Bachelor Thesis Project 1

Simulating Quantum Circuits using the Qiskit library

Submitted By -

Hardik Soni
20CS30023

Supervised By -

Professor Indranil Sengupta
Department of Computer Science and Engineering

Introduction to Qiskit

Qiskit is an open-source **software development kit** (SDK) for working with quantum computers at the level of circuits, pulses, and algorithms. It provides tools for creating and manipulating quantum programs and running them on **prototype quantum devices** on the IBM Quantum Platform or on **simulators on a local computer**. Qiskit is developed by IBM Research and the Qiskit community, and is released under the Apache 2.0 license. It is one of the most popular quantum computing SDKs available, and is used by researchers, developers, and educators around the world.

Quantum Circuit Construction

Qiskit provides a variety of tools for constructing quantum circuits, including a circuit compiler, optimizer, and transpiler. It provides a library of quantum algorithms for a variety of tasks, including machine learning, optimization, and search which aids the circuit construction.

Quantum Circuit Simulation

Qiskit provides a simulator that can be used to simulate the behavior of quantum circuits on a variety of hardware backends. It also provides a variety of tools for visualizing quantum circuits and results.

Quantum backend support

Qiskit supports a variety of quantum hardware backends, including superconducting qubits, trapped ions, and photonic qubits. It additionally provides a variety of tools for developing and testing quantum programs, such as a debugger and profiler.

Motivation

- Quantum computing is a rapidly developing field with the potential to revolutionize many industries. Simulating quantum circuits is a key step in developing and testing new quantum algorithms. The primary motivation behind this thesis project is to work with development and simulation of these quantum circuits, who recently have contributed in “quantum revolution” (Rapid recent use of quantum concepts in modern scientific research).

- Qiskit is a popular and well-supported open-source library for quantum computing. It provides a high-level interface for designing, compiling, and running quantum circuits on a variety of backends, including simulators.

- The extension of motivation for this project involves recognizing the profound impact that Simulation of Quantum Algorithms can have on its further development and its benefits:

    - Simulating quantum circuits can help you to gain a deeper understanding of quantum mechanics and quantum computing. By simulating different circuits and observing their results, you can learn about concepts such as superposition, entanglement, and interference.

    - Simulating quantum circuits can be used to solve a variety of problems in physics, chemistry, and materials science. For example, you can simulate the behavior of molecules and materials to develop new materials with desirable properties.

Objectives

GOAL : To develop and test the simulation of quantum circuits and algorithms that are being made use into recent developments of quantum field.

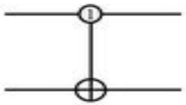
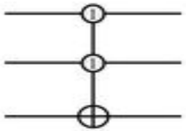
Defining Domain of Problem Statement: This involves identifying the specific aspects of quantum computing and Qiskit that will be investigated. For example, the problem statement could focus on the following: Simulating specific quantum circuits, such as the Deutsch-Jozsa algorithm or the Shor's algorithm, Comparing the performance of different quantum simulators.

Choice of Circuit: Once the domain of the problem statement has been defined, the next step is to choose the specific quantum circuits that will be simulated. This should be done carefully, taking into account the following factors:The complexity of the circuits, availability of resources for simulating the circuits and relevance of the circuits to the problem statement.

Design: Once the quantum circuits have been chosen, the next step is to design the simulation experiments. This involves determining the following: The parameters that will be varied in the simulations, metrics that will be used to evaluate the results of the simulations and the methods that will be used to analyze the data.

Plots and Analyzing the Results : Plots can be used to verify the working of the simulations and to analyze the results. For example, plots can be used to: Visualize the evolution of the quantum state of the circuit over time, Identify the eigenstates and eigenvalues of the circuit Hamiltonian and Compare the results of the simulation to the theoretical predictions.

