CSBB 311 : QUANTUM COMPUTING

LAB ASSIGNMENT 7 : Quantum Error Correction

Submitted By:

Name: KARTIK MITTAL

Roll No: 221210056

Branch: CSE

Semester: 5th Sem

Group: 2

Submitted To: Dr. VS Pandey

Department of Computer Science and Engineering

NATIONAL INSTITUTE OF TECHNOLOGY DELHI

2024

Theory -

1. Introduction to Quantum Error Correction

- Quantum error correction is a critical field in quantum computing aimed at protecting quantum information from errors caused by noise, decoherence, and imperfections in quantum hardware.

2. Quantum Error Correction Codes

- Quantum error correction codes, such as the Shor code, Steane code, and surface codes, are designed to detect and correct errors in quantum systems. These codes work by redundantly encoding quantum information in multiple qubits, allowing errors to be identified and corrected through syndrome measurements..

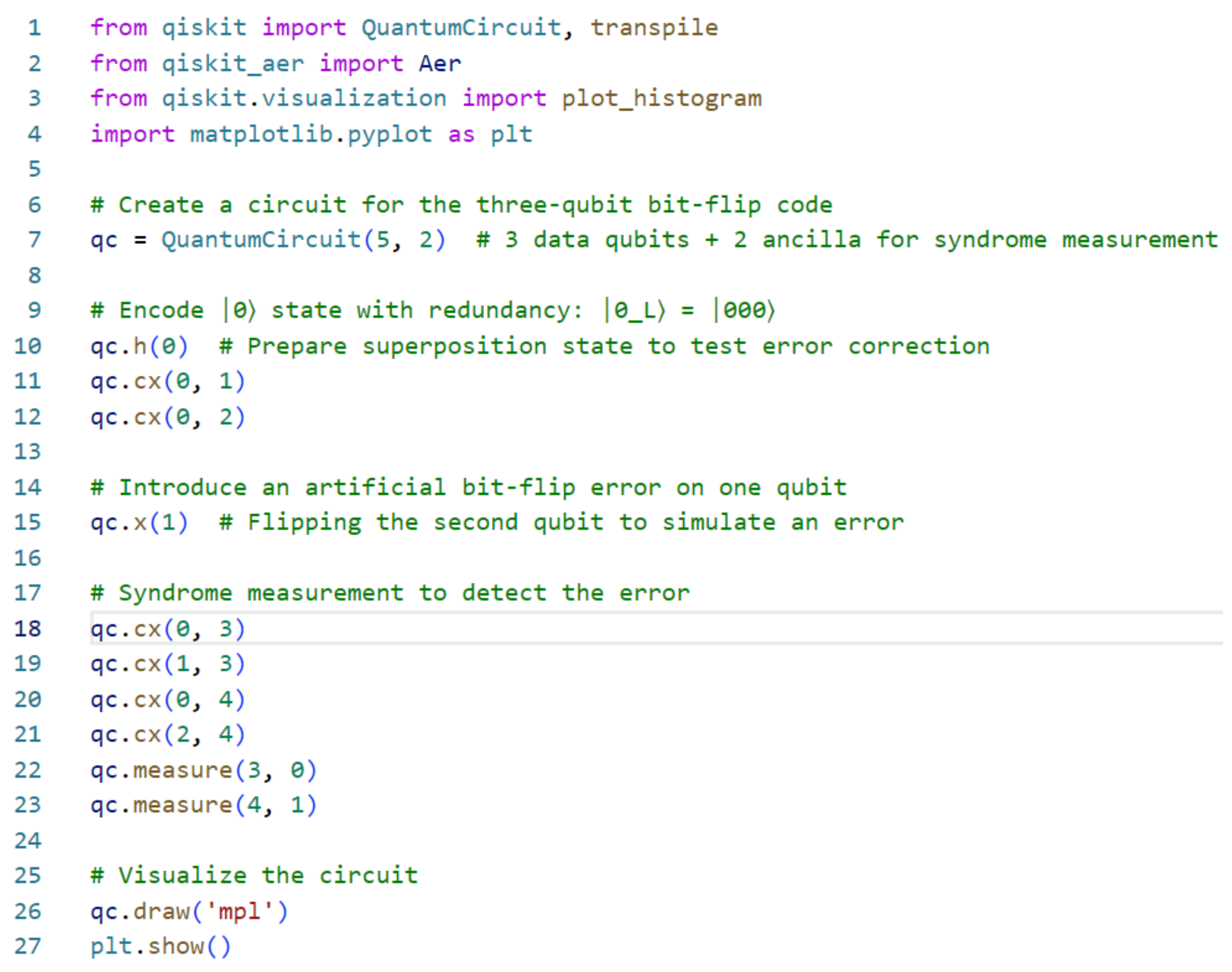
3. Workflow of Quantum Error Correction

- **Quantum Circuit Initialization**The quantum system is initialized into a state that encodes logical information into multiple physical qubits using a quantum error correction code.
- **Syndrome Measurement**:A series of measurements are made to detect errors. These measurements do not collapse the quantum state but instead provide information about potential errors affecting the qubits.
- **Error Detection and Correction**: After syndrome measurements, a classical post-processing step is performed to determine the errors and apply corrective operations.
- **Fault-Tolerant Computation**: Quantum error correction ensures that even if errors occur during the error-correction process, the system can still continue functioning correctly.

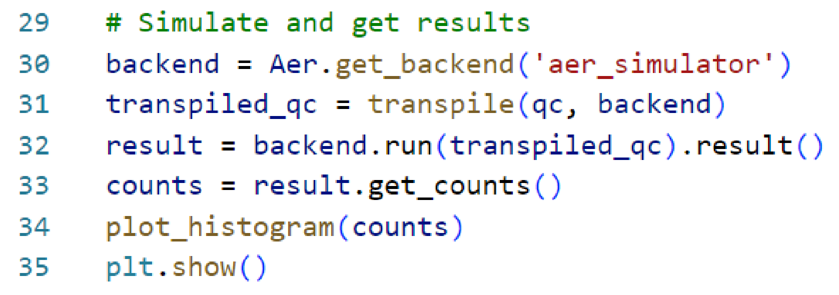
4. Importance of Quantum Error Correction

- Quantum error correction is essential for achieving practical quantum computing. It allows quantum algorithms to be executed on noisy intermediate-scale quantum (NISQ) devices by ensuring that errors do not propagate uncontrollably.
- The development of efficient quantum error correction methods is vital for scaling quantum computing to solve problems that are currently intractable for classical computers.

