Quantum Algorithms

Figen Yılmaz

Submitted to the
Institute of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Master of Science
in
Physics

Eastern Mediterranean University
January 2020
Gazimağusa, North Cyprus

Approval of the Institute of Graduate Studies and Research

Prof. Dr. Ali Hakan Ulusoy
Director

I certify that this thesis satisfies the requirements as a thesis for the degree of Master of Science in Physics.

Prof. Dr. İzzet Sakallı
Chair, Department of Physics

We certify that we have read this thesis and that in our opinion, it is fully adequate, in scope and quality, as a thesis of the degree of Master of Science in Physics

Asst. Prof. Dr. Mustafa Rıza
Supervisor

Examining Committee

1. Prof. Dr. Ayhan Bilsel
2. Prof. Dr. Mustafa Halilsoy
3. Asst. Prof. Dr. Mustafa Rıza

ABSTRACT

This thesis deals with Quantum Computing Algorithms. The most important Quantum Computing algorithms available in the literature have been summarized. After reviewing the foundations of Quantum Computing and the necessary parts of Quantum Mechanics, the quantum search algorithm by Grover was analyzed theoretically, and the simulation on the publicly available real Quantum Computer provided by IBM using the Qiskit package was carried out. The results from the simulation on the real quantum computer agree with theoretically obtained results.

Keywords: Quantum Mechanics, Quantum Algorithms, Grover's Algorithm

ÖZ

Bu tez, kuantum bilgisayarlar ve algoritmaları üzerine bir çalışmadır. En önemli kuantum bilgisayar algoritmaları 'Giriş' bölümünde kısaca anlatılmıştır. Kuantum bilgisayarlar için önemli olan kuantum mekaniğinin önemli postulatları bu tezde anlatılmıştır. Grover tarafından sunulan Grover Algoritması teorik olarak anlatılmış olunup gerçek bir Kauntum Bilgisayar ile denemesi yapılmıştır. Bu tez çalışmasında, IBM'in kuantum bilgisayalarına Qiskit ile bağlanılmış olup sonuçları sunulmuştur ve simulasyon sonuçları da eklenmiştir. Teorik olarak öngörülen sonuçlar halka açık olan IBM bilgisayarları ile doğrulanmıştır.

Anahtar Kelimeler: Kuantum Mekaniği, Kuantum Algoritmaları, Grover Algoritması

DEDICATION

To my music lists and coffees

ACKNOWLEDGMENTS

I know very well how important motivation is in one's academic endeavour. I feel so lucky to be part of a family to which I would like to extend my gratitude. This family is comprised of Prof. Dr. H. Tanju Besler, Asst. Prof. Dr. Erkan Arkın, Dr. İbrahim Avcı, Asst. Prof. Dr. Mustafa Babagil, Asst. Prof. Dr. Hüseyin Etikan and Süheyla Teksu, as well as Havva İlter and Akile Serinkanlı. And also to my lovely friend Hamit Çobanoğlu for reminding all deadlines during thesis process.

I would like to thank to my dear bestie Efecan Ünal and the strong woman-academic figure Hanife Erişen for always being with me and to Chimobi Onwualu and all my lovely universal friends for bearing with me no matter what ...

I am very well aware that this thesis would not be possible without the guidance, help and support of my supervisior, Dr. Mustafa Rıza. I feel privilaged to have worked with him in this study, so my heartfelt gratitude goes to my supervisor. Thank you very much for all you have done for me, Hocam.

Last but not least, I am grateful to my ”shero” sister, my brother and my mom and dad. I love you all.

TABLE OF CONTENTS

ABSTRACT iii

ÖZ iv

DEDICATION v

ACKNOWLEDGMENTS vi

LIST OF TABLES viii

LIST OF FIGURES x

1 INTRODUCTION 1

2 QUANTUM MECHANICS 5

2.1 Superposition 6

2.1.1 Young's Double Slit Experiment 8

2.2 Entanglement 10

3 QUANTUM COMPUTING 14

3.1 Circuit Theory 27

4 GROVER'S ALGORITHM 31

4.1 Worked Example with 3 Qubits 46

5 CONCLUSION 50

REFERENCES 52

APPENDIX 55

LIST OF TABLES

Table 3.1: How X Gate Changes the states. 19

Table 3.2: How CNOT Gate changes the states. 24

Table 3.3: How Toffoli Gate Changes the states................................ 28

Table 3.4: Quantum Gates... 30

LIST OF FIGURES

Figure 2.1: An experiment of light's behavior. 7

Figure 2.2: Light's behavior of one particle 8

Figure 2.3: Acting like wave.. 9

Figure 2.4: When there is an observer 10

Figure 3.1: State of a qubit on the Bloch Sphere 15

Figure 3.2: States of deterministic classical, probabilistic classical, and quantum bits.. 15

Figure 3.3: Bloch sphere representation of the Hadamard operator applied to $|0\rangle$ and $|1\rangle$.. 22

Figure 3.4: Visualization of the Hadamard gate on the Bloch sphere, acting on the input state $(|0\rangle + |1\rangle)/\sqrt{2}$.. 22

Figure 3.5: The representation of CNOT gate of control and target 24

Figure 3.6: The representation of CNOT gate on 3-qubit states.................... 26

Figure 4.1: Circuit diagram for Grover's algorithm, with a scratch qubit for the oracle.. 31

Figure 4.2: The general view of Grover's algorithm as circuit..................... 42

Figure 4.3: The general view of Grover's algorithm as circuit..................... 46

Figure A.1: The general schematic overview of the circuit for Grover's Algorithm 57

Figure A.2: All circuits of Grover's Algorithm for 2-qubits........................ 57

Figure A.3: The first step is 'Initialisation'.. 58

Figure A.4: The Oracle showed by dashed rectangle 59

Figure A.5: The Grover diffusion gate... 59

Figure A.6: The measurements are performed on first and second qubits 59

Figure A.7: Whole Grover’s circuit with 3-qubits……………………………….. 60

Figure A.8: IBM Quantum simulator for $|11\rangle$ state 61

Figure A.9: The result of $|11\rangle$ state from IBM Quantum simulator 61

Figure A.10: IBM Quantum simulator for $|10\rangle$ state 61

Figure A.11: The result of $|10\rangle$ state from IBM Quantum simulator 61

Chapter 1

INTRODUCTION

A computer is a physical machine and it computes all operations performed by such a machine is in aspect a physical process. Nowadays, computers are classified generally in two types as classical or quantum computers. Laws of computations are based on the laws of physics.

In 1930s, Alan Turing, who is an English logician and mathematician and who obeyed to physics laws and assumed that computation is performed by an idealized mechanical computer, formulated the classical theory of computation [19]. Now, this model is known as Turing Model and this has proved to have enough capability to the description of computational operations by modern electronic or mechanical computers.

Here is some explanation about a new type of computers, called Quantum Computers. Quantum Computers have the ability of storing information, loading and running programs and reading output by the laws of Quantum Physics. This type of computers works differently than classical computers. In this respect, we can compare the difference between classical computers and quantum computers to the difference between classical physics and quantum physics. Considering the history of quantum computing, it dates back to 1982, when Richard P. Feynman suggested the idea at a conference [9].

He suggested the notation of simulating quantum mechanics with computers, and later on it became universal quantum computers. In fact an earlier history of quantum computers dates back to Maxwell's Demon. It is an imaginary demon, which refers to the second law of thermodynamic by part of separation of cold and hot particles through opening an imaginary gate. Later on, Bennett, Fredkin, Toffoli and others proposed the idea about reversible operations to general computation in the 1970s. They showed that without erasing information, all computation can be reversible [3].

Much earlier, in 1935, Einstein, Podolsky and Rosen experimented with entanglement [8]. The Quantum Theory can define the state of one particle out of two which affect each other [3] (see following chapter).

Later, Bell came out with an acute result in 1964, showing that non-local interactions can exist [1]. This theorem was supported by Aspect, Dalibard and Roger in 1982. They showed which particles are in entanglement state and argued that their interactions must travel faster than speed of light and therefore it is impossible [3].

The real developments of quantum computers took place with Benioff, in 1980 [2]. He described one hybrid Turing machine which one can store qubits on the tape instead of traditional bits. Nevertheless, his machine could not use any quantum theory effects and for that reason each qubit was measured from the tape at each step.

A real breakthrough came with Feynman's speech at a physics conference in 1982 [9]. In his speech, he clearly discussed the architecture of a machine that would operate on quantum mechanical laws. He also spoke about the universal quantum simulator. The machine would use quantum properties to explore the rest of quantum effects and run simulators.

The first primitive quantum computing system was credited by Deutsch in 1985 [4]. However, there were some doubts about his system like including prohibition for the gates and operational states of his machine.

Simon explained an oracle problem in 1993, and then the difference between quantum and classic computers emerged with having the exponential fast [17]. Simon's algorithm solves the black-box problems exponentially faster than classical ones as Deutsch-Jozsa algorithm does [5]. And in 1994 [16], Shor described his quantum algorithm for an efficient factorization of large numbers, which drew attention on quantum computers. Shor's algorithm solves integer factorization problems in polynomial time where classical algorithms run in super polynomial time. Shor improved the algorithm from Simon's algorithm.

Meanwhile, Weisner and Bennett explored and introduced the idea of quantum key exchange in the early 1980's, which enabled security systems to deal with quantum computers by the ability of computational doable factorization [3].

In 1996, Grover described a quantum search algorithm [10]. This algorithm searches a marked entry in an unordered database with N entries. Instead of $\mathscr{O}(N)$ with classical way, it requires only $\mathscr{O}(\sqrt{N})$. But this algorithm does not work exponential speed-up, it works quadratic speed-up. In this thesis we work on Grover's Algorithm.

Finally, in 1998, the first functional two qubit nuclear magnetic resonance computer was introduced at UC Berkeley. Later on, the efficiency and effectiveness of the systems were proved, and in 2001, a 7-qubit NMR system was demonstrated at IBM Almaden to execute Shor's algorithm [3].

In sum, thanks to the aforementioned developments in quantum physics we have the ability to predict probabilities.

My research problem is that how much fast and accurate is the Grover's Algorithm.

Chapter's Survey: In the first chapter is introduction. Chapter two is basic definition of Quantum Mechanics, including superposition, Young's double slit experiment and entanglement. Chapter three is about Quantum Computing including Qubits and Circuit Theory. Chapter four focuses on Grover's Algorithm. And the last chapter presents Conclusion.

Chapter 2

QUANTUM MECHANICS

“Nature isn’t classical, dammit, and if you want to make a simulation of nature, you’d better make it quantum mechanical, and by golly it’s a wonderful problem, because it doesn’t look so easy.” Richard P. Feynman [9]

Like what Feynman said, nature is complicated and that is why it is not easy to explain it with classical physics. Quantum mechanics can help us for simulating the nature. It gives us a quite accurate description of nature. Nature is so complex and quantum mechanics predicts all probabilities of nature. Still, it is not known how it exactly works in that way. Quantum mechanics can tell how it works but not why it works that way. Here are the basic parts of quantum mechanics;

This is the wave function of a particle:

$$i\hbar\frac{\partial\psi}{\partial t} = -\frac{\hbar^2}{2m}\frac{\partial^2\psi}{\partial x^2} + V\psi \tag{2.1}$$

It is called the Schrödinger equation, derived by Erwin Schrödinger in 1925 [15] . Here, $\hbar = h/(2\pi)$. ψ is the wave function, V potential and m mass. Explaining Quantum Mechanics can take many chapters but essentially there is need to explain two main topics that are fundamental for Quantum Computing and Quantum Information. These are:

1) Superposition

2) Entanglement

2.1 Superposition

Superposition is one of the main topics which gives the difference between classical computing and quantum computing. It is the one of strange effects of quantum mechanics that describe a new way of states. Superposition works on photons, phonons, electrons and so on. In short, it means that a particle's quantum state can be at a different state. Here is a short version of superposition explanation with quantum physics. The main idea of superposition can be explained by Figure 2.1. This experiment can be observed in three parts.

First, there is the equipment of this experiment: a light source, semi-silvered mirror (it helps to divide the light) and very sensitive detectors. These detectors are sensitive enough even if an individual light passes through it will catch it. When the light falls on the dimmer, the detector will observe it by giving a click. In the first part of this experiment, there is a semi-silvered mirror placed in front of the light source [3]. The result is the detector 1 and 2 observe the light with the same probability. The question is 'How the light decides which way to go?'

In the second part of this experiment there is something really unusual. When two more full mirrors are placed on the way of lights, signals are gathered from detector 1 only. How would this be possible?

The classical explanation is that light might have predisposed to pass through the mirrors. Or another classical explanation is that different photons may have a path of an object moving through space. And comes the third part of this experiment. The result could be

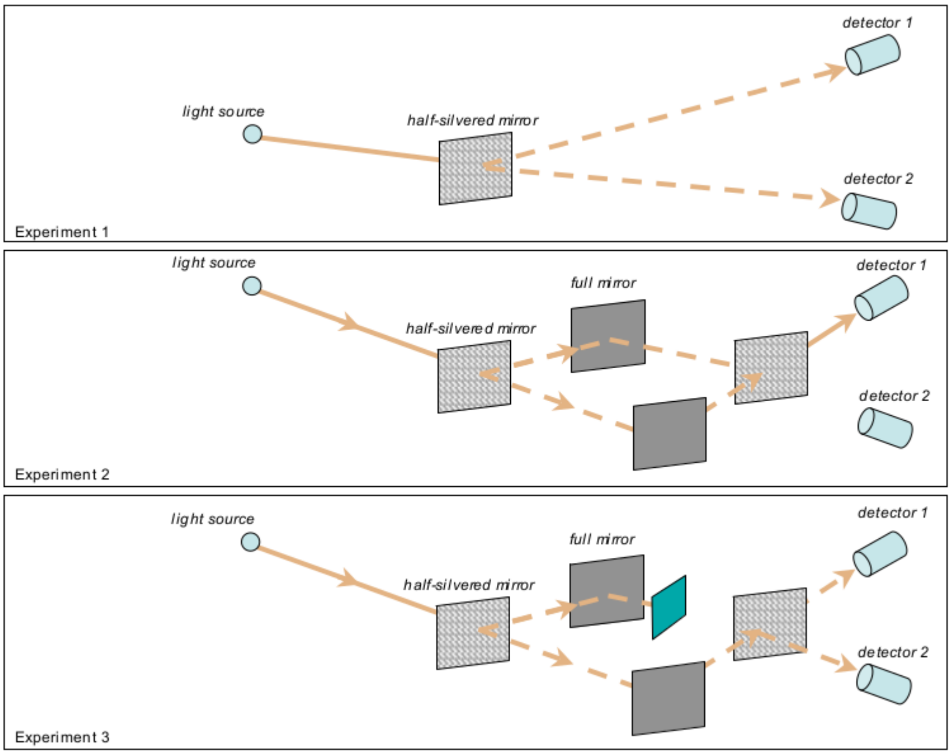


Figure 2.1: An experiment of light's behavior. [3]

estimated by classic physics but it would not work. This part of the experiment can not be explained by any classical way.

Regarding the last part of this experiment, as it can be seen in Figure 2.1, if the way of one light is blocked, light on both detectors with equal probability will be detected.

Surely, this phenomena can not be explained by classical physics. However, quantum mechanics proves helpful to explain and understand this phenomena, which is called superposition. It will be explained in detail in later sections. But the question is how superposition is related with quantum computing.

In classical explanation, one bit is either 0 or 1, but in quantum theory qubits are 0 or 1 and can be both at the same time. Bits are sharply 0 or 1 but qubits are like moving between 0 and 1. Having both states posits many advantages. However, when the

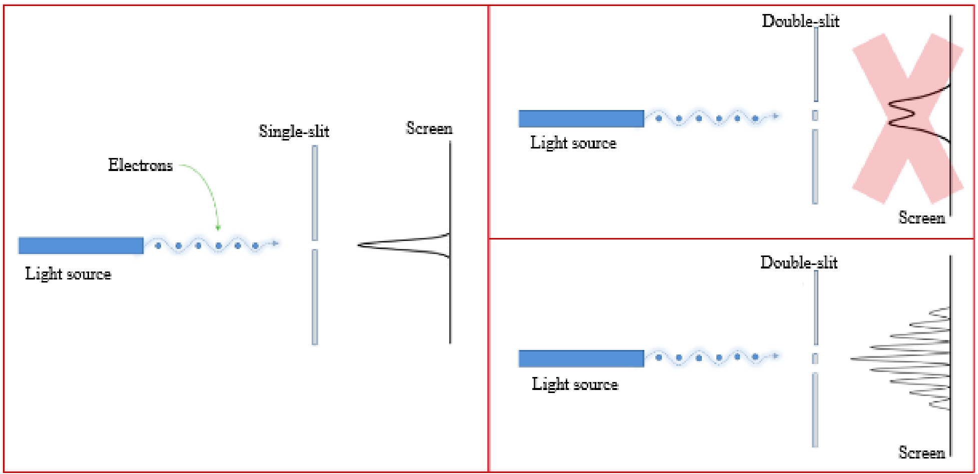


Figure 2.2: Light's behavior of one particle [7].

particle is observed the superposition state is broken and it collapses in one state, 0 or 1 state. Plus, it can not be estimated in which state the particle will collapse. The particle will choose what to do. It is the same with nature of light. As it was explained before, light has two identities: particle and wave. Light decides which identity it is going to have. It depends on the experiment. For example in 'Young's Double Slit Experiment'; it was expected that light would act as a particle but this experiment showed that light also has a wave identity.

A brief explanation would be useful to describe what 'Young's Double Slit Experiment' is in order to answer the questions brought up earlier.

2.1.1 Young's Double Slit Experiment

This experiment is the best way to understand what superposition exactly is. Plus, this experiment is the beginning of the paradox that light is wave or particle in the Modern Physics. In this experiment only a light source, some block which has slit on it and a screen are used. First, the behavior of light acting as particle is observed. There is only one slit on the block and there is one graph on the screen (Figure 2.2). All looks like normal but when the same is applied on double slit block something different is

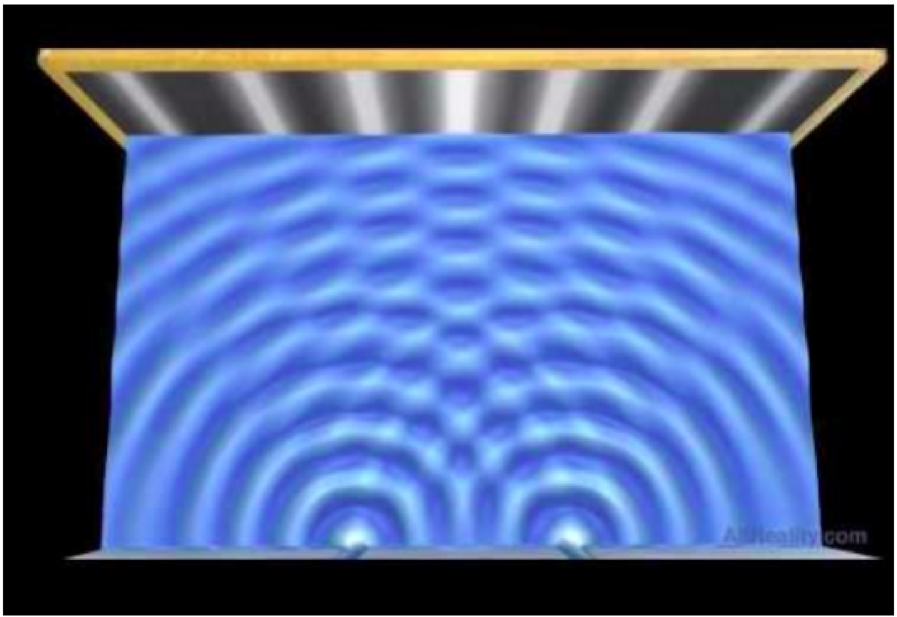


Figure 2.3: Acting like wave [7].

observed on the right side (Figure 2.2). Surprisingly, this result is completely different from what is expected. This result is like the particles are coming from two different sources showing that the light acts as wave. Dark and bright lines can be seen on the screen, (Figure 2.3).

This result is completely the same when the same experiment is done with only one particle. Even if there is only one particle, the results are the same. The question is although there is only one particle and how it interferences or with what it gets interference. The answer of this question is as follows. A particle in a superposition state can be at two different places meaning it can get interference with itself (Figure 2.4). When a sensor is placed in there and when an electron passes through it will send a signal and the observed result is on the screen what is seen is what was expected in the beginning. The light decides to act like a particle now, (Figure 2.4). A new question now is why and how? Is it too shy from sensor?

Actually it has got nothing to do with the sensor's physical place. When the sensor

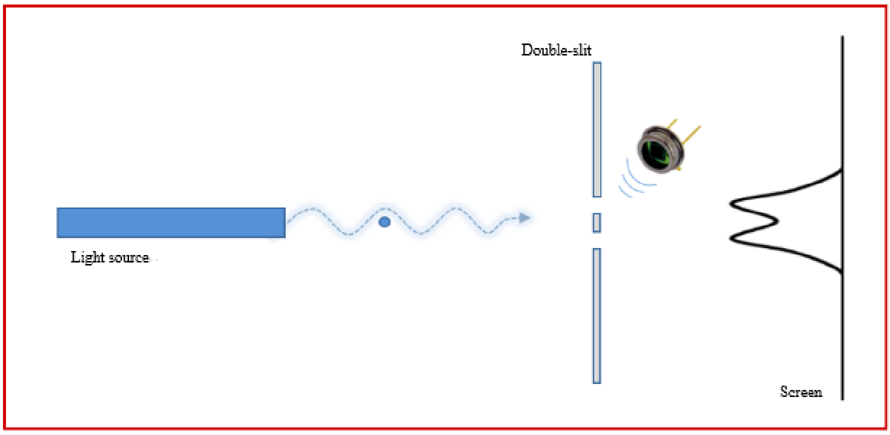


Figure 2.4: When there is an observer [7].

is plugged out again the interference shapes can be seen on the screen. How? The unusual but beautiful reality is when it is observed, the light will collapse in one state. That's why it acts like a particle. Because the properties of being wave is gone now. Another way of explanation is when the experiment is observed, the interference is completely gone. A quantum system behaves gracefully when it is observed than when it is not observed. This is a phenomenon called *particle-wave duality*. This experiment is completely different from others. Some results can be predicted from an observation of other physics experiments, but not any longer after quantum physics. This awkward result is both a blessing and a curse at the same time for the design of quantum computers.

Superposition is one of the mostly needed explanations like Entanglement for Quantum Computers and Quantum Communication. The following section discusses Entanglement.

2.2 Entanglement

Entanglement is the second phenomenon in quantum mechanics which can not be explained by classical theories. Sometimes it is called as quantum entanglement too. Entanglement is an interaction between two or more particles. Their physical properties are entangled together but this can not be explained by classical physics. Considering any two photons, it is not possible to tell anything about the first one's state or the second one's state. This is sort of the same as superposition rule. Their state can not be estimated unless looking at them because they will collapse in one state, entangled particles are both 0 and 1 till they are checked. Plus, it does not matter how much they are far from each other if one of them is observed, information from second one will be gathered at the same time.

This part might be complicated so an explanation can be made using gloves as examples. There is a pair of gloves and they are put in separate boxes and then one box is sent to planet 986F4574E and the second one stays on Earth. Before the box is opened one can not guess which hand is in the box and which one in the one on Earth and only when the box is opened one can understand at the same time which hand was in the other box on the other planet. This might seem like classical physics but the example was given on macroscopic material. Reality is with the particle. And also information was clear in the boxes even when the boxes were not opened. Entanglement is not working with that way because each particle is still 0 and 1.

It does not seem normal but photons affect each other at the same time and Einstein did not like the idea. He said that this caused non-local results. That is why there are some ”secret effects” which are not seen. After a long time John Bell fixed this phenomena in 1960s [1]. He proved that ”secret effect” by math. Right after that Bell-test showed Bell's theory is correct by experiments with Bell-CHSH. This test proved to be working because with that it was possible to observe if the particles are entangled and how strong they are.

In 1935, Einstein, Podolski and Rosen formulated the Bell basis [8].

Bell States: These basis are known as Bell basis and it has four states; $|\beta_{00}\rangle, |\beta_{01}\rangle, |\beta_{10}\rangle, |\beta_{11}\rangle$ Also known EPR pairs or EPR states. [13]

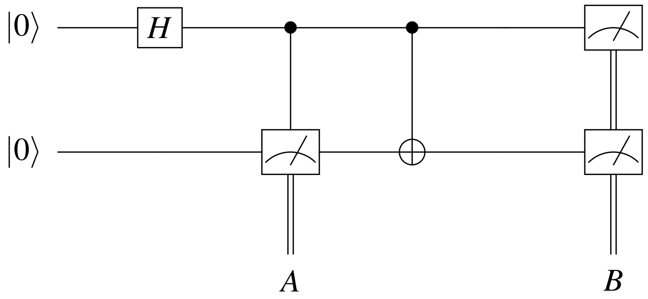
$$|\beta_{00}\rangle = \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle \tag{2.2}$$

$$|\beta_{01}\rangle = \frac{1}{\sqrt{2}}|01\rangle + \frac{1}{\sqrt{2}}|10\rangle \tag{2.3}$$

$$|\beta_{10}\rangle = \frac{1}{\sqrt{2}}|00\rangle - \frac{1}{\sqrt{2}}|11\rangle \tag{2.4}$$

$$|\beta_{11}\rangle = \frac{1}{\sqrt{2}}|01\rangle - \frac{1}{\sqrt{2}}|10\rangle \tag{2.5}$$

The Entanglement State can be created as:



It starts with $|00\rangle$, when it is measurement after Hadamard gate A is the result of that point and B is the result of all circuit.

$$\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)|0\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |10\rangle) \tag{2.6}$$

After Hadamard gate the CNOT gate is observed by: $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$

The details are mentioned in the next chapter.

$A = \frac{1}{\sqrt{2}}(|00\rangle + |10\rangle)$ and $B = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$

After Hadamard gate state is not Bell state. That is why the *CNOT* gate needs to be used. Finally, there is Entanglement state.

$$\Psi = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$

Chapter 3

QUANTUM COMPUTING

The states 0's and 1's can be represented as either with Dirac notation $|0\rangle$ and $|1\rangle$ or matris notation. The 0's and 1's are shown as vector way on Hilbert Space. Computer language is all about 0's and 1's. All math and operations are done with only 0's and 1's. But there is little difference between classical and quantum computers. That small but significant difference is that quantum computers use 0's and 1's at once and, not only thing to do operations on 0's and 1's. Plus, phases, complex numbers and quantum wave interference of them. This was explained in the previous section, 'superposition'.

A qubit can be in any linear combination of states. Qubits are represented in Dirac notation [6] with *ket*.

The representation of a qubit's state is: $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$. Where α, β are complex numbers and normalization coefficients : $|\alpha|^2 + |\beta|^2 = 1$. Here $|\alpha|^2$ is the probability of having 0 state and $|\beta|^2$ is 1 state. For example if $|\alpha|^2 = 0.6$ so, $|\beta|^2 = 0.4$. This means that 60% possibility will collapse in 0 state and 40% in 1 state. As this was mentioned before α and β can be complex numbers as well.

Representation of a qubit on the Bloch Sphere (Figure 3.1).

In Figure 3.2, there is states of classical computers, it is clear to see either 0 or 1. And

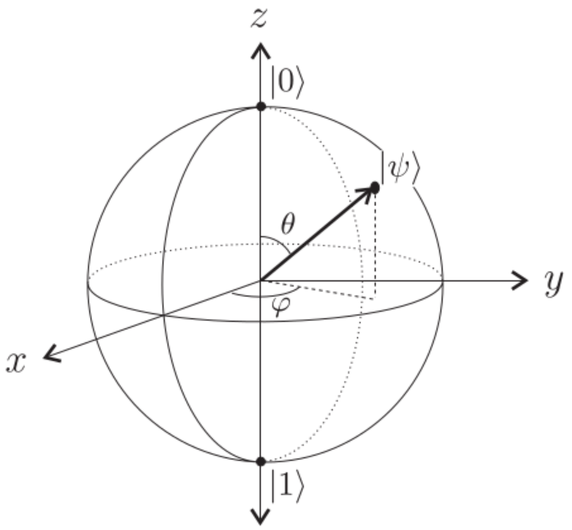


Figure 3.1: State of a qubit on the Bloch Sphere. [13]

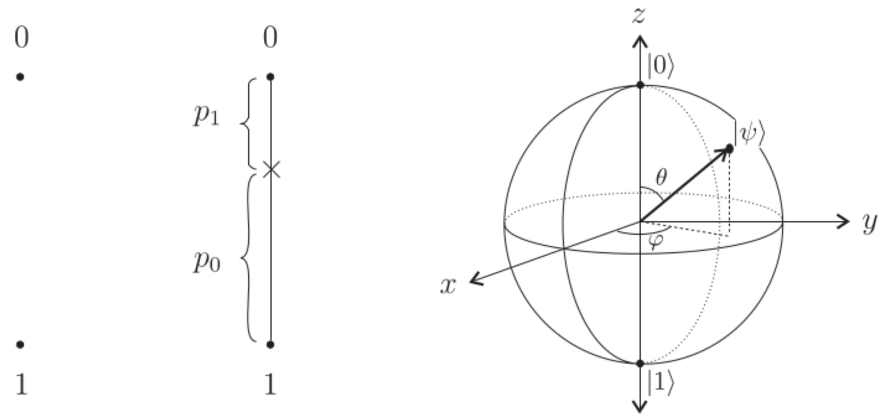


Figure 3.2: States of deterministic classical, probabilistic classical, and quantum bits. [13]

then, in the middle probabilistic computer is shown. In the last one is a quantum bit's state. Qubit's state can be anywhere on Bloch Sphere.

Quantum Bits - Qubits:

So far, the notion “bit” has been explained in terms of classical computers and quantum computers. ‘Bits’ are the fundamental units of classical information and computation. But now it is time to start to use ’qubit’. What is a qubit? Qubit is a basic unit of quantum computing. It is the same as classical bit at storing binary value. However the difference is qubit has 0 and 1 values at the same time.

A qubit can be represented with Dirac or matris notation:

$$\alpha\,|0\rangle + \beta\,|1\rangle \text{ or } \begin{pmatrix} \alpha \\ \\ \beta \end{pmatrix}$$

where α, β are complex numbers and normalization coefficients : $|\alpha|^2 + |\beta|^2 = 1$.

One-qubit with vector notation:

$$|0\rangle = \begin{pmatrix} 1 \\ \\ 0 \end{pmatrix} \text{ground state}$$

$$|1\rangle = \begin{pmatrix} 0 \\ \\ 1 \end{pmatrix} \text{excited state}$$

Multiple Quantum Bits:

And here 2-qubit states representation is shown:

$|\psi\rangle = \alpha\,|00\rangle + \beta\,|01\rangle + \omega\,|10\rangle + \gamma|11\rangle$

with α, β, ω, $\gamma \in \mathbb{C}$. This is called a two-qubit quantum register as well.

and normalization coefficients : $|\alpha|^2 + |\beta|^2 + |\omega|^2 + |\gamma|^2 = 1$.

Vector representation is this:

$$|\psi\rangle = \alpha \begin{pmatrix} 1 \\ \\ 0 \\ \\ 0 \\ \\ 0 \end{pmatrix} + \beta \begin{pmatrix} 0 \\ \\ 1 \\ \\ 0 \\ \\ 0 \end{pmatrix} + \omega \begin{pmatrix} 0 \\ \\ 0 \\ \\ 1 \\ \\ 0 \end{pmatrix} + \gamma \begin{pmatrix} 0 \\ \\ 0 \\ \\ 0 \\ \\ 1 \end{pmatrix}$$

A single qubit lives in the two-dimension Hilbert (complex) space which is $\mathbb{C}^2$. Two-qubit quantum register be a normalized vector in $\mathbb{C}^4$. $\mathbb{C}^4$ can be built in $\mathbb{C}^2$ vector space by tensor product. And n-qubits is an element of 2^n-dimensional complex vector space which is $\mathbb{C}^{2^n}$.

Now a calculation on what is

$|00\rangle$, $|01\rangle$, $|10\rangle$ and $|11\rangle$ $\{|0\rangle,|1\rangle\} \otimes \{|0\rangle,|1\rangle\}$

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

or $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$

In Hilbert (complex) space: $H^{AB} = H^A \otimes H^B$.

And here 2-qubits systems notations:

$$|00\rangle = |0\rangle \otimes |0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \tag{3.1}$$

$$|01\rangle = |0\rangle \otimes |1\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} \tag{3.2}$$

$$|10\rangle = |1\rangle \otimes |0\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \tag{3.3}$$

$$|11\rangle = |1\rangle \otimes |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \tag{3.4}$$

The gates need to be separated as one-qubit gates and multiple-qubit gates. In the Table 3.4 is the explanation of some of the most important gates.

Single Qubit Gates: These gates work on one-qubit; that is why they are called one-qubit operations. The Bloch Sphere was introduced before (Figure 3.1). One can consider one qubit in Bloch Sphere as 3-dim vector and these operations are turning around the axis. This way a qubit can be located at any point on Bloch Sphere.

There is Pauli matrices and they are coming from Pauli Spin matrices. Here, identity **1** matris, “which we always have to put in there to complete our mathematics-it doesn’t do a damn thing!” [9].

$$\sigma_z = Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \text{ and } \sigma_x = X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \tag{3.5}$$

Table 3.1: How X Gate Changes the states.

1-qubit states	X Gate	After X Gate
$\vert 0\rangle$	$\rightarrow$	$\vert 1\rangle$
$\vert 1\rangle$	$\rightarrow$	$\vert 0\rangle$

$$\sigma_y = Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \text{ and } \sigma_0 = I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \tag{3.6}$$

And these are Hermitian.

Pauli-X Gate: Pauli-X Gate gives a tour around x-axis through π. This gate has equivalent on classical computer as NOT Gate. It changes $|0\rangle$ state to $|1\rangle$ state and opposite way like: $|1\rangle$ to $|0\rangle$.

The Pauli-X gate's circuit representation is:

$-\boxed{X}-$

The matrix representation for this operator is: $X \equiv \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$

The Dirac notation is: $X = |1\rangle\langle 0| + |0\rangle\langle 1|$.

Pauli-Y Gate: With Pauli-Y Gate, qubit will turn around y-axis through π. Now X-gate has extra phase. $|0\rangle$ to $i|1\rangle$ and $|1\rangle$ to $-i|0\rangle$.

The Pauli-Y gate's circuit representation is:

$-\boxed{Y}-$

The matrix representation for this operator is: $Y \equiv \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$

The Dirac notation is: $Y = i\left|1\right\rangle\left\langle 0\right| - i\left|0\right\rangle\left\langle 1\right|$.

Pauli-Z Gate: Again as can be understood by its name, with this gate, qubit will turn around z-axis through π. $\left|0\right\rangle$ to $\left|0\right\rangle$ and $\left|1\right\rangle$ to $-i\left|1\right\rangle$.

The Pauli-Z gate's circuit representation is:

$-\boxed{Z}-$

The matrix representation for this operator is: $Z \equiv \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$

The Dirac notation is: $Z = \left|1\right\rangle\left\langle 0\right| - \left|0\right\rangle\left\langle 1\right|$.

There is one more matrix called as Identity I.

$$I \equiv \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Plus, $X^2 = I$, $Y^2 = I$ and $Z^2 = I$. Because all Pauli gates are unitary. That is why their square will be equal to Identity.

For example:

$$X \otimes X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \otimes \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = I$$

Hadamard Gate:

This is the one of the most important gates, because, on which qubit this gate is used it will be in superposition state. And also Hadamard gate is used on same qubit it will become or return same qubit as before. This gate belongs to quantum computers, no equivalent on classical computers.

The Hadamard gate's circuit representation is:

$-\boxed{H}-$

And matrix representation is: $H \equiv \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$

$$|0\rangle \;——\boxed{H}——\; \qquad \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle$$

$$|1\rangle \;——\boxed{H}——\; \qquad \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle$$

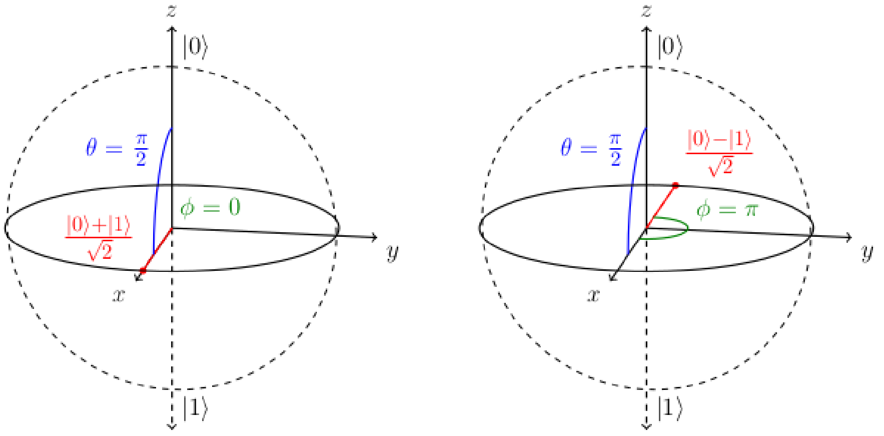
$$H \otimes |0\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{2}} \Bigg[\begin{pmatrix} 1 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} \Bigg]$$

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

$$H \otimes |1\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix} = \frac{1}{\sqrt{2}} \Bigg[\begin{pmatrix} 1 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ -1 \end{pmatrix} \Bigg]$$

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

Here the new basis states are observed: $|+\rangle$ and $|-\rangle$. These are orthonormal basis states.



(a) Basis state $|0\rangle$

(b) Basis state $|1\rangle$

Figure 3.3: Bloch sphere representation of the Hadamard operator applied to $|0\rangle$ and $|1\rangle$ [18].

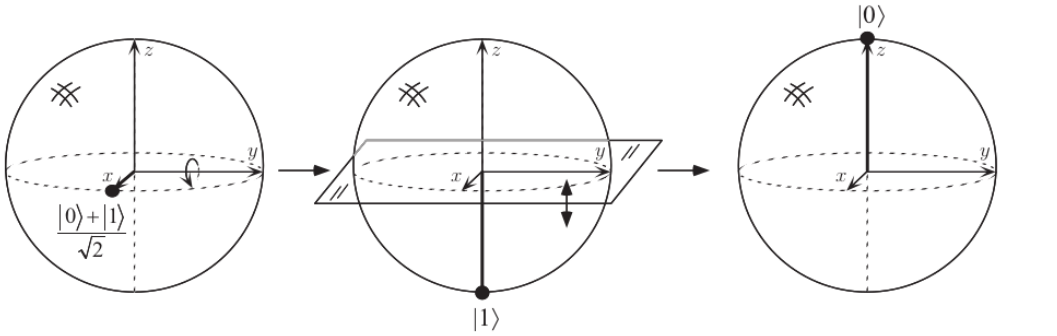


Figure 3.4: Visualization of the Hadamard gate on the Bloch sphere, acting on the input state $(|0\rangle+|1\rangle)/\sqrt{2}$. [14]

They called Hadamard basis too.

$$|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle+|1\rangle) \tag{3.7}$$

$$|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle-|1\rangle) \tag{3.8}$$

Measurement of $|+\rangle, |-\rangle$ probabilities are $|\alpha+\beta|^2/2$ and $|\alpha-\beta|^2/2$, respectively.

Hadamard gate does rotations and reflections. In figure 3.4, it starts $\frac{1}{\sqrt{2}}(|0\rangle+|1\rangle$. It does 90° around y axis and then a rotation about x axis 180°. The result is $|0\rangle$.

$$
\begin{aligned}
H\left[\frac{|0\rangle+|1\rangle}{\sqrt{2}}\right] = \frac{|0\rangle+|1\rangle}{2}\langle 0|0\rangle + \frac{|0\rangle-|1\rangle}{2}\langle 1|0\rangle \\
+\frac{|0\rangle+|1\rangle}{2}\langle 0|1\rangle + \frac{|0\rangle-|1\rangle}{2}\langle 1|1\rangle \\
= \frac{|0\rangle+|1\rangle}{2} + \frac{|0\rangle-|1\rangle}{2} = |0\rangle
\end{aligned} \tag{3.9}
$$

$$
\begin{aligned}
H\left[\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right] = \frac{|0\rangle+|1\rangle}{2}\langle 0|0\rangle + \frac{|0\rangle-|1\rangle}{2}\langle 1|0\rangle \\
-\frac{|0\rangle+|1\rangle}{2}\langle 0|1\rangle - \frac{|0\rangle-|1\rangle}{2}\langle 1|1\rangle \\
= \frac{|0\rangle+|1\rangle}{2} - \frac{|0\rangle-|1\rangle}{2} = |1\rangle
\end{aligned} \tag{3.10}
$$

Multiple Qubit Gates:

CNOT Gate: The name CNOT comes from *Controlled-NOT*. This gate is one of the other most important gates. This gate has a difference from NOT gate. The difference is that: there is control qubit and target qubit. It checks first bit and if it is 0 no changes on second bit, but if it is 1 it changes second bit, see Figure 3.5.

It can be used on 2-qubit states $(00, 11, 01, 10)$. This does not mean, it can only be used on 2-qubit states, also used for 3-qubit and n-qubit states as well, See Figure 3.6.

The CNOT gate's circuit representation is:

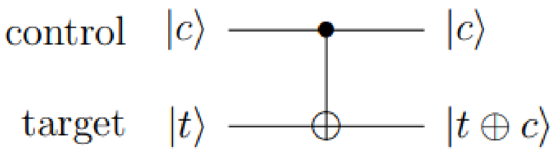
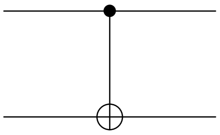


Figure 3.5: The representation of CNOT gate of control and target [18].

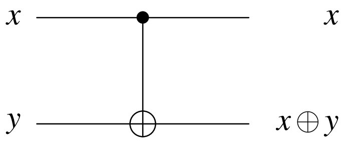
Table 3.2: How CNOT Gate changes the states.

2-qubit states	CNOT Gate	After CNOT Gate
$\|00\rangle$	→	$\|00\rangle$
$\|01\rangle$	→	$\|01\rangle$
$\|10\rangle$	→	$\|11\rangle$
$\|11\rangle$	→	$\|10\rangle$

CNOT Gate



Here, how the CNOT gate works:



and here matrix form of CNOT Gate:

$$CNOT \equiv \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

CNOT operator with Dirac notation:

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

And now it is necessary to look at how it changes the states of $(00, 11, 01, 10)$.

$$(CNOT) \otimes |00\rangle = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \otimes \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} = |00\rangle \tag{3.11}$$

$$(CNOT) \otimes |01\rangle = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \otimes \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} = |01\rangle \tag{3.12}$$

$$(CNOT) \otimes |10\rangle = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \otimes \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = |11\rangle \tag{3.13}$$

$$(CNOT) \otimes |11\rangle = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \otimes \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} = |10\rangle \tag{3.14}$$

CNOT gate is the gate qubits can be put in entanglement state with it.

SWAP Gate: This gate changes two qubits with each other. It is mostly used with super conductive circuits.

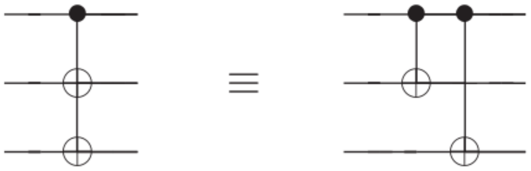
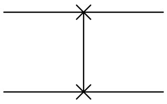


Figure 3.6: The representation of CNOT gate on 3-qubit states. [14].

The SWAP gate's circuit representation is:

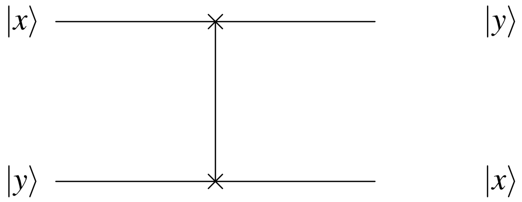
SWAP Gate



And matrix representation is:

$$SWAP \equiv \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

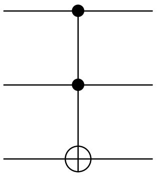
How SWAP Gate works:



Toffoli Gate: It is called as CCNOT gate too. The name is CCNOT coming from *controlled-controlled-NOT*. The gate is mostly like CNOT gate. For this gate 3-qubits are needed. Now first 2-qubits are control bits and it changes the third one. The rule is first two qubits should be $|1\rangle$ and then X gate will do operation on 3. qubit.

The Toffoli gate's circuit representation is:

Toffoli Gate



And matrix representation:

$$Toffoli \equiv \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Toffoli operator with Dirac notation:

$Toffoli = |000\rangle\langle 000| + |001\rangle\langle 001| + |010\rangle\langle 010| + |011\rangle\langle 011| + |100\rangle\langle 100| + |101\rangle\langle 101| + |111\rangle\langle 110| + |110\rangle\langle 111|$

And now how it changes the states of $(000, 001, 010, 011, 100, 101, 110, 111)$, see Table 3.3.

3.1 Circuit Theory

The Linear Algebra Formulation of the Circuit Model: A qubit in state of ‘0‘ probability shown with p_0 and in state of ‘1‘ probability is p_1. And here is 2-dim

Table 3.3: How Toffoli Gate Changes the states.

3-qubit states	Toffoli Gate	After Toffoli Gate
$\lvert 000\rangle$	→	$\lvert 000\rangle$
$\lvert 001\rangle$	→	$\lvert 001\rangle$
$\lvert 010\rangle$	→	$\lvert 010\rangle$
$\lvert 011\rangle$	→	$\lvert 011\rangle$
$\lvert 100\rangle$	→	$\lvert 100\rangle$
$\lvert 101\rangle$	→	$\lvert 101\rangle$
$\lvert 110\rangle$	→	$\lvert 111\rangle$
$\lvert 111\rangle$	→	$\lvert 110\rangle$

vector notation:

$$\begin{pmatrix} p_0 \\ \\ p_1 \end{pmatrix}$$

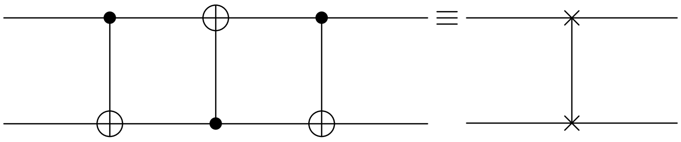
With this information $|0\rangle$ and $|1\rangle$ can be written as

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

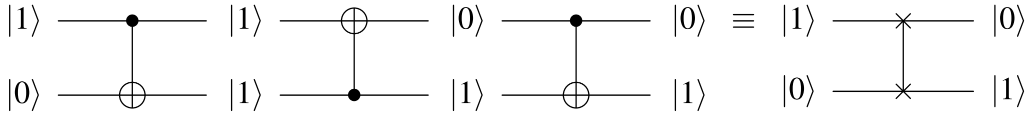
$p_0 = 1$ because it is written as $|0\rangle$ so, having probability of state ‘0‘ is *one* and ‘1’s is *zero*. That is why $p_1 = 0$. With same logic $|1\rangle$ is written as

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

This part is really important. Because a particular gate can be used instead of some other gates. For example, CNOT gate is used three times on a qubit it will be equivalent to SWAP gate. Here;

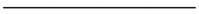
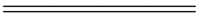
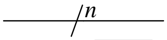
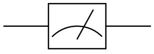
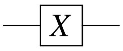
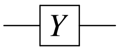
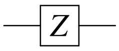
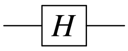
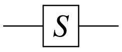
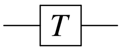
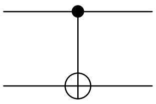
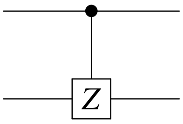
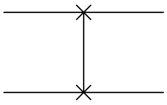
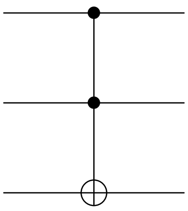


It is now time to show how after three times CNOT Gate will be equal to SWAP Gate, starting with $|10\rangle$ and ending with $|01\rangle$, showing how SWAP gate works.



Here the most important gates are shown below.

Table 3.4: Quantum Gates

Name	Circuit Representation	Matrix Representation
Qubit		Wire carrying a single qubit (time goes left to right)
Classical Bit		Wire carrying a single classical bit
n qubits		Wire carrying n qubits
Measurement		Projection onto $\|0\rangle$ and $\|1\rangle$
Pauli-X		$\begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}$
Pauli-Y		$\begin{bmatrix}0 & -i\\ i & 0\end{bmatrix}$
Pauli-Z		$\begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix}$
Hadamard		$\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}$
S		$\begin{bmatrix}1 & 0\\ 0 & i\end{bmatrix}$
T		$\begin{bmatrix}1 & 0\\ 0 & e^{i\pi/4}\end{bmatrix}$
CNOT		$\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & 1 & 0\end{bmatrix}$
C(Z)		$\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & -1\end{bmatrix}$
SWAP		$\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}$
Toffoli		$\begin{bmatrix}1 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1\\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0\end{bmatrix}$

Chapter 4

GROVER'S ALGORITHM

Considering a map of a city is given and one is supposed to find the shortest route by passing through all streets on this map. To find the shortest route, one needs to check all streets one by one. On the other hand, it might be said a search algorithm is created and the answer may be found in a quicker way. This is possible with Quantum Search Algorithm which is known Grover's Algorithm as well. It is known as the fastest search algorithm. Grover's Algorithm is a quantum search algorithm and it is a quantum algorithm in which the qualities of quantum systems can be used, because it works with quantum superposition of states. It can do a search in unordered sets. It uses iteration to find the answer. It is quicker than classical algorithms, especially if there are too many items to search to find the answer, since it has a quadratic speed up. For trying to find the shortest route with classical way it takes N times but with Grover's Algorithm it is root of N times. Like most of quantum algorithms, Grover's Algorithm uses 'amplitude amplification'.

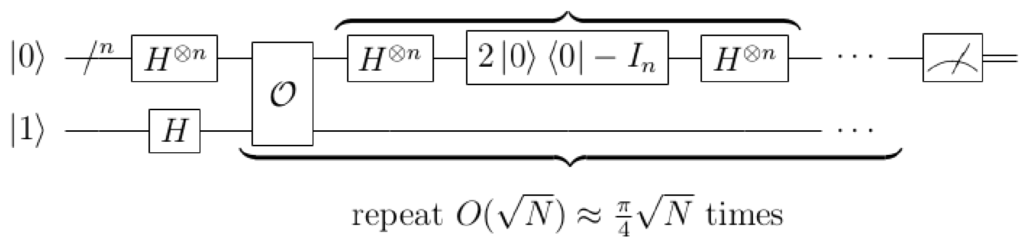


Figure 4.1: Circuit diagram for Grover's algorithm, with a scratch qubit for the oracle [18].

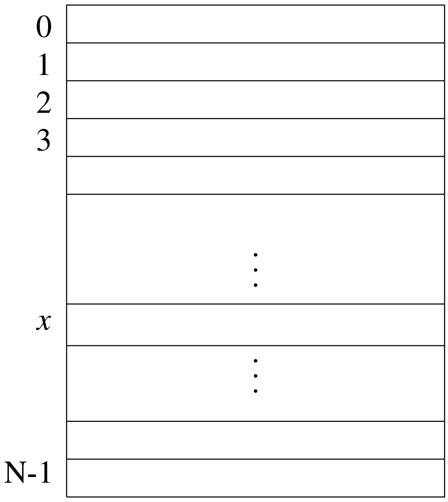
The way of these algorithms is shifting of selected phase which one has one state of a quantum system and satisfies some conditions for all iterations. Algorithm shifts a phase as π and change that state to '$-$'.

Probability of that state stays the same but amplitude changes. As it was said earlier the name is amplitude amplification.

One can consider there is an unordered set of $N = 2^n$ items and only one of them is marked. The mission is finding this marked item.
One can only suppose that there is a function called as f and it has following properties:
$f(x) = 1$ if x is marked
$f(x) = 0$ otherwise

By 'oracle' the list of f can be searched. With classical way $\mathcal{O}(N)$ times need to be calculated but with quantum only $\mathcal{O}(\sqrt{N})$ operations. This is quadratic speed up. It works with two different ways. First one is Phase Inversion and second one is Inversion About Mean. In next chapter the steps of Grover's Algorithms and why it runs in $\sqrt{N}$ steps will be described. However this is not enough to understand that how it is implemented the steps yet. Thus, the next part is about how one can implement the steps.

Here an explanation is given about how Grover's Algorithm works. To begin with, a "Digital haystack" can be considered:



Here is the problem.

Problem: Given $f : \{0,1,...,N-1\} \rightarrow \{0,1\}$ Find $x : f(x) = 1$

There is one special entry and one needs to find that special entry. Only in two steps that special entry in Grover's Algorithm will be reached;

1) Phase Inversion

2) Inversion About Mean

1) Phase Inversion:

Assuming that the special entry is $f(x^*) = 1$, that is what we are looking for at any given iteration of the algorithm. So, the algorithm will work in a number of iteration and actually the number of iteration is going to be $\sqrt{N}$ but how? At any iteration what the algorithm continues is a superposition over all 'x's. So, it can be shown as,

$$\sum_x \alpha_x |x\rangle$$

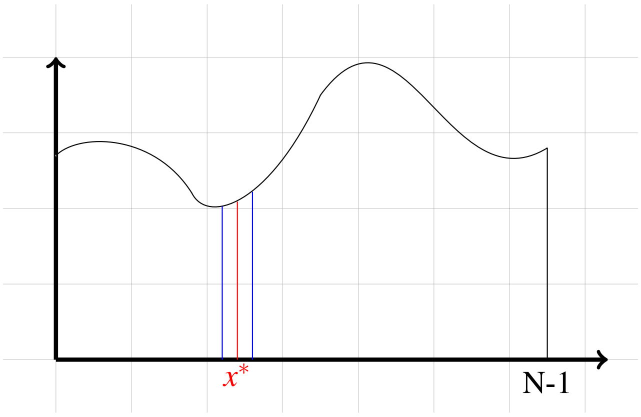
Actually in the beginning, one may not have any idea which value of x is looked for.

