Quantum Project

# Team Composition

Qasim Naveed 21L-5231

Laraib Akhtar 21L-5294

Aatiqa Hussain 21L-5397

1. **Introduction**

This document provides a detailed explanation of the provided code, which focuses on simulating quantum circuits using Qiskit, a comprehensive quantum computing framework developed by IBM. The code demonstrates how to create quantum circuits, introduce noise models, run simulations, and visualize results.

1. **Code Overview**

The provided code can be divided into several sections:

1. **Importing Libraries and Setting Up IBM Quantum Runtime Service:** The code begins by importing necessary libraries such as Qiskit and defining IBM Quantum Runtime Service credentials. This service allows users to access IBM Quantum devices and simulators for quantum computations.

2. **Defining Noise Model:** The code defines a noise model to simulate errors that occur during quantum computation. It includes functions to introduce quantum errors like bit-flip errors.

3. **Creating a Quantum Circuit:** Next, a quantum circuit is created using Qiskit. This circuit includes Hadamard gates, CNOT gates, and CZ gates, along with measurements.

4. **Running Ideal Simulation:** The quantum circuit is simulated without noise to obtain an ideal result. This is done using Qiskit's AerSimulator.

5. **Introducing Noise:** The noise model defined earlier is applied to the quantum circuit to simulate realistic quantum environments where errors occur during computation.

6. **Running Noisy Simulation:** The noisy quantum circuit is simulated using Qiskit's AerSimulator with the defined noise model.

7. **Visualizing Results:** The results of both ideal and noisy simulations are plotted using histograms to visualize the outcomes of the quantum circuits.

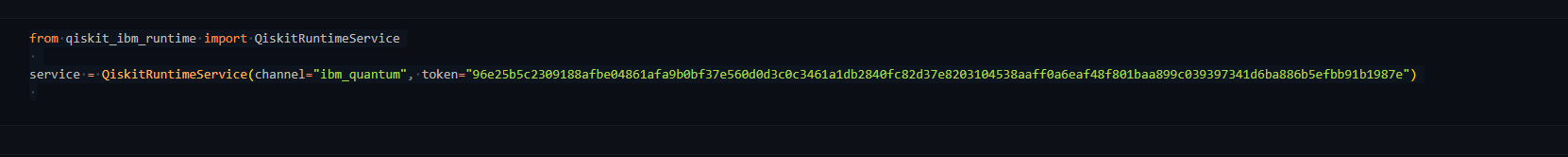
8. **Running Quantum Circuit on IBM Quantum Simulator:** Finally, the quantum circuit is executed on an IBM Quantum Simulator via the IBM Quantum Runtime Service. The results are retrieved and plotted.

1. **Detailed Explanation**

Let's break down each part of the code and provide a detailed explanation:

**Importing Libraries and Setting Up IBM Quantum Runtime Service**

The code starts by importing necessary libraries including Qiskit and defining IBM Quantum Runtime Service credentials. This service enables access to IBM Quantum devices and simulators.

****

1. **Defining Noise Model**

The code defines a noise model to simulate errors occurring during quantum computation. This includes functions to introduce quantum errors like bit-flip errors.

**A computer screen shot of a program code

Description automatically generated**

1. **Creating a Quantum Circuit**

A quantum circuit is created using Qiskit. This circuit includes Hadamard gates, CNOT gates, CZ gates, and measurements.

**A diagram of a circuit

Description automatically generated**

1. **Running On different Backends**

**Aer Simulator:** The ideal quantum circuit is simulated without noise to obtain the expected result using Qiskit's AerSimulator.

**Introducing Noise and Running Noisy Simulation:** The defined noise model is applied to the quantum circuit to simulate realistic quantum environments where errors occur during computation. The noisy quantum circuit is then simulated using Qiskit's AerSimulator with the defined noise model.

**Running Quantum Circuit on IBM Quantum Simulator:** Finally, the quantum circuit is executed on an IBM Quantum Simulator via the IBM Quantum Runtime Service. The results are retrieved and plotted.

**Additional Statevector and Density Matrix Calculations:** Additional calculations involving statevectors and density matrices are performed to visualize the quantum state of the circuit.

**Conclusion:** This document provides a detailed overview of the provided code for simulating quantum circuits using Qiskit. It covers the creation of quantum circuits, introduction of noise models, running simulations, and visualizing results. By following this explanation, users can gain a better understanding of quantum circuit simulations and utilize Qiskit for their quantum computing experiments. To see the results refer to the quantum.ipynb jupter notebook. A pip list is shared at the end if you need are missing any dependencies.

**Once again thank you sir for everything. Learning Quantum Computing from you was super fun and super COOL.**