Code (Quantum Error Correction) -

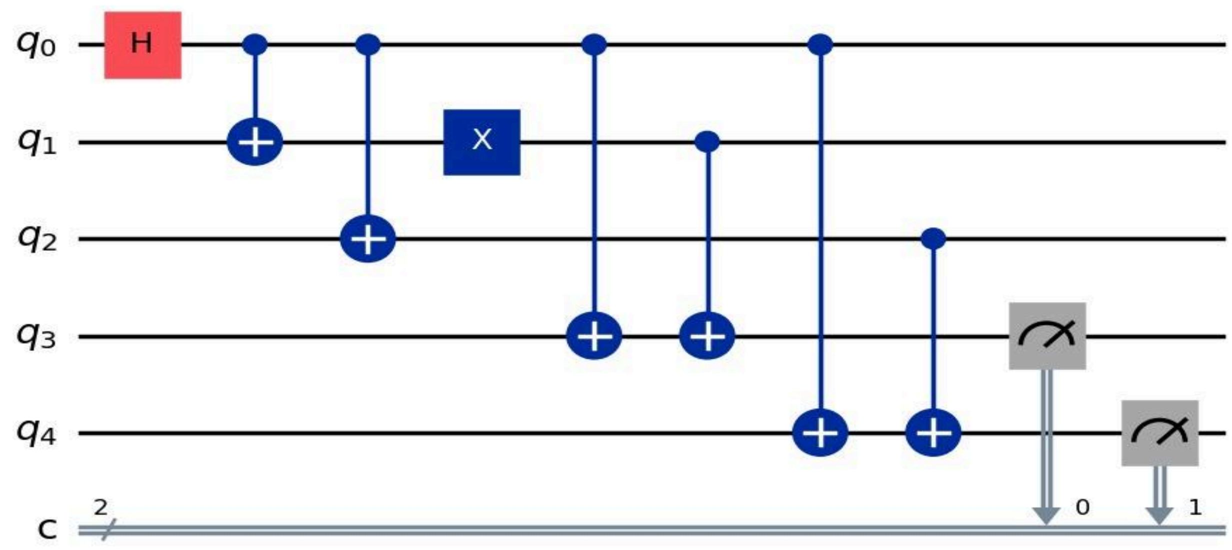


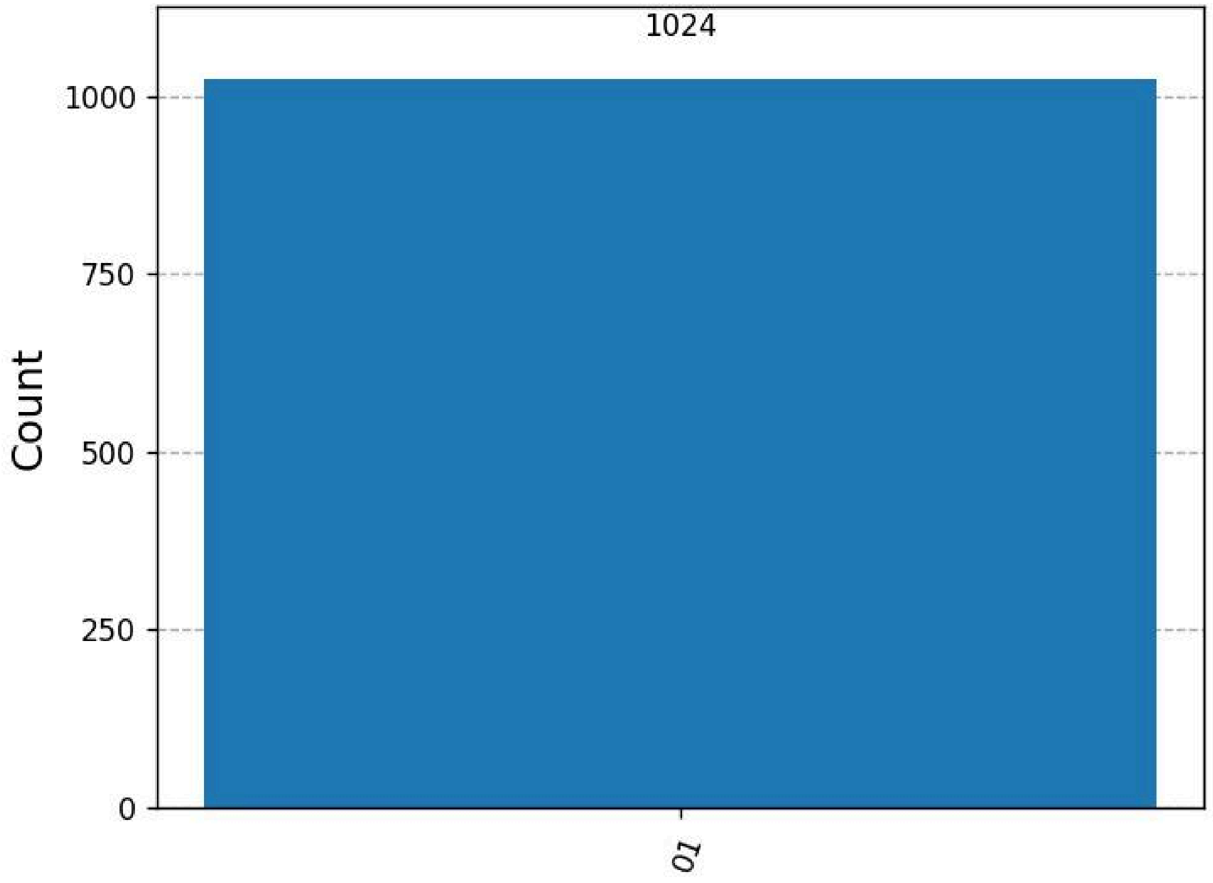
```python
from qiskit import QuantumCircuit, transpile
from qiskit_aer import Aer
from qiskit.visualization import plot_histogram
import matplotlib.pyplot as plt

# Create a circuit for the three-qubit bit-flip code
qc = QuantumCircuit(5, 2)  # 3 data qubits + 2 ancilla for syndrome measurement

# Encode |0⟩ state with redundancy: |0_L⟩ = |000⟩
qc.h(0)  # Prepare superposition state to test error correction
qc.cx(0, 1)
qc.cx(0, 2)

# Introduce an artificial bit-flip error on one qubit
qc.x(1)  # Flipping the second qubit to simulate an error

# Syndrome measurement to detect the error
qc.cx(0, 3)
qc.cx(1, 3)
qc.cx(0, 4)
qc.cx(2, 4)
qc.measure(3, 0)
qc.measure(4, 1)

# Visualize the circuit
qc.draw('mpl')
plt.show()
```



```python
# Simulate and get results
backend = Aer.get_backend('aer_simulator')
transpiled_qc = transpile(qc, backend)
result = backend.run(transpiled_qc).result()
counts = result.get_counts()
plot_histogram(counts)
plt.show()
```