Quantum Gates

Gate	Equation	Matrix	Transform	Notation
Identity (I)	$I = \vert 0\rangle\langle 0\vert + \vert 1\rangle\langle 1\vert$	$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$	$I\vert 0\rangle = \vert 0\rangle$ $I\vert 1\rangle = \vert 1\rangle$	
Pauli-X (X or NOT)	$X = \vert 0\rangle\langle 1\vert + \vert 1\rangle\langle 0\vert$	$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$	$X\vert 0\rangle = \vert 1\rangle$ $X\vert 1\rangle = \vert 0\rangle$	
Hadamard (H)	$H = \frac{\vert 0\rangle + \vert 1\rangle}{\sqrt{2}}\langle 0\vert + \frac{\vert 0\rangle - \vert 1\rangle}{\sqrt{2}}\langle 1\vert$	$\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$	$H\vert 0\rangle = \frac{1}{\sqrt{2}}(\vert 0\rangle + \vert 1\rangle)$ $H\vert 1\rangle = \frac{1}{\sqrt{2}}(\vert 0\rangle - \vert 1\rangle)$	
Controlled-NOT (CNOT)	$\mathbf{CNOT} = \vert 0\rangle\langle 0\vert \otimes I + \vert 1\rangle\langle 1\vert \otimes X$	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$	$\mathbf{CNOT}\vert 00\rangle = \vert 00\rangle$ $\mathbf{CNOT}\vert 01\rangle = \vert 01\rangle$ $\mathbf{CNOT}\vert 10\rangle = \vert 11\rangle$ $\mathbf{CNOT}\vert 11\rangle = \vert 10\rangle$	
Toffoli (T or CCNOT)	$\mathbf{T} = \vert 0\rangle\langle 0\vert \otimes I \otimes I + \vert 1\rangle\langle 1\vert \otimes \mathbf{CNOT}$	$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$	$\mathbf{T}\vert 000\rangle = \vert 000\rangle, \mathbf{T}\vert 001\rangle = \vert 001\rangle$ $\mathbf{T}\vert 010\rangle = \vert 010\rangle, \mathbf{T}\vert 011\rangle = \vert 011\rangle$ $\mathbf{T}\vert 100\rangle = \vert 100\rangle, \mathbf{T}\vert 101\rangle = \vert 101\rangle$ $\mathbf{T}\vert 110\rangle = \vert 111\rangle, \mathbf{T}\vert 111\rangle = \vert 110\rangle$	

Increment

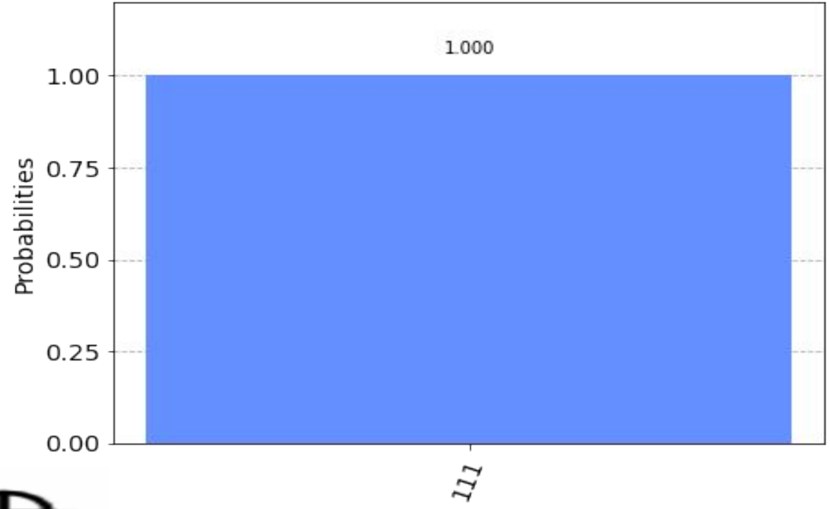
Takes an arbitrary input state |x⟩ on n and takes it to the |(x + 1)mod N⟩, where N = 2^n state in the binary representation, which can be done by applying the following circuit U. Let |x⟩ be an arbitrary state on n qubits,

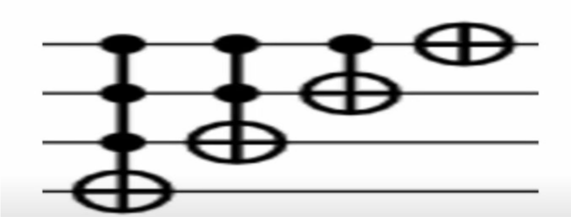
$$U|x\rangle = |(x+1) \bmod N\rangle$$

Where U defines an operation that applies multiple 1 multi-controlled X gates and flips the first qubit.

