VISVESVARAYA TECHNOLOGICAL UNIVERSITY,
BELAGAVI

Seminar Report
on

QUANTUM LOGIC GATES

In partial fulfillment of requirements for the degree of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE & ENGINEERING

SUBMITTED BY:
Sumantha Shettigar
4MW19CS122

Under the Guidance of

Mr. Shrinivasa
Asst. Professor (Selection Grade)
Dept. of CSE
SMVITM Bantakal

Ms. Yashaswini Jogi
Asst. Professor (Senior)
Dept. of CSE
SMVITM Bantakal

Department of Computer Science and Engineering
Shri Madhwa Vadiraja Institute of Technology & Management
Vishwothama Nagar, Bantakal - 574 115
[2022-2023]

Shri Madhwa Vadiraja Institute of Technology & Management

Vishwothama Nagar, Bantakal - 574 115

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that the Seminar on topic **Quantum Logic Gates** has been successfully presented at **Shri Madhwa Vadiraja Institute of Technology** by **Sumantha Shettigar,** bearing **USN 4MW19CS122,** in partial fulfillment of the requirements for the *VIII Semester degree of* ***Bachelor of Engineering in Computer Science and Engineering*** *of Visvesvaraya Technological University, Belagavi* during academic year 2022-2023.

Mr. Shrinivasa
Seminar Coordinator
Asst. Professor (Selection Grade)
Dept. of CSE
SMVITM Bantakal

Ms. Yashaswini Jogi
Seminar Coordinator
Asst. Professor (Sr)
Dept. of CSE
SMVITM Bantakal

Dr. Soumya J Bhat
Assoc. Professor and
Head of the Department
Dept. of CSE
SMVITM Bantakal

ACKNOWLEDGEMENT

First of all, I am indebted to the GOD ALMIGHTY for giving me an opportunity to excel in my efforts to complete this seminar on time.

I am extremely grateful to **Prof. Dr. Thirumaleshwara Bhat**, Principal, **SMVITM** and **Prof. Dr. Soumya J. Bhat**, Head of Department, Dept. of Computer Science and Engineering for providing all required resources for the successful completion of my seminar.

I express my deepest gratitude and respect to my seminar coordinators **Mr. Shrinivasa**, Asst. Professor (Selection Grade) **and Ms. Yashaswini Jogi**, Asst. Professor (Senior), Department of Computer Science and Engineering, for their extended support and valuable guidance while doing this work.

I remain grateful to the co-operation and help rendered by the teaching and non-teaching staffof the Computer Science and Engineering Department. Lastly, I take this opportunity to offer my regards to all of those who have supported me directly or indirectly in the successful completion of this work.

SUMANTHA SHETTIGAR

4MW19CS122

ABSTRACT

Quantum logic gates are fundamental components of quantum computers that enable the manipulation of quantum bits or qubits. They differ from classical logic gates, which operate on classical bits, in that they take advantage of the principles of quantum mechanics to perform operations on qubits. In this paper, we provide an overview of quantum logic gates, including their basic principles, operations, and applications. We begin by introducing the concept of quantum computing and its potential applications in various fields. We then discuss the different types of quantum logic gates, such as the Pauli gates, Hadamard gate, and phase gate, and their operations. We also review various quantum algorithms, such as Grover's algorithm and Shor's algorithm, and how they can be implemented using quantum logic gates. We then discuss the methodology used for implementing quantum logic gates, including the use of quantum circuits. We also present the applications of quantum logic gates in various fields, such as cryptography, machine learning, and chemistry. Finally, we provide a summary of the current state of research on quantum logic gates, including their advantages and limitations, and offer recommendations for future research in this area. Our paper aims to provide a comprehensive overview of quantum logic gates and their potential applications, highlighting the importance of quantum computing in the development of new technologies

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION

Quantum computing is a rapidly growing field that has the potential to revolutionize computing as we know it. Unlike classical computing, which relies on bits to store and manipulate information, quantum computing uses quantum bits, or qubits, which can exist in a superposition of states. Quantum logic gates are the building blocks of quantum circuits, which manipulate qubits to perform quantum computations. These gates operate on the principles of quantum mechanics, such as superposition and entanglement, and allow for complex computations that are not possible with classical logic gates.

The purpose of this report is to provide an overview of quantum logic gates and their importance in quantum computing. We will examine the different types of quantum logic gates, including the Hadamard gate, CNOT gate, and Toffoli gate, and their functions in quantum circuits. We will also explore the concept of quantum entanglement and how it is used in quantum computing.