Output -

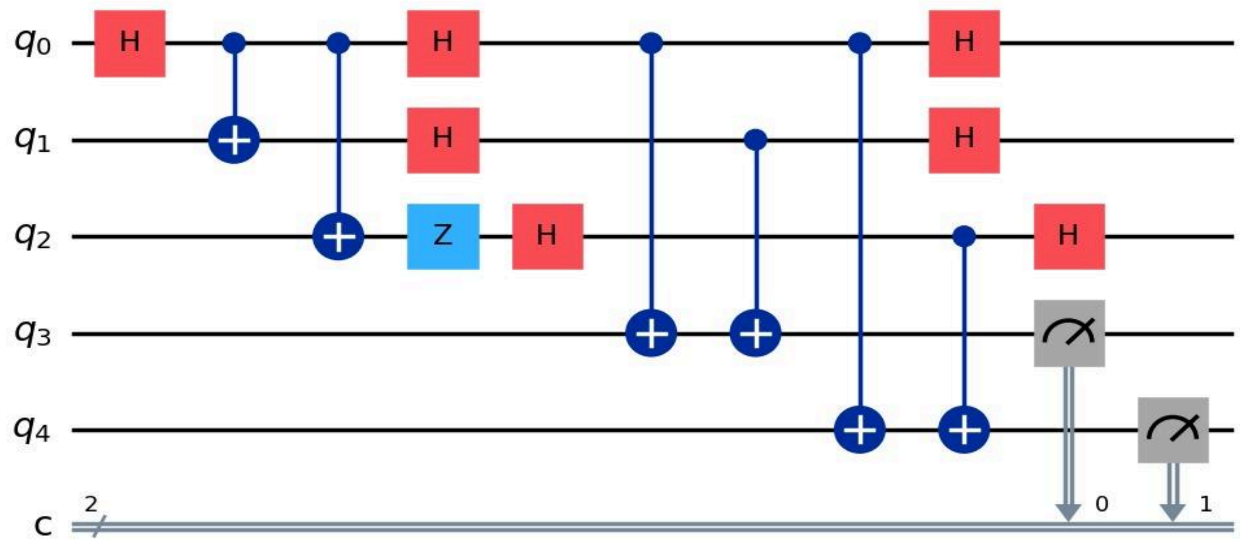




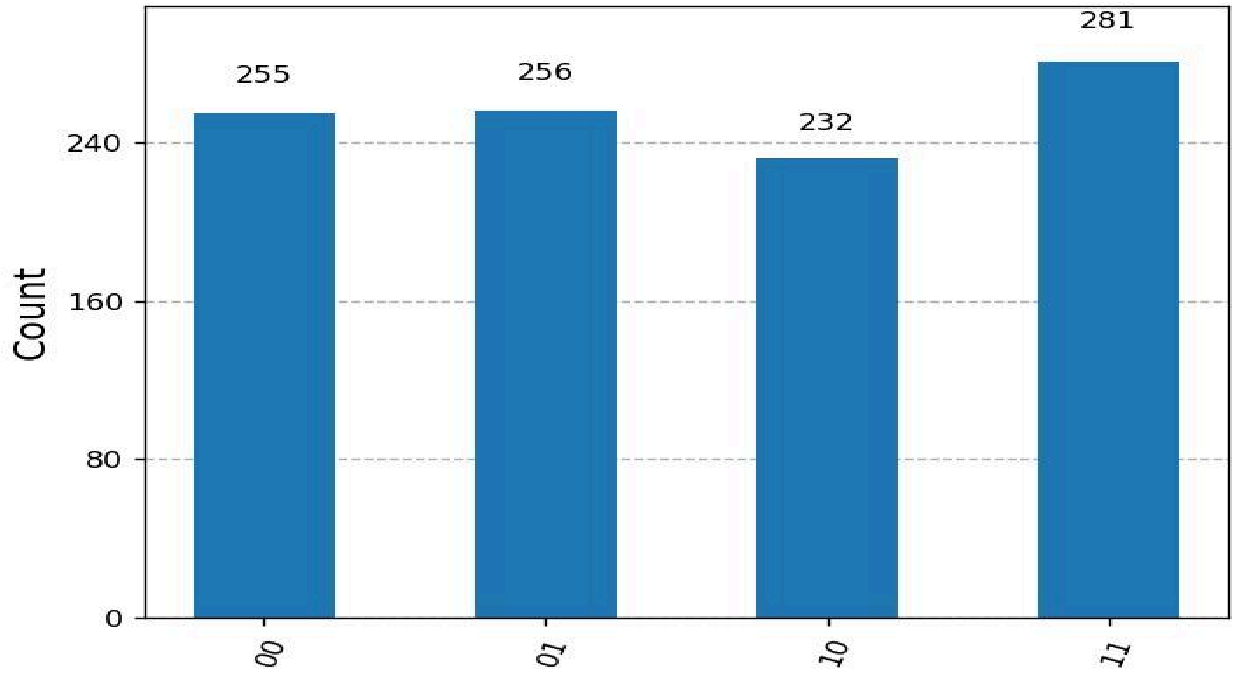
Code -

```
from qiskit import QuantumCircuit, transpile
from qiskit_aer import Aer
from qiskit.visualization import plot_histogram
import matplotlib.pyplot as plt

# Create a phase-flip code circuit (five qubits: 3 data qubits + 2 ancilla qubits)
qc = QuantumCircuit(5, 2)  # 3 data qubits + 2 ancilla qubits

# Encode the logical |0⟩ using a phase code: |0_L⟩ = (|000⟩ + |111⟩) / √2
qc.h(0)  # Apply Hadamard gate on qubit 0
qc.cx(0, 1)  # Apply CNOT gate between qubit 0 and qubit 1
qc.cx(0, 2)  # Apply CNOT gate between qubit 0 and qubit 2

# Introduce a phase-flip error on the third qubit (qubit 2)
qc.z(2)

# Syndrome measurement for phase-flip error detection
qc.h(0)
qc.h(1)
qc.h(2)
qc.cx(0, 3)
qc.cx(1, 3)
qc.cx(0, 4)
qc.cx(2, 4)
qc.h(0)
qc.h(1)
qc.h(2)

# Correct measurement of ancilla qubits
qc.measure(3, 0)  # Measure the ancilla qubit 3 to classical bit 0
qc.measure(4, 1)  # Measure the ancilla qubit 4 to classical bit 1

# Draw and display the circuit
qc.draw('mpl')
plt.show()

# Run simulation on the Aer simulator
backend = Aer.get_backend('aer_simulator')
transpiled_qc = transpile(qc, backend)
result = backend.run(transpiled_qc).result()
counts = result.get_counts()

# Plot the histogram of the results
plot_histogram(counts)
plt.show()
```

Output -



q_0
q_1
q_2
q_3
q_4
c
2
H
H
H
H
H
H
Z
H
0
1



Count
255
256
232
281
00
01
10
11
0
80
160
240

Conclusion -

- **Efficiency-Scalability Trade-off**: Quantum error correction presents a trade-off between efficiency and scalability due to the significant overhead required for encoding and correcting errors. As the complexity of the quantum system increases, the number of qubits needed for error correction also grows, impacting the overall quantum resources required
- **Algorithmic Significance**:Quantum error correction is vital for ensuring the reliability of quantum computations, making it a foundational component of scalable quantum computing. By protecting quantum information from errors and decoherence, it enables the execution of long and complex quantum algorithms with high fidelity.