Mr. Phiphat Chomchit 630631028

Test 7: Quantum Teleportation (10/17/2021)

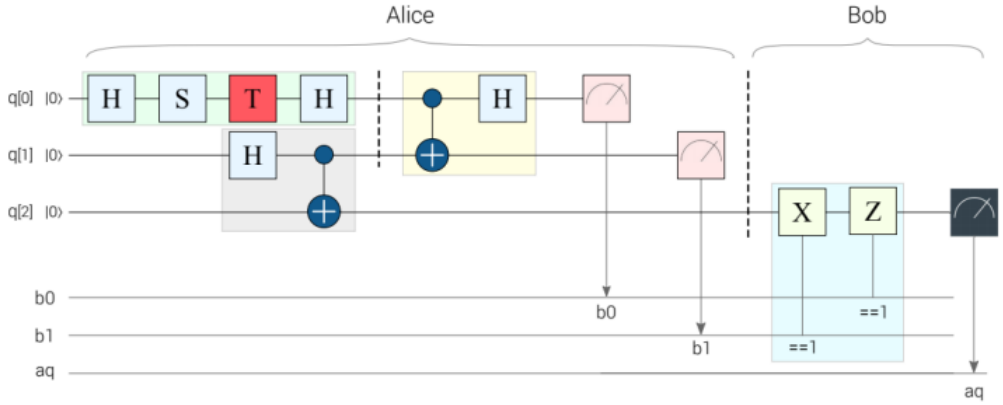
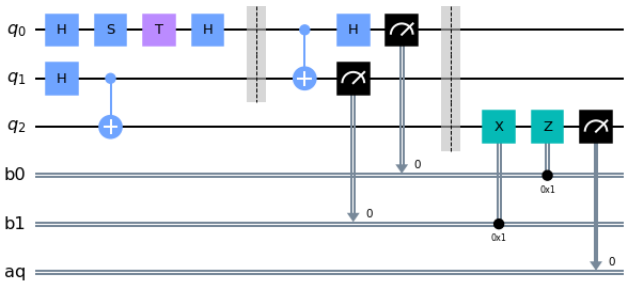


Figure 6-25. Quantum teleportation circuit: The state of qubit q[0] is teleported to qubit q[2].

1. Quantum Teleportation Circuit.

```
    from qiskit import QuantumRegister, ClassicalRegister
, QuantumCircuit
    qreg_q = QuantumRegister(3, 'q')
    creg_b0 = ClassicalRegister(1, 'b0')
    creg_b1 = ClassicalRegister(1, 'b1')
    creg_aq = ClassicalRegister(1, 'aq')
    circuit = QuantumCircuit(qreg_q, creg_b0, creg_b1,
creg_aq)

    circuit.h(qreg_q[0])
    circuit.s(qreg_q[0])
    circuit.t(qreg_q[0])
    circuit.h(qreg_q[0])
    circuit.h(qreg_q[1])
    circuit.cx(qreg_q[1], qreg_q[2])

    circuit.barrier(qreg_q[0], qreg_q[1])
    circuit.cx(qreg_q[0], qreg_q[1])
    circuit.h(qreg_q[0])
    circuit.measure(qreg_q[0], creg_b0[0])
    circuit.measure(qreg_q[1], creg_b1[0])
    circuit.barrier(qreg_q[0], qreg_q[1], qreg_q[2])

    circuit.x(qreg_q[2]).c_if(creg_b1, 1)
    circuit.z(qreg_q[2]).c_if(creg_b0, 1)
    circuit.measure(qreg_q[2], creg_aq[0])

    #drawing the circuit
    circuit.draw('mpl')


```

Listing 1: Circuit



2. Simulation and Visualization of the Result.

```
    # Adding the transpiler to reduce the circuit to QASM instructions
    # supported by the backend
    from qiskit import transpile

    # Use Aer's qasm_simulator
    from qiskit.providers.aer import QasmSimulator

    backend = QasmSimulator()

    # First we have to transpile the quantum circuit
    # to the low-level QASM instructions used by the
    # backend
    qc_compiled = transpile(circuit, backend)

    # Execute the circuit on the qasm simulator.
    # We've set the number of repeats of the circuit
    # to be 1024, which is the default.
    job_sim = backend.run(qc_compiled, shots=1024)

    # Grab the results from the job.
    result_sim = job_sim.result()

    from qiskit.visualization import plot_histogram
    plot_histogram(counts)

```

Listing 2: Simulation and Visualization