In addition, we will discuss the challenges associated with the implementation of quantum logic gates, including the issue of decoherence, which is the loss of quantum information due to interaction with the environment. We will also examine the current state of quantum computing technology and the potential applications of quantum logic gates in various fields, such as cryptography, chemistry, and machine learning.

Overall, the development of quantum logic gates has been instrumental in advancing the field of quantum computing, and their potential applications are vast and varied. This report will provide an in-depth look at the importance of quantum logic gates in quantum computing and their potential to transform computing as we know it.

Chapter 2

LITERATURE SURVEY

In this paper, they demonstrated the successful implementation of two-qubit quantum logic gates on a scalable semiconductor platform using silicon-based quantum dots. The authors achieved high gate fidelity and demonstrated the scalability of their approach, which is a key requirement for the development of practical quantum computers. This work is considered a significant step towards building large-scale, fault-tolerant quantum computers [1].

This paper proposes a scalable approach to implementing quantum logic gates using trapped ions. The authors use a hybrid quantum-classical algorithm to optimize the control parameters of the gates, which enables the gates to be more robust to experimental imperfections. The authors demonstrate the feasibility of their approach through simulations and experimental results [2].

This paper proposes an efficient method for implementing universal quantum gates on superconducting qubits. The authors use a combination of a global optimization algorithm and a quantum simulator to optimize the control parameters for the gates. The authors demonstrate the effectiveness of their approach through simulations and experimental results [3].

This paper proposes a method for synthesizing quantum gates using evolution strategies. The authors use a combination of classical simulation and quantum simulation to optimize the gate parameters. The authors demonstrate the effectiveness of their approach through simulations and experimental results [4].

This paper proposes a method for implementing fault-tolerant universal quantum gates using double-excited states. The authors use a combination of classical simulation and quantum simulation to optimize the gate parameters. The authors demonstrate the effectiveness of their approach through simulations and experimental results [5].

Chapter 3

METHODOLOGY

3.0 Abbreviations

Abbreviations used in this document are listed below.

$\mathbb{H}^n$	:	**n Dimensional Hilbert Space**
$\mathbb{C}^n$	:	**n Dimensional Complex Space**
$\mathbb{I}$	:	**Identity Matrix**
H	:	**Hadamard Matrix**
X	:	**Pauli X Matrix**
Y	:	**Pauli Y Matrix**
Z	:	**Pauli Z Matrix**
P	:	**Phase Shift Gate**
S	:	**S Gate**
T	:	**T Gate**
U	:	**Unitary Matrix**
$U^\dagger$	:	**Adjoint of Unitary Matrix (Hermitian)**

3.1 Welcome to Quantum

Classical computations are composed of a sequence of classical logic gates that manipulate a small set of classical bits at a time. Similarly, quantum computations can also be decomposed into a series of quantum logic gates, each operating on only a few qubits at once. However, the key difference lies in the fact that classical logic gates only manipulate the values of classical bits, which are either 0 or 1. In contrast, quantum gates can operate on complex multi-partite quantum states that may be entangled and in arbitrary superpositions of computational basis states. This enables quantum logic gates to be far more diverse than classical logic gates, allowing for more complex and powerful computations to be executed.

Just as classical computing involves connecting sets of logic gates to construct digital circuits, quantum computing leverages quantum logic gates to compute output states from input states, where the inputs are typically superposition states. These gates are mathematically described by unitary matrices, and their application involves multiplying the corresponding matrix by the state vector. For an 1-qubit gate, a 2x2 unitary matrix is required. On the other hand, a 2-qubit gate can be implemented with a 4x4 matrix.

3.2 Bloch Sphere Representation

The block sphere is a useful visualization tool for single quantum bits and unitary transform on them.

