Sudan University of Science and Technology

College of Graduate Studies

Quantum Error Correction Using Shor's Bit-Flip Code

تصحيح الاخطاء الكَمية بإستخدام طريقة شور

A dissertation submitted in partial fulfillment for the requirements of a master degree (M. Sc.) in physics

By: Sufuan Altegane Ali Hagana

Supervised by: Dr. Ammar Ibrahim Abdalgabar

April, 2017

Dedication

I dedicate this dissertation to my

Mother and Father

Brothers and Sisters

Acknowledgments

First of all warm thanks to the many people who have contributed to this Dissertation, especially Dr. Ammar Ibrahim Abdalgabar my direct M.S.c supervisor, without his help it would not have been possible to working on this research, and also Dr. Mohammed S. H. Suleiman who has been a terrific mentor whose boundless encouragement has supported me to keep on the track I am truly grateful, thank you for supporting me along the way. My deepest thanks goes to Dr. Badr Mohammed Awad Elseid, assistant professor of physics at California University, I would like to acknowledge him because he denoted me his time and effort mean while he was so busy, and for his patient during the course of this dissertation, the period I spent working with him have been so benefit-ion, I am extremely grateful to him. It is a great pleasure to thank the staff of physics department at Sudan University of Science and Technology, especially Dr. Magdi E. Y. Suliman and Dr. M.H Eisa who used to be my B.S.c supervisor, for their support and encouragement. A big thanks needs to go out to my family for all their love and support, and also to my friends and colleagues who were always there for me in particular Musa Alnoor Musa, and Mohammed I. Naile, thank you all so much!. I would like to thanks Edx online education platform, four years ago I had the opportunity to take a course in physics, about Quantum mechanics and quantum computing, by Umesh V.Vazirani professor of physics at University of California, Berkeley. Throughout the course I managed to learn so much about quantum computing, this course provided so much inspiration to me to take the path of quantum computing and quantum information. So thank you prof. V.Vazirani and the staff who are been working in that course. My thanks also to those creative people (Richard Feynman, Peter Shor, Richard Jozsa, Phillip Kaye Raymond, David Deutsch, Charles Bennett, Gilles Brassard, Rolf Landauer, Stephen Wiesner, T. Venkat Narayana Rao, Paul Benioff, and Michael Aaron Nielsen) without their contribution the physics of information would not exist as a field of research.

Abstract

This dissertation is slightly an introductory review to understand quantum computing and methods of quantum error correction, in this dissertation we focused on interpreting the theoretical part of quantum noise and errors such as decoherence. We found that decoherce is a type of quantum error among many other types, resulting from the qubits interaction with the environment and thus leads to errors in the qubits that carry the quantum information, for example bit-flip and phase-shift errors. In particular we attempted to correct the bit-flip quantum error theoretically by using quantum error correction code, through applying a tool known as shor's bit-flip code into the decohered state, and see if it is possible to correct this error and recover the lost quantum information encoded in the qubits. And we found that this method for quantum error correction is successful and can be developed to be more efficiently.

ملخص البحث

البحث عبارة عن مقدمة في الحوسبة الكمية Quantum Computing و تصحيح الاخطاء الكمية Quantum Error Correction التي تتعرض لها المعلومات الكمية عند نقلها او معالجتها او حتي عند تخذينها, هنالك العديد من الاخطاء الكمية quantum errors التي تحدث للمعلومات الكمية الممثلة في صورة Qubits نتيجة لتاثرها بعوامل خارجية مثل (المجالات الكهربائية و المغناطيسية و عوامل اخري), مما يؤادي الي تشوه Decoherence هذا المعلومات و بالتالي فقدها و عدم القدرة علي استرجاعها. احد هذه الاخطاء الكمية يعرف بـ Bit-flip quantum error ويحدث لـ qubits او qubit حيث يعمل علي تغير حالتها الكمية علي سبيل المثال من 1 الي 0 و هذا يعني تغير حالة النظام المعين الممثل بوسطة qubits وبالتالي تشوه المعلومة الكمية. هنالك طرق عديدة لمنع هذه الاخطاء من الحدوث منها العزل الكامل و الكلي Qubits Isolation لمنع التفاعل مع المؤثرات الخارجية, و كما ان هنالك طرق لمعالجة وتصحيح مثل هذه عند حدوثها مثل طريقة شور لتصحيح الاخطاء الكمية. في هذه الرسالة تم اخذ Bit-flip quantum error كمثال ومحاولة تصحيح هذا الخطأ و استرجاع المعلومات المفقودة بإستخدام طريقة شور لتصحيح الاخطاء الكمية و وجد انها جيدة ومجدية اذا تم تطبيقها بصورة مثالية وبالتالي يمكن تطويرها لتصحيح انواع اخري من الاخطاء الكمية بكفأة عالية.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURES

LIST OF TABLES

TABLES

CHAPTER I

INTRODUCTION

In this chapter we will give a brief introduction about quantum computer, current models of quantum computing, quantum noise and errors, then we shall formulate the research with problem and finally we give an outline of the dissertation.

1.1 Quantum Computer

Quantum computer (QC) is a device that makes use of quantum mechanical principles, to carry out a computation operations on data, some of the fundamental concepts that are used in quantum computing are super-positioning, and entanglement. Classical computers are based on transistors, it has a memory that is made up in the form of series of bits, where each bit takes a value either {0 or1} where as in a quantum computer, the information is represented in the form of quantum bits {Qubits} (T. Venkat, Shirish, 2010), a qubit is a unit vector in a 2- dimensional complex vector space with fixed orthonormal basis $|0\rangle$ and $|1\rangle$ which are relative to $|\downarrow\rangle$ and $|\uparrow\rangle$, and represent the classical bit values zero and one respectively. Qubits can be in a superposition of $|0\rangle$ and $|1\rangle$ such as $\alpha|0\rangle + \beta|1\rangle$. Thus, $|\alpha|^2$ is the probability of measured value to be in the state $|0\rangle$ and $|\beta|^2$ is the probability to be in state $|1\rangle$ respectively (Javier, 2009). A qubit is a form of quantum object, such as, an atom (an ion) which can occupy different quantum states. Two of these states are used to store digital information, for instance; atom in the ground state

corresponds to the value $\langle 0 \rangle$ of the qubit, the same atom in the excited state, corresponds to the value $\langle 1 \rangle$ of this qubit (Gennady, et al., 1999).

A qubit in general may have two spins, up and down, a single qubit can take the value of a 1 or 0, and may both of them simultaneously, similarly two qubits, three qubits can be in any quantum superposition of four states, eight states respectively, therefore, in general an $n-$qubits in quantum computer can have up to 2^n different states, compare to a classical computer that can only be in one of these 2^n states at any one time. A quantum computer operating by manipulates the qubits using a fixed sequence of series quantum logic gates, these gates can be applied by using quantum algorithm. In a quantum computer, computations are finished by using quantum properties to symbolize data and perform operations on these data. A theoretical model of a quantum computer is the quantum Turing machine, (the universal quantum computer) (T. Venkat, Shirish, 2010). The power of quantum computer is not connected with the density of qubits, but on its abilities which allows one to operate quantum states in superposition, one atom can be used to generate an infinite number of super positional states using just two basic quantum states, which correspond to $\langle 0 \rangle$ and $\langle 1 \rangle$, for instance; if two states have the energies, E_0 and E_1, one can prepare a superposition of states, 0 and 1, which corresponds to any average value of energy between the values E_0 and E_1. However, when measuring the energy of a single atom, we get only one of two results, E_0 and E_1. Utilization of super-positional states allows one to work with quantum states which simultaneously represent many different numbers; this is known as "Quantum parallelism".

Any quantum computer must satisfy certain requirements, which known as the DiVincenzo criteria, these requirements as following:

- A scalable physical system with well characterized qubits: A quantum computer must be made up of many quantum bits, which both exhibit quantum properties (superposition, entanglement).

- The ability to initialize the state of the qubits to a simple fiducially state such as the ground state: We must be able to initialize the computer in some state $|\psi\rangle$ $|000..\rangle$ for two reasons: First, any algorithm would require the computational register to be in some known state to begin a computation, second it is necessary to perform error correction on any quantum computer, which requires a steady stream of qubits in some pure state to extract entropy.
- Long relevant decoherence times much longer than the gate operation time: to successfully run an algorithm, a quantum computer must accurately store the information, any corruption of information can be understood as coupling to the environment, noise in control signals, and so on.
- A qubit-specific measurement capability: some sort of measurement mechanism is required, a measurement of some group property of qubits, like their projection onto a specific basis state, is insufficient, thus individual qubit measurement is crucial for extracting error syndromes to perform quantum error correction .
- A universal set of quantum gates: To run a quantum algorithm which is a set of unitary instructions that involve some number of qubits, one applies a Hamiltonian H_1 for some time on that algorithm followed by H_2, and so on. Experimentally it is very challenging to implement a series of arbitrary Hamiltonians, thus break them down into some set of constituent parts; as a result many quantum operations can be comprised from the others. For example, a controlled-phase gate can be turned into a Control_Not gate with the addition of Hadamard gates on the target qubit before and after. Thus, there are many possible sets of "universal" gates (Ronald, 2016).

1.2 Quantum Computation

1.2.1 What is Quantum Computation

Quantum computation is a field that investigates the computational powers of quantum computers and their properties by using principles of quantum mechanics. It aims on finding a quantum algorithms that are significantly faster than any classical algorithm which can give a perfect solution to the same problem (Ronald, 2016). Quantum computing makes use of the effect constructive interference in the "desired" direction of computation and destructive interference in all others, an example for this process is reflection of a light beam from a mirror surface (reflected light is a photons in a superposition moving in many different directions), only one direction is selected by nature which is corresponds to the law of reflection. The measurement outcome is represented in a discreet digital value, for instance if there is a voltage (represented by - 1), and there is no voltage (represented by-0) (Gennady, et al., 1999). Quantum computing relies on several phenomena and laws of the quantum world that are fundamentally different from those one encounters in classical computing, it offers radically a new possibilities and lead to different constraints than classical computations, which is based on the laws of classical physics, moreover, quantum computing seems to have the potential to deep our understanding of nature as well as to provide more reliable methods for information processing and communication tools (Jozef, 2011).

1.2.2 Present Models of Quantum Computing

Topological Quantum Computer: It is a theoretical model of quantum computer that utilizes two dimensional quasi-particles called (anyons), anyons world lines cross over one another to form braids in a 3-dimensional space-time (i.e., one temporal plus two spatial dimensions), these braids form the logic gates in the quantum computer, quantum computer based on quantum braids much more stable than that using trapped quantum particles, the smallest disturbance can cause a quantum particle to decohere and bring in errors in the computation, such small perturbations do not change the topological properties of the braids.

Adiabatic Quantum Computation (AQC): build on the adiabatic theorem to do calculations, a system with a simple Hamiltonian is prepared and initialized to the ground state, this Hamiltonian is then developed to the complex Hamiltonian in the ground state by using adiabatic quantum computation, system ground state describes the key to the problem of interest. By the adiabatic theorem, since the energy of the outside world is kept lower than the energy gap between the ground and the next higher energy state of the system, the system will has a lower probability of going to a higher energy state when interference with the outside world, similarly it cannot move to a lower state thus it will remain in the ground state, so at the end the state of the system can stay coherent as long as needed, that means AQC is a possible method to get around the problem of quantum decoherence. But experimentally it is a bit difficult to perform this, as the Hamiltonian is gradually change, a multiple qubits are close to a tipping point during computation, it is exactly at this point, the ground state gets arbitrarily close to a first energy state, by adding a slight amount of energy (external or slowly changing the Hamiltonian) the system will be taken out of the ground state, and this instantly damage the calculation. If we try to perform the calculation more quickly the external energy will increase a result; scaling the number of qubits makes the energy gap at the tipping points smaller.

Loss-DiVincenzo Quantum Computer: Also known as spin-qubit quantum computer is a scalable semiconductor-based quantum computer; the proposed computer was to use as qubits the intrinsic spin-1/2 degree of freedom of individual electrons restricted to quantum dots. This was done in a way that fulfilled DiVincenzo Criteria for a scalable quantum computer, namely:

- Identification of well-defined qubits.
- Consistent state preparation.
- Low decoherence.
- Precise quantum gate operations.
- Strong quantum measurements (T. Venkat, Shirish, 2010).

1.2.3 Advantages of Quantum Computing Over Present Computing Systems

There are several reasons to develop a practical quantum computer:

1. High Speed and Huge Security: atoms change energy states very quickly than fastest computer processors. Each qubit can take the place of an entire processor; a 1,000-processor computer could be replaced by 1,000 ions of say, carbon the main idea is finding the sort of problem a quantum computer is capable to solve. It is also extremely useful for decoding and encoding secret.
2. Large Data Storage and information Access: a quantum computer capable to store a large amount of data within a very minute scale. And also it could be used to search large databases in a fraction of the time that it would take a conventional computer.
3. High Efficiency and Competence: By adding up all the above qualities, it directly increases the efficiency and accuracy of a quantum computer to the next level.

4. Quantum Communication System: it allows a sender and receiver to agree on a code without ever meeting in person. If an eavesdropper tries to monitor the signal in transit the uncertainty principle ensure that it will be disturbed in such a way that the sender and receiver are alerted.
5. Quantum Cryptography Challenge: quantum computers expected to improve the world of cryptography.
6. Artificial Intelligence: according to theories of quantum computation, it is suggested that quantum computers will be capable of simulating conscious rational thought, will be the key to achieving true artificial intelligence (T. Venkat, Shirish, 2010).

1.2.4 Limits of quantum computation

It has been proven that quantum computing allows certain problems to be solved perfectly and doesn't provide an efficiently solutions to all problems, some problems which may take a very long time on classical computers a quantum one could solve it in a couple of days, it has proven that for some problems quantum computation cannot improve it on classical methods (Eleanor, 2008). Qubits in quantum computer, arranged symmetrically, during this process the spins of the qubits may change due to external noise (E-fields, heat), to protect the qubits from such external disturbance we may require a very advanced mechanism that would avoid or illuminate such noises to interfere our structure.