That's why all $\alpha_x = \frac{1}{\sqrt{N}}$.

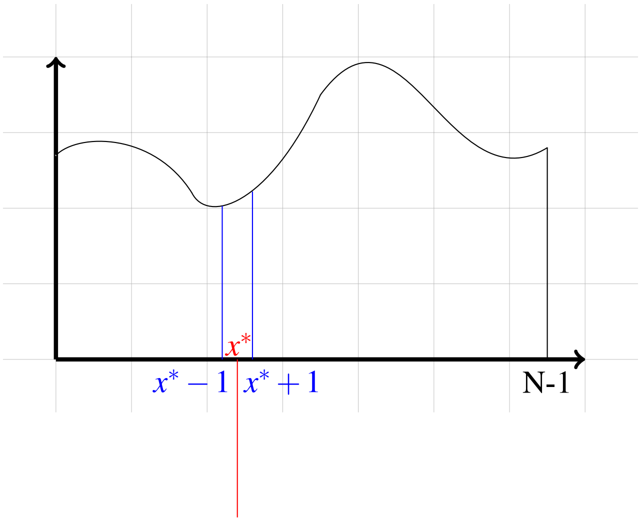
Phase inversion in step two does that it changes the superposition like this:

If $x \neq x^*$ it will leave it alone. This means that if it is not the special element. Otherwise, it will invert the phase ($x = x^*$).

$$\alpha_x = \frac{1}{\sqrt{N}} \rightarrow \sum_{x \neq x^*} \left(\alpha_x \left| x \right\rangle - \alpha_{x^*} \left| x^* \right\rangle \right)$$



the red line is amplitude of x^* and x^* will be inverted.



The red line inverted but blue lines will stay unchanged. Whatever x^* was stays same but inverted. That is the first operation.

2. Inversion About Mean:

This is the second operation which is inversion about the mean.

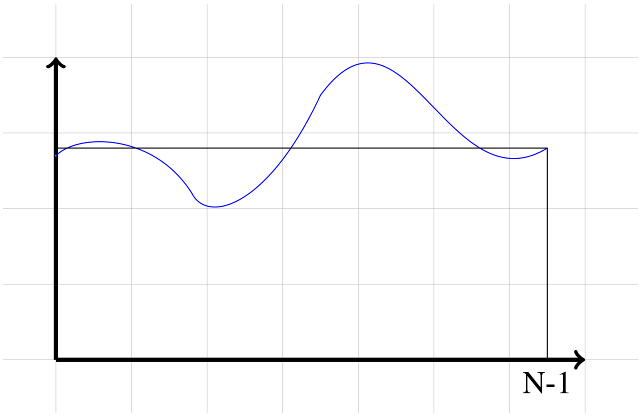
Starting point with a superposition: $\sum_x \alpha_x |x\rangle$.

What does it mean 'Inversion About Mean' ? μ can be described here. It means 'mean', and is equal to

$$\mu = \frac{\sum_{x=0}^{N-1} \alpha_x}{N}$$

This is the average value of all the amplitude. Then time to settle the average.

Now is time to flip the amplitude about the mean:

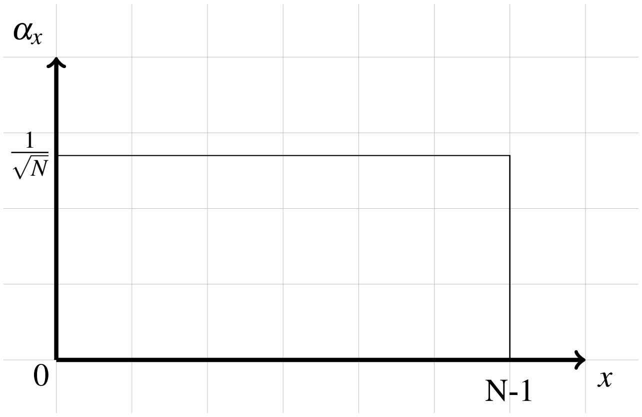


$$\alpha_x \rightarrow (2\mu - \alpha_x) = \mu + (\mu - \alpha_x) \sum_x \alpha_x |x\rangle \rightarrow \sum_x (2\mu - \alpha_x) |x\rangle$$

$\alpha_x < \mu$. This means that how much it is smaller than μ when it flips up. It is totally the same as when it flips down.

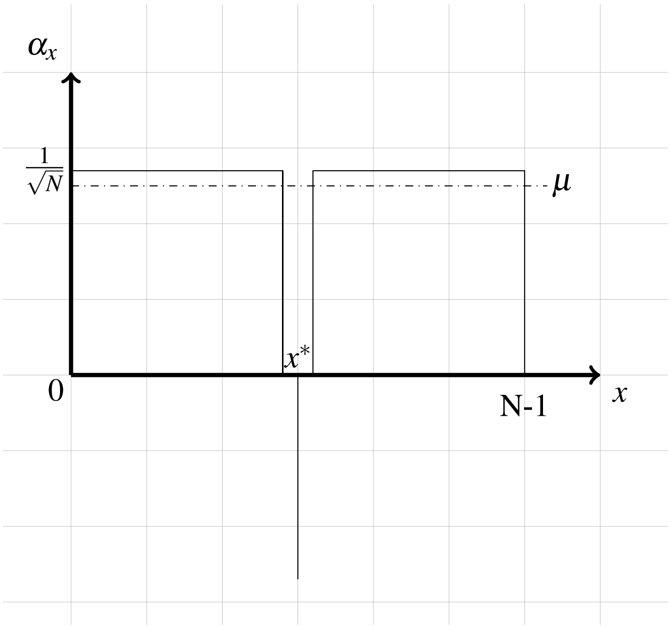
Now time to explain how Grover's Algorithm works.

Problem: Given $f : \{0,...,N-1\} \rightarrow \{0,1\}$ such that $f(x) = 1$ for exactly one x, find x.

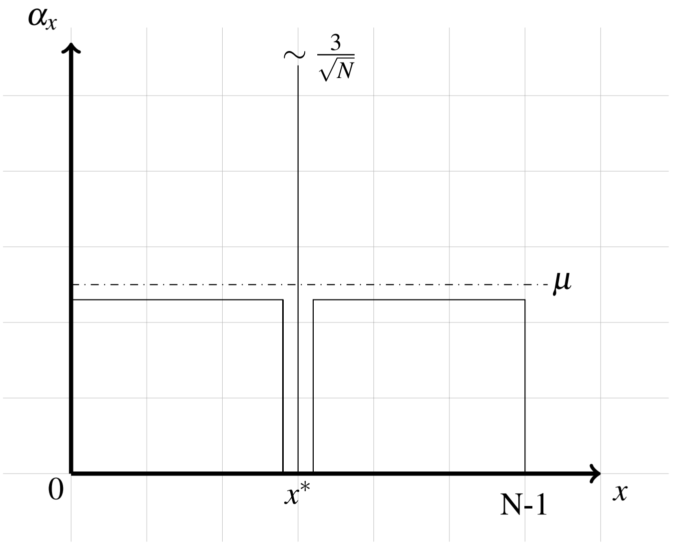


Initially nothing is known about the marked item (x^*, special entry). So, to start with all item's amplitude is equal to $1/\sqrt{N}$. Then the phase inversion is done.

Till here x^* was equal to $1/\sqrt{N}$ but now it is equal to $-1/\sqrt{N}$.



Now, it is time for the inversion about the mean. So, what is the mean? It would have been $1/\sqrt{N}$ if one had not done the phase inversion. What the phase inversion does is it lowers the mean just a little bit. Now what happens when about the is mean inverted?

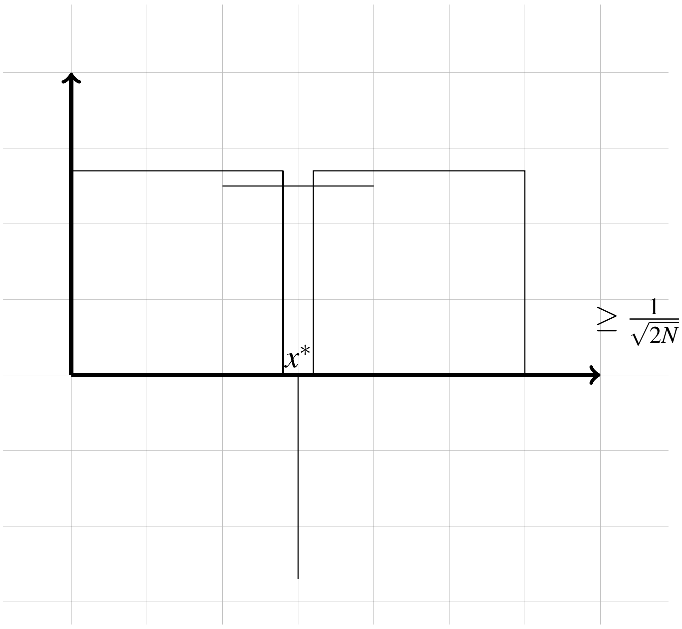


Everything except x^* amplitude drops a bit. It drops as much below the mean as it was about the mean before. So, what happened to x^*. When it is flipped up it goes up as much about this mean as it was below and it was below about $2/\sqrt{N}$. So it goes up $2/\sqrt{N}$ and approximately about this mean above this mean was approximately $1/\sqrt{N}$. So x^* got it's amplitude increased by about $2/\sqrt{N}$ in these two steps. x^* increased from $1/\sqrt{2}$ to $\sim 3/\sqrt{N}$. After this point proceeding will be same. It can continue these steps over and over again. Each time this is done, each time the amplitude of x^* increases about $2/\sqrt{N}$. With this case it will go to $5/\sqrt{N}$, $7/\sqrt{N}$ and etc. It goes on and on in roughly $\sqrt{N}$ steps it reaches this amplitude to about $1/\sqrt{2}$. At this point if this is measured, the chance that one can see the x^* is on the needle $\sim 1/\sqrt{2}$ and it is x^*'s amplitude. And now the marked item is founded.

How many steps are needed?

What is the amplitude of the rest of the elements when the needle in the haystack (the marked element) has $1/\sqrt{2}$? The rest amplitude should be at least $1/\sqrt{2N}$ (each of them).

At this point how much improvement is made per step? One will reach $1/\sqrt{2}$ in $\mathcal{O}(\sqrt{N})$ steps.



Improvement / step $\geq \frac{2}{\sqrt{2N}} = \sqrt{\frac{2}{N}}$. So, to reach $\frac{1}{\sqrt{2}} in \mathcal{O}(\sqrt{N})$ How many steps are needed?

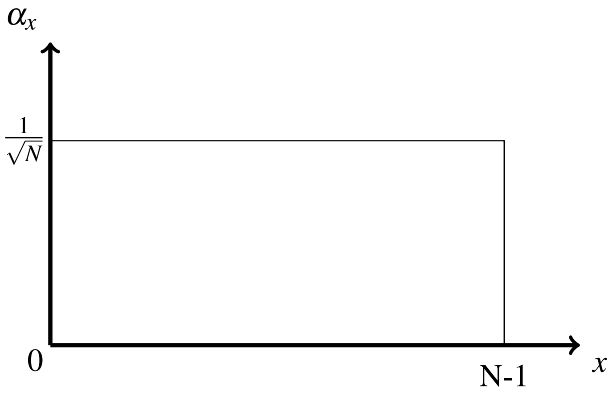
Number of steps $\leq \frac{\frac{1}{\sqrt{N}}}{\sqrt{\frac{2}{N}}} = \frac{\sqrt{N}}{2}$

This was analysis of Grover's Algorithm. How is Grover's Algorithm implemented in these two steps?

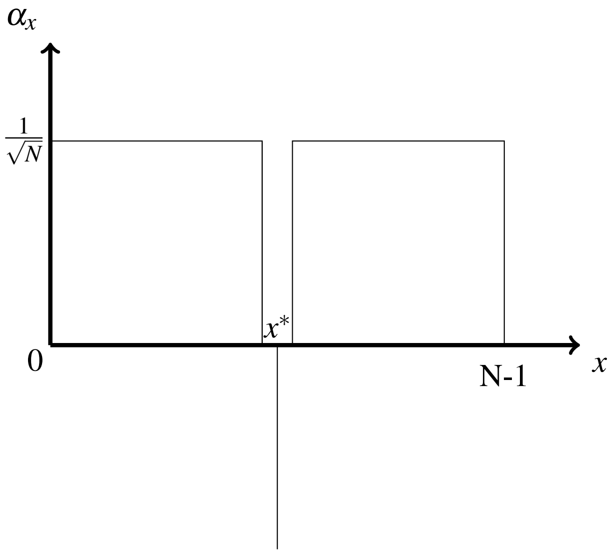
Phase Inversion:

What was done was to take the marked element the one where $f(x) = 1$ and invert it's face. That means if it's face was positive it was made negative as can be seen below in the figures.

