\,{}_x\langle +|\psi(t)\rangle|^2 = \frac{1}{2} \tag{3.25}$$

i.e. for a particle originally in state $|+\rangle$, the probability of measuring it to be in state $|+\rangle_x$ in this magnetic field is $1/2$, independently of time. This exactly describes the role of the X analyzer as

we saw it in Stern-Gerlach (see Figure 2.4). Instead, if we consider the state of the particle at time $t = \frac{\pi}{\omega}$ we find

$$\begin{aligned} |\psi(\pi/\omega)\rangle &= \frac{e^{-i\pi/2}\,|+\rangle_x + e^{i\pi/2}\,|-\rangle_x}{\sqrt{2}} \\ &= -\frac{i(|+\rangle_x - |-\rangle_x)}{\sqrt{2}} \\ &= -i\,|-\rangle \end{aligned} \tag{3.26}$$

The above state is physically equivalent to the state $|-\rangle$ and this is the exact output we would expect from $X\,|+\rangle$. We have now seen how the magnetic fields of Stern-Gerlach can simultaneously allow us to describe measurement operators and quantum gates. It is worth noting that the systems of Stern-Gerlach are not practical for real quantum computing devices, but these examples serve to develop an intuition for how quantum computers function.

Finally, we will also still deal with the concept of measurement in our discussion of generalized quantum computing. Instead of worrying about measuring the spin about the x, y, and z axes however, we will only concern ourselves with measurements of spin about the z-axis. Equivalently in our generalized notation this means we are concerning ourselves with measurements associated with the computational basis. Again after a measurement the quantum state collapses to the basis states associated with that measurement. So the state of qubits will collapse to either $|0\rangle$ or $|1\rangle$. This gives the feel that we are measuring the probability of a classical bit being either in state 0 or 1. We will further specify the relationship between classical computing and quantum computing in a later section.

Chapter 4

Toolbox of Quantum Computing

We now turn our attention solely to quantum computing. We begin by developing a "quantum toolbox," where we explore the mathematics of quantum computing theory. By the end of this section readers will have an understanding of how to represent qubits and systems of qubits as well as quantum gates and a universal set of quantum gates. We also explore the notion of quantum interference and some fundamental manipulations of quantum gates.

4.1 Qubits: A New Perspective

In classical computing the unit of information is called a bit. A bit takes the value of 0 or 1, and it can do so either deterministically or probabilistically. We can define the state of a deterministic bit as a discrete object of the set $\{0, 1\}$. We define a probabilistic bit as a continuous object $p_0 \in [0, 1]$, where p_0 describes the probability of the bit taking the value 0. We can thus define $p_1 = (1 - p_0)$ and represent a probabilistic bit as the vector (p_0, p_1).

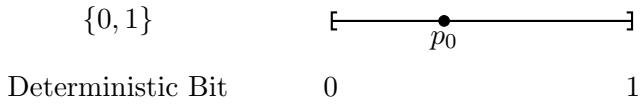

Figure 4.1: *State Space of Classical Bits.* (Left) State space of a deterministic bit. (Right) State space of a probabilistic bit, parametrized by p_0.

In the same manner that a classical bit is either a 0 or 1, a qubit is either in state $|0\rangle$ or $|1\rangle$ upon measurement. The important distinction here, and what gives quantum computing its power,

is that at a given moment in time a qubit is in a superposition of these two states, and only when measured it collapses entirely into the state $|0\rangle$ or the state $|1\rangle$. Just as with the probabilistic bit, the qubit can have different probabilities of being measured in each respective state, so we again describe them with a two dimensional vector, (a_0, a_1). Unlike the probabilistic bit, however, the entries of these vectors can be complex numbers, so they do not directly describe probabilities. We call the coefficients a_0 and a_1 amplitudes, and their magnitude squared describes the probabilities of a qubit being measured in a respective state. In particular the probability of a qubit being in state $|k\rangle$ upon measurement is $|a_k|^2$. Thus $|a_0|^2 + |a_1|^2 = 1$, or equivalently quantum states have norm 1, which we have seen from the Stern-Gerlach experiment of Figure 2.3. We can also represent an amplitude as $a_k = r_k e^{i\theta_k}$, where r_k describes the *magnitude of the amplitude*, and $e^{i\theta_k}$ describes the *phase of the amplitude*. (Here, r_k, θ_k are real numbers.) This formulation makes it clear that the probability of measuring state $|k\rangle$ is described by the magnitude squared of r_k, but the role of phase θ_k is murkier. However, the complex amplitudes of quantum states play an important role in quantum interference, a concept we will explore in a later section.

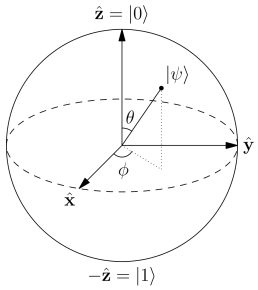

Figure 4.2: *The Bloch Sphere.* Visual representation of a quantum state, $|\psi\rangle = \cos\frac{\theta}{2}|0\rangle + e^{i\phi}\sin\frac{\theta}{2}|1\rangle$. Image "Bloch Sphere" by Glosser.ca is licensed under CC BY-SA 3.0.

It is also common to visualize the state of a qubit as a point in the Bloch Sphere, as seen in Figure 4.2. This alternative representation offers a visual reference of both phase and the probabilities of measuring a state as either of the basis states, the north and south poles of the sphere. To get a feel for this consider that—up to a factor in the unit circle—any quantum state

can be described as

$$|\psi\rangle = \cos\left(\frac{\theta}{2}\right)|0\rangle + e^{i\phi}\sin\left(\frac{\theta}{2}\right)|1\rangle \tag{4.1}$$

for certain $\theta \in [0, \pi]$ and $\phi \in [0, 2\pi]$; in particular, $\cos(\theta/2) \geq 0$ and $\sin(\theta/2) \geq 0$. Thus we can use the parameters θ and ϕ to describe a quantum state as a point on a sphere. For instance, the basis states, $|0\rangle$ and $|1\rangle$ correspond to $\theta = 0$ and $\theta = \pi$, or the north and south poles, respectively. In general, the probabilities of measurement in each state are controlled entirely by the parameter θ, which describes the quantum state's proximity to either pole. On the contrary, ϕ allows us to encode relative phase into a quantum state, and it has no effect on measurement probabilities. We will commonly refer to this relative phase, ϕ, as the *phase of the quantum state.*

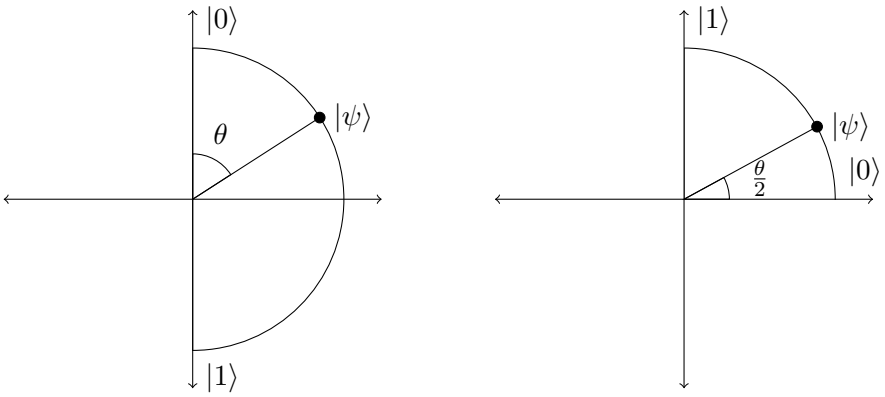

Figure 4.3: *Equivalent representations of a qubit.* (Left) State of a qubit $|\psi\rangle$ in the Bloch sphere representation when $\phi = 0$. (Right) Same state in $\mathbb{R}^2$, from the coordinates of the vector representation of $|\psi\rangle$. States with only real amplitudes are described by an arc of radius one in the first quadrant of $\mathbb{R}^2$.

4.2 Intermezzo: Kronecker Products

As we move our discussion towards systems with multiple qubits we require the use of the Kronecker product, denoted by $\otimes$. If $A \in \mathbb{C}^{m\times m}$ and $B \in \mathbb{C}^{n\times n}$ then we define the Kronecker

product between A and B as

$$A \otimes B = \begin{bmatrix} a_{11}B & a_{12}B & \dots & a_{1m}B \\ a_{21}B & a_{22}B & \dots & a_{2m}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}B & a_{m2}B & \dots & a_{mm}B \end{bmatrix} \tag{4.2}$$

It is clear that $(A \otimes B) \in \mathbb{C}^{mn \times mn}$ and that the Kronecker product is typically non-commutative. We will often denote the Kronecker product of several matrices C_i, for $i = 1, \dots, n$ as $\otimes_{i=1}^{n} C_i$.

We will often use the well-known result that for vectors $a \in \mathbb{C}^m$ and $b \in \mathbb{C}^n$ that

$$(A \otimes B)(a \otimes b) = (Aa) \otimes (Bb) \tag{4.3}$$

And more generally

$$(\otimes_{i=1}^{k} C_i)(\otimes_{i=1}^{k} c_i) = \otimes_{i=1}^{k} (C_i c_i) \tag{4.4}$$

4.3 Representation of multiple qubit states

For a quantum computer to be useful it needs to have multiple qubits. So far we have only discussed the states of single-qubits, but we need to extend those ideas to describe the states of multiple qubits at once. We will begin by describing multiple independent qubits. Ket notation yields a simple representation of such states. For instance, for a two qubit systen, given the first qubit is in state $|\psi_0\rangle$ and the second is in state $|\psi_1\rangle$ then we can describe the state of the system, $|\psi\rangle$, as

$$|\psi\rangle = |\psi_0\rangle \otimes |\psi_1\rangle \equiv |\psi_0\rangle\,|\psi_1\rangle \tag{4.5}$$

We will generally be describing states with respect to the computational basis i.e. the set of measurement states, or the output states of a quantum measurement. These states will be kets with bit strings of length n as their argument, where n is the number of qubits in the system. Just

as we can describe multiple classical bits as a bit string, such as 1010, the quantum equivalent can be described as

$$|1\rangle |0\rangle |1\rangle |0\rangle \equiv |1010\rangle \tag{4.6}$$

This makes it clear that after a quantum measurement, it is trivial to output the results as a bit string that can be sent to a classical computer. It will occasionally be useful to denote computational basis states through decimal numbers in correspondence with binary numbers. For example, a system composed of six qubits all in state $|1\rangle$ can be described as

$$|111111\rangle \equiv |2^6 - 1\rangle \equiv |63\rangle \tag{4.7}$$

While the above notation in some sense obscures the physical interpretation of the state $|111111\rangle$, it provides a much more compact method of identifying specific states as it does not rely on long binary representations of numbers. Further as long as the number of qubits in the system is explicitly defined, kets with decimal numbers as their arguments can always be translated back into their binary form, which restores the physical interpretation of the system.

Instead, when using the vector representation of qubits, we use the Kronecker product to represent the states of multiple qubits. For instance, if we have two qubits, $|x\rangle = (a_0, a_1)$ and $|y\rangle = (b_0, b_1)$, we can represent the state of the system, $|\psi\rangle$ as

$$|\psi\rangle = |x\rangle \otimes |y\rangle = \begin{bmatrix} a_0 |y\rangle \\ a_1 |y\rangle \end{bmatrix} = \begin{bmatrix} a_0 b_0 \\ a_0 b_1 \\ a_1 b_0 \\ a_1 b_1 \end{bmatrix} \tag{4.8}$$

(The Kronecker product between vectors of dimension m and n is a vector of dimension mn.) In particular, $|a_i b_j|^2$ is the probability that $|x\rangle = |i\rangle$ and $|y\rangle = |j\rangle$ upon measurement, when the state of each qubit is independent (in a probabilistic sense) of the other. From this, it is natural that the

computational basis states for a system of two qubits are as follows:

$$\begin{aligned} |00\rangle &= (1,0,0,0) \\ |01\rangle &= (0,1,0,0) \\ |10\rangle &= (0,0,1,0) \\ |11\rangle &= (0,0,0,1) \end{aligned} \tag{4.9}$$

As single-qubits are always of dimension 2, the dimension of any quantum state composed of n qubits will be 2^n. This leads to the intuition that for a system composed of n qubits the computational basis vectors are equivalent to the canonical basis vectors of dimension 2^n.

We define a quantum system, $|\psi\rangle$, to *separable* if it can be composed as a Kronecker product of single-qubit states. Namely, there are qubits $|\psi_i\rangle$ for for $i = 0, \ldots, (n-1)$ such that

$$|\psi\rangle = |\psi_0\rangle \cdots |\psi_{n-1}\rangle \tag{4.10}$$

The notion of a separable quantum system can therefore be thought of as a measurement of independent qubits—in a probabilistic sense.