Bloch Sphere Representation

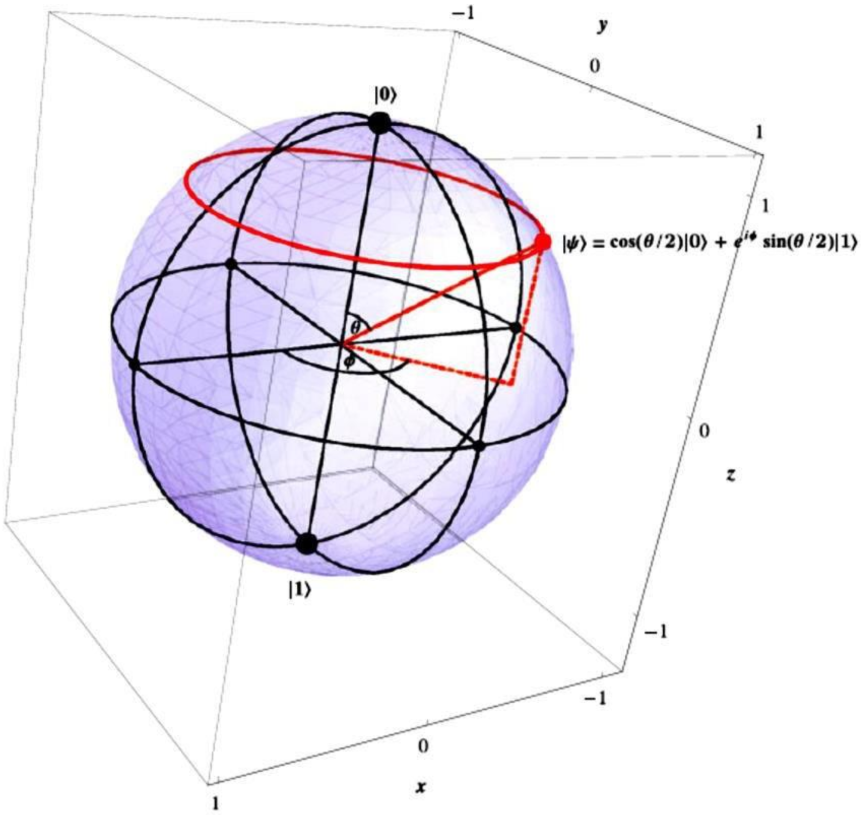

Fig 3.2.1

Bloch sphere showing the computational basis states |0 > and |1 >, and a general qubit state

$$|\Psi> = \text{Cos}\,\frac{\theta}{2}\,|0> + e^{i\phi}\text{Sin}\frac{\theta}{2}\,|1>$$

In the above picture, the North pole corresponds to the pure state |0> and the South pole corresponds to the (orthogonal) pure state |1>. All other points on the surface of the Bloch sphere correspond to the superposition states of the form $|\psi> = \alpha|0> + \beta\ |1>$ for all possible values of the complex numbers α *and* β such that

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

3.3 Quantum Logic Gates

By a quantum-logic gate, we shall mean a closed-system evolution(transformation) of the *n*- qubit state space $\mathbb{H}^n$. In particular, this means that no information is gained or lost during this evolution; thus, a quantum gate has the same number of input qubits as output qubits.

The ‘gates’ are unitary operations, we know that a unitary operator $\boldsymbol{U}$ is one where the adjointis equal to the inverse, meaning $\boldsymbol{U}^{\dagger} = \boldsymbol{U}^{-1}$. The defining relation for a unitary operator is thus

$$\boldsymbol{U}^{\dagger}\boldsymbol{U} = \boldsymbol{U}\boldsymbol{U}^{\dagger} = \mathbb{I}$$

The properties of unitary matrix ensures that we can always undo a quantum gate, i.e., that a quantum gate is logically reversible.

Quantum operators can be represented by matrices. Recall that a quantum gate with n inputs and outputs can be represented by a matrix of degree $\boldsymbol{2^n}$. For a single qubit, we require a matrix of degree $2^1 = 2$. That is, a quantum gate acting on a single qubit will be a 2×2 unitary matrix. A two-qubit gate can be implemented with a matrix of degree $2^2 = 4$ or a 4×4 matrix.

There are mainly 10 basic and most used Quantum gates that are essential bricks of quantum computing.

Operator	Gate(s)	Matrix
Pauli-X (X)	X	$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$
Pauli-Y (Y)	Y	$\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$
Pauli-Z (Z)	Z	$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$
Hadamard (H)	H	$\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$
Phase (S, P)	S	$\begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$
$\pi/8$ (T)	T	$\begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}$
Controlled Not (CNOT, CX)		$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$
Controlled Z (CZ)	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 (CCNOT, CCX, TOFF)		$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$

Fig 3.3.1

3.4 Single Qubit Gates

Single-qubit operations translate an arbitrary quantum state from one point on the Bloch sphere to another point by rotating the Bloch vector(spin) a certain angle about a particular axis. As shown in Fig. 2 and 3; there are several single-qubit operations, each represented by a matrix that describes the quantum operation in the computational basis represented by the eigenvectors of the Z operator, i.e., $|0>$ and $|1>$.

