Quantum Measurement and Visualization Project

Author: *Tymon Dydowicz*
Environment: Qiskit, Python 3.10.11
Student ID: 151936

Project Overview

This project demonstrates **quantum state preparation, measurement, and visualization** for a 4-qubit system using Qiskit.

Each task focuses on measuring quantum states under various operations and bases, aggregating results over multiple runs, and generating visualizations of the final quantum states.

Tasks

1. **Z-type projection measurement** — reading qubit states in the computational basis
2. **Quantum gate X operation** — applying and measuring the X (NOT) gate result
3. **Quantum gate H operation** — applying and measuring the Hadamard gate result
4. **Single-qubit state tomography** — performing X, Y, and Z basis measurements
5. **State visualizations** — generating Bloch, Hinton, City, and Qsphere plots

All simulations use:

- **4 qubits**
- **Target qubit:** q_0
- **2048 shots per run**
- **3 executions per task**, aggregated into combined histograms
- **Random seed:** 151936

Methodology

Key Code Snippets

Below are key Python functions used to construct and execute the quantum simulations:

```
SHOTS = 2048
SEED = 151936
OUTPUT_DIR = "results/"
N_QUBITS = 4
TARGET_QUBIT = 0
EXECUTIONS = 3
```

```python
def run_measurement(circ, simulator, shots=SHOTS, seed=SEED):
    transpiled = transpile(circ, simulator)
    result = simulator.run(transpiled, shots=shots, seed_simulator=seed).result()
    counts = result.get_counts()
    total = sum(counts.values()) if counts else shots
    probs = {k: v / total for k, v in counts.items()}
    return counts, probs
```

```python
def prepare_and_measure(prep_circ, measure_rotation=None, measure_qubits=None):
    if measure_qubits is None:
        measure_qubits = list(range(N_QUBITS))
    circ = QuantumCircuit(N_QUBITS, len(measure_qubits))
    circ.compose(prep_circ, inplace=True)
    if measure_rotation is not None:
        circ.compose(measure_rotation, inplace=True)
    for i, q in enumerate(measure_qubits):
        circ.measure(q, i)
    return circ
```

```python
def rotation_for_basis(basis, target_qubit=TARGET_QUBIT):
    r = QuantumCircuit(N_QUBITS)
    if basis == 'X':
        r.ry(np.pi/2, target_qubit)
        r.p(np.pi/2, target_qubit)
        r.h(target_qubit)
        return r
    if basis == 'Y':
        r.ry(np.pi/2, target_qubit)
        r.p(np.pi/2, target_qubit)
        r.sdg(target_qubit)
        r.h(target_qubit)
        return r
    if basis == 'Z':
        r.ry(np.pi/2, target_qubit)
        r.p(np.pi/2, target_qubit)
        return r
```

Measurement and Aggregation

- Circuits are prepared using standard Qiskit `QuantumCircuit` operations.
- Each configuration (Z, X, H, and tomography bases) is measured three times with distinct seeds.
- Results are stored and aggregated into histograms representing full 4-qubit states.
- Each bar in the histogram is labeled with **exact counts** and **probabilities**.

Tomography

Performed over three measurement bases:

- **X-basis**
- **Y-basis**
- **Z-basis**

The tomography reconstructs the reduced state of the target qubit by measuring the system in each basis, effectively allowing partial observation of the quantum density matrix.

Visualization

Four state visualizations are generated for each prepared state (`|0000⟩`, `X|0000⟩`, `H|0000⟩`):

1. **State City plot**
2. **State Hinton plot**
3. **State Bloch multivector**
4. **State Qsphere**