Problem: Given $f : \{0, ..., N-1\} \rightarrow \{0, 1\}$ such that $f(x) = 1$ for exactly one x, find x.



this turns to

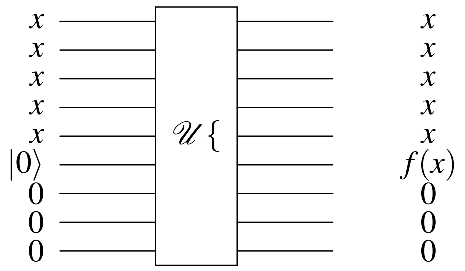


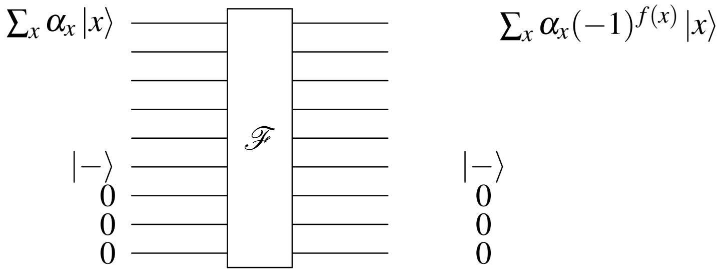
So, what exactly is going on here, if it is started with a superposition:

$$\sum \alpha_x |x\rangle \rightarrow^{phase}_{inversion} \sum_x \alpha_x (-1)^{f(x)} |x\rangle$$

The phase that needs to be applied is $(-1)^{f(x)}$.

This box can be changed to $(-1)^{f(x)} |x\rangle$. How will this be done? All needs to be done is change $|0\rangle$ to $|-\rangle$ state. $|0\rangle$ is the answer bit!




$\sum_x \alpha_x |x\rangle$
$|-\rangle$
0
0
0
$\mathscr{F}$
$\sum_x \alpha_x (-1)^{f(x)} |x\rangle$
$|-\rangle$
0
0
0

And now this $|-\rangle$ state will stay unchanged and effect of $f(U_f)$ is put that phase $(-1)^{f(x)}$ part. How? $|0\rangle$ and $|b\rangle$ are replaced. So, b is 0 or 1. Now what is the output? $|f(x) \oplus b\rangle$. If b start with 0 then output is $f(x)$. But, if b start with 1 output is opposite of $f(x)$.

b / $f(x)$	0	1
0	0	1
1	1	0
$\|-\rangle$	$\|-\rangle$	$-\|-\rangle$

For first case $f(x) = 0$;

when b is 0, output 0. Because output is the same with $f(x)$ when b is 0. On the other hand, when b is 1, output 1. Because, when b is 1 then the output is opposite of $f(x)$ that is why output will be 1.

And the second case $f(x) = 1$;

when b is 0, output 1. when b is 1, output 0.

Now it is time to see what happens if one starts with a state $|-\rangle$ which is: $|-\rangle = \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle$.

When $|-\rangle$ is put in it and when $f(x)$ is 0: In the case that when $f(x)$ and b was 0 output was same with input which was 1. That's mean if input is $|-\rangle$ output will be exactly $|-\rangle$. What happens in the case of when $f(x)$ was 1; when b was 0 output was 1. So the result of $|-\rangle$'s first part will be: $\frac{1}{\sqrt{2}}|0\rangle => \frac{1}{\sqrt{2}}|1\rangle$ and when b was 1 output was 0. So the result of $|-\rangle$'s second part will be: $\frac{1}{\sqrt{2}}|1\rangle => -\frac{1}{\sqrt{2}}|0\rangle$. The state that is obtained from output is exactly $-|-\rangle$.
$-|-\rangle = \frac{1}{\sqrt{2}}|1\rangle - \frac{1}{\sqrt{2}}|0\rangle$.

That's what was done up here. When input is $|-\rangle$ and if $f(x) = 0$ it is just exactly the state $-$, if $f(x) = 1$ one picks up a phase of -1. So, that's **Phase Inversion!** Now it can be said that output of $|-\rangle$ is $(-1)^{f(x)}|-\rangle$.

What about **Inversion About Mean?** Considering that inversion about mean, it has started with superposition

$$\sum_x \alpha_x |x\rangle$$

Define the mean of the all amplitudes to be average value of all the amplitudes.

$$\mu = \frac{\sum_{x=0}^{N-1}}{N} \tag{4.1}$$

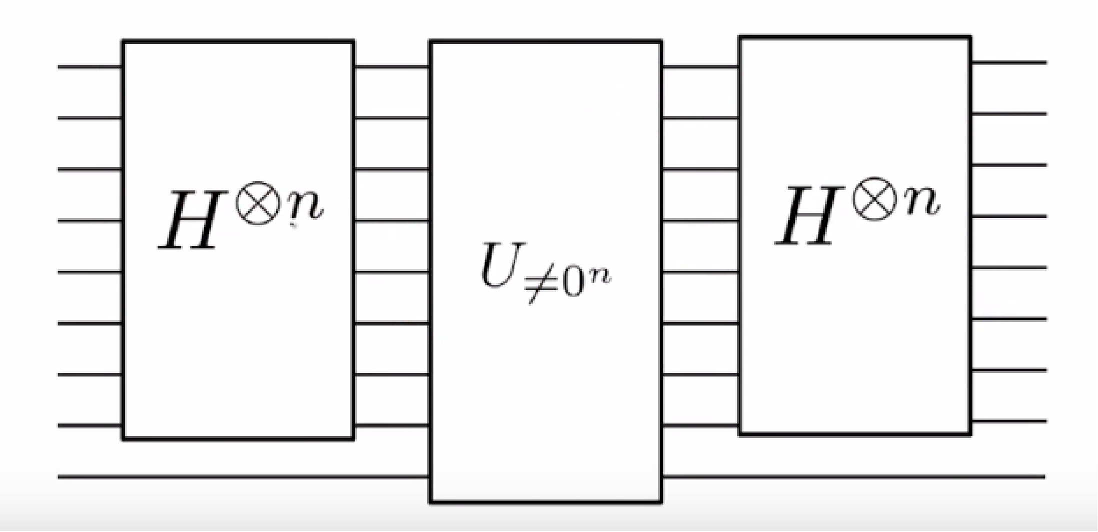


Figure 4.2: The general view of Grover's algorithm as circuit

Now, what has been reached is this transformation

$$\alpha_x \rightarrow (2\mu - \alpha_x) = \mu + (\mu - \alpha_x)$$

$$\sum_x \alpha_x |x\rangle \rightarrow \sum_x (2\mu - \alpha_x) |x\rangle$$

So what the quantum circuit to do is in the Figure 4.2. All inputs will be $\sum_x \alpha_x |x\rangle$ except last input. The last qubit's input will be $|-\rangle$ and then Hadamard transformation in the first step. Right after that one apply the function on n qubits. But what is going to happen on when last qubit input is $|-\rangle$? And inputs called as $y = \sum_x \alpha_x |x\rangle$.

If last qubit input is 0;

0 if $y = 0, ..., 0$

1 if $y \neq 0, ..., 0$

This is unitary transformation and now consider a function from n bits to one bit:

$g : \{0,1\}^n \rightarrow \{0,1\}$

the function $g(0,...0) = 0$ and $g(y) = 1$ if $y \neq 0, ...0$.

And the middle of the circuit is U_g. And then one can finish up by doing Hadamard transform on n qubits.

Going back to that Inversion About Mean part. It turns out that inversion about mean is the same as doing reflection about means looking a quantum state. $|u\rangle = \frac{1}{\sqrt{N}} \sum_x |x\rangle$.

One way to do it is to transform u in to the all 0's vector and than do reflection about 0's vector:

1) Transform $|u\rangle$ into $|0...0\rangle$

2) Reflection about $|0...0\rangle$

3) Transform $|0...0\rangle$ back into $|u\rangle$.

What unitary transform should be taken? It's clearly H transform $H^{\otimes n}$. How one reflects about all 0 vectors. All 0 vectors should be alone and everything is orthogonal to it, one multiply by -1. The transformation does this:

$$H^{\otimes n} \begin{bmatrix} 1 & 0 & . & . & 0 \\ 0 & -1 & . & & . \\ . & . & . & . & . \\ . & & . & . & 0 \\ 0 & . & . & 0 & -1 \end{bmatrix} \tag{4.2}$$

and than transform back $H^{\otimes n}$ its own inverse, so that is the transformation that needs to be carried out. If one checks again Figure 4.2 that is exactly what is done there. One applied phase of -1 if and only if $y \neq 0, ...0$.

$$H^{\otimes n} \begin{bmatrix} 1 & 0 & . & . & 0 \\ 0 & -1 & . & & . \\ . & . & . & . & . \\ . & & . & . & 0 \\ 0 & . & . & 0 & -1 \end{bmatrix} H^{\otimes n} = H^{\otimes n} \left(\begin{bmatrix} 2 & 0 & . & . & 0 \\ 0 & 0 & . & & . \\ . & . & . & . & . \\ . & & . & . & 0 \\ 0 & . & . & 0 & 0 \end{bmatrix} - I \right) H^{\otimes n}$$

$$= H^{\otimes n} \begin{bmatrix} 1 & 0 & . & . & 0 \\ 0 & -1 & . & & . \\ . & . & . & . & . \\ . & & . & . & 0 \\ 0 & . & . & 0 & -1 \end{bmatrix} H^{\otimes n} - \underbrace{H^{\otimes n} I H^{\otimes n}}_{\text{I}}$$

Remember $N = 2^n = 2^2$

$$= \begin{bmatrix} 2/\sqrt{N} & 0 & . & . & 0 \\ . & . & . & & . \\ . & . & . & . & . \\ . & . & . & . & 0 \\ 2/\sqrt{N} & . & . & 0 & 0 \end{bmatrix} H^{\otimes n} - I = \begin{bmatrix} \frac{2}{N} & \frac{2}{N} & . & . & \frac{2}{N} \\ . & . & . & & . \\ . & . & . & . & . \\ . & & . & . & . \\ \frac{2}{N} & . & . & . & \frac{2}{N} \end{bmatrix} - I$$

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

The question is why this matrix does an inversion about the mean? The answer is that when it operates on α's. So, what is $\frac{2}{N}\sum_y \alpha_y$? It is just equal to 2μ. Because $\mu = \frac{\sum_{x=0}^{N-1}}{N}$.

$$\begin{pmatrix} \frac{2}{N}-1 & \frac{2}{N} & . & . & \frac{2}{N} \\ . & . & . & & . \\ . & . & . & . & . \\ . & & . & . & . \\ \frac{2}{N} & . & . & . & \frac{2}{N}-1 \end{pmatrix} \begin{pmatrix} \alpha_0 \\ . \\ . \\ \alpha_x \\ . \\ . \\ \alpha_{N-1} \end{pmatrix} \rightarrow \begin{pmatrix} \\ \\ \\ \frac{2}{N}\sum_{y=0}^{N-1} \alpha_y - \alpha_x \\ \\ \\ \\ \end{pmatrix} = 2\mu - \alpha_x$$

The result is exactly the same with inversion about mean.

If one checks Figure 4.3, the first part is initialization and right after that (U_f) is the phase inversion and last three boxes are for inversion about mean. This whole circuit of Grover's algorithm but only for one iteration.

The computational complexity of the Grover's algorithm is only $\mathscr{O}(\sqrt{N})$ iterations.

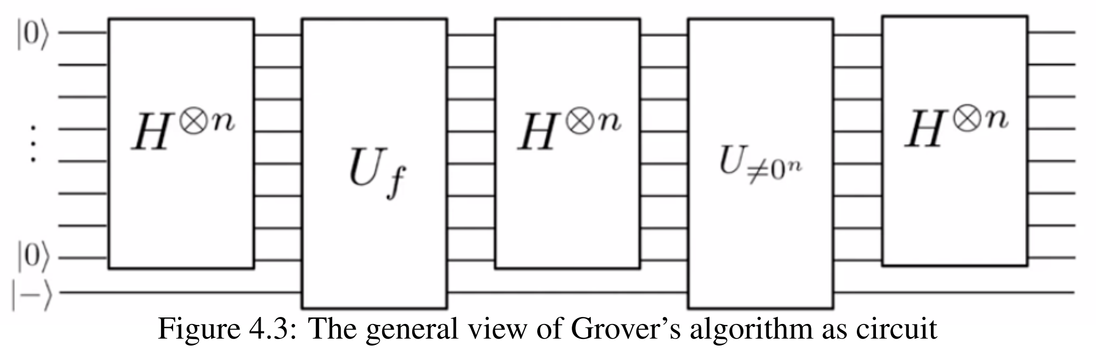


Figure 4.3: The general view of Grover's algorithm as circuit

4.1 Worked Example with 3 Qubits

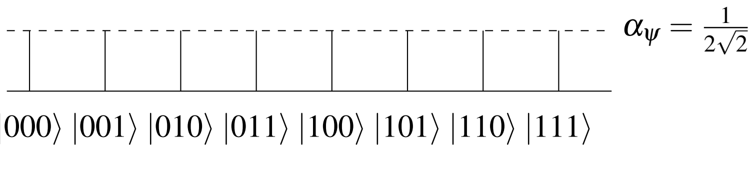
It can be shown how Grover's Algorithm works with 3 qubits. As it was explained before with $N = 2^n$ when there are 3 qubits it means that $N = 2^3$. That special item was called as x^* before. When it is assumed that $x^* = 011$. $n = 3$ qubits are needed and can be described as

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

α_i is the amplitude of the state of i. In Grover's Algorithm all states initialized to 0 and after Hadamard transformation normalized to 1: $1\,|000\rangle$. After the Hadamard transformation each state's amplitude will be $\frac{1}{\sqrt{N}}$. For this example $N = 8$. So, $N = \frac{1}{\sqrt{8}} = \frac{1}{2\sqrt{2}}$. Besides, all states have equal probability of being in any of the eight possible states which are:

$H^3\,|000\rangle = \frac{1}{2\sqrt{2}}\,|000\rangle + \frac{1}{2\sqrt{2}}\,|001\rangle + ... + \frac{1}{2\sqrt{2}}\,|111\rangle = \frac{1}{2\sqrt{2}}\sum_{x=0}^{7}|x\rangle = \psi$

After this step, this graph can be shown as the situation of these 3 qubits:

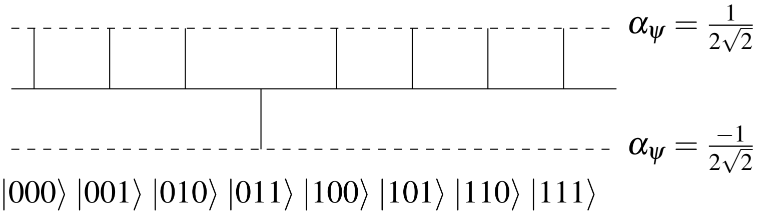


A calculation can be made about how many iterations needed to do with $\frac{\pi}{4}\sqrt{N}$.

For 3 qubits $N = 8$ so, $\frac{\pi}{4}\sqrt{8} = \frac{2\pi}{4}\sqrt{2} = \frac{\pi}{2}\sqrt{2} \approx 2.22$. It takes 2 rounds.

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

Geometric representation of this step:


$\alpha_\psi = \frac{1}{2\sqrt{2}}$
$\alpha_\psi = \frac{-1}{2\sqrt{2}}$
$|000\rangle$ $|001\rangle$ $|010\rangle$ $|011\rangle$ $|100\rangle$ $|101\rangle$ $|110\rangle$ $|111\rangle$

Now, time to that diffusion transform $2|\psi\rangle\langle\psi| - I$ after this step the x^* 's amplitude will be increased or decreased.

$$[2|\psi\rangle\langle\psi| - I]\,|x\rangle = [2|\psi\rangle\langle\psi| - I]\big[\,|\psi\rangle - \frac{2}{2\sqrt{2}}|011\rangle\,\big]$$

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

$\langle\psi|\,|\psi\rangle = 1$, Here $\langle\psi|\,|\psi\rangle = 8\frac{1}{2\sqrt{2}}\big[\frac{1}{2\sqrt{2}}\big] = 1$.

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

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

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

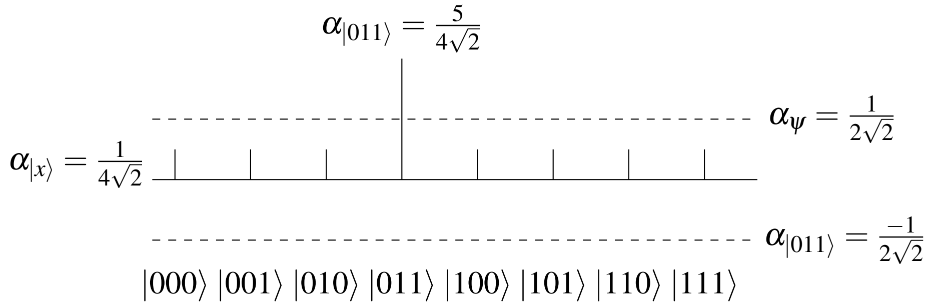
$$= \frac{1}{4\sqrt{2}}\sum_{x=0,x\neq 3}^{7}|x\rangle + \frac{1}{4\sqrt{2}}|011\rangle + \frac{1}{\sqrt{2}}|011\rangle$$

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

The notation will be as show before:

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

Again in geometric way:


$\alpha_{|011\rangle} = \frac{5}{4\sqrt{2}}$
$\alpha_{\psi} = \frac{1}{2\sqrt{2}}$
$\alpha_{|x\rangle} = \frac{1}{4\sqrt{2}}$
$\alpha_{|011\rangle} = \frac{-1}{2\sqrt{2}}$
$|000\rangle$ $|001\rangle$ $|010\rangle$ $|011\rangle$ $|100\rangle$ $|101\rangle$ $|110\rangle$ $|111\rangle$

Till here is the first completed iteration and now time for the second iteration with same two transformation:

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

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

$$= \frac{1}{4\sqrt{2}}\sum_{x=0}^{7}|x\rangle - \frac{6}{4\sqrt{2}}|011\rangle = \frac{1}{2}|\psi\rangle - \frac{3}{2\sqrt{2}}|011\rangle$$

after oracle touch and the diffusion transform:

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

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

$$= |\psi\rangle - \frac{1}{2}|\psi\rangle - \frac{3}{\sqrt{2}}\left(\frac{1}{2\sqrt{2}}\right) + \frac{3}{2\sqrt{2}}|011\rangle = -\frac{1}{4}|\psi\rangle + \frac{3}{2\sqrt{2}}|011\rangle$$

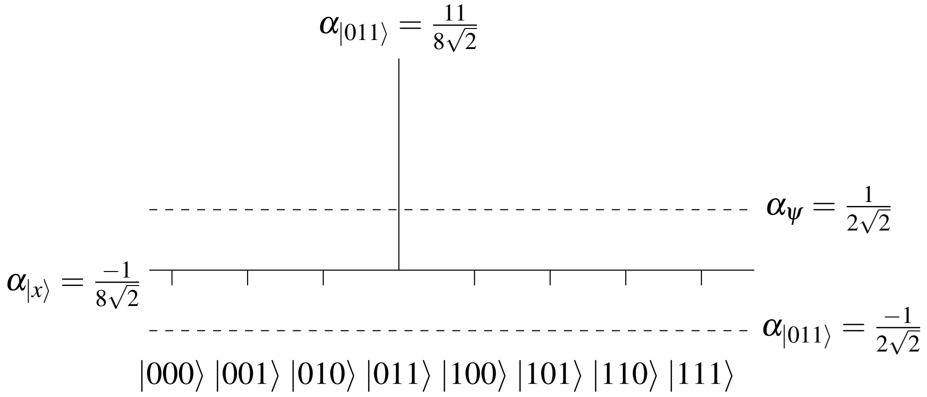
$$= -\frac{1}{4}\left[\frac{1}{2\sqrt{2}}\sum_{x=0,x\neq 3}^{7}|x\rangle + \frac{1}{2\sqrt{2}}|011\rangle\right] + \frac{3}{2\sqrt{2}}|011\rangle$$

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

and with other notation:

$|x\rangle = -\frac{1}{8\sqrt{2}}|000\rangle - \frac{1}{8\sqrt{2}}|001\rangle - \frac{1}{8\sqrt{2}}|010\rangle + \frac{11}{8\sqrt{2}}|011\rangle - ... - \frac{1}{4\sqrt{2}}|111\rangle$

and again geometrically:



$\alpha_{|011\rangle} = \frac{11}{8\sqrt{2}}$
$\alpha_{\psi} = \frac{1}{2\sqrt{2}}$
$\alpha_{|x\rangle} = \frac{-1}{8\sqrt{2}}$
$\alpha_{|011\rangle} = \frac{-1}{2\sqrt{2}}$
$|000\rangle$ $|001\rangle$ $|010\rangle$ $|011\rangle$ $|100\rangle$ $|101\rangle$ $|110\rangle$ $|111\rangle$

Now, it is easy to see the special item which is x^* that how much it clearly stays ahead from other states.

After this, it can be measured. Now it is time to find out the probability of $|011\rangle$ state.

$\left|\frac{1}{8\sqrt{2}}\right|^2 = 121/128 \approx 94.5\%$

And also the probability of other states is

$\left|\frac{-\sqrt{7}}{8\sqrt{2}}\right|^2 = 7/128 \approx 5.5\%$

It can be seen from this example that Grover's Algorithm is approximately 17 times more than to give the correct answer. This example was for when $N = 8$ means with 3 qubits. If the Grover's Algorithm is used with much bigger number of qubits then Grover's algorithm will give more correct answer. Because it grows quadratic as $\mathscr{O}\sqrt{N}$.

Chapter 5

CONCLUSION

Up to now, what has been explained is the difference between classical computers and quantum computers. What has been presented and discussed is expected to show how efficient quantum computers are. Plus, it has also been shown that it is not possible to simulate everything using classical computers. As has been mentioned before, nature is so complicated and it is not possible to understand nature simply by simulating it. That is why a machine is needed to simulate Nature. What is more, it has also been shown that quantum computers can solve problems in very short period of time, with more efficient results. This is because quantum algorithms are quicker than classical algorithms.

In addition to these advantages, having both states like 0 and 1 gives many other opportunities. For example infinite information can be stored in a qubit. The entire Hamlet novel can be stored in a qubit, because of superposition. However, if one observes that qubit, all information will be lost. It is going to act as classical bit. Having said that, however, there is no need to be pessimistic. Quantum computing is growing fast. Algorithms and simulations are evolving.

This study has showcased that Grover's Algorithm is significantly faster than any classical algorithm. There is an example of the difference between classical and Grover's algorithm.

Considering a set of data, 1024 unsorted data and it is needed to find the special entry.

When it is unsorted data, the result will be found in $1024/2 = 512$ steps. If all data are sorted then $n = 1024 = 2^{10}$ so, in approximately 10 steps special entry would be found. On the other hand, with Grover Algorithm it is $\sqrt{N} = \sqrt{2^{10}}$ and this is equal to 32. After 32 iterations the result is found. For Grover's Algorithm no one needs to have sorted data.

When using the classical methods, the special entry is found/calculated in 512 steps but with Grover Algorithm it can be calculated in only 32. Furthermore, if the number of entry increases Grover's algorithm will be more successful to give the answer in a shorter period of time. Because it has a quadratic speed up.

REFERENCES

[1] Bell, J. S. (1964). On the einstein podolsky rosen paradox.*Physics Physique Fizika, 1(3)*, 195.

[2] Benioff, P. (1980). The computer as a physical system: A microscopic quantum mechanical Hamiltonian model of computers as represented by Turing machines. *Journal of statistical physics*, 22(5), 563-591.

[3] Demmer, M., Fonseca R., Koushanfar F. (2001). *Richard Feynman: Simulating Physics with Computers.* CS294: Reading the Classics.

[4] Deutsch, D. (1985). Quantum theory, the Church–Turing principle and the universal quantum computer. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, 400(1818), 97-117.

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

[6] Dirac, Paul A. M. (1958). *The Principles of Quantum Mechanics*. Oxford University Press.

[7] Kutluer, K. (2019). *Kuantum süperpozisyon ve çift yarık deneyi nedir?* https://duzensiz.org/

[8] Einstein, A., Podolsky, B., Rosen, N. (1935). Can quantum-mechanical description of physical reality be considered complete?. *Physical review, 47*(10), 777.

[9] Feynman, R. P. (1981). Simulating physics with computers. *Int. J. Theor. Phys*, 21(6/7).

[10] Grover, L. K. (1996, July). A fast quantum mechanical algorithm for database search. *In Proceedings of the twenty-eighth annual ACM symposium on Theory of computing* (pp. 212-219).

[11] IBM. (2019). *IBM Quantum Computers Getting Started Notebooks*. World Scientific. https://quantum-computing.ibm.com/

[12] IBM. (2019). *IBM Quantum Computers Documentation and Support*. World Scientific. https://quantum-computing.ibm.com/support

[13] Kaye, P., Laflamme, R., Mosca, M. (2007). *An introduction to quantum computing*. Oxford university press.

[14] Nielsen, M. A., Chuang, I. (2002). *Quantum computation and quantum information*. Cambridge University Press

[15] Schrödinger, E. (1925). Die Erfüllbarkeit der Relativitätsforderung in der klassischen Mechanik. *Annalen der Physik, 382*(11), 325-336.

[16] Shor, P. W. (1996). Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. *IEEE Computer Society Press*

[17] Simon, D. R. (1997). On the power of quantum computation. *SIAM journal on computing*, 26(5), 1474-1483.

[18] Strubell, Emma. (2011). *COS498 Chawathe Spring, 13*, 19.

[19] Turing, A. M. (1937). On computable numbers, with an application to the Entscheidungsproblem. *Proceedings of the London mathematical society*, 2(1), 230-265.

[20] van der Lans, M. (2018). Quantum Algorithms and their Implementation on Quantum Computer Simulators. *Delft University of Tehcnology*.

APPENDIX

Steps in Grover’s Algorithms by circuits

Steps in Grover’s Algorithms by circuits:

An example with Qiskit. Qiskit is the IBM package which is public [11]. Here is a detailed explanation of Qiskit codes with the circuit. All circuits are showing in Figure A.1

Grover’s algorithm is followed as:

1) All qubits will be in superposition state with using Hadamard gate.

2) Implement the oracle to mark that special item. Oracle is the part that a phase inversion.

3) Implement an amplification circuit to find the marked item by decreasing other states amplitudes. Diffusion part will repeat $\frac{\pi}{4}\sqrt{N}$ times.