3.4.1 Pauli X - Gate

This gate is synonymous with the classical(reversible) NOT gate.

The X Operator is represented by the following matrix:

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

Hence,

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

So, with respect to the standards of computational basis, the X matrix acts as a NOT operator.

Truth Table:

Input	**Output**
|0>	|1>
|1>	|0>
α|0> + β|1>	α|1> + β|0>

Table 3.4.1

3.4.2 Pauli Y - Gate

The Y Operator is represented by the following matrix

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

Hence,

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

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

Truth Table:

Input	Output
|0>	$i\|1>$
|1>	$-i\|0>$
α|0> + β|1>	i α|1> $-i$ β|0>

Table 3.4.2

3.4.3 Pauli Z - Gate

The Z Operator is represented by the following matrix

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

Hence,

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

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

The Z operator is sometimes called the phase flip gate because it transforms a qubit.

Truth Table:

Input	Output
|0>	$\|0>$
|1>	$-\|1>$
α|0> + β|1>	α|0> - β|1>

Table 3.4.3

By using these Pauli's gates we can perform different actions on qubits, but we can't get the results other than |0> and |1>. To overcome this range limit, we use some other gates.

3.4.4 Hadamard Gate

The Hadamard H Operator is represented by the following matrix

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

Hence,

$$H\,|0> = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}\begin{pmatrix} 1 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{2}}\left[\begin{pmatrix} 1 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix}\right] = \frac{1}{\sqrt{2}}(|0> + |1>) = |+>$$

$$H\,|1> = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}\begin{pmatrix} 0 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ -1 \end{pmatrix} = \frac{1}{\sqrt{2}}\left[\begin{pmatrix} 1 \\ 0 \end{pmatrix} - \begin{pmatrix} 0 \\ 1 \end{pmatrix}\right] = \frac{1}{\sqrt{2}}(|0> - |1>) = |->$$

Truth Table:

Input	Output
|0>	$\frac{1}{\sqrt{2}}(\|0>+\|1>) = \|+>$
|1>	$\frac{1}{\sqrt{2}}(\|0>-\|1>) = \|->$
α|0> + β|1>	$\frac{(\alpha+\beta)\|0>}{\sqrt{2}} + \frac{(\alpha-\beta)\|1>}{\sqrt{2}} = \alpha\|+> + \beta\|->$

Table 3.4.4

Here, $|\alpha|^2 = (\frac{1}{\sqrt{2}})^2 = \frac{1}{2}$ and $|\beta|^2 = (\frac{1}{\sqrt{2}})^2 = \frac{1}{2}$ => $\frac{1}{2} + \frac{1}{2} = 1$

Therefore, $|\alpha|^2 + |\beta|^2 = 1$.

3.4.5 Phase Gate (S or P)

The Phase gate S Operator is represented by the following matrix

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

Hence,

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

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

Truth Table:

Input	Output
|0>	$\|0>$
|1>	$i\|1>$
α|0> + β|1>	$\alpha\|0> + i\beta\|1>$

Table 3.4.5

3.4.6 T - Gate

The T gate Operator is represented by the following matrix

$$T = \begin{pmatrix} 1 & 0 \\ 0 & e^{\frac{i\pi}{4}} \end{pmatrix}$$

Hence,

$$T|0> = \begin{pmatrix} 1 & 0 \\ 0 & e^{\frac{i\pi}{4}} \end{pmatrix}\begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} = |0>$$

$$T\,|1> = \begin{pmatrix}1 & 0\\ 0 & e^{\frac{i\pi}{4}}\end{pmatrix}\begin{pmatrix}0\\1\end{pmatrix} = \begin{pmatrix}0\\ e^{\frac{i\pi}{4}}\end{pmatrix} = e^{\frac{i\pi}{4}}|1>$$

Truth Table:

Input	**Output**
$\|0>$	$\|0>$
$\|1>$	$e^{\frac{i\pi}{4}}\|1>$
$\alpha\|0> + \beta\|1>$	$\alpha\,\|0> + e^{\frac{i\pi}{4}}\beta\|\,1>$

Table 3.4.6

These are few of the most used Single Qubit Gates. These Single Qubit Gates do not give much computational advantage, as they have limited applications. Hence, we go further for Multi–Qubit Gates.

