Quantum Computing Simulation Report

1. Simple Quantum Circuit using IBM Quantum Experience (Qiskit)

In this task, I used Qiskit, IBM's open-source quantum computing SDK, to simulate a simple 2-qubit quantum circuit that demonstrates quantum entanglement via a Bell State. This was done using the IBM Quantum Experience environment.

Circuit Overview:

- Hadamard Gate on qubit 0: Places the qubit in superposition.
- CNOT Gate between qubit 0 (control) and qubit 1 (target): Creates entanglement.
- Measurement of both qubits to observe the quantum state.

The simulation returned 50% probability for outcomes '00' and '11', which confirms successful entanglement a key feature of quantum advantage.

2. Optimization of AI Task: Drug Discovery

Why Quantum?

Traditional AI models used in drug discovery struggle with simulating the quantum properties of molecules due to complexity and computational limits. Quantum computing, particularly through entanglement and superposition, can simulate these quantum interactions *natively*.

Benefits of AI in Drug Discovery:

- Parallel Molecular Simulation: Quantum states represent many possibilities at once, speeding up simulations.
- Quantum-Assisted AI: Machine learning models can be enhanced with quantum subroutines for faster feature extraction and prediction.
- Compound Optimization: Quantum algorithms can explore a larger chemical space more efficiently than their classical counterparts.

By using quantum circuits in conjunction with AI, researchers can:

- Identify viable drug candidates more quickly.
- Simulate molecular binding more accurately.
- Reduce the time and cost of bringing new drugs to market.

---

3. Quantum Circuit Code (Bell State Simulation)

```
from qiskit import QuantumRegister, ClassicalRegister, QuantumCircuit, transpile
from qiskit.visualization import plot_histogram
from qiskit_aer import AerSimulator
import matplotlib.pyplot as plt

# Create quantum and classical registers
qreg_q = QuantumRegister(2, 'q')   # Only 2 qubits needed
creg_c = ClassicalRegister(2, 'c')
circuit = QuantumCircuit(qreg_q, creg_c)

# Apply Hadamard gate and CNOT gate to create Bell State
circuit.h(qreg_q[0])
circuit.cx(qreg_q[0], qreg_q[1])
circuit.barrier()

# Measure both qubits
circuit.measure(qreg_q[0], creg_c[0])
circuit.measure(qreg_q[1], creg_c[1])

# Display the circuit
print("Quantum Circuit:")
print(circuit)

# Simulate using AerSimulator
simulator = AerSimulator()
compiled_circuit = transpile(circuit, simulator)
result = simulator.run(compiled_circuit, shots=1024).result()
counts = result.get_counts(circuit)

# Display results
print("\nSimulation Results:")
print(counts)
plot_histogram(counts)
plt.title("Bell State Measurement Resu")
plt.show()
```

Conclusion

Using IBM Quantum Experience (via Qiskit), I successfully simulated a Bell State circuit to demonstrate entanglement a fundamental building block of quantum computation. This principle, when combined with AI, has powerful implications for drug discovery, leading to faster and more accurate exploration of molecular interactions.