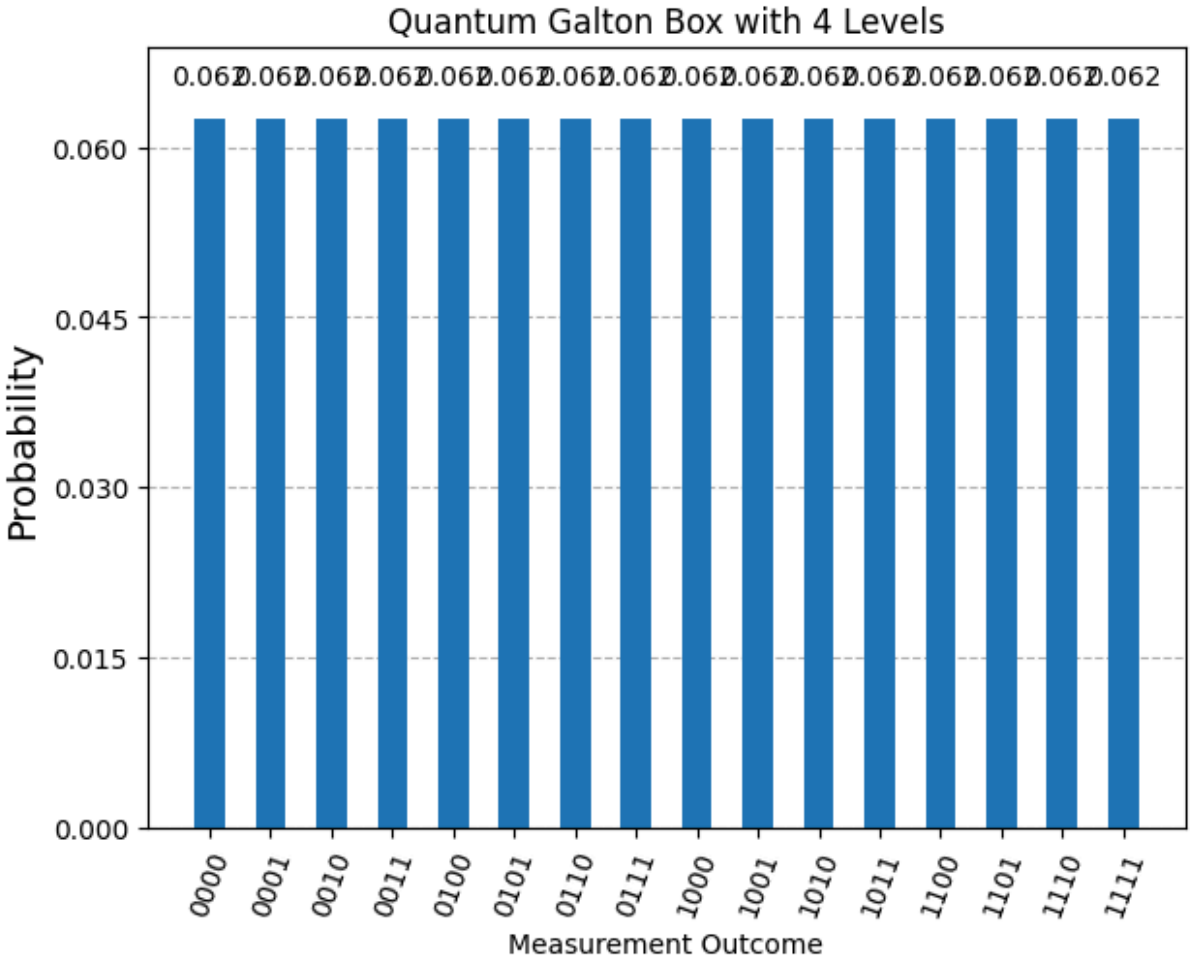
In [1]:

```
!pip install qiskit --quiet

# Ensure plots display inline
%matplotlib inline

from qiskit import QuantumCircuit
from qiskit.quantum_info import Statevector
from qiskit.visualization import plot_histogram
import matplotlib.pyplot as plt

def quantum_galton_box(n_layers=4):
    """
    Simulate a Quantum Galton Box using Hadamard-based quantum walks.

    n_layers: number of levels (qubits)
    """
    qc = QuantumCircuit(n_layers)

    for q in range(n_layers):
        qc.h(q)

    state = Statevector.from_instruction(qc)
    probs = state.probabilities_dict()

    # Plotting the histogram
    fig = plot_histogram(probs)
    plt.title(f"Quantum Galton Box with {n_layers} Levels")
    plt.xlabel("Measurement Outcome")
    plt.ylabel("Probability")
    plt.show()

    return probs

# Run the simulation
quantum_galton_box(n_layers=4)
```



Out[1]:
```
{np.str_('0000'): np.float64(0.06249999999999996),
 np.str_('0001'): np.float64(0.06249999999999996),
 np.str_('0010'): np.float64(0.06249999999999996),
 np.str_('0011'): np.float64(0.06249999999999996),
 np.str_('0100'): np.float64(0.06249999999999996),
 np.str_('0101'): np.float64(0.06249999999999996),
 np.str_('0110'): np.float64(0.06249999999999996),
 np.str_('0111'): np.float64(0.06249999999999996),
 np.str_('1000'): np.float64(0.06249999999999996),
 np.str_('1001'): np.float64(0.06249999999999996),
 np.str_('1010'): np.float64(0.06249999999999996),
 np.str_('1011'): np.float64(0.06249999999999996),
 np.str_('1100'): np.float64(0.06249999999999996),
 np.str_('1101'): np.float64(0.06249999999999996),
 np.str_('1110'): np.float64(0.06249999999999996),
 np.str_('1111'): np.float64(0.06249999999999996)}
```

In [ ]: