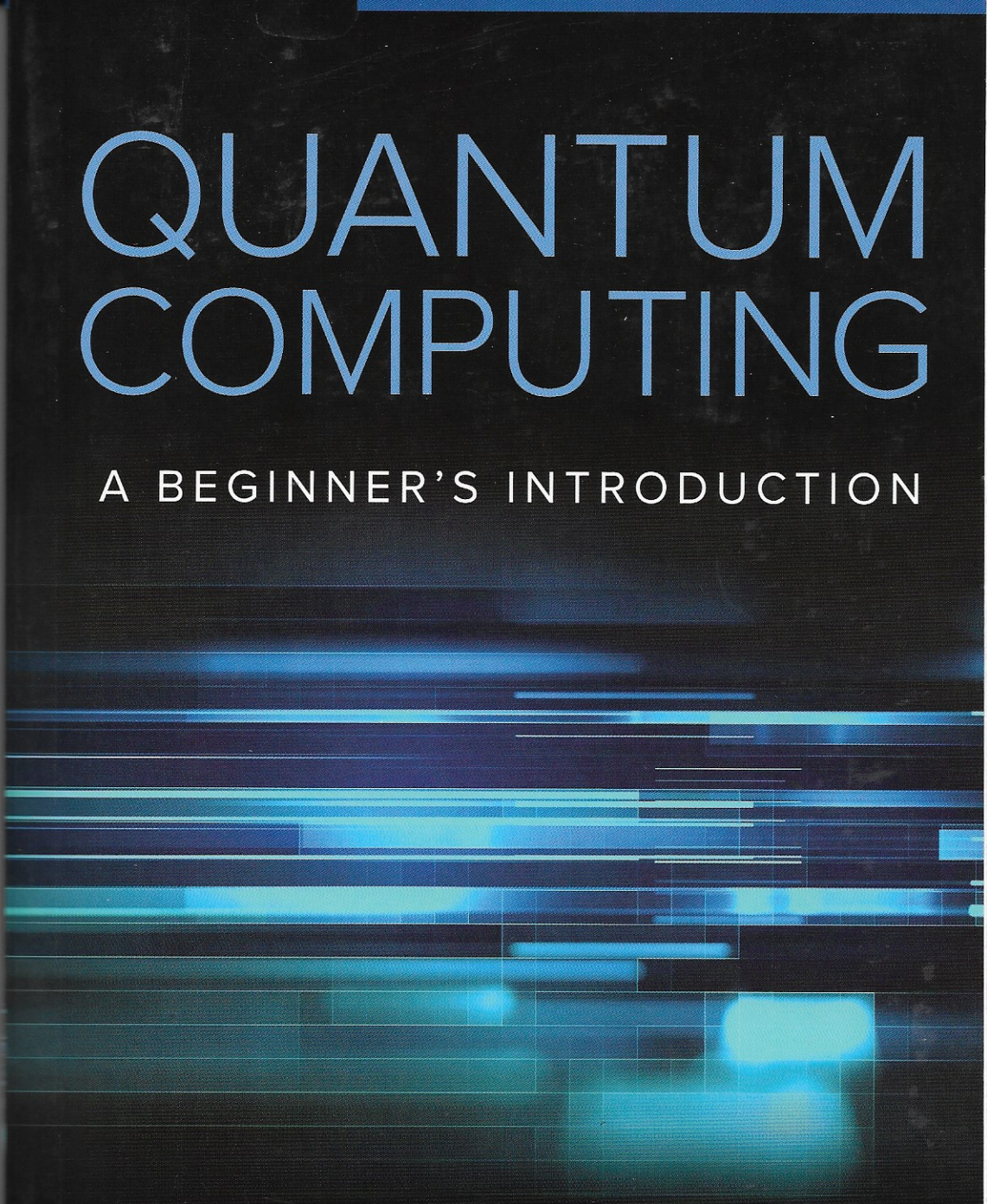
QUANTUM COMPUTING

A BEGINNER'S INTRODUCTION



PARAG K. LALA

Mc Graw Hill Education

Quantum Computing

A Beginner's Introduction

Parag K. Lala

New York Chicago San Francisco
Athens London Madrid
Mexico City Milan New Delhi
Singapore Sydney Toronto

Library of Congress Control Number: 2018951445

McGraw-Hill Education books are available at special quantity discounts to use as premiums and sales promotions, or for use in corporate training programs. To contact a representative please visit the Contact Us page at www.mhprofessional.com.

Quantum Computing: A Beginner's Introduction

Copyright ©2019 by McGraw-Hill Education. All rights reserved. Printed in the United States of America. Except as permitted under the United States Copyright Act of 1976, no part of this publication may be reproduced or distributed in any form or by any means, or stored in a data base or retrieval system, without the prior written permission of the publisher.

1 2 3 4 5 6 7 8 9 LCR 23 22 21 20 19

ISBN 978-1-260-12311-1
MHID 1-260-12311-1

The pages within this book were printed on acid-free paper.

Sponsoring Editor
Lara Zoble

Editing Supervisor
Donna M. Martone

Acquisitions Coordinator
Elizabeth Houde

Project Manager
Dolly Sarangthem,
Cenveo® Publisher Services

Copy Editor
Cenveo Publisher Services

Proofreader
Cenveo Publisher Services

Indexer
Robert Swanson

Production Supervisor
Lynn M. Messina

Composition
Cenveo Publisher Services

Art Director, Cover
Jeff Weeks

Information contained in this work has been obtained by McGraw-Hill Education from sources believed to be reliable. However, neither McGraw-Hill Education nor its authors guarantee the accuracy or completeness of any information published herein, and neither McGraw-Hill Education nor its authors shall be responsible for any errors, omissions, or damages arising out of use of this information. This work is published with the understanding that McGraw-Hill Education and its authors are supplying information but are not attempting to render engineering or other professional services. If such services are required, the assistance of an appropriate professional should be sought.

Contents

Preface

Quantum computing is based on the principles of quantum mechanics, which provides a description of the behavior of very small, atomic and subatomic, particles. Due to the way these particles behave, operations in a quantum computer can be done much faster than in traditional computers. In the past two decades, quantum computing has developed into a major area of research for physicists, computer scientists, and electrical engineers.

The book is aimed to be a "gentle introduction" to quantum computing. It presents the concepts and workings of quantum computing systems in a way that does not require readers to have technical knowledge beyond senior level in a typical degree program in electrical engineering, computer science, or physics.

The primary objective of the book is to present the material in a way that is readable and easy to understand without getting overwhelmed with the mathematical details. It is meant to be self-contained; all the necessary prerequisite material is included at appropriate places in the text.

The book is divided into ten chapters.

Chapter 1 provides a refresher coverage of complex numbers and vectors. It also introduces Dirac's bra and ket notation that is widely used in quantum mechanics to represent a quantum state.

Chapter 2 discusses development of quantum mechanics and deals with the behavior of matter at the atomic and subatomic levels.

Chapter 3 provides a comprehensive coverage of matrices and operators. Operators are extensively used in quantum computing; they act on quantum states and change them. In quantum computing, all operators are of linear type and are represented by matrices.

Chapter 4 presents the basics of Boolean algebra and conventional logic gates. It also discusses in detail the principles of quantum information processing and introduces *qubits*. Just as a bit is the basic unit of information in a classical computer, a qubit is the basic unit of information in a quantum computer.

Chapter 5 presents a detailed survey of quantum gates. Quantum gates are mathematically represented as transformation matrices. The operation of both single-qubit and 2-qubit gates is presented in detail.

Chapter 6 discusses two intriguing aspects of quantum particles—superposition and entanglement; both are used in quantum computing. Some familiarity with tensor products is needed to understand the concepts of superposition and entanglement. This chapter begins with a short introduction to tensor products.

Two other unique features of quantum information—teleportation and superdense coding—are discussed in Chap. 7. Teleportation is the ability to transmit quantum data by sending only classical bits. Superdense coding on the other hand is used to transfer two classical bits by sending only one qubit.

A major problem in quantum computing systems is that the quantum bits get corrupted whenever they interact with their environment. However, as in its classical counterpart it is possible to detect and correct errors in qubits by using error correcting codes. Chapter 8 discusses the possible types of errors in quantum systems and the techniques employed for correcting such errors.

Chapter 9 discusses one of the primary reasons why quantum computing has attracted so much attention. It can manipulate quantum information using mathematical operations borrowed from linear algebra. This has led to the development of quantum algorithms that can perform certain operations such as database searching, factoring large integers in a fraction of a time; it takes a conventional computer to execute these operations. This chapter discusses some of the well-known quantum algorithms.

The last chapter, Chap. 10, begins with a discussion of classic cryptographic systems and various techniques for data encryption. Next, the principle of quantum cryptography that uses intrinsic quantum properties of photons (particles of light) to encode data is presented. Several important techniques for quantum key distribution protocols are also discussed.

I owe gratitude to my family for the encouragement and support I have been given during this project. My wife Meena was the main source of support in this endeavor from the start to the finish. She went through various drafts of the manuscript, suggesting ways to make the material more reader-friendly and correcting mistakes. My daughter Nupur and son Kunal also helped out. I am particularly indebted to Nupur who basically copyedited the entire manuscript during the break prior to starting her medical residency program. I am also thankful to Ms. Dolly Sarangthem, the production manger of the book for her co-operation and for her patience. Finally I must express my sincere thanks to my former colleague and a dear friend Dr. Ugur Tanriver for many interesting discussions I have had with him. He also helped me in many other ways for which I remain grateful.

CHAPTER 1

Complex Numbers, Vector Space, and Dirac Notation

1.1 Complex Numbers

An understanding of quantum computing requires some knowledge of the properties of complex numbers. This section is written with the assumption that the reader has come across complex numbers in the past, know that they are composed of two types of fundamental numbers: real and imaginary, and that they can be solution to quadratic equations.

A *complex number* c is represented as

$$c = a + bi$$

where a and b are real numbers and $i = \sqrt{-1}$

The numbers a and b represent the real and the imaginary part of the complex number:

a = the real part of $c = \text{Re}(a + bi)$

b = the imaginary part of $c = \text{Im}(a + bi)$

For example, if $c = 6 + 4i$, then $a = 6$ and $b = 4$.

It should be clear from the above definition that the real numbers and the imaginary numbers are subsets of the complex numbers. If $\text{Re}(a + bi) = 0$, then c is a pure imaginary number, and if $\text{Im}(a + bi) = 0$, then c is a real number.

A complex number has two real coordinates, namely, its real and imaginary parts [1]. Therefore, a complex number can be plotted as a point in a complex plane using the real part of the complex number as the x-coordinate of the point, and the imaginary part b as the y-coordinate (Fig. 1.1). The horizontal axis of the complex plane is called the *real axis* and the vertical axis is called the *imaginary* axis.

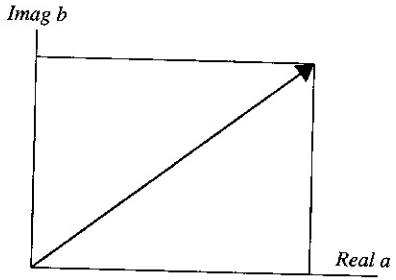


Figure 1.1 Complex number plane.

As indicated above there are two special cases: $b = 0$ or $a = 0$. In the special case when $b = 0$,

$$c = a + 0i = a$$

Thus every real number is also a complex number.
When $a = 0$, the complex numbers are called *pure imaginary* numbers. Note that

$$0 = 0 + 0i$$

Therefore, 0 is *both* a real number and a pure imaginary number.

It is clear from Fig. 1.1, that the *magnitude* (or *length*) of the complex number can be derived by using the Pythagorean theorem

$$|c| = \sqrt{a^2 + b^2}$$

Thus, the magnitude of the complex number is the square root of the sum of the squares of the real and imaginary parts of the complex number.

Complex numbers can be added and multiplied as the conventional real numbers keeping in mind that $i^2 = -1$. The rules for these operations are presented below:
Addition:

$$(a + ib) + (c + id) = (a + c) + i\,(b + d)$$

Multiplication:

$$(a + ib) \times (c + id) = (ac - bd) + i\,(ad + bc)$$

Division:

$$\frac{a + ib}{c + id} = \frac{a + ib}{c + id} \times \frac{(c + id)^*}{(c + id)^*}$$

[multiplying the numerator and the denominator by $(c + id)^*$]

$$= \frac{a + ib}{c + id} \times \frac{c - id}{c - id}$$

$$= \frac{(ac + bd) + i(bc - ad)}{(c^2 + d^2)}$$

1.2 Complex Conjugation

The operation of complex conjugation has a very important part in the theory of complex numbers. The complex conjugate of c above is obtained by replacing i with $-i$. Thus the complex conjugate of c, denoted as c^*, is

$$c^* = a - ib$$

As shown in Fig. 1.2, it is a reflection about the real axis.

The complex conjugate of algebraic expressions can be derived using the following two relations:

$$(a + b)^* = a^* + b^*$$

$$(ab)^* = a^* b^*$$

The *modulus* or *absolute value*, of a complex number $c = a + ib$, is denoted by $|c|$ and is the distance from point c to the origin in the complex plane; equivalently, $|c|$ is the *length* of the vector corresponding to c.

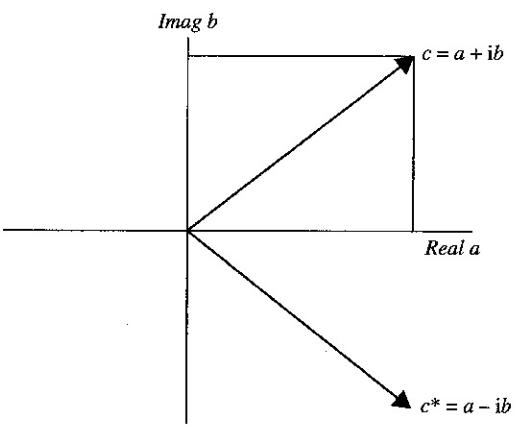


Figure 1.2 Complex conjugate.

The *square modulus* of c denoted by $|c|^2$, is the product of c with its complex conjugate c^*; it can be shown that $cc^* = |c|^2$ is a real number and is positive

$$\begin{aligned} cc^* &= (a + ib)(a - ib) \\ &= a^2 + b^2 \\ &= (\sqrt{a^2 + b^2})(\sqrt{a^2 + b^2}) \\ &= |c|^2 \end{aligned}$$

The absolute value of a complex number, also known as its *magnitude*, is expressed as

$$|c| = \sqrt{cc^*} = \sqrt{a^2 + b^2}$$

Note that the square *modulus* of c is not the same as c^2. Since $c = a + ib$,

$$c^2 = (a^2 - b^2) + i(2ab)$$

The square modulus is always a positive real number whereas the square is, in general, a complex number.

Some of the properties of complex conjugates are

1. The conjugate of a product is a product of conjugates:

$$(c_1 c_2)^* = c_1^* c_2^*$$

2. The conjugate of a sum is the sum of conjugates:

$$(c_1 + c_2)^* = c_1^* + c_2^*$$

3. The conjugate of a conjugate is the complex number itself:

$$(c^*)^* = c$$

1.3 Vector Space

A vector space V is a collection of elements called *vectors*, a *field* F of elements called *scalars* and two operations, vector addition and scalar multiplication. A field is a set of scalars with the property that if a and b belong to F then $a + b$, $a - b$, ab, and a/b are also in F (assuming that $b \neq 0$ when a/b is derived).

Addition: The addition operation takes any two vectors $\mathbf{u}$ and $\mathbf{v}$ in V and produces a third vector in V, written as $\mathbf{u} + \mathbf{v}$ in V. The addition operation obeys the following conditions:

1. $\mathbf{u}+\mathbf{v}$ is a vector in V (*closure*)
2. $\mathbf{u}+\mathbf{v}=\mathbf{v}+\mathbf{u}$ (*commutativity*)
3. $(\mathbf{u}+\mathbf{v})+\mathbf{w}=\mathbf{u}+(\mathbf{v}+\mathbf{w})$ (*associativity*)
4. There is a zero vector 0 in V such that for every $\mathbf{u}$ in V, $(\mathbf{u}+0)=\mathbf{u}$ (*identity*)
5. For every $\mathbf{u}$ in V, there is a vector in V denoted by $-\mathbf{u}$ such that $\mathbf{u}+(-\mathbf{u})=0$ (*inverse*)

Multiplication: The scalar multiplication that takes a scalar c in F, a vector $\mathbf{v}$ in V and produces a new vector written as $c\mathbf{v}$ in V that satisfy the following conditions:

1. $c\mathbf{v}$ is in V (*closure*)
2. $c(\mathbf{u}+\mathbf{v})=c\mathbf{u}+c\mathbf{v}$ (*distributivity*)
3. $(c+d)\mathbf{u}=c\mathbf{u}+d\mathbf{u}$ (*distributivity*)
4. $c(d\mathbf{u})=(cd)\mathbf{u}$ (*associativity*)
5. $1(\mathbf{u})=\mathbf{u}$ (*identity*)

As an example consider two rows of complex numbers X and Y in a complex vector space C^n:

$$X=(x_1, x_2, \ldots\ldots, x_n)$$
$$Y=(y_1, y_2, \ldots\ldots, y_n)$$

The addition operation is done component-wise as shown below:

$$X+Y=(x_1+y_1)(x_2+y_2)\ldots\ldots\ldots, x_n+y_n)$$

The multiplication operation is done by multiplying each component of X (or Y) by a complex number c:

$$c\,(X)=(cx_1, cx_2, \ldots\ldots\ldots, cx_n)$$

It can be easily shown that all the conditions of the vector space are satisfied, and thus C^n is indeed a vector space.

1.4 Basis Set

In classical physics the position of a particle at any point in a three-dimensional space can be determined from the x-, y-, z-coordinates of the point; the three axes are orthogonal. In quantum mechanics there is a similar idea of expressing a *state vector* using a *basis set*. For a vector space, a basis set is a subset of vectors that enables every vector in the vector space to be represented as a unique linear combination of the vectors in this subset.

However, none of the vectors in the subset can be expressed as a linear combination of the remaining vectors. In general, an n-dimensional vector space will have a basis set of n distinct vectors. If the vectors in the basis are *unit vectors*, that is, each has a magnitude 1, and each is perpendicular to others then the basis is called an *orthonormal* basis. For example, the *unit vectors* **i**, **j**, and **k** constitute a orthonormal basis for normal three-dimensional vector space. The unit vectors **i**, **j**, and **k** are the standard unit vectors:

$$\mathbf{i} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \quad \mathbf{j} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \quad \mathbf{k} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

Any vector can be written as a linear combination of the standard unit vectors:

$$V = \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix}$$

$$= v_1 \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} + v_2 \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} + v_3 \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

$$= v_1\mathbf{i} + v_2\mathbf{j} + v_3\mathbf{k}$$

As indicated above, in a Cartesian three-dimensional space a vector **v** is a set of three numbers, called *components* ($\mathbf{v}_x$, $\mathbf{v}_y$, $\mathbf{v}_z$). Any vector **v** in this space can be expanded in terms of the three unit vectors **i**, **j**, **k** (Fig. 1.3).

A function called an *inner product* enables the formation of numbers from vectors. The inner product of two vectors, **u** and **v**, in this space is defined as

$$\mathbf{u} \cdot \mathbf{v} = \mathbf{u}_x\mathbf{v}_x + \mathbf{u}_y\mathbf{v}_y + \mathbf{u}_z\mathbf{v}_z$$

The *length* also known as the *norm* of a vector **u** is $\sqrt{\mathbf{u} \cdot \mathbf{u}}$

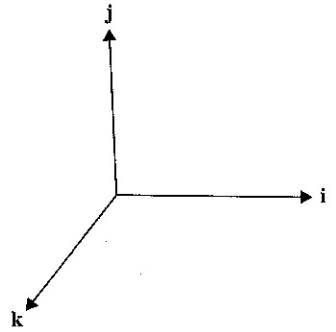


Figure 1.3 Unit vectors **i**, **j**, **k** directed along the positive x axis, y axis, and z axis, respectively.

In quantum computing the vectors are members of a *complex vector space*. A vector **v** in an *N*-dimensional complex vector space is stated as

$$\mathbf{v} \varepsilon C^N$$

A vector space in *N* dimensions would have *N* basis vectors and *N* components. A vector in this space can be represented by

$$|v> = \begin{pmatrix} v_0 \\ v_1 \\ . \\ . \\ v_{N-1} \end{pmatrix}$$

The *inner product* of two vectors **u** and **v** in the complex space, denoted $< u\ v >$, is a function that takes u and v as inputs and produces a complex number as the output. Assuming the column representation of vectors **u** and **v** as

$$\mathbf{u} = \begin{pmatrix} u_0 \\ u_1 \\ : \\ : \\ u_{N-1} \end{pmatrix} \quad \text{and} \quad \mathbf{v} = \begin{pmatrix} v_0 \\ v_1 \\ : \\ : \\ v_{N-1} \end{pmatrix}$$

the inner product is

$$< uv > = \begin{pmatrix} u_0 \\ u_1 \\ : \\ : \\ u_{N-1} \end{pmatrix} \bullet \begin{pmatrix} v_0 \\ v_1 \\ : \\ : \\ v_{N-1} \end{pmatrix} = u^* v = (u_0^*, u_1^*, \ldots \ldots u_{N-1}^*) \begin{pmatrix} v_0 \\ v_1 \\ : \\ : \\ vN-1 \end{pmatrix}$$

$$= \sum_{i=0}^{N-1} u_i^* v_i$$

where u^* $(u_0^*, u_1^*, \ldots \ldots u_{N-1}^*)$ is the *complex conjugate* of u.

As an illustration, the inner product of quantum states w_1 and w_2 represented by two column vectors in the complex vector space is derived by taking the complex conjugate of

$$w_1 = \begin{pmatrix} 3+i \\ 4-i \end{pmatrix} \quad \text{and} \quad w_2 = \begin{pmatrix} 3i \\ 4 \end{pmatrix}$$

the first vector w_1 and then multiplying the corresponding components of two vectors, and finally summing these products:

$$< w_1 w_2 > = < w_1{}^* w_2 >$$
$$= \begin{pmatrix} 3-i \\ 4+i \end{pmatrix} \begin{pmatrix} 3i \\ 4 \end{pmatrix}$$
$$= 19 + 13i$$

Since the length (*norm*) of a vector is derived by taking the square root of the inner product of the vector with itself, the length of the vector

$$\begin{pmatrix} 1-i \\ 2 \end{pmatrix}$$

is

$$\sqrt{< \begin{pmatrix} 1-i \\ 2 \end{pmatrix} \begin{pmatrix} 1+i \\ 2 \end{pmatrix} >}$$
$$= \sqrt{1+1+4} = \sqrt{6}$$

1.5 Dirac Notation

In quantum mechanics, a different notation called *Dirac notation* is used to represent quantum states. In this notation the inner product of two vectors **u** and **v** is denoted by $<\mathbf{u}\,|\,\mathbf{v}>$; the left part $<\mathbf{u}\,|$ is called the *bra* and the right part $|\,\mathbf{v}>$ the *ket*. Thus in the Dirac notation, also known as the *bra-ket notation*, an inner product is denoted by a $<>$ (*braket*).

1.5.1 Ket

A vector **v** in a complex vector space V is denoted by a *ket*

$$|\,\mathbf{v}> \varepsilon\, V$$

A ket is analogous to a column vector. Since the vector space is complex, all items in a column vector are complex numbers. For example, with N basis vectors $|\,i>$ where $i = 0, 1, \ldots \ldots N-1$, any vector $|\,v>$ can be written as

$$|\,v> = \sum_{i=0}^{N-1} c_i\,|\,i>$$

where c_i is an arbitrary set of complex numbers. It can also be arranged as a column vector of N complex numbers

$$|\,v> = \begin{pmatrix} 0 \\ 1 \\ . \\ . \\ N-1 \end{pmatrix}$$

The addition of two kets gives another ket. The addition of kets is commutative

$$(|\psi>+|\varphi>)=|\omega>=(|\varphi>+|\psi>)$$

and associative

$$(|\psi>+|\varphi>)+|\omega>=|\psi>+(|\varphi>+|\omega>)$$

Notice that the set of kets is *closed* under the addition operation.

Properties of Kets

$$(c_1+c_2)|u>=c_1|u>+c_2|u>$$
$$c_1(c_2|u>)=(c_1 c_2)|u>$$
$$c(|u>+|v>)=c|u>+c|v>$$
$$|1|u>=|u>$$
$$|u>+|0>=|u>$$

Since quantum computing deals exclusively with complex vector space, so scalar c is also assumed to be complex. The product of a complex number c and a ket $|u>$ can be written in either order,

$$c|u>=|u>c$$

A ket can also be written as a linear combination of other kets:

$$|v>=\sum_{i=0}^{N-1}|v_i>$$

that is

$$|v>=\begin{pmatrix} v_0 \\ v_1 \\ v_2 \\ . \\ . \\ v_{N-1} \end{pmatrix}$$

Note that there is a special vector in a vector space, the *null* or *zero vector*. The zero vector is written as 0, not $|0>$, so

$$|u>+0=|u>$$
$$0|u>=0$$

1.5.2 Bra

In linear algebra both column vectors and row vectors are used. The Dirac notation provides a symbol called a *bra* to represent a row vector. The bras corresponding to the kets $|0\rangle$ and $|1\rangle$ are

$$\langle 0| = (1\ 0)$$

$$\langle 1| = (0\ 1)$$

In a complex vector space V for every ket there is a unique bra; the bra corresponding to a ket is obtained by taking the conjugate transpose of the ket (and vice versa). A bra $\langle u|$ acting on a vector $\mathbf{v}$ turns it into a complex number c; this can be written as

$$\langle u| : \mathbf{v} \rightarrow c$$

For example, if $\mathbf{v}$ is written in the ket notation then

$$\langle u|(|\mathbf{v}\rangle) = c$$

or simply,

$$\langle u|\mathbf{v}\rangle = c$$

This indicates that $\langle u|$ can be combined with any ket vector $|\mathbf{v}\rangle$ to get the complex number c; in that case notation $\langle u|$ needs to be written as $(\langle u|)$; where any ket vector could be placed in the blank spot. The quantity $\langle u|$ can be considered as a function that assigns to every vector in V a complex number. Such a function is called a *linear functional*. Alternately, a bra vector can be interpreted as an operator that acts on a ket vector to produce a complex number [2, 3].

The bras corresponding to the kets $|1\rangle$ and $|0\rangle$ are the *conjugate transpose* of the kets:

$$\langle 1| = [0\ 1] \qquad \langle 0| = [1\ 0]$$

Bras are useful in calculating probability amplitudes. For example, bearing in mind the general state of a qubit is $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$, the probability of a qubit being in state $|1\rangle$ can be determined by combining the general qubit state with the bra $\langle 1|$ as below [5]:

$$\begin{aligned}\langle 1||\psi\rangle &= \langle 1|\psi\rangle \\ &= \langle 1|(\alpha\,|0\rangle + \beta\,|1\rangle) \\ &= \alpha\langle 1||0\rangle + \beta\langle 1||1\rangle \\ &= \alpha\,[0\ \ 1]\begin{bmatrix}1\\0\end{bmatrix} + \beta\,[0\ \ 1]\begin{bmatrix}0\\1\end{bmatrix}\end{aligned}$$

$$= \alpha \cdot 0 + \beta \cdot 1$$
$$= \beta$$

Note that the quantity $<1|\psi>$ can be interpreted as the probability amplitude for being in state $|1>$, and similarly $<0|\psi>$ is the probability amplitude for being in state $|0>$. It can be shown that $<0|1>$ and $<1|0>$ are both equal to zero and are known as *orthogonal* states. The amplitudes $<0|0>$ and $<1|1>$ are both equal to one. In general, for any quantum state $|\psi>$, the amplitude $<\psi|\psi> = 1$ as shown below:

Since $|\psi> = \alpha\ |0> + \beta\,|1>$, the dual or bra associated with it is

$$<\psi| = \alpha^*<0| + \beta^*<1|$$

Thus the amplitude $<\psi|\psi>$ can be calculated as

$$<\psi|\psi> = (\alpha^*<0| + \beta^*<1|)\,(\alpha\ |0> + \beta\,|1>$$
$$= \alpha^*\alpha<0|0> + \alpha^*\beta<0|1> + \beta^*\alpha<1|0> + \beta^*\beta<1|1>$$

Since $<0|1> = 0$ and $<1|0> = 0$, and both $<0|0>$ and $<1|1>$ are equal to 1.

$$<\psi|\psi> = |\alpha^2| + |\beta^2| = 1$$

The sum of two bras can be derived as below:

$$(<c_1| + <c_2|)\,|v> = <c_1|\,(|v>) + <c_2|\,(v>)$$

As an example, consider the following two kets:

$$|u> = \begin{pmatrix} p \\ q \end{pmatrix} \quad |v> = \begin{pmatrix} i \\ j \end{pmatrix}$$

Since a bra is the conjugate transpose of a ket, the bras corresponding to the kets above are

$$<u| = (p^*, q^*) \quad <v| = (i^*, j^*)$$

The sum of the two bras is

$$(p^* + i^*, q^* + j^*)$$

which is the conjugate transpose of the sum of the two kets:

$$|u> + |v> = \begin{pmatrix} p + i \\ q + j \end{pmatrix}$$

Bras and kets obey the following rules (a is some constant) [4]:

$$\langle v \mid au \rangle = a\langle v \mid u \rangle$$

$$\langle av \mid u \rangle = a^*\langle v \mid u \rangle$$

$$\langle v \mid u \rangle^* = \langle uv \rangle$$

$$\langle u \mid + \langle v \mid = \langle u + v \mid$$

Note that the order changes in which the bras and the kets are expressed based on how they are used as operators. For example, $\langle v \mid w \rangle \neq \langle w \mid v \rangle$.

Scalars, that is, complex numbers, however, can be moved through the expressions. For example,

$$\mid v \rangle\langle v \mid w \rangle\langle w \mid = \langle v \mid w \rangle \mid v \rangle\langle w \mid$$

since $\langle v \mid w \rangle$ is a complex number.

1.6 Inner Product

As stated earlier, the inner product of two vectors **u** and **v** in the complex space, denoted as $\langle \mathbf{uv} \rangle$, generates a complex number as the output.

The inner product must satisfy the following properties:

1. Linearity: $(\langle a \mid (w \mid b \rangle + v \mid c \rangle) = w\langle a \mid b \rangle + v\langle a \mid c \rangle$
2. Symmetry: $(u \mid v) = (v \mid u)^*$
3. Positivity: $\langle u \mid u \rangle \geq 0$ for $\mid u \rangle \neq 0$

The vector space V with an inner product is called an *inner product space*. In quantum mechanics a unit vector in a Hilbert Space is associated with a physical system. A *Hilbert Space* is basically a complex vector space with an inner product.

Since the inner product between two vectors results in a complex number, the Dirac notation for the inner product of the vectors $\mid u \rangle$ and $\mid v \rangle$ in a complex vector space is denoted by

$$\langle u \mid v \rangle = c$$

where c is a complex number. The term $\langle u \mid v \rangle$ is called a *bracket* and $\langle u \mid v \rangle = \langle v \mid u \rangle^*$

A vector $\mid u \rangle$ is *normalized* if

$$(\mid u \rangle, \mid u \rangle) = 1$$

Alternatively, if for two vectors

$$(\mid u \rangle, \mid v \rangle) = 0$$

then vectors ($\mid u \rangle$ and $\mid v \rangle$) are *orthogonal*.

1.7 Linearly Dependent and Independent Vectors

A set of vectors is *linearly independent* if a member of the set cannot be derived from the other members. More formally a set of vectors $v_0, v_1, v_2 \ldots \ldots \ldots, v_{n-1}$ is *linearly independent* if for any complex coefficients $c_0, c_1, c_2, \ldots \ldots \ldots, c_{n-1}$,

$$c_0v_0 + c_1v_1 + c_2v_2 \ldots \ldots \ldots + c_{n-1}v_{n-1} = 0$$

implies

$$c_0 = c_1 = c_2 = \ldots \ldots = c_{n-1} = 0$$

Conversely, the vectors $v_0, v_1, v_2, \ldots \ldots \ldots, v_{n-1}$ are *linearly dependent* if some of the complex coefficients $c_0, c_1, c_2, \ldots \ldots \ldots, c_{n-1}$ are not 0s, so that

$$c_0v_0 + c_1v_1 + c_2v_2 + \ldots \ldots \ldots + c_{n-1}v_{n-1} \neq 0$$

1.8 Dual Vector Space

The concept of a basis set was discussed earlier in Sec. 1.4. Formally, a set of vectors $\{v_0, v_1, \ldots \ldots, v_n\}$ is said to be a *basis* for a vector space if the set satisfies the following criteria:

1. The set of vectors $\{\mathbf{v}_0>, |\mathbf{v}_1> \ldots \ldots |\mathbf{v}_n>\}$ *spans* the vector space; that is, every state of the vector space can be represented by a linear combination of the states in the set. In other words, any state ψ of the vector space can be written as
$$\psi = \sum_i c_1 |v_i>$$
2. The set of vectors $\{\mathbf{v}_0, \mathbf{v}_1, \ldots \ldots, \mathbf{v}_n\}$ is linearly independent.
3. They are also *complete*. Completeness indicates that no additional basis are needed to describe any possible physical state of a quantum system. The vectors $\{|\mathbf{v}_0, \mathbf{v}_1, \ldots \ldots, \mathbf{v}_n\}$ are called the *base states* for the vector space.

Any arbitrary vector $|\mathbf{x}>$ in the complex vector space can be expressed uniquely as a linear combination of these basis vectors as

$$|\mathbf{x}> = \sum_{i=0}^{N-1} c_i |v_i>$$

where the complex numbers c_i are called the components of $|\mathbf{x}>$ with respect to the basis *vector* $\{\mathbf{v}_0, \mathbf{v}_1, \ldots \ldots, \mathbf{v}_n\}$

A set of vectors $\{\mathbf{e}_0 \ldots \ldots \mathbf{e}_n\}$ in V is *orthonormal* if

$$<\mathbf{e}_i | \mathbf{e}_j> = \delta_{ij}$$

where δ_{ij} is known as the *Kronecker* delta; it is equal to 1 if $i = j$ and zero if $i \neq j$ [5].

An orthonormal set is also a basis of V if it satisfies the *closure relation*

$$|x> = \sum_{i=0}^{N-1} <\mathbf{e}_i|\mathbf{e}_j> = 1$$

Earlier, in the Dirac notation of the inner product, the right side is just the *ket* notation of vector **y**. The quantity $<x|$ on the left side is a *linear functional* that maps any ket $|y>$ into the complex number generated by the inner product $(|x>, |y>)$.

The set of all linear functionals $(<x| \ldots \ldots)$ forms a complex vector space V^*, the dual space of V. In other words, a complex vector space V has a *dual vector space* V^* that contains the set of all bras. For example, for a set of ket vectors $(|\mathbf{v}_1>, |\mathbf{v}_2>, \ldots \ldots |\mathbf{v}_n>)$ in V, the dual space V^* consists of $(<\mathbf{v}_1|, <\mathbf{v}_2| \ldots \ldots \ldots \ldots <\mathbf{v}_n|)$. The sum of any two elements of V^* also belongs to V^*, and the product of a complex scalar with an element of V^* is also an element of V^*.

Note that V and V^* are not identical spaces, however for each ket vector $|\mathbf{y}>$ in V there is a corresponding bra vector $<\mathbf{y}|$ in V^*:

$$|\mathbf{y}> = (<\mathbf{y}|)^* <\mathbf{y}| = (|\mathbf{y}>)^*$$

The sum of two bras in the *dual space* V^* is also a bra in V^*, and the product of a complex number with a bra results in another bra in V^*.

The dimension of a vector space V and its dual V^* are equal. Vectors in V and V^* are represented by column and row matrices, respectively. Assume that $|\varphi>$ represent the following n-dimensional column vector

$$|\varphi> = \begin{pmatrix} \varphi_0 \\ \varphi_1 \\ . \\ . \\ . \\ \varphi_{N-1} \end{pmatrix}$$

then its dual vector denoted by $<\varphi|$ is defined by

$$<\varphi| = (\varphi_0^* \varphi_1^* \ldots \ldots \ldots \varphi_{N-1}^*)$$

In other words, if kets are represented as column vectors then bras are viewed as row vectors. Thus, a bra to the left of a ket in the inner product notation satisfies the matrix multiplication rule that a row vector times a column vector results in a number.

1.9 Computational Basis

Every vector space has an infinite number of orthonormal bases. In practice a basis set that is easy to work with is selected rather than any particular one; this basis is called the *computational basis*. Although the basis set does not need to be unique, the number of the base vectors contained in the set is invariant. For example, suppose the basis in complex vector space C^n has n linearly independent vectors; then any vector $|x>$ in C^n can be uniquely expressed as a linear combination of these n vectors

$$|x> = \sum_{i=0}^{n-1} c_i |v_i>, \quad c_i = \text{complex number}$$

where $|v_i>$ is the ith basis vector. The basis vectors are just column vectors with a single 1 at the location corresponding to a particular basis vector

$$|0> = \begin{pmatrix} 1 \\ 0 \\ 0 \\ . \\ . \\ 0 \end{pmatrix} \quad |1> = \begin{pmatrix} 0 \\ 1 \\ 0 \\ . \\ . \\ 0 \end{pmatrix} \quad |2> = \begin{pmatrix} 0 \\ 0 \\ 1 \\ . \\ . \\ 0 \end{pmatrix} \quad \ldots\ldots\ldots \quad |N-1> = \begin{pmatrix} 0 \\ 0 \\ 0 \\ . \\ . \\ 1 \end{pmatrix}$$

In quantum information processing states $|0>$ and $|1>$ for a qubit are used as the standard computational basis. But other bases are also employed. For example, the following vectors can be used as an orthonormal basis

$$\frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 1 \end{pmatrix}, \quad \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

Each of these vectors can be defined in terms of the standard computational basis; together they form an alternate basis +/.

$$\frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 0 \end{pmatrix} + \frac{1}{\sqrt{2}}\begin{pmatrix} 0 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{2}}(|0> + |1> = |+>$$

$$\frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ -1 \end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 0 \end{pmatrix} - \frac{1}{\sqrt{2}}\begin{pmatrix} 0 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{2}}(|0> - \frac{1}{\sqrt{2}}(|->) = |->$$

The +/− basis is a complete orthonormal basis. Thus any qubit state can be represented using the basis states +/−. The +/− basis is also known as the *Hadamard basis* or the *diagonal basis* [6].

1.10 Outer Product

The outer product between u and v is written as

$$|v><u|$$

Assuming $u = \begin{pmatrix} a_0 \\ a_1 \end{pmatrix} v = \begin{pmatrix} b_0 \\ b_1 \end{pmatrix}$

the outer product between u and v is

$$= \begin{pmatrix} b_0 \\ b_1 \end{pmatrix} (a_0^* \quad a_1^*)$$

$$= \begin{pmatrix} a_0^* b_0 & a_1^* b_0 \\ a_0^* b_1 & a_1^* b_1 \end{pmatrix}$$

The result is a matrix rather than a complex number [6]. The inner product between u and v is

$$= (a_0^* \quad a_1^*) \begin{pmatrix} b_0 \\ b_1 \end{pmatrix}$$

$$= a_0^* b_0 + a_1^* b_1$$

The outer product $|v><u|$ can be interpreted as an operator A. If $A = |v><u|$, the action of A on a state φ is

$$A\varphi = (|v><u|)\,\varphi = |v>(<u|\varphi>) = |v><u|\varphi>$$

References

1. Anthony A. Tovar, *Complex Numbers Review and Tutorial*, Eastern Oregon University, January 28, 2009.
2. Robert Littlejohn, Physics 221A Fall 2017 Notes 1—The Mathematical Formalism of Quantum Mechanics, Univ. of California, Berkley.
3. Dave Bacon, CSE 599d—Quantum Computing, lecture notes, University of Washington, Winter 2006.
4. Roman Koniuk, Quantum Mechanics (PHYS 4010) lecture notes, York University, 2011.
5. Mark M.Wilde, Quantum Information Processing Basics: Lecture I, *www.dias.ie/wp-content/uploads/2012/06/lecture-1-wilde.pdf.5* (June, 2012).
6. John Watrous, CPSC 519/619: Quantum Computation, University of Calgary, 2006.

CHAPTER 2

Basics of Quantum Mechanics

Classical physics represents the physics that was built on the twin foundations of Newton's laws that describe the motion of *macroscopic* (large) objects, and Maxwell's laws that describe the behavior of the electromagnetic radiation. Newton unveiled his laws of motion in 1686. Based on the work of Galileo and others, these laws describe with precision the behavior of objects when at rest, in motion, and when acted upon by external forces. There are three laws of motion. According to Newton's first law, an object moving in a straight line will continue to do so unless an external *force* is applied. At that point, the direction of the object and speed will change depending on the magnitude and the direction of the applied force. Alternatively, if an object is at rest, it will remain at rest unless an external force acts on it. This is often termed the *law of inertia.*

2.1 Limitations of Classical Physics

In classical physics, the state of a macroscopic object is described using its *position* and *momentum.* Newton's second law states that an acceleration is produced when the forces acting on an object are *unbalanced;* that is, a net force resulting from the combination of all the forces acts on the object. As the force acting upon the object is increased, the acceleration of the object increases, and as the mass of an object is increased, the acceleration of the object decreases. Thus the second law provides an exact relationship between force, mass, and acceleration:

$$Force = mass \times acceleration$$

So if the initial state of such an object at any time is known, it is possible to predict where it will go, that is, the position, and how fast it will, that is, the momentum by using Newton's second law of motion. In other words, Newton's second law allows determination of the dynamical evolution of an object over time, starting from an initial state.

Newton's third law states that for every action, there is an equal and opposite reaction. When two objects interact, they exert forces upon each other; these two forces are called *action* and *reaction* forces. The size of the force on the first object *equals* the size of the force on the second object, and the direction of the force on the first object is *opposite* to the direction of the force on the second.

Till the late 19th century, the laws of physics were based on mechanics, the law of gravitation from Newton, Maxwell's equations describing electricity and magnetism, and statistical mechanics describing the state of large collection of matter. These laws of physics were sufficient to describe nature well under most conditions. However, they do not apply to *microscopic*, that is, very small systems, such as individual atoms and the particles from which they are made because position and momentum are not appropriate variables to describe their state.

A number of fundamental difficulties at the levels of the atomic and subatomic were identified during the late 19th and early 20th centuries that could not be explained using the principles of classical physics [1]. Among these include *blackbody radiation*, *photoelectric effect*, and the *Rutherford-Bohr model of atom*.

2.1.1 Blackbody Radiation

Blackbody radiation is an example of a major contradiction between theory and experiment in classical physics. A *blackbody* is defined in classical physics as an object that absorbs all electromagnetic radiation that falls on it. This means that a blackbody does not reflect any radiation (hence it appears *black*) nor does it allow any radiation to pass through it. The abilities of a blackbody to absorb radiation and to reemit it are closely related to each other. A blackbody is capable of absorbing all wavelengths of electromagnetic radiation so it is also capable of emitting all wavelengths of electromagnetic radiation. Thus a blackbody is also an ideal source of radiation, known as *blackbody radiation*.

When a blackbody is cold, it does not emit any radiation. As it heats up, it starts emitting radiation. The wavelength of the emitted radiation depends only on the temperature of the blackbody, not its composition. The energy emitted per unit area is known as *intensity* of radiation. As indicated earlier, electromagnetic radiation results if there is any change in the movement of an electric charge. When a blackbody is heated, the electrons within it move in random directions, thus producing electromagnetic radiation. As a blackbody heats up, it moves through the spectrum becoming red, orange, yellow, green, and then blue, regardless of its composition.

At the beginning of the 20th century, two British scientists Raleigh and Jeans tried to analyze the spectrum of blackbody radiation.

They were primarily interested in determining how much of the emitted radiation is stored as blue light, how much as red light, and so on. They derived a formula that describes the intensity of radiation w as a function of frequency f for a fixed temperature T is

$$w(f, T) \propto f^2 \mathrm{T}$$

$$\propto \mathrm{T}/\lambda^2$$

At smaller frequencies (or longer wavelengths), the formula agrees with the experimental results showing radiation intensity decreasing. However, the radiation intensity simply gets higher and higher in the high-frequency end of the spectrum. This means for extremely high UV frequencies, for example, at very low wavelengths, the radiation intensity will be infinity. Yet the experimental results do not support the predictions of these calculations. This failure is named *ultraviolet catastrophe*; it exposed the shortcomings of classical physics because the derivation of the formula was based on basic concepts of thermodynamics and electromagnetic theory.

2.1.2 Planck's Constant

The development of quantum mechanics began in 1900 when Max Planck found the correct explanation for the *blackbody radiation* spectrum. In a paper on blackbody radiation, he proposed that radiation need not be considered a continuous wave. Instead, it could be assumed to be composed of smaller chunks which he termed "quanta." He assumed each such quanta has an energy E that is proportional to the frequency of radiation f with a constant of proportionality h ($= 6.626075 \times 10^{-34}$ J·s), which was later named Planck's constant in his honor:

$$E = hf$$

2.2 Photoelectric Effect

In 1887, Hertz noted that light incident on a clean metal plate in a vacuum emits electrons. The electrons ejected upon the metal's surface absorb the energy contained in the incident light. According to Maxwell's wave theory of light, the intensity of the incident light determines the number of electrons emitted from the metal plate. However, the energy of the emitted electrons is independent of the intensity of the incident radiation. It depends on the frequency of the incident radiation. No emission takes place if the frequency is below a certain threshold value, which is determined by the composition of the metal plate.

A few years later, Einstein showed that light consists of a stream of quanta, which he called *light quanta* (now called *photons*). Photons are electrically neutral and do not have a mass. However, a photon has an energy equal to E as in a quanta proposed earlier by Planck, and travels at the speed of light c:

$$E = hf$$

Note that this equation combines the wave and the particle natures of light, with E being the energy of one *particle* of light while f, the frequency on the right side, points to the wave nature of the same light. The constant h, as indicated earlier, is Planck's constant.

The momentum p of a photon, according to the theory of relativity is

$$p = E/c$$

Since $E = hf$,

$$p = hf/c$$

To reexpress this in terms of wavelength substitute $c = \lambda f$, where λ is the wavelength of the light wave. Thus

$$p = hf/\lambda f = h/\lambda$$

In order to remove an electron from a metal surface, a minimum energy ϕ, called the *work function*, must be expended. Suppose a single photon of energy hf is absorbed by an electron on a metal surface. Then:

1. If $hf < \phi$, the electron cannot be dislodged because it does not have the necessary energy to overcome the work function.
2. If $hf > \phi$, the electron has the energy to escape from the metal surface; any additional energy is used as kinetic energy by the electron. Mathematically, this can be written as

$$hf = \phi + \text{kinetic energy of the emitted electron}$$
$$= \phi + \tfrac{1}{2}mv^2$$

or

$$\tfrac{1}{2}mv^2 = \phi - hf$$

where m is the rest mass of the *photoelectron* (dislodged electron) and v its velocity.

The above equation shows that the kinetic energy of the photoelectron depends only on the frequency of the incident radiation, not

on its intensity. Furthermore, a photoelectron is emitted as a result of the interaction of a single electron with a single photon in the incident radiation wave, not the whole wave.

The photoelectric effect proved clearly that light is composed of particles. This result was unexpected because until then, light was considered to behave only as a wave.

2.3 Classical Electromagnetic Theory

The fundamental idea in electromagnetic theory is that an electromagnetic wave exists when a changing electric field creates a changing magnetic field, which in turn creates another changing electric field, and so on. The concept of a *field* is very important in physics and is utilized to explain forces that occur in nature in the absence of any physical contact. Each type of force (electric, magnetic, or gravitational) has its own appropriate field, that is, a region through which a force may be exerted. A field is thus a kind of medium for transmitting forces through empty space without any physical medium. For example, the gravitational, electric, and magnetic fields transmit forces that originate from masses, electric charges, and magnets, respectively. Alternatively, a field may be considered a continuous entity like fluid that fills the space surrounding the origin of the force. For example, an electric charge creates an electric field in the space around the charge, and anything that entered the space would feel a force that is stronger nearer to the charge from which it originated. Thus this space is altered by the presence of the charge; other charges in that space would feel unusual changes within that space. The electric field exists irrespective of if another charge enters that space or not. A magnetic field is caused by a moving electric charge. Thus electric and magnetic fields couple together to form electromagnetic waves.

Electric and magnetic fields of an electromagnetic wave are perpendicular to each other as shown in Fig. 2.1. Thus electromagnetic waves are *transverse waves*. Waves in which electric and magnetic fields are parallel to a pair of perpendicular axes are denoted as *linearly polarized*. Maxwell calculated the propagation speed of electromagnetic waves and found it to be equal to the speed of light (c). He concluded that light itself is an electromagnetic wave.

Waves have certain characteristics like *wavelength, frequency,* and *energy,* which basically define the nature of a particular wave (Fig. 2.2). A wavelength (λ) is the distance between two adjacent wave crests and is often expressed in centimeters. The frequency f represents the number of wave crests that pass a given point in unit time.

It is usually expressed in *hertz* (cycles/second). These two quantities are related to the speed of light c by the equation:

$$c = f \cdot \lambda$$

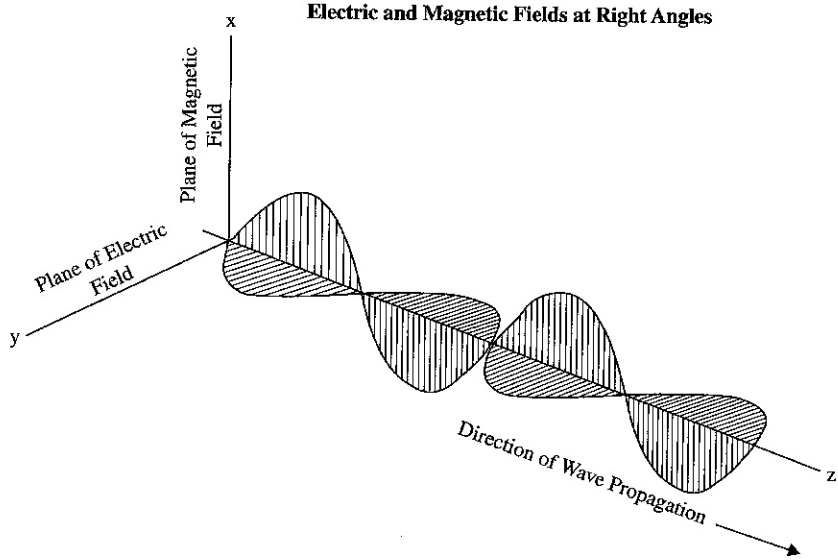


Figure 2.1 Sketch of electromagnetic waves [4].

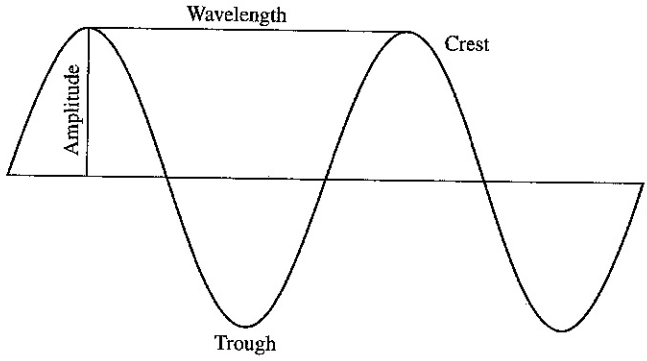


Figure 2.2 Nature of a wave.

Note the relationship between frequency f and wavelength λ from the above expression:

$$f = c/\lambda \qquad or \qquad \lambda = c/f$$

Thus the frequency is inversely proportional to the wavelength. The longer the wavelength, the lower the frequency and vice versa.

Electromagnetic waves span an enormous range of wavelengths and frequencies. This range is known as the *electromagnetic spectrum.*

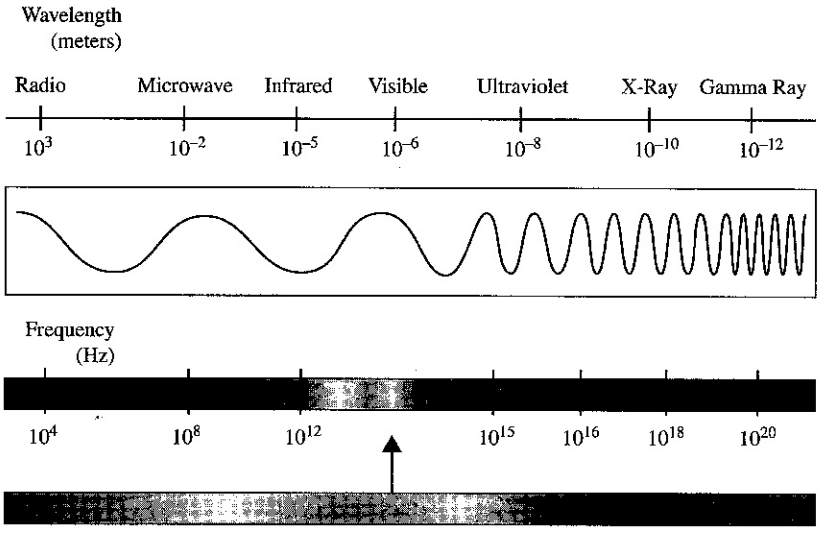


FIGURE 2.3 Electromagnetic spectrum.

The electromagnetic spectrum is generally divided into seven regions, in order of decreasing wavelength and increasing energy and frequency as shown in Fig. 2.3.

The drawback of Maxwell's electromagnetic theory is that it only expressed the wave nature of radiation; it could not explain the photoelectric effect which can only be understood if radiation is assumed to be composed of photons [2].

2.4 Rutherford's Model of the Atom

Atoms are the basic building blocks that combine to produce everyday matter in the world around us. Not all atoms are the same, however. In order to determine the difference between atoms, it is necessary to know from what they are made. The constituents of an atom are known as *subatomic particles*. In 1898, J. J. Thompson discovered the negatively charged *electron*, which was the first subatomic particle identified.

In 1914, Rutherford proposed that most of the mass of an atom and positively charged protons were concentrated in an extremely small volume of the atom. He called this region the *nucleus*, with negatively charged electrons surrounding it. He also proposed that these electrons revolve around the nucleus with very high speed in circular paths, which he named *orbits*. Since electrons are negatively charged and the nucleus is densely populated with positively charged protons, there is a strong electrostatic force that holds together the nucleus and electrons. For example, a hydrogen atom consists of a

single proton at the nucleus and an electron orbiting around it. In short, the Rutherford model of an atom is an imitation of the solar system.

Rutherford's model was widely accepted till the second half of 19th century when Maxwell developed the classical theory of electromagnetism. Classical electromagnetic theory postulates that a charged particle changing either speed, direction, or both would emit electromagnetic radiation in the form of light. An orbiting electron in Rutherford's model continuously emits radiation. As a result, the electron must lose energy and will be pulled more and more strongly by the nucleus and eventually crash into it. In practice, however, the electron does not collapse into the nucleus. Also, there is no continuous emission of radiation. On the contrary, the energy emission is confined to discrete wavelengths called a *line spectrum*, which could not be explained by the Rutherford model.

2.5 Bohr's Model of Atoms

Bohr ignored some of the concepts of classical physics utilized by Rutherford in his theory of atomic structure. Instead he used experimentally observed facts and Planck's idea of energy quantization to propose a new model in 1913. He developed his theory of the atom based on the hydrogen (one-electron) atom and used the following postulates:

1. The electrons in an atom move in a circular orbit around the positively charged nucleus under the influence of the electrostatic attraction between an electron and the nucleus; this force is balanced by the centrifugal force due to the velocity of an electron in its orbit. The orbits used by the electrons are specific distances away from the nucleus. The electron in a hydrogen atom normally stays in the first orbit, known as the *ground state*, which is closest to the nucleus.
2. The energy of the electron remains constant as long as it remains in a permitted orbit.
3. Radiation is emitted or absorbed when the electron makes a transition from an allowed orbit to another of lower or higher energy, respectively (Fig. 2.4). The energy difference between the two orbits ΔE is given by

$$E_2 - E_1 = |\Delta E| = hf$$

When an electron moves from a higher (energy) orbit E_2 to a lower one E_1 thereby losing energy, the lost energy manifests as a photon, that is, light. The energy of the emitted light is related to its color.

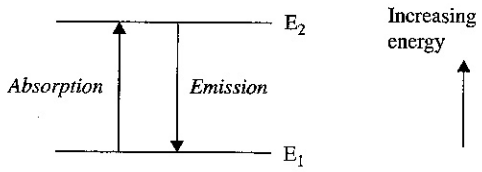


FIGURE 2.4 Transition from lower orbit to higher orbit and vice versa.

The electron loses the same energy it absorbed when it jumped from a lower orbit to a higher one. Bohr noticed that all of the possible combinations of jumps the electron could make to return to the ground state from the higher orbits account for the existence of all the spectral lines in the hydrogen atom.

Quantum computers utilize certain unique properties of subatomic particles to dramatically speed up computation that is not possible with traditional computers. Traditional computers are bound by the theories of classical physics, Newton's mechanics and Maxwell's electromagnetism. As indicated previously, Newton's mechanics are used to describe with precision the location and motion of *macroscopic* objects whereas electromagnetism is utilized to explain phenomena such as light and electromagnetism. Newton's mechanics for macroscopic objects gave contradictory results to theoretical predictions, however, when applied to objects at the *microscopic* level, that is, atomic and subatomic particles. This led to the development of a new type of physics called *quantum mechanics*, which is basically a branch of physics that deals with the *very small.*

2.6 Particle and Wave Nature of Light

One of the most well-known demonstrations of wave nature of particles is the double slit experiment by Young at the beginning of the 19th century. It provided strong experimental evidence to support the wave nature of light. A screen that had two slits cut into it was placed in front of a monochromatic light source. If light is composed of particles, only the couple of rays of light that hit the exact location of the slits would be able to pass through. However, if light is a wave, two light rays as mentioned in the previous case will go through the slits but will diffract, that is, spread out. At some points on the screen, the crest of one ray will meet the crest of the other (*constructive interference*), and in the other case, the crest will meet troughs of the other ray, canceling each other out (*destructive interference*) as shown in Fig. 2.5. In constructive interference, the phases of the waves are such that they add to form a combined wave of greater amplitude. The figure shows the greatest possible effect of constructive interference. All the parts of the two waves line up to interfere constructively everywhere.

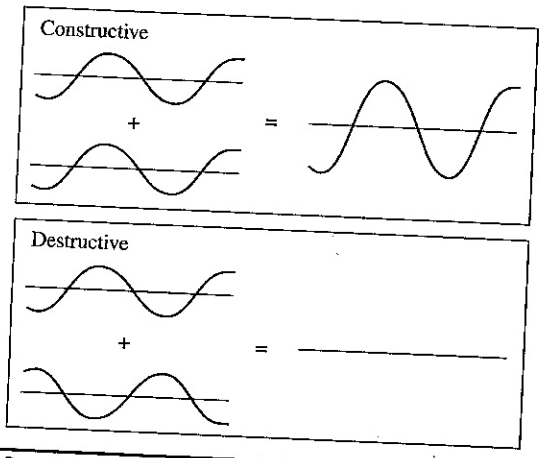


Figure 2.5 Constructive and destructive interference of light.

In destructive interference, the phases are such that the waves subtract to cancel out. The figure shows the greatest possible effect of destructive interference. All parts of the two waves line up in such a way as to interfere destructively everywhere. The constructive interference will appear as a bright spot on the screen whereas the destructive interference will manifest as a dark spot. Thus the double slit experiment conclusively proves the wave nature of light.

The paradox is that the experimental evidence provided in early decades of the 20th century, such as the photoelectric effect, showed just as clearly that light is composed of particles. The work of Arthur Compton in 1923 confirmed that particles of light, that is, photons, are scattered after collision with particles of matter (*electrons*), just like hard balls colliding with other hard balls. In the photoelectric effect, the energy of an incident photon is transferred to an electron. Instead of photon absorption by an electron, Compton's approach was to study scattering of photons after their collision with electrons. X-rays are electromagnetic waves of very short wavelength. Compton's experiment involved firing a beam of x-rays into a gas. The energy of a portion of photons from the incident x-rays was transferred to the electrons in the gas in the form of kinetic energy, as seen in particle collisions. An electron hit by x-rays is either ejected or moved into a higher orbit. The photon transfers its momentum to the electron and is scattered with a lower momentum.

Compton scattering led to the inevitable conclusion that electromagnetic radiation has a wave nature as well a particle nature. The interference and diffraction of light can be explained only if light behaves like a wave whereas the photoelectric effect and Compton scattering can only be explained if light behaves like particles. This characteristic of light has become known as *wave-particle duality*.

The dual nature of electromagnetic radiation led De Broglie in 1924 to conjecture that since waves have particle nature by symmetry, particles will have wave nature. Using Einstein's equation, De Broglie derived an equation that combined the particle nature of waves with wave nature of particles. This equation showed the wavelength associated with a photon can be obtained by dividing the Planck's constant (a very tiny number) with the momentum of the photon:

$$\lambda = \frac{h}{p}$$

De Broglie concluded that this equation is universally true. This wave nature is not a unique characteristic of photons only; it is exhibited by all particles of matter. When electrons travel through double slits for example, they do not pile up at two different locations on the screen behind the two slits. Instead they produce a bright (*constructive interference*) and a dark band (*destructive interference*) exactly like the ones produced by photons. This clearly exhibits the wave particle duality of electrons.

In 1927, Davisson and Germer showed that a beam of electrons of momentum p produced a diffraction pattern similar to that produced by a wave with wavelength λ. In short, the photon, as well as the electron, was found to behave sometimes as a particle and sometimes as a wave. This evidently implied some form of *wave-particle duality* for subatomic particles in nature that could not be explained by using the concepts of classical physics. A radically different approach was needed. Within a few years, a new physics was introduced that could account for the phenomenon of wave-particle duality of very small particles and developed leads to the inherently probabilistic nature of the universe.

2.7 Wave Function

In quantum mechanics, the state of a particle is not represented by position and momentum but by the *wave function*. The wave function contains all the measurable information about that particle as a function of position and time. Using the wave function, one can calculate a system's future behavior but only with a certain probability.

Erwin Schrödinger reasoned that if quantum particles behave as waves, it should be possible to describe them using a *wave equation*. This equation known as *the Schrödinger equation* shows how the quantum wave function evolves in time. The wave function is a complex function and is often denoted by the letter ψ.

The wave function of a particle of energy E can be specified as a linear combination of the following form of wave functions [3]:

$$\psi(x, t) = Ae^{i(kx - \omega t)}$$

where A is the amplitude of the wave, ω is the *angular frequency* given by $\omega = 2\pi f$, $k = 2\pi/\lambda$, x is the position, and t is time.

Since $k = 2\pi/\lambda$ and $\omega = 2\pi f$,

$$\psi(x, t) = Ae^{i(2\pi(x/\lambda - ft)}$$

Substituting $\lambda = h/p$ and $E = hf$

$$\psi(x, t) = Ae^{i(2\pi/h(px - Et)}$$

Finally, replacing $\hbar$ (*h bar*) by $h/2\pi$ gives,

$$\psi(x, t) = Ae^{i(px - Et)/h}$$

Irrespective of the state of a particle, its wave function assigns a *complex number* called the *amplitude* to each possible measurement outcome. Max Born showed that the integral of $|\psi|^2$ between any two points gives the probability of the particle appearing between those two points. This is known as Born's rule and can be restated as:

The probability of obtaining any possible measurement outcome is equal to the square of the corresponding amplitude; the wave function is just the set of all the amplitudes.

Thus

$$\text{Probability}(x) = |\text{amplitude}(x)|^2$$

For instance, the probability of finding the particle somewhere between $x = 0$ and $x = 1$ is:

$$\int_0^1 |\psi(x)|^2 | dx$$

This is the *normalization* condition on a probability density; the integral of the density over the range for which it is defined must sum to 1 since whatever number is generated must come from that interval.

In 1927, Werner Heisenberg proposed that there is a fundamental limit to what can be known about a quantum particle. His proposition, known as the *uncertainty principle* or *indeterminacy principle*, is one of the fundamental postulates of quantum mechanics. Heisenberg uncertainty principle states that certain attributes of quantum particles are linked together such that they cannot be simultaneously measured with accuracy; such an attribute is known as a *conjugate attribute*. If one attribute of a conjugate pair is measured with good precision, the other attribute cannot be measured as precisely. The most important conjugate attributes of a quantum particle are its position and its momentum. For example, it is impossible to know

simultaneously the exact position and momentum of a particle. That is, the more exactly the position is determined, the less likely is to be known about the momentum and vice versa. This is not due to the limitation of the measurement technology but the manifestation of a fundamental limit on what can be known about a particle at any given moment. This uncertainty in measurement arises because the act of measuring affects the object being measured.

Heisenberg's uncertainty principle indicates that the uncertainty in position measurement (Δx) and the uncertainty in momentum measurement (Δp) are approximately related by

$$\Delta x \cdot \Delta p \geq h$$

where h is Planck's constant. Thus, greater accuracy in position measurement is possible only with greater uncertainty in momentum measurement and vice versa. For example, if the position of a particle is to be measured with very high accuracy, the uncertainty in the position of the particle Δx must be 0; that is, the exact location of the particle is known. Since the product of Δx and Δp must be equal to or greater than a fixed quantity (Planck's constant), Δp will be infinite if $\Delta x = 0$. Hence if the position of the particle is known, the uncertainty in the momentum of the particle is infinite; that is, there will be total uncertainty in momentum. On the other hand, if the particle is at rest and the uncertainty in the momentum of the particle $\Delta p = 0$, then the position uncertainty Δx will be infinite; that is, the position of the particle is completely unknown. A similar relationship for uncertainty in energy measurement (ΔE) and the uncertainty in elapsed time (Δt) was also derived by Heisenberg:

$$\Delta E \cdot \Delta t \geq h$$

2.8 Postulates of Quantum Mechanics

A postulate is a statement that is assumed to be true without proof, whereas a theorem is a true statement that can be proven. The essence of quantum postulates is represented in this section with a set of seven postulates. These postulates provide a connection between the physical world and the mathematical framework of the quantum mechanics. The postulates are:

1. Every physical system has an associated complex vector space with an inner product (Hilbert space) known as the *state space* of the system. A unit vector represents the physical system in the state space; this unit vector known as the *state vector* contains all the information that can be known about the system.

2. All observables (measurable properties) of a physical system are represented by Hermitian operators acting in the system's state space. Any measurement performed on a quantum system necessarily involves some interaction between the system and the observer. Thus when an observable is measured obtaining a certain outcome λ, the measurement leaves the system in a state that is an eigenvector of the operator with eigenvalue λ.
3. The probability of measuring an observable with eigenvalue λ in a quantum state is

$$\text{prob}(\lambda) = |\langle a | \lambda \rangle|^2$$

where $|a\rangle$ is an eigenvector of the corresponding Hermitian operator with eigenvalue λ.

4. Measurements in systems having the same state vector may not result in the same outcomes. Only the probabilities of various outcomes can be known.
5. The *expectation value* of a measurable quantity Q is defined as the average of values that is likely to be obtained if Q is measured on a large number systems having the same state vector $|u\rangle$. The expectation value is postulated to be

$$\langle u | Q | u \rangle$$

provided $\langle u | u \rangle = 1$.

6. The state space of a composite physical system is the tensor product of the state spaces of the component systems.
7. The evolution of a quantum system in time is described by a unitary transformation.

References

1. Physics of the Universe (*http://www.lukemastin.com/physics/topics_quantum.html*).
2. Bernard D'Espagnat, *Conceptual Foundations of Quantum Mechanics*, 2nd ed., Westview Press, 1999.
3. J. D. Cresser, Quantum Physics Notes, Department of Physics, Macquarie University, Australia, 2011.
4. The BB84 protocol for quantum key distribution, *Quantum Gazette: Exploration into Quantum Physics and Science Writing*, September 22, 2016

CHAPTER 3

Matrices and Operators

In quantum mechanics, physical processes such as measurement, rotation, and time evolution cannot be done directly. Only the quantum state corresponding to these can be acted upon by a mathematical operation, the *operator.*

There are many observables in quantum mechanics; an *observable* is the property of a quantum object that can be measured, such as position, momentum, or energy. Each observable corresponds to a *linear operator* that acts on a state and gives another state, that is, it changes one function into another. In other words, a linear operator A is a linear function from a vector space to itself. In notations:

$$A:V \rightarrow V$$

which satisfies

$$\mathbf{A}(au + bv) = a(\mathbf{A}u) + b(\mathbf{A}v)$$

for all a, b in the underlying field and vectors u, v.

In elementary calculus, a *function* $f(x)$ is considered a *rule* that associates with each number x another number $y = f(x)$. This concept of a function can be extended to apply to vectors as well; in such cases the term function is called an *operator.* More formally, an operator A specifies a rule that transforms a vector $\mathbf{v}_i$ in a vector space to another vector $\mathbf{v}_j$ in the space. This is read as operator O acting on vector $\mathbf{v}_i$ and transforming it into vector $\mathbf{v}_j$:

$$O(\mathbf{v}_i) = \mathbf{v}_j$$

Like an operator, a matrix can also transform a vector into another vector. Thus, there is a correlation between linear operators and matrices. Each linear operator can be represented by a corresponding matrix and the converse is also true; each matrix generates a corresponding linear operator. Since linear operators can be represented by matrices, all relevant properties of such matrices also apply to operators.

3.1 Matrices

A matrix is a rectangular array of real (or complex) numbers containing m rows and n columns; an $m \times n$ matrix contains m rows and n columns and can be expressed as

$$A = |a_{ij}| = \begin{vmatrix} a_{11} & a_{12} \cdots a_{1n} \\ a_{21} & a_{22} \cdots a_{2n} \\ & \cdots \\ & \cdots \\ a_{m1} & a_{m2} \cdots a_{mn} \end{vmatrix}$$

The quantities a_{ij} are called the *entries*, or *components*, of the matrix. The entry at the ith row and jth column of matrix A is denoted by the corresponding lowercase alphabet with subscripts, for example, a_{ij}. As in the case of vectors, the components of a matrix may be real or complex. If they are real numbers, the matrix is called *real*; otherwise it is *complex*. If the number of rows and the number of columns in a matrix are finite, the matrix is said to be *finite*. Otherwise, it is infinite.

Two matrices are *equal* if they have the same order and their entries are equal. Two matrices can be added together if they have the same size. The sum of two $m \times n$ matrices $A[a_{ij}]$ and $B[b_{ij}]$ is obtained by adding the corresponding entries together and results in another $m \times n$ matrix C:

$$C_{ij} = A[a_{ij}] + B[b_{ij}]$$

The subtraction of matrix B from matrix A results in another $m \times n$ matrix D:

$$D_{ij} = A[a_{ij}] - B[b_{ij}]$$

A matrix A of any order can be multiplied by a *scalar* (number). If the scalar is s, every entry in the matrix is multiplied by s:

$$s[A] = [sa_{ij}]$$

Two matrices can be multiplied if the number of columns in the first matrix is equal to the number of rows in the second matrix. Thus, a $p \times q$ matrix $A = [a_{ij}]$ can multiply a $q \times r$ matrix $B = [b_{ij}]$. This results in a product matrix C of order $p \times r$:

$$AB = [c_{ij}]$$

where $c_{ij} = a_{i1}\, b_{1j} + a_{i2}\, b_{2j} + a_{i3}\, b_{3j} + \cdots + a_{ip}\, b_{pj}$

3.2 Square Matrices

A *square matrix* is the matrix in which the number of rows is equal to the number of columns, that is, $m = n$. An $n \times n$ matrix is known as a square matrix of *order* n. Square matrices have some unique properties such as *symmetry* and *antisymmetry*. Furthermore, finding determinants and computing eigenvalues are only possible in square matrices. A complex matrix of order 3 is shown below:

$$\begin{vmatrix} 2-i & 3 & 7 \\ 3 & 6i & 1-i \\ 5 & 3+2i & 8 \end{vmatrix}$$

In a square matrix, the entries a_{ii} $(i = 1,..., n)$ form the *main diagonal*. They lie on the imaginary line that runs from the top left corner to the bottom right corner of the matrix. The *anti-diagonal*, or *cross-diagonal*, runs from bottom left to upper right. For the matrix shown above, the main diagonal is $(2–i, 6i, 8)$ and the anti-diagonal is $(5, 6i, 7)$.

Two square matrices A and B *commutate* if

$$AB = BA$$

Otherwise, they do not. For example, if A and B are as follows:

$$A = \begin{vmatrix} i & 0 & 0j \\ 0 & 1 & 0 \\ 1 & 0 & 2i \end{vmatrix} \quad B = \begin{vmatrix} 2j & 4 & 0 \\ 3 & 1 & 0 \\ -1 & -4 & 1 \end{vmatrix}$$

then

$$AB = \begin{vmatrix} -2 & 4i & 0 \\ 3 & 1 & 0 \\ 0 & 4-8i & 2i \end{vmatrix} \quad \text{and} \quad BA = \begin{vmatrix} -2 & 4 & 0 \\ 3i & 1 & 0 \\ 1-i & -4 & 2i \end{vmatrix}$$

Therefore, A and B do not commutate.

The *trace* of an $n \times n$ square matrix M, denoted as $Tr(M)$, is the sum of the diagonal entries of the matrix:

$$Tr\, M = \sum_{i=1}^{n} m_i^i$$

Thus the trace of matrix A in the previous example is

$$Tr\, A = i + 1 + 2i = 1 + 3i$$

While matrix multiplication does not commute, the trace of a product of matrices does not depend on the order of multiplication.

3.3 Diagonal (or Triangular) Matrix

A square matrix of order n is called a *diagonal matrix* if

$$a_{ii} = d_i$$

$$a_{ij} = 0 \text{ if } i \neq j$$

For example, consider the 3×3 matrix shown below. The elements are $a_{11} = 8$, $a_{22} = 10$, $a_{33} = 3$. All other entries above and below the diagonal line are 0s. Hence this is a *diagonal matrix:*

$$\begin{vmatrix} 3 & 0 & 0 \\ 0 & 5i & 0 \\ 0 & 0 & 4 \end{vmatrix}$$

Note that a diagonal divides the matrix into two blocks; one above the diagonal and the other one below it. If the lower block consists of all 0s, the matrix is identified *upper-triangular*. Alternatively if the upper block consists of all 0s, the matrix is called *lower-triangular.* For example, the matrix

$$\begin{matrix} a_{11} & a_{12} & a_{13} & a_{14} \\ 0 & a_{22} & a_{23} & a_{24} \\ 0 & 0 & a_{33} & a_{34} \\ 0 & 0 & 0 & a_{44} \end{matrix}$$

is *upper-triangular,* while the matrix

$$\begin{matrix} a_{11} & 0 & 0 & 0 \\ a_{21} & a_{22} & 0 & 0 \\ a_{31} & a_{32} & a_{33} & 0 \\ a_{41} & a_{42} & a_{43} & a_{44} \end{matrix}$$

is *lower-triangular.*

Formally, a square matrix Y is upper triangular if

$$y_{ij} = 0 \qquad i > j$$

A square matrix X is lower triangular if

$$x_{ij} = 0 \qquad i < j$$

A square matrix that is either lower or upper triangular is known as a *triangular matrix*. A diagonal matrix, on the other hand, is both upper and lower triangular.

3.4 Operators

As mentioned earlier in this chapter, an operator A can be considered a *rule* that transforms one function $f(x)$ into a new function $Af(x)$. For example, the operator $D\left(=\frac{d}{dx}\right)$ derives the derivative of the function:

$$D(2x^2 + 3x) = 4x + 3$$

An operator in a vector space is a mapping between two vectors in that space. More precisely, this mapping is a transformation that takes a vector as an input and produces another vector as the output. For example, if an operator A acts on vector $\mathbf{v}_1$ in a vector space to transform it into another vector $\mathbf{v}_2$ in the same space, this is written as

$$A|\mathbf{v}_1\rangle = |\mathbf{v}_2\rangle$$

A measurable property of a physical system is denoted as an *observable* in quantum mechanics. An operator is often used to represent an observable. For example, a particle's momentum and position can both be observed and represented as operators. An operator on a vector space is a mapping between two vectors in that space. More precisely, this mapping is a transformation that takes a vector as an input and produces another vector as the output. For example, if an operator O acts on vector $\mathbf{v}_1$ on a vector space to transform it into another vector $\mathbf{v}_2$ in the same space, this is written as

$$O|\mathbf{v}_1\rangle = |\mathbf{v}_2\rangle$$

In general, operators could be of any form, but in quantum mechanics only a class of operators known as *linear operators* is of particular interest. In order to be *linear*, operator O must transform a linear combination of vectors into a linear combination of transformations of the vectors.

3.4.1 Rules for Operators

There are certain rules for working with operators that are collectively known as *algebra for operators:*

$$(A + B)|\varphi\rangle = A|\varphi\rangle + B|\varphi\rangle$$

$$(AB)|\psi\rangle = A[B|\psi\rangle]$$

Note that in the product (AB) $|\psi>$ the rightmost operator, B, acts first on vector $|\psi>$, and then the operator A acts on the vector resulting from the action of the operator B. The operator multiplication is associative. For any three operators A, B, and C:

$$A(BC) = (AB)C$$

But the operator multiplication is not commutative:

$$AB \neq BA$$

It should be noted that the difference is:

$$AB - BA = [A, B]$$

If AB = BA, then [AB] = 0 and the operators are said to commutate with each other. Note that the *commutator* of A and B is another linear operator. In quantum mechanics two physical quantities are simultaneously observable if their operator representations *commutate.*

3.5 Linear Operator

An operator can transform a vector $\mathbf{v}_i$ in a vector space V to another vector $\mathbf{v}_j$ that may belong to a vector space other than V. In general, operators could be of any form, but in quantum mechanics only a class of operators known as *linear operators* are of particular interest. If an operator A acts on vector $\mathbf{v}_1$ in the vector space V to transform it into another vector $\mathbf{v}_2$ also in the vector space V, then the operator A is referred to as a *linear operator*:

$$A|\mathbf{v}_1> = |\mathbf{v}_2>; \text{ for all } \mathbf{v}_1, \mathbf{v}_2 \in V$$

A linear operator A has the following properties:

1. $A(u + v) = Au + Av$, for all $u, v \in V$
2. $A(cu) = cAu$, for all $c \in F$ and $u \in V$

It should be clear that a linear operator A preserves linear combinations:

$$A(ru + sv) = r\,A(u) + sT(\mathbf{v})$$

Like a linear operator that can transform a function into another function, a matrix can transform a vector into another vector. Thus there is a definite correlation between linear operators and corresponding matrices. In fact, there exists a unique $n \times n$ matrix corresponding to a linear operator that acts on a vector space of dimension n and a basis producing another vector with respect to the given basis.

Assume for example that the action of operator O on the vectors $|f>$ and $|g>$ is given by

$$O|f> = |p> \text{ and } O|g> = |q>$$

where $|p>$ and $|q>$ are new vectors resulting from the action of the operator O on $|f>$ and $|g>$. Therefore the effect of the operator O on the quantum state $(c_1|f> + c_2|g>)$, where c_1 and c_2 are two arbitrary complex numbers is

$$O(c_1|f> + c_2|g>) = c_1 O|f> + c_2 O|g>)$$
$$= c_1|p> + c_2|q>$$

In quantum mechanics, as mentioned earlier, all operators with one exception are linear. The exception known as an *antilinear* operator Q has the property:

$$Q(c_1|f> + c_2|g>) = c_1^* Q|f> + c_2^* Q|g>$$

The *product of two operators* yields a new operator. For example, assume an operator C acting on a vector $|s>$ to produce another vector $|t>$:

$$C|s> = |t>$$

Next, assume another operator D acts on $|t>$ to produce a new vector $|u>$, thus

$$D(C|s>) = C|t> = |u>$$

This can be rewritten as

$$DC|s> = |u>$$

and indicates that an operator C acts on a ket vector first, and then another operator D acts on the resulting ket vector; the combination DC is known as the *product of operators* C and D. Thus the product of two operators E = DC can be written as

$$E|s> = DC|s>$$

3.6 Commutator

It should be noted that the order in which two operators C and D are multiplied is important because in general:

$$CD \neq DC$$

If the difference between the two product terms is 0, that is, CD = DC, the operators *commute*. Otherwise, they do not. For a quantum particle, the observable properties corresponding to two *noncommuting* operators cannot be known simultaneously. For example, if C is the *momentum* operator and D is the *position* operator, then:

$$(CD - DC)\,\varphi \neq 0$$

Therefore, C and D do not commute.

3.7 Matrix Representation of a Linear Operator

Assuming $B = (v_1, v_2, \ldots, v_n)$ is an ordered basis of V, every vector **v** of V can be expressed as a linear combination of the vectors in B. The term *ordered basis* indicates that the basis vectors in B must be listed such that it is clear which one is first, which one is second, etc. In other words, the order in which the vectors are listed is important. Thus $B_1 = (v_2, v_1, \ldots, v_n)$ is a different ordered basis for V.

Since each vector in V can be expressed as a linear combination of basis vectors in B

$$\mathbf{v} = \{c_1 v_1, c_2 v_2, \ldots, c_n v_n\}$$

Therefore, an operator A acting on V results In

$$A(\mathbf{v}) = c_1 A(v_1) + c_2 A(v_2) + \cdots + c_n A(v_n)$$

Assume $\{g_1, g_2, \ldots, g_m\}$ is a basis for vector **W**. Then the matrix of operator A with respect to this basis can be obtained by computing $A(v_1) \ldots A(v_n)$, and then expanding them using the basis $\{g_1, g_2, \ldots, g_m\}$ as below, where c_{ij} $(i = 1 \ldots m, j = 1 \ldots n)$ are constants:

$$A(v_1) = c_{11} g_1 + c_{21} g_2 + \cdots + c_{m1} g_m$$

$$A(v_2) = c_{12} g_1 + c_{22} g_2 + \cdots + c_{m2} g_m$$

$$\ldots$$

$$A(ve_n) = c_{1n} g_1 + c_{2n} g_2 + \cdots + c_{mn} g_m$$

By arranging all the scalars c_{ij} column by column in a matrix form, it results in matrix **A** relative to the given basis $\{b_1, b_2, \ldots, b_n\}$ and $\{g_1, g_2, \cdots, g_n\}$:

$$A = \begin{pmatrix} c_{11} & c_{12} \ldots\ldots\ldots c_{1n} \\ c_{21} & c_{22} \ldots\ldots\ldots c_{2n} \\ & \ldots\ldots\ldots \\ c_{m1} & c_{m2} \ldots\ldots\ldots c_{mn} \end{pmatrix}$$

An alternative way of showing the matrix representation of a linear operator is based on the definition that when a linear operator acts on a vector, the result is another vector. If an operator acts on a column matrix representing a vector, the result will be another column matrix. Consequently, the action of the operator on a vector can be viewed as a matrix multiplication [1]. That is

$$|\varphi\rangle = A\,|\psi\rangle$$

3.8 Symmetric Matrix

A square matrix is *symmetric* if it is equal to its transpose. In other words, an $n \times n$ matrix A is symmetric if $A = A^T$ such a matrix is symmetric about its main diagonal (from top left to bottom right).

Some of the important properties of symmetric matrices:

1. The transpose of a symmetric matrix A is also symmetric.
2. The product of a symmetric matrix A and a scalar c, that is, cA is also symmetric.
3. The inverse of the transpose of a symmetric matrix A is equal to the inverse of the matrix.

$$(A^T)' = A'$$

4. If A and B are two symmetric matrices and $(A + B)^T = A^T + B^T = A + B$ then $(A + B)$ is a symmetric matrix.
5. The product of two symmetric matrices is also symmetric if the two matrices commute, that is, $AB = BA$.

In quantum mechanics, two symmetric matrices are of particular importance, *Hermitian matrices* and *unitary matrices*. In order to understand these matrices, it is necessary to be familiar with certain operations discussed in the following sections.

3.9 Transpose Operation

The transpose of an $m \times n$ matrix A, denoted by A^T, is an $n \times m$ matrix with entries

$$(A^T)_{ij} = a_{ji}$$

Thus the transpose of a matrix can be derived by replacing ith row of the matrix of A by the ith column of A. The transpose operation is illustrated below:

$$A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix} \qquad A^T = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}$$

The following results are easy to prove:

1. $(A^T)^T = A$ and $(A^*)^* = A$
2. $(A \pm B)^T = A^T \pm B^T$ (and also for *)
3. $(cA)^T = c\,(A^T)$
4. $(AB)^T = B^T A^T$
5. $(A^T)^T$

3.10 Orthogonal Matrices

A square matrix is called *orthogonal* if the transpose of the matrix A is equal to its inverse:

$$A^T = A^{-1}$$

or equivalently, if $A^T A = I$

For example, the matrix A shown below is orthogonal:

$$A = \begin{vmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{vmatrix}$$

$$A^T = \begin{vmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{vmatrix}$$

Since $A\,A^T = \begin{vmatrix} 1 & 0 \\ 0 & 1 \end{vmatrix} = I$, matrices A and A^T are orthogonal.

Some of the properties of orthogonal matrices are

1. In an orthogonal matrix, the inner product of any two row vectors or any two column vectors is equal to zero.
2. Identity matrices are orthogonal.
3. The product of an orthogonal matrix and its transpose matrix is equal to the identity matrix of same order.
4. The product of two orthogonal matrices is also orthogonal.
5. An orthogonal matrix is always a symmetric matrix.

3.11 Identity Operator

The *identity operator*, written *I*, is a square matrix in which all items on the main diagonal are ones and all remaining items are zeroes. Thus an identity matrix can be specified as

$$I_{ii} = 1, \text{ and}$$
$$I_{ij} = 0 \text{ if } i \neq j$$

For example, a 4×4 identity matrix is shown below:

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

It should be clear that a diagonal matrix can also be used to denote the *identity matrix:*

$$I_n = (a_{11}, a_{22}, \ldots, a_{nn})$$

where n is the *trace* of the matrix.

The multiplication of a 4×4 matrix by the identity matrix is shown below:

$$\begin{pmatrix} 2 & 0 & 5 & 0 \\ 0 & 3 & 0 & 6 \\ 4 & 1 & 1 & 3 \\ 0 & 2 & 0 & 7 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 2 & 0 & 5 & 0 \\ 0 & 3 & 0 & 6 \\ 4 & 1 & 1 & 3 \\ 0 & 2 & 0 & 7 \end{pmatrix}$$

As can be seen above, the multiplication of the matrix by the identity matrix reproduces the original matrix without any change. In other words, the effect of multiplying a matrix by a *unity* (*identity*) *matrix* has the same effect as multiplying by 1.

3.12 Adjoint Operator

One of the most important operations in complex linear algebra is the *Hermitian conjugate,* or *adjoint,* of a linear operator. The adjoint of a operator A, denoted as $A^{\dagger}$, is obtained by taking the complex conjugate of all of its entries and then interchanging the rows and columns. Thus

$$A^{\dagger}(i, j) = A(j, i)^*$$

As an example, consider the following 3×3 matrix:

$$A = \begin{vmatrix} 2 & -1 & 4 \\ 1-2i & 5 & 2-5i \\ -5i & 2+i & 3+7i \end{vmatrix}$$

The complex conjugate of A, denoted as A*, is derived first by taking the complex conjugate of each entry:

$$A^* = \begin{vmatrix} 2 & i & 4 \\ 1+2i & 5 & 2+5i \\ 5i & 2-i & 3-7i \end{vmatrix}$$

Next, A* is transposed to obtain the *adjoint* (or *Hermitian conjugate*) of the matrix:

$$A^\dagger = \begin{vmatrix} 2 & 1+2i & 5i \\ i & 5 & 2-i \\ 4 & 2+5i & 3-7i \end{vmatrix}$$

The *adjoint* of a linear operator A can be written in bra-ket notation [2]:

$$\langle A^\dagger \varphi \mid \psi \rangle = \langle \varphi \mid A\psi \rangle \tag{3.1}$$

By changing the order of the bras and kets on the left, this can be rewritten as

$$(\langle \psi \mid A^\dagger \varphi \rangle)^* = \langle \varphi \mid A\psi \rangle$$

Taking the complex conjugates of both sides:

$$(\langle \psi \mid A^\dagger \mid \varphi \rangle) = (\langle \varphi \mid A \mid \psi \rangle)^*$$

Next, changing the order of bras and kets in Eq. (3.1):

$$\langle \psi \mid A^\dagger \varphi \rangle = \langle A\psi \mid \varphi \rangle$$

By eliminating $\mid \varphi \rangle$ from both sides:

$$\langle \psi \mid A^\dagger \equiv \langle A\psi \mid$$

The bra operator $\langle A\psi \mid$ is known as the *adjoint* of A, and is denoted as $A^\dagger$. In other words, the ket $A \mid \psi \rangle$ has a corresponding bra $\langle \psi \mid A^\dagger$ [2].

The Hermitian adjoint of an expression composed of bras, kets, constants and operators can be obtained by following the steps below:

1. Replace any complex constants with their complex conjugates. The Hermitian adjoint of a complex number is the complex conjugate of that number, that is, $a^\dagger = a^*$.
2. Replace kets with their corresponding bras and replace bras with their corresponding kets. Note that the bras and kets need to be swapped while deriving the adjoint of an operator.

3. Replace operators by their adjoints.
4. Reverse the order of the factors.

3.13 Hermitian Operator

A class of operators of special importance are the *Hermitian operators* because the *eigenvalues* of a Hermitian operator are the possible values of the observables; this implies that measured values are real numbers, not complex numbers.

In quantum mechanics, operators that are equal to their adjoints are called *Hermitian* or *self-adjoint*. In other words, an operator A is Hermitian if and only if satisfies the following equation:

$$A^{\dagger} = A$$

As an example, derive the adjoint of the following matrix representing a linear operator and determine whether or not it is Hermitian:

$$A = \begin{pmatrix} 2 & 2+i & 4 \\ 2-i & 3 & i \\ 4 & -i & 1 \end{pmatrix}$$

First, determine the complex conjugate of the matrix A^*

$$A^* = \begin{pmatrix} 2 & 2-i & 4 \\ 2+i & 3 & -i \\ 4 & i & 1 \end{pmatrix}$$

Next, A^* is transposed resulting in

$$A^{\dagger} = \begin{pmatrix} 2 & 2+i & 4 \\ 2-i & 3 & i \\ 4 & -i & 1 \end{pmatrix}$$

Since $A = A^{\dagger}$, the matrix is Hermitian.

Note that every entry in $A^{\dagger}$ is equal to the complex conjugate of the entry symmetrically located with respect to the diagonal of the matrix.

Hermitian operators have some special properties:

1. They are the complex analogue of symmetric matrices. The diagonal entries of a Hermitian matrix must be real since the conjugation process must not change these entries. Moreover, each entry and its image across the diagonal must be complex conjugates.

2. The expectation value of a Hermitian operator is real on any state. It is also true, however, that any operator whose expectation value is real for all states must be Hermitian.
3. The eigenvalues of Hermitian operators are real.

3.14 Unitary Operators

An operator represented by an $n \times n$ *matrix* A is unitary if it produces an identity matrix when multiplied by its conjugate transpose:

$$AA^{\dagger} = I$$

In other words, A is a *unitary matrix* if its conjugate transpose is equal to its inverse:

$$A^{\dagger} = A^{-1}$$

For example, the following matrix is unitary:

$$A = \frac{1}{2}\begin{pmatrix} 1+i & 1-i \\ 1-i & 1+i \end{pmatrix}$$

The conjugate transpose of the matrix is

$$A^{\dagger} = \frac{1}{2}\begin{pmatrix} 1-i & 1+i \\ 1+i & 1-i \end{pmatrix}$$

Therefore,

$$AA^{\dagger} = \frac{1}{4}\begin{pmatrix} (1+i)(1-i)+(1-i)(1+i) & (1+i)(1+i)+(1-i)(1-i) \\ (1-i)(1-i)+(1+i)(1+i) & (1-i)(1+i)+(1+i)(1-i) \end{pmatrix}$$

$$= \frac{1}{4}\begin{pmatrix} 2+2 & 2-2 \\ 2-2 & 2+2 \end{pmatrix}$$

$$= \frac{1}{4}\begin{pmatrix} 4 & 0 \\ 0 & 4 \end{pmatrix}$$

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

$$= I$$

3.14.1 Properties of Unitary Operators

1. A unitary operator preserves the inner product of any two vectors **u** and **v** in the vector space V.

   To illustrate, assume

   $$<c| = <U \mid a>$$

   and

   $$<d| = <U|b>,$$

   then:

   $$<c|d> = <U|a><U|b>$$
   $$= <a|U\dagger U|b$$
   $$= <a|b>$$

2. The operator is *invertible*. Notice that the matrix corresponding to the operator is *invertible* if and only if the matrix is not *singular*. The determinant of a singular matrix is zero.
3. The eigenfunctions are orthonormal and they are complete.
4. Eigenvalues need not be real.

3.15 Projection Operator

Consider the expansion of an arbitrary ket $|\varphi>$ in a basis $(v_0, v_1, v_2, \ldots,)$

$$|\varphi> = \sum_{i=0}^{n} c_i |v_i>$$

Suppose the linear operator $<v_1|v_1>$ is applied to vector $|\varphi>$; this will pick up the vector's projection in the $|v_1>$ direction:

$$|v_1><v_1|\varphi>$$

If the vectors are orthonormal, the operator $|v_0><v_0| + |v_1><v_1|$ projects the vector onto the two-dimensional subspace defined by $|v_1>$ and $|v_2>$ [3]:

$$|v_0><v_0|\varphi> + |v_1><v_1|\varphi>$$

In a similar manner the sum can be extended to the sum over all the basis vectors

$$|v_0><v_0|\varphi> + |v_1><v_1|\varphi> + \cdots + |v_{n-1}><v_{n-1}|\varphi>$$

Formally, if an operator composed of the sum of all the single projection operators for a given set of orthonormal basis states acts on a vector v_i, the resulting vector is

$$\begin{aligned}|\varphi> &= \sum_{i=0}^{n-1} < v_i \varphi > \ v_i > \\ &= \sum_{i=0}^{n-1} < v_i \ \varphi > v_i > \\ &= \sum_{i=0}^{n-1} v_i >< v_i \ \varphi > \\ &= \left(\sum_{i=0}^{n-1} |v_i >< v_i|\right) |\ \varphi>\end{aligned}$$

Thus

$$\sum_{i=0}^{n-1} |v_i >< v_i| = I$$

where I is the identity operator. This is formally known as the *completeness relation* for the chosen orthonormal basis; it indicates that the sum of the projections of a vector in all possible directions returns the original. The set $|v_i >< v_i|$ acts as an operator in the direction $|v_i>$ and is called *projection operator* P_v [4].

Note that $P_v^2 = P_v P_v = |v >< v|\ |v >< v| = |v >< v| = P_v$

There are two important properties of a projection operator:

1. It is Hermitian, that is, $P = P^{\dagger}$.
2. A projection operator is equal to its own square.

References

1. J. W. Van Orden, Quantum mechanics lecture notes, Old Dominion Uiversity, August 21, 2007.
2. B. Zwiebach, Dirac's Bra and ket notations, Lecture note for Quantum Physics II, MIT, Fall 2013.
3. Intermediate Quantum Mechanics, Lecture 6 (Notes), Rutgers University, Fall 2012.
4. Quantum Physics, UCSD Physics 130, April 2, 2003, page 183. *https://quantum-mechanics.ucsd.edu/ph130a/130_notes/node246.html.*

CHAPTER 4

Boolean Algebra, Logic Gates, and Quantum Information Processing

Boolean algebra uses symbols to represent statements or propositions. A *proposition* may be either true or false but not both. Boolean Algebra uses symbol 1 to represent a proposition that is true, and symbol 0 to represent the inverse of the proposition that is false. If symbols, *a, b, c,...* stand for elementary propositions that can be either true or false, then the logic connectives: *conjunction* (AND), *disjunction* (OR), and *negation* (NOT), can be used to combine these propositions to make *complex* propositions.

4.1 Boolean Algebra

Boolean algebra [1] is defined for a set $\mathcal{A}$ in terms of two binary operations $\cdot$ and $+$. The symbols $\cdot$ and $+$ are called the AND and *inclusive* OR, respectively. The operations in Boolean Algebra are based on the following *axioms* or *postulates*:

1. If $x, y \in \mathcal{A}$, then $x + y \in \mathcal{A}$; $x \cdot y \in \mathcal{A}$
   This is known as the *closure property*.
2. If $x, y \in \mathcal{A}$, then $x + y = y + x$; $x \cdot y = y \cdot x$
   that is, $+$ and $\cdot$ operations are *commutative*

3. If $x, y, z \in \mathcal{A}$, then

$$x + (y \cdot z) = (x + y) \cdot (x + z)$$
$$x \cdot (y + z) = (x \cdot y) + (x \cdot z)$$

that is, + and · operations are *distributive.*

4. Identity elements, denoted as 0 and 1 must exist such that $x + 0 = x$ and $x \cdot 1 = x$ for all elements of $\mathcal{A}$.
5. For every element x in $\mathcal{A}$ there exists an element x', called the *complement* of x such that

$$x + x' = 1 \qquad x \cdot x' = 0$$

Note that the basic postulates are grouped in pairs. One postulate can be obtained from the other by simply interchanging all OR and AND operations, and the identity elements 0 and 1. This property is known as *duality.*

There are several theorems that can be used for manipulating Boolean functions.

Theorem 1. The identity elements 0 and 1 are unique.

Theorem 2. The *idempotent laws*

i. $x + x = x$ *ii.* $x \cdot x = x$

Theorem 3.

i. $x + 1 = 1$ *ii.* $x \cdot 0 = 0$

Theorem 4. The *absorption laws*

i. $x + xy = x$ *ii.* $x \cdot (x + y) = x$

Theorem 5. Every element in $\mathcal{A}$ has a unique complement

Theorem 6. *Involution law*

$(x')' = x$

Theorem 7.

i. $x + x'y = x + y$ *ii.* $x(x' + y) = xy$

Theorem 8. *DeMorgan's law*

i. $(x + y)' = x' \cdot y'$ *ii.* $(xy)' = x' + y'$

If a 1 is used to denote a true proposition and a 0 is used to denote a false proposition, then the AND (·) combination of two propositions can be written as follows:

$$0 \cdot 0 = 0$$
$$0 \cdot 1 = 0$$
$$1 \cdot 0 = 0$$
$$1 \cdot 1 = 1$$

The AND combinations of two propositions are known as the *product* of two propositions. The OR combinations of two statements known as the *sum* of two propositions, can be written as follows:

$$0 + 0 = 0$$
$$0 + 1 = 1$$
$$1 + 0 = 1$$
$$1 + 1 = 1$$

The NOT operation of a statement is true if and only if the operation is false. The NOT operation also known as *complementation* or *negation* can be stated as follows:

$$(0)' = 1$$
$$(1)' = 0$$

An AND gate implements the logical *product* operation and an OR gate implements the logical *sum* operation. The complementation is performed by an inverter (NOT) gate. Any complex proposition can be decomposed into a collection of elementary propositions, each of which can then be realized by an appropriate gate and connected together as implied in the proposition. Thus, a complex proposition can be converted into a Boolean circuit using *gates* and *wires*. The gates perform simple logic operations and the wires carry information around the circuit. A common basis for Boolean circuits is the set {AND, OR, NOT}, from which all other Boolean functions can be constructed. In addition to the operators AND, OR, and NOT, three other operators can be added to the set of elementary logic operators: *identity, fanout,* and *exchange.* The identity operator does not change any bit it operates on, that is, a 0 remains a 0 and a 1 remains a 1. Thus the identity operator can be considered as a wire that takes a signal from one place to another. *Fanout* splits a signal into two identical copies of itself. *Exchange* swaps two input signals when they are not equal.

4.2 Classical Circuit Computation Model

Several models have been investigated over the years for the study of classical computation, namely, Turing machines, high-level programming languages, and Boolean circuits [2]. The Boolean circuit model is not only the most appropriate because logic circuits are the basic building blocks of real-world computers, it is also the easiest to generalize for the study of quantum computation. The Boolean circuit model can be represented by the block diagram of Fig. 4.1.

It shows f as a function of n variables $(x_1, x_2, \ldots, x_n)$. If each variable can independently assume either a true (1) or a false (0) value then they are known as binary variables, and the function is referred to as a *Boolean function* of n variables. A classical circuit models of computation evaluates function f and produces an m-bit output. Thus, it computes a binary function

$$f: (0, 1)^n \rightarrow (0, 1)^m$$

that maps its n input variables to the values of its m outputs.

A Boolean function can be described by a truth table. Since each variable can be a 0 or a 1, there can be 2^n combinations of values for n variables. For each combination of values, a function can have a value of either 0 or 1. A truth table displays the value of a function for all possible 2^n combinations of its variables. Figure 4.2 shows the truth table for the function

A small set of circuit elements known as *logic gates* can be used to implement Boolean functions; this set is called the *basis*. The most common basis contains the following three gates: AND, OR, and NOT; they are sufficient to realize any Boolean function of the form shown in Fig. 4.1. Each gate maps its inputs to a 1-bit output, that is, $n = 2$, $m = 1$ in Fig. 4.1:

$$f: (0, 1)^2 \rightarrow (0, 1)^1$$

The AND gate produces a 1 output if and only if all input variables are 1s. Figure 4.3 shows a two-input AND gate with four possible input combinations.

Figure 4.4 shows a two input OR gate. The OR gate produces a 0 output only if all the inputs are 0.

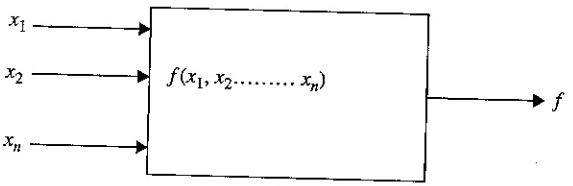
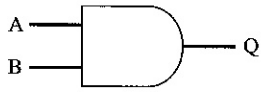


Figure 4.1 A function of n variables.

$f(a, b, c) = ab + bc + ac$

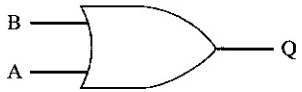
a	b	c	$f(a, b, c)$
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

FIGURE 4.2 Truth table for $f(a, b, c) = ab + bc + ac$.



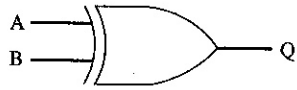
A	B	Q
0	0	0
0	1	0
1	0	0
1	1	1

FIGURE 4.3 AND gate.



A	B	Q
0	0	0
0	1	1
1	0	1
1	1	1

FIGURE 4.4 OR gate.



A	B	Q
0	0	0
0	1	1
1	0	1
1	1	0

FIGURE 4.5 EX-OR gate.

A variant of the OR gate known as EXCLUSIVE-OR or XOR gate, has also been found to be very useful. The only difference between XOR and the conventional OR gate is that XOR produces an output 0 when both inputs are 1. Figure 4.5 shows the symbol and the truth table of the XOR gate.

The NOT gate produces an output of 1 when the input is 0, and an output of 0 when the input is 1. Figure 4.6 shows the symbol for the NOT gate. It is sometimes referred to as an *inverting buffer* or simply an *inverter*.

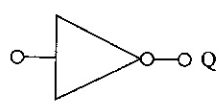
4.3 Universal Logic Gates

A set of gates is considered *universal* if every Boolean function can be implemented using only gates in this set; thus universal gates are *functionally complete*. For example, the following sets of gates are universal

(AND, NOT), (OR, NOT), (AND, XOR)

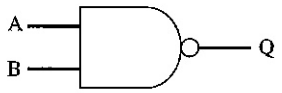
Since AND and NOT can be combined into a single gate, that is, NAND (Fig. 4.7*a*), and also OR and NOT can be combined into a NOR gate (Fig. 4.7*b*); both NAND and NOR gates are universal.

The truth table and the symbol for the NAND and the NOR gate are shown in Fig. 4.7. The bubble on the output in the graphic symbol in Fig. 4.7*a* and 4.7*b* denotes a complement operation that is performed on the output of the gates.



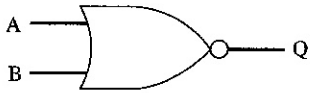
A	Q
0	1
1	0

FIGURE 4.6 NOT (inverter).



A	B	Q
0	0	1
0	1	1
1	0	1
1	1	0

(a)



A	B	Q
0	0	1
0	1	0
1	0	0
1	1	0

(b)

FIGURE 4.7 (*a*) NAND, (*b*) NOR.

4.4 Quantum Computation

Quantum computers utilize certain unique properties of subatomic particles in conjunction with the theories of computer science to process and store information. This merging of quantum mechanics and computer science has been extensively explored during the last three decades, and has led to the development of techniques for a class of computational problems, for example, deciphering codes, factoring large numbers, searching an unsorted collection etc. that can be solved much more efficiently using a quantum computer.

Such advances in information processing capability can be attributed to the fact that the data bits in a quantum computer, unlike their counterparts in classical computers can simultaneously exist in more than one state at a time and can be manipulated simultaneously.

Information in conventional digital representation uses a sequence of *bits*. Each bit is basically the charge of an electron. If the electron is charged, the bit is assumed to carry a value 1; alternatively the bit carries a value 0 if the electron is not charged. Thus a bit also known as a *classical bit* can be in state 0 or state 1, and measuring a bit at any time results in one of two possible outcomes.

4.5 The Quantum Bit and Its Representations

In quantum computing systems as in classical systems, two distinguishable states of the system are needed to represent a single bit of data. For example, consider the electron in a hydrogen atom. It can be in its ground state or in an excited state as depicted in Fig. 4.8.

If this were a classical system, it could be assumed as shown in the figure the *excited* state represent a $|1>$ and the *ground* state represent a $|0>$. In general, the electron is a quantum system, may exist in a linear superposition of the ground and excited state. It is in the ground state (0) with probability amplitude α and in the

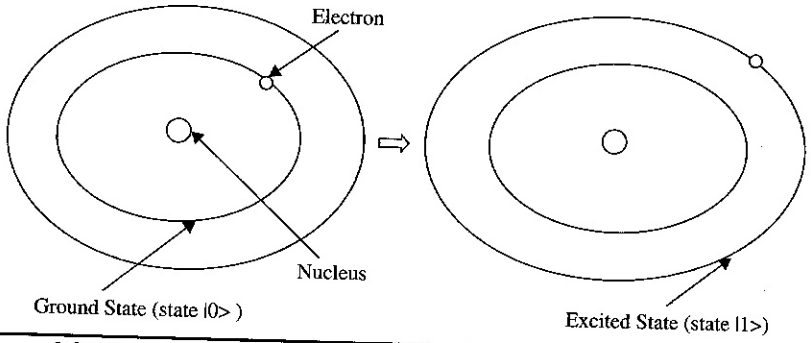


Figure 4.8 Electron in a hydrogen atom.

excited state (1) with probability amplitude β. Such a two-state quantum system is referred to as a *qubit*, and its actual state ψ can also be any linear combination (or *superposition*) of these basis states.

The state space of a qubit can be visualized by using an imaginary sphere (Fig. 4.9) known as the *Bloch sphere*. It has a unit radius. The arrow on the sphere represents the state of the qubit. Its north and south poles are selected to represent the basis states $|1>$ and $|0>$, respectively; the other locations are superpositions of $|0>$ and $|1>$. While the state of a classical bit can be either the north and the south pole of the equator, a qubit can be any point on the sphere.

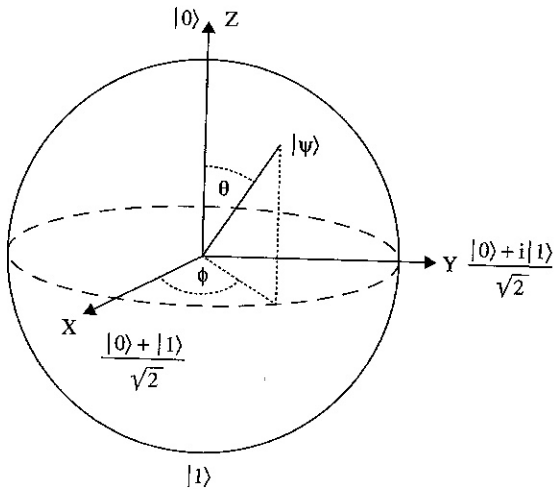


Figure 4.9 A qubit represented as a Bloch sphere.

The Bloch sphere allows the state of a qubit to be represented using unit spherical coordinates, for example, the polar angle θ and the azimuth angle ϕ. The Bloch sphere representation of a qubit is

$$|\psi> = \cos\frac{\theta}{2}\,|0> + e^{i\varphi}\sin\frac{\theta}{2}\,|1>$$

where $0 \le \theta < \pi$ and $0 \le \phi < 2\pi$.

The normalization constraint is

$$\left|\cos\frac{\theta}{2}\right|^2 + \left|\sin\frac{\theta}{2}\right|^2 = 1$$

Note that $|\psi> = |0>$ when $\theta = 0$, and $|\psi> = |1>$ when $\theta = \pi$, regardless of ϕ. In the Bloch sphere representation a qubit can not only be in either the north or the south pole of the sphere but also in states that are blend of these two states. In other words, a qubit can exist in multiple states simultaneously. This is basically the essence of the *principle of superposition* that happens because of the wave nature of subatomic particles.

A qubit can be physically implemented by two states of an electron orbiting a hydrogen atom (as shown in Fig. 4.8), by the spin-1/2 system with the two states $|\uparrow>$ and $|\downarrow>$, by the horizontal and the vertical polarizations of a photon or by any other two-state quantum system. A qubit responds in the same way as a classical bit when measured, that is, produces an output 0 or 1.

A unique property that makes quantum computing so special and offers such an unparallel potential, is that qubits unlike classical bits can also work with the overlap of both 0 and 1 states. For example, a 4-bit (classical) register can store one number from 0 to 15 at a time, whereas a 4-qubit register can store all 16 numbers in a superposition. All values in a qubit register can be simultaneously accessed and operated on, thus allowing truly parallel computation. A quantum state in superposition can be written as a linear combinations of $|0>$ and $|1>$

$$|\psi> = \alpha\,|0> + \beta\,|1> \tag{4.1}$$

where $|\psi>$ is the state of the qubit and $|0>$ and $|1>$ are the *computational basis states*. The coefficients α and β are complex numbers, they are called *probability amplitudes*; the actual probabilities are given by the absolute value squared of the associated amplitude. That is, if α is the probability amplitude of 0 state then the probability of the qubit being in 0 state is $\alpha\alpha^* = |\alpha|^2$, where α^* is the complex conjugate of α. Similarly the probability of the qubit being in 1-state is $|\beta|^2$. Normalization requires that the sum of the probabilities must be 1:

$$|\alpha|^2 + |\beta|^2 = 1$$

The quantum state ψ in Eq. (4.1) can be written as a *unit column* vector in a two-dimensional complex plane spanned by the two basis states. Figure 4.10 illustrates the two basis states; these states are called *normal basis*. Note that vectors $|0>$ and $|1>$ are *orthogonal*, that is, perpendicular to each other. A qubit with state $|0>$ is represented by the column vector $\begin{pmatrix}1\\0\end{pmatrix}$ and a qubit with state $|1>$ is represented by the column vector $\begin{pmatrix}0\\1\end{pmatrix}$, that is

$$|0> = \begin{pmatrix}1\\0\end{pmatrix} \qquad |1> = \begin{pmatrix}0\\1\end{pmatrix}$$

Thus

$$\begin{aligned}\psi &= \alpha|0> + \beta|1> \\ &= \alpha\begin{pmatrix}1\\0\end{pmatrix} + \beta\begin{pmatrix}0\\1\end{pmatrix} \\ &= \begin{pmatrix}\alpha\\0\end{pmatrix} + \begin{pmatrix}0\\\beta\end{pmatrix} \\ &= \begin{pmatrix}\alpha\\\beta\end{pmatrix}\end{aligned}$$

In other words, an arbitrary qubit state is represented by the vector $\begin{pmatrix}\alpha\\\beta\end{pmatrix}$.

Figure 4.10 illustrates the two basis states; these two states are called *normal basis*. Note that vectors $|0>$ and $|1>$ are *orthogonal* that is, perpendicular to each other. The arrow in the diagram is the hypotenuse of the right-angled triangle formed by $|0>$ and $|1>$, and the square of the hypotenuse equals the sum of the squares of the vertical and horizontal sides. Since this sum has the value one so the hypotenuse has the length one. Thus given an arrow of unit length its projection on the vertical and horizontal directions gives a pair of numbers, the sum of the squares of these numbers is 1. In other words, the arrow provides

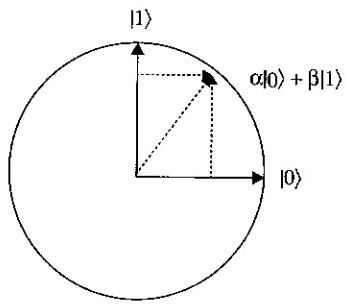


Figure 4.10 Basis states of a qubit.

all the necessary information about state of the configuration and is called the *state vector* [3].

Two other states can be derived from Eq. (4.1) by assuming $\alpha = \frac{1}{\sqrt{2}}$ and $\beta = \frac{1}{\sqrt{2}}$

$$| + > = \frac{1}{\sqrt{2}} | 0 > + \frac{1}{\sqrt{2}} | 1 >$$

$$| - > = \frac{1}{\sqrt{2}} | 0 > - \frac{1}{\sqrt{2}} | 1 >$$

$| + >$ and $| - >$ also form a computational basis. The basis states $| 0 >$ and $| 1 >$ in Eq. (4.1) can be rewritten as

$$| 0 > = \frac{1}{\sqrt{2}} | + > + \frac{1}{\sqrt{2}} | - >$$

$$| 1 > = \frac{1}{\sqrt{2}} | + > - \frac{1}{\sqrt{2}} | - >$$

Hence, a qubit state in Eq. (4.1) can be expressed using basis $| + >$ and $| - >$.

As stated earlier classical bit can only be in a single state whereas a *qubit* cannot only be in one of the two discrete states, it can also exist simultaneously in a blend of some of these states. The proportions of $| 0 >$ and $| 1 >$ in the blend need not be equal, and can be arbitrary. Thus an infinite number of possible combinations of $| 0 >$ and $| 1 >$ is possible in a qubit provided the constraint: $| \alpha |^2 + | \beta |^2 = 1$ is satisfied. Thus, in principle, it is possible to store a vast amount of information on a single qubit but it is impossible to retrieve the information. When the value in a qubit is measured, it returns $| 0 >$ with probability α^2 or it returns $| 1 >$ with probability β^2, and then the qubit assumes the state just returned.

4.6 Superposition in Quantum Systems

Superposition is a fundamental principle of quantum physics. It states that all states of a quantum system may be superimposed, that is, combined together like waves in classical physics to yield a *coherent* quantum state that is distinct from its component states. The state however collapses into a random state once it is measured.

For example, assume an electron as a qubit with *spin-up* orientation representing state $| 0 >$ and *spin-down* state $| 1 >$. However, unlike a classical bit that can only be in a single state at any time, a *qubit* can be in state *up*, *down*, or a combination of both states at the same time because of the wave-like characteristics of subatomic particles.

A qubit in superposition behaves as if it were in both $|0>$ and $|1>$ states simultaneously. This new state $|\psi>$ of the qubit can be written as:

$$|\psi> = \alpha|\uparrow> + \beta|\downarrow> = \alpha|0> + \beta|1>$$

where α and β are complex numbers and are known as *probability amplitudes* as indicated earlier, satisfying the relation

$$\alpha^2 + \beta^2 = 1$$

This indicates that a qubit has the probability α^2 of being in *spin-up* (classical 0-state) and has probability β^2 of being in state *spin-down* (classical 1-state); it can also be in a coherent superposition of both. Thus, $|\psi>$ can be considered as a vector in the two-dimensional complex vector space C^2 spanned by two basis states $|0>$ and $|1>$. In other words, a quantum bit can be in all of its positions at the same time.

It should be clear from the above that the major advantage of a qubit over its classical counterpart is that an operation on a qubit while it is in a superposed state, can simultaneously affect both of its values. However, when a qubit in superposition is measured it irreversibly collapses into either 0 or 1 state, thereby destroying the superposition. After that if the qubit is measured again it gives the same result. This implies that no additional information can be obtained by repeating the measurement. It should be mentioned here that whether a particle is in two places at the same time can never in practice be observed, only a measurement can identify one state or another.

4.7 Quantum Register

A quantum register is composed of a number of qubits; the size of the register is determined by the number of qubits. For example, a quantum register of size 4 can store individual number from 0 to 15. At any particular time the 4 qubits can be in any one of 16 possible configurations:

$$0000, 0001, 0010, \ldots\ldots\ldots\ldots\ldots 1111$$

Thus, a 4-qubit register can be represented in a superposition of the above 16 states:

$$|\psi> = c_0|0000> + c_1|0001> + c_2|0010>, \ldots\ldots\ldots + c_{14}|1110> + c_{15}|1111>$$

where the numbers $c_0, c_1, c_2, \ldots \ldots, c_{15}$ are complex coefficients such that

$$|c_0|^2 + |c_1|^2 + |c_2|^2 + \ldots \ldots \ldots \ldots \ldots \ldots + |c_{15}|^2 = 1$$

A unique advantage of quantum system is that a linear increase in the number of qubits in a register leads to the exponential growth in the state space of the register. The state of a quantum register with m qubits can be represented as a 2^m-dimensional vector in complex vector space. Since all the states of a quantum register can be in all of its states at the same time; this allows parallel processing capability to solve certain problems in a time that is many-fold faster than that is possible in classical computers.

References

1. Parag K. Lala, *Principles of Modern Digital Design*, John Wiley and Sons, 2007.
2. Ryan O'Donnell, Lecture 1: Introduction to the Quantum Circuit Model, CMU Quantum Computation. Carnegie-Mellon University, Pittsburgh, Fall 2015.
3. Steven Weinberg, *Dreams of a Final Theory*, Prentice Hall, 1992.

CHAPTER 5

Quantum Gates and Circuits

In quantum circuits, gates are mathematically represented as transformation matrices or linear operators that must be *unitary*. This is a required condition because the norm of a system wave function must be equal to 1 at all times; unitary transformations preserve the norm. Each unitary transformation U has an inverse transformation $U^{-1} = U^{\dagger}$, where $U^{\dagger}$ is the conjugate of U. Thus, quantum computations are reversible.

All computations are performed by applying a succession of these unitary matrices on single or multi-qubit systems. A single qubit is represented as a 2×1 matrix, 2 qubits as a 4×1 matrix, and 3 qubits as a 8×1 matrix. If a gate acts on a single qubit then it is called a *single-qubit gate;* a 2-qubit gate and a 3-qubit code can be defined in a similar manner. Every unitary operator (U) is represented by a 2×2 matrix, in fact every unitary operator is valid single-qubit gate [1,5].

5.1 *X* Gate

It is the quantum equivalent of a classical NOT gate, that is, if $|k>$ is the input to an *X* gate, the output of the gate is $|k'>$. Since the states of a qubit $|0>$ and $|1>$ are represented by the column vectors

$$\begin{pmatrix}1\\0\end{pmatrix} \quad \text{and} \quad \begin{pmatrix}0\\1\end{pmatrix}$$

respectively, therefore

$$X\,|0> \quad \begin{pmatrix}0 & 1\\1 & 0\end{pmatrix}\begin{pmatrix}1\\0\end{pmatrix}=\begin{pmatrix}0\\1\end{pmatrix}=|1>$$

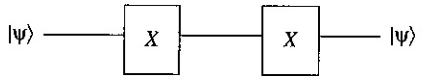
$$X\,|1> \quad \begin{pmatrix}0 & 1\\1 & 0\end{pmatrix}\begin{pmatrix}0\\1\end{pmatrix}=\begin{pmatrix}1\\0\end{pmatrix}=|0>$$

the *X* gate is also called *bit-flip* gate because it inverts each input bit.

The matrix for the X gate is shown below and is the same as Pauli matrix σ_x; it is called X gate because of this reason.

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

Suppose two X gates are connected in series to form a quantum circuit as shown below



A line in the circuit is considered as a *quantum wire* and basically represents a single qubit, thus an input at the input of the first X gate is transformed to $X|\psi>$ and the second X gate acts on it to form $XX|\psi>$. Replacing each X by its matrix representation results in an *identity matrix I:*

$$X \cdot X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = I$$

Thus two NOT gates in series is basically equivalent to a *quantum wire,* that is, nothing happens and the output is the same as the original input.

The matrix representation of the state of a superposed qubit $\alpha|0> + \beta|1>$ is given by

$$\begin{pmatrix} \alpha \\ \beta \end{pmatrix}$$

Therefore, if a superposed qubit goes through an X gate, the result will be

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} \beta \\ \alpha \end{pmatrix} = \alpha|1> + \beta|0>$$

It should be mentioned here that the X gate "negates" the computational basis states $|0>$ and $|1>$ correctly, it cannot correctly negate an arbitrary superposition state.

5.2 *Y* Gate

This gate is represented by the Pauli matrix σ_y. It maps $|0>$ to $i|1>$ and $|1>$ to $-i|0>$.

$$Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$$

$$Y|0> \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}\begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ i \end{pmatrix} = i|1>$$

$$Y|1> \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}\begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} -i \\ 0 \end{pmatrix} = -i|0>$$

Therefore, the matrix Y defines the transformation:

$$Y(\alpha|0> + \beta|1>) = \alpha Y|0> + \beta Y|1> = i\alpha|1> - i\beta|0>$$

5.3 Z Gate

This gate maps input $|k>$ to

$$(-1)^k |k>$$

Thus for an input $|0>$ the output of the Z gate is not changed, that is, also $|0>$ and for an input $|1>$ the output is $-|1>$.

It should be clear from the above definition that the matrix for the Z gate can be written as

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

that is, the same as the Pauli matrix σ_z.

The mapping of $|0>$ and $|1>$ using the matrix for the Z gate shown below:

$$Z|0> = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} = |0>$$

$$Z|1> = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ -1 \end{pmatrix} = -|1>$$

Thus the matrix Z define the transformation:

$$Z(\alpha|0> + \beta|1>) = \alpha Z|0> + \beta Z|1> = \alpha|0> - \beta|1>$$

$$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} \alpha \\ -\beta \end{pmatrix} = \alpha|1> + \beta|0>$$

Due to this nature, this transformation is sometimes called a *phase-flip*.

5.4 $\sqrt{NOT}$ (Square Root of NOT) Gate

A square-root NOT gate is a 1-qubit gate that is designed to implement the expression:

$$\sqrt{\text{NOT}} \cdot \sqrt{\text{NOT}} = \text{NOT}$$

There is no such operation in classical logic. The $\sqrt{\text{NOT}}$ gate is a good example of how a gate can exist even though Boolean algebra cannot be used to describe its operation.

A $\sqrt{\text{NOT}}$ gate can be represented by the following matrix:

$$\sqrt{\text{NOT}} = \frac{1}{\sqrt{2}} \begin{vmatrix} 1 & -1 \\ 1 & 1 \end{vmatrix}$$

When a $|0>$ is applied to a $\sqrt{\text{NOT}}$ gate, the output is

$$\sqrt{\text{NOT}}\,|0> = \frac{1}{\sqrt{2}} \begin{vmatrix} 1 & -1 \\ 1 & 1 \end{vmatrix} \begin{vmatrix} 1 \\ 0 \end{vmatrix} = \frac{1}{\sqrt{2}} \begin{vmatrix} 1 \\ 1 \end{vmatrix} = \frac{1}{\sqrt{2}} \left(\begin{vmatrix} 1 \\ 0 \end{vmatrix} + \begin{vmatrix} 0 \\ 1 \end{vmatrix} \right)$$

$$= \frac{1}{\sqrt{2}} (|0> + |1>)$$

Similarly, when a $|1>$ is applied to a $\sqrt{\text{NOT}}$ gate, the output is

$$\sqrt{\text{NOT}}\,|1> = \frac{1}{\sqrt{2}} \begin{vmatrix} 1 & -1 \\ 1 & 1 \end{vmatrix} \begin{vmatrix} 0 \\ 1 \end{vmatrix} = \frac{1}{\sqrt{2}} \begin{vmatrix} -1 \\ 1 \end{vmatrix} = \frac{1}{\sqrt{2}} \left(\begin{vmatrix} 0 \\ 1 \end{vmatrix} - \begin{vmatrix} 1 \\ 0 \end{vmatrix} \right)$$

$$= \frac{1}{\sqrt{2}} (|1> - |0>)$$

Thus when a qubit at state $|0>$ or state $|1>$ is applied to a $\sqrt{\text{NOT}}$ gate, it leaves the qubit in an equal superposition of states $|0>$ and $|1$. If the output is applied to another $\sqrt{\text{NOT}}$ gate, a superposition of $|0>$ and $|1>$ states will be the input to the second $\sqrt{\text{NOT}}$ gate since the unmeasured output of the first gate cannot be assigned any definite value. Thus, the output of the second gate when two $\sqrt{\text{NOT}}$ gates are connected in series can be derived as follows from the unmeasured output states of $\sqrt{\text{NOT}}\,|0>$ and $\sqrt{\text{NOT}}\,|1>$.

When the output of the first $\sqrt{\text{NOT}}$ gate $\frac{1}{\sqrt{2}}(|0> + |1>)$, the output of the second $\sqrt{\text{NOT}}$ gate will be

$$\frac{1}{\sqrt{2}} \begin{vmatrix} 1 & -1 \\ 1 & 1 \end{vmatrix} \begin{vmatrix} 1 \\ 0 \end{vmatrix} \frac{1}{\sqrt{2}} (|0> + |1>)$$

$$= \frac{1}{\sqrt{2}} \begin{vmatrix} 1 & -1 \\ 1 & 1 \end{vmatrix} \begin{vmatrix} 1 \\ 0 \end{vmatrix} \frac{1}{\sqrt{2}} \left[\begin{pmatrix} 1 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right]$$

$$= \frac{1}{2} \begin{pmatrix} 1 \\ 1 \end{pmatrix} + \frac{1}{2} \begin{pmatrix} -1 \\ 1 \end{pmatrix}$$

$$= \frac{1}{2} \left[\begin{pmatrix} 0 \\ 1 \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \end{pmatrix} \right] + \frac{1}{2} \left[\begin{pmatrix} 0 \\ 1 \end{pmatrix} - \begin{pmatrix} 1 \\ 0 \end{pmatrix} \right]$$

$$= \begin{pmatrix} 0 \\ 1 \end{pmatrix} = |1>$$

Similarly if the output of the first $\sqrt{\text{NOT}}$ gate is $\frac{1}{\sqrt{2}}(|1\rangle - |0\rangle)$, the output of the second $\sqrt{\text{NOT}}$ gate is

$$\frac{1}{\sqrt{2}}\begin{vmatrix} 1 & -1 \\ 1 & 1 \end{vmatrix}\begin{vmatrix} 1 \\ 0 \end{vmatrix}\frac{1}{\sqrt{2}}(|1\rangle - |0\rangle)$$

$$= \frac{1}{\sqrt{2}}\begin{vmatrix} 1 & -1 \\ 1 & 1 \end{vmatrix}\frac{1}{\sqrt{2}}\left[\begin{pmatrix} 0 \\ 1 \end{pmatrix} - \begin{pmatrix} 1 \\ 0 \end{pmatrix}\right]$$

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

$$= \frac{1}{2}\left[\begin{pmatrix} 0 \\ 1 \end{pmatrix} - \begin{pmatrix} 1 \\ 0 \end{pmatrix}\right] - \frac{1}{2}\left[\begin{pmatrix} 0 \\ 1 \end{pmatrix}\begin{pmatrix} 1 \\ 0 \end{pmatrix}\right]$$

$$= \frac{1}{2}\left[\begin{pmatrix} 0 \\ 1 \end{pmatrix} - \begin{pmatrix} 1 \\ 0 \end{pmatrix} - \begin{pmatrix} 0 \\ 1 \end{pmatrix} - \begin{pmatrix} 1 \\ 0 \end{pmatrix}\right]$$

$$= -\frac{1}{2}\cdot 2\begin{pmatrix} 1 \\ 0 \end{pmatrix} = -1\,|0\rangle = |0\rangle$$

The quantum NOT gate operates exactly like its classical counterpart. If the NOT gate is implemented using two unknown gates in series then these two gates can be assumed to performing as $\sqrt{\text{NOT}}$ gates. Thus the logic operation of the NOT gate can be written as

$$\sqrt{\text{NOT}} \cdot \sqrt{\text{NOT}} = \text{NOT}$$

However, no single input and single output classical gate can reproduce the function of a traditional NOT gate when connected in series with another similar gate. In other words, a $\sqrt{\text{NOT}}$ gate is a truly nonclassical gate.

5.5 Hadamard Gate

The Hadamard gate is a truly quantum gate and is one of the most important in quantum computing. It has some similar characteristics to the $\sqrt{\text{NOT}}$ gate. However, the Hadamard gate, unlike the $\sqrt{\text{NOT}}$ gate, is *self-inverse*. It maps input $|m\rangle$ to

$$H\,|m\rangle = \frac{1}{\sqrt{2}}\sum_{k=0,1}(-1)^{mk}\,|k\rangle$$

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

The effect of the Hadamard gate on the computational basis states $|0>$ and $|1>$ are as follows, assuming $m = 0$ and 1, respectively:

$$H|0> = H\begin{pmatrix}1\\0\end{pmatrix} = \frac{0> + (-1)^0|1>)}{\sqrt{2}}$$

$$= \frac{0> + |1>)}{\sqrt{2}} = |+>$$

$$H|1> = H\begin{pmatrix}0\\1\end{pmatrix} = \frac{|0> + (-1)^m|1>)}{\sqrt{2}}$$

$$= \frac{|0> - |1>)}{\sqrt{2}} = |->$$

Hence the matrix for the Hadamard gate can be defined as

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

Notice that a Hadamard gate converts a $|0>$ state and a $|1>$ to a superposition state of $|0>$ and $|1>$. When the superposed qubit is measured, there is an equal probability of it being in the state $|1>$ or $|0>$.

A Hadamard gate acts on a superposed state $\alpha|0> + \beta|1>$ linearly:

$$H(\alpha|0> + \beta|1>) = H\alpha|0> + H\beta|1>$$

$$= \alpha H|0> + \beta H|1>$$

$$= \alpha \cdot \frac{|0> + |1>)}{\sqrt{2}} + \beta \cdot \frac{|0> - |1>)}{\sqrt{2}}$$

$$= \frac{\alpha + \beta}{\sqrt{2}}|0> + \frac{\alpha - \beta}{\sqrt{2}}|1>$$

A quantum register of size n can not only store an individual string but all possible combinations of n qubit at the same time. A 2-qubit register, for example, can store $|01>$ or $|10>$ at any time:

$$|01> = |0> \otimes |1>$$

$$|11> = |1> \otimes |1>$$

However, the 2-qubit register can also store both strings simultaneously. This can be done by setting the first qubit to a superposition of $|0>$ and $|1>$ instead of just $|0>$ or $|1>$; this can be done by applying the qubit to a Hadamard gate:

$$(\frac{1}{\sqrt{2}}|0> + |1>) \otimes |1>$$

$$= \frac{1}{\sqrt{2}}|01> + |11>$$

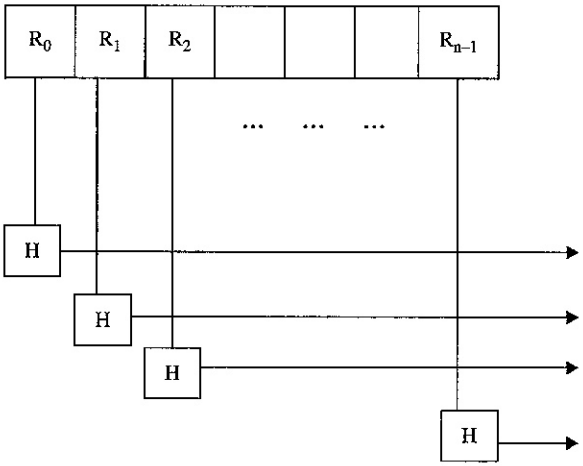


Figure 5.1 Generation of all possible bit strings corresponding to n qubits.

This can be extended to all qubits in a quantum register of size n by first preparing each qubit in the state $|0>$ then applying the Hadamard gate to each of the n qubits in parallel as shown in Fig. 5.1. The resulting state of the register is an n-qubit superposition containing 2^n component states; these states are all the possible bit strings corresponding to n qubits.

$$H|0> \otimes H|0> \otimes \cdots \otimes H|0> = \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} |k>$$

5.6 Phase Gate

This gate turns a $|0>$ into $|0>$, and a $|1>$ into $i|1>$. It is represented by the following matrix:

$$S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$$

The mapping of $|0>$ and $|1>$ using the matrix for the S gate shown below:

$$S|0> = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} = |0>$$

$$S|1> = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ -i \end{pmatrix} = i|1>$$

Thus, the matrix S defines the transformation:

$$S(\alpha|0>+\beta|1>) = \alpha S|0>+\beta S|1>$$
$$= \alpha\ |0>+i\beta|1>$$

Note that a Z gate can be obtained by connecting two S gates in series:

$$S^2 = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}\begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} = Z$$

5.7 *T* Gate

The matrix for the T gate, also known as the $\frac{\pi}{8}$ gate, is

$$T = \begin{pmatrix} 1 & 0 \\ 0 & \exp\left(\frac{i\pi}{4}\right) \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & \frac{(1+i)}{\sqrt{2}} \end{pmatrix}$$

Note that an S gate can be formed by connecting two T gates in series:

$$T^2 = \begin{pmatrix} 1 & 0 \\ 0 & \frac{(1+i)}{\sqrt{2}} \end{pmatrix}\begin{pmatrix} 1 & 0 \\ 0 & \frac{(1+i)}{\sqrt{2}} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} = S$$

5.8 Reversible Logic

Loss of energy is a major problem in digital circuits designed using classical gates. In classical gates energy dissipation results due to information loss. Landauer [2] showed that the loss of each bit of information leads to at least

$$kT\log(2)$$

amount of energy release in the form of heat, where k is the Boltzman's constant and T is temperature at which the computation is performed. Although this is a trivial amount of heat, it becomes significant in any major computing system. Such a system carries out many millions of operations per second; as a result it becomes noticeably hot in a short amount of time. The loss of information and the resulting heat generation can be avoided by replacing classical gates with *reversible gates.*

In a reversible gate it is possible to unambiguously determine bits on the inputs of the gate from the outputs of the gate. Classical gates are *irreversible;* all possible input combinations in such a gate are mapped to one of two possible outputs 0 and 1. For example, in two-input *AND gates the possible input combinations are 00, 01, 10, and 11.* The combinations 00, 01, and 10 map to output 0, only the combination 11 produces an output 1. Thus, when an AND gate produces a 0, the input could be one of the three possibilities and it is not possible to infer what the actual input was. Since in an AND gate a complete knowledge about the input cannot be determined from the output, the operation of the AND gate is not reversible. In a similar manner in all classical gates, the input values cannot be determined from the output value. The NOT gate is an exception. A straightforward way explaining this is that since in classical gates such as two-input AND, OR, NAND, NOR gates a 2-bit input maps into a 1-bit output, one-bit of information is lost on every operation and it is not possible to recover. Thus, classical gates are irreversible. Reversible gates do not lose information. Hence, a reversible gate must have exactly the same number of inputs and outputs; this allows the input value to be uniquely determined from the output value and vice versa. This is the reason a NOT gate with a single input and a single output is reversible. Bennett [3] showed that any computing system can be made reversible by replacing each classical gate in the system with its reversible equivalent. An extremely useful reversible gate for quantum computing circuits is the two-input/two-output controlled-NOT or *CNOT* gate.

5.9 CNOT Gate

A CNOT gate basically implements a reversible EX-OR. It can be used to generate entanglement. The CNOT gate can be graphically represented as in Fig. 5.2, The control and the target inputs are shown as two horizontal lines. The dependence of output y on control a is shown by a vertical line from a to one of the inputs of the EX-OR gate,

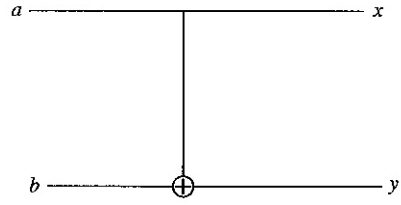


Figure 5.2 CNOT gate.

a	b	x	y
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0

TABLE 5.1 Truth Table of CNOT Gate

the other input of the gate is driven by target input b. The truth table of CNOT is shown in Table 5.1.

The input a is typically called the *source,* and input b is known as the *target.* Output $x = a$, that is, x takes the value of source s. The source is also called the *control input* and controls the application of the NOT operation on the target input. Output $y = a \oplus b$, that is, y is the inverse of target b when source is 1 otherwise $y = b$. In other words, whether y gets the inverted value of the target b (or not) is *controlled* by source a. For this reason, CNOT is known as a *controlled NOT gate.* It can be seen from the truth table that in CNOT gate the inputs can be uniquely determined from the outputs, thus verifying the reversibility of the gate and is represented by the matrix:

	00	01	10	11
00	1	0	0	0
01	0	1	0	0
10	0	0	0	1
11	0	0	1	0

As discussed above, if the target qubit is $|0>$ and the control qubit is either $|0>$ or $|1>$ then the target takes the value of the control qubit, that is, becomes a copy of the control qubit, but the control qubit itself does not change. However, a superposition in the control qubit results in the entanglement of control and target qubits. To illustrate, assume the control input is $|1>$ and the target input is $|0>$. Then the combined state of the control and the target inputs is

$$\frac{1}{\sqrt{2}}(|1>+|0>)|0>$$

$$=\frac{1}{\sqrt{2}}(|10>+|00>)$$

Thus the combined input state is a superposition of states $|10>$ and $|00>$. However, when the control input of the CNOT gate is a superposition of $|0>$ and $|1>$, that is, $\alpha|1>+\beta|0>$ and the target qubit $|0>$, the output of the gate is an entangled state that is a superposition of the output states of the two individual inputs $\frac{1}{\sqrt{2}}|10>$ and $\frac{1}{\sqrt{2}}|00>$. Thus the combined output state of the CNOT gate is

$$(\frac{1}{\sqrt{2}}|10>+\frac{1}{\sqrt{2}}|00>)$$

Since for a CNOT gate

$$|00>\rightarrow|00>,\ |10>\rightarrow|11>$$

the combined output state is

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

Note that this output state cannot be separated as a product of two single states unlike the combined input state. In other words, the output state is *entangled*.

5.10 Controlled-*U* Gate

A CNOT gate can be extended in a way that it can work on two qubits based upon a single control qubit. Assume U is a single qubit gate that can be represented with a unitary matrix:

$$U=\begin{vmatrix} u_{00} & u_{01} \\ u_{10} & u_{11} \end{vmatrix}$$

Then the *controlled-U gate* can operate on two qubits in such a way that the first qubit serves as a control. The schematic of the gate is shown in Fig. 5.3.

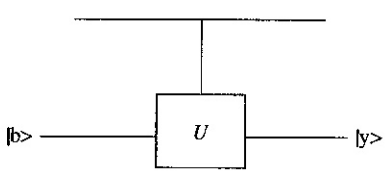


Figure 5.3 Controlled-*U* gate.

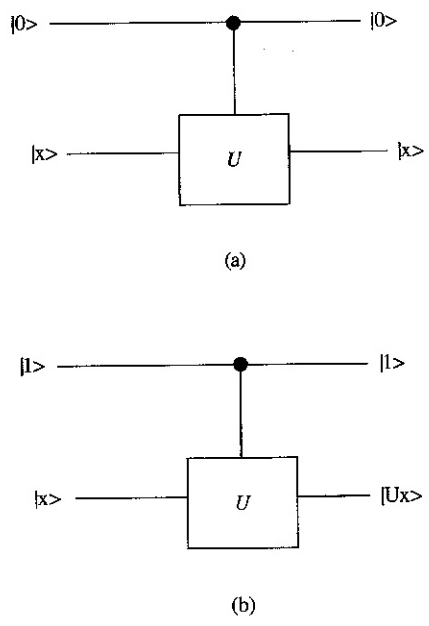


FIGURE 5.4 (*a*) $|0x\rangle \rightarrow |0x\rangle$; (*b*) $|1x\rangle \rightarrow |1, Ux\rangle$.

The outputs of the gate corresponding to the control bit $a = |0\rangle$ or $|1\rangle$ are shown in Figs. 5.4(*a*) and 5.4(*b*), respectively.

Thus the input-to-output mapping of the gate can be represented as

a	b	x	y
$\|0\rangle$	$\|0\rangle$	$\|0\rangle$	$\|0\rangle$
$\|0\rangle$	$\|1\rangle$	$\|0\rangle$	$\|1\rangle$
$\|1\rangle$	$\|0\rangle$	$\|1\rangle$	$\|1,U\rangle$ $\|0\rangle$
$\|1\rangle$	$\|1\rangle$	$\|1\rangle$	$\|1,U\rangle$ $\|1\rangle$

Since,

$$U\rangle|0\rangle = \begin{vmatrix} u_{00} & u_{01} \\ u_{10} & u_{11} \end{vmatrix} \begin{vmatrix} 1 \\ 0 \end{vmatrix} = \begin{vmatrix} u_{00} \\ u_{10} \end{vmatrix}$$

$$= (u_{00}\,|0\rangle + u_{10}\,|1\rangle)$$

Therefore,

$$|1\rangle|U\rangle|0\rangle = |1\rangle(u_{00}|0\rangle + u_{10}|1\rangle)$$

Similarly,

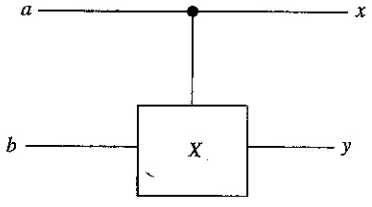
$$|U>|1> = \begin{vmatrix} u_{01} \\ u_{11} \end{vmatrix} = (u_{01}|0> + u_{10}|1>)$$

$$|1>|U>|1> = |1>(u_{01}|0> + u_{10}|1>$$

Hence, the matrix representing the controlled-U gate can be written as

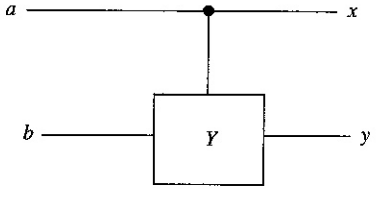
$$\begin{array}{c} \\ 00 \\ 01 \\ 10 \\ 11 \end{array} \begin{array}{c} \begin{array}{cccc} 00 & 01 & 10 & 11 \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & u_{00} & u_{01} \\ 0 & 0 & u_{10} & u_{11} \end{bmatrix} \end{array}$$

This indicates that the controlled version of a U-gate matrix can be obtained by substituting the lower two-by-two submatrix of a four-by-four identity matrix with the single qubit matrix of the chosen gate. If U is a Pauli matrix X, Y, or Z, the resulting controlled gates are identified as *controlled-X*, *controlled-Y*, and *controlled-Z* gates, respectively, as depicted in Fig. 5.5[4].



a b \ xy	00	01	10	11
00	1	0	0	0
01	0	1	0	0
10	0	0	0	1
11	0	0	1	0

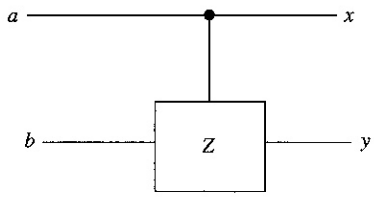
(a)



	00	01	10	11
00	1	0	0	0
01	0	1	0	0
10	0	0	0	$-i$
11	0	0	i	0

(b)

Figure 5.5 Controlled-U gate. (*a*) Controlled-X gate. (*b*) Controlled-Y gate. (*c*) Controlled-Z gate.



	00	01	10	11
00	1	0	0	0
01	0	1	0	0
10	0	0	1	0
11	0	0	0	−1

(c)

Figure 5.5 *(Continued)*

5.11 Reversible Gates

Landauer [2] showed that three-input/three-output reversible logic gates are extremely useful for classical reversible computation. Two such gates are *Fredkin gate* and *Toffoli gates;* these have been proven to be *universal gates* for irreversible computation. A universal gate allows the building of a circuit corresponding to any Boolean function. Both NAND and NOR gates have been used as universal gates in classical digital circuit design. Fredkin gates and Toffoli gates have been shown to function as NAND gates.

5.11.1 Fredkin Gate (Controlled Swap Gate)

Unlike the quantum gates discussed so far, the Fredkin gate is a three-input gate. It can be seen from the truth table of the gate shown in Fig. 5.6 that when $a = 0$, b is transferred to x and c to y. Alternatively, when $a = 1$, outputs x and y are *swapped,* that is, b and c are transferred to y and x, respectively. Because of this feature, a Fredkin gate is also known as a *controlled swap* (CSWAP) *gate.*

The mapping from input to output is

$$\begin{pmatrix} 000 \\ 001 \\ 010 \\ 011 \\ 100 \\ 101 \\ 110 \\ 111 \end{pmatrix} \rightarrow \begin{pmatrix} 000 \\ 001 \\ 010 \\ 011 \\ 100 \\ 110 \\ 101 \\ 111 \end{pmatrix}$$

a	b	c	w	x	y
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	0	0
1	0	1	1	1	0
1	1	0	1	0	1
1	1	1	1	1	1

Figure 5.6 Truth table of Fredkin gate.

Therefore, the matrix representing the gate is

	000	001	010	011	100	101	110	111
000	1	0	0	0	0	0	0	0
001	0	1	0	0	0	0	0	0
010	0	0	1	0	0	0	0	0
011	0	0	0	1	0	0	0	0
100	0	0	0	0	1	0	0	0
101	0	0	0	0	0	0	1	0
110	0	0	0	0	0	1	0	0
111	0	0	0	0	0	0	0	1

The structural representation of the gate is shown in Fig. 5.7. It can be seen from the diagram that

$$w = a$$

$$x = a'b \oplus ac$$

$$y = a'c \oplus ab$$

This gate is universal as well as *conservative;* in a conservative gate the number of 0s and 1s remain constant as signals pass through from the input to the output of the gate.

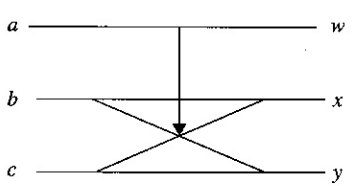


Figure 5.7 Fredkin gate.

a	b	c	w	x	y
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	0	0
1	0	1	1	0	1
1	1	0	1	1	1
1	1	1	1	1	0

Figure 5.8 Truth table for Toffoli gate.

5.11.2 Toffoli Gate (Controlled-Controlled-NOT)

A Toffoli gate also known as a CCNOT gate. It has three inputs; the outputs are the same as the inputs except the third qubit which flips only if the first two qubits are both 1s. The truth table of the Toffoli gate is shown in Fig. 5.8.

It can be seen from the truth table that the mapping from input to output is

$$\begin{pmatrix} 000 \\ 001 \\ 010 \\ 011 \\ 100 \\ 101 \\ 110 \\ 111 \end{pmatrix} \rightarrow \begin{pmatrix} 000 \\ 001 \\ 010 \\ 011 \\ 100 \\ 101 \\ 111 \\ 110 \end{pmatrix}$$

Thus, the matrix representing the Toffoli gate can be represented as shown in Fig. 5.9.

	000	001	010	011	100	101	110	111
000	1	0	0	0	0	0	0	0
001	0	1	0	0	0	0	0	0
010	0	0	1	0	0	0	0	0
011	0	0	0	1	0	0	0	0
100	0	0	0	0	1	0	1	0
101	0	0	0	0	0	1	0	0
110	0	0	0	0	0	0	0	1
111	0	0	0	0	0	0	1	0

Figure 5.9 Matrix corresponding the Toffoli gate.

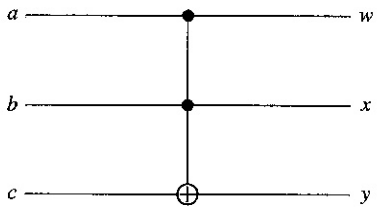


Figure 5.10 Toffoli gate.

Figure 5.10 shows the structural representation of the Toffoli gate that indicates:

$$w = a$$

$$x = b$$

$$y = ab \oplus c$$

5.11.3 Peres Gate

The Peres gate is a modified form of the Toffoli gate. The truth table of the gate is shown Fig. 5.11.

It can be seen from the truth table that the mapping from input to output is

$$\begin{pmatrix} 000 \\ 001 \\ 010 \\ 011 \\ 100 \\ 101 \\ 110 \\ 111 \end{pmatrix} \rightarrow \begin{pmatrix} 000 \\ 001 \\ 010 \\ 011 \\ 110 \\ 111 \\ 101 \\ 100 \end{pmatrix}$$

a	b	c	w	x	y
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	1	0
1	0	1	1	1	1
1	1	0	1	0	1
1	1	1	1	0	0

Figure 5.11 Truth table for Peres gate.

	000	001	010	011	100	101	110	111
000	1	0	0	0	0	0	0	0
001	0	1	0	0	0	0	0	0
010	0	0	1	0	0	0	0	0
011	0	0	0	1	0	0	0	0
100	0	0	0	0	0	0	1	0
101	0	0	0	0	0	0	0	1
110	0	0	0	0	0	1	0	0
111	0	0	0	0	1	0	0	0

Figure 5.12 Matrix for Peres gate.

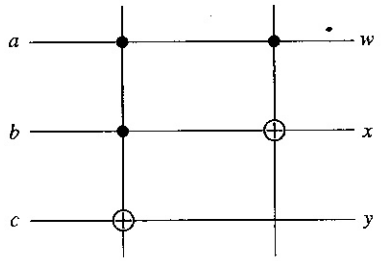


Figure 5.13 Peres gate.

Thus the matrix for the Peres gate is as shown in Fig. 5.12.

The structural representation of the gate is shown in Fig. 5.13. As can be seen in the diagram

$$w = a$$
$$x = a \oplus b$$
$$y = ab \oplus c$$

References

1. Michael Nielsen and Issac Chuang, *Quantum Computation and Quantum Information*, Cambridge University Press, 2000.
2. R. Landauer, Irreversibility and Heat Generation in the Computational Process, *IBM J.Res.& Dev.*, **5**, 183–191, 1961.
3. C. H. Bennett, Logical Reversibility of Computation, *IBM J.Res.& Dev.* **30**, 525–532, 1973.
4. Quantum logic gate, Root edit on August 1, 2018. https://en.wikipedia.org/wiki/Quantum_logic_gate#Controlled_(cX_cY_cZ)_gates.
5. David McMohan, *Quantum Computing Explained*, Wiley Interscience, 2008.

CHAPTER 6

Tensor Products, Superposition, and Quantum Entanglement

Quantum computing utilizes two major features of quantum mechanics: *superposition* and *entanglement*. Superposition, as indicated earlier, refers to the quantum phenomenon where a quantum system can exist in multiple states or places at the same time. Entanglement is an extremely strong correlation that exists between two or more quantum particles. These particles are so inextricably linked that even if separated by great distances, they change their states instantaneously in perfect unison. This might seem almost impossible, but is fundamental to the quantum world. Some knowledge of tensor products is needed to understand the concepts of superposition and entanglement.

6.1 Tensor Products

The state of a single quantum bit can be represented as a unit (column) vector in a two-dimensional vector space C^2. However, quantum information processing systems in general use multiple qubits. The joint state of such a system can only be described using a new vector space that takes into account the interaction among the qubits. This vector space is generated by using a special operation known as a *tensor product*. The tensor product, denoted by $\otimes$, combines the smaller vector spaces of the individual qubits and forms a bigger space; the elements of the bigger vector space are identified as *tensors*. A tensor product is also called *Kronecker product* or *direct product* [1, 2].

For example, the tensor product of two two-dimensional vectors $\mathbf{U} = \begin{pmatrix} x_1 \\ y_1 \end{pmatrix}$ and $\mathbf{V} = \begin{pmatrix} x_2 \\ y_2 \end{pmatrix}$ is illustrated below:

$$\mathbf{U} \otimes \mathbf{V} = \begin{pmatrix} x_1 \\ y_1 \end{pmatrix} \otimes \begin{pmatrix} x_2 \\ y_2 \end{pmatrix}$$

$$= \begin{bmatrix} x_1 \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} \\ y_1 \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} \end{bmatrix}$$

$$= \begin{bmatrix} x_1 x_2 \\ x_1 y_2 \\ y_1 x_2 \\ y_1 y_2 \end{bmatrix}$$

It has been shown in Chap. 1 that the vector representation of single qubit states $|0>$ and $|1>$ are $|0> = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ and $|1> = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$. Using these vector representations and the definition of the tensor product the two-qubit basis states can be represented as

$$|00> = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad |01> = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \quad |10> = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \quad |11> = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

In general, if $\mathbf{U}$ is an m-dimensional vector space with bases $\{g_0 \ldots \ldots g_{m-1}\}$ and $\mathbf{V}$ is an n-dimensional vector space with bases $\{h_0 \ldots \ldots h_{n-1}\}$, the tensor product $\mathbf{U} \otimes \mathbf{V}$ is a mn-dimensional vector space that is spanned by elements of the form $g \otimes h$ and the coefficients are given by $u_i v_j$ for each basis vector [2]:

$$\mathbf{U} \otimes \mathbf{V} = \sum_{i=0}^{m-1} u_i g_i \otimes \sum_{j=0}^{n-1} v_j h_j$$

$$= \sum_{i=0}^{m-1} u_i \sum_{j=0}^{n-1} v_j (g_i \otimes h_j)$$

Assuming both **U** and **V** are two-dimensional, that is, $m = n = 2$,

$$\mathbf{U} \otimes \mathbf{V} = \sum_{i=0}^{1} u_i g_i \otimes \sum_{j=0}^{1} v_j h_j$$

$$= (u_0 g_0 + u_1 g_1) \otimes (v_0 h_0 + v_1 h_1)$$

$$= u_0 v_0 (g_0 \otimes h_o) + u_0 v_1 (g_0 \otimes h_1) + u_1 v_0 (g_1 \otimes h_o) + u_1 v_1 (g_1 \otimes h_1)$$

The new vector space generated by the tensor product of two two-dimensional vectors **U** and **V** is four-dimensional and the new basis vectors are

$$(u_0 v_0, u_0 v_1, u_1 v_0, u_1 v_1)$$

This vector is shown below as a four-dimensional column vector

$$\mathbf{U} \otimes \mathbf{V} = \begin{pmatrix} u_0 v_0 \\ u_0 v_1 \\ u_1 v_0 \\ u_1 v_1 \end{pmatrix}$$

For example, if **V** is a vector space with two basis vectors $|\mathbf{u}>$ and $|\mathbf{v}>$ that correspond to two quantum bits, then the joint state of the quantum bits is

$$u> \otimes |v>$$

and it is an element of $\mathbf{U} \otimes \mathbf{V}$.

Some of the basic properties of tensor products are:

i. If A and B are operators on m and n dimensional vectors respectively, then $A \otimes B$ is an operator on $n \times m$ dimensional vector.

ii. The product of any scalar s with tensor product $A \otimes B$ is equal to

$$s(A \otimes B) = (sA) \otimes B = A \otimes (sB)$$

iii. $(A \otimes B)^* = (A^* \otimes B^*)$ (and similarly for inverse and transpose)

iv. If A is an $m \times n$ matrix and B an $p \times q$ matrix, then their tensor product is an $mp \times nq$ matrix.

v. If A, B, C, and D are matrices then

   a. $A \otimes (B + C) = (A \otimes B) + (A \otimes C)$

   b. $A \otimes (B \otimes C) = (A \otimes B) \otimes C$

that is, derivation of a tensor product is independent of the order of evaluation; thus it is an associative operation.

c. $(A \otimes B)(C \otimes D) = (AC) \otimes (BD)$

that is, the distribution law holds.

vi. If A, B are matrices, and **U**, **V**, and **W** are vectors, then

$(A \otimes B)(\mathbf{U} \otimes \mathbf{W}) = A\mathbf{U} \otimes B\mathbf{W}$

$(\mathbf{U} + \mathbf{V}) \otimes \mathbf{W} = \mathbf{U} \otimes \mathbf{W} + \mathbf{V} \otimes \mathbf{W}$

$\mathbf{U} \otimes (\mathbf{V} + \mathbf{W}) = \mathbf{U} \otimes \mathbf{V} + \mathbf{U} \otimes \mathbf{W}$

vii. If s and t are scalars, **U** and **V** are vectors, then

$s\mathbf{U} \otimes t\mathbf{V} = s\,t\,(\mathbf{U} \otimes \mathbf{V})$

6.2 Multi-Qubit Systems

A multi-qubit quantum system can be formed by putting together a number of known quantum subsystems. A subsystem based on a single qubit is generally described by its state that is defined to be a unit vector in some complex Hilbert space. The mathematical framework for describing a multi-qubit system utilizes the concept of a *tensor product* of vector spaces. The state space of the ith constituent of such a system is given by a separable Hilbert space H_i. Each Hilbert space H_i has an orthonormal basis given by

$$\{|i, k_i> k_i = 1, 2, \ldots\}, i = 1, 2, \ldots, n$$

The state space of the multi-bit system is then the tensor product space H defined by

$$H = H_1 \otimes H_2 \otimes \ldots \otimes H_n$$

The basis states of a multi-qubit system are constructed from the basis-states of a single qubit using *tensor product* of vectors. The tensor product of two vectors **u** and **v**, assuming **u** and **v** are m-dimensional and n-dimensional, respectively, has the dimension $m \times n$. Since in a single qubit system $m = 2 = n$, a two-qubit system has four basis states. These basis states are constructed from the single-qubit basis $|0>$, $|1>$ using the following rule

$$|\mathbf{u}> \otimes |\mathbf{v}> = |\mathbf{u}>|\mathbf{v}> = |\mathbf{uv}>$$

where $\mathbf{u},\mathbf{v} \in \{0, 1\}$.

For instance, if the states of the two qubits are given by

$$|\psi_1> = \alpha_0|0> + \alpha_1|1> \text{ and } |\psi_2> = \beta_0|0> + \beta_1|1>$$

then the state of the composite system is

$$|\psi> = |\psi_1> \otimes |\psi_2>$$
$$= (\alpha_0|0> + \alpha_1|1>) \otimes (\beta_0|0> + \beta_1|1>)$$

Using the following rule

$$|\mathbf{u}> \otimes |\mathbf{v}> = |\mathbf{u}>|\mathbf{v}> = |\mathbf{uv}> \qquad \text{where } \mathbf{u}, \mathbf{v} = 0, 1$$

$|\psi>$ can be written as

$$|\psi> = \alpha_0\beta_0|0>|0> + \alpha_0\beta_1|0>|1> + \alpha_1\beta_0|1>|0>$$
$$+ \alpha_1\beta_1|1>|1> \ldots\ldots\ldots\ldots (6.1)$$

Thus a two-qubit system has four basis states:

$$|0> \otimes |0>, |0> \otimes |1>, |1> \otimes |0>, |1> \otimes |1>$$

The first state in the two-qubit system, for instance,

$$|0> \otimes |0>$$

indicates that the first qubit is in state $|0>$ and the second qubit is also in state $|0>$. Similarly, for the state $|0> \otimes |1>$ the first qubit is in state $|0>$ and the second qubit is in state $|1>$.

Using the following rule

$$|\mathbf{u}> \otimes |\mathbf{v}> = |\mathbf{u}>|\mathbf{v}> = |\mathbf{uv}> \text{ where } (\mathbf{u},\mathbf{v}) \in (0, 1)$$

Expression (6.1) can be written as

$$|\psi> = \alpha_0\beta_0|0>|0> + \alpha_0\beta_1|0>|1> + \alpha_1\beta_0|1>|0> + \alpha_1\beta_1|1>|1>$$

and also as

$$|\psi> = \alpha_0\beta_0|00> + \alpha_0\beta_1|01> + \alpha_1\beta_0|10> + \alpha_1\beta_1|11>$$

It should be noted, however, that every two-qubit state cannot be separated into two single-qubit states. In general, the state of a two-qubit system has the form

$$|\psi> = c_0|00> + c_1|01> + c_2|10> + c_3|11>$$

where

$$|c_0|^2 + |c_1|^2 + |c_2|^2 + |c_3|^2 = 1$$

As indicated previously a collection of n qubits is referred to as a *quantum register* of size n. An n-qubit register has 2^n basis states, each

of the form $|c_0 \otimes |c_1 \otimes \ldots \ldots |c_{n-1}$, with $c_i \varepsilon \{0, 1\}$. A *basis state* can be represented by a number 0 to 2^{n-1}. For example, the state 1001 in a quantum register of size 4 is denoted by $|9>_4$.

A quantum register of n qubits can be in any superposition of 2^n states

$$c_0 |0> + c_1 |1> + c_2 |2> + \cdots \cdots \cdots \cdots + c_2^{n-1} |2^{n-1}>$$

However, it is not possible to retrieve the states within a superposition because this process leads to the collapse of the superposition and produce just one of the original states $|j>$ with probability $|c_j|^2$.

6.3 Superposition

The principle of superposition was introduced in Chap. 4. A quantum system with k distinguishable states can exist partly in two or more mutually exclusive states, and can also be placed in a linear superposition of these states with complex coefficients. Conversely, two or more states can be superimposed to give a new state. After measurement the system falls to one of the basis states that form the superposition, thus destroying the original configuration.

To illustrate, assume a system of two hydrogen atoms. An electron in a hydrogen atom can be regarded as a two-state quantum system since each electron can either be in the ground or in the excited state. Thus, the electrons in a system of two hydrogen atoms constitute a system of four classical states and the system can be in one of four states: 00, 01, 10, or 11.

However, unlike a classical bit which can only be in a single state at any time, a *qubit* it can not only be in one of the two discrete classical states but may also exist in more than one state at the same time due to the wave-like characteristics of subatomic particles. This is basically the essence of the *principle of superposition*; it states that if s_1 and s_2 are two *distinct* physical states, then the *complex linear superposition* of s_1 and s_2

$$\frac{1}{\sqrt{2}} |s_1> + \frac{1}{\sqrt{2}} |s_2>$$

also is a *quantum state* of the system, and

$$\frac{1}{\sqrt{2}} |s_1> - \frac{1}{\sqrt{2}} |s_2>$$

also represents a legitimate quantum state.

Thus, by the superposition principle, the quantum state of the two electron system can be in any linear combination of the four classical states:

$$|\varphi> = \alpha_{00} |0>|0> + \alpha_{01} |0>|0> + \alpha_{10} |1>|0> + \alpha_{11} |1>|1>$$

where $\sum_{i,j} |\alpha_{ij}|^2 = 1$. Note that in this case $\alpha_{ij} = \frac{1}{2}$.

The overall state can be written as a product of the individual states of the two electrons:

$$\frac{1}{2}|00>+\frac{1}{2}|01>+\frac{1}{2}|10>+\frac{1}{2}|11>$$

$$=(\frac{1}{\sqrt{2}}|0>+\frac{1}{\sqrt{2}}|1>)\otimes(\frac{1}{\sqrt{2}}|0>+\frac{1}{\sqrt{2}}|1>)$$

The fact that the overall state is *factorizable*, that is, the product of the states of the two electrons indicates that the electrons are independent of each other. This means any operation on one electron has no impact on the other. Moreover, since the overall state is an *equal* superposition of the four states, the measurement outcome will be one of these randomly and with equal probability.

As an example, consider a system with two states A and B. Suppose the observation of the system in state A gives an output *a*, and the observation in state B gives *b*. Then the observation of the system when equal percentage of A and B are in superposition, always gives an output *a* or *b* with equal probability, nothing else.

In the world of subatomic particles there may potentially be an unlimited number of different states in which the particles may be located at once. In reality, whether a particle is in an indeterminate state or in two places at the same time, can never actually be observed, only a measurement can verify this.

The working of a quantum computer relies on processing all the particles in superposition simultaneously [3]. This gives parallel processing capability to quantum computers. For example, a traditional computer can process only *one* combination of *n* bits (currently 64 bits). A quantum computer on the other side of the spectrum can process all 2^n combinations of two states at the same time. This corresponds to a conventional computer with 2^n processors. For example, the capability to process 64 *qubits* simultaneously instead of 64 classical *bits* increases the speed of the computation by a factor of 2^{64} ($= 2 \times 10^{19}$)!

However, a major problem with the superposition state is that it collapses into a random state once it is measured. For example, assume an electron as a qubit with spin-up orientation representing state $|0>$ and spin-down state $|1>$. If the particle enters a superposition of states, it behaves as if it were in both $|0>$ and $|1>$ states simultaneously.

Thus an operation on a single qubit affects both values of the qubit at the same time. Similarly, any operation on a system with two qubits will allow simultaneous operation on four values, and on eight values in a three-qubit system. For example, in a four-qubit system at any particular time the 4 qubits can be in any one of 16 possible configurations:

(0000, 0001, 0010, ... 1111)

Thus, a 4-qubit register can be represented in a superposition of the above 16 states:

$$|\Psi> = c_0|0000> + c_1|0001> + c_2|0010> \cdots \cdots \cdots + c_{14}|1110> + c_{15}|1111>$$

where the numbers c_0, c_1, c_2, ..., c_{15} are complex coefficients such that

$$|c_0|^2 + |c_1|^2 + |c_2|^2 + \cdots \cdots \cdots \cdots \cdots + |c_{15}|^2 = 1$$

The states of the qubit register can be represented as tensor products resulting in

$$\begin{aligned}|\Psi> &= C_{0000}|0>\otimes|0>\otimes|0>\otimes|0> + C_{0001}|0>\otimes|0>\otimes|0>\otimes|1> \\ &\quad + C_{0010}|0>\otimes|0>\otimes|1>\otimes|0>) + \cdots \cdots \cdots \cdots \cdots + C_{1110}|1>\otimes| \\ &\quad 1>\otimes|1>\otimes|0>) + C_{1111}|1>\otimes|\ 1>\otimes|1>\otimes|1>) \\ &= C_{0000}|0000> + C_{0001}|0001> + C_{0010}|0010> + \cdots \cdots \cdots \cdots \cdots \\ &\quad + C_{1110}|\ 1110> + C_{1111}|1111>\end{aligned}$$

In a similar way, the state of an n-qubit register can be written as a normalized vector in 2^n-dimensional complex Hilbert space; this allows the superposition of 2^n different base states as indicated earlier.

A unique feature of quantum computing is that with a linear increase in the number of qubits in a register, the dimension of quantum register state space grows exponentially. The additional qubits allow parallel processing capability to quantum computers and thus help in solving certain problems in a time that is many times faster than that is possible in classical computers.

6.4 Entanglement

Entanglement is a unique kind of correlation that exists only in quantum systems. It has no analogue in classical physics and refers to the strange behavior of quantum particles such as electrons, photons that have interacted in specific ways in the past and then moved apart. It is the essential ingredient in such phenomena as *superdense coding*, which allows any of four possible messages to be transmitted via a single quantum particle, and *teleportation*, in which a quantum state is transmitted from one location to another without going through the intervening space.

EPR (Einstein, Podolsky, Rosen) showed that two particles in a quantum system may be correlated such that any measurement on one particle identifies the outcome of the same measurement on its partner particle *instantaneously*, irrespective of how wide is the physical distance between them. In other words, if two particles are entangled, any measurement on one of them can impact the behavior of its partner particle instantaneously, no matter how far away this second

particle moved. This phenomenon is so strange that Einstein called it "spooky action at a distance."

For example, if the particles are correlated in spins and one of them is in the spin-up orientation then the other one must be in spin-down orientation. Thus, the state of the two particle system can be written as [4]:

$$|\text{spin-up}\rangle_1 \otimes |\text{spin-down}\rangle_2$$

It is also possible to have another state of the two particle system in which the first particle has spin-down orientation and the second one is in spin-up orientation:

$$|\text{spin-down}\rangle_1 \otimes |\text{spin-up}\rangle_2$$

Alternatively, the first particle could be in a superposition state of spin-up and spin-down orientation. If the second particle is in a spin-up orientation, the combined state of the particles is

$$(\alpha\,|\text{spin-up}\rangle_1 + \beta\,|\text{spin-down}\rangle_1) \otimes |\text{spin-up}\rangle_2$$

If the spin of the first particle is measured, the probability of getting a spin-up orientation is α^2 and the probability of getting a spin-down orientation is β^2. In either case the second particle is not affected since it has not been measured. Hence, if the outcome of the measurement of the first particle is spin-up, then the state of the system after the measurement will be

$$|\text{spin-up}\rangle_1 \otimes |\text{spin-up}\rangle_2$$

Thus in the quantum states of the kind seen above, a measurement of one particle has no impact on the state of the other.

Based on the fundamental principle of superposition one could create a new quantum state from the two-particle states

$$|\text{spin-up}\rangle_1 \otimes |\text{spin-down}\rangle_2$$

and

$$|\text{spin-down}\rangle_1 \otimes |\text{spin-up}\rangle_2$$

as follows

$$\frac{1}{\sqrt{2}}\left(|\text{spin-up}\rangle_1 \otimes |\text{spin-down}\rangle_2 + |\text{spin-down}\rangle_1 \otimes |\text{spin-up}\rangle_2\right)$$

where $\frac{1}{\sqrt{2}}$ is the normalization constant.

Notice that in the superposition state above, the second particle is in spin-up as well as in spin-down orientation. Both orientations of the

second particle, however, is associated with a particular orientation of the first particle. For example, in the first term spin-up of the first particle is attached to the spin-down orientation of the second, whereas in the second term the spin-down of the first is attached to the spin-up of the second. So if the measurement of second particle results in a spin-up orientation then the final state of the two-particle system will be

$$|\text{spin-down}>_1 \otimes |\text{spin-up}>_2$$

that is, the second term of the expression. Alternatively, if the orientation of the second particle were spin-down then the final state will be the first term of the expression, that is

$$|\text{spin-up}>_1 \otimes |\text{spin-down}>_2$$

Next a measurement is made on the first particle. In the first term the measurement of the second particle had indicated spin-down orientation; therefore, the first particle can only be spin-up since this is the only orientation of the particle present in the combined state. In the second case, the second particle have a spin-up orientation thus the first particle will be in spin-down orientation. Thus, the measurement result of the first particle is determined by the outcome of the earlier measurement on the second particle.

It should now be clear that the measurement of spin on one particle of the pair will correctly predict the subsequent measurement result of the other particle in the pair. The reason for this is that the original state of the two-particle system

$$\frac{1}{\sqrt{2}}(|\text{spin-up}>_1 \otimes |\text{spin-down}>_2 + |\text{spin-down}>_1 \otimes |\text{spin-up}>_2)$$

cannot be factored into a simple tensor product of one state involving only the first particle and another state involving only the second particle; such states are called *entangled*.

It seems from the above discussion that two entangled particles have some kind of invisible link between them. The existence of this linkage can be avoided only if it is possible to write the entangled qubits as a sum of two independent qubits. To illustrate, suppose an entangled state of the two qubits could be created from two two-qubit registers, $p = |00>$ and $q = |11>$, by combining the two states where each has equal weight (ω):

$$\Psi_{00} = \omega|\ 00> + \omega|11> \ldots\ldots\ldots\ldots\ (6.2)$$

Assume that Ψ could be created by taking the tensor product of the individual states of qubits u and v, that is

$$\Psi_{00} = (u_0|0> + u_1|1>) \otimes (v_0|0> + v_1|1>)$$

If the derived values of u_0, u_1, v_0, and v_1 are found to satisfy the expression Ψ, then it can be written as a separable state. By expanding out the tensor product in Ψ,

$$\Psi_{00} = (u_0 v_0 |00> + u_0 v_1 |01> + u_1 v_0 |10> + u_1 v_1 |11>)$$

Since the expression (6.2) does not contain state $|10>$ nor $|01>$, both coefficients $u_1 v_0$ and $u_0 v_1$ will be 0. This means for the first coefficient either $u_1 = 0$ or $v_0 = 0$. However, u_1 is not allowed to be 0 since that will result in $u_1 v_1 = 0$, eliminating state $|11>$. Similarly, if $v_0 = 0$ then $u_0 v_0 = 0$ resulting in the elimination of state $|00>$. In short, neither u_1 nor v_0 can be 0. Hence, state $|10>$ will have a nonzero weight that is in contradiction with expression (6.2); hence, quantum state Ψ needs to be modified as

$$\Psi_{00} = u_0 v_0 |00> + u_1 v_1 |11> + u_1 v_0 |10>$$

Alternatively, if coefficient $u_0 v_1 = 0$ instead of $u_1 v_0$, then state $|01>$ will have a nonzero weight. Thus, the modified quantum state Ψ will be

$$|\Psi_{00}> = u_0 v_0 |00> + u_0 v_1 |01> + u_1 v_1 |11>$$

A simple circuit for generating entanglement is shown in Fig. 6.1.

The first qubit is passed through a Hadamard gate and then both qubits are entangled by a CNOT gate. If the input to the circuit is $|0> \otimes 0>|$, then the Hadamard gate changes the state to

$$\frac{|0> + |1>}{\sqrt{2}} \otimes |0>$$
$$= \frac{1}{\sqrt{2}} (|0> + |1>) \otimes |0>$$
$$= \frac{1}{\sqrt{2}} (|00> + |10>)$$

and after passing through the CNOT gate the state changes to

$$= \frac{1}{\sqrt{2}} (|00> + |11>)$$

which is one of the four Bell states that are maximally entangled; this state is known as the Bell state $|\Phi^+>$.

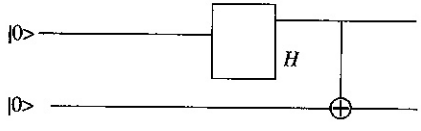


Figure 6.1 Entangled state generation with a Hadamard gate and a CNOT gate (From Ref. 3).

Next assume each output of the circuit is passed through two H gates, that is,

$$H \otimes H\left(\frac{|00> + |11>}{\sqrt{2}}\right)$$

$$= \frac{1}{\sqrt{2}}(H|0> \otimes H|0> + \frac{1}{\sqrt{2}}(H|1> \otimes H|1>)$$

$$= \frac{1}{\sqrt{2}}\left(\frac{1}{\sqrt{2}}(|0> + \frac{1}{\sqrt{2}}(|1>) \otimes \frac{1}{\sqrt{2}}\left(\frac{1}{\sqrt{2}}(|0> + \frac{1}{\sqrt{2}}(|1>)\right.\right.$$

$$\left.\left. + \frac{1}{\sqrt{2}}\left(\frac{1}{\sqrt{2}}(|0> - \frac{1}{\sqrt{2}}(|1>) \otimes \frac{1}{\sqrt{2}}\left(\frac{1}{\sqrt{2}}(|0> - \frac{1}{\sqrt{2}}(|1>)\right.\right.\right.$$

$$= \frac{|00> + |11>}{\sqrt{2}}$$

that is, the Bell state remains the same as the original.

It can be shown that each input combination to the circuit of Fig. 6.1, results in a Bell state; they are all listed below and are also known as the *EPR states*

$$|\psi_{00}> = \frac{|00> + |11>}{\sqrt{2}} = |\Phi^+>$$

$$|\psi_{01}> = \frac{|01> + |10>}{\sqrt{2}} = |\psi^+>$$

$$|\psi_{10}> = \frac{|00> - |11>}{\sqrt{2}} = |\Phi^->$$

$$|\psi_{11}> = \frac{|01> - |10>}{\sqrt{2}} = |\psi^->$$

As another example of entanglement consider a two-qubit system with standard basis chosen for both qubits. Then the basis for the system is

$$(|0>, |1>) \otimes (|0>, |1>)$$

$$= (|00>, |01>, |10>, |11>)$$

On the other hand, if the Hadamard basis is chosen for the first qubit and the standard basis for the second qubit, the basis for the two-qubit system is

$$(|+>, |->) \otimes (|0>, |1>)$$

$$= (|+0>, |+1>, |-0>, |-1>)$$

where

$$(|+0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle \otimes |0\rangle$$

$$= \frac{1}{\sqrt{2}}(|00\rangle + |10\rangle$$

$$(|+1\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle \otimes |1\rangle$$

$$= \frac{1}{\sqrt{2}}(|01\rangle + |11\rangle$$

$$(|-0\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle \otimes |0\rangle$$

$$= \frac{1}{\sqrt{2}}(|00\rangle - \frac{1}{\sqrt{2}}(|10\rangle$$

$$(|-1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle \otimes |1\rangle$$

$$= \frac{1}{\sqrt{2}}(|01\rangle - \frac{1}{\sqrt{2}}(|11\rangle$$

Therefore, the Hadamard-Standard base is

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

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

6.5 Decoherence

The superposition principle states that *any* two states $|A\rangle$, $|B\rangle$ of a quantum system may be superimposed, yielding a new state. Thus, a quantum bit can yield a new state from the arbitrary superposition of its two states. However, a superposed state is very fragile and therefore difficult to control [5]. As a consequence, any kind of interaction with their environment can cause superposed states to eradicate certain *coherences*, therefore preventing the associated states from interfering with each other. This effectively destroys the superposition and the system *collapses* randomly into one of the states that constitute the superposition state; this process is known as *decoherence*.

Decoherence is an undesirable effect in quantum systems. It destroys many possible advantages of quantum systems over classical systems. For example, entanglement which has potential applications in quantum computation, quantum cryptography may be lost due to decoherence. As another example, the superposition of states that allow parallel processing of quantum information, are the ones that are most susceptible to decoherence. Hence qubits need to be designed such that the effects of environmental interactions that make it difficult to maintain the quantum superposition feature for prolonged periods of time are eliminated. This is a necessary requirement in quantum computing systems. Currently decoherence is a major barrier to the development of quantum information processing systems. It is generally accepted now that reliable computation in the presence of decoherence is possible only by incorporating some form of quantum error correction.

References

1. 221A Lecture Notes on Tensor Product, University of California Berkeley, Hitoshi Murayama, Fall 2006.
2. Mark Wilde, Quantum Information Processing Basics (Lecture 1), Louisiana State, University, Department of Physics and Astronomy, Baton Rouge, June 2012.
3. Ryan O'Donnell, Quantum Computation (CMU 18-859BB, Fall 2015) Lecture 3: The Power of Entanglement, Carnegie Mellon University, Pittsburgh.
4. Oliver Morsch, *Quantum Bits and Quantum Secrets*, Wiley-VCH, 2008.
5. Maximilian A. Schlosshauer, *Decoherence and the Quantum-to-Classical Transition*, Springer, 2007.

CHAPTER 7

Teleportation and Superdense Coding

Quantum teleportation is used to replace the state of one qubit with that of another over a long distance without the qubits directly interacting with each other. It works only at the level of individual quantum particles such as photons, electrons etc., and has not even vague resemblance with what is presented in television shows and/or science fictions stories. Superdense coding can be viewed as the process in which two classical bits of information are transmitted by sending just one quantum bit.

7.1 Quantum Teleportation

The goal of teleportation is to transfer the unknown state information of the source (first) qubit without measuring or observing, to the destination (second) qubit, thereby avoiding the disturbance of the first [1, 2, 4]. The second qubit, therefore, does not receive a *copy* of the quantum state of the first since it is impossible to produce an exact copy of an arbitrary quantum state (*no-cloning theorem*). As it turns out, the second qubit does not need a copy of the state information of the first—the *original* state of the first is *teleported* to it. Note that the process is not faster than light and a pair of entangled states has to be distributed ahead of time:

1. At the start, assume a single-qubit state

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

   at a location A, and that α and β in the state are unknown. Therefore, the necessary information to specify the state at location A are not available.
2. Generate an entangled state of a pair of qubits; assume the entangled state is a *Bell* (EPR) state and is written as

$$|\vartheta> = \frac{1}{\sqrt{2}}(|00> + |11>)$$

The first half of the Bell state is sent to location A and the second half to location B. Thus, there are two qubits in location A (state $|\Psi>$ and half of the Bell state $|\vartheta>$), and one in location B (the second half of the Bell state).

3. To teleport the qubit at location A to location B, create a tensor product of qubit at A with ϑ

$$\omega_1 = \psi \otimes \vartheta$$

$$= (\alpha|0> + \beta|1>) \otimes \frac{1}{\sqrt{2}}(|00> + |11>)$$

$$= (\alpha|0> \otimes \frac{1}{\sqrt{2}}|00> + |11>) + \beta|1> \otimes \frac{1}{\sqrt{2}}(|00> + |11>)$$

$$= \frac{1}{\sqrt{2}}(\alpha|000> + \alpha|01\,1> + \beta|10\,0> + \beta|111>)$$

Note that there are three qubits at the start:

a. Qubit 1 is in an unknown state that is to be teleported and is located in A.
b. Qubit 2 is the first half of the entangled pair and is located in A.
c. Qubit 3 is the second half of the entangled pair and is located in B.

4. Next the two qubits in location A are sent through a CNOT gate. As indicated in Chap. 5 a CNOT gate inverts the state of the second qubit if the first qubit is in state 1, otherwise nothing changes. Thus, the second qubit of terms 3 and 4 in state ω_0 change giving a new state:

$$\omega_1 = \frac{1}{\sqrt{2}}(\alpha|00\,0> + \alpha|01\,1> + \beta|11\,0> + \beta\,|10\,1>)$$

5. Next Qubit 1, that is the first qubit that initially contains the state to be teleported, is sent through a Hadamard gate. There are four terms in state ω_1 with the first qubit being in state 0, 0, 1, and 1, respectively. As indicated previously a Hadamard gate transforms state $|0>$ and $|1>$ into

$$|0> = \frac{1}{\sqrt{2}}(|0> + |1>)$$

$$|1> = \frac{1}{\sqrt{2}}(|0> - |1>)$$

respectively.

By substituting $|0>$ and $|1>$ in the first qubit of the terms of ω_1 with their Hadamard transformations, another quantum state ω_2 is obtained:

$$\omega_2 = \frac{1}{\sqrt{2}}(\alpha|00\,0> + \alpha|01\,1> + \beta|110> + \beta|10\,1>)$$

$$= \frac{1}{\sqrt{2}}[\alpha(\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\ |0\,0\rangle + \alpha(\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\ |1\,1\rangle$$

$$+ \beta(\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)\ |1\,0\rangle + \beta(\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)\ |0\,1\rangle]$$

This indicates a superposition of eight states that can be rearranged as follows:

$$\omega_2 = \frac{1}{\sqrt{2}}[\alpha((\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\ |0\,0\rangle + (\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\ |1\,1\rangle)$$

$$+ \beta((\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)\ |1\,0\rangle + (\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)\ |0\,1\rangle)]$$

$$= \frac{1}{2}[\alpha|000\rangle + \alpha|100\rangle + \alpha|011\rangle + \alpha|111\rangle + \beta|010\rangle$$

$$- \beta|11\,0\rangle + \beta|001\rangle - \beta|101\rangle]$$

$$= \frac{1}{2}[|00\rangle(\alpha|0\rangle + \beta|1\rangle) + |01\rangle(\alpha|1\rangle + \beta|0\rangle)$$

$$+ |10\rangle(\alpha|0\rangle - \beta|1\rangle) + |11\rangle(\alpha|1\rangle - \beta|0\rangle)]$$

Note that at this stage that qubits Q1 and Q2 are in location A, and the third qubit Q3 is in location B.

Using the two-dimensional unit matrix I and the three Pauli matrices:

$$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \qquad X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

$$Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \qquad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

the state ω_2 can be rewritten as

$$= \frac{1}{2}[|00\rangle \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} |\psi\rangle + |01\rangle \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} |\psi\rangle$$

$$+ |10\rangle \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} |\psi\rangle + |11\rangle i \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} |\psi\rangle]$$

$$= \frac{1}{2}[|00\rangle \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} |\psi\rangle + |01\rangle \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} |\psi\rangle$$

$$+ |10\rangle \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} |\psi\rangle + |11\rangle \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} |\psi\rangle]$$

$$= \frac{1}{2}[|00\rangle I|\psi\rangle + |01\rangle X|\psi\rangle + |10\rangle Z|\psi\rangle + |11\rangle XZ|\psi\rangle]$$

Result of the Measurement of Q1 and Q2 in Location A	State of Q3 in Location B
$c_0\, c_1$	
00	$(\alpha\|0> + \beta\|1>)$
01	$(\alpha\|1> + \beta\|0>)$
10	$(\alpha\|0> - \beta\|1>)$
11	$(\alpha\|1> - \beta\|0>)$

Table 7.1 One of Four Possible States of Q3 After Measurement

Note that qubits 1 and 2 are different in each term. If the qubits in location A are measured, the outcomes can be encoded using one of the following pairs of classical bits:

$$c_0c_1 = 00, 01, 10, \text{ or } 11$$

In other words, the four possible outcomes upon measuring qubits 1 and 2 result in two bits of classical information: c_0 and c_1. This measurement has an impact on Qubit 3 in location B and leaves it in one of the four distinct states as shown in Table 7.1; the classical bits c_0 and c_1 identify the state.

6. The next step is to send the classical bits c_0 and c_1 to location B via a classical channel. Depending on the values of c_0 and c_1, one of the four possible unitary operations is performed on Qubit 3 (qubit in location B) as shown in Table 7.2. This step restores state ψ, that is the original state of Qubit 1.

   As an example, suppose the qubits in location A (Qubits 1 and 2) are measured and the result is 00, then as shown previously the qubit in location B (Qubit 3) is in state $(\alpha\,|\,0> + \beta\,|\,1>)$. Note that this is the state that was initially intended for teleportation, thus the teleportation from A to B already

State of Q1 and Q2 in Location A	Unitary Operation Needed to Restore Original State of Location A
00	I
01	X
10	Z
11	ZX

Table 7.2 Unitary Operation Performed on Qubit 3

happened in this instance. However, this is not the only result of the measurement, there are four possible results, each of which gives a different state for the qubit in location B. If the result of the measurement in location A is 00, 01, 10, or 11 then the state of the qubit at location B becomes $(\alpha|0> + \beta|1>)$, $(\alpha|1> + \beta|0>)$, $(\alpha|0> - \beta|1>)$, and $(\alpha|1> - \beta|0>)$, respectively.

These four possible outcomes upon measuring two qubits in location A are encoded using the two classical bits c_0 and c_1. In each case a unitary transformation is applied on the state of Qubit 3 in location B so that the state of Qubit 1 and Qubit 2 in A are restored to their original value as discussed below:

i. State of the qubits 1 and 2 = 00

$$\begin{array}{ll} q_1q_2 & q_3 \\ |0\,0> & (\alpha|0> + \beta|1>) \end{array}$$

Classical bits $c_0c_1 = 00$ are sent from location A to B. Since both bits are 0s, both operators X and Z are idle, and unitary operator I is applied to Qubit 3 as indicated in Table 7.2, so it retains its state $\alpha|0> + \beta|1>$:

$$\begin{aligned} I(\alpha|0> + \beta|1>) &= \alpha I|0> + \beta I|1> \\ &= \alpha\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} 1 \\ 0 \end{pmatrix} + \beta\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} 0 \\ 1 \end{pmatrix} \\ &= \alpha\begin{pmatrix} 1 \\ 0 \end{pmatrix} + \beta\begin{pmatrix} 0 \\ 1 \end{pmatrix} \\ &= \alpha|0> + \beta|1> = \psi \end{aligned}$$

ii. State of the qubits 1 and 2 = 01

$$\begin{array}{ll} q_1q_2 & q_3 \\ |0\,1> & (\alpha|1> + \beta|0>) \end{array}$$

Classical bits $c_0c_1 = 01$ are sent from location A to B. Since $c_0 = 0$ and $c_1 = 1$, operator X is applied to the qubit in location B, that is, Qubit 3 as indicated in Table 7.2.

$$X(\alpha|1> + \beta|0>) = \alpha X|1> + \beta X|0>$$

Thus the state of the qubit changes to $\alpha|0> + \beta|1> = \psi$

iii. State of the qubits 1 and 2 = 10

$$\begin{array}{ll} q_1q_2 & q_3 \\ |1\,0> & (\alpha|0> - \beta|1>) \end{array}$$

Classical bits $c_0c_1 = 10$ are sent from location A to B as shown in Table 7.1. Since $c_0 = 1$ and $c_1 = 0$, operator Z is applied to the qubit in location B. Thus, the state of the qubit changes to $\alpha|0\rangle + \beta|1\rangle) = \psi$:

$$Z(\alpha|0\rangle - \beta|1\rangle) = \alpha Z|0\rangle - \beta Z|1\rangle$$

$$= \alpha\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}\begin{pmatrix} 1 \\ 0 \end{pmatrix} - \beta\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}\begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

$$= \alpha\begin{pmatrix} 1 \\ 0 \end{pmatrix} + \beta\begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

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

iv. State of the qubits 1 and 2 = 11

q_1q_2	q_3
$\|1\,1\rangle$	$(\alpha\|1\rangle - \beta\|0\rangle)$

Classical bits $c_0c_1 = 11$ are sent from location A to B. Since $c_0 = 1$ and $c_1 = 1$, both X and Z are applied to the qubit in location B:

$$ZX(\alpha|1\rangle - \beta|0\rangle) = \alpha ZX|1\rangle - \beta ZX|0\rangle$$

$$= \alpha Z|0\rangle - \beta Z|1\rangle$$

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

Thus, the state of the qubit changes to $\alpha|0\rangle + \beta|1\rangle$.

In each of the above cases there is a unitary transformation that restores the state of the qubit in location B to original state ψ.

7.2 No-Cloning Theorem

In classical computing systems, it is taken for granted that digital data can be copied with perfect accuracy. The *no-cloning theorem* describes one of the most fundamental properties of quantum systems, namely, there is no unitary operation that will perfectly copy an arbitrary quantum state [3]. An arbitrary state in this context means any state of a specified Hilbert space that is being considered. This obviously limits the available resources for programming a quantum computer. However, the no-cloning feature is extremely important in quantum cryptography because the inability of copying an unknown quantum state is a contributing factor to the system security.

To illustrate the operation of cloning assume a hypothetical machine that accepts the state of a qubit as an input and produces two exact copies of the state, that is *clones* of the state. For example, a state $|\varphi\rangle$ is transformed into $|\varphi\varphi\rangle$ by the machine. Similarly, another

state $|\phi\rangle$ is converted to $|\phi\phi\rangle$. However, if a state that is a linear combination of two states is sent through the cloning machine, the output obtained is

$$|\omega\rangle = (a|\varphi\varphi\rangle + b|\phi\phi\rangle)$$

that is, a superposition of the two copies of $|\varphi\rangle$ and two copies of $|\phi\rangle$ because in quantum systems the linearity property is preserved. However, the output of the machine is expected to be

$$|\psi\rangle|\psi\rangle = (a|\varphi\rangle + b|\phi\rangle)(a|\varphi\rangle + b|\phi\rangle)$$

that is the original and a copy of $|\psi\rangle$, and not $|\omega\rangle$ produced by the machine! The no-cloning theorem formally states this result:

Theorem: There is *no* valid quantum operation that maps an arbitrary state $|\psi\rangle$ to $|\psi\rangle|\psi\rangle$ [5,6].

Assume an initial state $|s\rangle$ that is to be converted into any other state $|\phi\rangle$ or $|\varphi\rangle$. For example, if $|s\rangle$ is to be converted to $|\varphi\rangle$ then the initial pair of state $|s\rangle$ and $|\varphi\rangle$ is transformed into two copies of $|\varphi\rangle$ by a unitary transformation U as shown in Fig. 7.1.

The copying of a state using an unitary operator U can be written as

$$U|\varphi\rangle \otimes |s\rangle = |\varphi\rangle \otimes |\varphi\rangle \tag{7.1}$$

Similarly for another state $|\phi\rangle$,

$$U|\phi\rangle \otimes |s\rangle = |\phi\rangle \otimes |\phi\rangle \tag{7.2}$$

Take the inner product of the left-hand sides of the above equations

$$U|\varphi\rangle \otimes |s\rangle\, U|\phi\rangle \otimes |s\rangle$$

Replace the first half of the equation with its complex conjugate,

$$= \langle s| \otimes \langle\varphi|\ U^* U\ |\phi\rangle \otimes |s\rangle$$
$$= \langle s| \otimes \langle\varphi|\ U^* U\ |\phi\rangle \otimes |s\rangle$$

Since $U^* U = I$,

$$= \langle s| \otimes \langle\varphi|\phi\rangle \otimes |s\rangle$$
$$= \langle\varphi|\phi\rangle \langle s|s\rangle$$

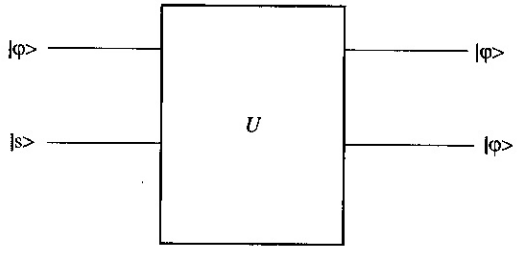


Figure 7.1 Unitary transformation U for copying state.

Since $\langle s \mid s \rangle = 1$

$$= \langle \varphi \mid \phi \rangle \tag{7.3}$$

Similarly, taking the inner product of the right-hand sides of Eqs. (7.1) and (7.2) gives

$$\langle \varphi \mid \otimes \langle \varphi \mid \mid \phi \rangle \otimes \mid \phi \rangle$$
$$= (\langle \varphi \mid \phi \rangle)^2 \tag{7.4}$$

However, Eqs. (7.3) and (7.4) must be equal; this implies

$$\langle \varphi \mid \phi \rangle = (\langle \varphi \mid \phi \rangle)^2$$

Thus, $\langle \varphi \mid \phi \rangle$ is either 0 or 1, that is, $\mid \phi \rangle$ and $\mid \varphi \rangle$ are either orthogonal or the same state. Perfect copying is only possible for a set of states that are orthogonal, not for any arbitrary state.

An alternative proof of the no-cloning theorem is as follows. Assume a unitary operator that can copy an unknown state $\mid \alpha \rangle = a \mid 0 \rangle + b \mid 1 \rangle$ onto a state $\mid s \rangle$, then

$$U(\mid \alpha \rangle \mid s \rangle) = \mid \alpha \rangle \mid \alpha \rangle = (a \mid 0 \rangle + b \mid 1 \rangle)(a \mid 0 \rangle + b \mid 1 \rangle)$$
$$= a^2 \mid 00 \rangle + ab \mid 01 \rangle + ab \mid 10 \rangle + b^2 \mid 11 \rangle$$

However, if the linear combination corresponding to state $\mid \alpha \rangle$ is sent through the cloning machine then a different state is produced:

$$U(a \mid 0 \rangle + b \mid 1 \rangle) \mid s \rangle = (a \mid 00 \rangle + b \mid 11 \rangle)$$
$$\neq (a \mid 0 \rangle + b \mid 1 \rangle)(a \mid 0 \rangle + b \mid 1 \rangle)$$

This contradiction in result shows there does not exist an operator for cloning.

7.3 Superdense Coding

Superdense coding can be viewed as teleportation in reverse. The idea is to transmit two classical bits of information by sending a single qubit through the quantum channel. Suppose Alice wants to send a two-bit message to Bob [1, 4, 6]. She could send two qubits with the message encoded in them. A single qubit on its own cannot transmit two classical bits of information. However, superdense coding allows a single qubit to perform this. This option requires that the two parties initially share a pair of entangled qubits. Assume initially Alice and Bob share a Bell state:

$$\frac{1}{\sqrt{2}}(\mid 00 \rangle + \mid 11 \rangle)$$

The first qubit in each term is Alice's half of the state and the second qubit is used by Bob. Note that the Bell state is a fixed state; Alice and Bob do not need to prepare the state. It is assumed that some third party prepared the Bell state beforehand and sent one of entangled qubits to Alice, and the other one to Bob. Hence, Alice and Bob each possess half of the Bell state, that is they share one unit of the entangled pair.

For example, assume $s_1 s_0$ is the two-bit string Alice wants to send to Bob. There are four possible combinations of $s_1 s_0$ (= 00, 01, 10, or 11) that Alice can send to Bob using the qubit she has. She chooses one of four unitary operations U(= I, X, Y or Z), on the entangled bit in her possession based upon which bit string she wishes to send to Bob. Applying the transformation only to her qubit means she needs to apply an identity (I) operation on the second qubit (in Bob's possession) so that it does not change. This coding process is described below, and the combined state after the application of the chosen unitary operation is also shown with Alice's qubit in bold:

i. *Classical bits to be sent: 00.* Nothing needs to be done, so Alice applies $U = I \otimes I$ on her part of the Bell state

$$I \otimes I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Hence

$$(I \otimes I)\left(\frac{1}{\sqrt{2}}(|00> + |11>)\right) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \left(\frac{1}{\sqrt{2}}(|00> + |11>\right)$$

$$= \left(\frac{1}{\sqrt{2}}(|0>|0> + |1>|1>\right)$$

ii. *Classical bits to be sent: 01.* Alice applies $U = X \otimes I$ on her part of the Bell state

$$U = X \otimes I$$

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

Hence

$$(X \otimes I)\left(\frac{1}{\sqrt{2}}(|00> + |11>)\right) = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \left(\frac{1}{\sqrt{2}}(|00> + |11>\right)$$

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

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

$$= \frac{1}{\sqrt{2}}(|1>|0> + |0>|1>)$$

iii. *Classical bits to be sent: 10.* Alice applies $U = Z \otimes I$ on her part of the Bell state

$$U = Z \otimes I$$

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

Hence

$$(Z \otimes I)(\frac{1}{\sqrt{2}}(|00> + |11>)$$

$$= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} (\frac{1}{\sqrt{2}}(|00> + |11>)$$

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

$$= \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 0 \\ 0 \\ -1 \end{pmatrix}$$

$$= \frac{1}{\sqrt{2}}(|0>|0> - |1>1>)$$

iv. *Classical bits to be sent: 11.* Alice applies $U = XZ \otimes I$ on her part of the Bell state

Hence

$$U = iY \otimes I \text{ since } XZ = iY$$

$$= i \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

Thus

$$(XZ \otimes I)\left(\frac{1}{\sqrt{2}}(|00>+|11>)\right)$$

$$=\begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{pmatrix}\left(\frac{1}{\sqrt{2}}(|00>+|11>)\right.$$

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

$$=\frac{1}{\sqrt{2}}\begin{pmatrix} 0 \\ 1 \\ -1 \\ 0 \end{pmatrix}=\frac{1}{\sqrt{2}}(|0>|1>-|1>|0>)$$

Table 7.3 shows the starting state, the unitary operation applied, and the resulting final state for sending a two-bit string. Note that like the initial states, the final states are also Bell basis states.

After the application of the unitary operation Alice sends her half of the entangled qubits, that is, q_0 to Bob. Bob combines it with his qubit (q_1) and applies a controlled-NOT operation to the pair (q_0 q_1), where q_0 is assumed to be the control bit. Next a Hadamard transform on the first qubit of the pair leads to the untanglement of the Bell state and results in a unique state that corresponds to the two-bit string. This process can be explained as follows:

Classical Bits to Be Sent	Initial State	Unitary Operation	Final State
00	$\frac{1}{\sqrt{2}}(\|00>+\|11>)$	$I \times I$	$\frac{1}{\sqrt{2}}(\|0>\|0>+\|1>\|1>)$
01	$\frac{1}{\sqrt{2}}(\|00>+\|11>)$	$X \times I$	$\frac{1}{\sqrt{2}}(\|1>\|0>+\|0>\|1>)$
10	$\frac{1}{\sqrt{2}}(\|00>+\|11>)$	$Y \times I$	$\frac{1}{\sqrt{2}}(\|0>\|0>-\|1>\|1>)$
11	$\frac{1}{\sqrt{2}}(\|00>+\|11>)$	$X \times Z$	$\frac{1}{\sqrt{2}}(\|0>\|1>-\|1>\|0>)$

Table 7.3 Transmission of One of a Four Possible Two-Bit Strings via a Single Qubit

Case 00: The controlled-NOT operation to the pair $(q_0\, q_1)$ has no effect on the $|0\rangle|0\rangle$ part of the Bell state

$$\frac{1}{\sqrt{2}}(|0\rangle|0\rangle+|1\rangle|1\rangle$$

but since bit q_0 is 1, it changes the $|1\rangle|1\rangle$ part into $|1\rangle|0\rangle$; thus the Bell state is transformed into

$$\frac{1}{\sqrt{2}}(|0\rangle|0\rangle+|1\rangle|0\rangle \tag{7.5}$$

Bob then applies H (Hadamard transform) on the first qubit (q_0) of the entangled pair; this changes Bell state Eq. (7.5):

$$\frac{1}{\sqrt{2}}[\frac{1}{\sqrt{2}}(|0\rangle+|1\rangle)|0\rangle+\frac{1}{\sqrt{2}}(|0\rangle-|1\rangle)|0\rangle]$$

$$=\frac{1}{2}[(|0\rangle+|1\rangle)|0\rangle+(|0\rangle-|1\rangle)|0\rangle]$$

$$=\frac{1}{2}[|00\rangle+|10\rangle+(|00\rangle-|10\rangle]$$

$$=|00\rangle$$

Bob measures both qubits and gets Alice's message 00.

Case 01: The controlled–NOT operation changes the $|1\rangle|0\rangle$ part of the Bell state

$$\frac{1}{\sqrt{2}}(|1\rangle|0\rangle+|0\rangle|1\rangle)$$

into $|1\rangle|1\rangle$, because the first that is, the control bit is 1; thus the Bell state is transformed into

$$\frac{1}{\sqrt{2}}(|1\rangle|1\rangle+|0\rangle|1\rangle \tag{7.6}$$

Bob then applies H (Hadamard transform) on the first qubit (q_0) of the entangled pair, this changes Bell state Eq. (7.6):

$$\frac{1}{\sqrt{2}}[\frac{1}{\sqrt{2}}(|0\rangle-|1\rangle)|1\rangle+\frac{1}{\sqrt{2}}(|0\rangle+|1\rangle)|1\rangle]$$

$$=\frac{1}{2}[|01\rangle-|11\rangle+(|01\rangle+|11\rangle]$$

$$=|01\rangle$$

Bob measures both qubits and gets Alice's message 01.

Case 10: The controlled–NOT operation changes the $|1\rangle|1\rangle$ part of the Bell state

$$\frac{1}{\sqrt{2}}(|0\rangle|0\rangle - |1\rangle|1\rangle)$$

and transforms it to

$$\frac{1}{\sqrt{2}}(|0\rangle|0\rangle - |1\rangle|0\rangle) \tag{7.7}$$

The application of Hadamard transform on the q_0 bit, changes the Bell state Eq. (7.7):

$$\begin{aligned}
&\frac{1}{\sqrt{2}}[\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)|0\rangle - \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)|0\rangle] \\
&= \frac{1}{2}[|00\rangle + |10\rangle - (|00\rangle + |10\rangle] \\
&= |10\rangle
\end{aligned}$$

Bob measures both qubits and gets Alice's message 10.

Case 11: The controlled–NOT operation changes the $|1\rangle|0\rangle$ part of the Bell state

$$\frac{1}{\sqrt{2}}(|0\rangle|1\rangle - |1\rangle|0\rangle)$$

to

$$\frac{1}{\sqrt{2}}(|0\rangle|1\rangle - |1\rangle|1\rangle) \tag{7.8}$$

The application of Hadamard transform on the q_0 bit, changes the Bell state (7.8):

$$\begin{aligned}
&\frac{1}{\sqrt{2}}[\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)|1\rangle - \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)|1\rangle] \\
&= \frac{1}{2}[|01\rangle + |11\rangle - (|01\rangle + |11\rangle] \\
&= |11\rangle
\end{aligned}$$

Bob measures both qubits and gets Alice's message 11.

Thus, in each of the four cases above Bob needed two bits to decode the status of the final state. This means Alice's message to Bob was composed of two bits not one, otherwise it was not possible to decode four separate states!

References

1. Mark Oskin, Quantum Computing—Lecture Notes, University of Washington.
2. C/CS/Phys C191 Teleportation Fall 2009 Lecture 5, UC Berkley, 2009.
3. Ryan Odonnell, Quantum Computation Lecture Notes, Lecture 3: The Power of Entanglement, Carnegie Mellon University, 2015.
4. John Watrous, CPSC 519/619: Quantum Computation, Lecture 3: Superdense Coding, Quantum Circuits, and Partial Measurements, University of Calgary, 2006.
5. G. Benenti, G. Casati, and G. Strini, *Principles of Quantum Computation and Information Vol. I Basic Concepts*, World Scientific, 2004.
6. D. McMahon, *Quantum Computing Explained*, Wiley Interscience, 2008.

CHAPTER 8

Quantum Error Correction

A major drawback of quantum computers is that any interaction of qubits with their environment can corrupt the qubits. However, by using error correcting codes, it is possible to detect and correct errors in qubits without disturbing their coherence. Quantum error coding techniques are very similar to their classical counterparts in which additional bits are used to recover from an erroneous state to the original one.

In classical communication or computer systems, information bits transmitted from one point to another are subject to changes because of the environmental interference or a physical defect in the communication medium. As a result a single-bit value in a group of data can change from a 0 to a 1 or vice versa. This is known as a *single-bit error*.

The coding of the data is generally used for detecting and correcting an error(s) to ensure information is transferred intact from its source to its destination. Almost all systems that handle digital data employ some form of encoding of data bits at the source end for transmission. The encoding process uses extra bits before the data is transmitted. The extra bits are redundant to the information; they are discarded as soon as the accuracy of the transmission has been determined. This redundancy enables detection of any error introduced during transmission at the destination end.

Assume, for example, a data string given by

$$0\ 1\ 1\ 1\ 0\ 1\ 0\ 0$$

Suppose after passing through a transmission medium the data becomes

$$0\ 1\ 1\ \underline{0}\ 0\ 1\ 0\ 0$$

One bit is erroneous in the received data, but this is not obvious unless a bit-by-bit comparison of the transmitted and received strings are made.

However, if instead of storing the original string, every bit in the string is duplicated before storing

$$00 \quad 11 \quad 11 \quad 11 \quad 00 \quad 11 \quad 00 \quad 00$$

then it is easy to see that in the absence of any bit errors, the string should consist of only pairs of 00 and 11. Now suppose some bits got flipped as shown below:

$$00 \quad 11 \quad 10 \quad 11 \quad 00 \quad 11 \quad 01 \quad 00$$

It is clear now that two pairs of the string (pairs 3 and 7) are not coded identifying the presence of single-bit errors in two separate pairs of the string. Similarly, if two individual bits in a pair are erroneous, for example,

$$00 \rightarrow 11$$

$$11 \rightarrow 00$$

the presence of a double-bit error is indicated. However, flipping of two bits simultaneously transforms a valid code word into another valid code word, hence the error cannot be detected. This strategy of including additional bits to allow the receiver to decide whether the encoded data is corrupt or not is known as *error-detecting codes*.

Alternatively, error correction in classical computing systems is concerned with coding strategies that guarantee any errors occuring during the transmission of information bits are automatically detected and corrected. Suppose a source (Alice) wishes transmit classical information bits via a communication channel to a receiver (Bob) as shown in Fig. 8.1. In practice the channel is noisy, so the information is prone to errors during transmission. To protect the information from the effects of the noise Alice incorporates some redundant (*check*) bits to the information bits and sends the mixed (*encoded*) data bits to Bob. After receiving the mixed data Bob identifies the erroneous bits, corrects the errors and removes the check bits.

8.1 Classical Error-Correcting Codes

The simplest example of a classical error-correcting code is a *repetition* code. It uses three bits to encode the value of a single bit:

$$0 \rightarrow 000$$

$$1 \rightarrow 111$$

If an error causes one of the three bits to flip, then the error can be eliminated by majority voting.

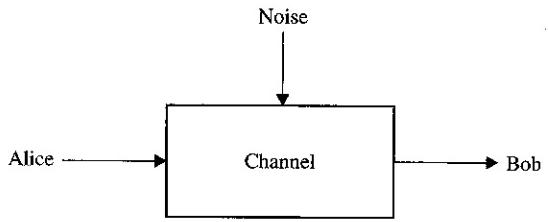


Figure 8.1 Transmission of data in presence of noise.

The general strategy for identifying an error is to send data bits with repetition. This repetition allows for some of the data to be corrupted while still retaining the ability to decode the original message. Figure 8.1 shows a simple model of noisy channels known as the *Binary Symmetric Channel*. Suppose the channel carries one bit at a time from Alice to Bob; it is assumed that the bit is usually transmitted correctly. However, there is noise on the channel and its effect is such that there is a small probability $p > 0$ that the bit sent by Alice is flipped and consequently the bit being sent is not received correctly. This probability known as the *crossover probability* represents the probability of receiving a bit that is different than what was transmitted. Thus, if Alice sends bit b, then the probability of Bob receiving bit b is $(1 - p)$ and not receiving bit b is p.

A simple means of protecting a bit against the effects of noise is to replace the bit with three copies of itself; this is known as the *3-bit repetition code*. That is, each time a logic 0 needs to be sent, 3 bits all in state 0 are sent through the channel. Similarly, each time a logic 1 needs to be sent, 3 bits all in state 1 are sent:

$$0 \rightarrow 000$$

$$1 \rightarrow 111$$

At the receiving end, if among the received 3 bits at least two have the same value v, then v is accepted to be the correct output by the rule of majority decision. This decoding process is known as the *majority decoding*. Thus if there is a single-bit flip error, it can be corrected by choosing the majority of the three bits, for example, $101 \rightarrow 1$. The error-correction scheme, using three copies of a bit and majority decoding, works as long as only one error occurs and is known as *TMR (triple modular redundancy)*. Table 8.1 shows how the 3-bit repeated data is decoded, and the associated probability of error during decoding. The total probability of error is therefore

$$= p^2(1-p) + p^2(1-p) + p^2(1-p) + p^3 = 3p^2(1-p) + p^3$$

It is clear from Table 8.1 that if only one error occurs, the 3-bit repetition code decodes the correct value with 100% accuracy. The overall

Sent	Received	Decoded	Probability of Error
000	000	0	$(1-p)^3$
	001	0	$p(1-p)^2$
	010	0	$p(1-p)^2$
	100	0	$p(1-p)^2$
	011	1	$p^2(1-p)$
	110	1	$p^2(1-p)$
	101	1	$p^2(1-p)$
	111	1	p^3

TABLE 8.1 Probability of Error During Decoding

probability of error is just the likelihood that two or three errors occurring simultaneously, that is $3p^2(1-p)+p^3$, hence the probability of decoding correctly is $(1-p^3)+3p^2(1-p)$. Without encoding the correct bit is received with probability $(1-p)$. Hence the encoding of the data improves reliability if the overall probability of error is $<p$. If $p=\frac{1}{2}$, then the probability of accurately receiving a transmitted bit is only 50%; hence, the channel is obviously of no use.

8.2 Quantum Error-Correcting Codes

Error correction in quantum computing systems, as in its classical counterpart is concerned with coding strategies that guarantee any errors occuring during the transmission of information bits are automatically detected and corrected.

The 3-bit repitition can also be employed in a quantum setting. However, detecting quantum errors is not straightforward at all. For example, unlike in a classical system the *bit flip errors* (a bit changes from 0 to 1 or vice versa) cannot be detected and corrected just by copying the same bit three times and taking the output of the majority of error free outputs as the correct output. This approach is not reliable because qubit states are fragile and any attempt to directly copy a qubit state might alter its state. Moreover, qubits can be affected by a different type of error known as the *phase error*. Because of the superposition phenomenon in quantum physics, each qubit in a quantum system can temporarily exist as a 0, and as a 0 and 1 simultaneously. A phase error can change the superposition *sign* of the phase relationship between the 0 and 1 values [1–3].

As discussed in Chap. 1, a qubit can be in one of two different states $|0>$ or $|1>$; these two constitute a standard base for the qubit. Moreover, a qubit can be in any complex superposition of these two basic states, $a\ |0>+b\ |1>$, where a and b are complex numbers.

Although arbitrary values of coefficients a and b are allowed, usually only normalized combinations are considered. Since a qubit may be in any one of an infinite number of possible superposition states, many more complex errors can occur in quantum systems than in classical systems. In fact, quantum computation will not be possible without incorporating some kind of error correction feature. Consequently, the need for error correction is extremely important in quantum systems.

Quantum error correcting codes use essentially the same principles as classical error correcting codes. Encoding involves the addition of *check* (redundant) bits to the information bits at the transmitting end such that if the encoded information bits are corrupted by the transmission channel, the original data might be recovered at the receiver end. The number of check bits needed to successfully correct the erroneous information bits depends on the coding technique utilized. Let **E** and **D** be the encoding and decoding operators and let **F** be an operator that represents the error(s) that can occur during the manipulation or transmission of a qubit. ϕ then:

$$\phi \xrightarrow{\mathbf{E}} E(\phi) \longrightarrow \mathbf{F}(E(\phi)) \xrightarrow{\mathbf{D}} \phi$$

In order to develop feasible quantum error correcting codes, it is necessary to be familiar with unique problems associated with quantum information processing:

i. No-cloning theorem prohibits copying. It is impossible to create identical copies of an arbitrary, unknown quantum state $|\psi>$ to obtain multiple copies. In short, a repetition code cannot be designed by duplicating a state several times.

ii. A qubit may be in any one of an infinite number of possible superposition states, that is, linear combinations of states $|0>$ and $|1>$. Different errors on a single qubit may form a combination of error, and as a result identification of a particular error becomes unfeasible.

iii. It is not possible to measure a qubit to detect an error. Any measurement that tests whether a state is correct destroys hidden superposition and the qubit collapses to a single state, thereby making recovery impossible.

Suppose that the quantum channel is transmitting qubits such that a bit is changed from 0 to 1 or vice versa with probability p, thus the probability that the bit remains unchanged is $1 - p$. Such an error in a channel is known as a *bit-flip* error and its effect on a qubit is the same as if it has been acted upon by an X operator:

$$\psi = \alpha\,|0> + \beta\,|1> \xrightarrow{X} |\,\alpha\,|1> + \beta\,|0>$$

Another important class of errors on qubits is known as *phase flips*. The effect of a phase flip on a qubit is similar to the application of a Z operator to the qubit:

$$\psi = \alpha|0> + \beta|1> \xrightarrow{Z} \alpha|0> - \beta|1>$$

8.3 Shor's 3-Qubit Bit-Flop Code

As shown earlier, a bit-flip is the same as an *X* gate in the quantum computing context:

$$|0> \rightarrow X\,|0> = |1>$$
$$|1> \rightarrow X\,|1> = |0>$$
$$|\varphi> = |0> + |1> \rightarrow X\,|\varphi> = |1> + |0>$$

The concept of the classical 3-bit repetition code can be imitated to protect a qubit against bit-flip errors. Both encoding and error correction, however, are done by quantum operations. Assume a qubit in some unknown state:

$$|\varphi> = \alpha|0> + \beta|1>$$

The *embedding* of this single qubit state, $\alpha|0> + \beta|1>$, as three entangled qubits provides protection from bit-flip errors:

$$(\alpha|0> + \beta|1>)|0>|0> \rightarrow \alpha|000> + \beta|111>$$

It is known from the non-cloning theorem that the following encoding of the unknown qubit:

$$(\alpha|0> + \beta|1>) \otimes (\alpha|0> + \beta|1>) \otimes (\alpha|0> + \beta|1>)$$

cannot be performed.

The 3-qubit code encodes a single logical qubit into three physical qubits with the property that it can correct a single-bit flip error. Assume the two logical states of the qubit are defined as

$$|0>_L = |000> \text{ and } |1>_L = |111>$$

Thus, an arbitrary single qubit state $\psi = \alpha|0> + \beta|1>$ can be mapped to

$$\begin{aligned}\alpha|0> + \beta|1> &\rightarrow \alpha|0>_L + \beta|1>_L \\ &= \alpha|000> + \beta|111> \\ &= \psi_L\end{aligned}$$

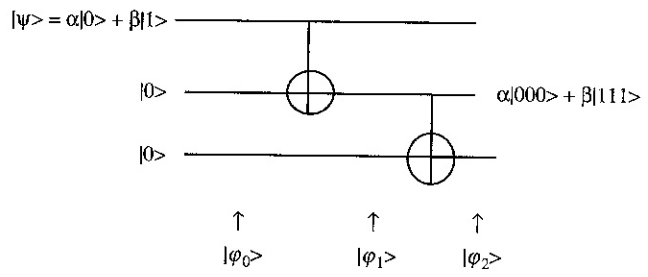


FIGURE 8.2 Encoding of a single qubit.

Figure 8.2 shows the quantum circuit required to encode a single qubit via the initialization of two *ancilla* qubits and two CNOT gates; the ancilla qubits are set to 0. The two CNOT gates act upon the ancilla qubits based on the state of the information qubit.

The inputs to the circuit in Fig. 8.2 are the state $|\psi\rangle$, and two ancillary qubits in the state $|00\rangle$. When the target bit is $|\psi\rangle = |0\rangle$, the ancillary qubits are left unchanged and the output is $|000\rangle$. However, when the input is $|\psi\rangle = |1\rangle$, the output is $|111\rangle$. By the superposition principle, the input $|\psi\rangle|00\rangle = (a|0\rangle + b|1\rangle)|00\rangle$ results in the output

$$|\psi\rangle_L = a|000\rangle + b|111\rangle$$

as expected. This is explained further as follows:

During the $|\varphi_0\rangle$ phase the inputs to the top CNOT gate are as below in Fig. 8.2:

$$|\varphi_0\rangle = \alpha|000\rangle + \beta|100\rangle$$

control input = 0, target input = 0

control input = 1, target input = 0

When the control bit is $|\psi\rangle = |0\rangle$, the ancillary qubits are left unchanged and the output is $|000\rangle$. However, when the input is $|\psi\rangle = |1\rangle$, the output is $|110\rangle$. Hence

$$|\varphi_1\rangle = \alpha|000\rangle + \beta|110\rangle$$

since the top CNOT gate flips from $|10\rangle$ to $|11\rangle$.
Similarly, during $|\varphi_1\rangle$ phase the inputs to the lower CNOT gate are

$$|\varphi_1\rangle = \alpha|000\rangle + \beta|110\rangle$$

control input = 0, target input = 0

control input = 1, target input = 0

Hence

$$|\varphi_2\rangle = \alpha|000\rangle + \beta|111\rangle$$

since the lower CNOT gate flips from $|10\rangle$ to $|11\rangle$.

Thus, the circuit shown in Fig. 8.2 encodes a given qubit $\alpha|0\rangle + \beta|1\rangle$ with two other *ancillary* qubits into a composite state of three qubits, namely, $\alpha|000\rangle + \beta|111\rangle$. Notice that the qubit state has been repeated only in the computational basis. More importantly, the superposition states have redundant encodings but are not repeated, hence the no-cloning theorem is not violated.

8.4 Error Correction

The principle of quantum error correction is the same as its classical counterpart and can be performed by following the steps below:

i. Measure all three encoded qubits.

ii. Identify the one which differs from the others.

iii. Flip back the erroneous qubit.

If the three encoded qubits are transmitted one at a time through a noisy channel, the error protection mechanism will allow detection and correction of a bit-flip error without destroying the superposition. It is assumed that only a single qubit out of the three has flipped.

To illustrate the error correction process, assume during the transmission the first qubit is affected by the noise in the channel and flips but the second and the third qubits are unaffected. Hence the state of the three qubits becomes

$$\alpha|100\rangle + \beta|011\rangle \ldots\ldots\ldots\ldots\ldots\ldots (8.1)$$

At the receiving end of the transmitted code word, however, the presence of an error has to be detected and corrected without having any knowledge of any particular bit flip! In the absence of any additional information, it seems the only option available for identifying an erroneous qubit is to measure the individual outputs of the three encoded qubits, compute the majority, and compare the result with the individual outputs of the qubits. Like in a majority voter-based classical system, a single-qubit error is assumed. This measurement, however, will ruin the information in the qubit sent. Hence, the primary objective of quantum error detection (and correction) is to prevent any changes in the encoded state while retrieving information for the detection of an error in the received code word.

In a quantum system composed of three qubits, it is necessary to determine whether two qubits are different without measuring their actual values. In practice, this is achieved by using *syndrome measurement*

that can identify an erroneous qubit from the extracted *error syndrome*. The error syndrome indicates whether there is an error and identifies its location. The error syndrome ($s_1\, s_0$) for the three-qubit system is determined by comparing the parities of Qubit 1 and Qubit 2, and comparing parities of Qubit 2 and Qubit 3:

$$s_1 = \text{Qubit 2} \oplus \text{Qubit 3}$$

$$s_0 = \text{Qubit 1} \oplus \text{Qubit 2}$$

Notice that s_1 does not need to know whether the Qubit 2 or Qubit 3 is equal to 1 or 0; only whether they have equal parity or not. Similarly, s_0 needs to know only whether Qubit 1 and Qubit 2 have equal parity or not. Table 8.2 shows the error syndromes for all possible combinations of qubits.

It can be seen in Fig. 8.2 that if Qubit 1 and Qubit 2 are the same then ancilla z_1 will have value 0, and if Qubit 2 and Qubit 3 are the same then ancilla z_0 will also have value 0; otherwise the value will be 1 for z_1 and z_0. Thus, the two ancillas are actually the error syndrome s_1 and s_0, and more importantly, measurements of the syndrome do not disturb the quantum state:

s_1	s_0	Bit-Flip Error
0	0	None
1	0	Qubit 1
1	1	Qubit 2
0	1	Qubit 3

Table 8.2 Error Syndromes

8.4.1 Bit-Flip Error Correction

The correction circuit for a single bit-flip error is illustrated in Fig. 8.3 [2]. Like in the encoder circuit, two ancilla qubits are included in this circuit and the parities of the Qubit 1 and Qubit 2 as

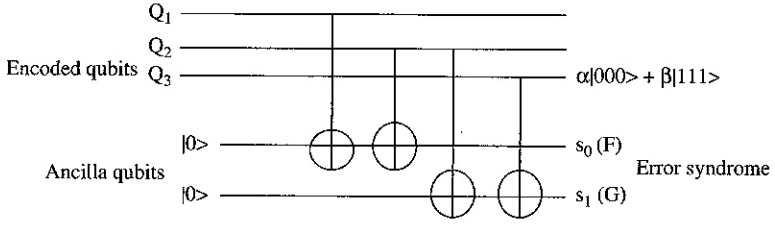


Figure 8.3 Correction circuit.

well as the parity of Qubit 2 and Qubit 3 are checked using four CNOT gates. The control bits of these CNOT gates are the encoded qubits, while the target qubits are the two ancillary qubits introduced at the receiving end. A pair of CNOT gates is used to check the parity of the three-qubit data.

To illustrate how the correction circuit works, assume the encoded data be $Q_1Q_2Q_3$, and further assume that F and G be the output state of the first and the second ancilla qubit, respectively. As can be seen in Fig. 8.3,

$$F = Q_1 \oplus Q_2 \quad \text{and} \quad G = Q_2 \oplus Q_3$$

Therefore, for the data $Q_1Q_2Q_3 = |100\rangle$, the ancilla qubits $|00\rangle$ will be transformed into $|10\rangle$ and the encoded data will be

$$(\alpha|100\rangle + \beta|011\rangle)|00\rangle \rightarrow (\alpha|100\rangle + \beta|011\rangle)|10\rangle$$

The set of two ancillary qubits form the *syndrome* as discussed earlier.

The measurement of the two ancilla qubits leads to the identification of a particular qubit among the three that is corrupted, otherwise all three are error-free as shown in Table 8.3. When an error is detected, an X operator is applied to the erroneous qubit in order to recover from the error. In the absence of any error, only an I (identity) operator is applied since no change in the data is necessary. Thus the direct measurement of any of the qubits can be avoided for the error detection because the measurement result of the ancilla qubits is sufficient to detect and identify a qubit flip error.

Note that in the three-qubit system there can be four pairs of states with each pair having exactly the same ancilla bits as shown in Table 8.4.

The syndrome bit generator circuit can be derived as shown below by noting that syndrome s_1 corresponds to the even parity of the first two encoded qubits while s_0 is the even parity of the last two qubits as discussed earlier. The parity generation circuit is composed of two pairs of CNOT gates (Fig. 8.4), each pair acting as an XOR gate.

State	Ancilla	Error
$(\alpha\|000\rangle + \beta\|111\rangle)$	$\|00\rangle$	No error
$(\alpha\|100\rangle + \beta\|011\rangle)$	$\|10\rangle$	Qubit 1
$(\alpha\|010\rangle + \beta\|101\rangle)$	$\|11\rangle$	Qubit 2
$(\alpha\|001\rangle + \beta\|110\rangle)$	$\|01\rangle$	Qubit 3

Table 8.3 Ancilla Measurement For Single-Bit Flip Errors

State	Ancilla	Error
$(\alpha\|000> + \beta\|111>)$	$\|00>$	No error
$(\alpha\|111> + \beta\|000>)$	$\|00>$	No error
$(\alpha\|100> + \beta\|011>)$	$\|10>$	Qubit 1
$(\alpha\|011> + \beta\|100>)$	$\|10>$	Qubit 1
$(\alpha\|010> + \beta\|101>)$	$\|11>$	Qubit 2
$(\alpha\|101> + \beta\|010>)$	$\|11>$	Qubit 2
$(\alpha\|001> + \beta\|110>)$	$\|01>$	Qubit 3
$(\alpha\|110> + \beta\|001>)$	$\|01>$	Qubit 3

TABLE 8.4 Ancilla Measurement for Single-Bit Flip Errors

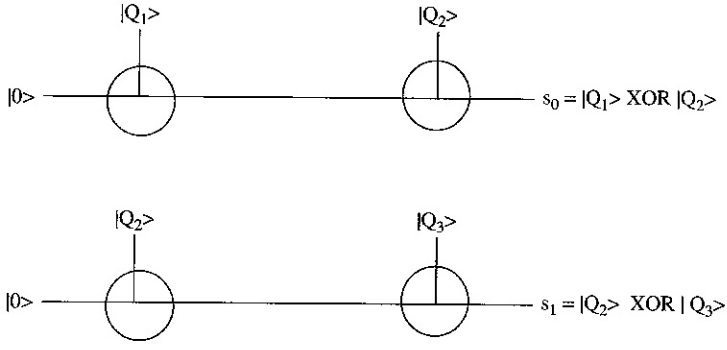


FIGURE 8.4 Parity generation circuit.

To illustrate how a bit-flip error is corrected assume a bit-flip error on the second qubit Q_2 of the entangled qubits $|Q_1Q_2Q_3>$; hence the encoded qubits, $\alpha|000 + \beta|111>$, will be transformed into

$$\begin{aligned}(\alpha|000 + \beta|111>)|00> &\rightarrow \alpha|010>|00> + \beta|101>|00>) \\ &= \alpha|010>|11> + \beta|101>|11>) \\ &= (\alpha|010> + \beta|101>)\ |11>)\end{aligned}$$

The syndrome will also change from $|00>$ to $|11>$.

The error decoding procedure, that is, steps needed to correct an error are as follows:

i. If the syndrome is 00, there is no error. No action is needed.

ii. If the syndrome is 01, 10, or 11, then the erroneous qubit is Q_3, Q_1, or Q_2, respectively.

Apply X operation to the qubit identified by the syndrome bits.

In this example, the syndrome is 11, thus a bit-flip error occurred on qubit Q_2 (or 11 in binary). This error can be corrected with a X operation on the second qubit.

8.4.2 Phase Error Correction

The previous section discusses how the quantum information, $\alpha|0>+\beta|1>$, can be coded is not significantly different from the classical error correcting codes. However, there are other types of quantum errors that affect single qubits in a way that the repetition code cannot shield against. For example, a *phase-flip* error:

$$|k> \rightarrow (-1)^k|k> \quad \text{where } k \in \{0,1\}$$

may affect a qubit when it is transmitted through the channel to the receiver and could flip the relative phase of $|0>$ and $|1>$. In other words, the sign between $|0>$ and $|1>$ in a quantum state can become inverted when a *phase-flip error* occurs during transmission. Thus, a phase-flip error can be represented by the **Z** matrix:

$$\mathbf{Z}=\begin{pmatrix}1 & 0\\ 0 & -1\end{pmatrix}$$

The effect of a phase-flip error on a quantum state, $|\psi>$, can be understood by applying the phase-flip operator **Z** to $|0>$ and to $|1>$:

$$\mathbf{Z}|0>=\begin{pmatrix}1 & 0\\ 0 & -1\end{pmatrix}\begin{pmatrix}1\\ 0\end{pmatrix}=\begin{pmatrix}1\\ 0\end{pmatrix}=|0>$$

$$\mathbf{Z}|1>=\begin{pmatrix}1 & 0\\ 0 & -1\end{pmatrix}\begin{pmatrix}0\\ 1\end{pmatrix}=\begin{pmatrix}0\\ -1\end{pmatrix}=-|1>$$

Thus **Z** maps

$$(\alpha|0>+\beta|1>) \rightarrow (\alpha|0>-\beta|1>)$$

Obviously, there is no way of correcting this kind of error using the three-qubit code employed for bit-flip errors, since the syndrome measurement will produce 00. Consequently, the error correction operation does nothing and the error is not corrected.

Recall that two important basis states of a qubit are the $|+>$ and $|->$ states:

$$|+>=\frac{|0>+|1>}{\sqrt{2}}$$

$$|->=\frac{|0>-|1>}{\sqrt{2}}$$

The operator **Z** maps $|+>$ to $|->$ and $|->$ to $|+>$ as shown below:

$$|+> \rightarrow |-> \quad \text{and} \quad |-> \rightarrow |+>$$

$$\mathbf{Z}|+> = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \left(\frac{|0> + |1>}{\sqrt{2}} \right)$$

$$= \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} (|0> + |1>)$$

$$= \frac{1}{\sqrt{2}} (|0> - |1>) = |->$$

Similarly,

$$\mathbf{Z}|-> = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \left(\frac{|0> - |1>}{\sqrt{2}} \right)$$

$$= \frac{1}{\sqrt{2}} (|0> + |1>) = |+>$$

Hence a phase-error maps $|+>$ to $|->$ and $|->$ to $|+>$.

Since $|+> = H(1)$ and $|-> = H(0)$ a phase error can be converted to a bit-flip error if the basis is changed from the computational to the Hadamard. This change can be accomplished by using the identity *H-Z-H*:

$$H\text{-}Z\text{-}H = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}$$

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

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

$$= \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} = X$$

Thus, the identity *H-Z-H* converts phase-flip errors to bit errors *X*. (Note that in a similar manner, the *H-X-H* identity can be employed to transform from a bit-flip to a phase-flip error). Since,

$$|+> = \frac{|0> + |1>}{\sqrt{2}} = H(0), \quad |-> = \frac{|0> - |1>}{\sqrt{2}} = H(1)$$

it follows that

$$Z|+> = |-> \quad \text{and} \quad Z|-> = |+>$$

Since in the $(|+>, |->)$ basis a phase-flip error acts like a bit-flip error, the error correcting code for bit-flips can also be used for phase flips provided that the basis for each qubit is changed. In other words, if the quantum *g* state is encoded using the $(|+>, |->)$ basis, the error-detection and correction operations for phase errors can be performed just as in case of the bit-flip error. This allows the 3-qubit repetition-code to be applied to correct single phase-flip errors.

The quantum circuit for phase-flip error correction is obtained by encoding a single qubit state in triplicate as for the bit-flip errors:

$$\psi = \alpha|000> + \beta|111>$$

Then, a Hadamard gate is applied to each qubit as shown in Fig. 8.5. The Hadamard gates change the computational basis so that the phase-flip errors behave like bit-flip errors:

$$|0_L> \rightarrow |000> = H|0>H|0>H|0>$$

$$= \frac{(|0> + |1>)}{\sqrt{2}} \frac{(|0> + |1>)}{\sqrt{2}} \frac{(|0> + |1>)}{\sqrt{2}}$$

$$= |+>|+>|+>$$

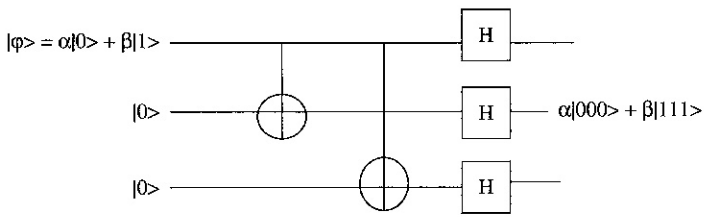


Figure 8.5 Circuit for making phase-flip errors behave as bit-flip errors.

and

$$|1_L\rangle \rightarrow\rangle |111\rangle = H|1\rangle H|1\rangle H|1\rangle$$
$$= \frac{(|0\rangle - |1\rangle)}{\sqrt{2}} \frac{(|0\rangle - |1\rangle)}{\sqrt{2}} \frac{(|0\rangle - |1\rangle)}{\sqrt{2}}$$
$$= |-\rangle|-\rangle|-\rangle$$

In the {+, −} basis, the phase-flip operator **Z** performs as the bit-flip operator *X*. Therefore, the bit-flip error correction code can be used for phase errors.

Next the basis is changed back from {|+>, |−>} to the original one, that is, {|0>, |1>} by using another set of three Hadamard gates (Fig. 8.6) so that the correction approach used for bit-flip errors can be employed to recover from the bit-phase errors.

To illustrate the phase error correction, assume a phase flip on Qubit 3. Then the encoding

$$\alpha(|+\rangle|+\rangle|+\rangle) + \beta(|-\rangle|-\rangle|-\rangle)$$

changes to

$$\alpha(|+\rangle|+\rangle|-\rangle) + \beta(|-\rangle|-\rangle|+\rangle)$$

The second Hadamard transform is then applied to all three qubits to restore the basis back from (|+>, |−>) to {|0>, |1>} resulting in

$$\alpha(|001\rangle + \beta(|110\rangle$$

Thus, the phase-flip error on Qubit 3 is transformed into a bit-flip error. Since,

$$\text{Qubit 1} \oplus \text{Qubit 2} = 0$$

$$\text{Qubit 2} \oplus \text{Qubit 3} = 1$$

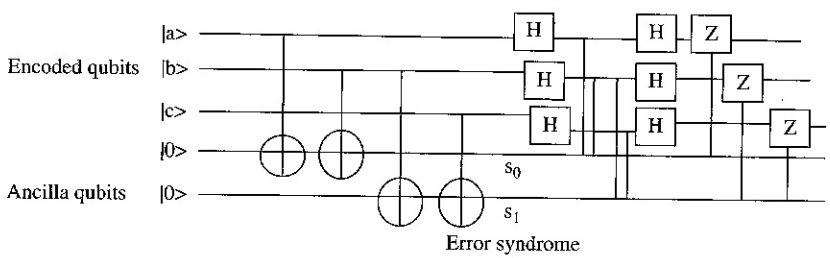


Figure 8.6 Correction circuit for both bit-flop and phase-flip error (Ref. 2).

for $|001>$ and $|110>$, the ancillary qubits, that is, the syndrome, will flip to 01 identifying Qubit 3 as the one that got corrupted by a phase-flip error during transmission. This error can be corrected by performing an X operation on the third qubit.

8.5 Shor's 9 Qubit Code

This code employs nine-qubits for each logical bit and can correct X (bit-flip), Z (phase-flip) errors or a combination of both ($Y = ZX$) on a qubit. It is based on the concept of majority voting and assumes the presence of a single erroneous qubit only. It has been shown that a quantum code that can correct both bit-flip and phase-flip errors, can correct an arbitrary error on a single qubit [4,5].

Shor's code is based on the 3-qubit repetition code, and encodes a single logical qubit into three physical qubits.

$$|0_L> \rightarrow |000> \quad \text{and} \quad |1_L> \rightarrow |111>$$

It then uses-two separate steps in the following order for encoding a *physical qubit*:

i. Employ phase encoding for the qubit state

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

to guard against phase-flip errors.

The phase-flip error, as shown earlier, is similar to the bit-flip error in the basis $\{|+>, |->\}$, thus the qubit state can be encoded as

$$\psi = \alpha|+> + \beta|->$$

Since the qubit state is encoded in the three-qubit repetition code, the following encoding of a qubit is appropriate for protection against phase-flip errors:

$$|0>_L = |+++>, \; |1>_L = |---|$$

Thus,

$$\psi = \alpha|+> + \beta|-> = |+++> + |---|$$

ii. Next, encode each of the resulting three qubits to protect each against bit-flip errors.

Figure 8.7 shows the encoding circuit. The first set of CNOT gates (on the left) in conjunction with the ancillary $|0>$ bits convert the quantum state

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

to

$$\psi = \alpha|000> + \beta|111>$$

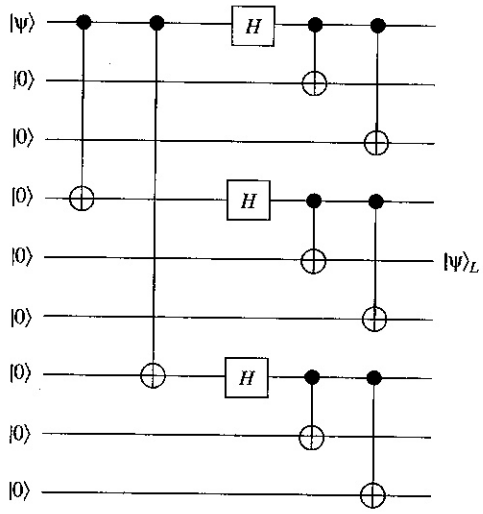


FIGURE 8.7 Encoding circuit for 9-qubit code [2].

Next, the Hadamard gates transform this state to

$$\alpha\left[\frac{1}{\sqrt{2}}(|000>+|111>)\cdot\frac{1}{\sqrt{2}}|000>+|111>)\cdot\frac{1}{\sqrt{2}}|000>+|111>)\right]$$

$$+\beta\left[\frac{1}{\sqrt{2}}(|000>-|111>)\cdot\frac{1}{\sqrt{2}}(|000>-|111>)\cdot\frac{1}{\sqrt{2}}|000>-|111>)\right]$$

Each output of the Hadamard gate drives the control inputs of a pair of CNOT gates as shown in Fig. 8.7, with $|0>$ feeding the other inputs of the CNOT gates.

The final outputs of the circuit show the single qubit state $\alpha|0>+\beta|1>$ is mapped into a product of 9 qubits, that is, an extension of 3-qubit repetition code:

$$\alpha|0>+\beta|1>\rightarrow$$

$$\frac{\alpha(|000>+|111>)\,(|000>+|111>)(|000>+|111>)+\beta(|000>-|111>)(|000>-|111>)(|000>-|111>)}{2\sqrt{2}}$$

$$=\frac{\alpha(|000>+|111>)^{\otimes 3}-\beta(|000>-|111>)^{\otimes 3}}{2\sqrt{2}}$$

The basis states for the code are encoded as

$$|0\rangle_L = \frac{(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)}{2\sqrt{2}}$$

$$|1\rangle_L = \frac{(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)}{2\sqrt{2}}$$

The first, fourth, and seventh qubits are used for the phase-flip code while all three block's of qubits, that is, (1, 2, 3), (4, 5, 6), and (7, 8, 9) are used for the bit-flip code. The overall encoded states, therefore consist of three *blocks*, each block containing three qubits:

1 2 3		4 5 6		7 8 9

Notice that all 9-qubit code words have the structure of a phase-flip code, but in each of them the $(|0\rangle \pm |1\rangle)$ part is replaced by $(|000\rangle \pm |111\rangle)$, that is, bit-flip encoding. This type of coding, that is, the nesting of one code inside another is called a *concatenated code*. This concept is very important in guarding against quantum computer's vulnerability to noise.

Single bit-flip or phase-flip errors in any of these blocks can be detected and corrected by using the error syndromes. As shown earlier, error syndromes for 3-qubit codes are generated by measuring whether or not two qubits are the *same* or *different* in standard basis. In Shor's code, there are three 3-qubit codes; they form three blocks with each block representing an encoding of one such code [6].

In block 1 the syndrome measurement, Z_1Z_2, compares the first and second qubit; if they are the same:

$Z_1Z_2 = +1$ if the first qubit and the second qubit are the same;

$= -1$ otherwise.

Similarly, syndrome measurement Z_2Z_3 compares the first and the third qubit in block 1;

$Z_2Z_3 = +1$ if the second qubit and the third qubit are the same;

$= -1$ otherwise.

From all possible combinations of the measurement results it is possible to determine (using the following table) whether there is a qubit flip error in block 1 and its location:

$Z_1Z_2 = +1$	$Z_2Z_3 = +1$	no error
$Z_1Z_2 = -1$	$Z_2Z_3 = +1$	qubit 1 has flipped
$Z_1Z_2 = -1$	$Z_2Z_3 = -1$	qubit 2 has flipped
$Z_1Z_2 = +1$	$Z_2Z_3 = -1$	qubit 3 has flipped

In a similar manner in block 2 syndrome measurement, Z_4Z_5 compares values of the fourth and the fifth qubits, and Z_5Z_6 compares values of the fifth and the sixth qubit, resulting in

$Z_4Z_5 = +1$	$Z_5Z_6 = +1$	no error
$Z_4Z_5 = -1$	$Z_5Z_6 = +1$	qubit 4 has flipped
$Z_4Z_5 = -1$	$Z_5Z_6 = -1$	qubit 5 has flipped
$Z_4Z_6 = +1$	$Z_5Z_6 = -1$	qubit 6 has flipped

Finally, in block 3 syndrome measurement, Z_7Z_8 compares values of the seventh and the eight qubits, and Z_7Z_9 compares values of the eighth and the ninth qubit, resulting in

$Z_7Z_8 = +1$	$Z_8Z_9 = +1$	no error
$Z_7Z_8 = -1$	$Z_8Z_9 = +1$	qubit 7 has flipped
$Z_7Z_8 = -1$	$Z_8Z_9 = -1$	qubit 8 has flipped
$Z_7Z_8 = +1$	$Z_8Z_9 = -1$	qubit 9 has flipped

To illustrate the application of Shor's 9-bit coding, assume that a qubit $\alpha|0> + \beta|1>$ is encoded as $\alpha|000> + \beta|111>$, and the three qubits are sent one at a time via a channel.

Suppose a bit-flip error occurs on the seventh qubit (shown in bold below),

$$\frac{\alpha(|000> + |111>)\,(|000> + |111>)(|\mathbf{1}00> + |\mathbf{0}11>) + \beta(|000> - |111>)(|000> - |111>)(|100> - |\mathbf{1}11>)}{2\sqrt{2}}$$

Then the qubits in the three blocks will be

Blocks	1	2	3
	000	000	100

Error syndrome measurements can be obtained from three blocks of qubits; the six observables are derived by comparing the appropriate qubits in blocks 1, 2, and 3:

Z_1Z_2	Z_2Z_3	Z_4Z_5	Z_5Z_6	Z_7Z_8	Z_8Z_9
+1	+1	+1	+1	−1	+1

It is clear that the bit-flip error occurred on the seventh qubit, that is, the first qubit of the third block above. The original state can then be recovered by applying an X gate to qubit 7.

References

1. Jeremy Hsu, IBM shows first full error detection for quantum computers. IEEE Spectrum, April 29, 2015.
2. Simon J. Devitt, William J. Munro and Kae Nemoto, Quantum Error Correction for Beginners, arXiv:0905.2794v4 [quant ph] 21 June, 2013.
3. John Watrous, CPSC 519/619: Quantum Computation Lecture 16: Quantum error correction. University of Calgary, 2006.
4. A. R. Calderbank and Peter W, Shor, Good quantum error-correcting codes exist, *Phys. Rev. A*, 54:1098–1105, 1996. quant-ph/9512032.
5. Andrew M. Steane, Multiple particle inference and quantum error correction, *Proc. Roy. Soc. A.*, 452:2551, May 1996. quant-ph/9601029.
6. Michael Nielsen and Issac Chuang, *Quantum Computation and Quantum Information*, Cambridge University Press, 2000.

CHAPTER 9
Quantum Algorithms

Quantum computers utilize the unique characteristics of the quantum systems to process exponentially large amount of data in a very short time. Obviously this kind of computing power has the potential application in mathematics, cryptography, and also in the simulation of quantum systems themselves. A fundamental feature of many of these algorithms is that they allow a quantum computer to evaluate a function $f(x)$ for many different values of x simultaneously.

9.1 Deutsch's Algorithm

Deutsch's algorithm does not solve any particularly important problem in computer science. However, it is the first to provide proof that in certain cases quantum computers are significantly faster than traditional computers [1, 3, 7, 8]. Deutsch found solution for a certain problem in a single step whereas a classical approach needs two.

Consider a function f that maps $\{0, 1\}$ into $\{0, 1\}$. If it is known that $f(x)$ is either *constant* (0 or 1 for all values of x) or *balanced* (0 for exactly half for all possible x and 1 for the other half), then the problem is to decide what type it is. All possible function mappings are shown in Fig. 9.1.

Assume this function is implemented inside a *black box* (also called an *oracle*). Thus, the only way to gain any information about the function is to apply some input $x \in \{0, 1\}$ and check the device output $f(x) \in \{0, 1\}$.

If a classical computer is used to determine whether f is balanced or constant, the computer needs to evaluate the function both for $x = 0$ and for $x = 1$ separately, and then compare the two outputs to decide whether $f(x = 0)$ is indeed equal to $f(x = 1)$. Notice that in the left column of Fig. 9.1 the function produces a constant value 0 or 1 for the inputs, thus f is a *constant function*. Alternatively, on the right column for one-half of the inputs the function produces 0 and for the other half it produces 1; in this case f is a *balanced function*.

Deutsch's algorithm evaluates the function by framing the problem in a slightly different manner. Instead of checking out the values $f(0)$ and $f(1)$ separately the algorithm determines value of f simultaneously for $x = 0$ and $x = 1$. A function $f(x)$ is balanced if the value of $f(x = 0) \oplus f(x = 1) = 1$, even though it is not possible to determine the

$f(x)$	$f(x)$
$0 \rightarrow 0$	$1 \rightarrow 0$
$1 \rightarrow 0$	$0 \rightarrow 1$
$0 \rightarrow 1$	$1 \rightarrow 1$
$1 \rightarrow 1$	$0 \rightarrow 0$

Figure 9.1 All possible mappings of a function {0,1} to {0,1}.

actual value. Alternatively, the function is constant if $f(x = 0) = f(x = 1)$. Thus, determining $f(x = 0) \oplus f(x = 1) = 1$ provides the answer to the question whether the function f is constant or balanced. A quantum circuit explaining Deutsch's algorithm is shown in Fig. 9.2.

The given function $f(x)$ is first transformed into $|x, y \oplus f(x)>$ using a two-qubit quantum register $|x\ y>$ and a unitary transform operation that is identified as U_f:

$$U_f(|x>|y>) = |x>\ |y \oplus f(x)>$$

U_f leaves the first bit x unchanged and inverts the second bit y if the function applied to x, that is, $f(x) = 1$, otherwise x and y remain unchanged. Assume in the description of the algorithm, (00) means that both qubits have value 0 and likewise for 01, 10, and 11; these bits represent the two possible inputs to $f(x)$. They are applied in superposition. For example, if the quantum register is prepared in state (01), state (01) is fed to $f(x)$. This is done by first passing $|x>$ and $|y>$ through Hadamard gates H_x and H_y, respectively, as shown in Fig. 9.2.

H_x transforms $|x>$ that is $|0>$ into $\frac{1}{\sqrt{2}}(|0>+|1>)$ and H_y transforms $|y>$ that is $|1>$ into $\frac{1}{\sqrt{2}}(|0>-|1>)$. Then the two-qubit input to U_f is

$$\frac{1}{\sqrt{2}}(|0>+|1>)\cdot\frac{1}{\sqrt{2}}(|0>-|1>)$$

$$=\frac{1}{2}[|00>-|01>+|10>-|11>]$$

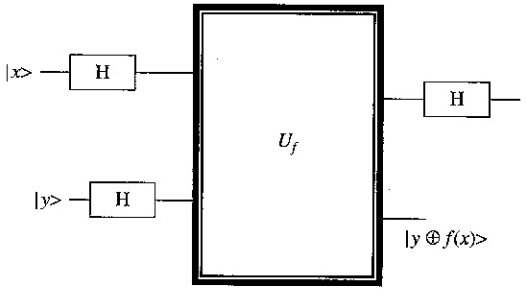


Figure 9.2 Quantum circuit for Deutsch algorithm.

Next the above result is passed through the U_f transform that yields as shown in Fig. 9.2.

$$\begin{aligned}
&\frac{1}{2}\,|0>(|0>\oplus f(0)>)-\frac{1}{2}\,|0>(|1>\oplus f(0)>)\\
&\quad+\frac{1}{2}\,|1>(|0>\oplus f(1)>)-\frac{1}{2}\,|1>(|1>\oplus f(1)>)\\
&=\frac{1}{2}\,|0>(|0>\oplus f(0)>)-|1>\oplus f(0)>)\\
&\quad+\frac{1}{2}\,|1>(|0\oplus f(1)>)-|1\oplus f(1)>)
\end{aligned}$$

Using the following formula [3]

$$|0\oplus\alpha>-|1\oplus\alpha>=(-1)^{\alpha}(|0>-|1>) \quad \text{where} \quad \alpha\in\{0,1\}$$

the above equation can be represented as

$$\begin{aligned}
&\frac{1}{2}|0>[(-1)^{f(0)}(0>-\frac{1}{2}-(-1)^{f(0)}(1>]\\
&\quad+\frac{1}{2}-[|1>[(-1)^{f(1)}(0>-\frac{1}{2}-(-1)^{f(1)}(1>]\\
&=\frac{1}{2}-[(-1)^{f(0)}\,|0>(|0>-|1>]+\frac{1}{2}[(-1)^{f(1)}\,|1>(|0>-|1>)]\\
&=\frac{1}{\sqrt{2}}-[(-1)^{f(0)}\,|0>+(-1)^{f(1)}\,|1>]\,\frac{(|0>-|1>)}{\sqrt{2}}\\
&\quad\text{(factoring out } (|0>-|1>)/\sqrt{2})
\end{aligned}$$

Notice that the second qubit has not been changed by the U_f transformation. Hence, the output of the second qubit could be discarded. The state of the first qubit remains unchanged:

$$\frac{1}{\sqrt{2}}-[(-1)^{f(0)}\,|0>+(-1)^{f(1)}\,|1>]$$

This can be rewritten as

$$(-1)^{f(0)}[\frac{1}{\sqrt{2}}|0>+\frac{1}{\sqrt{2}}(-1)^{f(0)\oplus f(1)}\,|1>]$$

based on the following result [3]

$$(-1)^{f(0)}(-1)^{f(1)}=(-1)^{f(0)\,\oplus\,f(1)}.$$

The Hadamard gate at output x transforms the state to

$$(-1)^{f(0)}\,|0>$$

If $f(0) \oplus f(1) = 1$, that is, f is a balanced function then the state of the first qubit is

$$\left[(-1)^{f(0)} \frac{|0> - |1>}{\sqrt{2}}\right]$$

and so the Hadamard gate at output x transforms the state to

$$(-1)^{f(0)} |1>$$

The final Hadamard gate results in the value

$$(-1)^{f(0)} |f(0) \oplus f(1)>$$

This means that the first qubit at output x provides the value of $f(0) \oplus f(1)>$, thereby indicating whether the function is constant or balanced.

9.2 Deutsch-Jozsa Algorithm

Deutsch's algorithm works on a single input bit in the simple case where f:(0, 1) → (0, 1). A generalization of the algorithm known as Deutsch-Jozsa algorithm can act on an n-bit function f:$(0, 1)^n \rightarrow (0, 1)$. Assuming a 2-bit function of the form $(0, 1)^2 \rightarrow (0, 1)$ is provided and it is *known* at the outset that the function is one of the four shown below, the problem is to identify which one this is [2, 3, 8, 9].

Example [9]

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

In the worst case, the function (in a black box) needs to be called $2^{n-1} + 1$ times to check whether it is balanced or constant. Deutsch and Jozsa introduced an algorithm that can answer this with 100% accuracy using only *one* query [2].

The Deutsch-Jozsa algorithm uses two quantum registers x and y, x having n qubits and y having only 1 qubit. The circuit diagram of the Deutsch-Jozsa algorithm is shown in Fig. 9.3. As can be seen in the diagram, the total number of input qubits is $(n + 1)$ that is the sum of qubits in the x and the y register. The n qubits in x register are

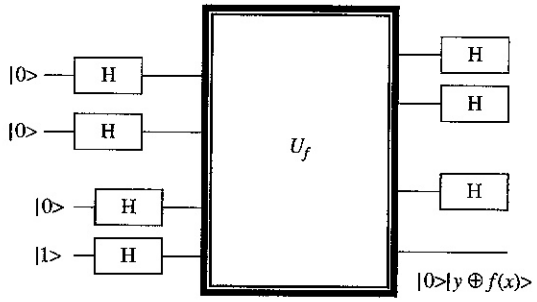


FIGURE 9.3 Quantum circuit for Deutsch-Jozsa algorithm.

initialized to $|0\rangle$, and the qubit in y register is initialized to a $|1\rangle$. Hence, the quantum state of the circuit at the start is $|0\rangle^{\otimes n}|1\rangle$

$$|\psi_0\rangle = |0\rangle \ldots \ldots |0\rangle|1\rangle$$
$$= |0\rangle^{\otimes n}|1\rangle$$

The notation $|0\rangle^{\otimes n}$ means n consecutive $|0\rangle$ qubits.

Next, the Hadamard transformation H is applied separately to each qubit in the x and y register. This results in tensor products of $(n + 1)$ 1-qubit Hadamard gates (acting in parallel). For example, the Hadamard transformation of each qubit of a two-qubit register assuming the register is initialized to $|0\rangle$ can be represented as

$$H|0\rangle \otimes H|0\rangle$$
$$= \left(\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\right) \otimes \left(\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\right)$$

Similarly, the Hadamard transformation of the contents of an n-qubit register initially at state $|0\rangle$ will result in

$$\left(\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\right) \otimes \left(\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\right) \cdots\cdots\cdots \otimes \left(\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\right)$$

The expansion of the tensor product terms will result in the following representation of the above result

$$\sum_{x \in (0,1)^n} \frac{1}{\sqrt{2^n}} |x\rangle$$

The notation $(0, 1)^n$ means all possible bit strings of size n. For example, if $n = 2$, this would be 00, 01, 10, and 11. Thus, the 1-qubit Hadamard gates acting on the n-qubit state of all zeros yields a

sum of 2^n terms, each of which is a unique string of $(0, 1)^n$ and the amplitudes of each is $\frac{1}{\sqrt{2^n}}$.

The Hadamard transformation applied to a 1-qubit register initialized to $|1>$ will result in

$$H|1> = \frac{1}{\sqrt{2}}|0> - \frac{1}{\sqrt{2}}|1>$$

Hence in the Deutsch-Jozsa algorithm the quantum state, $|\Psi_1>$ of the $(n + 1)$-qubit register after the first Hadamard transformation is

$$|\psi_1> = H^{\otimes n}|0>^{\otimes n} H|1>$$

$$= \sum_{x\epsilon(0,1)^n} \frac{1}{\sqrt{2^n}}|x>\left(\frac{|0> - |1>}{\sqrt{2}}\right)$$

Next the U_f operation is applied to $|\Psi_1>$ that is transformed into

$$|\Psi_2> = \sum_{x\epsilon(0,1)^n} \frac{1}{\sqrt{2^n}}|x>\left(\frac{|0 \otimes f(x)>) - |1 \otimes f(x)>)}{\sqrt{2}}\right)$$

When $f(x) = 0$, $\Psi_2> = \sum_{x\epsilon(0,1)^n} \frac{1}{\sqrt{2^n}}|x>\left(\frac{|0> - |1>}{\sqrt{2}}\right)$

but when $f(x) = 1$, $\Psi_2> = \sum_{x\epsilon(0,1)^n} \frac{1}{\sqrt{2^n}}|x>\left(\frac{|1> - |0>}{\sqrt{2}}\right)$

Both possibilities can be included in $|\Psi_2>$ by rewriting it as

$$|\Psi_2> = \sum_{x\epsilon(0,1)^n} \frac{(-1)^{f(x)}}{\sqrt{2^n}}|x>$$

As in the case of Deutsch algorithm the last qubit is discarded at this stage, and another Hadamard transformation is applied to each of the remaining qubits.

The action of the last set of Hadamard gates on an n-qubit state $|x> = |x_1>|x_2>...|x_n>$ is given by

$$H^{\otimes n}|x> = H|x_1> \otimes H|x_2> \otimes H|x_3> \ldots\ldots\ldots \otimes H|x_n>$$

$$= \frac{1}{\sqrt{2^n}} \sum_{y\epsilon(0,1)} (-1)^{x_1 \cdot y_1}|y_1> \ldots\ldots\ldots \sum_{y\epsilon(0,1)} (-1)^{x_n \cdot y_n}|y_n>$$

$$= \frac{1}{\sqrt{2^n}} \sum_{y\epsilon(0,1)^n} (-1)^{x_1 \cdot y_1 + \cdots + x_n \cdot y_n}|y>$$

$$= \frac{1}{\sqrt{2^n}} \sum_{y\epsilon(0,1)^n} (-1)^{x \cdot y}|y>$$

where $x \cdot y$ is $\sum_{i=1}^{n} a_i b_i$.

The state after the final set of Hadamard gates is

$$|\psi_3> = |\psi_2> \cdot H^{\otimes n}$$

$$= \frac{1}{\sqrt{2^n}} \sum_{x\epsilon(0,1)^n} (-1)^{f(x)} \frac{1}{\sqrt{2^n}} \sum_{y\epsilon(0,1)^n} (-1)^{x \cdot y} |y>)$$

$$= \frac{1}{\sqrt{2^n}} \sum_{x\epsilon(0,1)^n} (-1)^{f(x)} \frac{1}{\sqrt{2^n}} \sum_{y\epsilon(0,1)^n} (-1)^{x \cdot y} |y>)$$

$$= \frac{1}{2^n} \sum_{x\epsilon(0,1)^n} \left(\sum_{y\epsilon(0,1)^n} \right) (-1)^{f(x)+xy} |y>$$

This implies that the function $f(x)$ is *constant* if there is a unity probability of obtaining $|00 \ldots 0>$ on measurement of the n qubit register, otherwise the function is *balanced*.

9.3 Grover's Search Algorithm

Grover's algorithm performs a search over an unstructured and unsorted database of N entries for accessing a particular entry [4, 5, 6]. Using a classical computation model a solution can be obtained by checking every item in the database to find the desired one. This search in the worst case will require a time-complexity $O(N)$. Grover developed an algorithm that finds a *marked* item, that is, the item of interest x^* in a set of N elements $(x_1, x_2, \ldots, x_n)$; it completes the search in time complexity $O(\sqrt{N})$ by utilizing the nature of quantum systems.

All N items in the database are simultaneously encoded in n qubits, where n is the number of qubits necessary to represent the search space of $2^n = N$. The items in the database are labeled as n-bit Boolean strings in $\{0, 1\}^n$, not indexed from 1 to N. For example, for two qubits there are four possible combinations. These values are obtained by putting both qubits in the superposition state $\frac{1}{\sqrt{2}}(|0> + (|1>)$ and taking their tensor product

$$\frac{1}{\sqrt{2}}(|0> + (|1>) \otimes \frac{1}{\sqrt{2}}(|0> + (|1>)$$

$$= \frac{1}{2}|00> + \frac{1}{2}|01> + \frac{1}{2}|10> + \frac{1}{2}|11>.$$

Grover's algorithm begins with a quantum register of n qubits all initialized to $|0\rangle$:

$$|0 \otimes n\rangle = |0\rangle$$

The first step is to the put the system into an equal superposition of states by using Hadamard gates:

$$|\psi\rangle = H^{\otimes n} |0>^{\otimes n} = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x>$$

Next a series of transformations, often referred to as the *Grover iteration*, is applied R times to the superposition state where

$$R = \frac{\pi}{4} \sqrt{2^n}$$

The first step in the Grover iteration is a call to the quantum oracle O, that can observe and modify the system without collapsing it to a classical state

$$|x\rangle \dashrightarrow O(-1)^{f(x)} f(x) |x\rangle$$

where $f(x) = 1$ if x is the searched state, and 0 otherwise.

Then a selective phase inversion is performed which switches the sign of the amplitude of the searched state; for the purposes of this illustration the searched state is the fourth state.

Finally, the inversion about average operation is performed, which increases the amplitude of the state that was inverted in the previous step.

9.3.1 Details of Grover's algorithm

The algorithm begins with a quantum register of n qubits, where n is the number of qubits necessary to represent the search space of size $2^n = N$, all initialized to $|0>$

$$|0>^{\otimes n} = |0>$$

The next step is to put the system into an equal superposition of states; this is achieved by applying the Hadamard transform H to the qubits. In a system of n qubits, n applications of the Hadamard transformation is needed to form a superposition of each of the states individually:

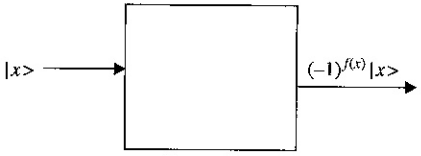
$$\begin{aligned} &H|0> \otimes H|0> \otimes \ldots \otimes H|0> \\ &\quad = (H \otimes H \otimes \ldots \otimes H)\ |00\ldots0> \\ &\quad = \frac{1}{\sqrt{2^n}} (|0> + |1>) \otimes (|0> + |1>) \otimes \ldots \ldots \otimes (|0> + |1>) \\ &\quad = \frac{1}{\sqrt{2^n}} (|00\ldots.00> + |00\ldots.01> + |00\ldots.11> + \ldots\ldots + |11\ldots11>) \end{aligned}$$

This represents the sum for the decimal numbers from 0 to $2^n - 1$ written in the binary notation, that is,

$$= \frac{1}{\sqrt{2^n}} \sum_{n=0}^{2^n-1} |n>$$

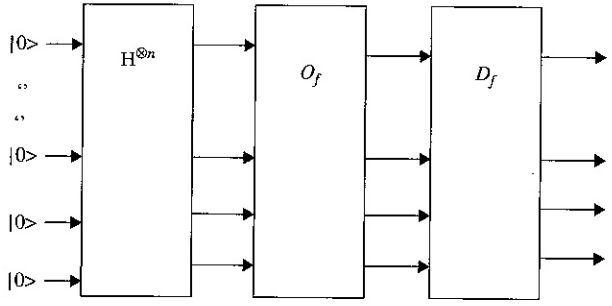
This transformation leads to equal amplitude of $\frac{1}{\sqrt{2^n}}$ with every possible configuration of qubits in the system, and thus can be in any of the n possible states with equal probability of $\frac{1}{\sqrt{2^n}}$.

As stated earlier Grover's algorithm requires an oracle that maps an input $(0, 1)^n$ to (0, 1). The oracle may be considered as a black box, that is, the details of how it works are not of concern. A gate corresponding to an oracle produces an 1-bit output for an n-bit input. Since the oracle is not unitary nor reversible, it is not a valid quantum gate. One possible solution is to add an extra bit c, called an *ancilla*, to use for the output. If $f(x) = 0$, the ancilla qubit is left unchanged; if however $f(x) = 1$, the phase of the ancilla is inverted. The inclusion of an additional bit however can be avoided by flipping the input when $f(x) = 1$ as shown in the gate below:



$$f(x) = (-1)^{f(x)} |x> = \begin{cases} 1 & if \quad f(x) = 0 \\ -|x> & if \quad f(x) = 1 \end{cases}$$

The uniform superposition state obtained by using Hadamard gates is then queried by the oracle gate that flips the amplitude of the $x*$ item and leaves everything else unchanged. The amplitude of $x*$ is increased next by incorporating a new gate that functions as what is called the *Grover diffusion operator* D_f as depicted below:



As an example of the Grover's algorithm, consider a system consisting of $N = 8 = 2^3$ states [5, 6]. Suppose the state under search, for example, x^* is represented by the bit string 100.

The system can be represented by 3 qubits, $N = 3$. Assume that the superposition state $|x>$ corresponding to three qubits is

$$|x> = \alpha_0 |000> + \alpha_1 |001> + \alpha_2 |010> + \ldots\ldots\ldots\ldots\ldots \alpha_7 |111>$$

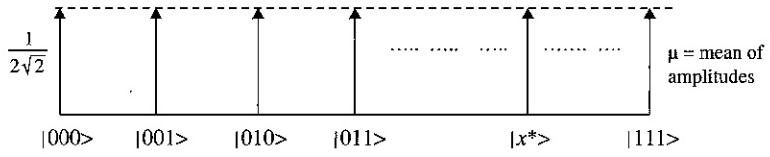
where α_i is the amplitude of the state $|i>$. Each combination of the three qubits is the binary notation of numbers 0 to 7. These are obtained by putting the three qubits into the superposition state $\frac{1}{\sqrt{2}}$ $(|0> + |1>)$ so that the combined state of the three qubits is

$$\frac{1}{2\sqrt{2}}(|0> + |1>) \otimes \frac{1}{2\sqrt{2}}(|0> + |1>) \otimes \frac{1}{2\sqrt{2}}(|0> + |1>)$$

$$= \frac{1}{2\sqrt{2}} |000> + \frac{1}{2\sqrt{2}} |001> \ldots\ldots\ldots\ldots \frac{1}{2\sqrt{2}} |111>$$

Since the amplitude of each state is $\frac{1}{2\sqrt{2}}$, the associated probability is $(\frac{1}{2\sqrt{2}})^2 = \frac{1}{8}$. Thus, the total probability of measuring one of these eight states is $8 \cdot \frac{1}{8} = 1$.

The amplitudes may be visualized as lines perpendicular to an axis as shown below; the length of a line is proportional to the amplitude it represents. For example, the equal superposition of states resulting from the first Hadamard transformation appears as follows:



The states (bit strings) are labeled as {000, 001… …. 111} and x^* is the unknown marked state. The uniform amplitudes of all the n-bit strings indicate that x^* at this stage cannot be distinguished from the other strings.

Next the superposition state is fed to the *oracle* G_f.

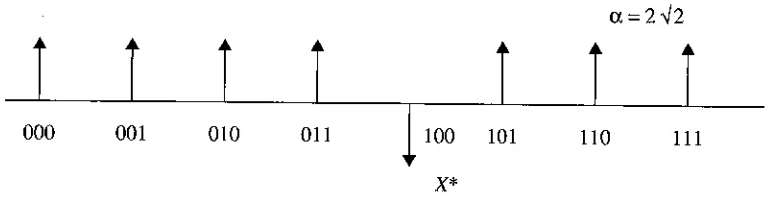
If $f(x) = 1$ then $|x>$ is the correct state and the qubit picks up a phase factor of -1 that recognizes the solution, however, if $f(x) = 0$ the state remains unchanged:

$$f(x) = \begin{cases} 1 & \text{if } x \text{ is the search element } (x^*) \\ 0 & \text{otherwise} \end{cases}$$

Assume the application of the oracle G_f on this state flips the amplitude of the state marked state 100 (denoted as x^* in the diagram) but leaves everything else unchanged, resulting in the following state:

$$-\frac{1}{\sqrt{N}}|x^*>+\sum_{\substack{x=0\\x\neq x^*}}\frac{1}{\sqrt{N}}|x>$$

Note that the phase of $|100>$ in superposition is inverted and it becomes $-|100>$, and the corresponding line points down as shown below:



Define $|u>$ as

$$|u>=\sum_{\substack{x=0,\\x\neq 100}}^{7}\frac{1}{\sqrt{7}}|x>\ldots\ldots$$

$$=\frac{1}{\sqrt{7}}|000>+\frac{1}{\sqrt{7}}|001>+\frac{1}{\sqrt{7}}|010>+\frac{1}{\sqrt{7}}|011>$$

$$+\frac{1}{\sqrt{7}}|101>+\frac{1}{\sqrt{7}}|110>+\frac{1}{\sqrt{7}}|111>$$

Thus, u is a superposition of all basis states with equal amplitudes given by $1/\sqrt{N}$

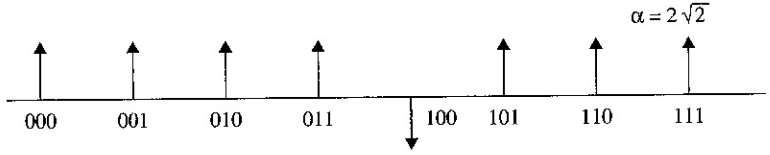
Then $|\psi>$ can be represented as

$$|\psi>=\frac{\sqrt{7}}{2\sqrt{2}}|u>+\frac{1}{2\sqrt{2}}|100>$$

Next this state is applied to the oracle defined earlier, this results in the following

$$|\psi>=\frac{\sqrt{7}}{2\sqrt{2}}|u>-\frac{1}{2\sqrt{2}}|100>$$

The phase of $|100>$ in superposition is inverted and the corresponding line points down as shown below; there are no other changes:



Although the application of the oracle identifies a marked entry because of its phase difference from the others, it cannot be distinguished from other entries since the amplitudes of all are the same. Next an additional operation known as the *diffusion transformation* is utilized; it increases the amplitudes by their difference from the average, and decreases the amplitudes if the difference is negative.The alteration of the amplitudes of unmarked entries makes it easier to measure the final state and gives the correct result with a very high probability.

The application of diffusion transmission operator to $|\psi_1>$ results in

$$|\psi_{2>} = [2|\psi><\psi| - I]|\psi_1>$$

$$= [2|\psi><\psi| - I]\left[|\psi> - \frac{1}{2\sqrt{2}}|100>\right]$$

$$= 2|\psi><\psi|\psi> - |\psi> - \frac{2}{2\sqrt{2}}|\psi><\psi|100> + \frac{1}{\sqrt{2}}|100>$$

Since $<\psi|100>$ is one of the basis vectors, $<100|\psi> = <\psi|100> = \frac{1}{2\sqrt{2}}$,

$$= 2|\psi> - |\psi> - \frac{2}{\sqrt{2}}\left(\frac{1}{2\sqrt{2}}\right)|\psi> + \frac{1}{\sqrt{2}}|100>$$

$$= \frac{1}{2}|\psi> + \frac{1}{\sqrt{2}}|100>$$

Substituting $|\psi> = \frac{1}{2\sqrt{2}}\sum_{0}^{7}|u>$,

$$|\psi> = \frac{1}{2\sqrt{2}}\sum_{0}^{7}|u> + \frac{1}{\sqrt{2}}|100>$$

$$= \frac{1}{4\sqrt{2}}\sum_{0}^{7}|u> + \frac{1}{4\sqrt{2}}|100> + \frac{1}{\sqrt{2}}|100>$$

$$= \frac{1}{4\sqrt{2}}\sum_{0}^{7}|u> + \frac{5}{4\sqrt{2}}|100>$$

$$= \frac{1}{4\sqrt{2}}\sum_{\substack{x=0 \\ x\neq 4}}^{7}|u> + \frac{5}{4\sqrt{2}}$$

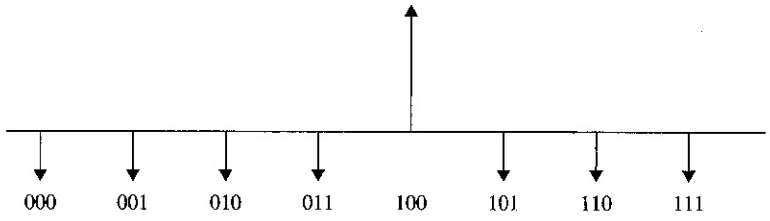


FIGURE 9.4 Amplitude of 100 is much larger than that of other states.

This expression can be rewritten as shown previously

$$|x\rangle = \frac{1}{4\sqrt{2}}|000\rangle + \frac{1}{4\sqrt{2}}|001\rangle + \frac{1}{4\sqrt{2}}|010\rangle + \frac{1}{4\sqrt{2}}|011 - \frac{5}{4\sqrt{2}}|100\rangle$$

$$+ \frac{1}{4\sqrt{2}}|101\rangle \ldots\ldots + \frac{1}{4\sqrt{2}}|111\rangle]$$

$$= \frac{1}{4\sqrt{2}}\sum_{x=0,}^{7}|x\rangle - \frac{5}{4\sqrt{2}}|100\rangle$$

Another application of the Grover iteration will result in the following transformation:

$$= -\frac{1}{8\sqrt{2}}\sum_{x=0,}^{7}|x\rangle + \frac{11}{8\sqrt{2}}|100\rangle$$

The graphical representation of the expression is shown in Fig. 9.4. Note that the amplitude of $|100\rangle$ is much larger than the amplitude of any other state in the expression. Indicating that it is the state with the desired information.

9.4 Shor's Factoring Algorithm

Shor's algorithm reduces the prime factorization problem into a problem of *order* (or *period*) finding [8, 10, 11]. It is based on a result of the number theory that the function

$$f(r) = x^r \bmod N$$

is a periodic function when x is an integer that is co-prime to N, that is, they do not share common factors. Shor's algorithm tries to find r, the period of $x^a \bmod N$, where N is assumed to be the number to be factored.