On the other hand, when we cannot describe qubits independently, we say they are *entangled*. We will further explore this concept in a later section, however it is assumed that at the start of a quantum algorithm, all qubits are separable, and we will only entangle them by performing special operations upon pairs of qubits.

4.4 Single-qubit Gates

Quantum gates are linear operators that act on qubits and build the foundation of quantum algorithms. As we have seen, qubit states must have norm 1, and thus quantum gates must preserve that norm. This implies that quantum gates are unitary matrices. A defining property of a unitary matrix, U, is that $U^*U = UU^* = I$, or that its conjugate transpose is also its inverse. This immediately implies that for every quantum operation, there exists an associated inverse, and quantum computing is thus reversible. This is in stark contrast to classical computing where

fundamental logic gates, such as the AND gate ($\wedge$) is not reversible because $0 \wedge 0 = 0 \wedge 1 = 1 \wedge 0$. Similarly, the classical OR gate ($\vee$) is not reversible because $0 \vee 1 = 1 \vee 0 = 1 \vee 1$. In particular, as we proceed in our discussion we will have to handle reversible implementations of non-reversible classical gates.

For now we will begin by discussing some fundamental quantum gates. The first four we are already familiar with from the Stern-Gerlach experiment: the *Pauli gates* defined as

$$I \equiv \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad X \equiv \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad Y \equiv \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \quad Z \equiv \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \tag{4.11}$$

Obviously the identity gate, I, serves only as a state preservation gate because $I|\psi\rangle = |\psi\rangle$, for any $|\psi\rangle$.

The gate X is directly related to the classical NOT, or bit flip, gate. In classical computing the NOT gate maps a bit x_1 to $x_1 \oplus 1$, effectively turning a 0 into a 1 and vice versa (here $\oplus$ denotes the exclusive-or operation, XOR). Instead, it is easy to verify that $X(a, b) = (b, a)$, or that the gate X effectively swaps the amplitudes of the basis states, and thus implicitly sends the state $|0\rangle$ to $|1\rangle$ and $|1\rangle$ to $|0\rangle$. Thus, from now on, we will refer to the gate X as the NOT gate of quantum computing.

The gate Z induces a change in the phase of a given state. Specifically, it twists the phase of a quantum state by π radians, and can be visualized as a rotation about the z-axis on the Bloch sphere.

A twist gate is generally defined as

$$T(\theta) = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{bmatrix} \tag{4.12}$$

Just like with the Z gate, a twist gate $T(\theta)$ twists the phase of a quantum state by θ radians, and can be visualized as a rotation of θ about the z-axis of the Bloch sphere. Thus Z is just a special case of a twist gate, namely $Z = T(\pi)$.

One can then notice that excluding a global phase factor, $Y = X \cdot Z$. Hence, the gate Y swaps the amplitudes of a given state while simultaneously changing the relative phase.

Another fundamental gate is the *Hadamard gate*, which is a very common gate in the initial steps of many quantum algorithms. This is because it sends the computational basis states into a uniform superposition state. It is defined as

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{4.13}$$

Superposition is the main feature of quantum computing that leads to the significant speed increases compared to classical algorithms, and as such it is very common to begin an algorithm by sending qubits into a uniform superposition.

Gate	**Action**	**Gate** $\|0\rangle$	**Gate** $\|1\rangle$
I	Identity	$\|0\rangle$	$\|1\rangle$
X	NOT	$\|1\rangle$	$\|0\rangle$
Z	Relative phase twist of π	$\|0\rangle$	$-\|1\rangle$
$T(\theta)$	Relative phase twist of θ	$\|0\rangle$	$e^{i\theta}\|1\rangle$
Y	NOT with relative phase twist of π	$\|1\rangle$	$\|0\rangle$
H	Uniform superposition	$\frac{\|0\rangle+\|1\rangle}{\sqrt{2}}$	$\frac{\|0\rangle+\|1\rangle}{\sqrt{2}}$

Table 4.1: *Common single-qubit Gates.* Summary of common single-qubit gates and their action over the computational basis.

4.5 Multiple Qubit Gates

Quantum gates map quantum states to quantum states. As all single-qubit states are of dimension 2, single-qubit gates are all 2×2 unitary matrices. In general, because a quantum state composed of n qubits is of dimension 2^n, quantum gates that act on n qubits are $2^n \times 2^n$ matrices.

Let us consider a system of two qubits, $|q_0\rangle$ and $|q_1\rangle$. Imagine wanting to apply the NOT gate to $|q_0\rangle$ but the Hadamard gate to $|q_1\rangle$. Due to the identities in equations (4.3) and (4.4), to design a gate U that acts this way on the two qubits we just need to consider the Kronecker

product between the X and H gates. As a result:

$$U = (X \otimes H) = \begin{bmatrix} 0 \cdot H & 1 \cdot H \\ 1 \cdot H & 0 \cdot H \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{array}{c} \begin{matrix} |00\rangle & |01\rangle & |10\rangle & |11\rangle \end{matrix} \\ \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 \\ 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix} \end{array} \begin{matrix} |00\rangle \\ |01\rangle \\ |10\rangle \\ |11\rangle \end{matrix} \tag{4.14}$$

We note that for a general gate it is useful to view the columns as the outputs given an input. For instance, the column corresponding with the input $|00\rangle$ (i.e. first column) is mapped by U to the quantum state $(|10\rangle + |11\rangle)/\sqrt{2}$. In particular, the first qubit was flipped, whereas the second qubit is now in uniform superposition. Further observation of this gate shows it performs exactly as we would like it to.

Quantum algorithms generally assume that the initial state of a system has every qubit in state $|0\rangle$. In particular for a system of n qubits the intial state is $|0\rangle^{\otimes n}$ or equivalently $|0^n\rangle$, where $0^n = 0 \cdots 0$, a string of n characters. Many quantum algorithms rely on first sending state of the system into a uniform superposition. Since the Hadamard gate sends the state $|0\rangle$ to $(|0\rangle + |1\rangle)/\sqrt{2}$, a common initialization step for several quantum algorithms is

$$H^{\otimes n} |0\rangle^{\otimes n} = \frac{1}{2^{n/2}} \sum_{i=0}^{2^n - 1} |i\rangle \tag{4.15}$$

where we here use the notation of equation (4.7).

In what follows, we say that a quantum gate is *separable* if it can be represented as the Kronecker product of single-qubit gates. This reinforces the intuition that separable systems can be thought of as leading to independent measurements.

When working with a large number of qubits, the matrix representation of quantum gates can be difficult to work with as they grow in size exponentially. Further, expressing separable gates that act on multiple qubits can quickly become confusing, as it is unclear what gate acts on what qubit. Quantum circuits are diagrams that can be used to describe operations on qubits in an easily readable way.

To fix ideas, Figure 4.4 shows an example quantum circuit for a three qubit system. Each qubit has an associated *qubit line*, with different quantum gates on each line representing their application to that qubit. Quantum circuits are read left to right, and due to their resemblance to musical scores, are sometimes referred to as *quantum scores*. Quantum circuits make it much easier to quickly ascertain what operations are being applied to each qubit in a quantum algorithm.

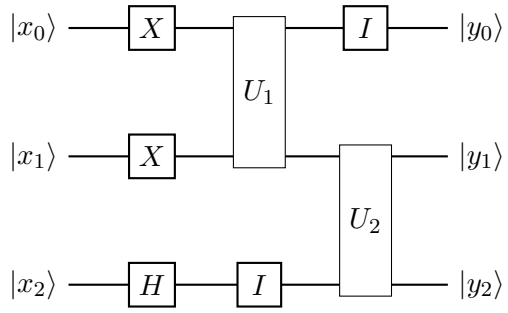

Figure 4.4: *Example of Quantum Circuit.* Given the input state $|x_0\rangle\,|x_1\rangle\,|x_2\rangle$ the system is in the output state $|y_0\rangle\,|y_1\rangle\,|y_2\rangle$ after this circuit. Single-qubit gates, such as X, H and I are denoted by single and vertically aligned boxes along the qubit lines. Gates that take multiple qubits as inputs such as U_1 and U_2 are denoted as boxes on multiple qubit lines. Note here we have drawn I as a quantum gate, but it is common for the identity gate to be omitted from quantum circuits. One could equivalently denote $|y_0\rangle\,|y_1\rangle\,|y_2\rangle = (I \otimes U_2)(U_1 \otimes I)(X \otimes H \otimes H)(|x_0\rangle\,|x_1\rangle\,|x_2\rangle)$.

Thus far we have only discussed separable quantum systems, or systems defined as the Kronecker product of single-qubit states. An *entangled state* is one that cannot be composed in such a way. Just as we relate separable systems to independent measurements, entangled systems relate to dependent measurements. An entangling quantum gate similarly cannot be composed as the Kronecker product of single-qubit gates. The most basic entangling gate is the controlled-not or

CNOT gate, which we will denote $\wedge_1(X)$. In matrix form, this is defined as

$$\wedge_1(X) \equiv \begin{array}{c} \begin{matrix} |00\rangle & |01\rangle & |10\rangle & |11\rangle \end{matrix} \\ \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \end{array} \begin{matrix} |00\rangle \\ |01\rangle \\ |10\rangle \\ |11\rangle \end{matrix} \tag{4.16}$$

To understand this gate it is convenient to refer to the first qubit as the control, and to the second qubit as the target. Equation (4.16) now makes it clear that when the control qubit is in state $|0\rangle$ the identity gate is applied to the target qubit, however, when the control is in state $|1\rangle$ the NOT gate is applied to the target. Equivalently, $\wedge_1(X)$ transforms $|x\rangle\,|y\rangle$ into $|x\rangle\,|x \oplus y\rangle$ i.e. the CNOT gate is encoding the exclusive-or (XOR) of the control and target qubits into the target qubit. In particular, after the application of the CNOT gate the target qubit state is dependent on the control qubit state. The standard quantum circuit representation of the CNOT gate can be seen in Figure 4.5.

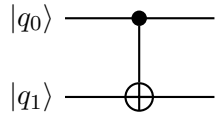

Figure 4.5: *Controlled-NOT Quantum Circuit.* Representation of a CNOT gate, $\wedge_1(X)$, as a quantum circuit. Here q_0 is the control qubit and q_1 is the target qubit.

Notice that applying the CNOT gate to a basis state results in another basis state which is clearly separable. Let us consider instead applying it with a control qubit that is in superposition.

If we define the gate $U_B = \wedge_1(X)(H \otimes I)$ we find (see Figure 4.6):

$$U_B = \frac{1}{\sqrt{2}} \begin{array}{c} \begin{matrix} |00\rangle & |01\rangle & |10\rangle & |11\rangle \end{matrix} \\ \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & -1 \\ 1 & 0 & -1 & 0 \end{bmatrix} \end{array} \begin{matrix} |00\rangle \\ |01\rangle \\ |10\rangle \\ |11\rangle \end{matrix} \tag{4.17}$$

In particular,

$$\begin{aligned} U_B |00\rangle &= \frac{|00\rangle + |11\rangle}{\sqrt{2}}; \quad U_B |01\rangle = \frac{|01\rangle + |10\rangle}{\sqrt{2}} \\ U_B |10\rangle &= \frac{|00\rangle - |11\rangle}{\sqrt{2}}; \quad U_B |11\rangle = \frac{|01\rangle - |10\rangle}{\sqrt{2}} \end{aligned} \tag{4.18}$$

The four states of Equation (4.18) are known as the *Bell States*, and are classic examples of entangled quantum states. The Bell states are often described as the maximally entangled states and are important in quantum information theory.

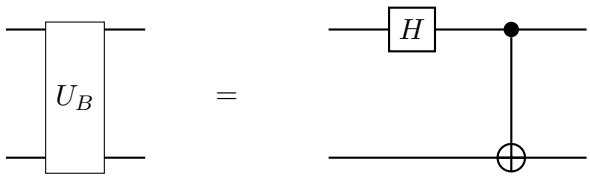

Figure 4.6: *Bell State Entanglement.* Quantum circuit that maps the computational basis states to the Bell States.

While we have only discussed one entangling gate, CNOT, it is the only entangling gate we will need. In fact, Theorem 4.3.7 of [16] states the set of gates $\{H, T(\frac{\pi}{4}), \wedge_1(X)\}$ is universal for quantum computing. Namely, one can approximate any quantum operation using a quantum circuit composed only of gates from this set. Explicitly, for any gate E we can compose $\tilde{E}$ such that $\|E - \tilde{E}\|$ is arbitrarily small. Note that as matrix norms are equivalent there is no need to explicitly define which norm is used for this calculation.

4.6 Quantum Interference

Quantum interference is a key property of superposition. It allows us to bias the measurement of a qubit toward a specific basis state. This is an important concept of many quantum algorithms, where we design gates that allow us to quickly shape uniform distributions towards some previously unknown, but desired state. In order to see how quantum interference works as a property of superposition we must consider an alternative representation of states.

While a quantum state can be in a superposition we can also describe them as *mixed or ensemble states*. These are more of a direct analogue to the probabilistic bit, and we define a mixed state as a qubit that is in *pure state* $|\psi_i\rangle$ with probability p_i for $i = 1 : k$, where $(p_1 + \ldots + p_k) = 1$. A pure state is defined simply as quantum state with a distinct state vector and so far we have only dealt with such states in our discussions. We denote a mixed state as

$$[(|\psi_1\rangle, p_1), (|\psi_2\rangle, p_2), \ldots, (|\psi_k\rangle, p_k)] \tag{4.19}$$

We will explore the different behaviors of mixed states and superposition states through the following example.

Consider the Hadamard gate:

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

If we apply H to $|0\rangle$ or $|1\rangle$ then there is equal probability of observing either $|0\rangle$ or $|1\rangle$ upon measurement. Similarly, if we apply H to the mixed state $[(|0\rangle, \frac{1}{2}), (|1\rangle, \frac{1}{2})]$ then $|0\rangle$ and $|1\rangle$ are equally likely. However if we apply the Hadamard gate to a state in superposition such as $\frac{|0\rangle - |1\rangle}{\sqrt{2}}$, we will observe state $|1\rangle$ with certainty upon measurement. This is because the negative entry of the superposition state cancels out the probability of observing $|0\rangle$ after applying the Hadamard gate. This is an exhibit of quantum interference and it is clear that such a phenomena can only occur in superposition states, but not mixed states.

When we begin working with the physical quantum computer, we will see that this difference in behavior between superposition states and mixed states is significant. In particular the exact

scenario of the above example may happen if an incidental measurement occurred on a qubit in superposition, before we had meant to perform a measurement. This clearly would lead to an error in our final result, and this loss of quantum information is called *quantum decoherence*.

4.7 Encoding Boolean Functions as Quantum Gates

Consider some given function $f : \{0,1\}^n \to \{0,1\}^m$ and suppose we would like to encode it as a quantum gate U_f. This is easier said than done because, unlike the function f, the linear operator U_f must be reversible. Furthermore, it must also be a unitary matrix. A way around this is to use extra qubits [16]. Specifically, we define $U_f : \mathbb{C}^{n+m} \to \mathbb{C}^{n+m}$ as

$$U_f \left|x\right\rangle \left|y\right\rangle = \left|x\right\rangle \left|f(x) \oplus y\right\rangle \tag{4.20}$$

i.e. if $f(x) = (z_1, z_2, \ldots, z_m)$ then $U_f \left|x\right\rangle \left|y\right\rangle = \left|x\right\rangle \left|z_1 \oplus y_1, z_2 \oplus y_2, \ldots, z_m \oplus y_m\right\rangle$. In this context, $\left|x\right\rangle$ and $\left|y\right\rangle$ are commonly described as the control and the target qubit, respectively.

To fix ideas, note that if $I(x) = x$, for $x \in \{0,1\}$, then $U_I \left|x\right\rangle \left|y\right\rangle = \left|x\right\rangle \left|x \oplus y\right\rangle$ i.e. U_I is the same as the CNOT gate.

More generally, it is easy to see that U_f is reversible because $U_f \left|x\right\rangle \left|f(x) \oplus y\right\rangle = \left|x\right\rangle \left|y\right\rangle$, and from equation (4.20) one can see that U_f is in fact just a permutation matrix. This implies that U_f is indeed unitary as well.

Let us now consider a general function $f : \{0,1\} \to \{0,1\}$ and its quantum counterpart of U_f. It is very clear that we can use U_f to evaluate the function f, just as we would on a classical computer. Due to superposition, however, this quantum gate offers new possibilities that are impossible using classical computing techniques.

To start appreciating this we first explore the effects of having the target qubit in a superposition. Let the control qubit be a basis state, $\left|x\right\rangle \in \{\left|0\right\rangle, \left|1\right\rangle\}$, and the target qubit be one of the

states from the Hadamard basis, $|y\rangle = (|0\rangle - |1\rangle)/\sqrt{2}$ and consider the effect of applying U_f:

$$\begin{aligned} U_f |x\rangle |y\rangle &= \frac{U_f |x\rangle |0\rangle - U_f |x\rangle |1\rangle}{\sqrt{2}} \\ &= \frac{|x\rangle |f(x)\rangle - |x\rangle |f(x) \oplus 1\rangle}{\sqrt{2}} \\ &= (-1)^{f(x)} |x\rangle \frac{(|0\rangle - |1\rangle)}{\sqrt{2}} \\ &= (-1)^{f(x)} |x\rangle |y\rangle \end{aligned} \tag{4.21}$$

Thus $|x\rangle |y\rangle$ is an eigenstate for U_f, with the sign of the associated eigenvalue determined by the value of $f(x)$. This practice, where we use an eigenstate of a quantum gate to kick back the associated eigenvalue as a phase factor, is known as *phase kick back*. It is important to notice that we have encoded $f(x)$ in the global phase, which we cannot access from measurement. Thus we will need to do a bit more work to utilize this concept of phase kick back. As it turns out it is an integral concept to Deutsch's algorithm, which we will study in a later section and see how to take advantage of this global phase factor.

Instead, if we now let the control qubit be the superposition state $|x\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$, we can in some sense evaluate $f(0)$ and $f(1)$ simultaneously. There is, however, an important subtlety here: one is not able to determine both $f(0)$ and $f(1)$ explicitly at once. To see this consider the target as a basis state $|y\rangle \in \{|0\rangle, |1\rangle\}$:

$$\begin{aligned} U_f |x\rangle |y\rangle &= \frac{U_f |0\rangle |y\rangle + U_f |1\rangle |y\rangle}{\sqrt{2}} \\ &= \frac{|0\rangle |f(0) \oplus y\rangle + |1\rangle |f(1) \oplus y\rangle}{\sqrt{2}} \end{aligned} \tag{4.22}$$

Thus, while in the above we have certainly encoded the values of both $f(0)$ and $f(1)$ in our qubits, when performing a measurement, the states would collapse out of superposition so that we would only observe one of the two values. So again it is unclear how to gain a quantum advantage from this formulation of U_f. We will see the utility of this when we study Deutsch's algorithm.

4.8 Conditionally Applying Quantum Gates

It is very common to conditionally apply a quantum gate, a concept described as adding a control to a quantum gate. We have already seen this with the controlled-not gate, $\wedge_1(X)$, where the NOT gate is applied to the target qubit only when the control qubit is in state $|1\rangle$. In this section, we see how to extend this idea to a general quantum gate U with an arbitrary number of controls. We denote a controlled-U gate with n controls as $\wedge_n(U)$ (see Figure 4.7).

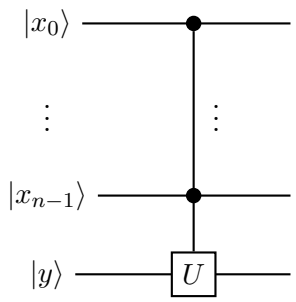

Figure 4.7: *Quantum Circuit of* $\wedge_n(U)$. Quantum circuit with controls on qubits $|x_0\rangle, \ldots, |x_{n-1}\rangle$ to determine whether the gate U is applied to $|y\rangle$. The thick dots on the qubit lines designate which qubits are the controls.

Consider a single-qubit gate U. The controlled-U gate, denoted as $\wedge_1(U)$, applies the U gate to the target qubit $|y\rangle$ only when the control qubit $|x_0\rangle$ is in state $|1\rangle$. Otherwise, it leaves the target qubit untouched. Clearly, such a gate is typically entangling and may be represented in block form as

$$\wedge_1(U) = \begin{bmatrix} I & 0 \\ 0 & U \end{bmatrix} \tag{4.23}$$

While this construction is trivial, we have we seen that $\wedge_1(X)$ is the only entangling gate needed for a universal set of gates. In fact, when we move to programming on a real quantum computer, $\wedge_1(X)$ will be the only entangling gate at our disposal. Thus—in practice—we need to construct $\wedge_1(U)$ via the controlled-not gate. As we will see, this relies on factorizations of the operator U that use the X gate.

To fix ideas let us consider the construction of a controlled-Z gate, $\wedge_1(Z)$. Notice that $Z = HXH$ and that H is its own inverse. This suggests defining $\wedge_1(Z)$ via the circuit in Figure 4.8. Indeed, if $|x_0\rangle = |0\rangle$ then $HIH = I$ is applied to the target qubit $|y\rangle$. However, if $|x_0\rangle = |1\rangle$ then $HXH = Z$ is applied to $|y\rangle$. We have thus successfully constructed a controlled-Z gate.

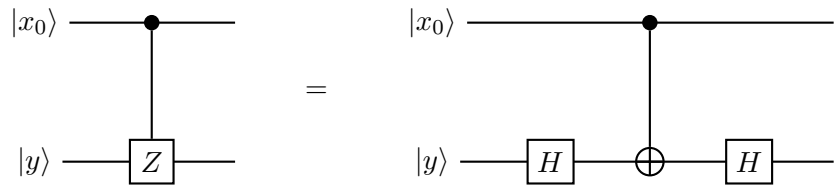

Figure 4.8: *Circuit for Controlled-Z gate,* $\wedge_1(Z)$.

4.8.1 Gates with One Control

Define

$$\mathrm{SU}(2) := \{W \in \mathbb{C}^{2\times 2} \text{ such that } \det(W) = 1\} \tag{4.24}$$

i.e. the *special unitary group of degree* 2 [2].

In this section we will discuss implementing gates with one control for any single-qubit gate U. We first must recognize that any (2×2) unitary matrix, U, can be factorized in the form of $U = \mathrm{Ph}(\delta) \cdot W$, for certain $\delta \in [0, 2\pi)$ and $W \in \mathrm{SU}(2)$ [2]. Here $\mathrm{Ph}(\delta)$ is called a *phase gate* and induces a change of δ radians in the global phase of a single-qubit. It is defined as

$$\mathrm{Ph}(\delta) = \begin{bmatrix} e^{i\delta} & 0 \\ 0 & e^{i\delta} \end{bmatrix} \tag{4.25}$$

Clearly, the circuit associated with $\wedge_1(U)$ is as displayed in Figure 4.9. In particular:

$$\wedge_1(U) = \wedge_1(V) \cdot \wedge_1(W), \text{ with } V = \mathrm{Ph}(\delta) \tag{4.26}$$

To determine $\wedge_1(U)$ we just need therefore to determine $\wedge_1(V)$ and $\wedge_1(W)$. The former

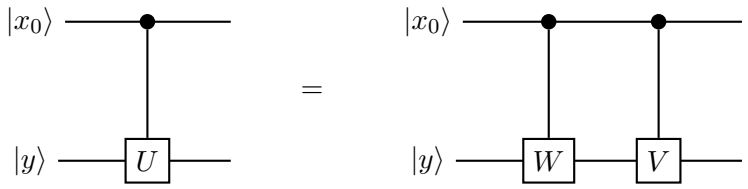

Figure 4.9: *Quantum Circuit for* $\wedge_1(U)$. Circuit for $\wedge_1(U)$ for an arbitrary unitary gate, U, such that $U = V \cdot W$ with $V = \mathrm{Ph}(\delta)$ and $W \in \mathrm{SU}(2)$.

operator is straightforward. In fact, we find that:

$$\wedge_1(V) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & e^{i\delta} & 0 \\ 0 & 0 & 0 & e^{i\delta} \end{bmatrix} = T(\delta) \otimes I \tag{4.27}$$

where $T(\delta)$ is a phase twist gate of δ radians (see Table 4.1). In particular, we may represent the single-qubit controlled-V gate with the circuit displayed in Figure 4.10.

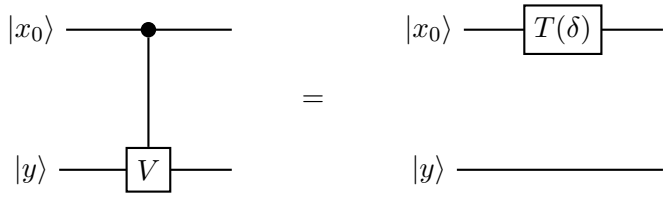

Figure 4.10: *Quantum Circuit for* $\wedge_1(V)$. Adding a control to the gate $V = \mathrm{Ph}(\delta)$.

Next we see how to construct $\wedge_1(W)$. In particular, we only want to apply a W gate to the target qubit, $|y\rangle$, if the control qubit, $|x_0\rangle$, is in state $|1\rangle$. For this we invoke Lemma 4.3 of [2], which asserts that for any $W \in \mathrm{SU}(2)$ there are matrices $A, B, C \in \mathrm{SU}(2)$ such that $ABC = I$ and $AXBXC = W$. This suggests the circuit in Figure 4.11 to obtain $\wedge_1(W)$. Indeed, if $|x_0\rangle = |0\rangle$ then X acts as the identity operator; in particular, the operator $AIBIC = ABC = I$ is applied to $|y\rangle$. However, if $|x_0\rangle = |1\rangle$ then the operator $AXBXC = W$ is applied to $|y\rangle$. The circuit in

Figure 4.11 therefore recreates the controlled-W gate. In particular:

$$\wedge_1(W) = (I \otimes C) \cdot \wedge_1(X) \cdot (I \otimes B) \cdot \wedge_1(X) \cdot (I \otimes A) \tag{4.28}$$

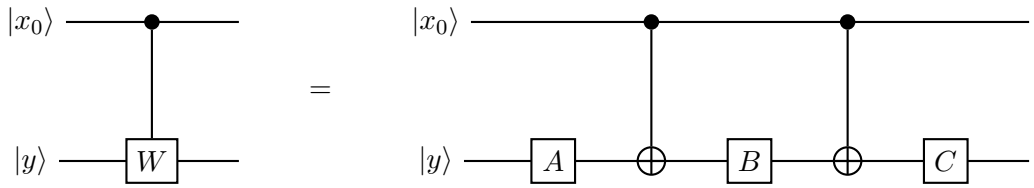

Figure 4.11: *Quantum Circuit for controlled-W gate,* $\wedge_1(W)$*, for* $W \in SU(2)$. Here W admits the factorization $W = AXBXC$ for certain $A, B, C \in \mathrm{SU}(2)$.

4.8.2 Gates with Two Controls

We now will explore adding two controls to a single-qubit gate U. In other words given qubits $|x_0\rangle$, $|x_1\rangle$ and $|y\rangle$, apply gate U to qubit $|y\rangle$ if and only if $|x_0\rangle = |x_1\rangle = |1\rangle$. We achieve this by using a controlled-gate $\wedge_1(S)$, with S a quantum gate such that $U = S^2$, and constructing the circuit in Figure 4.12. (Such a factorization always exists because every unitary matrix U is normal and can thus be factorized as $U = VDV^*$, where V is unitary and D is diagonal. Thus $S = VD^{1/2}V^*$.) The appropriateness of this circuit to simulate $\wedge_2(U)$ can be seen in Table 4.2, which lists all the possible gates applied to the target qubit based on the values of the control qubits.

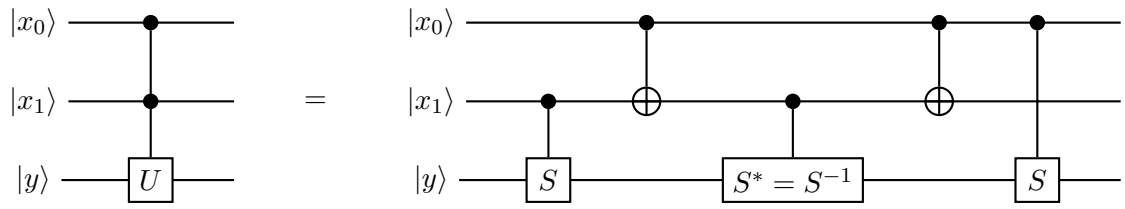

Figure 4.12: *Quantum Circuit for* $\wedge_2(U)$. Implementation of $\wedge_2(U)$ when $U = S^2$ and $S^* = S^{-1}$. The two controlled-NOT gates applied to $|x_1\rangle$ are to have this control qubit unchanged at the end of the circuit.

There is an implicit logic behind the construction of this circuit that is worth understanding before addressing cases with more controls. For this, it is convenient to identify basis states for the

$\|x_0x_1\rangle$	Gate Applied to $\|y\rangle$
$\|00\rangle$	$III = I$
$\|01\rangle$	$IS^*S = I$
$\|10\rangle$	$SS^*I = I$
$\|11\rangle$	$SIS = U$

Table 4.2: *Possible actions on target qubit for* $\wedge_2(U)$.

control qubits $|x_i\rangle$ with binary values x_i. As seen in Figure 4.12, due to the left most controlled-S gate, the gate S^{x_1} is first applied to $|y\rangle$. (Note that $S^{x_1} = I$ if $x_1 = 0$ but $S^{x_1} = S$ if $x_1 = 1$.) Following this, the left most CNOT gate between the first two qubit lines causes the gate $S^{-(x_0 \oplus x_1)}$ to be applied to $|y\rangle$. Finally, due to the right most controlled-S gate, S^{x_0} is the last gate applied to $|y\rangle$. As a result, the quantum gate

$$S^{x_1} \cdot S^{-(x_0 \oplus x_1)} \cdot S^{x_0} = S^{x_1 - (x_0 \oplus x_1) + x_0} \tag{4.29}$$

is applied to the target qubit $|y\rangle$. It turns out, however, that no matter the values for x_0 and x_1, the following identity holds [2]: $x_0 + x_1 - (x_0 \oplus x_1) = 2(x_0 \wedge x_1)$. As a result, the gate $S^{2(x_0 \wedge x_1)}$ is applied to the target gate, and this is $S^2 = U$ if and only if $x_0 = x_1 = 1$. Since $S^{2(x_0 \wedge x_1)} = I$ when $x_0 = 0$ or $x_1 = 0$, the correctness of the circuit is assured.

4.8.3 Gates with Several Controls

The construction of gates with n control qubits $|x_0\rangle, \ldots, |x_{n-1}\rangle$ is based on the identity [2]:

$$\sum_i x_i - \sum_{i<j} x_i \oplus x_j + \sum_{i<j<k} x_i \oplus x_j \oplus x_k - \sum_{i<j<k<\ell} x_i \oplus x_j \oplus x_k \oplus x_\ell + \ldots = 2^{n-1}(x_0 \wedge \ldots \wedge x_{n-1}) \tag{4.30}$$

and the factorization of a single-qubit quantum gate U as $U = V^{2^{n-1}}$, with $V^* = V^{-1}$ (The existence of such a unitary V can be shown in much the same way as before).

To fix ideas, Figure 4.13 displays a circuit associated with the controlled-U gate $\wedge_3(U)$. This construction relies on having a quantum gate V such that $U = V^4$; in particular, $V^* = V^{-1}$. As we advance through the circuit, we see that the following gates are sequentially applied to the target

qubit $|y\rangle$: V^{x_0}, $V^{-(x_0 \oplus x_1)}$, V^{x_1}, $V^{-(x_1 \oplus x_2)}$, $V^{(x_0 \oplus x_1 \oplus x_2)}$, $V^{-(x_0 \oplus x_2)}$ and V^{x_2}. Due to the identity in equation (4.30) with $n = 3$, the overall action of these gates is equivalent to applying $V^{4(x_0 \wedge x_1 \wedge x_2)}$ to $|y\rangle$, and the latter gate equals U when $(x_0 \wedge x_1 \wedge x_2) = 1$ and I otherwise.

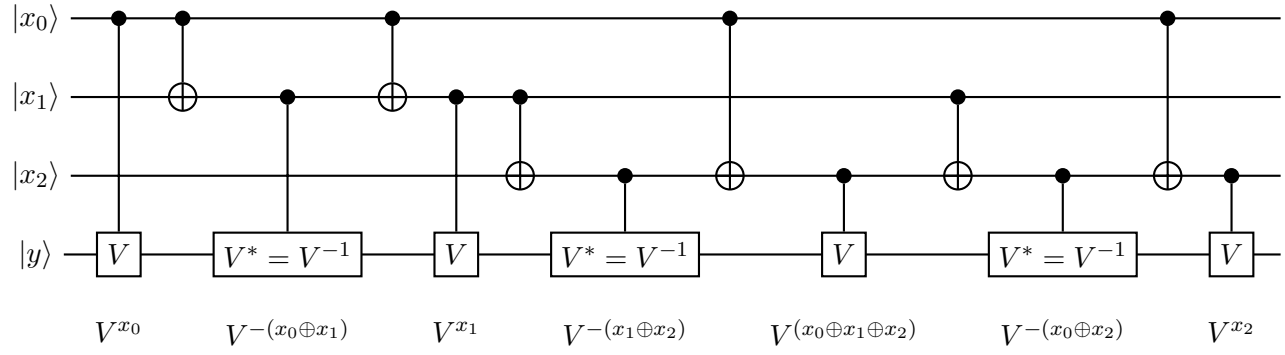

Figure 4.13: *Quantum Circuit for* $\wedge_3(U)$. Implementation of $\wedge_3(U)$ when $U = V^4$ and $V^* = V^{-1}$. Last row shows the sequence of quantum gates applied to the target qubit.

Chapter 5

Quantum Algorithms

The main goal of quantum computing is to utilize the unique structure and properties of quantum systems to gain algorithmic speed ups over classical approaches. Here we will explore two foundational quantum algorithms that demonstrate such speed up. Deutsch's algorithm is the first quantum algorithm that demonstrated the computational gains only possible with quantum computers [9]. Grover's quantum search algorithm is often toted as the prototypical quantum algorithm and is described as "finding a needle in a haystack" in sub-linear time [14]. The importance of these algorithms is mostly theoretical as they do not offer immediate practical use. They illustrate, however, key quantum concepts and the computational power of quantum architectures.

5.1 Deutsch's Algorithm

In 1985, a few years after Feynman's famous lectures on quantum computing, described in the Introduction, David Deutsch published a paper in which he formalized quantum complexity theory and quantum Turing machines [9]. From this the theory of quantum computing was fully realized, and it was possible to compare the computational speed of a classical computer against a quantum machine. In the same paper, Deutsch posed a problem, and a quantum algorithm to solve it, that demonstrated that quantum computers offer computational speed up compared to classical machines.

Deutsch's algorithm is one of the simplest quantum algorithms. It demonstrates a marginal but unexpected improvement in time complexity over its classical counterpart. Ultimately, it

serves as a nice introduction to how to think about and approach quantum algorithms. In fact, this algorithm demonstrates the utility of quantum concepts such as superposition and phase kick back. The manipulation of phase in superposition states is a fundamental technique of quantum computing that appears in almost all quantum algorithms.

The problem statement for Deutsch's algorithm is as follows.

Problem: Given $f : \{0,1\} \to \{0,1\}$, determine if f is a constant function with the least number of evaluations of the function f.

Let us first consider this problem classically. The only way we could determine if f is constant or not would be through two evaluations i.e. computing $f(0)$ and $f(1)$ and comparing the results. Alternatively, $f(0) \oplus f(1)$ would identify if f is constant or not because

$$f(0) \oplus f(1) = \begin{cases} 0, & \text{if } f \text{ is constant;} \\ 1, & \text{if } f \text{ is not constant.} \end{cases} \tag{5.1}$$

The key idea of Deutsch's algorithm is taking advantage of superposition to evaluate $f(0)$ and $f(1)$ simultaneously. This problem statement is of course very similar to that of Section 4.7, where we have a Booelan function, f, encoded as a quantum gate, U_f. In particular, if we utilize equation (4.21) we can encode in the relative phase of the control qubit the value of $f(0) \oplus f(1)$. To see this let $|x\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$ and $|y\rangle = (|0\rangle - |1\rangle)/\sqrt{2}$. Then

$$\begin{aligned} U_f |x\rangle |y\rangle &= \left(\frac{(-1)^{f(0)} |0\rangle + (-1)^{f(1)} |1\rangle}{\sqrt{2}} \right) |y\rangle \\ &= (-1)^{f(0)} \left(\frac{|0\rangle + (-1)^{f(0) \oplus f(1)} |1\rangle}{\sqrt{2}} \right) |y\rangle \end{aligned} \tag{5.2}$$

We can see in the above that we have encoded the value of $f(0) \oplus f(1)$ in the relative phase of the control qubit's superposition. Moreover, from equation (5.1), the control qubit will be in state $(|0\rangle + |1\rangle)/\sqrt{2}$ if f is constant but state $(|0\rangle - |1\rangle)/\sqrt{2}$ if not. We can recognize these as the states of the Hadamard basis. Applying a Hadamard gate to the control qubit will then map it to $|0\rangle$

if f is constant and to $|1\rangle$ otherwise. Thus, by measuring the state of the control qubit, we can deterministically determine if f is constant or not—with only one query to the circuit U_f.

Figure 5.1 shows a quantum circuit associated with Deutsch's algorithm.

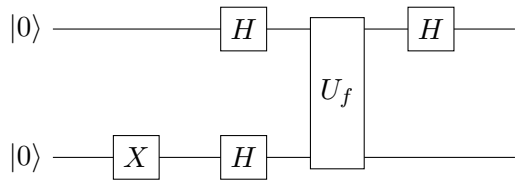

Figure 5.1: *Circuit diagram of Deutsch's algorithm.*

Another way to think about the problem of Deutsch's algorithm is in determining if the function f is balanced or constant. A function $f : \{0,1\}^n \to \{0,1\}$ is considered balanced if exactly half of its inputs map to 0 and the other half to 1. Given a function $f : \{0,1\}^n \to \{0,1\}$ that is either balanced or constant, the *Deutsch-Josza algorithm* determines which it is. Deutsch's algorithm is therefore a special case of Deutsch-Josza algorithm when $n = 1$ [11].

5.2 Grover's Quantum Search Algorithm

Grover's quantum search algorithm is often described as "finding a needle in a haystack." It is a remarkable algorithm in that it demonstrates a quantum computer's ability to find a particular item, the needle, in a large list of items, the haystack, with a quadratic speed up compared to any classical algorithm. It is often used to illustrate the potential speed up offered by quantum computers. Much like Deutsch's algorithm, however, it is more of a theoretical than practical exercise.

Consider a large, unsorted list containing N items, such as a randomly ordered phone directory with N names, where $N = 2^n$ for some integer $n \geq 0$. To determine the location (i.e. index) of a specific phone number in the list, a classical algorithm would take $O(N)$ iterations. By harnessing the power of superposition, however, Grover's algorithm can perform this task in $O(\sqrt{N})$ iterations [14]. Here we will explore Grover's algorithm and discuss its optimal time complexity as described

in [3]. We first formalize the associated problem statement:

Problem: Given an unsorted list of $N = 2^n$ items, find the index of a specific item, w by repeatedly querying an oracle S_w.

With a classical mindset, all one can really do to approach this problem is to start at the beginning of the list and query each item until the solution state has been found. One could be lucky and find the solution on the first few queries, or be unlucky and not find it until the very last query. Assuming the list-index of the solution is uniformly distributed, in average we need $N/2$ i.e. $O(N)$ queries to determine said index, and there is no way to rig the problem to guarantee better performance than this.

In contrast, Grover's quantum algorithm exploits superposition to—in some sense—query every index in the list at once. It manipulates the amplitudes of each index to bias the probability of identifying the correct item. This is an important distinction with the classical approach. In fact, if we were to run the classical algorithm long enough, we would guarantee its success. The quantum querying in contrast can only achieve a high probability of success, but not guarantee it. In fact, running the quantum algorithm for too long can decrease its success probability, necessitating a bound for the optimal number of iterations [3].

5.2.1 Defining the Grover iterate

There are two steps to Grover's algorithm: an initialization step, and the repeated application of the Grover iterate, G. The initialization consists only of setting the system to a uniform superposition. This is achieved by initializing the system as $|\psi\rangle = H^{\otimes n}|0\rangle$. The Grover iterate is defined as $G = D\,S_w$, where D is the so-called *Diffusion transform* and S_w is called the *quantum oracle*.

The quantum oracle changes the phase of the solution state $|w\rangle$ as follows

$$S_w |x\rangle = \begin{cases} -|x\rangle & , \text{ if } |x\rangle = |w\rangle \\ |x\rangle & , \text{ else} \end{cases} \tag{5.3}$$

For now we will assume we are given some quantum circuit that encodes S_w as described above.

The Diffusion transform is often described as an *inversion about the mean.* For a generic state $|\psi\rangle = \sum_{i=0}^{N-1} \alpha_i |i\rangle$, the *average amplitude* is defined as $\alpha = \frac{1}{N} \sum_{i=0}^{N-1} \alpha_i$. Applying the Diffusion transform to $|\psi\rangle$ inverts each amplitude about this average. In other words, if α_i is some amount below α before the transform then it will be that amount above α afterwards. Similarly, if α_i is above α by some amount, it will be shifted to be that amount below α by the transform. Explicitly:

$$D\ |\psi\rangle := \sum_{i=1}^{N-1} (\alpha + (\alpha - \alpha_i)) |i\rangle \tag{5.4}$$

From Theorem 1 in [14], $D = H^{\otimes n} \cdot S_0 \cdot H^{\otimes n}$. In particular

$$(S_0)_{ij} = \begin{cases} (-1) & , \ i = j = 0 \\ 1 & , \ i = j \neq 0 \\ 0 & , \text{ else} \end{cases} \tag{5.5}$$

Notice that S_0 is a real symmetric matrix, that is its own inverse. Thus it is clear D is unitary and is a valid quantum operator. Further notice that both D and S_w are real matrices, so the Grover iterate does not introduce complex amplitudes.

Let us consider the first few steps of Grover's algorithm to understand at a high-level how it works. Given $|\psi_0\rangle = H^{\otimes n} |0\rangle$ is in a uniform superposition, we first apply the oracle S_w. This will mark the solution state $|w\rangle$ by multiplying its amplitude by (-1). If we assume n is fairly large then the average amplitude, α, will still be fairly close to $1/\sqrt{N}$. After an application of the Diffusion transform the amplitude of the solution state $|w\rangle$ will be $1/\sqrt{N} + (1/\sqrt{N} + 1/\sqrt{N}) = 3/\sqrt{N}$. The amplitudes of all the non-solution states will approximately be $1/\sqrt{N} + (1/\sqrt{N} - 1/\sqrt{N}) = 1/\sqrt{N}$. Consequently, the Diffusion transform is amplifying the amplitude of the solution state, while relatively decreasing the amplitudes of all the other possible states. Figure 5.2 demonstrates this concept in the initial steps of Grover's Algorithm.

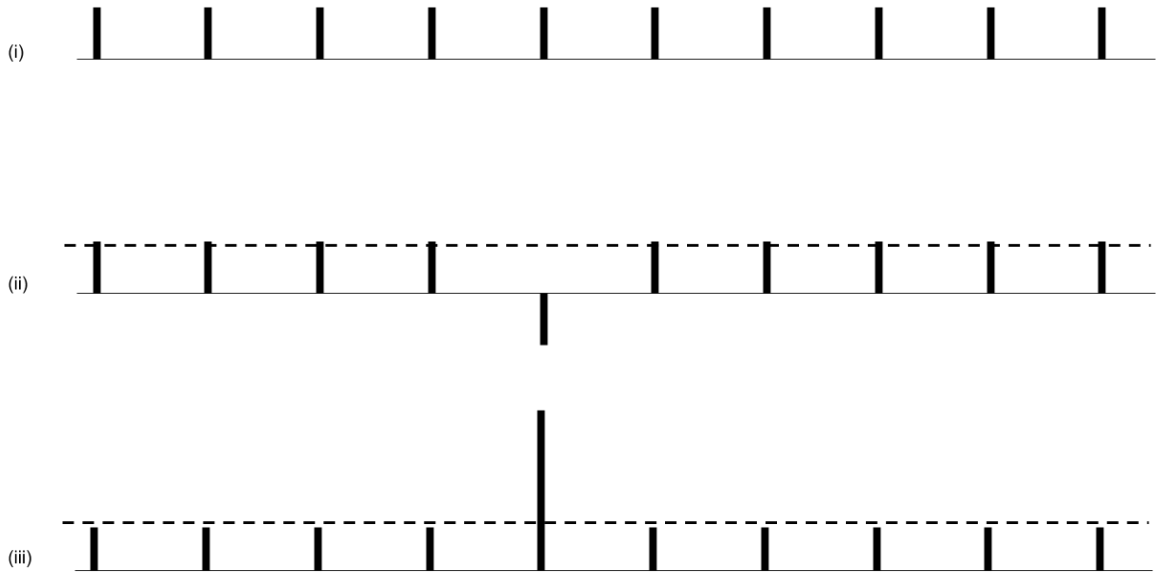

Figure 5.2: *Initial Steps of Grover's Algorithm.* (i) Initial uniform superposition of all states. (ii) Amplitudes after the solution state has been marked by the oracle. The dotted line represents the average amplitude, α. (iii) Amplitudes after the Diffusion transform. The solution amplitude is amplified. Notice that as all the amplitudes in the initial state are real, and as the Grover iterate is a real matrix, all amplitudes will remain real throughout the iteration process.

5.2.2 Finding the Optimal Number of Iterations

We can develop a recursion describing the amplitude changes for each iteration. Let $k, l \in \mathbb{R}$ such that $k^2 + (N-1)l^2 = 1$. From the above approximation we can see we can represent the state of our quantum system as $|\psi(k,l)\rangle = k\,|w\rangle + \sum_{i \neq w} l\,|i\rangle$. The oracle changes the above state to $|\psi(k,l)\rangle = -k\,|w\rangle + \sum_{i \neq w} l\,|i\rangle$. Hence the average amplitude of $|\psi\rangle$ before the application of the Diffusion transformation is

$$\alpha = \frac{(N-1)l - k}{N} \tag{5.6}$$

If we let the amplitude of the solution state $|w\rangle$ after the j-th application of the Diffusion transform be described by k_j, and similarly for the non solution states with l_j, we find from equation (5.4)

that

$$k_{j+1} = 2\alpha - k_j = \frac{2(N-1)l_j - 2k_j}{N} + \frac{Nk_j}{N} \tag{5.7}$$

$$l_{j+1} = 2\alpha - l_j = \frac{2(N-1)l_j - 2k_j}{N} - \frac{Nl_j}{N} \tag{5.8}$$

Thus we can describe the following recursion for the amplitudes:

$$\begin{aligned} l_0 = k_0 &= \frac{1}{\sqrt{N}} \\ k_{j+1} &= \frac{2(N-1)}{N} l_j + \frac{N-2}{N} k_j \\ l_{j+1} &= \frac{(N-2)}{N} l_j - \frac{2}{N} k_j \end{aligned} \tag{5.9}$$

The above can be represented as a closed form expression:

$$\begin{aligned} k_j &= \sin((2j+1)\theta) \\ l_j &= \frac{1}{\sqrt{N-1}} \cos((2j+1)\theta) \end{aligned} \tag{5.10}$$

where θ satisfies $\sin^2(\theta) = 1/N$ [3].

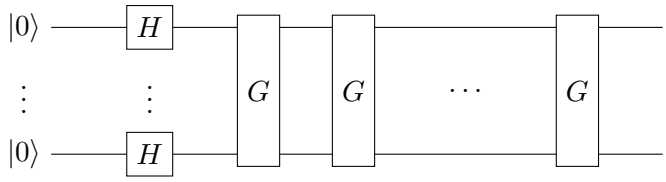

Figure 5.3: *Quantum Circuit for Grover's Quantum Search Algorithm.*

It is now clear why applying the Grover iterate too many times can in fact harm the probability of identifying the correct solution, as the closed form expression in equation (5.10) is periodic. To identify the optimal number of iterations we want $\sin((2j+1)\theta) = 1$, which occurs when $j = \frac{\pi - 2\theta}{4\theta}$. However j must be an integer, but error is bounded by $1/N$ when $j = \lfloor \pi/(4\theta) \rfloor$[3]. When N is large, it holds that $1/\sqrt{N} = \sin(\theta) \approx \theta$, so an appropriate number of applications of the Grover iterate is

$$j = \left\lfloor \frac{\pi}{4} \sqrt{N} \right\rfloor \tag{5.11}$$

In fact because of the floor argument, the approximation of $\theta \approx 1/\sqrt{N}$ does not affect the value of j for any $N = 2^n$ with $n \geq 0$ an integer. We now have all the pieces in place to formalize Grover's Quantum Search as seen in Algorithm 1.

Initialize $|\psi_0\rangle = H^{\otimes n} |0\rangle$ **for** $j = 1, \ldots, \left\lfloor \frac{\pi}{4}\sqrt{N} \right\rfloor$ **do**
  Apply the Grover iterate, G, defined as $G = H^{\otimes n} \cdot S_0 \cdot H^{\otimes n} \cdot S_w$ such that $|\psi_j\rangle = G\,|\psi_{j-1}\rangle$
Measure $|\psi\rangle$.
Query resulting state with oracle.
if Solution state then
  Exit
else
  Start From Beginning

Algorithm 1: *Pseudocode for Grover's Algorithm.*

Chapter 6

Programming IBM's Public Quantum Computer

With the IBM Quantum Experience [1], for the first time quantum computing is available to the general public. Using an intuitive web-app, called the Quantum Composer, users can design their own quantum circuits and use credits to run them remotely on a physical 5-qubit computer. Further, to aid in the design of algorithms users can also simulate their circuits using real or theoretical constraints, which we will refer to as real or theoretical quantum topology. With real quantum topology users must consider the physical limitations and constraints of IBM's machine, which we will discuss further in a later section, while with the theoretical topology users can simulate up to 20 qubits to test and explore quantum concepts without considering physical constraints. Users can earn credits to use the physical computer by completing tutorials and taking part in the quantum community on the website.

In this section we will explore the IBM Quantum Experience. We will begin by discussing the quantum gates available in the Composer and their implications to the algorithms users can develop. We will then review the physical constraints of IBM's machine as well as discussing the options available for simulating quantum algorithms. We finally discuss how to implement both Deutsch's and Grover's algorithms. Due to limited access to the quantum computer, we were only able to run a select number of tests of these algorithms.