Even if we design a machine that could cut the noise, the temperature comes in as another problem ("For the qubits to act normally, we require to maintain a temperature of -200 degrees Fahrenheit"), this can be done by using chemicals in order to bring the temperature, but there is another problem, as we are working at the quantum level it is somewhat difficult to determine the two parameters of the qubit (Momentum, Position) because of the uncertainty principle, Thus, the tools with which we may be working on the qubits may become a problem for the

qubits itself (T. Venkat, Shirish, 2010). Quantum computers are affected by noise and imprecision, noise is the coupling between the computer and all other systems, normally referred to as the (environment), and by imprecision we refer to an inaccuracy in the process of applying quantum gates which are applied to the computer in order to make it compute. The latter case can be regarded as noise acting all the qubits involved in the gate, followed by a perfect implementation of the gate (A. M. Steane, 1998).

1.3 Quantum Noise and Error

In quantum computation, a quantum states is manipulated in such a way that coherence is preserved, coherence means essentially if a computer can be in states $|\psi_1\rangle$ and$|\psi_2\rangle$, then during the process of a computation it might adopt the state: $|A\rangle = \frac{1}{2}(|\psi_1\rangle + e^{i\phi}|\psi_2\rangle)$ to preserve coherence of the states, it requires $|\psi_1\rangle$ and $|\psi_2\rangle$ stay unchanged by noise and imprecision, also the value of the phase in the superposition is well denned. However, the phase becomes undefined either if an imprecise operation produces rotation of the state through an unknown angle of order π, or if the coupling to the environment lead to an entangled state such as: $(|\psi_1\rangle|e_1\rangle + e^{i\phi}|\psi_2\rangle|e_2\rangle)/\sqrt{2}$. Where $|e_1\rangle$ are states of the environment and the $\langle e_2|e_1\rangle = 0$. For instance if there are K qubits in the computer, and these qubits are taken to be physically separated systems such as atoms, quantum computation is only more efficiently than classical computation if states of the form $|A\rangle$feature predominantly, such that $O(K)$ of the qubits are involved in the interference (e.g. $|\psi_1\rangle = |000\ldots.0\rangle$, $|\psi_2\rangle = |111\ldots.1\rangle$), therefore at any stage of the computation the computer fails if any one of the K qubits decohered, that is, becomes entangled with the environment or randomly processes. If the probability for a decohering process is p for any qubit, during the time it takes to perform one computational step, and there are S steps in the whole computation, then the probability that the quantum algorithm succeeds is of order $(1-p)^{KS} \simeq 1 - KSp$. For a successful computation, we therefore require $p \leq$

$O(1/KS)$ (A. M. Steane, 1998). Quantum information is highly sensitive to errors and noise more than classical information (Seth Lloyd, 2009), quantum noises are caused by imperfect quantum operations, and coupling between the quantum system and with its environment (Ronald, 2016). An error on a quantum bit could take the form of a rotation by an unknown (angle θ, axis), a qubit can either be flipped about the x-axis (σ_x), y-axis (σ_y) and z-axis (σ_z), or a combination of these effects simultaneously (Seth Lloyd,2009), such process is analyzed to a sum of "error operators", known as tensor products of Pauli spin operators, these are the analogues of classical error vectors (Ronald, 2016). Decoherence is another form of quantum error which caused by environmental monitoring, decoherence destructs the quantum coherence between preferred states associated with the observables monitored by the environment (Wojciech, 2003). (Qubit initialization, measurement errors, qubit loss and leakage) are considerable sources of error which can effect the computation of quantum information processing (Simon, et al., 2013).

1.4 Problem of The Research in Question

It turns out that quantum information susceptible to decoherence due to quantum noise, which causes errors in quantum state of the qubits. In particular we will consider only one type of quantum errors; the bit-flip error (flips the state $|0\rangle \leftrightarrows \langle 1|$). Here we will attempt to find a complete or partial solution to this problem, and certainly we will go to use the approach of quantum error correction to protect qubits of the quantum system in question from coupling with environment, and applying a correcting code on the decohered qubits to correct the error on qubit state and restore the lost information.

1.5 Outline of The Dissertation

- chapter I is an introduction to quantum computers, and quantum computing, we are going to show why quantum computer has the upper hand over classical one by illustrating some advantages of quantum computing, we also going to know about the limits and problems of quantum computation, and the methods for solving some of this problems will be suggested briefly at the end of the last chapter.
- In chapter II we will describe in a bit of details quantum gates (which are act on qubits), and its properties, in additional to reversible computation.
- In chapter III we are going to use these quantum gates to perform some simple computations on qubits, in addition we show the combination of these gates together to generate new quantum gates and form quantum circuits.
- Chapter IV is all about an importance quantum algorithms as (Grover's searching algorithm, Shor's factoring algorithm), we will explain these algorithms and show how it works and use them to correct some quantum errors, and also we will know about some common techniques used in quantum algorithms such as quantum Fourier transform.
- Finally in chapter V we will discuss the current challenge and obstacles that face the field of quantum computing and we suggest some techniques and ideas which can help to boost the practical side of quantum computing and error correction.

CHAPTER II

QUANTUM GATES

Processing of classical information is accomplished by various logic gates which act on the bits being processed (Phillip, et al., 2007). In quantum computation, quantum gates are required to perform operations on quantum information (qubits), these gates are essentially evolutions of quantum states (Abdullah, 2011). Quantum gates (as opposed to classical gates) are unitary transformations of a quantum state chosen from a continuous set, the unitary transform U is composed of elementary gates acting on a fixed number of qubits (Julia, 2005). The effect of a unitary transformation Uon a state s is described by the corresponding rotation of the vector $|s\rangle$ in Hilbert space. For this reason, U stand both for the quantum mechanical transformation as well as for the unitary rotation:

$$|U(\mathrm{s})\rangle = U|\mathrm{s}\rangle = U\left(\sum_i \alpha_i |i\rangle\right) = \sum_i \alpha_i U|i\rangle = \sum_i \alpha_i \sum_j U_{ji}|j\rangle \qquad (2.1)$$

Where U_{ji} denotes the matrix element of U positioned at the j-th row and the i-th column. It follows from the associativity of matrix multiplication that the effect of two consecutive transformation U and W is the same as the single transformation ($W \cdot U$). Just as matrix multiplication does not commute, so does the order of a sequence of unitary transformations matter: in general $W\,U \neq U\,W$.

We can restate this in a more intuitive way by saying that it makes a difference if we first do U and then W, or the other way around. A typical example of this phenomenon is given by the matrices

$$W = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \quad \text{and} \quad U = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \qquad (2.2)$$

With clearly $WU \neq U\,W$ (Willem, 2004).

2.1 Elementary Quantum gates Operations

Pauli operators represent unitary evolution (U-transformation) which may take place on a single qubit; four standard operators acting on a single qubit are the Pauli sigma operators, defined by:

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

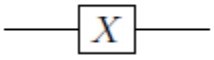
$$Y \equiv \sigma_2 = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \qquad Z \equiv \sigma_3 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \tag{2.3}$$

(Where these matrices are written in the basis $|0\rangle, |1\rangle$), the standard notation for the Pauli operators is $\sigma_i, i = 0, 1, 2, \ldots, n$, the Pauli operators form a basis set for the vector space to operators on a single qubit. The X Pauli operator is known as the quantum not gate, as it flips the basis states, $X|0\rangle = |1\rangle$ and $X|1\rangle = |0\rangle$ much as the classical not gate interchanges 0 and 1. The Z Pauli operator is a phase flip gate, as it flips the relative phase of the basis state, $Z|0\rangle = |0\rangle$ and $Z(X|0\rangle) = -|1\rangle$. At present there is no widely accepted term for the Yoperator. I is the identity gate (Michael, 1998). Quantum gates are unitary operation, the most important elementary quantum gate operations are as follows:

2.1.1 Not-Gate

Is the simple quantum operation on one qubit, it acts linearly on a general quantum state as

$$X(\alpha|0\rangle + \beta|1\rangle) = \alpha|1\rangle + \beta|0\rangle \tag{2.4}$$



In fact quantum Not-gate operator is the Pauli operator X as in eq. (2.3) (Xinlan, 2002).

2.1.2 Hadamard Gate

The Hadamard gate H is defined as that gate mapping the basis states as follows:

$$H|0\rangle = \frac{1}{\sqrt{2}}\,(|0\rangle + |1\rangle\,, \qquad H|1\rangle = \frac{1}{\sqrt{2}}\,(|0\rangle - |1\rangle) \tag{2.5}$$

The Hadamard gate has the following matrix representation:

$$H = 1/\sqrt{2}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix} = \begin{bmatrix}\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}}\\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}}\end{bmatrix} \tag{2.6}$$

One useful property of the Hadamard gate is that it is self-inverse, meaning $H = H^{-1}$

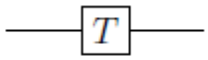
$$H(\frac{1}{\sqrt{2}}\,(|0\rangle + |1\rangle)) = |0\rangle$$

$$H(\frac{1}{\sqrt{2}}\,(|0\rangle - |1\rangle)) = |1\rangle \text{ (Phillip, et al., 2007).} \tag{2.7}$$

2.1.3 Phase-Shift Gate

Is a one qubit gate where $\theta = \pi/4$ is referred to as $\pi/8$ gate, or T which acts on the basis states as follows:

$$T|0\rangle = |0\rangle, \qquad T|1\rangle = e^{i\frac{\pi}{4}}|1\rangle \tag{2.8}$$



The name $\pi/8$-phase gate is from the fact that transformation can be represented with $\theta = \pi/8$ as following matrix representation:

$$T = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix} \quad = e^{i\pi/8}\begin{bmatrix} e^{-i\pi/8} & 0 \\ 0 & e^{i\pi/8} \end{bmatrix} \tag{2.9}$$

which is why we call it a $\pi/8$-gate (Emma, 2011), any unitary operator acting on a 2-dimensional quantum system (a qubit) is called a '1-qubit quantum gate' (Michael, 1998). Every 1-qubit pure state is represented as a point on the surface of the Bloch sphere, or equivalently as a unit vector whose origin is fixed at the centre of the Bloch sphere. A 1-qubit quantum gate U transforms a quantum state $|\psi\rangle$ into another quantum, state $U|\psi\rangle$. In terms of the Bloch sphere, the action of U on $|\psi\rangle$ can be thought of as a rotation of the Bloch vector for $|\psi\rangle$ to the Bloch vector for $U|\psi\rangle$. For example, the not gate takes the state $|0\rangle$ to the state $|1\rangle$ (and takes $|1\rangle$ to $|0\rangle$). In terms of the Bloch sphere, this action can be visualized as a rotation through an angle π about the x axis, as illustrated in Figure (2.1):

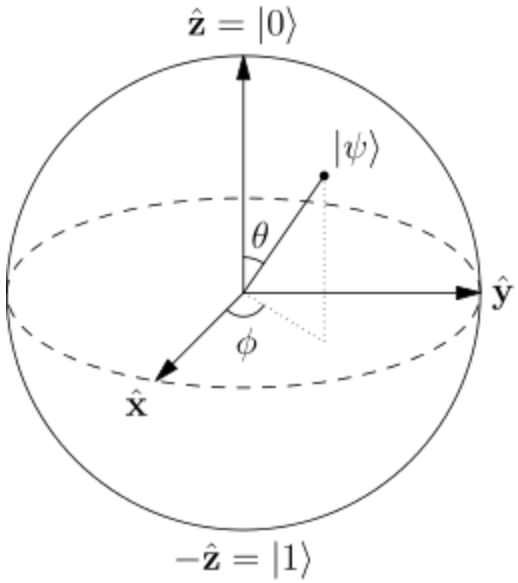


Figure (2.1): Bloch sphere

If we take the exponential of Pauli gates, we get unitary operators corresponding to very important classes of 1-qubit gates. These are the rotation gates, which correspond to rotations about the $x, y\ and\ z-axis$ of the Bloch sphere. They are defined in terms of the Pauli gates as shown in eq. (2.3).The rotation gates are defined as follows:

$$R_x(\theta) = e^{-\frac{1}{2}(\theta_x)} \qquad R_y(\theta) = e^{-\frac{1}{2}(\theta_y)} \qquad R_z(\theta) = e^{-\frac{1}{2}(\theta_z)} \tag{2.10}$$

Consider an arbitrary 1-qubit state, written in terms of its Bloch vector angles σ and:

$$\cos\left(\frac{\sigma}{2}\right)\ |0\rangle + \ e^{i\tau}\sin(\frac{\sigma}{2})\,|1\rangle \tag{2.11}$$

In the basis, this can be written as the column vector

$$\begin{pmatrix} cos\left(\frac{\sigma}{2}\right) \\ e^{i\tau}\,sin(\frac{\sigma}{2}) \end{pmatrix} \tag{2.12}$$

The effect of applying $R_z(\theta)$ on this state can be seen by performing a matrix multiplication:

$$\begin{pmatrix} e^{-i\frac{\theta}{2}} & 0 \\ 0 & e^{i\frac{\theta}{2}} \end{pmatrix} \begin{pmatrix} cos\left(\frac{\sigma}{2}\right) \\ e^{i\tau}\, sin(\frac{\sigma}{2}) \end{pmatrix} = \begin{pmatrix} e^{-i\frac{\theta}{2}} cos\left(\frac{\sigma}{2}\right) \\ e^{i\frac{\theta}{2}}\, e^{i\tau}\, sin(\frac{\sigma}{2}) \end{pmatrix} = e^{-i\frac{\theta}{2}} \begin{pmatrix} cos\left(\frac{\sigma}{2}\right) \\ e^{i\theta}\, e^{i\tau}\, sin(\frac{\sigma}{2}) \end{pmatrix}$$

$$= e^{-i\frac{\theta}{2}} \left(cos\left(\frac{\sigma}{2}\right) \ |0\rangle \ + \ e^{i(\tau+\theta)}\, sin(\frac{\sigma}{2})\, |1\rangle \right) \qquad (2.13)$$