The *order* of an integer x mod N is the smallest integer r such that

$$x^r = 1 \bmod N$$

For example, consider the sequence of numbers

$$2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, \ldots$$

If mod 15 is taken for each of the above numbers, a new sequence of numbers consisting of the remainders of the division of the above numbers by 15 is produced:

$$2, 4, 8, 1, \ldots$$

As shown above taking the powers of two mod 15 yields a periodic sequence whose *period* (or *order*) is four. For another example, consider the same powers of two, with mod 21 taken for each number; the resulting sequence is also periodic with a period 6:

$$2, 4, 8, 16, 11, 1 \ldots$$

The key idea underlying Shor's algorithm is based on a result of the number theory that the function

$$f(r) = x^r \bmod N$$

is a periodic function when x is an integer that is co-prime to N, that is, they do not share common factors. Shor's algorithm tries to find r, the period of x^a mod N, where N is assumed to be the number to be factored:

i. Choose an integer q that is equal to power of 2 and is defined to be

$$N^2 \leq q \leq 2N^2$$

ii. Choose a random integer x that it is a co-prime to N. Two numbers are co-primes if their greatest common divisor is 1, that is, GCD(x, N) = 1.

iii. Create a quantum register R and partition it into two separate registers: Register 1 and Register 2. Register 1 is called the *input register* and must have sufficient number of qubits to represent any integer up to $q - 1$. Register 2 known as the *output register* must also have sufficient qubits to represent any integers up to $N - 1$. Registers 1 and 2 must be entangled so that the collapse of the input register leads to the collapse of the output register.

iv. Apply Hadamard transformation to each qubit of R_1. This initializes register R1 with an equally weighted superposition of all integers a (from 0 to $q - 1$); also initialize R2 with all 0s. The combined state of the quantum memory register after this step will be

$$\frac{1}{\sqrt{q}} \sum_{a=0}^{q-1} | a, 0 >$$

v. Compute the function $x^a \bmod N$ for each number stored in Register 1 and store the result in Register 2. Because of the quantum parallelism the computation $x^a \bmod N$ can be completed in one step on a quantum computer. After this step the quantum memory register will be in the state

$$\frac{1}{\sqrt{q}} \sum_{a=0}^{q-1} |a, x^a \bmod N>$$

vi. Measure the output register and get the collapsed output $|c>$. Since the output register collapsed to c the input register in turn collapses into an equal superposition of each value of a between 0 and $q-1$ that yielded the collapsed output $|c>$

$$x^a \bmod N = c$$

This operation is executed on a quantum computer. The state of the quantum memory register after this step is

$$|\psi_3> = \frac{1}{\sqrt{||A||}} \sum_{a' \in A} |a', c>$$

where A is the set of a's such that $x^a \bmod N = c$, and $||A||$ is the number of elements in set A.

vii. Apply the QFT (Quantum Fourier Transform) to register R1. The Fourier transform when applied to a state $|a>$ changes it to

$$|a> = \frac{1}{\sqrt{q}} \sum_{c=0}^{q-1} e^{2\pi i ac/q} |c>$$

This step is performed on the quantum computer in one step. The state of the quantum memory register after this step is

$$\frac{1}{\sqrt{q}} \sum_{a=0}^{q-1} |a> |x^a \bmod(N)>$$

$$= \frac{1}{\sqrt{q}} \sum_{a=0}^{q-1} \frac{1}{\sqrt{q}} \sum_{c=0}^{q-1} e^{2\pi i ac/q} |c> |x^a \bmod(N)>$$

$$= \frac{1}{q} \sum_{a=0}^{q-1} \sum_{c=0}^{q-1} e^{2\pi i ac/q} |c> |x^a \bmod(N)>$$

viii. Next the state of the input register is measured. This integer referred to as m is very likely to be a multiple of $\frac{q}{r}$, where r is the desired period. QFT increases the probability amplitudes of all values that $x^a \bmod N$ yielded while the other values in the registrar remain unaffected; this step is performed on a classical computer.

ix. The value of r is derived based on the knowledge of m and q on a classical computer using, for example the *continuous fraction expansion* (11).

x. Next r is checked to determine whether it is even, and also whether $x^{\frac{r}{2}} \bmod n \neq -1$. If both conditions hold, the factors of N can be determined by taking the greatest common divisor (GCD) of N with respect to $x^{\frac{r}{2}} + 1$ and $x^{\frac{r}{2}} - 1$.

As an example of the Shor's algorithm consider factoring of $N = 21$. Choose an integer q such that the condition in step 2 of the algorithm, $N^2 < 2^q < 2N^2$, is satisfied; let q be 9, this is the smallest value of q that allows 2^q between N^2 and $2N^2$. Choose a random integer x such that GCD(x, 21) = 1; suppose $x = 11$.

Assume the initial state of the quantum register consisting of Registers 1 and 2 of length l (= $x + q$) be

$$|\psi_1> = |0>|0>$$

where the first register x has 9 qubits and the second q has 5 qubits.

The combined wave function of input register R1 and output register R2 after this step is

$$|\psi_1> = \frac{1}{\sqrt{512}} \sum_{a=0}^{511} |a>|0>$$

Initialize Register 2 with superposition of all states $x^a \pmod N$

$$|\psi_1> = \frac{1}{\sqrt{512}} \sum_{a=0}^{511} |a>|11^a (\bmod 21)>$$

The next step is to compute the function $f(a) = 11^a \bmod 21$ on Register 2, that is, the output register; this yields

$$|\psi_2> = \frac{1}{\sqrt{512}} \sum_{a=0}^{511} |a>|f(a)>$$

$$= \frac{1}{\sqrt{512}} \sum_{a=0}^{511} |a>|11^a \bmod 21>$$

$$= \frac{1}{\sqrt{512}} (|0>|1> + |1>|11> + |2>|16> + |3>$$
$$|8> + |4>|4> + |5>|2> + |6>|1> + |7>$$
$$|11> + |8>|16> + |9>|8> + |10>|4>$$
$$+ |11>|2> \cdots + \cdots)$$

This expression can be rewritten by separating Registers 1 and 2 as shown below:

$$\leftarrow - - \text{Register 1} - - - \rightarrow \leftarrow \text{Register 2} \rightarrow$$

$$= \frac{1}{\sqrt{512}} [(|0> + |6> + \ldots\ldots\ldots\ldots + |510>) |1> +$$

$$(|1> + |7> + \ldots\ldots\ldots\ldots + |511>) |2> +$$

$$(|4> + |10> + \ldots\ldots\ldots\ldots) |16> +$$

$$(|3> + |9> + \ldots\ldots\ldots\ldots\ldots) |8> +$$

$$(|2> + |8> + \ldots\ldots\ldots\ldots\ldots) |4> +$$

$$(|5> + |11> + \ldots\ldots\ldots\ldots) |11>] \ldots\ldots\ldots\ldots\ldots \quad (9.1)$$

As can be seen from the above expression Register 2 has an order of six and will be in a superpostion of the following six states:

$$(|1>, |2>, |4>, |8|, |11>, |16>)$$

When measured Register 2 will randomly collapse into one of the six states with the probability of collapse being equal in all cases. Since Registers 1 and 2 are entangled, measuring output register R2 will also lead to the collapse of input register R1 into an equal *superposition* of each state between 0 and 511($= q - 1$) that yielded the value c in the output register. Since the output register R2 collapsed to $|4>$, the input register R1 will be in an equal superposition of all 85 terms as shown in Eq. (9.1):

$$\frac{1}{\sqrt{85}} (|2> + |8> + |14> + \cdots\cdots\cdots + |506>) |4>$$

Notice that in the above expression the states have a periodic pattern; the period of this pattern can be determined by applying QFT register R1. The application of QFT to Register 1 results in

$$|\psi_1> = \frac{1}{512} \sum_{a=0}^{511} \sum_{c=0}^{511} e^{2\pi i a c/512} \, |4> |11^a (\text{mod } 21)>$$

The QFT peaks the probability amplitudes at multiples of q/r, where r is the period, which is 6 in this case

$$|1>, |2>, |4>, |8>, |11>, |16>$$

Since r (= 6) is even and $x^{r/2} \bmod N = 11^3 \bmod 21 \neq -1$, the factors for N (= 21) can be determined as shown below:

$$x^{r/2} \bmod N - 1 = (11^{6/2} \bmod 21) + 1 = 9$$

$$x^{r/2} \bmod N - 1 = (11^{6/2} \bmod 21) - 1 = 7$$

The two factors are

$$\text{GCD}\,(9, 21) = 3 \text{ and } \text{GCD}\,(7, 21) = 7$$

References

1. D. Deutsch, Quantum theory, the Church-Turing Principle and the universal quantum computer, *Proc. Royal Society*, London A, 400:97, 1985.
2. D. Deutsch and R. Jozsa, Rapid solution of problems by quantum computation, *Proc. Royal Society*, London A, 439:553, 1992.
3. Phillip Kaye, Raymond Laflamme, and Michele Mosca, *An Introduction to Quantum Computing*, Oxford University Press, 2007.
4. John Wright, Quantum Computation (CMU 15-859BB) Lecture Notes, Lecture 4: Grover's Algorithm, Carnegie Mellon University, Pittsburgh, Sep. 2015.
5. C. Lavor, L. R. U. Manssur, and R. Portugal, Grover's Algorithm: Quantum Database Search, arXiv:quant-ph/0301079v1, Cornell University Lebrary, Ithaca, New York, 2003.
6. E. Strubell, *An Introduction to Quantum Algorithms*, UMass-Amherst Tutorial, COS498, Spring 2011.
7. John Watrous, CPSC 519/619: Lecture 4, Quantum Teleportation; Deutsch's Algorithm, University of Calgary, January 26, 2006.
8. Mark Oskin, Quantum Computing—Lecture Notes, University of Washington, February, 2014.
9. John Watrous, CPSC 519/619: Lecture 5, A Simple Searching Algorithm; Deutsch-Jozsa Algorithm, University of Calgary, January 31, 2006.
10. C. Lavor, L. R. U. Manssur, and H. Portugal, Shor's algorithm for factoring large integers, arXiv:quant-ph/0303175, Cornell University Library, Ithaca, New York, 2005.
11. Elisa Baumer, Jan-Grimo Sobez, and Stefan Tessarini, Shor's Algorithm, qudev .phys.ethz.ch/content/QSIT15/, Switzerland, May 15, 2015.

CHAPTER 10

Quantum Cryptography

Secure transmission of messages and data is of utmost importance in both commercial and defense applications. This involves transmission of digital bit streams or digitized analog signals through various means from one location to another over a secure channel. A major weakness of such systems is the physical channel used by a system for interconnecting users and the system. An unauthorized user must not have access to the data transmitted via the channel.

Cryptography is used to make a message unintelligible to any person who is not authorized to receive it; this is achieved by combining the message with some additional information known as the *key*. The process of disguising a message to hide its content is called *encryption*. Many secure transmission methods require a type of *encryption*. A message before it is encrypted is known as a *plain text;* an encrypted message is also known as a *ciphertext*. The reverse process of converting a ciphertext back to a plaintext is known as *decryption* The two main components required to encrypt information are an algorithm and a key. The algorithm is generally known but the key is kept secret; the message cannot be extracted from the encrypted data without using the key.

For example, a message can be encrypted by using the following rule:

Replace every A in a message with a D, every B with an E, and so on through the alphabet.

Thus, using this rule the

Plain text: RETREAT

can be converted into

Ciphertext: UHWUHDW

Only someone who has the *shift-back-by-3* key can decipher this message; this is done by deriving the *mod* 26 of the new position of a shifted letter. Thus,

Ciphertext: DWWDFN

can be decrypted to

Plaintext: ATTACK

The primary goal of cryptography has been to provide the following four services for enhancing information security:

Confidentiality: Protection from disclosure to an unauthorized person, that is, keep the information from any person whose identity has not been verified.

Integrity: Identifying any alteration to the data. A receiver should be able to verify that a message has not been modified during its transition from the sender to the receiver; in other words, an intruder cannot substitute the original message with a false one without being detected. Thus, data integrity allows a means for the detection of any unauthorized manipulation of data although it cannot prevent it.

Authentication: The process of confident identification of the originator of a message by a recipient. In other words, it confirms that the message received by one party has been sent by another party whose identity has already been verified.

Non-repudiation: An originator of a message cannot falsely deny later that he or she sent the message.

In summary, a secure system must satisfy the following requirements:

1. Allow legitimate users have access when they need it.
2. Keep out unauthorized users.

10.1 Principles of Information Security

Information security can be achieved by using *symmetric key cryptography* or by *public key cryptography*. Claude Shannon of Bell Laboratories published the fundamental theory behind symmetric key cryptography in 1949. In this type of cryptography a single key is used to encrypt and decrypt the message. The main advantage of sharing a key is that a large amount of information can be communicated in secret by sharing a small number of key bits.

A major requirement in symmetric system is that a secure key establishment mechanism is in place. If a key is compromised, impersonators can decrypt messages and the security of the system can no longer be assured. A separate pair of keys can be employed for each pair of users to enhance security; this however increases the number of keys rapidly. For example, in a group of n people, the number of keys required will be $[n(n-1)]/2$. Secure key distribution remains the biggest challenge in using symmetric key cryptography.

Another class of security systems known as public key cryptographic system is very convenient to use and rely on a publicly known algorithms. The security of the Internet, for example, is partially based on such systems. A public key system uses separate keys for encryption and decryption. These keys are mathematically related; one key is used for encryption, the other one can decrypt the encrypted message and retrieve the original message back. For example, if one party A wants to communicate with another party B, then party A chooses a *private key* first. This key is not disclosed to anyone. A and B then exchange their public keys. To send a message to B, A first encrypts the message with the public key and then transmits it to B. B uses his/her private key to extract the corresponding plain text from the encrypted message.

The public-key system in a group of n people requires $2n$ keys, that is, n public and n private keys. The most important advantage of public-key systems over their private counterpart is that the need for a sender and a receiver to share secret keys via some secure channel is eliminated; all communications involve only public keys, a private key is never transmitted or shared.

In a private key cryptography the secrecy of the key is dependent only on the secrecy of the key. The key must be composed of sufficiently long string of randomly chosen bits private key cryptography suffers from a major weakness—as indicated earlier it requires sharing of a secret key between two parties. An intruder can copy the secret key as it is being exchanged, thereby severely compromising the security of the system. Thus, a private key cryptographic system depends entirely on secrecy of the key.

Public key cryptography does not have a key distribution problem. Its security relies on the fact that determining the factors of a number that is the product of two very large prime numbers is not computationally feasible. It has been shown that a quantum computer can derive the prime factors of very large numbers in polynomial time (see Shor's algorithm in Chap. 9). Public key cryptography will therefore become insecure if quantum computing becomes a reality. Quantum cryptography avoids all these issues by encrypting the shared key using a series of photons.

10.2 One-Time Pad

Vernam [1] proposed a scheme known as *one-time pad,* that encrypts data using a random key. The term "one-time pad" indicates that the key is used one time, and never used again. In topics of cryptographic communication the sender is identified as Alice, the receiver as Bob, and the intruder as Eve. The key must have the same number of bits as the data to be transmitted and must also consist of completely random bits that are kept secret from everyone except the sender and the receiver. The keys are used only once as indicated above; both the sender and the receiver must destroy their keys after use. The principle of operation of one-time pad is as follows:

Encryption by Alice:

$$c_i = d_i + k_i \quad i = 1, 2, 3 \ldots \ldots$$

where d_i are data bits; k_i are key bits; c_i are encrypted data bits.

Decryption by Bob:

$$d_i = c_i + k_i \quad i = 1, 2, 3 \ldots$$

Thus Alice encrypts the data she sends to Bob by EX-ORing it with randomly generated key bits. Bob retrieves the encrypted data by EX-ORing the received data with the same key bits. Figure 10.1 illustrates the scheme assuming key bits are 100010000101100.

A major drawback of one-time pads is that the sender and the receiver must somehow exchange the secret key bits that they use. For instance, in the example above (Fig. 10.1) Alice and Bob share the key bits k_1 before the transmission of the encrypted information bits. A third party might intercept the communication between the sender and the receiver and access the key, thereby comprising the security of the data transmission. Thus the secure distribution of keys is the prerequisite for secure communication; this is known as the *key distribution problem.*

Alice		Bob	
d_i	000010010110000	100000010011100	c_i
+			+
k_i	100010000101100	100010000101100	k_i
c_i	100000010011100	000010010110000	d_i

Figure 10.1 An example of one-time pad.

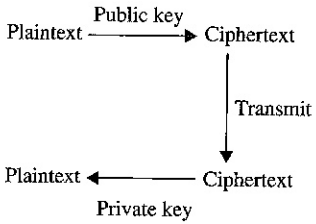


Figure 10. 2 Public key cryptographic system.

10.3 Public Key Cryptography

As discussed earlier in this chapter, the communicating parties in public key cryptographic system uses two separate keys—a public key and a private key. The public key as the name suggests may be made accessible to anybody. The private key on the other hand is kept secret. Figure 10.2 shows the encryption and decryption process in a public key cryptographic system.

Public key cryptography uses a method of encoding and decoding that employs a special case of the one-way function known as a *trapdoor or one-way function* [2]. A function $f(x)$ is considered to be a one-way function if it is easy to compute the function $f(x) = y$ for any input x, but the opposite, that is, computing x from $f(x)$, is significantly more complicated unless some piece of information (the trapdoor) is known. For example, it is relatively easy to multiply two prime numbers to generate a composite number, but is extremely difficult to factor a composite number (especially a very large integer) into a product of two prime numbers unless one of the numbers is known. It is intuitively clear that calculating 67×83 is much faster than finding the prime factors of 5761. However, the problem can be easily solved, if some additional information is given, for instance, knowing that 67 is one of the prime factors of 5761.

An important point to remember is that unlike in symmetric encryption, the two keys in public cryptography behave differently; the public key is the *only* key that can encrypt the data to be sent out. A public key is freely available to anyone who wants to use it. They can be distributed as email attachments or through a public key chain server that stores a large number of public keys. Although any one can have access to the public key, the encrypted data can only be decrypted by a party who knows the corresponding private key. The distribution of the private key is avoided, thus preventing any unauthorized party from accessing the key. Moreover even if the public key falls into hands of unauthorized persons, it is practically impossible to derive the decryption key from the encryption key because this is computationally infeasible.

The process of using public key cryptography is relatively straightforward. To send a message, the sender (Alice) obtains a copy of Bob's (recipient) public key, either by email or from a key chain

server that stores a large number of public keys. The resulting encrypted message is then sent to Bob who uses their shared private key to restore the original message.

The advantage of public key cryptography is that it does not require any initial secure exchange of secret keys for encrypting a message. However, it requires far longer keys to offer the same level of protection as symmetric encryption. A newer type of public key cryptography, known as *elliptic curve cryptography*, can be just as secure as symmetric encryption using similar key lengths [3].

10.4 RSA Coding Scheme

The widely used RSA (Rivest, Shamir, and Adelman) technique is a public key cryptographic system [4]. It facilitates the generation of public and private keys by choosing two large prime numbers p and q, and making $N = p.q$. Next a random positive integer e is chosen such that it is relatively prime to $(p - 1)$ $(q - 1)$; e is called the encryption constant. Then the decryption constant d is derived such that $e.d = 1 \ mod \ (p - 1)(q - 1)$. The public key is (N, e), and the private key is d.

It should be mentioned that although N is revealed to all, the factors p and q of N are kept secret. Obviously if an intruding party can factor N to find p and q, then it can use e of the public key to derive the private key d from the expression $e.d = 1 \ mod \ (p - 1)(q - 1)$. The steps of the RSA algorithm are as follows:

i. Generate two large prime numbers, p and q, and let $n = p.q$.

ii. Let $\varphi = (p - 1)(q - 1)$.

iii. Choose another number e which is relatively prime to φ; two numbers a and b that have no common factors other than 1 are said to be *co-prime* or *relatively prime*.

iv. Select e, $1 < e < \varphi$ such that $gcd(e, \varphi) = 1$; *gcd* (the greatest common divisor) of two integers a and b is the largest integer that divides both numbers.

v. Find d, such that $de \equiv 1 \bmod \varphi$. The notation "$a \equiv b \ (\text{mod } n)$" means a is *congruent* to b, that is, a and b have the same remainder when divided by n.

vi. Encryption: compute $c = m^e \bmod n$, where m is message block represented as a number $0 < m < n - 1$ and c is the encrypted message.

vii. Decryption: compute: $m = c^d \bmod n$.

The security of RSA system is based on the fact that currently no algorithm is available for factoring a large number into a product of two

prime numbers in a reasonable amount of known for some time that Shor's quantum algorithm (discussed in Chap. 9) is capable of factoring very large numbers efficiently. Thus, the security of public key cryptographic system can be guaranteed only till quantum computers become technologically feasible.

Public key cryptography in general requires far longer keys to offer the same level of protection as symmetric encryption. Elliptic curve cryptography, on the other hand, can be just as secure as symmetric encryption using similar key lengths.

10.5 Quantum Cryptography

According to the quantum theory, light is an electromagnetic wave and is made up from a lot of particles known as photons, each photon has a specific energy hf and a wavelength c/f. Recall that light has a pair of electric and magnetic fields that are perpendicular to each other as shown in Fig. 10.3. If the electric field component of a beam of light vibrates along a single direction (like the vertical direction Fig. 10.3) then the beam of light is said to be *polarized* in that direction [5].

Polarized photons can be created by passing a normal beam of light through a filter set for a specific angle of polarization. A photon incident on the filter will either pass through it or will be blocked; if the photon emerges it will be aligned to the angle of the filter regardless of its initial polarization angle of the filter regardless of its initial polarization.

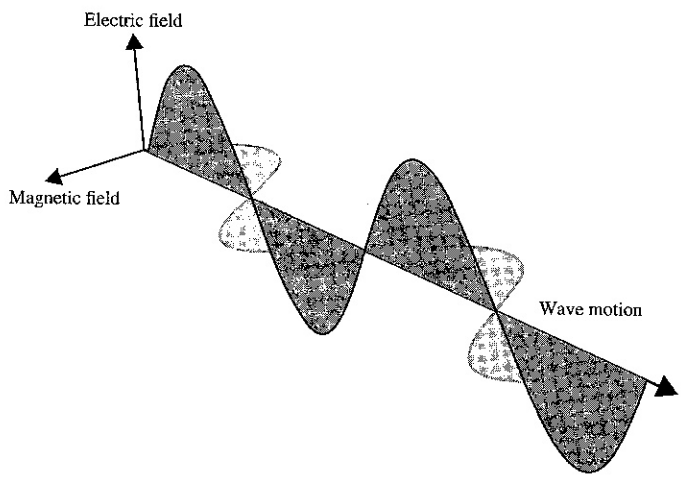


Figure 10.3 Propagation of electromagnetic waves (Ref.5).

Figure 10.4 Polarized modes of a photon.

Photons can be polarized in one of two bases: *rectilinear* (⊕) or *diagonal* (⊗), using an appropriate filter as shown in Fig. 10.4. A filter allows the transmission of a photon through it only if the polarization of the photon is aligned with the filter. In the rectilinear mode only photons with horizontal or vertical polarization pass through the polarizing filter. In the diagonal mode, on the other hand, only photons with polarization that are at an angle of +45° or −45°, to the horizontal axis can pass through the polarizing filter. Thus in the rectilinear mode, orientations | and — represent 0° and 90°, respectively, and in the diagonal mode orientations \ and / represent +45° and −45°, respectively.

Heisenberg's uncertainty principle shows that certain pairs of properties, known as *noncommutating* properties, are related in a way that it is impossible to measure these simultaneously; such pairs are called *conjugate pairs*. Rectilinear and diagonal polarizations constitute such a conjugate pair of noncommuting properties. Thus, a filter with 0° /90° orientation can correctly detect a rectilinearly polarized photon; a filter similarly with +45°/−45° orientation can detect a diagonally polarized photon. On the other hand, if a diagonally polarized filter is used to detect a rectilinearly polarized photon or vice versa, the outcome will be random with equal probabilities and the photon will lose all the information of its previous state.

10.6 Quantum Key Distribution

Key distribution as discussed earlier enables the sharing of cryptographic keys such as a private key between two or more parties so that they can securely share information such as a private key; the key can then be used to encrypt messages that are being communicated over an insecure channel. As indicated previously, the distribution of keys is a major weakness of private key cryptography.

Quantum cryptography overcomes this drawback by providing a secure way of sharing a random key between two separate parties. An additional advantage of quantum keys is that the sender and receiver can easily verify whether the key has been tampered with. It should be emphasized here that QKD is not a technique for encryption and decryption of data; it allows only secure distribution of private keys.

As the name implies, quantum cryptography is a particular form of cryptography that relies on the laws of quantum mechanics in

order to ensure unconditional security. It has its origin in a novel idea of Stephen Wiesner, a graduate student at Columbia University in 1969 [6]:

i. The polarization of photons cannot be simultaneously measured in incompatible bases (rectilinear/diagonal).

ii. Information of an individual property of a quantum particle, for example, the polarization of a single photon cannot be obtained.

iii. It is not possible for an intruder to access a message between Alice and Bob without changing its meaning.

iv. It is not possible to copy an unknown quantum state.

Once a key has been successfully transmitted, it can be used to encrypt a message in a classical symmetric cipher, for example, the one-time pad and can be transmitted by conventional means such as telephone or email. Thus, symmetric keys in conjunction with quantum key distribution can guarantee secure generation and transmission of private keys. More importantly, quantum key distribution is secure against new attacking strategies and against determined eavesdropping.

10.7 BB84

Bennet and Basard, inspired by Wiesner's scheme, proposed a quantum key distribution (QKD) protocol [7]. This protocol, known as BB84, allows a sender (Alice) to send photons to a receiver (Bob). Alice and Bob communicate via a one-way quantum channel and a two-way public channel. Alice has a source of single photons and two polarizing filters—one rectilinear and one diagonal. Figure 10.5 illustrates a quantum key distribution system:

The quantum channel, however, is vulnerable to any possible manipulation from an eavesdropper. Alice and Bob must ensure that

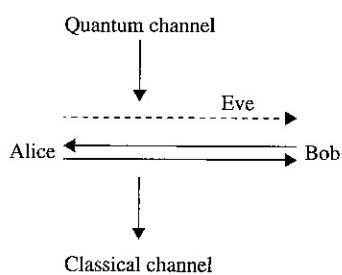


Figure 10.5 A quantum key distribution system.

the eavesdropper, usually called Eve, cannot tap on the quantum channel and listen to the exchanges over the classical channel.

A photon has three types of spins: *horizontal, vertical,* and *diagonal.* An unpolarized photon has all three spin states at the same time. By passing it through a polarization filter, a photon can be polarized to let through only a particular spin. All the unwanted types are eliminated. Furthermore, single photons cannot be copied; this is because the linearity of quantum mechanics does not allow cloning of unknown quantum states as explained (see Chap. 7).

The BB84 protocol can be implemented using polarized single photons. Alice can transmit single photons randomly in either a rectilinear (⊕) or diagonal (⊗) basis. In each basis, one orientation of the photon is used to represent the logic value 0 and the other one to represent 1; this is agreed upon by Alice and Bob before the key formation begins.

Note that in practical quantum cryptography, a qubit is represented by the polarization state of a photon. The polarization state of a randomly polarized photon is a superposition of any pair of orthogonal states such as:

- Horizontal (H) and vertical (V) polarization
- +45° and −45° diagonal polarization

The polarization of a photon can thus be modeled by a qubit; the state of the qubit is represented as

$$|\psi> = a|H> + b|V> = c|+45°> + d|-45°>$$

where a, b, c, and d are complex numbers and $|a|^2 + |b|^2 = |c|^2 + |d|^2$.

For a chosen orthogonal state pair, one polarized state is assumed to be $|0>$ and the other one as $|1>$. For example, for a rectilinear state pair:

$$|V> = 1 \quad \text{and} \quad |H> = 0$$

for a diagonal state pair

$$|+45°> = 1 \quad \text{and} \quad |-45°> = 0$$

Thus, a photon can be assumed to be a qubit with one bit of quantum information.

Figure 10.6 shows the basis, angle, polarization, and logic value of single photons. For example in the rectilinear mode, orientations | and — represent 1 and 0, respectively. In the diagonal (⊗) mode, orientations \ and / represent 1 and 0, respectively.

Basis	Angle	Polarization	Logic value
⊕	0°	—	0
⊕	90°	|	1
⊗	45°	/	0
⊗	−45°	\	1

FIGURE 10.6 Characteristics of single photons.

To perform BB84, Alice and Bob have to first agree on how bits will be encoded in the polarization directions for each filter. This means they should form a bit table like the one shown above in Fig. 10.6.

The following steps describe the BB84 protocol:

i. Alice generates a random sequence of 0s and 1s. She then replaces each bit in the binary sequence with a randomly chosen polarization shown in Fig. 10.6. Theoretically any quantum particle can be used to replace the bits. The photon is preferred, however, because they can be transmitted over longer distances without decoherence.

ii. Alice sends the photons corresponding to each bit replacement in the binary sequence to Bob via the quantum channel, while keeping record of the polarization basis and the logic value of the transmitted photons.

iii. As Bob is not aware of which basis Alice has chosen for a photon, he randomly chooses one of the two bases. If he chooses the same basis as Alice, the polarization is recorded correctly. Alternatively, if he chooses a different basis, the initial polarization of the received photon is lost and the result is a random polarization. It is also possible that sometimes Bob does not register anything because of errors in the detection or in the transmission.

iv. Once Bob receives all the photons sent by Alice, he confirms that he has received and measured all of them. The bit string corresponding to the photons received by Bob is called a *raw key*.

v. Next, Bob announces via the public channel his choice of basis for each photon. This does not lead to any security compromise since Bob reveals only which bases he used, not which result he obtained. Thus, an eavesdropper cannot get any information related to the key formation.

vi. Alice and Bob then compare the bases they selected and discard all non-matching bases. In other words, Alice and Bob keep only the bits corresponding to the same bases. Since both Alice and Bob have randomly chosen the bases, there is an equal probability of getting matched and unmatched results. As a consequence, almost 50% of the qubits are available for forming the secret key. Note that the key is truly random because neither Alice nor Bob can decide which key will result at the end of the procedure.

To illustrate, assume that Alice decides to send the following bits to Bob:

Bits	1	0	0	1	1	0	1	0

and chooses the following bases to convert the bits:

Basis	+	×	+	×	×	×	+	+

The polarization of the resulting single photons are:

Polarization	|	/	—	\	\	/	|	—

Bob detects the state of each photon he receives by randomly picking one of the bases of photons. If he makes the correct guess in picking the base Alice used for sending a particular photon, he obviously detects the correct orientation of the photon and the correct logic value the photon represents. For example, if Alice sends a 1 using the rectilinear mode (as in the first bit in the above bit sequence) and Bob chooses the same polarization mode, he is guaranteed to receive a 1. On the other hand, if Bob picks the diagonal base, the probability of his receiving a 1 is reduced to 50% and there is a 50% probability of his receiving a 0 instead. The modes Bob selected and the polarizations of the resulting photons are:

Mode	+	+	×	×	+	×	+	×
Polarization	|	—	/	\	|	/	|	/

After all the bits have been sent, Alice informs Bob of the bases she used to send each photon but not its orientation via the public channel. Bob also communicates to Alice via the public channel the bases he used. If Bob used a base that is different from Alice's photon, he ignores the corresponding bit. Alice and Bob keep only those bits for which their bases match perfectly; the remaining bits are discarded. Statistically, only 50% of the transmitted bits agree. These bits

are used as the key. This shorter key is called a *sifted key*. The following example illustrates the BB84 key generation protocol:

Alice random Bits	1	1	1	1	0	1	0	1	0	1	1	0
Alice base	⊗	⊗	⊕	⊗	⊕	⊗	⊕	⊗	⊕	⊕	⊕	⊗
Alice polarization	\	\	|	\	—	\	—	\	—	|	|	/
Bob base	⊗	⊕	⊗	⊗	⊗	⊕	⊕	⊗	⊗	⊗	⊕	⊕
Bob's polarization	\	|	\	\	/	|	—	\	/	\	|	—
	↑			↑			↑	↑			↑	
Secret key	1			1			0	1			1	

Note that in this case, the sifted key has less than 50% of the original key bits. Thus, BB84 is inherently inefficient in its use because many key bits, as high as 50%, are discarded during the key formation process as shown in the above example.

A simplified version of the BB84 protocol was proposed in Ref. [8]. This version uses two states, rectilinear and diagonal, for representing 0 and 1 respectively instead of four states in BB84. Pasquinucci et al. [9] proposed a protocol that uses three orthogonal bases and six states to encode the key bits. Thus an intruder has to correctly choose the base used by the sender and receiver out of three possible bases. This increases the probability of the intruder making more errors in selecting the correct base, thus allowing easier intrusion detection.

Scarani et al. [10] proposed another variation of the BB84 protocol known as SARG04 in 2004. The first phase of the protocol is the same as the first phase of BB84. In the second phase when Alice and Bob determine for which bits their bases matched, Alice announces a pair of non-orthogonal bases. Instead of the exact bases that she used to encode her bit, one of the bases in this pair is the base she used to encode the key data bit. At the receiving end, Bob will correctly measure the polarization state if he chooses the same basis as Alice. Otherwise, the data will have an unpredictable value. If there are no errors, then the length of the key remaining after the sifting stage is ¼ of the raw key.

The SARG04 protocol provides almost the same security as BB84. However, SARG04 provides better protection against the PNS (photon number splitting) vulnerability of the BB84 protocol. It arises because Eve can take away a photon and can obtain all the information from it after the public key sifting stage [11].

10.8 Ekart 91

It is a three-state protocol that uses the Einstein-Podolsky-Rosen (EPR) paradox discussed in section 6.4. This protocol can be described in terms of the three polarization states of an EPR photon pair [12]. For example, three possible states are:

$$|\varphi_0\rangle = \frac{1}{\sqrt{2}}(|0\rangle_1 |90\rangle_2 - |90\rangle_1 |0\rangle_2)$$

$$|\varphi_1\rangle = \frac{1}{\sqrt{2}}(|30\rangle_1 |120\rangle_2 - |120\rangle_1 |30\rangle_2)$$

$$|\varphi_2\rangle = \frac{1}{\sqrt{2}}(|60\rangle_1 |150\rangle_2 - |150\rangle_1 |60\rangle_2)$$

The symbols for bits 0 and 1 in these states are

$$|\varphi_0\rangle \; |0\rangle = \text{Bit } 0$$

$$|90\rangle = \text{Bit } 1$$

$$|\varphi_1\rangle \; |30\rangle = \text{Bit } 0$$

$$|120\rangle = \text{Bit } 1$$

$$|\varphi_2\rangle \; |60\rangle = \text{Bit } 0$$

$$|150\rangle = \text{Bit } 1$$

As in BB84, there are two stages of communication between Alice and Bob; one over a quantum channel and the other over a public channel.

An EPR pair of photons is first created in randomly selected states from the set $\{|\varphi_0\rangle, |\varphi_1\rangle, |\varphi_3\rangle\}$. One photon of the EPR pair is sent to Alice and the other to Bob using the quantum channel. For each photon they receive, Alice and Bob select randomly and independently an operator from the set of three that were chosen for measuring the photons. They measure their respective photons with the selected operator. Alice records her measured bit while Bob records the complement of his measured bit. This process is repeated for all the needed photons.

In the first part of the second stage, they release via the public channel which basis they used for each bit slot measurement. They separate the measurement results into two groups:

1. The first group consists of bit slots for which the same measurement operators are used.

2. The second group consists of bit slots for which different operators are used.

Any photons which are not measured in the first or the second group are discarded. The first group is used to establish a *raw key*, while the second group that includes all the remaining bit slots is called a *rejected key*.

The EPR protocol, unlike BB84, does not discard the rejected key bits. Instead, they are used to test the presence of an intruder. During the second phase of the second stage, Alice and Bob publicly announce the results obtained for those cases in which they used *different* operators. Assuming Alice and Bob picked the measurement operators randomly and independently, the correlation between their results is found to be the same as *CHSH* (Clauser, Horn, Shimony, and Holt) inequality and is equal to $-2\sqrt{2}$ [13]. A major variation of this value would indicate the presence of an eavesdropper. On the other hand, if the CHSH inequality is not violated, Alice and Bob can trust that the perfectly anti-correlated results that they obtained can then be converted into a secret key.

For each measurement where Alice and Bob used the *same* basis, they should expect opposite results due to the principle of quantum entanglement. This means that if Alice and Bob both interpret their measurements as bits (as before), they each have a bit string which is the binary complement of the other. Either party could invert their key and share a secret key. They can then use this set of common private keys to encrypt and decrypt their messages and communicate secretly. However, each key can only be used once and cannot be repeated in order to keep the keys completely random.

The eavesdropper Eve cannot get any information from the photons while they are in transit because the information is *formed* in a photon only after it is measured and the result is communicated to legitimate users.

If Eve tries to detect photons coming from the source, she has to randomly choose her own measurement base since she does not know what base Bob will use. Therefore, about half the time she will choose a different basis than Bob. Suppose Alice sends one type of polarized photons, some of which are received by Eve and Bob. If Eve decides to use a different basis for measurement and Bob decides to use the same basis as Alice, Alice and Bob keep the resulting data since they both used the same basis. But since Eve used the wrong basis, she does not know what their result was.

Suppose instead that Eve decided to measure the polarized photons according to same basis Alice used, but Bob decides to measure these using a different basis. Here Eve would know the polarization that Alice sent, but since Bob did not choose the correct basis, Alice and Bob would throw the results out.

References

1. G. S. Vernam, "Cipher Printing Telegraph Systems for secret wire and radio telegraphic communications," *J. AIEE* 45, pp. 109–115, 1926.
2. W. Diffie and M. Hellman, "New directions in cryptograph," *IEEE Trans. Information Theory*, Vol. 22, No. 6, 1976.
3. I. F. Blake, G. Seroussi and, N. P. Smart, "Elliptic curves in cryptography," London Mathematical Society, Lecture Note Series, 265. Cambridge University Press, Cambridge, 2000.
4. R. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public key cryptosystems," *Communications of the ACM*, 21, pp. 120–126, 1978.
5. "The BB84 protocol for quantum key distribution," Quantum Gazette: Exploration into quantum physics and science writing Sept. 22, 2016.
6. Nikolena Ilic, "The Ekert protocol," *J. Phy.* Vol. 334, No. 1, 2007.
7. C. H. Bennett and G. Bassard, "Quantum cryptography: Public key distribution and coin tossing," *International Conference on Computers, Systems & Signal Processing*, pp. 175–179, 1984.
8. C. H. Bennett, "Quantum cryptography using any two non-orthogonal states," *Phys. Rev. Lett.*, Vol. 68, pp. 3121–3124, 1992.
9. H. Bechmann-Pasquinucc and N. Gisin, Incoherent and coherent eavesdropping in the six state protocol of quantum computing," *Phys. Rev.*, Vol. A59, pp. 4238–4248, 1999.
10. A. Scarani, A. Acin, G. Ribordy, and N. Gisin, "Quantum cryptography protocols robust against photon number splitting attacks," *Phys. Rev. Lett.*, Vol. 92, No. 5, pp. 057901, 2004.
11. Sheila Cobourne, "Quantum Key Distribution Protocols and Applications," Technical Report, RHUL-MA-2011-05, 2011.
12. A. K. Ekert, "Quantum cryptography based on Bell's Theorem," *Phys. Rev. Lett.*, Vol. 67, p. 661, 1991.
13. J. F. Clauser, M. A. Horne, A. Shimony, and R. A. Holt, "Proposed experiment to test local hidden-variable theories," *Phys. Rev. Lett.*, 23 (15):880–884, 1969.

Index

Note: Page numbers followed by *f* denote figures.

N

O

P

Q

R

S

ELECTRICAL ENGINEERING

A self-contained, reader-friendly introduction to the principles and applications of quantum computing

Especially valuable to those without a prior knowledge of quantum mechanics, this electrical engineering text presents the concepts and workings of quantum computing systems in a clear, straightforward, and practical manner. The book is written in a style that helps readers who are not familiar with non-classical information processing more easily grasp the essential concepts; only prior exposure to classical physics, basic digital design, and introductory linear algebra is assumed.

Quantum Computing: A Beginner's Introduction presents each topic in a tutorial style with examples, illustrations, and diagrams to clarify the material. Written by an experienced electrical engineering educator and author, this is a self-contained resource, with all the necessary prerequisite material included within the text.

Coverage includes:

- Complex Numbers, Vector Space, and Dirac Notation
- Basics of Quantum Mechanics
- Matrices and Operators
- Boolean Algebra, Logic Gates, and Quantum Information Processing
- Quantum Gates and Circuit
- Tensor Products, Superposition, and Quantum Entanglement
- Teleportation and Superdense Coding
- Quantum Error Correction
- Quantum Algorithms
- Quantum Cryptography

ISBN 978-1-260-12311-1
MHID 1-260-12311-1
90000
9 781260 123111

Learn more. Do more.™
MHPROFESSIONAL.COM
Follow us on Twitter @MHEngineering
ALSO AVAILABLE AS AN EBOOK