6.1 Quantum Gates and Measurement

The IBM Quantum Experience includes 10 built-in, basic quantum gates. These are the Pauli Gates (I, X, Y and Z), the Hadamard gate (H), two twist gates and their inverses (S and $S^\dagger$, as well as T and $T^\dagger$) [1] , and the controlled-not gate ($\wedge_1(X)$). The twist gates are defined as

$$S = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}, \quad S^\dagger = \begin{bmatrix} 1 & 0 \\ 0 & -i \end{bmatrix}, \quad T = \begin{bmatrix} 1 & 0 \\ 0 & e^{\frac{i\pi}{4}} \end{bmatrix}, \quad T^\dagger = \begin{bmatrix} 1 & 0 \\ 0 & e^{\frac{-i\pi}{4}} \end{bmatrix} \tag{6.1}$$

As we saw in Section 4.5 the set of gates $\{H, T, \wedge_1(X)\}$ is universal (i.e. we can approximate with arbitrary precision any quantum gate, using only gates from this set). In particular, we have at our disposal more gates than we minimally need, as can be seen explicitly in Table 6.1.

Gate	**Composition from** $\{T, H\}$
$T^\dagger$	T^7
S	T^2
$S^\dagger$	T^6
Z	$S^2 = T^4$
X	$HZH = HT^4H$
Y	$ZX = T^4HT^4H$

Table 6.1: *Representation of IBM's Quantum Composer Gates.* Factorization of all the single-qubit gates from the Quantum Composer in terms of the twist gate T and the Hadamard gate H.

While it might seem redundant to include all these gates, it is in fact quite useful to have definitions of them in the composer. One reason for this is a higher degree of code readability. Another is the so-called *quantum decoherence*, or the difficulty in maintaining a qubit state over time. This makes time an extremely valuable resource in a physical quantum computer. In fact, if each gate takes some number of unit time operations to evaluate, clearly having an explicit definition of a common gate such as the NOT gate, X, is much more advantageous than composing $X = HT^4H$, which would take 6 quantum operations to evaluate.

In actuality the IBM quantum computer uses three advanced gates, sometimes described as the physical gates, $U_1(\lambda)$, $U_2(\phi, \lambda)$ and $U_3(\theta, \phi, \lambda)$, that form a universal set of single-qubit gates.

[1] Note here $U^\dagger$ is the standard notation of physicists for the conjugate transpose of U, and is used in the Quantum Composer. We will use it interchangeably with U^*.

We will focus primarily on just the basic gates here, but more information on these physical gates can be found in [7]. Accordingly, will define an operation as one matrix of dimension $2^n \times 2^n$ composed of basic gates using just the Kronecker product. For example, with two qubits $(H \otimes H)$ would represent one operation whereas $(H \otimes H) \cdot (X \otimes X)$ would represent two. In this sense we are counting the number of vertically aligned gates on a quantum circuit, i.e. the number of columns in a quantum circuit. Thus, we define the number of operations when comparing different gates or algorithms in this way. Each of the physical gates has a different time evaluation cost and one could similarly discuss the number of unit time operations needed to run a gate or algorithm from the specifications of [7]. We will not address time computations in this manuscript, however, and only focus on the number of operations as just defined.

The Quantum Composer also includes a measurement gate that collapses a qubit to a basis state. This is the last operation that can be performed on a qubit in the Quantum Composer.

In both real and theoretical topology users must select the number of shots (i.e. trials) for their quantum algorithm. In either case, results are returned both as a bar graph, representing the statistics of the experiment, or the number of trials that yield each possible result, and the theoretical state of the quantum system as a point on the Bloch sphere before measurement. Thus for both real and theoretical topology the same Bloch sphere representation is generated, which can be a useful tool to understand the bar graph results. With theoretical topology, results come directly from simulation and match what one would expect if they were constructing the operators directly via linear algebra. With real topology, physical constraints of the machine are at play and the results are much more prone to errors. We discuss more about the latter in the next section.

6.2 Physical Constraints

To understand the physical constraints of a quantum computer, it is necessary to have a feeling for how it operates, in particular knowing the type of qubits that it uses. It is important to realize that, while the theory of quantum computing is developed for any type of qubit, independently of its physical nature, in a real quantum computer those properties dictate our ability to maintain

and manipulate the states of the qubits. So while the physical details of qubits are irrelevant to the theory of quantum computing, once we move to a real machine we must understand the limitations of the type of qubit we are using in the context of our algorithms. The IBM machine uses a type of charge qubit, meaning the basis state $|0\rangle$ corresponds to some sort of unexcited or ground state whereas the state $|1\rangle$ is associated with an excited or charged state. Qubits that are in a superposition are thus in some sense simultaneously charged and uncharged, or equivalently simultaneously $|0\rangle$ and $|1\rangle$, as usual. For more details about the qubits used in the IBM quantum computer see [16].

6.2.1 Maintaining Qubit States

IBM's quantum computer has five qubits, labelled q_0, q_1, q_2, q_3, q_4, that each have physical constraints pertaining to their loss of information of a quantum state over time, or *quantum decoherence*. There are two main machine-dependent time constants associated with decoherence for each qubit: T_1, which is associated with energy relaxation, and T_2, which is associated with dephasing. Current research in the development of quantum computers focuses on improving these constants to allow longer, more complex algorithms, to be implemented with greater consistency and accuracy.

Energy relaxation describes the tendency for a qubit in the excited state, $|1\rangle$, to fall to the unexcited state, $|0\rangle$. We have described the quantum gate I as the state preserving gate; applications of I to a state $|\psi\rangle$ do not change its state. While theoretically this is true, using the physical quantum computer and a series of I gates we can see that over time the excited state, $|1\rangle$, will relax to the ground state $|0\rangle$. The time constant T_1 describes this process for each individual qubit as each qubit has its own rate of energy relaxation. When implementing algorithms on the physical machine this should be taken into consideration as certain qubits will lose information at much faster rates than others.

Due to energy relaxation there is a built in limit to the number of quantum gates one can use per qubit in a quantum score. In real topology this cap is 80 operations per qubit, which is

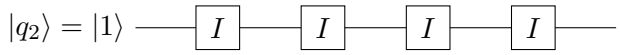

Figure 6.1: *Energy Relaxation Circuit.* Theoretically, the measurement $|q_2\rangle = |1\rangle$ after passing through this simple circuit should result in $|1\rangle$ with probability 1. The effect of running this circuit on the real machine can be seen in Table 6.2 and demonstrates that time will be a significant constraint on the physical quantum computer.

Number of I gates after Excitement	Percentage of Maintained States
4	95.1%
16	90.1%
78	44.2%

Table 6.2: *Energy Relaxation Results.* Results from running quantum circuit of Figure 6.1, with varying numbers of I gates on qubit q_2. The maximum number of state preserving gates that can be used on an excited qubit is 78 (the overall cap is 80 and it requires one gate to excite the qubit to $|1\rangle$ and one gate to perform a measurement). This circuit is the most trivial quantum circuit and demonstrates the significant restrictions energy relaxation imposes on quantum algorithms. Note running the circuit of Figure 6.1 on each qubit would generate different results as they each have their own value of T_1 that describes their energy relaxation rate.

sufficient for many of the algorithms that can be implemented with 5-qubits though, as we will see with Grover's algorithm, this does pose practical limitations.

Though superposition is a key concept in quantum computing, it is very difficult to maintain superposition states. In fact, any interaction a qubit has with its surrounding environment may act as a measurement and hence collapse the qubit to a basis state (as if the environment had "observed" the qubit). When these premature measurements occur quantum states are no longer represented as superpositions and are instead mixed states. As we saw in Section 4.6, mixed states behave quite differently than superposition states and, in particular, quantum interference will no longer occur, fundamentally changing what we observe under measurement. Dephasing, and its associated time constant T_2, is related to this process and also relies on the energy relaxation rate from T_1. Thus T_2 describes the ability to maintain qubit superposition states without incidentally inducing a measurement and collapsing the system to a mixed state.

6.2.2 Entangling Qubits

Another constraint associated with programming under real topology is a limitation in the entangling gate, $\wedge_1(X)$. In particular, only q_1, q_2, q_4 can be the target qubits of $\wedge_1(X)$. There are further restrictions on which qubits can be the control qubit depending on which is the target, as outlined in Figure 6.2.

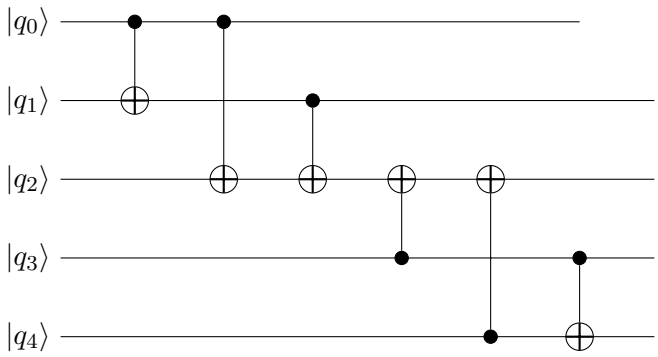

Figure 6.2: *Physical Entanglement Restrictions.* Quantum circuit representations of all possible implementations of $\wedge_1(X)$ on the physical IBM quantum computer.

Many algorithms require entanglement between multiple pairs of qubits so this limitation must be addressed. Here we introduce two quantum gates that we can use to work around this entanglement restriction: a reversed CNOT and a SWAP gate. We can easily construct both from the gates of the Quantum Composer.

The reversed CNOT gate allows us to switch the target and control qubits of an ordinary CNOT gate, see circuit on the left hand-side of Figure 6.3. This gate is necessary only because of IBM's CNOT restrictions. For example, in real topology, it allows q_2 to be the control qubit and one of q_0, q_1, q_3, q_4 to be the target.

The reversed CNOT gate maps $|01\rangle \rightarrow |11\rangle$ and $|11\rangle \rightarrow |01\rangle$, but keeps $|00\rangle$ and $|10\rangle$ un-

changed. In particular, its matrix form representation is:

$$\begin{array}{c} \begin{matrix} |00\rangle & |01\rangle & |10\rangle & |11\rangle \end{matrix} \\ \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \end{array} \begin{matrix} |00\rangle \\ |01\rangle \\ |10\rangle \\ |11\rangle \end{matrix} = (H \otimes H) \cdot \wedge_1(X) \cdot (H \otimes H) \tag{6.2}$$

where the last identity is a straightforward exercise. Consequently, we may simulate the controlled CNOT gate using the circuit in Figure 6.3.

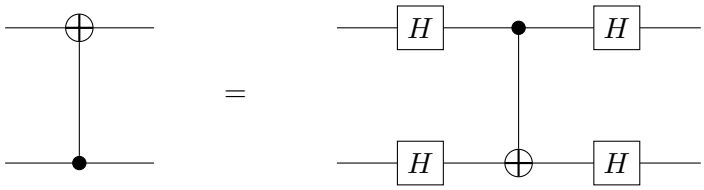

Figure 6.3: *Quantum Circuit for Reverse CNOT.* Quantum to circuit to switch the target and control qubits of a standard CNOT gate. Its cost is 3 quantum operations.

The SWAP gate switches the states of two qubits and can be visualized as crossing two qubit lines in a quantum circuit, as seen on the left hand-side of Figure 6.4. This gate relies on the reversed CNOT gate and can be used to entangle any of the qubits in real topology. For example, if we wanted to have q_0 be the target qubit of a CNOT operation we could swap it with q_2, apply CNOT, and swap back. Thus we can entangle any two qubits in real topology with at most 5 quantum operations.

The SWAP gate maps $|01\rangle \to |10\rangle$ and $|10\rangle \to |01\rangle$, while keeping $|00\rangle$ and $|11\rangle$ invariant. It

is then easy to verify that the following identities apply:

$$\text{SWAP} = \begin{array}{c} \begin{matrix} |00\rangle & |01\rangle & |10\rangle & |11\rangle \end{matrix} \\ \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{array} \begin{matrix} |00\rangle \\ |01\rangle \\ |10\rangle \\ |11\rangle \end{matrix} = \wedge_1(X) \cdot (H \otimes H) \cdot \wedge_1(X) \cdot (H \otimes H) \cdot \wedge_1(X) \quad (6.3)$$

In particular, since $(H \otimes H) \cdot \wedge_1(X) \cdot (H \otimes H)$ is equivalent to the reverse CNOT gate, the circuit in Figure 6.4 can be used to simulate SWAP.