We see that effect of $R_z(\theta)$ has been changing the angle τ to $\tau + \theta$, which is a rotation of θ about the $z-axis$ of the Bloch sphere. To see that $R_X(\theta)$ and $R_y(\theta)$ implement rotations about the $x\ and\ y-axis$ of the Bloch sphere is trickier, because such rotations involve changes to both angles σ and so using the result in eq.(2.13) we can write the rotation gates as:

$$\begin{aligned} R_X(\theta) \equiv \ e^{\frac{-i\theta x}{2}} = \ cos\left(\frac{\theta}{2}\right) \ I + (-\ i\, sin(\frac{\theta}{2})\, X) \\ R_y(\theta) \equiv \ e^{\frac{-i\theta y}{2}} = \ cos\left(\frac{\theta}{2}\right) \ I + (-\, i\, sin(\frac{\theta}{2})\, Y) \\ R_z(\theta) \equiv \ e^{\frac{-i\theta z}{2}} = \ cos\left(\frac{\theta}{2}\right) \ I + \ (-i\, sin(\frac{\theta}{2})\, Z\ ) \end{aligned} \qquad (2.14)$$

Knowing the matrices for I, X, Y and Z in the basis, we can now write the rotation gates as matrices in the basis:

$$R_X(\theta = \begin{bmatrix} cos\left(\frac{\theta}{2}\right) & -i\, sin\left(\frac{\theta}{2}\right) \\ -i\, sin\left(\frac{\theta}{2}\right) & cos\left(\frac{\theta}{2}\right) \end{bmatrix} \qquad R_y(\theta) = \begin{bmatrix} cos\left(\frac{\theta}{2}\right) & i\, sin(\frac{\theta}{2}) \\ i\, sin(\frac{\theta}{2}) & cos\left(\frac{\theta}{2}\right) \end{bmatrix}$$

$$R_z(\theta) = \begin{bmatrix} e^{-i\frac{\theta}{2}} & 0 \\ 0 & e^{i\frac{\theta}{2}} \end{bmatrix} \quad \text{(Phillip, et al., 2007)} \qquad (2.15)$$

Small rotations

What if we want to apply only a small rotation about one of these axes? Let us consider the time-dependent Schrödinger equation, given by

$$ih\frac{\partial}{\partial t}\,|\psi\rangle\, = \hat{H}|\psi\rangle \tag{2.16}$$

Where $\hat{H}$ is the Hamiltonian which governing the time evolution. For time independent $\hat{H}$, we can solve this equation with $|\psi(t)\rangle = e^{-iHt/\hbar}|\psi\rangle$. If $\hat{H} = \frac{1}{2}\Omega\,\vec{n}.\vec{\sigma}$, where $\vec{n}$ is an arbitrary vector which defines our rotation axis, the corresponding unitary is given by $\hat{U} = e^{-i\hat{H}\,t} = e^{-i\frac{\Omega t}{2}\vec{n}.\vec{\sigma}}$. Taylor-expanding the exponential $\hat{U} = \sum_{j=0}^{\infty}\frac{1}{j!}\,(\frac{-i\Omega t}{2}\,\vec{n}.\vec{\sigma})^{j}$, and utilizing the Pauli operator identities, we are left with $\sum_{j\in evens}^{\infty}\frac{1}{j!}\,(\frac{-i\Omega t}{2})^{j}\,I + \sum_{j\in odd}^{\infty}\frac{1}{j!}\,(\frac{-i\Omega t}{2})^{j}\,(\hat{n}\,.\,\hat{\sigma}\,)$. We can identify the two sums as sines and cosines, giving us $\hat{U}(t) = cos\left(\frac{\Omega t}{2}\right)I - \,i\,sin\left(\frac{\Omega t}{2}\right)(\hat{n}\,.\,\hat{\sigma}) = R_{\hat{n}}^{\Omega t}$. Thus, we can control the amount of rotation driven by our Hamiltonian by simply changing the period of time for which we apply it. Equivalently, we could change the parameter Ω, which represents the coupling or drive strength of our rotation. For example, if we take $\hat{n} = \hat{z}$, then $\hat{U}$ is diagonal in the basis as

$$\hat{U}(t) = \begin{pmatrix} e^{\frac{-i\Omega t}{2}} & 0 \\ 0 & e^{\frac{+i\Omega t}{2}} \end{pmatrix} = \, e^{\frac{-i\Omega t}{2}}\begin{pmatrix} 1 & 0 \\ 0 & e^{\frac{i\Omega t}{2}} \end{pmatrix} \tag{2.17}$$

We can control the difference between $|0\rangle$and $|1\rangle$ by applying the σ_z operator. Geometrically, this corresponds to rotations about the z−axis as a function of time, our state processes about z. For arbitrary $\vec{n}$, if we choose our qubit basis as states pointing parallel and anti-parallel to $\vec{n}$, the unitary operation is exactly as written above, where in the second equation we have factored out the irrelevant global phase, so we can arbitrarily control the phase (Matthew, 2013).

2.2 Multi-Qubits Quantum Gates

Quantum gates of 2-qubits and higher are classified into: Firstly, analog of classical gates where the inputs determines the out puts (the out puts depends on the inputs). Secondly, what so called "controlled " gates, they have the property that one of the (input-output) pair known as the target qubit has an action done on it if and only if the other (input-output) pairs called the control qubit have a certain value, the control qubits are unaffected by the gate in either case (Abdullah, 2011). Gates operate on a manifold of multiple qubits must also be realized, since the state vector has 2^N elements, these operators must be 2^N by 2^N matrices. Consider a set of $k = (x, y, z)$ Pauli operators that each act on only the jth qubit σ_k^i, where the superscript denotes which qubit it addresses. For example, if we have two qubits, an $X-$operation on the first qubit would be given by the tensor product of σ_x^1, and σ_i^2. A single qubit gate is one where all but one of the operators in the tensor product are I; having two or more non-identity operations constitutes a multi-qubit gate. For example, an X-operation on two qubits simultaneously would be given by $\sigma_x^1 \otimes \sigma_x^2$, and is commonly abbreviated as σ_{xx} or simply XX. Some particularly common gates include the "swap gate," which maps $|01\rangle \Longleftrightarrow |10\rangle$, and does nothing to $|00\rangle$ or $|11\rangle$, and is given by the matrix

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

There are also $\text{Control}-\text{Not}$ gates, where a target qubit is flipped if and only if a control is excited, and is given by the matrix

$$\text{Control}-\text{Not} = \Lambda(\sigma_x) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

The $\text{Control}-\text{Not}$ gate is naturally extendible to being "controlled" by more than one qubit; for example, the three-qubit Toffoli gate flips some qubit if and only if two controls are excited, and is therefore also known as a $\text{Control}-\text{Control}-\text{Not}$ gate as in Figure. (2.2), another common two-qubit gate is the $\text{Control}-\text{Phase}$ gate, which flips the phase of only the $|11\rangle$ basis state:

$$\text{Control}-\text{Phase} = \Lambda(\sigma_z) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}$$

The $Control-Phase$ gate is also has a multi-qubit generalization which is known as $(\text{Contro}-\text{Control}-\text{Phase})$ gate which flips the phase of the basis state$|1..1\rangle$. Many of these multi-qubit gates can be related to one another with single-qubit rotations. Experimentally, single-qubit gates are implemented essentially the same regardless of the number of qubits (Matthew, 2013).

Note :

Quantum Control Not (Cont.Not) gate can be denoted by this operator, $\text{Cont.Not} = |00\rangle\langle 00| + |01\rangle\langle 01| + |10\rangle\langle 11| + |11\rangle\langle 10|$. The Cont.Not gate is a 2-qubit operator where the first qubit is the control and the second qubit is the target. If the control qubit is in the ground state$|0\rangle$, the value of the target qubit does not change after the action of the Cont.Not gate. In the opposite case, the target qubit changes its value. This situation is described by the first two terms and the second two terms respectively. The Cont.Not operator, like the N operator, is unitary and Hermitian,

$(\text{Cont.Not})^{\dagger} = \text{Cont.Not.} \quad (\text{Cont.Not.})^{\dagger} = |00\rangle\langle 00| + |01\rangle\langle 01| + |10\rangle\langle 10| + |11\rangle\langle 11|$

where the matrix form of the Cont.Not gate in decimal notation is,

$$\text{Cnot. Not} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

The matrix element $(\text{Cont. Not.})_{ik}$ corresponds to the term $|i\rangle\langle k|$where we count i and k from zero in Cont. Not.

The three qubit F -gate can be described by the operator, $F = |000\rangle\langle 000| + |001\rangle\langle 001| + |010\rangle\langle 010| + |011\rangle\langle 011| + |100\rangle\langle 100| + |101\rangle\langle 110| + |110\rangle\langle 101| + |111\rangle\langle 111|$. The left qubit in F is the control qubit. If the control qubit is in the ground state $|0\rangle$ the two target qubits do not change their states. This situation is described by first four terms in F. The second four terms in F describe the opposite case, with the control qubit in the excited state $|1\rangle$ and the target qubits exchanging their states. For example, $|001\rangle = |001\rangle\langle 001|001\rangle = |001\rangle$, and the state of the qubits does not change. At the same time, $F|101) = |110\rangle\langle 101|101\rangle = |110\rangle$, and the target qubits exchange their states. In decimal notation, the F gate can be written as, in matrix representation, has the form

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$ (Gennady, et al., 1999)

2.3 Universal gates set

In classical computing we implement complicated operations as a sequence of much simpler operations. In practice we want to be able to select these simple operations from some set of elementary gates. In case of quantum computing, we do the same thing using what is known universal gates sets (Phillip, et al., 2007). Universality is the ability to compute any mathematical function with a computational system and translates in quantum computation to the ability of implementing arbitrary single-qubit rotations on all qubits and entangling gates between the qubits (Andrea, 2009). There are finite sets of quantum gates that are universal in the following sense. Consider the networks that can be constructed from a countable set of gates each network will implement a unitary transformation, and we want to consider if any finite unitary transformation can be implemented in such a way, of course we cannot construct an exactly copy of every such element (Willem, 2004). Thus it suffices to have an implementation that approximates the desired unitary to some specified level of accuracy. Suppose we approximate a desired unitary transformation U by some other unitary transformation V. The error in the approximation is defined to be

$$E(U,V) \equiv \max_{|\Psi\rangle} \| (U-V)|\Psi\rangle \| \qquad (2.18)$$

When we say that an operator U can be 'approximated to arbitrary accuracy', we mean that if we are given any error tolerance $\varepsilon > 0$, we can implement some unitary V such that $E(U,V) < \varepsilon$ (Phillip, et al., 2007). It has been proven that there are indeed universal sets of quantum gates with which this can be achieved, with the (Hadamard, control–Not and the phase) gates any other unitary transformation can be approximated constructed with in an arbitrary small error (with respect to some distance measure on the set of operators) (Willem, 2004). Restrictions must be placed on the gates from which quantum circuits may be made, this is done by using a suitable finite set from the following list (representing a standard choice for a gate set):

- Erasure gates: A non-unitary gates, it takes a 1-qubit as input and has no output.

- Ncillary gates: A non-unitary gates, it takes no input and produce a 1-qubit in the ground state $|0\rangle$ as output.
- Toffoli gate: is a three-qubit unitary gate.
- Hadamard gate: is a single-qubit unitary.
- Phase – shift gate: is a single-qubit unitary.

The symbols used to denote these gates in quantum circuit, shown in Figure (2.2)

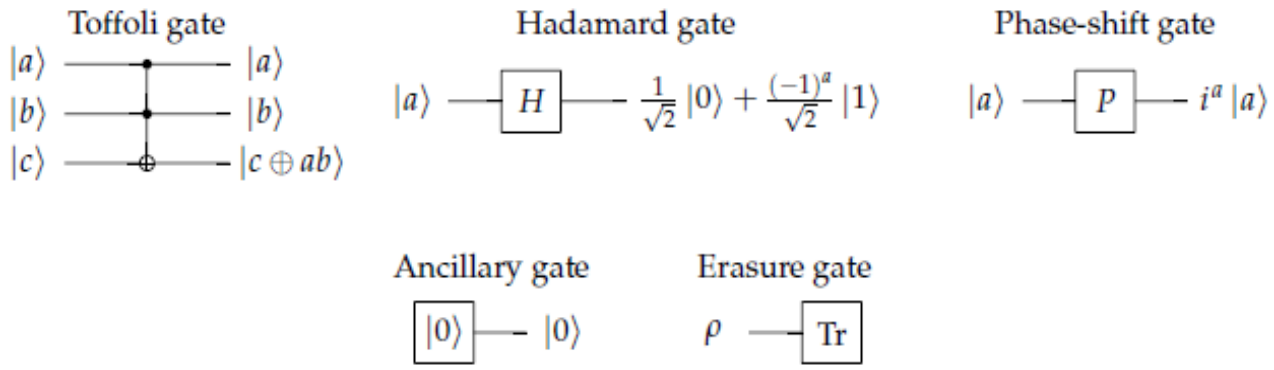


Figure (2.2)

Figure (2.2): A universal collection of quantum gates: Toffoli, Hadamard, phase– shif , ancillary, and Erasure gates (John, 2011).

2.4 Reverse of Quantum Gate

Each quantum gate has a property that numbers of inputs - outputs are equivalent, this feature comes from the fact that quantum mechanics is a theory that obeys time symmetry (A quantum mechanical system that is evolved from a given state to using a definite series of operations can be evolved backwards using the inverse of the series of operations on it), which means quantum

computer must be a reversible machine. A gate operating on an inputs to produce an outputs is be able to do the reverse if the number of inputs -outputs are similar (Abdullah, 2011).

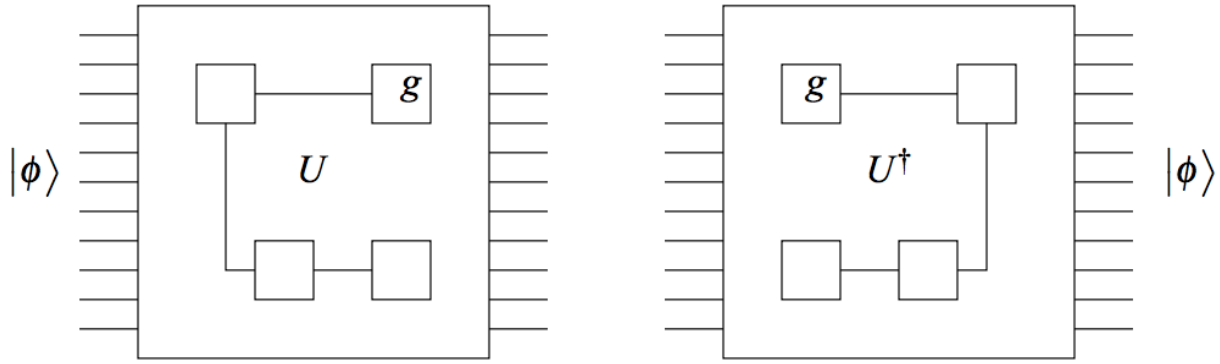


Figure (2.3) Reverse of quantum gate

A quantum circuit acting on N qubits is described by a $N \times N$ unitary operator U, since U is unitary, $UU^{\dagger} = U^{\dagger}U = I$. This implies that in a quantum circuit (each quantum gate has an inverse gate which is the mirror image of the original gate). The circuits for U and $U^{\dagger}$ are the same size and have mirror image gates which carries out the inverse operator $U^{\dagger}$(Umesh, 2012).