TABLE 2.5: Truth table for Quantum Increment Operation

Initial State $\|x\rangle$	Final State $U\|x\rangle$
$\|000\rangle$	$\|001\rangle$
$\|001\rangle$	$\|010\rangle$
$\|010\rangle$	$\|011\rangle$
$\|011\rangle$	$\|100\rangle$
$\|100\rangle$	$\|101\rangle$
$\|101\rangle$	$\|110\rangle$
$\|110\rangle$	$\|111\rangle$
$\|111\rangle$	$\|000\rangle$





Decrement

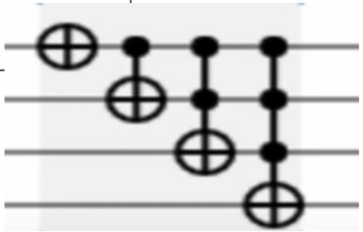
Takes an arbitrary input state |x⟩ on n and takes it to the |(x − 1)mod N⟩, where N = 2^n state, n in the binary representation, which can be done by applying the following circuit U. Let |x⟩ be an arbitrary state on n qubits,

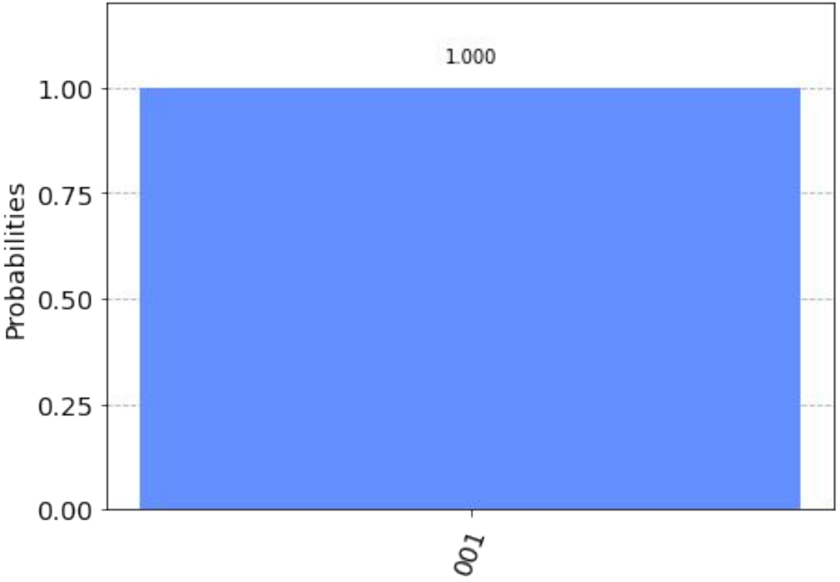
$$U |x\rangle = |(x - 1) \bmod N\rangle$$

Where U defines an operation that applies multiple 0 multi-controlled X gates and flipping the first qubit.

TABLE 2.6: Truth table for Quantum Decrement Operation

Initial State $\|x\rangle$	Final State $U\|x\rangle$
$\|000\rangle$	$\|111\rangle$
$\|001\rangle$	$\|000\rangle$
$\|010\rangle$	$\|001\rangle$
$\|011\rangle$	$\|010\rangle$
$\|100\rangle$	$\|011\rangle$
$\|101\rangle$	$\|100\rangle$
$\|110\rangle$	$\|101\rangle$
$\|111\rangle$	$\|110\rangle$

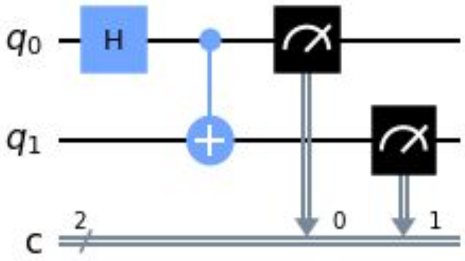


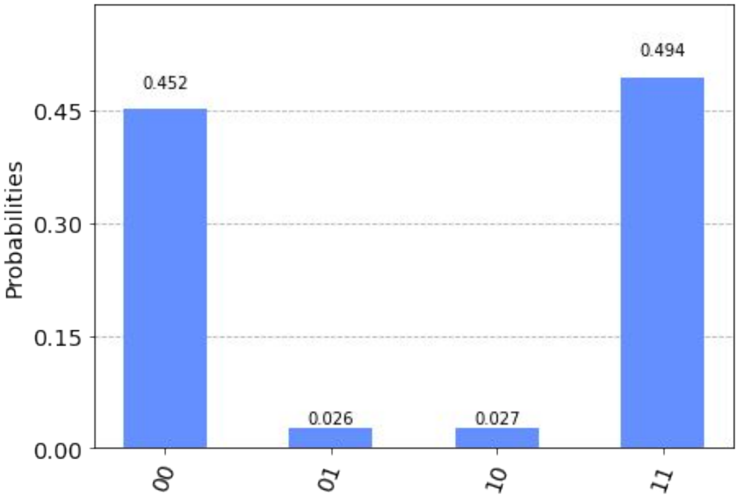


Build BELL State

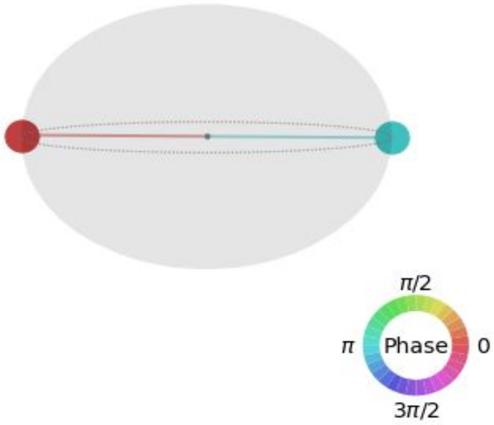
The Bell circuit is a quantum circuit that creates a Bell state. A Bell state is a maximally entangled state of two qubits. This means that the qubits are completely correlated, and measuring one qubit will instantly reveal the state of the other qubit, even if they are separated by a large distance.Receive a state on two qubits |xy⟩ and takes it to a given Bell State. This operation is done by applying the U gate U|x⟩ = |βxy⟩ as follows,The following equation describes the Bell circuit: |Φ+⟩ = H ⊗ CNOT |00⟩ Where: |Φ+⟩ is a Bell state, H is the Hadamard gate, CNOT is the CNOT gate, |00⟩ is the ground state of two qubits

$$|\beta_{00}^{+}\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$
$$|\beta_{01}^{-}\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle)$$
$$|\beta_{10}^{+}\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle)$$
$$|\beta_{11}^{-}\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle)$$

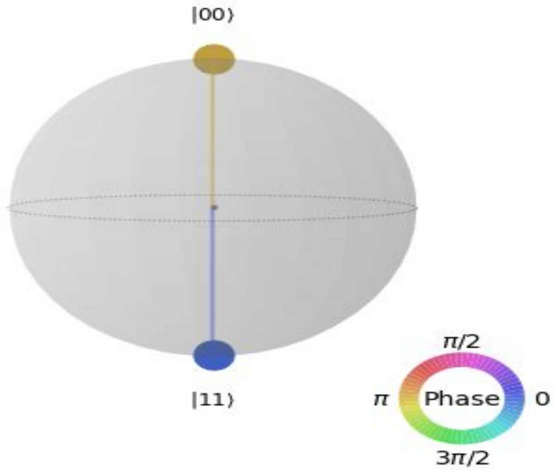




Build BELL State



Plot State qsphere for Bell State 11 qsphere



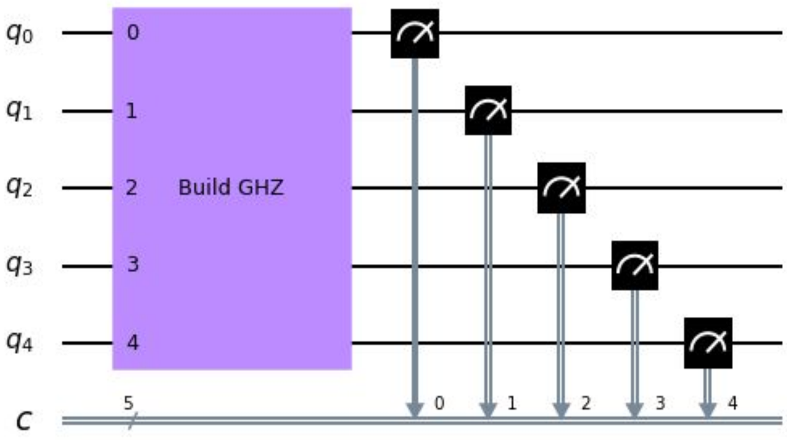
Plot State qsphere for Bell State 10 qsphere

