**Lab Manual: Simulating Basic Quantum Gates in Qiskit**

**Step 1: Theoretical Background**

Quantum gates are the building blocks of quantum circuits, analogous to classical logic gates. Here are some basic quantum gates:

1. **Hadamard Gate (H):**
   * Creates superposition.
   * Transforms |0⟩ to (|0⟩ + |1⟩)/√2 and |1⟩ to (|0⟩ - |1⟩)/√2.
2. **Pauli-X Gate (X):**
   * Flips the state of the qubit.
   * Transforms |0⟩ to |1⟩ and |1⟩ to |0⟩.
3. **Pauli-Y Gate (Y):**
   * Applies a combination of a bit flip and a phase flip.
   * Transforms |0⟩ to i|1⟩ and |1⟩ to -i|0⟩.
4. **Pauli-Z Gate (Z):**
   * Applies a phase flip.
   * Transforms |0⟩ to |0⟩ and |1⟩ to -|1⟩.

**Step 2: Implementation**

1. **Set Up Your Environment**
   * Ensure you have Python, Qiskit, and Matplotlib installed:

bash

Copy

pip install qiskit matplotlib

1. **Write the Code to Simulate Basic Gates**
   * The following code demonstrates the use of the Hadamard, Pauli-X, and Pauli-Z gates:

from qiskit import QuantumCircuit

from qiskit\_aer import AerSimulator

import matplotlib.pyplot as plt

# Function to visualize the circuit and state

def visualize\_circuit\_and\_state(qc, initial\_state, title):

    print(title)

    print(qc.draw())

    qc.measure\_all()  # Measure all qubits

    simulator = AerSimulator()

    result = simulator.run(qc).result()

    counts = result.get\_counts()

    plt.bar(counts.keys(), counts.values(), color='skyblue')

    plt.title(title)

    plt.xlabel('States')

    plt.ylabel('Counts')

    plt.show()

# Create a Quantum Circuit with 1 qubit for Hadamard gate

qc\_h = QuantumCircuit(1)

qc\_h.h(0)  # Apply Hadamard gate

# Visualize Hadamard Gate

visualize\_circuit\_and\_state(qc\_h, '|0⟩', 'Hadamard Gate')

# Create a Quantum Circuit with 1 qubit for Pauli-X gate

qc\_x = QuantumCircuit(1)

qc\_x.x(0)  # Apply Pauli-X gate

# Visualize Pauli-X Gate

visualize\_circuit\_and\_state(qc\_x, '|0⟩', 'Pauli-X Gate')

# Create a Quantum Circuit with 1 qubit for Pauli-Z gate

qc\_z = QuantumCircuit(1)

qc\_z.z(0)  # Apply Pauli-Z gate

# Visualize Pauli-Z Gate

visualize\_circuit\_and\_state(qc\_z, '|0⟩', 'Pauli-Z Gate')

1. **Run the Code**
   * Execute this script in your Python environment.

**Step 3: Test Results**

* **View the Quantum Circuit Diagram**  
  Each gate's corresponding quantum circuit will be printed to the console.
* **Measurement Results**  
  After executing the circuit for each gate, you will see a bar plot displaying the measurement results.

**Step 4: Analyze the Results**

1. **Hadamard Gate:**
   * Expected Result: Approximately equal counts for |0⟩ and |1⟩ due to the superposition created.
2. **Pauli-X Gate:**
   * Expected Result: All counts for |1⟩, as this gate flips |0⟩ to |1⟩.
3. **Pauli-Z Gate:**
   * Expected Result: All counts for |0⟩, since Z does not change |0⟩.

**Conclusion**

This lab manual demonstrates how to simulate basic quantum gates using Qiskit. By applying the Hadamard, Pauli-X, and Pauli-Z gates, you can explore the transformations of quantum states. Understanding these gates is fundamental for building more complex quantum algorithms.

**Further Exploration**

1. **Combine Gates:** Create circuits with combinations of these gates and analyze the results.
2. **Explore Other Gates:** Implement additional gates like CNOT, Toffoli, etc.
3. **Increase Qubits:** Experiment with multi-qubit circuits to understand more complex behaviors.