Results

1. Z-type Projection Measurement

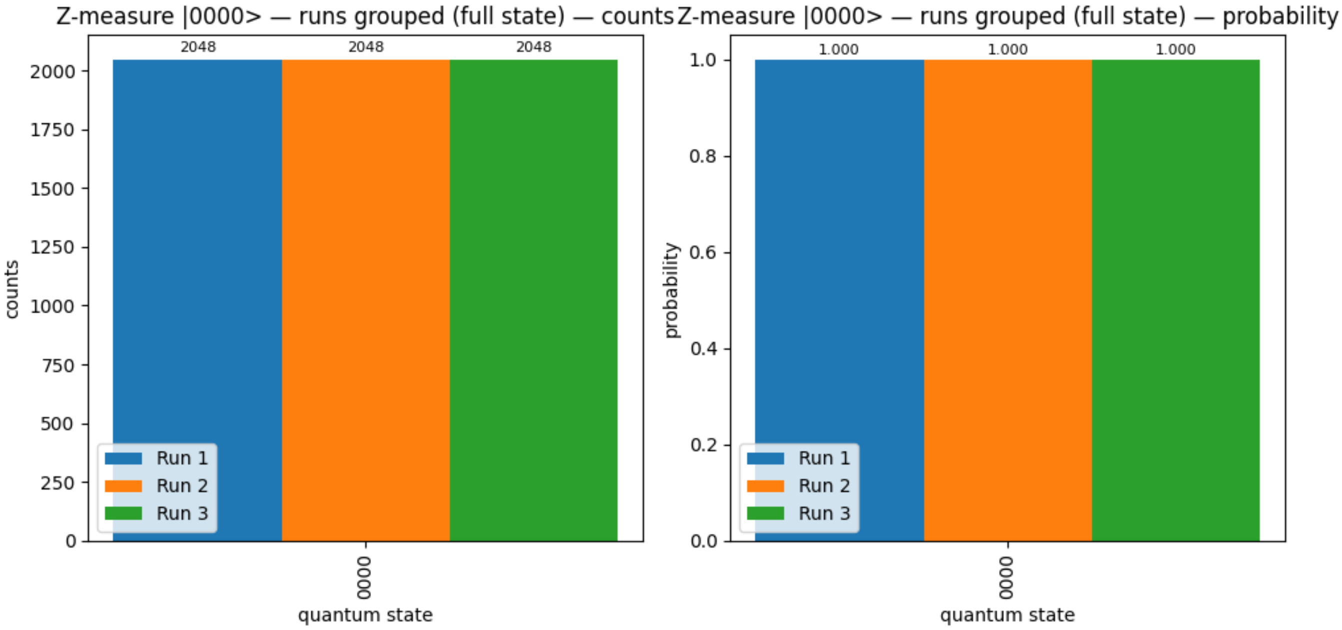
Measurement of the initial |0000⟩ state in the computational (Z) basis.

Code:

```
prep_zero = QuantumCircuit(N_QUBITS)

z_circ = prepare_and_measure(prep_zero, measure_rotation=None,
measure_qubits=None)
z_counts, z_probs = run_measurement(z_circ, simulator, shots=SHOTS, seed=run_seed)
```

Image:



---

2. Operation of Quantum Gate X

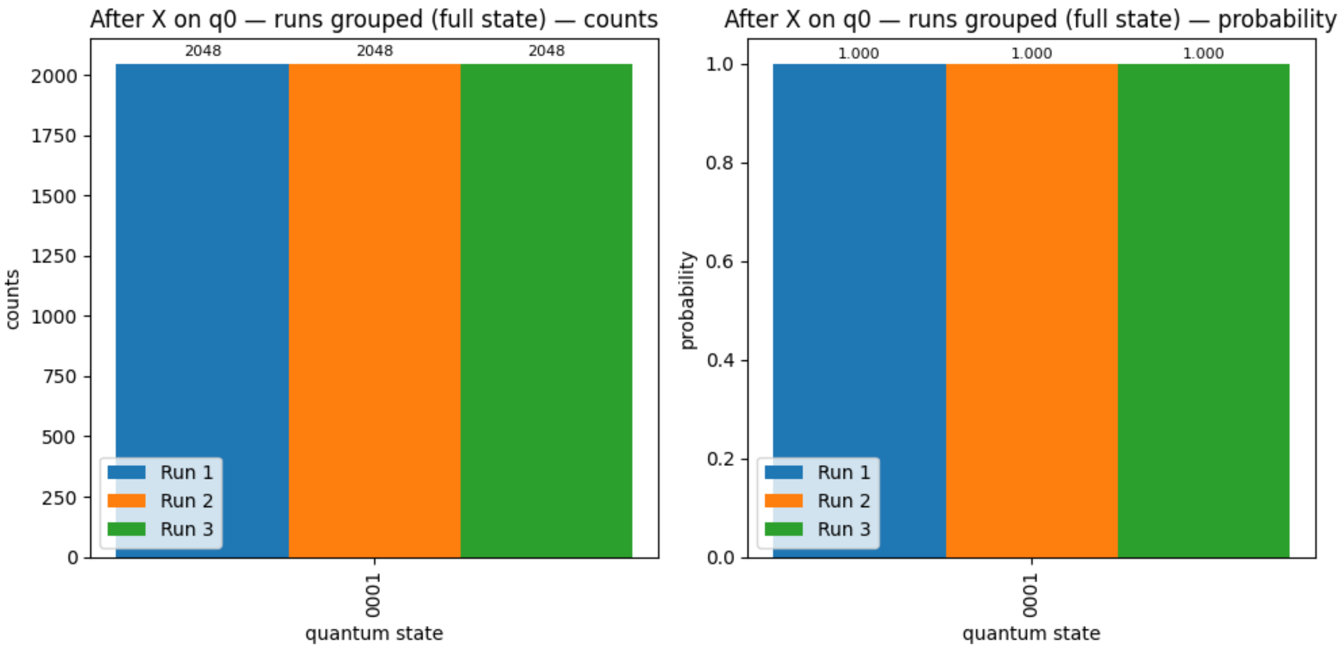
Application of an X gate on qubit 0, flipping |0000⟩ → |1000⟩.