CHAPTER III

SIMPLE QUANTUM COMPUTATIONS

In the previous chapter we knew about quantum gates and its properties, some of these gates act on a single qubit such as Hadamard gate (see subsection 2.1.1), while the others operate on more than one qubit system as illustrated in (section 2.2). In this chapter we will manipulate these gates by applying it both on qubits system to carry simple computations, and on itself to generate new and more complicated operators such as swap gate and C_H gate, as you will see in example 4 and 6 respectively. More important and in particular you see how one can apply the Hadramard gate to more than one qubit system (see example 4). Furthermore we shall combine gates together to implement circuit which can be use in to build quantum algorithms.

Example1

Let us implement the following gates which are the special cases of the single-qubit rotation operations and implemented by the rotation pulses.

1. Hadamard gate: If $R_y(\frac{\pi}{2})$ is the rotation operator around the x-axis , and $R_z(\pi)$ around z-axis:

$$R_y\left(\frac{\pi}{2}\right) = \begin{bmatrix} cos(\frac{\frac{\pi}{2}}{2}) & -i\,sin(\frac{\frac{\pi}{2}}{2}) \\ i\,sin(\frac{\frac{\pi}{2}}{2}) & cos(\frac{\frac{\pi}{2}}{2}) \end{bmatrix} = \begin{bmatrix} cos(\frac{\pi}{4}) & -i\,sin(\frac{\pi}{4}) \\ i\,sin(\frac{\pi}{4}) & cos(\frac{\pi}{4}) \end{bmatrix}$$

$$R_z(\pi) = \begin{bmatrix} e^{-i\frac{\pi}{2}} & 0 \\ 0 & e^{i\frac{\pi}{2}} \end{bmatrix}$$

Then

$$H = iR_y\left(\frac{\pi}{2}\right)R_z(\pi) = i\begin{bmatrix} -i^2 & 1 \\ 1 & -i^2 \end{bmatrix}\begin{bmatrix} cos(\frac{\pi}{4}) & -i\,sin(\frac{\pi}{4}) \\ i\,sin(\frac{\pi}{4}) & cos(\frac{\pi}{4}) \end{bmatrix}\begin{bmatrix} e^{-i\frac{\pi}{2}} & 0 \\ 0 & e^{i\frac{\pi}{2}} \end{bmatrix}$$

$$= 1/\sqrt{2}\begin{bmatrix} i & 1 \\ -1 & i \end{bmatrix}\begin{bmatrix} e^{-i\frac{\pi}{2}} & 0 \\ 0 & e^{i\frac{\pi}{2}} \end{bmatrix} = (\frac{1}{\sqrt{2}})\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

So

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

2. Pauli-x gate

$$\text{If } R_x(\pi) = \begin{bmatrix} \cos(\frac{\pi}{2}) & -i\sin(\frac{\pi}{2}) \\ -i\sin(\frac{\pi}{2}) & \cos(\frac{\pi}{2}) \end{bmatrix} \qquad i = i\begin{bmatrix} -i^2 & 1 \\ 1 & -i^2 \end{bmatrix}$$

$$X = i\,R_x(\pi) \quad = \quad i\begin{bmatrix} -i^2 & 1 \\ 1 & -i^2 \end{bmatrix}\cdot\begin{bmatrix} \cos(\frac{\pi}{2}) & -i\sin(\frac{\pi}{2}) \\ -i\sin(\frac{\pi}{2}) & \cos(\frac{\pi}{2}) \end{bmatrix}$$

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad \text{(Marek, 2011).}$$

Implementing the X-Pauli gate using H, Z gates

$$H = 1/\sqrt{2}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \qquad Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

$$X = HZH$$

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

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

Pauli-X gate also known as quantum Not-gate, one of it is application is to transform the state of the qubit from $|0\rangle \longrightarrow |1\rangle$ the unitary of quantum Not-gate is

$$U_{not} = \begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix} \qquad |0\rangle = \begin{bmatrix}1\\ 0\end{bmatrix} \qquad |1\rangle = \begin{bmatrix}0\\ 1\end{bmatrix}$$

$$U_{not}|0\rangle = \begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix} = \begin{bmatrix}1\\ 0\end{bmatrix} = \begin{bmatrix}0\\ 1\end{bmatrix} = |1\rangle$$

$$U_{not}|1\rangle = \begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix} = \begin{bmatrix}0\\ 1\end{bmatrix} = \begin{bmatrix}1\\ 0\end{bmatrix} = |0\rangle$$

Example 2

Implementing Control– Not gate as in Figure (3.1) by combining (H, Control– Z) gates together, the controlled not gate can be implemented (as it is equivalent) to two H gates and a Control– Z gate, as shown in Figure below:

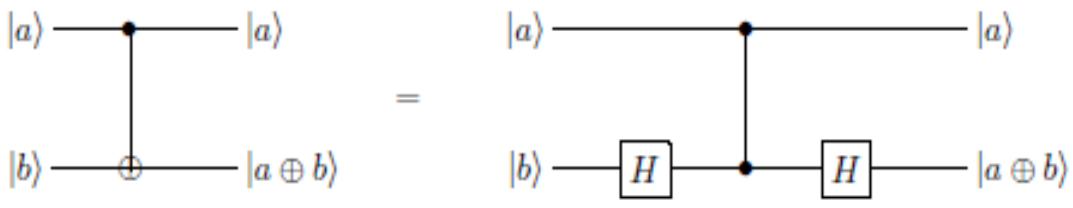


Figure (3.1): Equivalence between Control– Not gate and Control– Z gate

$$\text{Control}-\text{Not} = (I \otimes H)\ \text{Control}-\text{Z}\ (I \otimes H)$$

$$I = \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix} \qquad H = 1/\sqrt{2}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix} \qquad \text{Control}-\text{Z} = \begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & -1\end{bmatrix}$$

Therefore

$$\text{Control}-\text{Not} = \left(\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix} \otimes \tfrac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}\right)\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & -1\end{bmatrix}\left(\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix} \otimes \tfrac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}\right)$$

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

Control– Not gate is equivalent to classical gate XOR, it acts on a two qubits in superposition as following: In dirac notation

$$|\psi\rangle = |00\rangle + |01\rangle + |10\rangle + |11\rangle$$

so

$$U_{Cnot}|\psi\rangle = U_{Cnot}(|00\rangle + |01\rangle + |10\rangle + |11\rangle)$$

$$U_{Cnot}|\psi\rangle = U_{Cnot}|00\rangle + U_{Cnot}|01\rangle + U_{Cnot}|10\rangle + U_{Cnot}|11\rangle$$

$$= |00\rangle + |01\rangle + |11\rangle + |10\rangle$$

In matrix representation

$$|00\rangle = \begin{bmatrix}1\\0\\0\\0\end{bmatrix} \qquad |01\rangle = \begin{bmatrix}0\\1\\0\\0\end{bmatrix} \qquad |10\rangle = \begin{bmatrix}0\\0\\1\\0\end{bmatrix} \qquad |11\rangle = \begin{bmatrix}0\\0\\0\\1\end{bmatrix}$$

The unitary matrix of $Control-Not$ gate is: $U_{Cnot} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$

$$U_{Cnot}|\psi\rangle = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} |00\rangle \\ |01\rangle \\ |10\rangle \\ |11\rangle \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

(Moayad, et al., 2010).

Example 3

In the Bloch sphere picture, every operation (or gate) on a single qubit can be interpreted as a rotation around an axis of the sphere. Consider the operation on the following qubits where the initial state $|1\rangle$ is rotated by an angle $\pi/2$ around the x-axis and ends up along the y-axis in the state $|i\rangle$. Mathematically the state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ is then written as $\begin{bmatrix} \alpha \\ \beta \end{bmatrix}$, since operations on single qubits can be described as unitary 2×2 matrices, the above example is then

$$U|\psi\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & -i \\ -i & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} -i \\ 1 \end{bmatrix} \cong \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ i \end{bmatrix}$$

where the last equivalence is due to the irrelevant global phase factor (Lars, 2013).

Example 4

We want to apply Hadamard gate to two qubits at the same time, for this we have two qubit inputs, and so require a 4 by 4 matrix operator. To get this we find the tensor product of the Hadamard gate and itself.

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

Doing the same for three qubits would require $H \otimes H \otimes H$

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

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

$$= \left(\frac{1}{2}\right)^{1/3} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \end{bmatrix}$$

And for any n qubit inputs we have $H^{\otimes n}$. We can generalise this notion to any gate U by considering it as an application of a U gate to each qubit. Given n qubit inputs, we can perform the operator on all inputs by finding the operator $U^{\otimes n}$ (Joey, 2010) (Magnus, 2009).

Example 5

Let us create a useful operator by combining three gates together, the swap gate, illustrated in Figure (3.2)

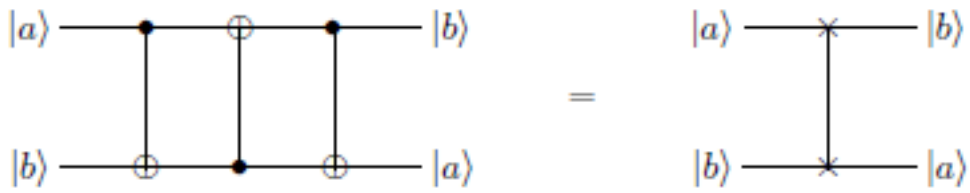


Figure (3.2): Defining the quantum swap gate in terms of three Control– Not gates

We use the Control– Not gate described above, and we will flip the gate such that our bottom qubit is the control, and our X–gate will be performed on the top qubit (Control– Not gate):

$$\text{Control– Not} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \qquad \text{Not– Control} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

From Figure (3.2), we require the result of the matrix multiplication of Control– Not gate, Not– Control gate:

$$(\text{Control– Not})(\text{Not– Control})(\text{Control– Not}) =$$

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

$$\text{swap} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Although we may represent the swap gate as a single gate in a circuit, we see it can be decomposed into Control_Not gates (Joey, 2010).

Example 6

Implementing the C_H gate, we define the C_H gate as $C_H = \text{Control–Not} \cdot H_1$, with $H_1 = H \otimes I$ which has the matrix form

$$H_1 = 1/\sqrt{2} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = 1/\sqrt{2} \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix}$$

so

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

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

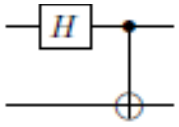
$C_H =$ 

Figure (3.3) C_H gate

Note that the C_H gate is the first example for the Bell transform which is a unitary transformation from the product basis to the Bell basis. With the quantum circuits of the H gate and the Control–Not gate, the associated quantum circuit of the C_H gate is in Figure (3.3) (Yong, Kun, 2016).

Example 7

Let us implement the Bell state $|\phi^{\pm}\rangle = \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle)$ using the following simple quantum circuit in Figure (3.4), it consist of Hadamard and Control– Not gate

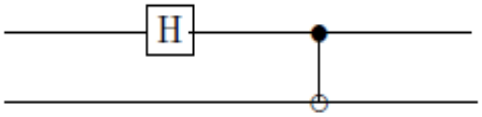


Figure (3.4): A circuit for implementing Bell state

The first qubit passed through Hadamard gate from left to right, after Hadamard both qubits are entangled by a Control– Not gate. If the input to the system is$|0\rangle\otimes|0\rangle$, then the Hadamard gate changes the state to

$$H|0\rangle = \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\ \langle 0| + \frac{|0\rangle + |1\rangle}{\sqrt{2}}\ \langle 1|\right)|0\rangle = \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right)$$

$$\left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right) \otimes |0\rangle = \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|10\rangle$$

and after the Contol– Not gate the state becomes $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$, the Bell state $|\varphi^{+}\rangle$. The action of the Control– Not gate is entangling the states, but not copying as our classical intuition would suggest. The state $|\varphi^{+}\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ is one of four Bell basis states:

$$|\varphi^{+}\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle), \qquad |\phi^{+}\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle)$$

$$|\varphi^{-}\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle), \qquad |\phi^{-}\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle)$$

(Umesh, 2012).

CHAPTER IV

QUANTUM ALGORITHMS

Quantum algorithms are defined as any quantum effects use to perform useful computational tasks on quantum systems. It is implemented by an appropriate sequence of unitary operations, applied on qubits as a form of unitary transformations known as (Quantum gates) (Sarah, 2008). A quantum algorithm is a step-by step procedure to solve a problem, with each step executed by a quantum computer (Yazhen, 2012). A computational problem is defined to be easy if there exist an efficient algorithm to solve it, and if no such algorithm exists then the problem is hard, thus a large class of interesting problems, such as (database searching) turn out to be easy, but there still some appear to be intrinsically hard (Tim, 2005). Quantum algorithms capable of solving certain problems faster than is possible classically, however and because of the difficulty of getting around the issue of measurement, only a few useful quantum algorithms have so far been discovered, the most well-known examples include Grover's search algorithm, which can search an unstructured database quadratically faster than is possible with a classical computer, the other is Shor's factoring algorithm (Matthew, 2013), which is exponentially faster than the classical algorithms (Jill, 2006), also the Deutsch-Jozsa algorithm for determining whether a function is balanced or constant, which also grants an exponential increase .The first two algorithms will only return a correct answer with a high probability and may require several repetitions, this is in contrast to Deutsch-Jozsa, which is deterministic and will always return the correct value if the algorithm was run successfully (Matthew, 2013).