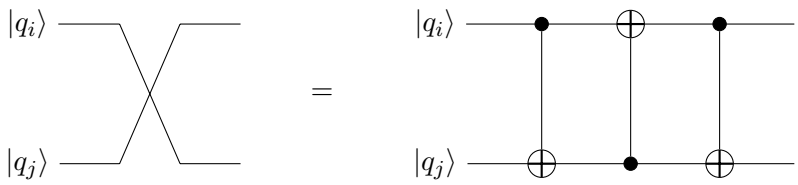

Figure 6.4: *Quantum Circuit for SWAP Gate.* Quantum circuit to switch two qubit lines, or swap the states of two qubits. With the cost of the reversed CNOT being 3 operations, it takes a total of 5 operations to swap two qubit states. This a key gate that allows any qubit to be the target of the entangling gate CNOT.

6.3 Implementation of Deutsch's Algorithm

In this section we will discuss how to implement Deutsch's algorithm on the IBM Quantum Experience. Recall the problem statement of Deutsch's algorithm:

Problem: Given $f : \{0, 1\} \to \{0, 1\}$, determine if f is a constant function with the least number of evaluations of the function f.

As a reminder, classically, this could only be solved by querying the function twice and comparing the values of $f(0)$ and $f(1)$. With Deutsch's algorithm, however, we can solve this problem in one query to the quantum representation of f i.e. the gate U_f—see equation (4.20).

Recall also that Deutsch's algorithm can be represented by the quantum circuit of Figure 5.1. In particular, beyond the unknown gate U_f, all the gates necessary for Deutsch's algorithm are already built into the Quantum Composer. To program and test Deutsch's algorithm, we must therefore design the gate U_f. While in practice, we would only receive U_f as a quantum black box i.e. some quantum circuit that we are given without knowing its function, here we create U_f ourselves and test the algorithm. Deutsch's algorithm is a simple algorithm in that we can consider all the possible outcomes and verify by brute force that it is behaving correctly. For our function $f : \{0, 1\} \to \{0, 1\}$ there are only four possibilities to consider. These are

$$\begin{aligned} f(x) &\equiv 0, \quad f(x) \equiv 1 \\ f(x) &= x, \quad f(x) = \neg x \end{aligned} \tag{6.4}$$

Thus we only have to consider four possible quantum gates U_f.

We first will consider the function $f(x) \equiv 0$. In this case, $U_f |x\rangle |y\rangle = |x\rangle |y\rangle$; in particular, $U_f = I_4 = (I_2 \otimes I_2)$ and we see U_f is separable. (Here I_d denotes the identity gate of dimensions $(d \times d)$.) On the other hand, for the function $f(x) \equiv 1$, $U_f |x\rangle |y\rangle = |x\rangle |\neg y\rangle$ i.e. $U_f = (I \otimes X)$, which is again separable.

We now consider the case of $f(x) = x$. In this case, $U_f |x\rangle |y\rangle = |x\rangle |x \oplus y\rangle$ i.e. $U_f = \wedge_1(X)$; in particular, and unlike the previous cases, U_f is now an entrangling gate.

We finally consider the function $f(x) = \neg x$. Since $(\neg x) \oplus y = x \oplus 1 \oplus y = x \oplus \neg y$, we see that $U_f |x\rangle |y\rangle = |x\rangle |x \oplus \neg y\rangle$ i.e. $U_f = \wedge_1(X)(I \otimes X)$ as we are performing a controlled-not gate on a flipped value of $|y\rangle$. The gate U_f is thus again entangling.

In summary, all the gates necessary to program and test Deutsch's algorithm are directly available in the IBM Quantum Experience. Deutsch's algorithm only requires entanglment to occur between two qubits in one direction (i.e. one qubit is solely the control and the other solely the target) so we can easily program it for the real quantum computer by using q_1 as the control qubit and q_2 as the target qubit without having to work around the limitations of entanglement. Thus the last step of Deutsch's algorithm is to perform a measurement on q_1.

Theoretically, it is easy to verify that for constant functions f this will yield $|0\rangle$ and for non-constant functions $|1\rangle$ with certainy. Table 6.3 shows the results of running Deutsch's algorithm over 1024 trials with $f(x) = x$. Clearly the accuracy of the results suffer due to quantum decoherency as, in total there are five quantum operations applied to each qubit. When compared to the results of the trivial circuit from Table 6.2 it is clear that both energy relaxation and dephasing are contributing to the errors of quantum decoherence.

Measurement Result	Theoretical Success Rate	Empirical Success Rate
$\|0\rangle$	0%	13.0%
$\|1\rangle$	100%	87.0%

Table 6.3: *Deutsch's Algorithm Results.* Success rates over 1024 trials of the control qubit for Deutsch's algorithm when $f(x) = x$.

6.4 Implementation of Grover's Algorithm

In this section we will discuss implementing Grover's algorithm on the IBM Quantum Experience. Recall the problem statement of the Grover's algorithm.

Problem: Given an unsorted list of $N = 2^n$ items, find the index of a specific item, w by repeatedly querying an oracle S_w.

As discussed in a previous section, the key component of Grover's Quantum Search Algorithm is the Grover iterate, G, defined as $G = H^{\otimes n} \cdot S_0 \cdot H^{\otimes n} \cdot S_w$, where S_w is the quantum oracle that marks the solution state's phase by rotating it by π radians, and S_0 performs a phase rotation by π radians on the state $|0\rangle$. These gates are considerably more complex than those of Deutsch's algorithm, and we will need to construct them from the gates of the Quantum Experience.

From equation (5.5), we know that S_0 is a diagonal matrix of all ones except for the first entry of (-1). This immediately is reminiscent of the Z gate. In fact, if we add controls to the first $(n-1)$ qubits to apply a Z gate to the n-th qubit, i.e. consider the operator $\wedge_{n-1}(Z)$, we get

a flipped version of S_0: a diagonal matrix of all ones except for a last entry of (-1). We can create a permutation matrix that allows us to transform $\wedge_{n-1}(Z)$ into S_0 through $X^{\otimes n}$. Further, one can prove inductively for $i, j = 0, 1, \ldots, (2^n - 1)$ that

$$(X^{\otimes n})_{ij} = \begin{cases} 1 & , \quad \text{if } (i+j) = (2^n - 1) \\ 0 & , \quad \text{else} \end{cases}$$

In particular, $X^{\otimes n}$ has the matrix form:

$$X^{\otimes n} = \begin{bmatrix} & & & & 1 \\ & & & 1 & \\ & & \iddots & & \\ & 1 & & & \\ 1 & & & & \end{bmatrix}$$

Applying this matrix to the left of a gate vertically flips all the rows, and from the right horizontally flips all the columns. Thus

$$S_0 = X^{\otimes n} \cdot \wedge_{n-1}(Z) \cdot X^{\otimes n} \tag{6.5}$$

By noticing that $S_w = \wedge_{n-1}(Z)$, when $|w\rangle = |2^n - 1\rangle$, we see that the controlled-Z gate will be key in the construction of S_w as well. Further, as stated above, $S_w = X^{\otimes n} \cdot \wedge_{n-1}(Z) \cdot X^{\otimes n}$ when $|w\rangle = |0^n\rangle$. This leads to the intuition that we can relate the gates X and I to the values 0 and 1 in bit strings, to create permutation matrices that allow us to construct any S_w from the controlled-Z gate, $\wedge_{n-1}(Z)$. In this relation X would be associated with 0, and I with 1. As an example, if $n = 4$ and the solution state is $|w\rangle = |0110\rangle$, then $S_w = (X \otimes I \otimes I \otimes X) \cdot \wedge_3(Z) \cdot (X \otimes I \otimes I \otimes X)$.

To see this explicitly, note that we are trying to create a permutation matrix, P, that maps $|2^n - 1\rangle \to |w\rangle$. Let $|w\rangle = |x_0, x_1, \ldots, x_{n-1}\rangle$ where $x_i \in \{0, 1\}$ for $i = 0, \ldots, (n-1)$. Consider constructing P such that $P = A_0 \otimes A_1 \otimes \ldots \otimes A_{n-1}$, where $A_i \in \{I, X\}$ and $P|w\rangle = |2^n - 1\rangle$. Then

$$\begin{aligned} |2^n - 1\rangle &= P|w\rangle \\ &= (A_0 \otimes A_1 \otimes \ldots \otimes A_{n-1})(|x_0\rangle \otimes |x_1\rangle \otimes \ldots \otimes |x_{n-1}\rangle) \\ &= A_0|x_0\rangle \otimes A_1|x_1\rangle \otimes \ldots \otimes A_{n-1}|x_{n-1}\rangle \end{aligned} \tag{6.6}$$

In the above expression, $A_i |x_i\rangle = |1\rangle$ must hold for all $i = 0, \ldots, (n-1)$. Thus, if $|x_i\rangle = |0\rangle$ then $A_i = X$, and if $|x_i\rangle = |1\rangle$ then $A_i = I$. As $X^2 = I$, then $A_i^2 = I$ for all i, and hence P is its own inverse. Thus $P|2^n - 1\rangle = |w\rangle$ and we have successfully created a permutation matrix such that $S_w = P \cdot \wedge_{n-1}(Z) \cdot P$.

The resulting quantum circuit for the Grover iterate is displayed in Figure 6.5. The key component to the phase shift gates necessary for the Grover iterate is the controlled-Z gate, $\wedge_n(Z)$. We have seen how to add controls to arbitrary gates in Section (4.8). Here we will go into the details of implementing controlled-Z gates for different values of n, the number of qubits at our disposal.

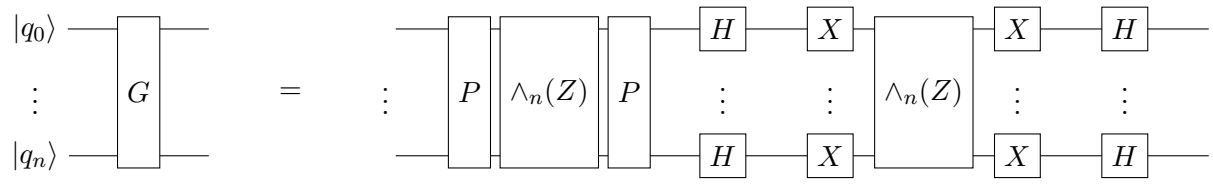

Figure 6.5: *Grover's iterate for* $(n+1)$ *qubits*

Note that we do not need to consider the trivial case of $n = 1$ as, in this case, a single query to the oracle would suffice to identify the solution.

6.4.1 Controlled-Z Gates when $n = 2$

We have already seen how to implement the gate $\wedge_1(Z)$ in the quantum circuit of Figure 4.8. We can implement $\wedge_1(Z)$ in the Quantum Composer with just three gates. Furthermore, the permutation gates, P, associated with each solution state, $|w\rangle$, are as given in Table 6.4. The

$\|w\rangle$	P
$\|00\rangle$	$X \otimes X$
$\|01\rangle$	$X \otimes I$
$\|10\rangle$	$I \otimes X$
$\|11\rangle$	$I \otimes I$

Table 6.4: *Permutation Gates for Quantum Oracle when* $n = 2$

permutation matrix P will always only require one quantum operation per qubit. In total the Grover iterate requires two controlled-Z gates, two permutation matrices, and two applications of X and H respectively, for a total cost of 12 quantum operations when $n = 2$. Further, if we use q_1 as the control qubit and q_2 as the target, we will have no entanglement restrictions. Using equation (5.11) we find that just one application of the Grover iterate is optimal. The initialization step of Grover's involves setting the quantum system in a uniform superposition via the Hadamard gate. So, in total 13 quantum operations plus two measurements (one for each qubit) are required for Grover's algorithm when $n = 2$.

Table 6.5 shows that we get fairly consistent results when running the algorithm on the physical machine, regardless of the solution state. Notice that from equation (5.10), the amplitude of the solution state will be $k_1 = \sin(3\theta)$ where $\theta = \sin^{-1}(1/2) = \pi/6$. Thus $k_1 = 1$, so theoretically Grover's returns the correct solution state with certainty when $n = 2$. The physical quantum computer of course does not achieve this level of accuracy but it does still perform fairly well.

Solution State	Theoretical Success Rate	Empirical Success Rate
$\|00\rangle$	100%	88.8%
$\|01\rangle$	100%	83.5%
$\|10\rangle$	100%	87.6%
$\|11\rangle$	100%	85.3%

Table 6.5: *Grover's Algorithm Results for $n = 2$.* Success rates for Grover's Quantum Search Algorithm over 1024 trials when $n = 2$ for the four possible solution states.

6.4.2 Controlled-Z Gates when $n = 3$

To implement the controlled-Z gate, $\wedge_2(Z)$, we need a matrix U such that $U^2 = Z$, see Figure 4.12. A moments thought, however, reveals that the phase gate S defined in equation (6.1) satisfies this requirement. Thus, we need to implement a controlled-S gate, $\wedge_1(S)$.