Code:

```
prep_x = QuantumCircuit(N_QUBITS)
prep_x.x(TARGET_QUBIT)

x_circ = prepare_and_measure(prep_x, measure_rotation=None, measure_qubits=None)
x_counts, x_probs = run_measurement(x_circ, simulator, shots=SHOTS, seed=run_seed)
```

Image:



---

3. Operation of Quantum Gate H

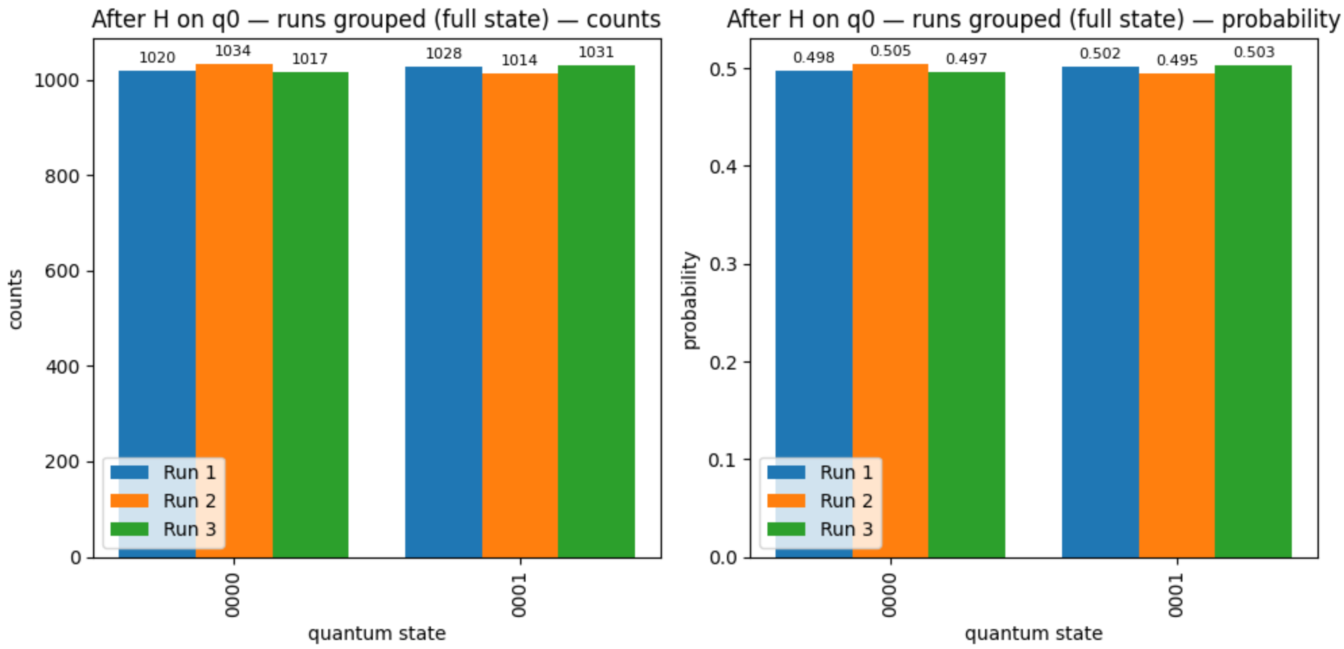
Application of a Hadamard gate on qubit 0, creating a superposition

Code:

```
prep_h = QuantumCircuit(N_QUBITS)
prep_h.h(TARGET_QUBIT)

h_circ_meas = prepare_and_measure(prep_h, measure_rotation=None,
measure_qubits=None)
h_counts, h_probs = run_measurement(h_circ_meas, simulator, shots=SHOTS,
seed=run_seed)
```

Image:



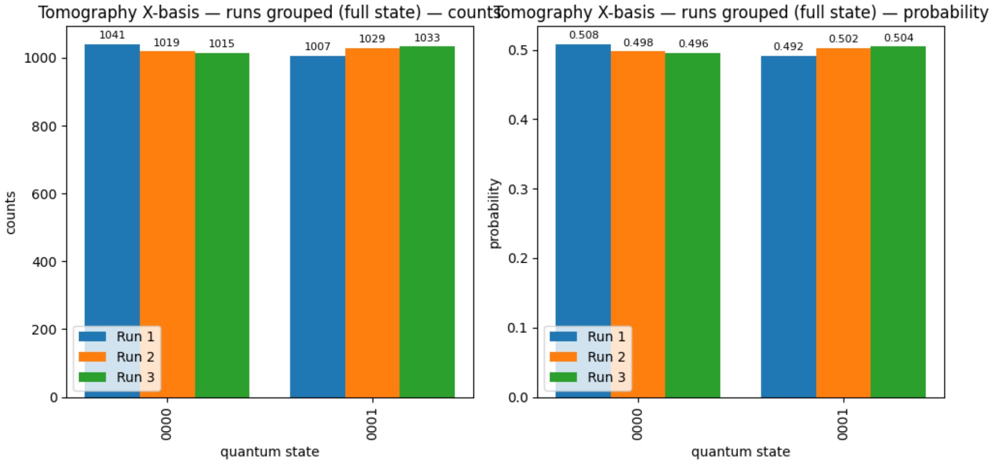
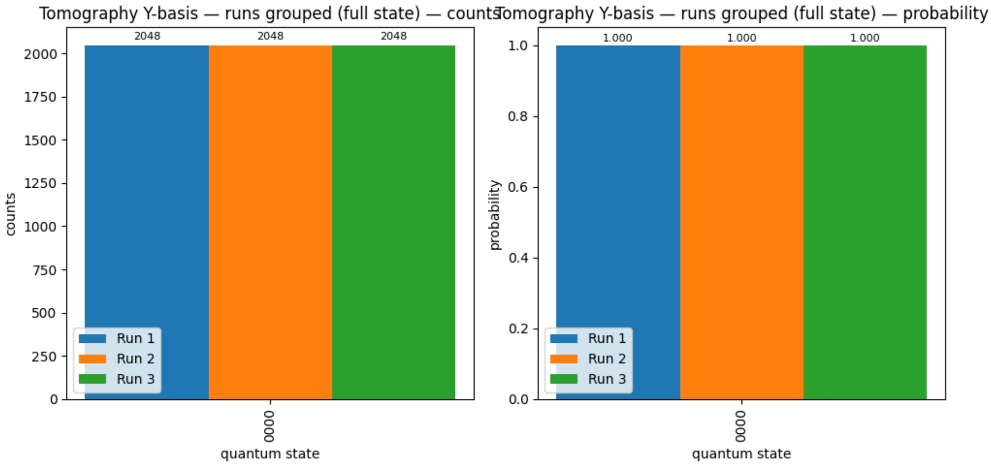
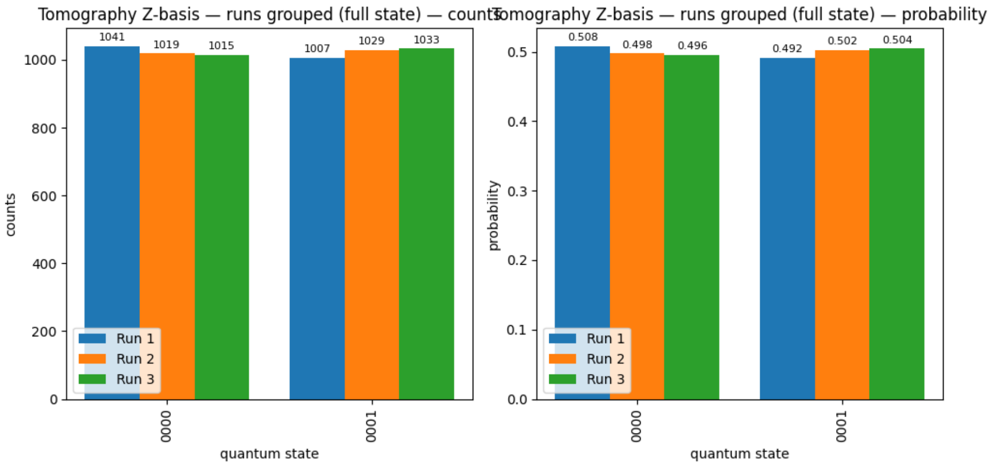
---

4. One-Qubit State Tomography

Measurements across three bases to reconstruct the qubit's state vector.

Code:

```
bases = ['X', 'Y', 'Z']
for basis in bases:
    rot = rotation_for_basis(basis, target_qubit=TARGET_QUBIT)
    circ = prepare_and_measure(prep_zero, measure_rotation=rot,
measure_qubits=None)
    counts_b, probs_b = run_measurement(circ, simulator, shots=SHOTS,
seed=run_seed)
```

Basis	Description	Image
X	Measurement in X-basis	
Y	Measurement in Y-basis	
Z	Measurement in Z-basis	

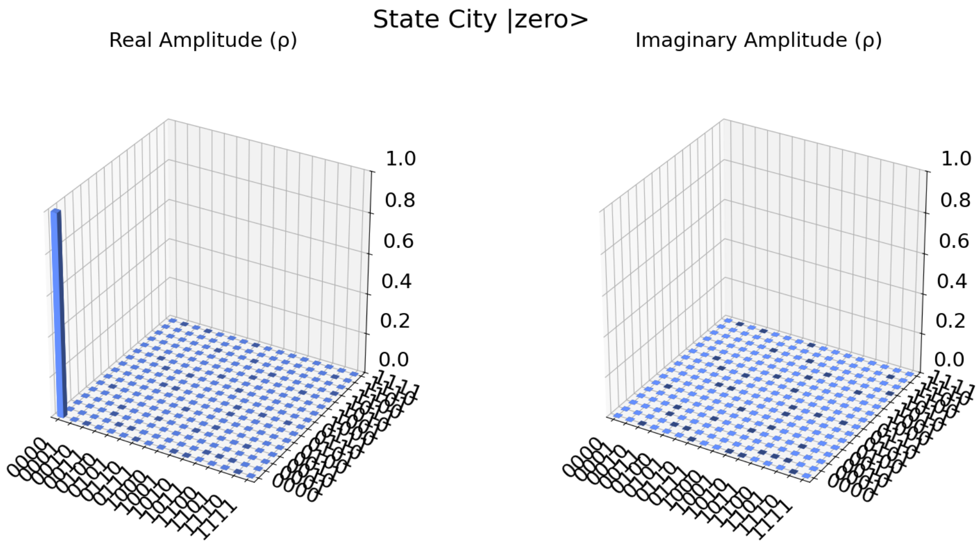
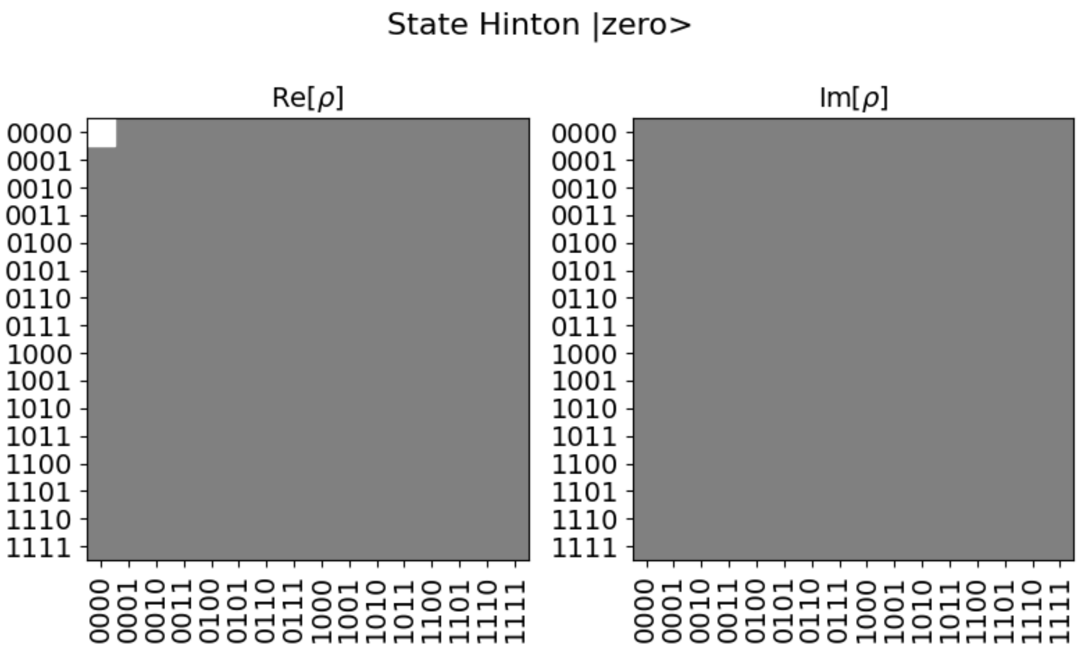
Quantum State Visualizations

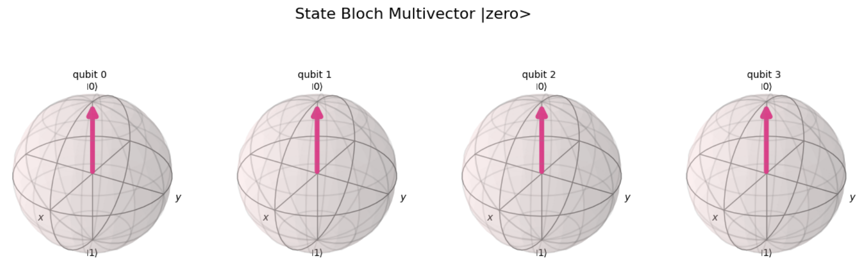
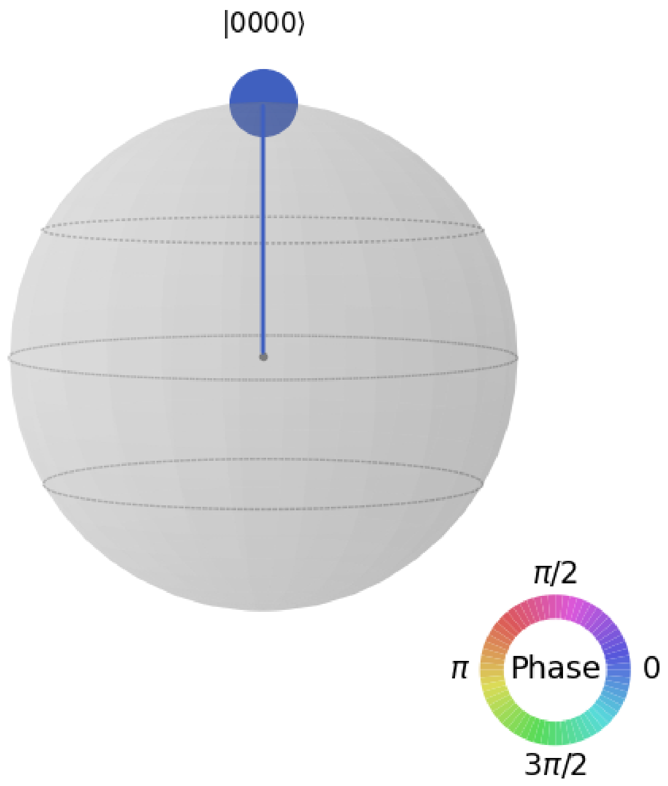
```
vis_circs = {"zero": prep_zero, "x": prep_x, "h": prep_h}
for name, circ in vis_circs.items():
    SV = StatevectorSimulator()
    result = SV.run(circ).result()
    psi = result.get_statevector()

    plot_state_city(psi, title=f"State City |{name}>",
```

```
filename=os.path.join(OUTPUT_DIR, f"state_city_{name}.png"))
    plot_state_hinton(psi, title=f"State Hinton |{name}>",
filename=os.path.join(OUTPUT_DIR, f"state_hinton_{name}.png"))
    plot_bloch_multivector(psi, title=f"State Bloch Multivector |{name}>",
filename=os.path.join(OUTPUT_DIR, f"state_bloch_{name}.png"))
    plot_qsphere = plot_state_qsphere(psi)
    plot_qsphere.savefig(os.path.join(OUTPUT_DIR, f"state_qsphere_{name}.png"))
```

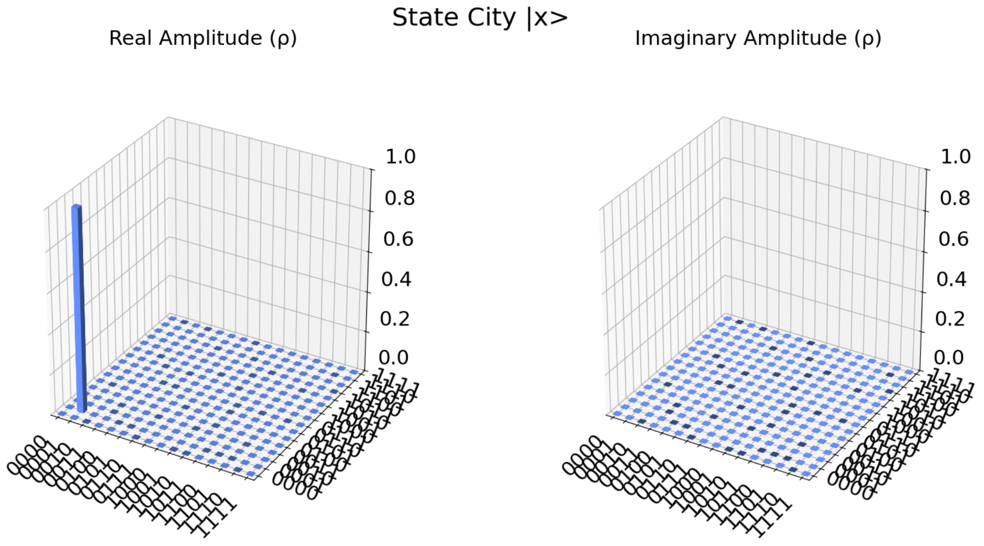
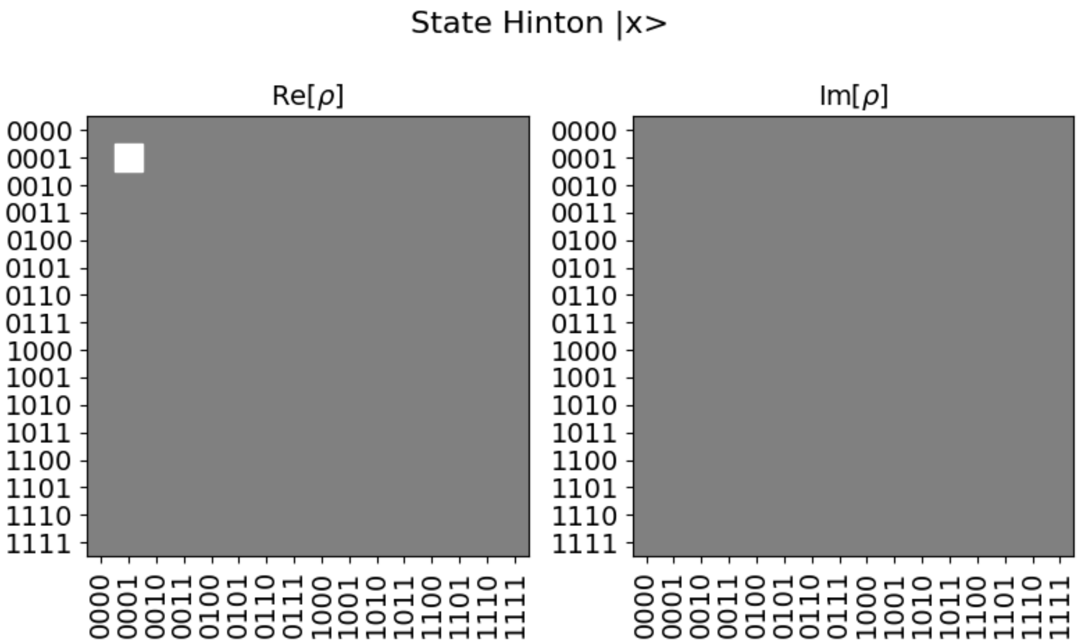
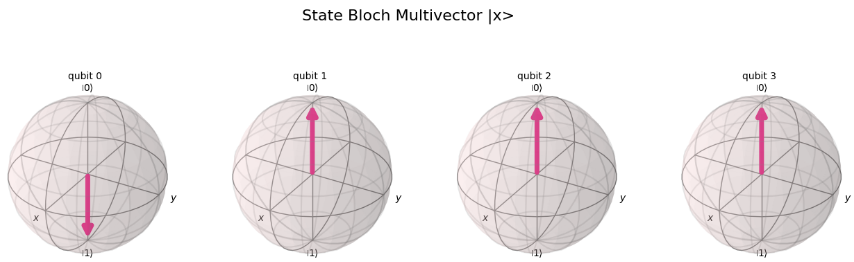
|0000⟩ — Ground State

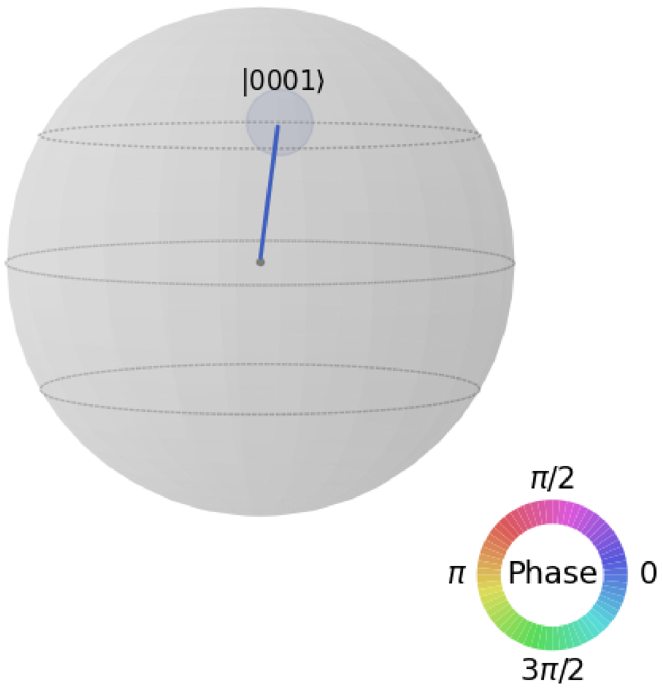
Visualization	Image
City plot	
Hinton plot	

Visualization	Image
Bloch multivector	
Qsphere	

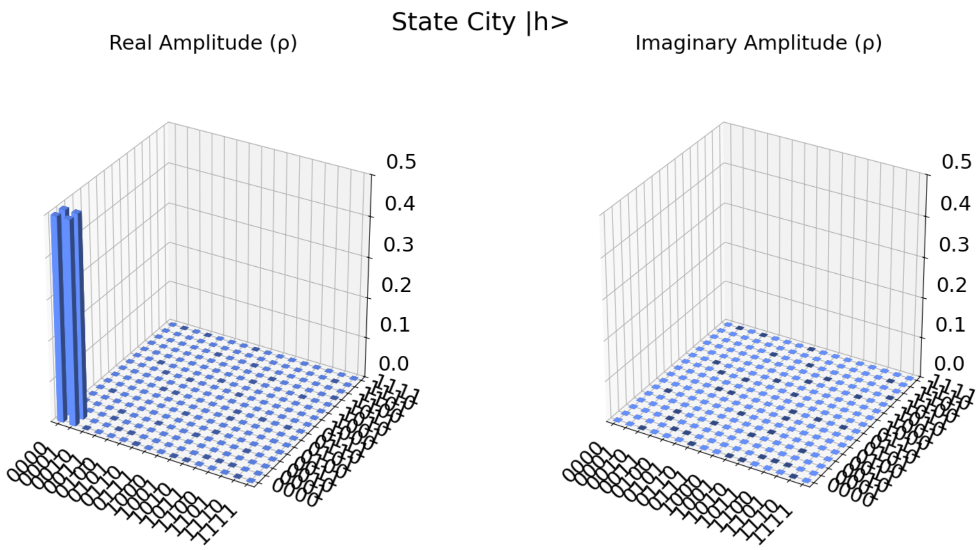
X|0000⟩ — After X Gate

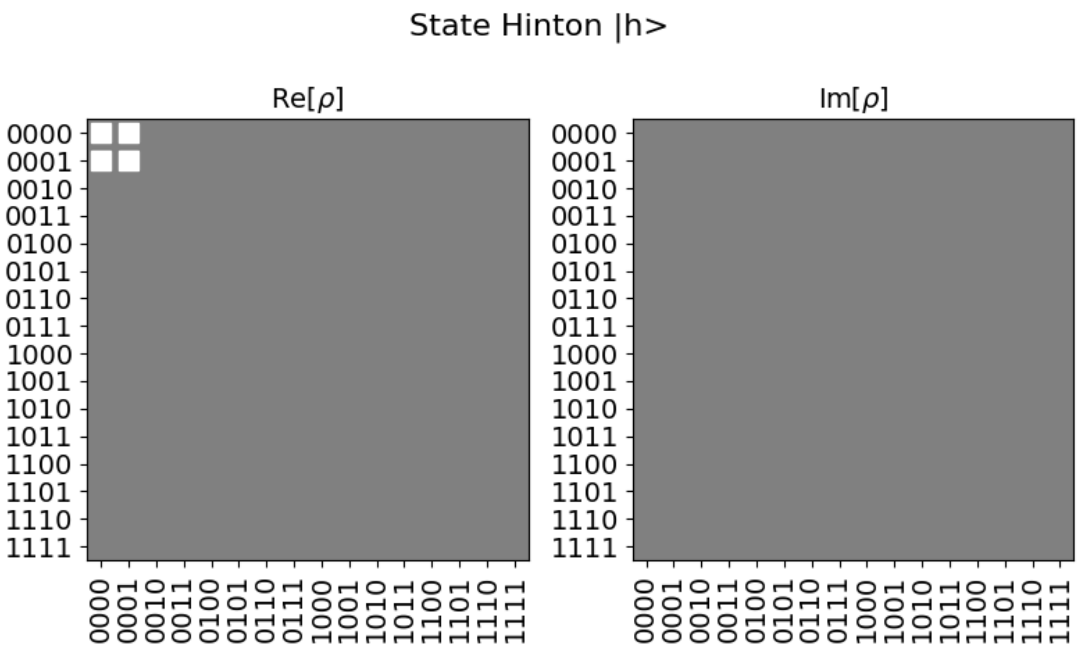
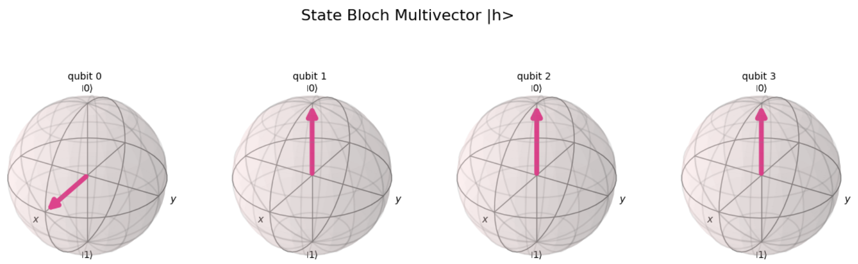
Visualization	Image

<table>
<tr><th>Visualization</th><th>Image</th></tr>
<tr><td>City plot</td><td>
</td></tr>
<tr><td>Hinton plot</td><td>
</td></tr>
<tr><td>Bloch multivector</td><td>
</td></tr>
</table>

Visualization	Image
Qsphere	

H|0000⟩ — After Hadamard Gate

Visualization	Image
City plot	

Visualization	Image
Hinton plot	
Bloch multivector	
Qsphere	

Visualization	Image