3.5 Multi Qubit Gates

There are mainly four Multi-Qubit Gates. These gates operate on multi-qubits. Multi-Qubits are obtained by operation of single qubits with itself or with another qubit. Let's move to the case of two-qubit gates. In this section the notion of a controlled gate will allow us to implement an if -then–else type of construct with a quantum gate. Controlled quantum (or controlled unitary) gates work in this type of fashion, using a control qubit to determine whether or not a specified unitary action is applied to a target qubit.

3.5.1 CNOT - Gate

This gate is synonymous with the Conditional (Controlled) NOT gate. Hence, can also be called as Controlled X – Gate (CX).

The CNOT Operator takes combination of two qubits as input. The combination of two qubits is obtained by tensor product of two different or same qubits.

We know that, $|0> = \begin{pmatrix}1\\0\end{pmatrix}$ and $|1> = \begin{pmatrix}0\\1\end{pmatrix}$

Then the tensor product of these two state column vectors is obtained as

$$|0>\otimes|0> = |00> = \begin{bmatrix}0\begin{pmatrix}0\\1\end{pmatrix}\\ 0\begin{pmatrix}0\\1\end{pmatrix}\end{bmatrix} = \begin{pmatrix}1\\0\\0\\0\end{pmatrix}$$

$$and$$

$$|0>\otimes|1> = |01> = \begin{bmatrix}0\begin{pmatrix}0\\1\end{pmatrix}\\ 1\begin{pmatrix}0\\1\end{pmatrix}\end{bmatrix} = \begin{pmatrix}0\\1\\0\\0\end{pmatrix}$$

In the same way we get,

$$|10> = \begin{pmatrix}0\\0\\1\\0\end{pmatrix} \ and \ |11> = \begin{pmatrix}0\\0\\0\\1\end{pmatrix}$$

The tensor product of 2 single qubits will have 4 elements, and hence, the tensor product of n single qubits will have 2^n elements (dimensions).

As discussed earlier, CNOT operator operates on two qubits, where first qubit acts as control (⬤) and the second qubit acts as the target (⊕).

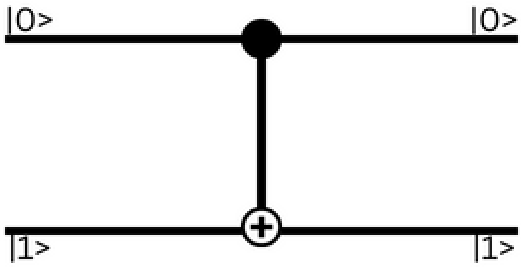

Fig. 3.5.1 (a)

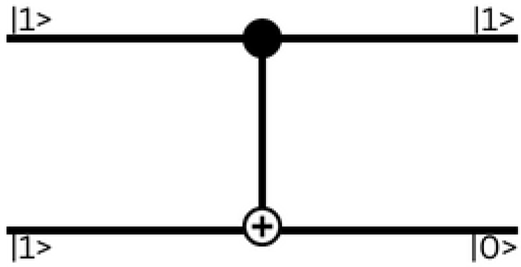

Fig. 3.5.1 (b)

Truth table:

Input	**Output**
|00>	|00>
|01>	|01>
|10>	|11>
|11>	|10>
|AB>	B⊕A

Table 3.5.1

By observation it is known that when the control qubit is |0> the gate won't affect the target, but when the control qubit is |1> the target is altered. Control qubit never changes through the gate.

The matrix representation of CNOT will be written as 4X4 matrix since it operates two qubits of two rows each.

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

Example-1, consider |01> as input to a CX gate:

$$\boldsymbol{CNOT}\,|01> = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} = |01>$$

The gate is inactive since control is |0>. When the control is |1> the gate will affect the target.

Consider another example.

Example-2, consider |11> as input to a CX gate:

$$\boldsymbol{CNOT}\,|11> = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} = |10>$$

So, we can conclude that to the standards of computational basis the CNOT gate will act like if then conditional control flow.

3.5.2 SWAP - Gate

While playing with multi qubit system, sometime we may need to interchange the states. SWAP gate appears there to help with this operation.

The SWAP gate made up of many CNOT gates as it is the continuation of CNOT (CX) gate. which helps to interchange control (⬤) and target (⨁) as per requirement.