As $S \notin \mathrm{SU}(2)$, we must construct two gates, $V = \mathrm{Ph}(\delta)$ and $W \in \mathrm{SU}(2)$, such that $S = VW$, see Figure 4.9. We expect the T gates to play a role in the construction of W, as the T gate

performs half the phase twist that an S gate does. Consider

$$W = TXT^{\dagger}X = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix} \begin{bmatrix} e^{-i\pi/4} & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} e^{-i\pi/4} & 0 \\ 0 & e^{i\pi/4} \end{bmatrix} \tag{6.7}$$

It is clearly easy to implement a controlled version of this gate, $\wedge_1(W)$, by replacing the X gates with $\wedge_1(X)$. Further, we can quickly deduce that the phase gate, $V = \mathrm{Ph}(\pi/4)$ would satisfy the requirement that $S = VW$. From Figure 4.10 we can implement $\wedge_1(V)$ with the gate

$$\begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix} \tag{6.8}$$

acting on the control qubit. The above is clearly the T gate, so we have all the gates we need to implement $\wedge_1(S)$ using the Quantum Composer.

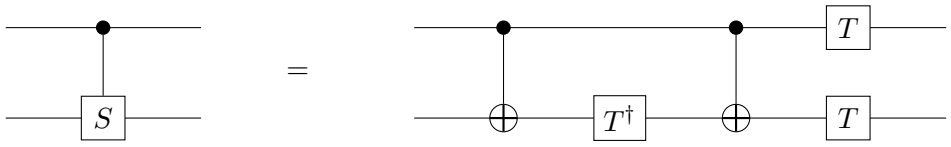

Figure 6.6: *Quantum Circuit of Controlled-S Gate*, $\wedge_1(S)$.

Figure 6.6 shows the quantum circuit for the $\wedge_1(S)$ gate. It requires four quantum operations. From Figure 4.8, it is clear we will need to apply $\wedge_1(S)$ twice and $\wedge_1(S^{\dagger})$ once, along with two additional applications of $\wedge_1(X)$ to perform $\wedge_1(Z)$. Thus, it will take us 14 quantum operations to perform $\wedge_1(Z)$. However, when expanding the $\wedge_1(S)$ and $\wedge_1(S^{\dagger})$ gates, several operators will cancel out and in fact we can implement $\wedge_1(Z)$ with just 12 operations. Thus each Grover iterate will require 30 operations. If qubits q_0 and q_1 are used as the controls and q_2 as the target, we again do not have to consider IBM's entanglement restrictions.

From equation (5.11) we find the optimal number of iterations is two, however, we must also consider the implications of how many operations are required by the Grover iterate in this case. Running Grover's with the theoretical optimal number of iterations would require a total of 61 operations plus three measurement gates. As we saw from the trivial algorithm in Table 6.2,

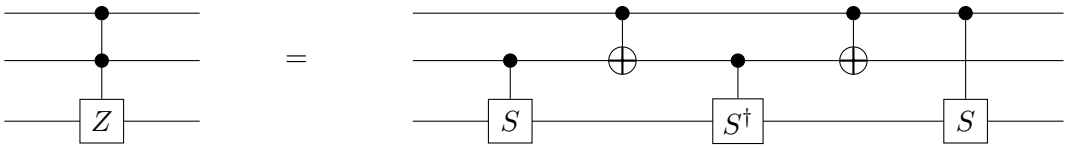

Figure 6.7: *Quantum Circuit for Controlled-Z gate,* $\wedge_2(Z)$.

we cannot expect to maintain significant accuracy over such a high number of operations. From equation (5.10), we can find the theoretical probability of success with two iterations is 94.5% and with one iteration it is 78.1%. Thus we have to balance these theoretical results with the physical constraints of the quantum computer.

Table 6.6 shows the errors associated with Grover's algorithm with three qubits and two Grover iterates for two solution states, $|010\rangle$ and $|111\rangle$. Due to limited access to the quantum computer, more trials could not be run for this test case. Energy relaxation appears to play a role in the loss of accuracy as when the solution state was $|111\rangle$, the state $|000\rangle$ was returned more often.

Two Iterations	$\|000\rangle$	$\|001\rangle$	$\|010\rangle$	$\|011\rangle$	$\|100\rangle$	$\|101\rangle$	$\|110\rangle$	$\|111\rangle$
$\|010\rangle$	13.2%	12.2%	**14.8%**	11.7%	11.7%	11.1%	13.6%	11.6%
$\|111\rangle$	15.0%	12.6%	12.9%	11.1%	10.3%	12.9%	11.9%	**13.3%**

Table 6.6: *Grover's Success and Error Rates with Two Iterations.* Success of Grover's Quantum Search algorithm with 3 qubits and two Grover iterates over 1024 trials. The first column denotes the actual target solution, and the subsequent columns show the percentage of trials they were returned as the solution under measurement. Bold items denote the success rate of the algorithm.

In Table 6.7 we see the errors for Grover's with three qubits and one Grover iterate. It is clear that these results performed better than two iterates, but were still fairly inaccurate. Again it seems energy relaxation causes a slight bias in the state $|000\rangle$ being returned. Even the limited 31 operations needed for Grover's with one iterate caused a very large amount of error, and shows that the main constraint of physical quantum computers today is the maintenance of qubit information, rather than just the limited number of qubits at our disposal.

One Iteration	$\|000\rangle$	$\|001\rangle$	$\|010\rangle$	$\|011\rangle$	$\|100\rangle$	$\|101\rangle$	$\|110\rangle$	$\|111\rangle$
$\|000\rangle$	**24.9%**	9.2%	12.9%	10.5%	13.5%	7.9%	11.0%	10.1%
$\|001\rangle$	12.6%	**17.1%**	11.2%	14.1%	10.4%	12.7%	10.1%	11.8%
$\|010\rangle$	13.9%	10.3%	**20.4%**	11.8%	12.8%	9.1%	13.1%	8.6%
$\|011\rangle$	13.0%	15.5%	9.8%	**18.0%**	7.5%	10.7%	12.3%	13.2%
$\|100\rangle$	15.3%	10.1%	15.1%	10.0%	**18.7%**	8.0%	14.3%	8.6%
$\|101\rangle$	13.9%	12.4%	10.8%	13.9%	11.2%	**15.7%**	11.1%	10.8%
$\|110\rangle$	16.0%	11.0%	15.2%	11.2%	13.6%	8.5%	**16.0%**	8.4%
$\|111\rangle$	12.6%	12.0%	10.5%	14.2%	9.8%	14.1%	9.0%	**17.9%**

Table 6.7: *Grover's Error Matrix with One Iteration.* Success of Grover's Quantum Search algorithm with 3 qubits and one Grover iterate over 1024 trials. The first column denotes the actual target solution, and the subsequent columns show the percentage of trials they were returned as the solution under measurement. Bold items denote the success rate of the algorithm.

6.4.3 Controlled-Z Gates for Arbitrary n

To implement Grover's algorithm with $n = 4$ qubits on the physical computer, one would quickly encounter a variety of issues. First, the $\wedge_3(Z)$ gate requires a $\wedge_1(T)$ gate, which is very difficult to make with the basic gates provided, and requires the physical gate $U_1(\pi/8)$. With this gate one can implement $\wedge_1(T)$ or $\wedge_1(T^\dagger)$ with four operations. It is clear from Figure 4.13 that we will need seven of these gates and 6 additional $\wedge_1(X)$ gates, for a total of 34 gates per controlled-Z gate. Thus, as a Grover iterate uses two controlled-Z gates and six additional gates, we need 74 quantum operations for a single Grover iterate. Therefore, with the initialization step and the required four measurement operators, one could simulate Grover's with 79 operations, one short of the max allowed. Unfortunately, due to IBM's current entanglement restrictions, we would need to take advantage of SWAP gates to implement this on the physical machine, so it is not possible to run Grover's with 4 qubits on the physical machine. Further one Grover iterate with 4 qubits, where $N = 2^4$, would be the largest list size one could simulate with IBM's current quantum computer capabilities.

6.5 Conclusions from the Physical Quantum Computer

In this section we discussed how to program IBM's physical quantum computer and implemented two quantum algorithms on it. We first laid out the quantum gates available on the quantum computer, and confirmed we had a universal set. We then explored the physical limitations of the machine and saw that errors due to quantum decoherence become significant as the scope of our algorithms increases. We finally implemented Deutsch's and Grover's algorithms with 2 qubits and saw reasonable success rates. However, we also saw that algorithms that utilize several quantum operations such as Grover's with 3 qubits, suffer from significant decoherence error. This implied that it was worth exploring simpler implementations of quantum algorithms that do not achieve the theoretical optimality, but that do perform better in practice.

Bibliography

[1] IBM Quantum Experience. https://quantumexperience.ng.bluemix.net/qstage/#/user-guide. Accessed: 2017-04-01.

[2] A. Barenco, C. H. Bennett, R. Cleve, D. P. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. A. Smolin, and H. Weinfurter. Elementary gates for quantum computation. Physical Review A, 52(5):3457, 1995.

[3] M. Boyer, G. Brassard, P. Hoyer, and A. Tapp. Tight bounds on quantum searching. Fortschritte der Physik, 46(4-5):493–505, 1998.

[4] K-A. Brickman, PC. Haljan, PJ. Lee, M. Acton, L. Deslauriers, and C. Monroe. Implementation of grovers quantum search algorithm in a scalable system. Physical Review A, 72(5):050306, 2005.

[5] J. Chiaverini, J. Britton, D. Leibfried, E. Knill, M. D. Barrett, RB. Blakestad, W. M. Itano, J. D. Jost, C. Langer, R. Ozeri, et al. Implementation of the semiclassical quantum Fourier transform in a scalable system. Science, 308(5724):997–1000, 2005.

[6] W. Christian. QM Spins Program, May 2008.

[7] A. W. Cross, L. S. Bishop, J. A. Smolin, and J. M. Gambetta. Open quantum assembly language. https://github.com/IBM/qiskit-openqasm/tree/master/spec, 2017.

[8] S. Debnath, NM. Linke, C. Figgatt, KA. Landsman, K. Wright, and C. Monroe. Demonstration of a small programmable quantum computer with atomic qubits. Nature, 536(7614):63–66, 2016.

[9] D. Deutsch. Quantum theory, the church-turing principle and the universal quantum computer. Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences, 400(1818):97–117, 1985.

[10] D. Deutsch, A. Barenco, and A. Ekert. Universality in quantum computation. Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences, 449(1937):669–677, 1995.

[11] D. Deutsch and R. Jozsa. Rapid solution of problems by quantum computation. Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences, 439(1907):553–558, 1992.

[12] R. P. Feynman. Simulating physics with computers. International Journal of Theoretical Physics, 21(6):467–488, 1982.

[13] A. Franklin and S. Perovic. Experiment in physics. In E. N. Zalta, editor, The Stanford Encyclopedia of Philosophy. Summer 2015 edition, 2015. http://plato.stanford.edu/archives/sum2015/entries/physics-experiment/.

[14] L. K. Grover. A fast quantum mechanical algorithm for database search. In Proceedings of the twenty-eighth Annual ACM Symposium on Theory of Computing, pages 212–219. ACM, 1996.

[15] S. Gulde, M. Riebe, G. PT. Lancaster, C. Becher, J. Eschner, H. Häffner, F. Schmidt-Kaler, I. L. Chuang, and R. Blatt. Implementation of the Deutsch-Jozsa algorithm on an ion-trap quantum computer. Nature, 421(6918):48–50, 2003.

[16] P. Kaye, R. Laflamme, and M. Mosca. An Introduction to Quantum Computing. Oxford University Press, 2007.

[17] J. Koch, T. M. Yu, J. Gambetta, A. A. Houck, D. I. Schuster, J. Majer, A. Blais, M. H. Devoret, S. M. Girvin, and R. J. Schoelkopf. Charge-insensitive qubit design derived from the cooper pair box. Physical Review A, 76:042319, Oct 2007.

[18] N. Linden, H. Barjat, and R. Freeman. An implementation of the DeutschJozsa algorithm on a three-qubit NMR quantum computer. Chemical Physics Letters, 296(12):61 – 67, 1998.

[19] D. H. McIntyre, C. A. Manogue, and J. Tate. Quantum Mechanics: A Paradigms Approach. Pearson, 2012.

[20] T. Monz, D. Nigg, E. A. Martinez, M. F. Brandl, P. Schindler, R. Rines, S. X. Wang, I. L. Chuang, and R. Blatt. Realization of a scalable Shor algorithm. Science, 351(6277):1068–1070, 2016.

[21] P. W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM Journal on Computing, 26(5):1484–26, 10 1997.

[22] P. W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM Review, 41(2):303–332, 1999.

[23] L. MK. Vandersypen, M. Steffen, G. Breyta, C. S. Yannoni, M. H. Sherwood, and I. L. Chuang. Experimental realization of Shor's quantum factoring algorithm using nuclear magnetic resonance. Nature, 414(6866):883–887, 2001.