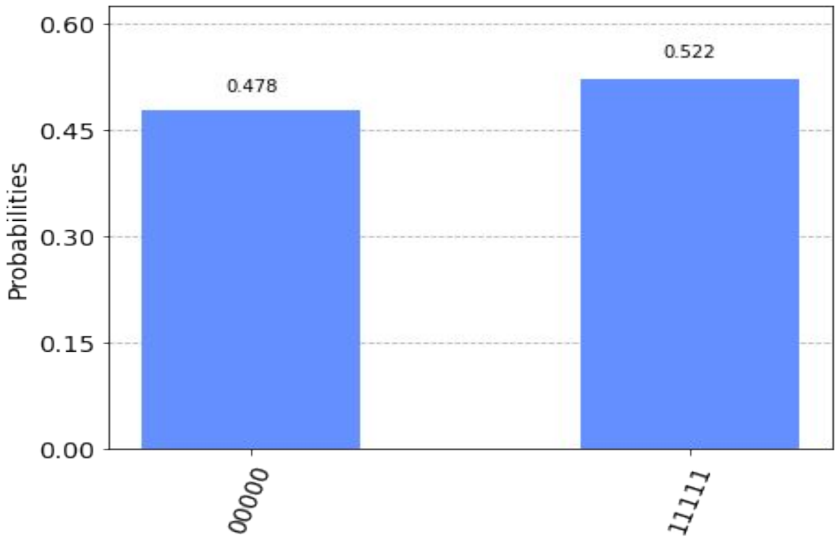
Build GHZ State

Takes an input state |x⟩ = |000…..0⟩ and takes it to the GHZ state $\frac{1}{\sqrt{2}}(|0\dots0\rangle + |1\dots1\rangle)$,which can be done by applying the following circuit |U⟩. Let |x⟩ be a all zeros state on n qubits,

$$U|x\rangle = \frac{1}{\sqrt{2}}(|000\dots0\rangle + |111\dots1\rangle) \quad |x\rangle$$

Where |U⟩ defines an operation that applies a Hadamard gate |H⟩ on the first qubit and Controlled |X⟩ gates, CNOT, on the remaining ones, taking the control qubit as the first one and the targets from the second to the n-th qubit.

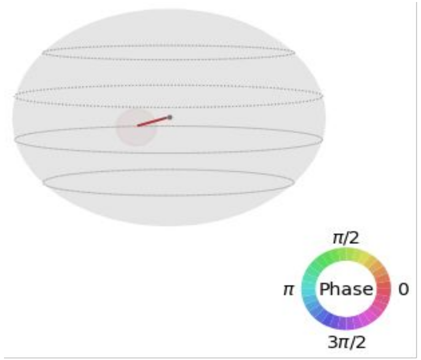




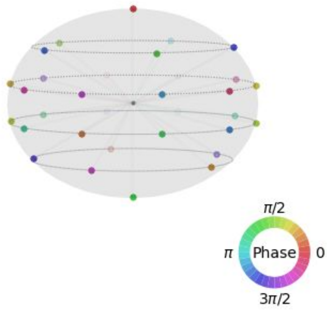
Quantum Fourier Transform

Takes an input state |x⟩ on the computational basis and takes it to the Fourier basis, which can be done by applying the following circuit |U⟩. Let |x⟩ be a all zeros state on |n⟩ qubits,