4.1 Importance of algorithms in quantum computer

Intrinsic irreversibility of the basic operations in usual classical computer is a source of energy consumption, a gate like AND maps two input bits to one output bit, it means the input cannot be reconstructed from the output, because one bit of information is erased during the operation of the AND gate, hence an amount of energy is dissipated to the environment, to avoid this, we restrict to reversible processes, it should be possible to reconstruct the input data from the output data. This is called reversible computations, and it is performed in terms of reversible gates included in quantum algorithms, which is in particular importance for quantum computing and quantum computer (Michael, 2002). Quantum algorithms able to solve problems in polynomial-time using quantum effects such as superposition, this gives quantum computer an exponential speedup in information processing and allow performing many classical computation in parallel (S.A. Duplij, 2007) (Samuel, 2011), this what makes quantum algorithms exciting because it is fast compared to classical algorithms, for solving some tough problems (Yazhen, 2012). For instance Shor's factoring algorithm which is based on quantum Fourier transform (a powerful principle that leads to quantum computers) can factor an integer in a time that grows polynomially faster than any current known classical computer (Kathy-Anne, 2007) (Sarah, 2008).

4.2 Quantum Searching algorithm

Unstructured searching data basis

Unstructured searching problems usually in the form of find some x in a set of possible solution such that statement $P(x)$, where nothing is known about the structure of the solution space and the statement P. For instance, determining$P(x_0)$, provides no information about the possible value of $P(x_1)\ for\ x_0 \neq x_1$ (Javier, 2009). A database is nothing but a collection of items; suppose we have a database consists of N words, we want to search for the word say

"Computing", we go through every word and check whether this is "Computing" or not. Solution of this problem on classical computer would take $N/2$ queries on average, and in the worst case it would need $N-1$ query. If the word "Computing" occurs M times, then it is easy to show that we require $O(N/M)$ trial to succeed with classical algorithm because any randomly chosen word will be computation with the probability M/N. For $M=1$, we require $O(N)$ trial to succeed (Debasis, 2012). On a quantum computer, this problem can be solved easily and efficiently by using Grover's Quantum Algorithm. The basic idea of this quantum algorithm is to rotate the initial state of the qubit system representing the database in the direction of the searching state with the help of a unitary quantum version of the oracle (Sarah, 2008).

4.2.1 Grover's Search Algorithm

Grover's Algorithm searches an N_object unsorted database for an object in $O(\sqrt{N})$ order of operations, offering quadratic speedup (Magnus, 2009). The heart of Grover's algorithm is a sequence of gates known as the Grover iterate, $G = ([2/N] - I)O_x$. Where I is the $N \times N$ matrix with $2/N$ in every entry, and O_x is the oracle $O_{\pm}$for the input x. The unitary matrix $([2/N] - I)$ is called the diffusion transform, implemented as $-H^{\otimes n} O_G H^{\otimes n}$, where $O_G |\vec{0}\rangle = -|\vec{0}\rangle$, and does nothing to all other states. Grover's algorithm is embedded as follows:

1. The initial state starts in the n-qubit state$|\vec{0}\rangle$.
2. Applying Hadamard transform H to all qubit in $|\vec{0}\rangle$, resulting a uniform state of superposition

$$\frac{1}{\sqrt{N}} \sum_i |i\rangle$$

3. Applying the Grover iterate $O(\sqrt{N})$ times.
4. Measuring the superposition, collapse it into a single state (Jill, 2006).

4.2.2 Grover's Search Operator

Consider a quantum system with an N-dimensional Hilbert space, whose basis states$|j\rangle, j \epsilon 0, 1, \ldots, N-1$, encode the N items. The target state $|j\rangle$ corresponds to the target item, while all other basis states are non-target states. If we begin our initial state with the uniform superposition of all states

$$|\psi\rangle = \frac{1}{\sqrt{2^n}} \sum_{j=0}^{2^n-1} |j\rangle$$

So, one can write the uniform superposition to be

$$|\psi\rangle = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} |j\rangle = \sqrt{\frac{M}{N}}\,|t\rangle + \sqrt{1-\frac{M}{N}}\,|t_\perp\rangle$$

We will now define a unitary operator $G = -I_s I_t$such that

$$I_s: 1 - 2\,|t\rangle\,\langle t|$$
$$I_t: 1 - 2\,|s\rangle\,\langle s|$$ (Debasis, 2012).

Figure (4.1) schematically shows the Grover's searching algorithm operates as follows: Beginning with the initial state $|0\rangle^{\otimes n}\,|1\rangle$. After Hadamard gates the state evolve to

$$2^{-\frac{n}{2}} \sum_{x=0}^{2^n-1} |x\rangle \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

After quantum oracle

$$2^{-\frac{n}{2}} \sum_{x=0}^{2^n-1} (-1)^{f(X)} |x\rangle \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

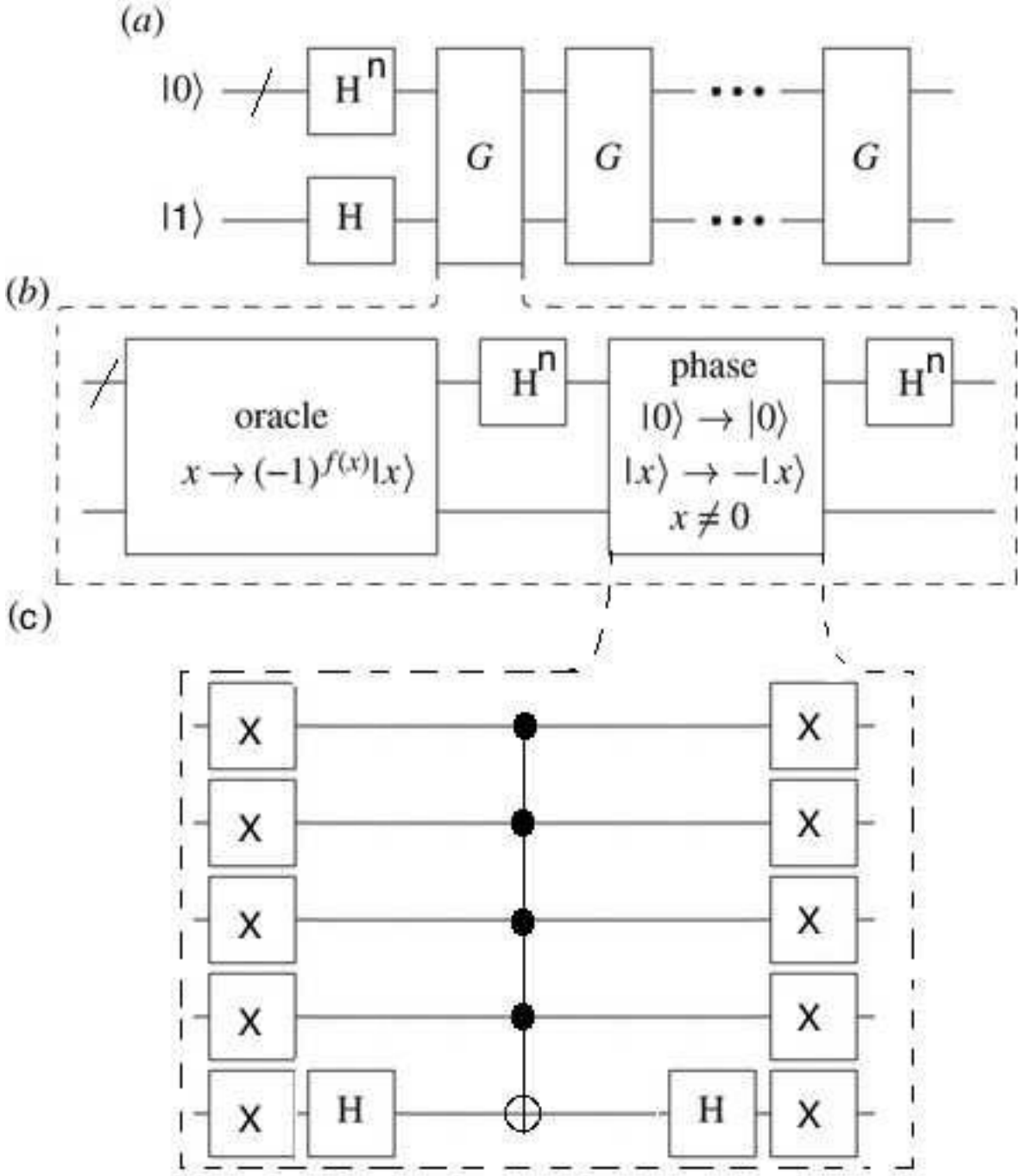


Figure (4.1): Circuit of Grover's searching algorithm

$$= 2^{-\frac{n}{2}}\left(\sum_{x\neq x_0}|x\rangle - |x_0\rangle\right)\frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

$$= 2^{-\frac{n}{2}}\left(\sum_{x=0}^{2^n-1}|x\rangle - 2|x_0\rangle\right)\frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

After Hadamard gates becomes

$$= \left(|0\rangle^{\otimes n} - 2^{-n}\sum_{x=0}^{2^n-1}(-1)^{x\cdot x_0}\,2|x\rangle\right)\frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

$$= \left(|0\rangle^{\otimes n} - 2^{1-n}\sum_{x=0}^{2^n-1}(-1)^{x\cdot x_0}\,|x\rangle\right)\frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

After Grover's phase operator we find

$$= \left((1-2^{1-n})|0\rangle^{\otimes n} + 2^{1-n}\sum_{x=1}^{2^n-1}(-1)^{x\cdot x_0}|x\rangle\right)\frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

$$= \left((1-2^{2-n})|0\rangle^{\otimes n} + 2^{1-n}\sum_{x=0}^{2^n-1}(-1)^{x\cdot x_0}|x\rangle\right)\frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

After Hadamard gates, yield

$$= \left((1-2^{2-n})2^{-\frac{n}{2}}\sum_{x=0}^{2^n-1}|x\rangle + 2^{1-n}\sum_{x=0}^{2^n-1}(-1)^{x\cdot x_0}\,2^{-\frac{n}{2}}\sum_{y=0}^{2^n-1}(-1)^{x\cdot y}|y\rangle\right)\frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

$$= \left((1-2^{2-n})2^{-\frac{n}{2}}\sum_{x=0}^{2^n-1}|x\rangle + \;2^{1-n}\,2^{-\frac{n}{2}}\sum_{x=0}^{2^n-1}\sum_{y=0}^{2^n-1}(-1)^{x\cdot(y+x_0)}|y\rangle\right)\frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

$$= \left((1-2^{2-n})2^{-\frac{n}{2}}\sum_{x=0}^{2^n-1}|x\rangle + 2^{1-\frac{n}{2}}|x_0\rangle\right)\frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

Note that the probability of measuring $|x_0\rangle$ is roughly nine times larger than the probability of measuring any other state:

$$p(|x\rangle) \approx \begin{cases} 9 \times 2 & for \ \ |x\rangle = |x_0\rangle \\ 2^{-n} & for \ \ |x\rangle \neq |x_0\rangle \end{cases}$$

Generally for more initial state

$$= 2^{-\frac{n}{2}}\left(N\sum_{x}^{2^n-1}|x\rangle + M|x_0\rangle\right)\frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

After oracle

$$= 2^{-\frac{n}{2}}\left(N\sum_{x}^{2^n-1}(-1)^{f(x)}|x\rangle - M|x_0\rangle\right)\frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

$$= 2^{-\frac{n}{2}}\left(N\sum_{x}^{2^n-1}|x\rangle\,(M + 2N)\,|x_0\rangle\right)\frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

After Hadamard

$$= 2^{-\frac{n}{2}}\left(N2^{\frac{n}{2}}|0\rangle^{\otimes n} - (M+2N)2^{-\frac{n}{2}}\sum_{x=0}^{2^n-1}(-1)^{x\cdot x_0}\,|x\rangle\right)\frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

After Grover's phase operator

$$= 2^{-\frac{n}{2}}\left((N\,2^{\frac{n}{2}} - (M+2N)2^{-\frac{n}{2}})|0\rangle^{\otimes n} - (M+2N)2^{-\frac{n}{2}}\sum_{x=1}^{2^n-1}(-1)^{x\cdot x_0}\,|x\rangle\right)\frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

$$= 2^{-\frac{n}{2}}\left((N\,2^{\frac{n}{2}} - 2(M+2N)2^{-\frac{n}{2}})|0\rangle^{\otimes n} - (M+2N)2^{-\frac{n}{2}}\sum_{x=0}^{2^n-1}(-1)^{x\cdot x_0}\,|x\rangle\right)\frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

$$= 2^{-\frac{n}{2}}\left((N - (M+2N)2^{1-n}\,)\sum_{x=0}^{2^n-1}|x\rangle - (M+2N)\,2^{-n}\sum_{x=0}^{2^n-1}\sum_{y=0}^{2^n-1}(-1)^{x\cdot(x_0+y)}|y\rangle\right)\frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

$$= 2^{-\frac{n}{2}}\left((N-(M+2N)2^{1-n})\sum_{x=0}^{2^n-1}|x\rangle-(M+2N)|x_0\rangle\right)\frac{|0\rangle-|1\rangle}{\sqrt{2}}$$

Thus after each iteration the probability of measurement of $|x_0\rangle$ increases and the probability of measurement of random states decreases as

$$N_{i+1} = N_i - (\mathrm{M}_i + 2N_i)\,2^{1-n}$$
$$N_{i+1} = M_i + 2N_i$$

Worked example

Consider a system with $N = 8 = 2^3$ states; we are searching for the state, x_0 represented by the bit string 011: To describe this system, $n = 3$ qubits are required, represented as:

$$|x\rangle = \alpha_0|000\rangle + \alpha_1|001\rangle + \alpha_2|010\rangle + \alpha_3|011\rangle + \alpha_4|100\rangle + \alpha_5|101\rangle + \alpha_6|110\rangle + \alpha_7|111\rangle$$

Where α_i is the amplitude of the state$|i\rangle$, Grover's algorithm begins with a system initialized to 0:

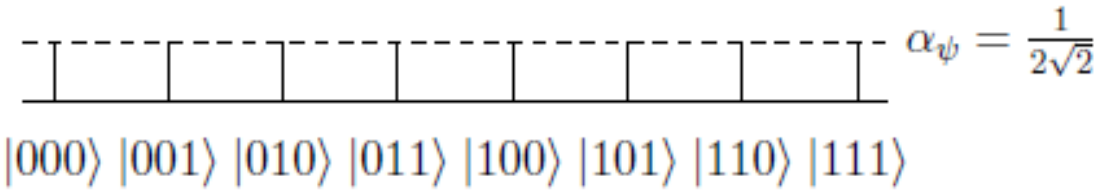
$$1\ |000\rangle$$

and then apply the $Hadamard$ transformation to obtain equal amplitudes associated with each state of $1/\sqrt{N} = 1/\sqrt{2^3} = 1/2\sqrt{2}$ and thus also equal probability of being in any of the 8 possible states:

$$H^3|000\rangle = \tfrac{1}{2\sqrt{2}}|000\rangle + \tfrac{1}{2\sqrt{2}}|001\rangle + \cdots + \tfrac{1}{2\sqrt{2}}|111\rangle = \tfrac{1}{2\sqrt{2}}\textstyle\sum_{x=0}^{7}|x\rangle = |\psi\rangle \qquad (4.1)$$

Throughout the execution of Grover's algorithm, the amplitudes of the states remain real, so we may visualized as lines perpendicular to an axis whose lengths are proportional to the amplitude

they represent. The equal superposition of states resulting from the first $Hadamard$ transform appears as follows:



It is optimal to now perform 2 Grover iterations in order to obtain the solution:

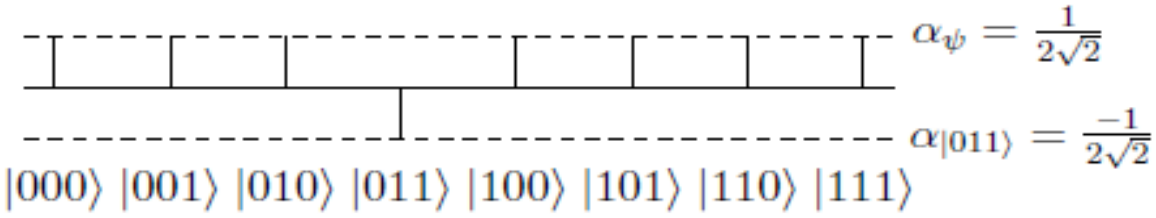
$$\frac{\pi}{4}\sqrt{8} = \frac{2\pi}{4}\sqrt{2} = \frac{\pi}{2}\sqrt{2} \approx 2.22$$

these rounds to 2 iterations.

In each iteration, the first step is to call the quantum oracle$\mathcal{O}$, then perform inversion about the average, or the diffusion transform. The oracle query will negate the amplitude of the state $|x_0\rangle$, in this case $|011\rangle$, giving the configuration:

$$|x\rangle = \frac{1}{2\sqrt{2}}|000\rangle + \frac{1}{2\sqrt{2}}|001\rangle + \frac{1}{2\sqrt{2}}|010\rangle - \frac{1}{2\sqrt{2}}|011\rangle + \cdots + \frac{1}{2\sqrt{2}}|111\rangle$$

And the geometric representation is



Now, perform the diffusion transform $2\,|\psi\rangle\langle\,\psi| - I$ which will increase the amplitudes by their difference from the average, and decreasing if the difference is negative:

$$[2\,|\psi\rangle\langle\,\psi| - I]\,|\psi\rangle$$

$$= [2\,|\psi\rangle\langle\,\psi| - I]\;\left[|\psi\rangle - \frac{2}{2\sqrt{2}}|011\rangle\right]$$

$$= 2|\psi\rangle\;\langle\psi|\psi\rangle - |\psi\rangle - \frac{2}{\sqrt{2}}|\psi\rangle\,\langle\psi\,|011\rangle + \frac{1}{2\sqrt{2}}|011\rangle$$

Note that $\langle\psi|\psi\rangle = 8\frac{1}{2\sqrt{2}}\left[\frac{1}{2\sqrt{2}}\right] = 1$.

Additionally, since $|011\rangle$ is one of the basis vectors, we can use the identity $\langle\psi|011\rangle = \langle 011|\psi\rangle = \frac{1}{2\sqrt{2}}$:

$$= 2|\psi\rangle\; - |\psi\rangle - \frac{2}{\sqrt{2}}\left(\frac{1}{2\sqrt{2}}\right)|\psi\rangle\; + \frac{1}{\sqrt{2}}|011\rangle$$

$$= |\psi\rangle - \frac{1}{2}|\psi\rangle + \frac{1}{\sqrt{2}}|011\rangle$$

$$= \frac{1}{2}|\psi\rangle + \frac{1}{\sqrt{2}}|011\rangle$$

Substituting from Equation (4.1) gives

$$= \frac{1}{2}\left[\frac{1}{2\sqrt{2}}\sum_{x=0}^{7}|x\rangle\right] + \frac{1}{\sqrt{2}}|011\rangle$$

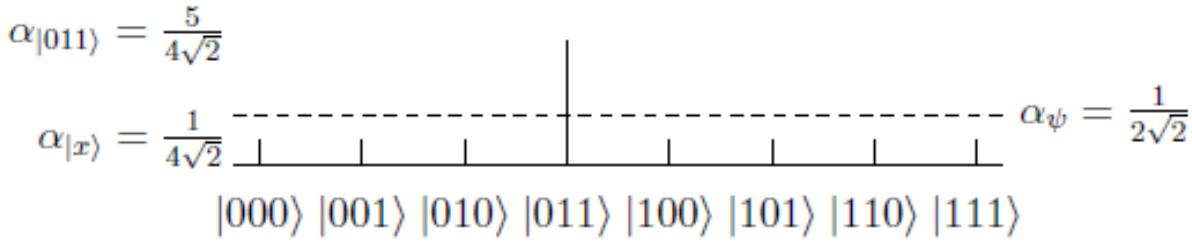
$$= \frac{1}{4\sqrt{2}}\sum_{\substack{x=0\\x\neq 3}}^{7}|x\rangle\; + \frac{1}{4\sqrt{2}}|011\rangle + \frac{1}{\sqrt{2}}|011\rangle$$

$$= \frac{1}{4\sqrt{2}}\sum_{\substack{x=0\\x\neq 3}}^{7}|x\rangle\; + \frac{5}{4\sqrt{2}}|011\rangle$$

In the notation used earlier:

$$|x\rangle = \frac{1}{4\sqrt{2}}\,|000\rangle + \frac{1}{4\sqrt{2}}\,|001\rangle + \frac{1}{4\sqrt{2}}\,|010\rangle + \frac{5}{4\sqrt{2}}\,|011\rangle + \ \ldots + \frac{1}{4\sqrt{2}}\,|111\rangle$$

This appears geometrically as



This completes the first iteration. We apply the same two transformations in the second iteration, yield:

$$|x\rangle = \frac{1}{4\sqrt{2}}\,|000\rangle + \frac{1}{4\sqrt{2}}\,|001\rangle + \frac{1}{4\sqrt{2}}\,|010\rangle - \frac{5}{4\sqrt{2}}\,|011\rangle + \ \ldots + \frac{1}{4\sqrt{2}}\,|111\rangle$$

$$= \frac{1}{4\sqrt{2}} \sum_{\substack{x=0 \\ x\neq 3}}^{7} |x\rangle - \frac{5}{4\sqrt{2}}\,|011\rangle$$

$$= \frac{1}{4\sqrt{2}} \sum_{x=0}^{7} |x\rangle - \frac{6}{4\sqrt{2}}\,|011\rangle$$

$$= \frac{1}{2}\,|\psi\rangle - \frac{3}{2\sqrt{2}}\,|011\rangle$$

After the oracle query, and after applying the diffusion transform we get

$$= [2\,|\psi\rangle\langle\,\psi| - I]\;\left[\frac{1}{2}|\psi\rangle - \frac{3}{2\sqrt{2}}|011\rangle\right]$$

$$= 2\left(\frac{1}{2}\right)|\psi\rangle\,\langle\psi|\psi\rangle - \frac{1}{2}|\psi\rangle - 2\left(\frac{3}{2\sqrt{2}}\right)|\psi\rangle\,\langle\psi\,|011\rangle + \frac{3}{2\sqrt{2}}|011\rangle$$

$$= \;|\psi\rangle\; - \frac{1}{2}|\psi\rangle - \frac{3}{\sqrt{2}}\left(\frac{1}{2\sqrt{2}}\right)|\psi\rangle + \frac{3}{2\sqrt{2}}|011\rangle$$

$$= \frac{-1}{4}\;|\psi\rangle + \frac{3}{2\sqrt{2}}\;|011\rangle$$

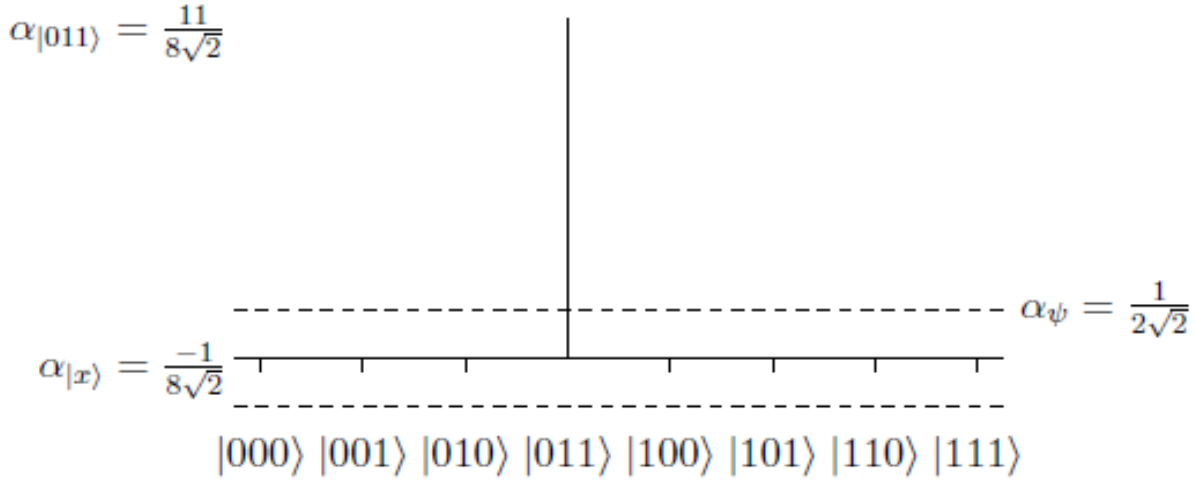
$$= \frac{-1}{4}\left[\frac{1}{2\sqrt{2}}\sum_{\substack{x=0\\x\neq 3}}^{7}|x\rangle + \frac{1}{2\sqrt{2}}|011\rangle\right] + \frac{3}{2\sqrt{2}}\;|011\rangle$$

$$= \;\frac{-1}{8\sqrt{2}}\sum_{\substack{x=0\\x\neq 3}}^{7}|x\rangle + \frac{11}{8\sqrt{2}}|011\rangle$$

Or in the expanded notation:

$$|x\rangle = \frac{1}{8\sqrt{2}}\;|000\rangle - \frac{1}{8\sqrt{2}}\;|001\rangle - \frac{1}{8\sqrt{2}}\;|010\rangle + \frac{11}{8\sqrt{2}}\;|011\rangle - \;\ldots - \frac{1}{8\sqrt{2}}\;|111\rangle$$

Geometrically, the success of the algorithm is clear:



Now when the system is measured, the probability that the state representative of the correct solution $|011\rangle$ will be measured is $\left|\frac{11}{8\sqrt{2}}\right|^2 = \frac{121}{128} \approx 94.5\%$. The probability of finding an incorrect state is $\left|\frac{-\sqrt{7}}{8\sqrt{2}}\right|^2 = \frac{7}{128} \approx 5.5\%$ (Emma, 2011).

4.3 Quantum Fourier Transform

Consider there is quantum circuit acting on n -qubits, by applying a Hadamard gate (unitary transformation) to each qubit, the unitary transformation $H^{\otimes n}$, or H tensored up with itself n times. The unitary transformation H_{2^n}can be define as the $2^n \times 2^n$ matrix in which the (x, y) entry is $2^{-n/2}(-1)^{x \cdot y}$, which is also known as (Fourier transform over Z_2^n) . Applying this unitary transformation to the state of all zeros gives an equal superposition over all 2^n states:

$$H_{2^n}|0 \ldots 0\rangle = \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle$$

In general, applying the $Hadamard$ transform to the basis state $|u\rangle$ yields:

$$H_{2^n}|u\rangle = \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} (-1)^{u\cdot x}|x\rangle$$

(Umesh, 2012).

The quantum Fourier transform (QFT) is defined to be a linear transformation on n -qubits that maps the basis states $|j\rangle,\ j = 0, 1, \ldots, 2^n - 1$, to superposition states as follows:

$$|j\rangle = \rightarrow \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{2\pi ijk/2^n}\,|k\rangle, \qquad i = \sqrt{-1}$$

Where $|j\rangle$ is a number of the orthonormal basis, the inverse of quantum Fourier transforms is given by:

$$|k\rangle = \rightarrow \frac{1}{\sqrt{2^n}} \sum_{j=0}^{2^n-1} e^{-2\pi ijk/2^n}\,|j\rangle, \qquad i = \sqrt{-1}$$

(Yazhen, 2012).

Quantum Fourier transform (QFT) is unitary, it preserves the inner product by the definition,

$\langle j|QFT^{\dagger}QFT|i\rangle =$

$$= \frac{1}{\sqrt{N}}\frac{1}{\sqrt{N}} \sum_{k=1}^{N} e^{-\frac{2\pi ljk}{N}} \sum_{k=1}^{N} e^{\frac{2\pi lik}{N}}\ \langle k|k\rangle$$

$$= \frac{1}{N} \sum_{\mathrm{k}=1}^{N} e^{2\pi lk(i-j)/N}$$

This is just a geometric series, whose sum is given by:

$$\frac{1 - e^{(2\pi lk(i-j))}}{1 - e^{(2\pi lk(i-j)/N)}} = \ \delta_{ij}$$

(Aaron, 2012).