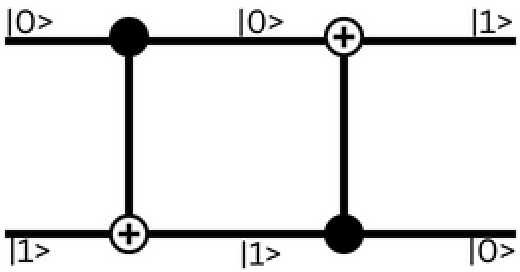

Fig. 3.5.2 (a)

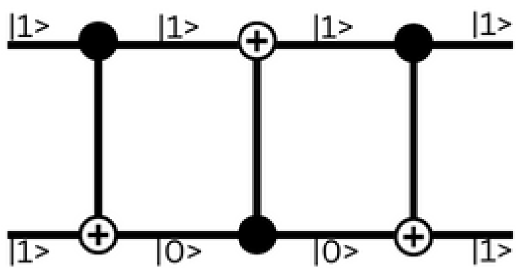

Fig. 3.5.2 (b)

Truth table:

Input	Output
|00>	|00>
|01>	|10>
|10>	|01>
|11>	|11>

Table 3.5.2

Some swap operation need only two CNOT gates (Fig. 3.5.2 (a)), where some swap operation need three or more CNOT gates (Fig. 3.5.2 (b)). It is always better to use minimum three CNOT gates to build a SWAP gate.

The matrix representation of SWAP operation will be written as 4X4 matrix since it operates two qubits of two rows each.

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

Example-1, consider |01> as input to a SWAP gate:

$$\boldsymbol{SWAP}\ |01> = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} = |10>$$

Consider another example.

Example-2, consider |10> as input to a SWAP gate:

$$\boldsymbol{SWAP}\ |10> = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} = |01>$$

So. We can conclude that SWAP gate swaps qubit state between two input state qubits.

3.5.3 CZ - Gate

We, know that CNOT gate will act like bit-flip operator. Unlike the CNOT gate sometime we may need to change the phase. CZ (Controlled Z) gate comes there to help with this operation.

In this CZ gate, we know that control (●) is inactive when first bit is |0>. But when the first bit is |1> and second bit is |0> the gate remains inactive. The CZ gate acts only when both qubits state are |> i.e., input = |11>.

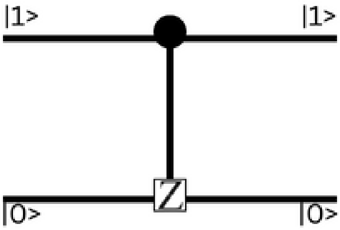

Fig. 3.5.3 (a)

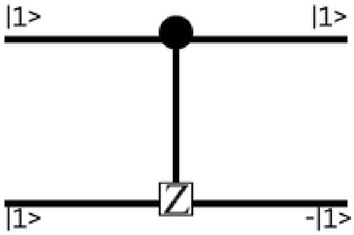

Fig. 3.5.3 (b)

Truth table:

Input	Output
|00>	|00>
|01>	|01>
|10>	|10>
|11>	−|11>

Table 3.5.3

We can observe that in Fig. 3.5.3 (a), control is active but Z gate is not ready for action. In Fig. 3.5.2 (b), the gate is active since control is |1> and Z gate also activated to perform phase-flip.

The matrix representation of CZ operation will be written as,

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

Example-1, consider |01> as input to a CZ gate:

$$\boldsymbol{U_Z}|01> = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}\begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} = |01>$$

Consider another example.

Example-2, consider |11> as input to a CZ gate:

$$\boldsymbol{U_Z}|11> = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}\begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ -1 \end{pmatrix} = |01>$$

So. We can conclude that CZ gate changes qubit phase when input is |11>.

3.5.4 Toffoli - Gate

Toffoli gate is also called as the Double Controlled CNOT gate. It takes three qubits combinations as input. We know that the combination of qubits is obtained by tensor product of two or more different or same qubits.

We know that,

$$|00> = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \; and \; |01> = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

Then,

$$|000> = |0> \otimes |00> = \begin{bmatrix} 1 & \begin{pmatrix} 1\\0\\0\\0 \end{pmatrix} \\ 0 & \begin{pmatrix} 1\\0\\0\\0 \end{pmatrix} \end{bmatrix} = \begin{pmatrix} 1\\0\\0\\0\\0\\0\\0\\0 \end{pmatrix}$$

$$and$$

$$|001> = |0> \otimes |01> = \begin{bmatrix} 1 & \begin{pmatrix} 0\\1\\0\\0 \end{pmatrix} \\ 0 & \begin{pmatrix} 0\\1\\0\\0 \end{pmatrix} \end{bmatrix} = \begin{pmatrix} 0\\1\\0\\0\\0\\0\\0\\0 \end{pmatrix}$$

Similarly, the 8 combinations are calculated.

$\|000>$	$\|001>$	$\|010>$	$\|011>$	$\|100>$	$\|101>$	$\|110>$	$\|111>$
$\begin{pmatrix} 1\\0\\0\\0\\0\\0\\0\\0 \end{pmatrix}$	$\begin{pmatrix} 0\\1\\0\\0\\0\\0\\0\\0 \end{pmatrix}$	$\begin{pmatrix} 0\\0\\1\\0\\0\\0\\0\\0 \end{pmatrix}$	$\begin{pmatrix} 0\\0\\0\\1\\0\\0\\0\\0 \end{pmatrix}$	$\begin{pmatrix} 0\\0\\0\\0\\1\\0\\0\\0 \end{pmatrix}$	$\begin{pmatrix} 0\\0\\0\\0\\0\\1\\0\\0 \end{pmatrix}$	$\begin{pmatrix} 0\\0\\0\\0\\0\\0\\1\\0 \end{pmatrix}$	$\begin{pmatrix} 0\\0\\0\\0\\0\\0\\0\\1 \end{pmatrix}$

Table 3.5.4 (a)

As discussed earlier, Toffoli gate operates on three qubits, where first two qubit acts as control (⬤) and the third qubit acts as the target (⊕).

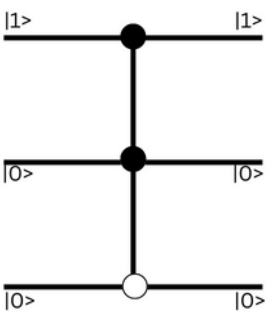

Fig. 3.5.4 (a)

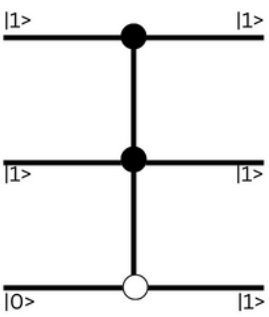

Fig. 3.5.4 (b)

Truth table:

Input	Output
|000>	|000>
|001>	|010>
|010>	|001>
|011>	|011>
|100>	|100>
|101>	|101>
|110>	|111>
|111>	|110>

Table 3.5.4 (b)

The Toffoli gate three qubits as input where first two qubits are control qubits and the last one is the target. In Fig. 3.5.4 (a), the gate is inactive where in Fig. 3.5.4 (b), the gate is active and alters the target. Because, the gate needs both controls to be in |1> state to get activated.

The matrix representation of Toffoli operation will be written as 8X8 matrix since it operates three qubits of two rows each (2^3 dimensions).

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

Example-1, consider |010> as input to a Toffoli gate:

$$\boldsymbol{Toffoli}\,|010> = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} = |010>$$

Here, the control qubits are |0> and |1>. So, the gate is inactive and did not alter the target qubit.

Let’s consider another example.

Example-2, consider |110> as input to a Toffoli gate:

$$\boldsymbol{Toffoli}\,|110> = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = |111>$$

Here, the control qubits are |1> and |1>. So, the gate is active and changed the target bit.

So. We can come to a conclusion that Toffoli gate changes the target qubit state only when both the control qubits are in |1> state.

3.6 Drawbacks of Quantum Computing

- Very difficult to monitor their interactions. Quantum systems are highly sensitive to noise and environmental interference, which can cause errors in quantum gates and result in computational errors.

- The quantum information will spread outside the quantum computer and be lost into the environment, thus spoiling the computation. This process is called de-coherence.

- The number of operations that can be performed before the information is lost due to de-coherency is therefore limited.

- Quantum chips must be kept at very low temperature to create super positions and entanglement of qubits

- Quantum gates are typically probabilistic, meaning that the output of a gate is not deterministic and may vary depending on the measurement of the qubits. Hence repeated operations are required to get a correct answer.

- Quantum gates are subject to the no-cloning theorem, which states that it is impossible to create an exact copy of an arbitrary unknown quantum state.