4) At the end measurement. See Figure A.2

Connecting to IBM Quantum Computer with Qiskit:

Here is the beginning of coding:

Listing A.1: Grover’s Algorithm - Phyton

```
exampleIn [1]: import numpy as np
In [2]: from qiskit import( QuantumRegister ,
ClassicalRegister , QuantumCircuit ,
: execute ,IBMQ, Aer)
In [3]: import math
In [4]: IBMQ.save_account('24fc*
/home/fgnyilmaz/.local/lib/python3.6/*
```

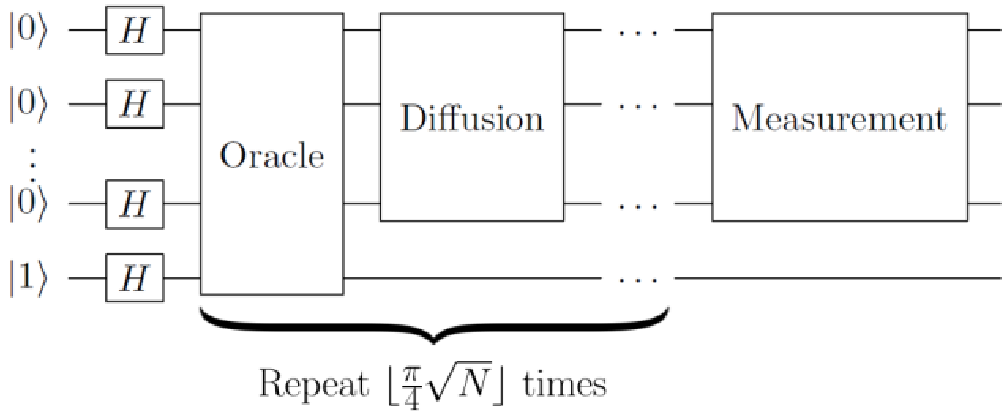


Figure A.1: The general schematic overview of the circuit for Grover's Algorithm. [20]

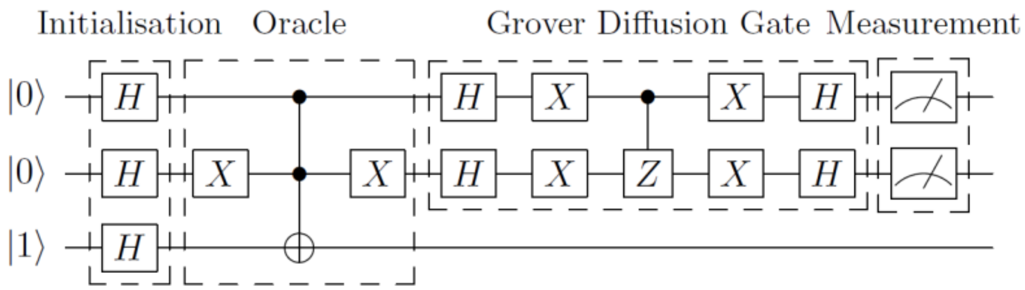


Figure A.2: All circuits of Grover's Algorithm for 2-qubits.[20]

```
warnings.warn('Credentials already present. '

In[5]:IBMQ.load_account()

/usr/lib/python3/dist−packages/s∗

In[6]:␣from␣qiskit␣import␣compile

In[9]:␣provider␣=␣IBMQ.get_provider(hub='ibm−q')

In[10]:␣pi␣=␣math.pi

:q=␣QuantumRegister(4,'q')

:c=␣ClassicalRegister(4,'c')

:qc=␣QuantumCircuit(q,c)
```

Initialising Circuit:

Here the first step of doing initialisation of qubit. One have this state with Hadamard

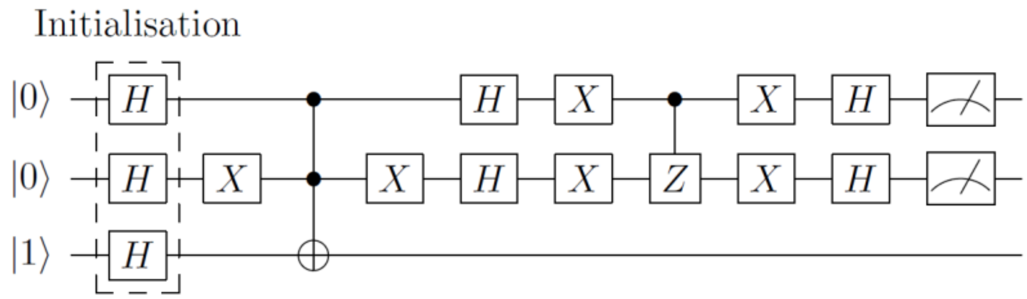


Figure A.3: The first step is ‘Initialisation’.[20]

gate as mentioned before. Hadamard gate on each qubit as Figure A.3.

```
Initialising Circuit...

In [12]: qc.h(q[0])

: qc.h(q[1])

: qc.h(q[2])

: qc.h(q[3])

Out[12]: <qiskit.circuit.instructionset.

InstructionSet at 0x7fa23a2a2eb8>
```

Having all states in Oracle:

```
Preparing Oracle circuit....


In [14]: qc.x(q[0])

: qc.x(q[1])

: qc.x(q[2])

: qc.x(q[3])

Out[14]: <qiskit.circuit.instructionset.

InstructionSet at 0x7fa23a2a2c18>
```

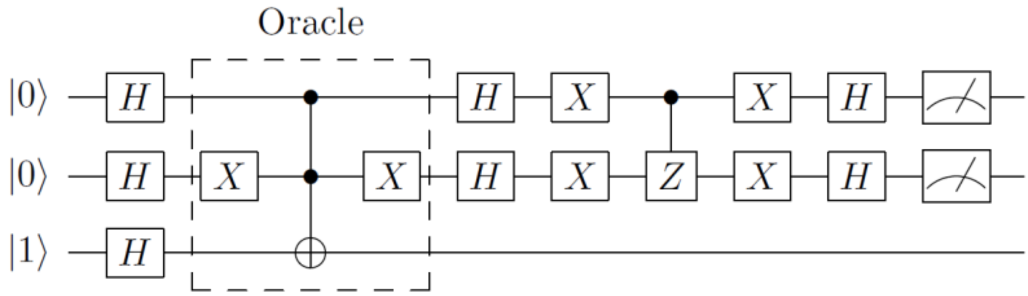


Figure A.4: The Oracle showed by dashed rectangle.[20]

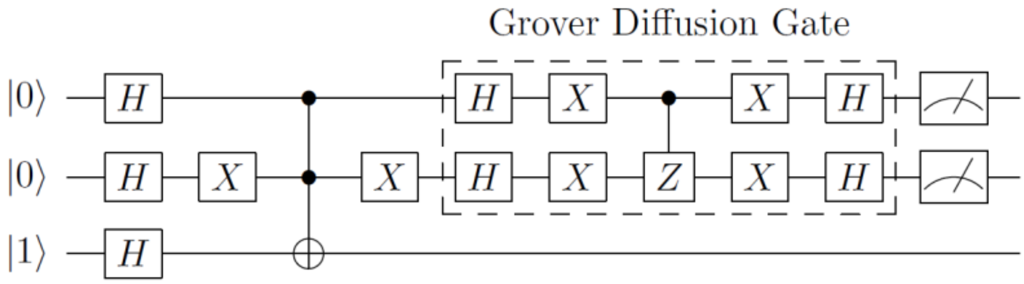


Figure A.5: The Grover diffusion gate.[20]

Measurement:

```
In [65]: qc.barrier(q)
: qc.measure(q[0], c[0])
: qc.measure(q[1], c[1])
: qc.measure(q[2], c[2])
: qc.measure(q[3], c[3])
Out[65]: <qiskit.circuit.instructionset.
InstructionSet at 0x7fa23a190cf8>
```

```
In [66]: backend = provider.get_backend
```

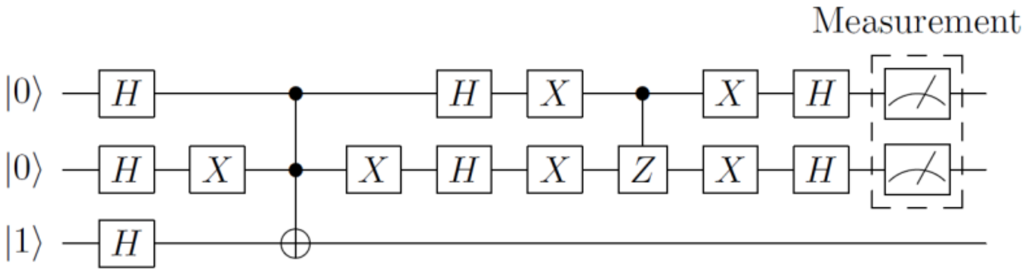


Figure A.6: The measurements are performed on first and second qubits.[20]

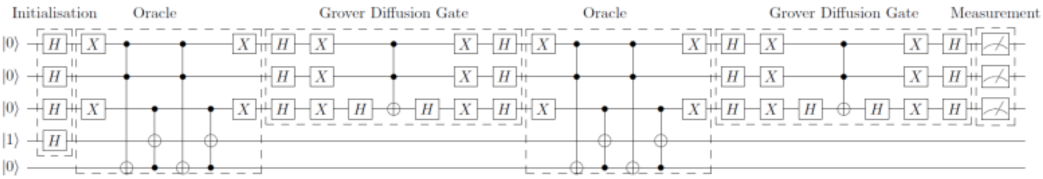


Figure A.7: Whole Grover's circuit with 3-qubits. [20]

```
('ibmq_qasm_simulator')

: print('\nExecuting job....\n')

: job = execute(qc, backend, shots=100)

Executing job....

In [67]: result = job.result()

: counts = result.get_counts(qc)


In [68]: print('RESULT: ',counts,'\n')

: print('Press any key to close')

: input()

RESULT:  {'0100': 1, '0101': 1,

'1011': 47, '1110': 1,

          '0111': 3, '1100': 2, '1000': 1,

          '1111': 2, '0110': 2, '1101': 2, '1010': 3}
```

Examples of two states by IBM Quantum Simulator

State of $|11\rangle$ circuit by IBM Quantum Simulator:

The circuit of Grover's algorithm that was executed on IBM's 2-qubit quantum computer and ibmqx4 was used [12]. These examples are on only 2-qubits and started with $|11\rangle$ state and then $|10\rangle$ state.

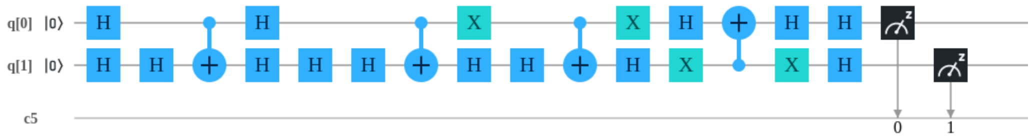


Figure A.8: IBM Quantum simulator for $|11\rangle$ state [12].

Each example was executed by 1024 shots. This is the result of $|11\rangle$ state Figure A.9. It isobvious result is the same as what we were looking for.

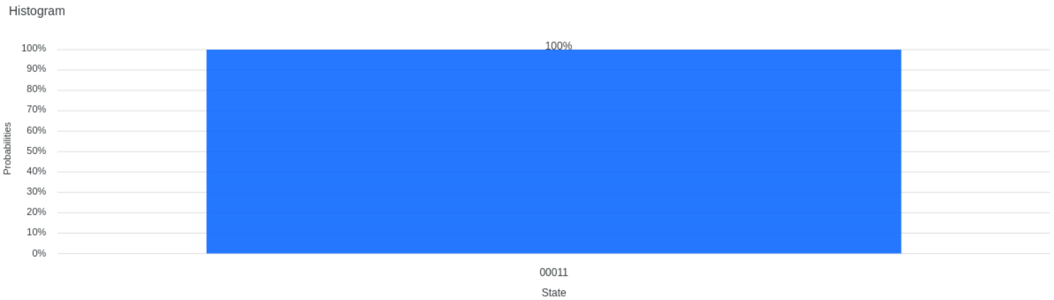


Figure A.9: The result of $|11\rangle$ state from IBM Quantum simulator [12].

And second example with $|10\rangle$. To having $|10\rangle$ state one uses X gates on first qubit.

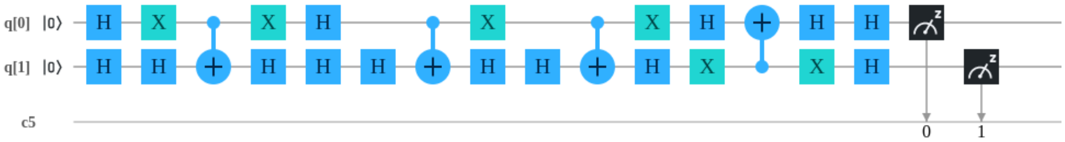


Figure A.10: IBM Quantum simulator for $|10\rangle$ state [12].

And here again result of $|10\rangle$ state.

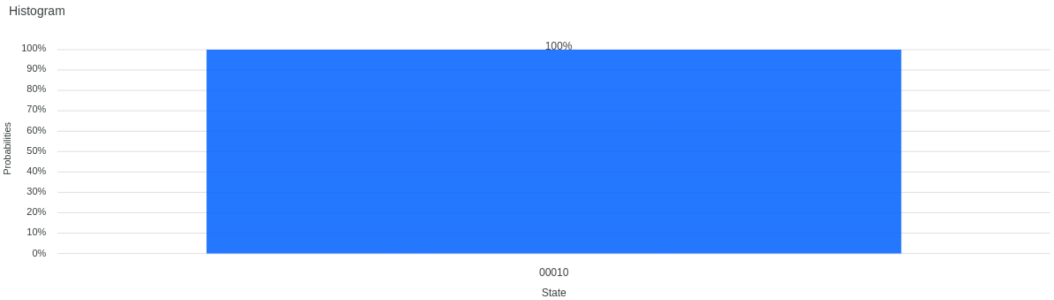


Figure A.11: The result of $|10\rangle$ state from IBM Quantum simulator [12].