4.4 Shor's Factoring Algorithm

Shor's algorithm is a quantum algorithm for factoring a number N in $O((log\, N)^3)$ time and $O(log\, N)$ space, Shor's algorithm is probabilistic, it gives the correct answer with high probability, and the probability of failure can be decreased by repeating the algorithm (S.A. Duplij, 2007). This algorithm uses quantum Fourier transform to find the factors of a large number. The basic steps of the algorithm are shown:

1. Initialize a first register of $n = 2\, log\, N$ bits to $|0\rangle \otimes ... \otimes |0\rangle \equiv |0\rangle$and a second register of $m = 2\, log\, N$ bits to $|0\rangle \otimes ... \otimes |1\rangle \equiv |1\rangle$
2. Apply a Hadamard gate to the first n qubits, so that the first register reaches

$$\sum_{x=0}^{2^n+1} \frac{|x\rangle}{\sqrt{2^n}}$$

3. Multiply the second register by $f(x) = a^x \bmod N$ to get.

$$\sum_{x=0}^{2^n+1} \frac{|x\rangle\, |1 \times a^x mod N\rangle}{\sqrt{2^n}}$$

   Since the first register is in a superposition of 2^n terms$|x\rangle$, the modular exponentiation is computed for 2^n values of x in parallel.
4. Perform the inverse QFT on the first register, giving

$$|\psi_3\rangle = \sum_{y=0}^{2^n-1} \sum_{x=0}^{2^n-1} e^{2i\pi xy/2^n}\, |y\rangle |a^x mod N\rangle / 2^n$$

5. Measure the qubits in the first register.

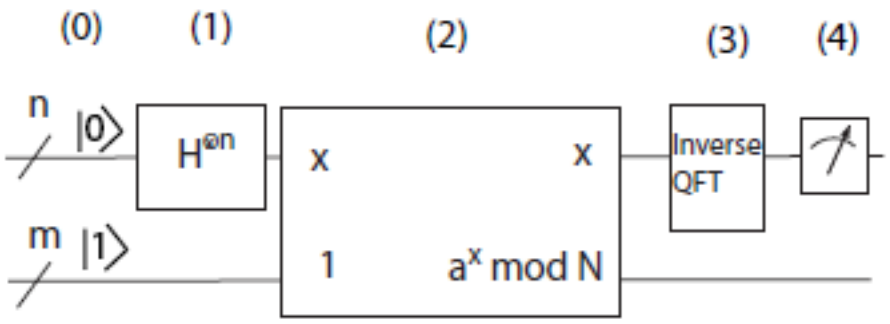


Figure (4.2): Circuit diagram to implement Shor's algorithm. 0) Initialize register one to the$|0\rangle$, and register two to the$|1\rangle$ state. 1) Apply a Hadamard gate on the first register of qubits to gives $\sum_{x=0}^{2^n+1} \frac{|x\rangle}{\sqrt{2^n}}$. 2) Multiply the second register by $f(x) = a^x \bmod N$ to get $\sum_{x=0}^{2^n+1} \frac{|x\rangle\,|1\times a^x mod N\rangle}{\sqrt{2^n}}$. 3) Perform the inverse QFT. 4) Measurement (Kathy-Anne, 2007).

Worked examples

Let see how knowing the period of a number $mod\ N$ can be used to factor N. Suppose we want to factor $N = pq$. We simply pick a random a which is co-prime to N. We then use Shor to compute the order of this a $(mod\ N)$. If this r is odd, we pick a new$a$. If it's even then we can calculate,

$$x = a^{r/2} (mod\ N)$$

Then we have,

$$x^2 \equiv 1\ (mod\ N)$$

This gives us,

$$(x-1)\ (x+1) \equiv 0\ (mod\ N)$$

We know $(x-1) = a^{\frac{r}{2}} - 1 \neq 0$, since, r is the smallest power of a for which $a^r \equiv 1\ (mod\ N)$, we check $x + 1 = a^{r/2} + 1$ is equal to zero or not$(mod\ N)$. If it is, we have been unlucky and we start over with a new a . If it is not, then we can continue. Since $N \nmid (x+1)$ and $N \nmid (x-1)$ but $N\,|(x-1)(x+1)$, then it must mean that $p|\,(x-1)$ and $q|\,(x+1)$.

Then it only remains to compute $gcd\ (N, x+1)$ and$gcd(N, x-1)$, which will yield p and q (Abdullah, 2011).

Period finding through Shor's algorithm

We need two registers of qubits to work with. They will be called the input and output registers respectively. The output register will be made of $n_0 = log\, N$ qubits. The input register will contain twice this number, $n = 2n_0$. We start with all our registers in the state$|0\rangle$:

$$|\psi_0\rangle = |0\rangle_{\otimes n}|0\rangle_{\otimes n_0}$$

We apply the Quantum Fourier Transform (QFT) to the first register. QFT transforms a given vector from one representation to another. Essentially, it transforms a given vector from one representation to another. The QFT is applied in the following way

$$\hat{F}|x\rangle_{\otimes n} = 1/2^{n/2} \sum_{y=0}^{2^n-1} e^{2\pi i \frac{x}{2^n} y}\, |y\rangle_{\otimes_n}$$

There is also an inverse transform QFT^{-1} that gives

$$\hat{F}^{-1} \frac{1}{2^{\frac{n}{2}}} \sum_{y=0}^{2^n-1} e^{2\pi i \frac{x}{2^n} y}\, |y\rangle = |x\rangle$$

In our case all the $x = 0$. So the exponential factor is always 1.

$$|\psi_1\rangle = \left(\, \hat{F}|0\rangle_{\otimes n}\, \right)|0\rangle_{\otimes n_0}$$

$$= 1/2^{n/2} \sum_{x=0}^{2^n-1} |x\rangle_{\otimes n}\, |0\rangle_{\otimes n_0}$$

Now we define $f(x) = b^x (mod\ N)$ and implement a quantum gate that implements this. More specifically, if we have a state like $|x\rangle\,|0\rangle$ then the quantum gate, $\hat{F}$ acts on this state to give $\hat{F}|x\rangle\,|0\rangle = |x\rangle\,|f(x)\rangle$.

This applied to our registers yield

$$|\psi_2\rangle = \hat{F}(x)\,|\psi_1\rangle$$

$$= 1/2^{n/2} \sum_{x=0}^{2^n-1} |x\rangle_{\otimes n}\,|f(x)\rangle_{\otimes n_0}$$

At this point we make a measurement of the output register. We get out some $f_0(x_r)$ where $x_r = x_0 + kr$ here $x_0$is the smallest value of x for which $f(x_0) = f(x_r)$. This is useful because of the measurement postulate the input register too is affected. Its state too changes and our overall state becomes

$$|\psi_3\rangle = \frac{1}{\sqrt{m}} \sum_{k=0}^{m-1} |x_0 + kr\rangle_{\otimes n}\,|f(x)\rangle_{\otimes n_0}$$

Here m is the smallest integer such that $x_0 + mr \geq 2^n$. At this point we don't really care about the output state, so from this point on we will only write down the input state.

$$|\psi_4\rangle = \frac{1}{\sqrt{m}} \sum_{k=0}^{m-1} |x_0 + kr\rangle_{\otimes n}$$

We now apply QFT again to our input register.

$$|\psi_5\rangle = \hat{F}\,|\psi_4\rangle$$

$$= \sum_{k=0}^{m-1} \frac{1}{\sqrt{m}} \frac{1}{2^{\frac{n}{2}}} \sum_{y=0}^{2^n-1} e^{2\pi i(x_0+kr)y/2^n} |y\rangle_{\otimes n}$$

We can make a few rearrangements and break up the exponential,

$$= \sum_{y=0}^{2^n-1} \left(e^{2\pi i x_0 y/2^n} \frac{1}{\sqrt{m2^n}} \sum_{k=0}^{m-1} e^{2\pi i k r y/2^n} \right) |y\rangle_{\otimes n}$$

Notice here is a case of an overall phase factor. Every $|y\rangle_{\otimes n}$ has the same factor associated with it, and that factor has its modulus squared equal to 1 i.e. $|e^{2\pi i x_0 y/2^n}|^2 = 1$. Consequently, we can drop it from our state

$$|\psi_6\rangle = \sum_{y=0}^{2^n-1} \left(\frac{1}{\sqrt{m2^n}} \sum_{k=0}^{m-1} e^{2\pi i k r y/2^n} \right) |y\rangle_{\otimes n}$$

We make a measurement of the input register. This yields one of the values of y with probability give by the squared modulus of its coefficient.

$$P(y) = \frac{1}{m2^n} \left| \sum_{k=0}^{m-1} e^{2\pi i k r y/2^n} \right|^2$$

The exponential has its maximum when its argument is of the form $2\pi l$ where l is integer. Classically our exponential will have maximums when y is close to an integer multiple of $2^n/r$. It can be shown that for integer j if $\left| y - \frac{j2^n}{r} \right| < \frac{1}{2}$ Then the probability of obtaining such a y is at least 40%. We can repeat the quantum part efficiently and with near certainty obtain such a y. Notice the above expression can be rewritten as

$$\left| \frac{y}{2^n} - \frac{j}{r} \right| < \frac{1}{2^{n+1}}$$

Since n is large, and we know y and can compute2^n, then $y/2^n$ is a good estimate of j/r. This is not as good as finding out r itself, but it's very close. There is a low chance that any two random numbers would have a common factor. If j and r don't have a common factor then r is simply the numerator. If not we can either use a classical computer to compute multiples of the numerator and find out which one of them is r. Or we can run the quantum part all over again and get a new j/r. To check if we have the right r we can simply compute $b^r (mod\ N)$. If it's equal to 1, we have our period (Abdullah, 2011).

4.5 Error correction

The basic types model of errors appears on quantum information are Pauli operator $\sigma_x\sigma_y\sigma_z$ (Daniel, 1997), the bit–flip errors corresponds to the Pauli matrix σ_x, exchanging the states $|0\rangle$ and $|1\rangle$, phase–flip errors changing the relative phase of $|0\rangle$ and $|1\rangle$ by π, and their combination. The phase–flip error to σ_z, and their combination to σ_y. It is sufficient to consider only bit-flip error to know about how error is corrected using quantum algorithm (Markus, et al., 1999). A single q-bit becomes entangled with environment and thus (decohere), the main idea is to fight decoherence (or entanglement with environment) with more entanglement. To protect a given q-bit

$$|\psi_0\rangle = a|0\rangle + b|1\rangle$$

one encodes it as a maximally entangled triplet of bits

$$a|000\rangle + b|111\rangle$$

which can be done with the following circuit in Figure (4.3)

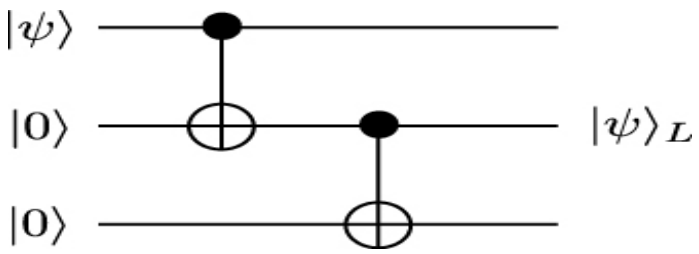


Figure (4.3): In coding circuit.

Shore's bit-flip code

Consider the following circuit as in Figure (4.3). To demonstrate how it works let us write the state of TMR (triple module redundancy) in classical error correcting codes. Bits written as (first, second and third):

$$a|z_1z_2z_3\rangle + b|\bar{z}_1\bar{z}_2\bar{z}_3\rangle$$

then the states of the auxiliary bit (fourth and fifth) is given by

$$|z_1 \oplus z_2, z_1 \oplus z_3\rangle$$

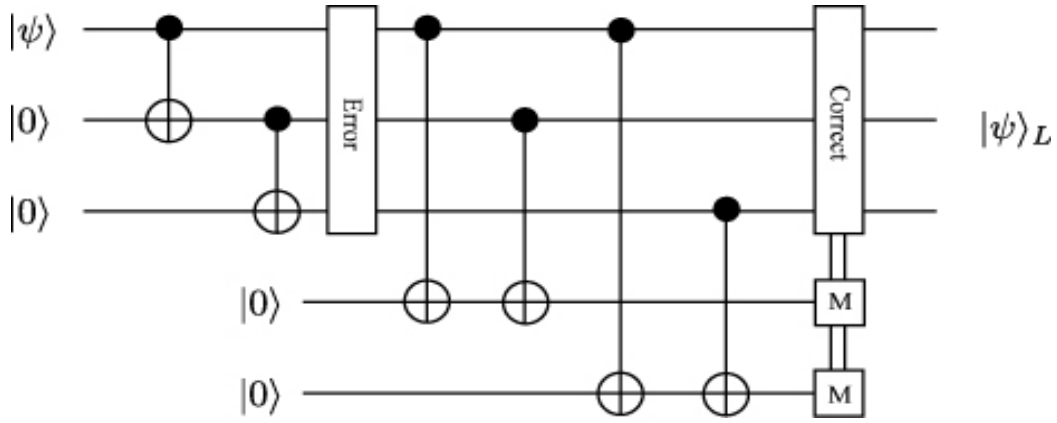


Figure (4.3) circuit implement shore's bit flip code

which can be written as logical table A as shown below in Table (4.1).

Table (4.1)

z_1	z_2	z_3	$z_1 \oplus z_2$	$z_1 \oplus z_3$
⇓	⇓	⇓	⇓	⇓
0	0	0	0	0
0	0	1	0	1
0	1	0	1	0
0	1	1	1	1
1	0	0	1	1
1	0	1	1	0
1	1	0	0	1
1	1	1	0	0

By analyzing the logic table A, (but without actually measuring the TMR bits) one can deduce the following as in logic table B shown in Table (4.2).