- The development of quantum algorithms and the design of quantum circuits require specialized knowledge and expertise, which may limit their accessibility to a wider audience.

- Quantum gates are currently limited by the size and complexity of current quantum hardware, which may require significant advancements in technology to achieve practical and commercially viable applications.

- Finally, quantum computing is currently in its early stages of development, and there are still many challenges that need to be overcome before it can become a practical and commercially viable technology.

Chapter 4

APPLICATION AND RESULTS

- Cryptography: Quantum logic gates have the potential to significantly enhance the security of cryptographic protocols by exploiting the properties of entanglement and superposition. For example, the quantum key distribution (QKD) protocol uses quantum gates to generate and transmit secure encryption keys.

- Simulation: Quantum logic gates can be used to simulate complex quantum systems that are difficult to study using classical computers. This has applications in fields such as material science, chemistry, and high-energy physics.

- Optimization: Quantum logic gates can be used to solve optimization problems more efficiently than classical algorithms, which has applications in fields such as finance, logistics, and transportation.

- Machine learning: Quantum logic gates can be used to enhance the performance of machine learning algorithms by exploiting the properties of quantum entanglement and superposition. This has applications in fields such as image recognition, natural language processing, and data analysis.

- Quantum error correction: Quantum logic gates can be used to implement error-correction techniques that help to mitigate the effects of noise and decoherence in quantum systems. This is essential for the development of robust and fault-tolerant quantum computing systems.

- Commercial applications: Quantum logic gates have the potential to revolutionize a wide range of industries, from finance and pharmaceuticals to transportation and logistics. For example, quantum computing can be used to optimize supply chain management, accelerate drug discovery, and improve financial modeling.

- Milestones: The development of quantum logic gates has led to several significant milestones in the field of quantum computing. For example, the implementation of the first quantum algorithm (Shor's algorithm) demonstrated the potential of quantum computing to solve problems that are intractable using classical algorithms. Additionally, the development of the first quantum error-correcting code demonstrated the feasibility of building fault-tolerant quantum computing systems.

- Overall, quantum logic gates have the potential to revolutionize various fields and industries, but their practical implementation requires significant advancements in technology and the development of new algorithms and techniques.

Chapter 5

CONCLUSION AND FUTURE ENHANCEMENTS

1. Quantum logic gates are a vital component of quantum computing, enabling the manipulation of quantum states and allowing for the implementation of quantum algorithms.
2. The Hadamard, CNOT, and Toffoli gates are just a few examples of the many types of quantum logic gates that can be used to construct quantum circuits.
3. The Bloch sphere provides a visual representation of the quantum state of a single qubit and is an essential tool for understanding and visualizing quantum operations.
4. Researchers are exploring the development of new types of quantum logic gates and algorithms that can take advantage of the unique properties of quantum computing.
5. These advancements could lead to significant breakthroughs in fields such as cryptography, machine learning, and drug discovery.
6. Additionally, there is a growing interest in developing more robust error-correction techniques and fault-tolerant quantum computing systems to overcome the challenges posed by noise and decoherence in quantum systems.
7. As research in this field continues to progress, the potential for quantum computing to revolutionize various fields and industries becomes more apparent.
8. However, there are still significant challenges that need to be overcome before quantum computing can become a practical and commercially viable technology.
9. These challenges include the development of scalable and reliable hardware, the improvement of error correction and fault-tolerant techniques, and the design of new quantum algorithms for solving real-world problems.
10. Despite these challenges, the potential of quantum computing to transform various fields and industries is too significant to ignore, and research in this area is likely to continue to accelerate in the coming years.

REFERENCES

[1] “Universal quantum gates on a scalable semiconductor platform” by M. Veldhorst, C. H. Yang, J. C. C. Hwang, W. Huang, J. P. Dehollain, J. T. Muhonen, S. Simmons, A. Laucht, F. E. Hudson, K. M. Itoh, A. Morello, and A. S. Dzurak, Nature (2015).

[2] “Scalable Quantum Logic Gates with Trapped Ions” by J. D. Sterk et al. (2020).

[3] “Efficient Universal Quantum Gates for Superconducting Qubits” by Y. Ge et al. (2021).

[4] “Gate Synthesis Using Evolution Strategies” by N. Sheth et al. (2021).

[5] “Fault-Tolerant Universal Quantum Gates with Double-Excited States" by X. Li et al. (2021).