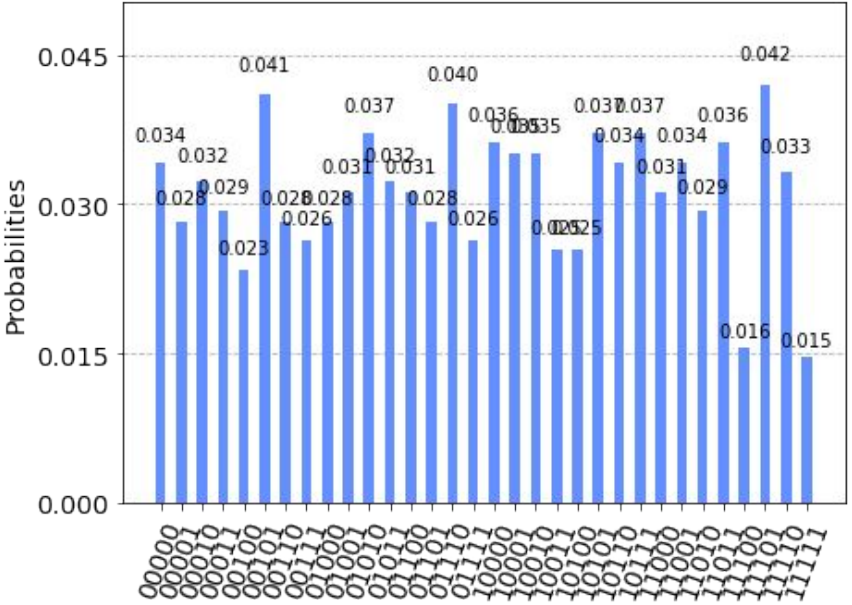
$$U|x\rangle = \frac{1}{\sqrt{N}} \sum_{y=0}^{N-1} \exp\left(\frac{2\pi i x y}{N}\right)|y\rangle$$



Plot State **qsphere** for **Computational** basis state qsphere



Plot State **qsphere** for **Fourier** basis state qsphere



Histogram Plot for **QFT**, Peaks at the **Coefficients** of Fourier Equation

Future Work

In my ongoing research for Bachelor's Thesis Part 2 (BTP2), I'm committed to further exploration in these areas:

Quantum Fourier Transform	To enhance precision, and further work to address the problem of Period Finding Problem in the next session of development.
Quantum Phase Estimation	I'll extend the work towards Phase Estimation, Precisely the problem of how can we extract the Θ parameter given the ability to prepare ψ and applying the unitary operator *U*.
Shor's Algorithm	In the Future we may focus working on certain implementations in Shor's algorithm: Enhancing the algorithm's efficiency for factoring large numbers, Exploring its applications in other areas beyond cryptography.
Grover's Algorithm	Grover's Algorithm is the solve the problem of "Searching a element into an unsorted database", which with a high probability finds the searched given element, using just $O(\sqrt{N})$ queries as in comparison to classic $O(N)$ algorithm.
Deutsch-Jozsa Algorithm	The Deutsch-Jozsa algorithm is a simple but important quantum algorithm that is exponentially faster than any classical algorithm for solving a certain type of problem. We hope to implement this quantum algorithm and work on it's open endpoints(available for research).

References

1. R. H. (Bo) Ewald. An introduction to quantum computing and its application. In (Vol. 11413 LNCS, pp. 3–8). Springer Verlag. https://doi.org/10.1007/978-3- 030-14082-31, 2019.
2. M. Coggins. Introduction to Quantum Computing with Qiskit. Scarborough Quantum Computing Ltd, December 2021.
3. R. P. da Silva. Quantum Circuits with Qiskit. PhD thesis, SSRN Electronic Journal, February 15, 2023.
4. J. A. F. Fathelrahman Mohamed, Ibrahim Savran. Quantum fourier transform :using five qubits on qiskit.
5. . C. I. L. Nielsen, M.A. Quantum computation and quantum information (10th anniv). In Cambridge University Press.,

   2010.
6. R. C. S. Pimenta and A. Bezerra. Leveraging qiskit to teach basic quantum computation algorithms. Parana Journal of Science and Education, July 2021.
7. Q. D. Team. Qiskit. In https://qiskit.org, 2023.
8. Q. Tutorials. https://nbviewer.jupyter.org/github/qiskit/qiskit-tutorial/blob/master/index.ipynb.
9. M. M. A. Yanofsky, N. S. Quantum computing for computer scientists. In New York, NY: Cambridge University Press, 2008.

Thank you!