Table (4.2)

		$z_1 \oplus z_2$	$z_1 \oplus z_3$
–		⇓	⇓
$No\ Error$	$\longrightarrow$	0	0
$bit\ 3\ flipped$	$\longrightarrow$	0	1
$bit\ 2\ flipped$	$\longrightarrow$	1	0
$bit\ 1\ flipped$	$\longrightarrow$	1	1

For example let say the third TMR bit was randomly rotated with matrix,

$$R_x(\theta) = \begin{pmatrix} cos\frac{\theta}{2} & -i \cdot sin\frac{\theta}{2} \\ -i \cdot sin\frac{\theta}{2} & cos\frac{\theta}{2} \end{pmatrix}$$

Then

$$\begin{array}{rcccc} initial\ state & \dashrightarrow & a|00000\rangle & + & b|10000\rangle \\ after\ 1st\ CNOT & \dashrightarrow & a|00000\rangle & + & b|11000\rangle \\ after\ 2nd\ CNOT & \dashrightarrow & a|00000\rangle & + & b|11100\rangle \end{array}$$

$$\begin{array}{rcccccc} after\ ERROR & \rightarrow & a\ cos\frac{\theta}{2}|00000\rangle - & ai\ sin\frac{\theta}{2}|00100\rangle - & bi\ sin\frac{\theta}{2}|11000\rangle + & b\ cos\frac{\theta}{2}|11100\rangle \\ after\ 3rd\ CNOT & \rightarrow & a\ cos\frac{\theta}{2}|00000\rangle - & ai\ sin\frac{\theta}{2}|00100\rangle - & bi\ sin\frac{\theta}{2}|11010\rangle + & b\ cos\frac{\theta}{2}|11110\rangle \\ after\ 4th\ CNOT & \rightarrow & a\ cos\frac{\theta}{2}|00000\rangle - & ai\ sin\frac{\theta}{2}|00100\rangle - & bi\ sin\frac{\theta}{2}|11000\rangle + & b\ cos\frac{\theta}{2}|11100\rangle \\ after\ 5th\ CNOT & \rightarrow & a\ cos\frac{\theta}{2}|00000\rangle - & ai\ sin\frac{\theta}{2}|00100\rangle - & bi\ sin\frac{\theta}{2}|11001\rangle + & b\ cos\frac{\theta}{2}|11101\rangle \\ after\ 6th\ CNOT & \rightarrow & a\ cos\frac{\theta}{2}|00000\rangle - & ai\ sin\frac{\theta}{2}|00101\rangle - & bi\ sin\frac{\theta}{2}|11001\rangle + & b\ cos\frac{\theta}{2}|11100\rangle \end{array}$$

We can now measure the auxiliary q-bits. The forth bit would always have value 0, but for the fifth one will have two possible outcomes with relative probabilities

$$p(0) = a^2\ cos^2\frac{\theta}{2} + b^2\ cos^2\frac{\theta}{2} = cos^2\frac{\theta}{2}$$

$$p(1) = a^2\ sin^2\frac{\theta}{2} + b^2\ sin^2\frac{\theta}{2} = sin^2\frac{\theta}{2}$$

but the final state of the TMR bits is either

$$\frac{a\ cos\frac{\theta}{2}|000\rangle + b\,cos\frac{\theta}{2}|111\rangle}{\sqrt{cos^2\frac{\theta}{2}}} \quad = \quad a\,|000\rangle + b|111\rangle$$

(Which is a desired final state) or

$$\frac{-ai\ \ sin\frac{\theta}{2}|001\rangle - bi\,sin\frac{\theta}{2}|110\rangle}{\sqrt{sin^2\frac{\theta}{2}}} \quad = \quad -i(a\,|001\rangle + b|110\rangle)$$

which can be corrected with a Control_Not gate acting on the third bit (with fifth bit being the controlled bit) (Vitaly, 2015).

CHAPTER V

DISCUSSIONS AND CONCLUSIONS

5.1 Discussions

A bit-flip error might occurs in any stage of the computational process, as it appears it can be correct easily and effectively through applying a correcting code as in the previous chapter. But the problems of decohernce still exist due to other source of errors and noises, such as loss and leakage, in additional to problem of measurements.

Current problems in quantum computation seems to be daunting, the biggest challenge is to isolate the quantum state to prevent particles representing qubits to interact with the external environment, which disturbs the quantum state and causes it to decohere, and even if we could isolate the system from interact to the environment, quantum gates cannot be implemented with perfect accuracy and the effects of small imperfections in the gates will accumulate, leading to an eventual failure of the computation. In additional quantum algorithms require many gates to be applied on many quantum bits, in order to keep the probability of error low enough; and there is a correcting code for each type of error, adding error correction codes to quantum algorithms increase the number of qubits required to provide the necessary redundancy to recover from errors, and the number of quantum gates needed to process the redundantly encoded data, and to diagnose and reverse the errors, this increases the likelihood of error, and mitigates the effect of decoherence, which makes correcting codes entail an enormous overhead in a quantum computation.

In other hand one might also try to implement the repetition code quantum mechanically by duplicating the quantum state three or more times, this is forbidden by the no cloning theorem, even if cloning were possible, it would not be possible to measure and compare the three

quantum states output from the channel. Error correction substantially more difficult in the quantum world, due to many reasons, for instance and in generally measurement that test whether a state is correct or not, can collapses and destroys the quantum state under observation, this in turn makes recovery of quantum information impossible. Furthermore we must not ignore that the correction and recovery procedure itself can introduce new errors, thus new ideas and techniques need to be introduced to make quantum error correcting codes possible.

5.2 Recommendations

Some suggestions for future boost of practical quantum computation

- Dealing with decoherence comes from the algorithmic rather than the physical side, a fast and robust algorithm can be found and develop by exploiting properties of the problem structure itself, this is important for the development of practical quantum computers. Be able to construct an arbitrary unitary gate on a potentially arbitrary number of quantum bits using a universal set of elementary gates is required for a successful quantum device. Various gates need to be applied fast and precisely enough, to allow quantum error correction to succeed. Also we need to know how to efficiently perform encoding and decoding, beside any reasonable correction scheme must thus protect against small unitary errors in the quantum gates as well as against decoherence. Quantum error correction is similar to classical error correcting codes, with considerable differences between the classical and quantum states. However, classical techniques can be modified to work for quantum systems.

- Fault-tolerant methods are significant to improve the reliability of a quantum computer; the idea is to keep errors under control in such a way that regular phases of error-correction don't get overwhelmed by the errors. When designing schemes for fault-tolerant computing, it is very important to ensure that errors don't spread quickly. Another potentially and more robust model is the model of adiabatic quantum

optimization, where computation is achieved by adiabatically tuning a set of Hamiltonians, to keep the system always in the instantaneous ground state, by creating a gap between the ground state and the first exited state at all times, which might make the state more robust to noise.

REFERENCES

A. M. Steane. (1998) Introduction to quantum error correction. Clarendon Laboratory. Oxford OX1 3NP, UK, Phil. Trans. R. Soc. Lond. A (1998) 356, 1739-1758, the Royal Society.

Andrea Del Duce. (2009) Quantum Logic Circuits for Solid-State Quantum Information Processing. A thesis submitted for the degree of Doctor of Philosophy. University College London**.**

Aaron Krahn. (2012) Quantum Computation and Grover's Algorithm. Available from: math.uchicago.edu/~may/ REU2012/REUPapers/Krahn.pdf. [Accessed 25th March 2017].

Abdullah Khalid. (2011) A Gentle Introduction to Quantum Computing. 2012-10-0168. School of Science and Engineering. Lahore University of Management Sciences.

Daniel Gottesman. (1997) Stabilizer Codes and Quantum Error Correction. Thesis Submitted for the Degree of Doctor of Philosophy. California Institute of Technology Pasadena. California. 2004.

Eleanor Rieffel. (2008) Quantum Computing. FX Palo Alto Laboratory. Available from: quant-ph/0804.22v2 [Accessed 25th March 2017].

Emma Strubell. (2011) An Introduction to Quantum Algorithms. COS498-Chawathe. Available from: http://www.scottaaronson.com/ [Accessed 25th March 2017].

Gennady P Berman, Gary D Doolen, Ronnie Mainieri and Vladimir I Tsifrinovich. (1999) Introduction to Quantum Computers. Published by World Scientific Publishing Co. Pte. Ltd. P 0 Box 128. Singapore 912805.

Javier Enciso. (2009) Introduction to Quantum Computing.

Jozef Gruska. (2011) Quantum Computing. Available from: http://mcgraw-hill.co.uk/gruska [Accessed 25th March 2017].

John Watrous. (2011) An introduction to quantum information and quantum circuits. Institute for Quantum Computing. University of Waterloo.

Julia Kempe. (2005) Approaches to Quantum Error Correction. Vol. 1. Université de Paris-Sud 91405 Orsay Cedex France. Séminaire Poincaré 1 (2005) 65 -93.

Jill Cirasella. (2006) Classical and Quantum Algorithms for Finding Cycles. MSc Thesis submitted for the degree of MSc in Logic. Universiteit van Amsterdam.

Joey Allcock. (2010) Emulating Circuit-Based and Measurement-Based Quantum Computation. MEng4 Individual . project Final Report. Department of Computing, Imperial College London.

Kathy-Anne Brickman. (2007) Implementation of Grover's Quantum Search Algorithm with Two Trapped Cadmium Ions. A dissertation submitted for the degree of Doctor of Philosophy (Physics). University of Michigan.

Lars Steffen. (2013) Quantum Teleportation and Efficient Process Verification with Superconducting Circuits. A dissertation submitted for the degree of Doctor of Sciences. ETH Zurich.

Michael Keyl. (2002) Fundamentals of quantum information theory. Mendelssohnstraβe. 3, D-38106, Braunschweig, Germany.

Magnus Gausdal Find. (2009) Algorithms for Quantum Computers. Bachelor Thesis. Department of Mathematics and Computer Science. University of Southern Denmark, Odense.

Markus Grassl, Willi Geiselmann and Thomas Beth. (1999) Quantum Reed–Solomon Codes. Institut f¨ur Algorithmenund Kognitive Systeme Arbeitsgruppe. Quantum Computing

Universit¨at Karlsruhe. Am Fasanengarten 5, 76 128 Karlsruhe. Germany. Marc Fossorier et al. (Eds.): AAECC-13, LNCS 1719, pp. 231–244.

Matthew David Reed. (2013) Entanglement and Quantum Error Correction with Superconducting Qubits. A Dissertation submitted for the Degree of Doctor of Philosophy. Faculty of the Graduate School. Yale University in Candidacy.

Marek Andrzej Perkowski. (2011) ECE 510 - Quantum Computing. Course. Spring 2011. Available from http://web.cecs.pdx.edu/~mperkows/class_future/new_materials_2011/lukac perkowski book introduction and quantum mechanics.pd. [Accessed 25th March 2017].

Michael Aaron Nielsen. (1998) Quantum Information Theory. Dissertation Submitted for the Degree of Doctor of Philosophy Physics. University of New Mexico. USA.

Moayad A. Fahdil, Ali Foud Al-Azawi, and Sammer Said. (2010) Operations Algorithms on Quantum Computer. IJCSNS International Journal of Computer Science and Network Security. VOL.10 No.1.Available from: paper.ijcsns.org/07_book/201001/20100112.pdf [Accessed 25th March 2017].

Morinosato-Wakamiya, Atsugi Kanagawa 243-0198. Japan. Electronic address: devitt@nii.ac.jp. Available from: quant ph/0905.2794v4 [Accessed 25th March 2017].

Phillip Kaye, Raymond Laflamme and Michele Mosca. (2007) An Introduction to Quantum Computing. ISBN 0-19-857000-7 978-0-19-857000-4. ISBN 0-19-857049-x 978-0-19-857049-3 (pbk).

Robert B. Griffiths. (2012) Quantum Error Correction. Version of 9. qitd213.

Ronald de Wolf. (2016) Quantum Computing. Lecture Notes. Amsterdam. Available from: homepages.cwi.nl/~rdewolf/qcnotes.pdf [Accessed 25th March 2017].

Samuel L. Braunstein. (2003) Quantum computation. University of York. UK. Available from: http://www.doc88.com/p-1367546833756.html [Accessed 25th March 2017].

Seth Lloyd. (2009) Quantum Information Science. Available from : http://web.mit.edu/2.111/www/notes09 [Accessed 25th March 2017].

Simon J. Devitt, William J. Munro and Kae Nemoto. (2013) Quantum Error Correction for Beginners. National Institute of Informatics 2-1-2 Hitotsubashi Chiyoda-ku Tokyo 101-8340. Japan. NTT Basic Research Laboratories, NTT Corporation 3-1.

Sarah Mostame. (2008) On Quantum Simulators and Adiabatic Quantum Algorithms. Dissertation zur Erlangung des akademischen Grades Doctor rerum naturalium. July 1978 in Tabriz, Iran, Dresden 2008.

S.A. Duplij, I.I. Shapoval. (2007) Quantum computations: fundamentals and Algorithms. Institute of Physics and Technology, Kharkiv. Ukraine. Problems of Atomic Science and Technology (PAST).-2007.-No.3 (1).-p.230-235.

T. Venkat Narayana Rao & Shirish Pathania. (2010) The next generation computing brainwave-quantum computing. International Journal of Hybrid Information Technology. Vol.3, No.4.

Tim Reid. (2005) On the Evolutionary Design of Quantum Circuits. A thesis submitted for the degree of Master of Mathematics in Combinatorics and Optimization. Waterloo, Ontario, Canada.

Umesh Vazirani. (2012) C191 Quantum information – EECS. Available from: http://wwwinst.eecs.berkeley.edu/~cs191/sp12/. [Accessed 25th March 2017].

Vitaly Vanchurin. (2015) PHYS4071/5071: Quantum Computation. Available from: d.umn.edu/~vvanchur/2015PHYS4071/Chapter6.pd. [Accessed 25th March 2017].

Wojciech Hubert Zurek. (2013) Decoherence, Einselection, and The Quantum Origins of the Classical. LANL, Mail Stop B288, Los Alamos, New Mexico 87545. Available from: quant-ph/0105127v3 [Accessed 25th March 2017].

Willem Klaas van Dam. (2004) On Quantum Computation Theory. academisch proefschrift ter verkrijging van de graad van doctor aan de Universiteit van Amsterdam. ISBN: 90–5776–091–6, ILLC Dissertation Series 2002-04. Second Version.

Xinlan Zhou. (2002) Quantum logic gate construction and quantum algorithms. A dissertation submitted for the degree of doctor of philosophy. Stanford University.

Yazhen Wang. (2012) Quantum Computation and Quantum Information. Statistical Science Vol. 27, No. 3, 373–394, DOI: 10.1214/11-STS378. Available from: stat.ME/1210.0736v1 2 Oct 2012 [Accessed 25th March 2017].

Yong Zhang and Kun Zhang. (2016) GHZ transform (I): Bell transform and quantum teleportation. School of Physics and Technology. Wuhan University. China. Available from: quant-ph/1401.7009v4 [Accessed 25th March 2017].