Introduction to Quantum Information and Quantum Machine Learning

Project 2

Mateusz Tabaszewski 151945

In [1]:
```
from qiskit import QuantumRegister, ClassicalRegister, QuantumCircuit
from qiskit_aer import Aer
from qiskit.compiler import transpile
from qiskit.visualization import *
from numpy import pi
from qiskit.visualization import plot_histogram
from qiskit.transpiler import generate_preset_pass_manager
```

In [2]:
```
backend = Aer.get_backend('statevector_simulator')
nx=2
shots=2048
```

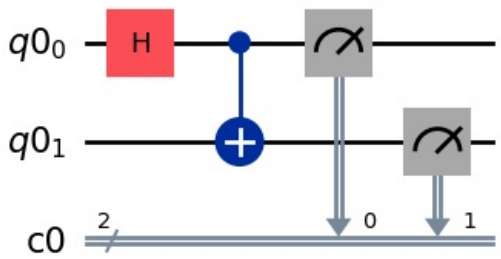
In [3]:
```
def run_simple_circuit(nx, shots, backend, gate_list, measurement=None):
    if measurement is not None:
        if measurement == "XX":
            gate_list += [
                    ("barrier", (0, 1)),
                    ("h", 0),
                    ("h", 1),
                    ("measure", (0, 0)),
                    ("measure", (1, 1))
            ]
        elif measurement == "YY":
            gate_list += [
                    ("barrier", (0, 1)),
                    ("sdg", 0),
                    ("sdg", 1),
                    ("h", 0),
                    ("h", 1),
                    ("measure", (0, 0)),
                    ("measure", (1, 1))
            ]
        elif measurement == "XZ":
            gate_list += [
                    ("barrier", (0, 1)),
                    ("measure", (0, 0)),
                    ("h", 1),
                    ("measure", (1, 1))
            ]
        else:
            raise ValueError("Invalid Measurement Type")

    qx = QuantumRegister(nx)
    cx = ClassicalRegister(nx)
    circuit = QuantumCircuit(qx, cx)

    for gate, qubits in gate_list:
        if gate == "x":
            circuit.x(qx[qubits])
        elif gate == "h":
            circuit.h(qx[qubits])
        elif gate == "sdg":
            circuit.sdg(qx[qubits])
        elif gate == "cx":
            control, target = qubits
            circuit.cx(qx[control], qx[target])
        elif gate == "barrier":
            if isinstance(qubits, tuple):
                circuit.barrier(qx[qubits[0]], qx[qubits[1]])
            else:
                circuit.barrier(qx[qubits])
        elif gate == "measure":
            q, c = qubits
            circuit.measure(qx[q], cx[c])

    results = []
    for i in range(3):
        job_result = backend.run(transpile(circuit, backend), shots=shots).result()
        results.append(job_result)
```

```
    job_result = results[-1]
    counts = job_result.get_counts(circuit)
    return circuit, results, counts
```

Task 1

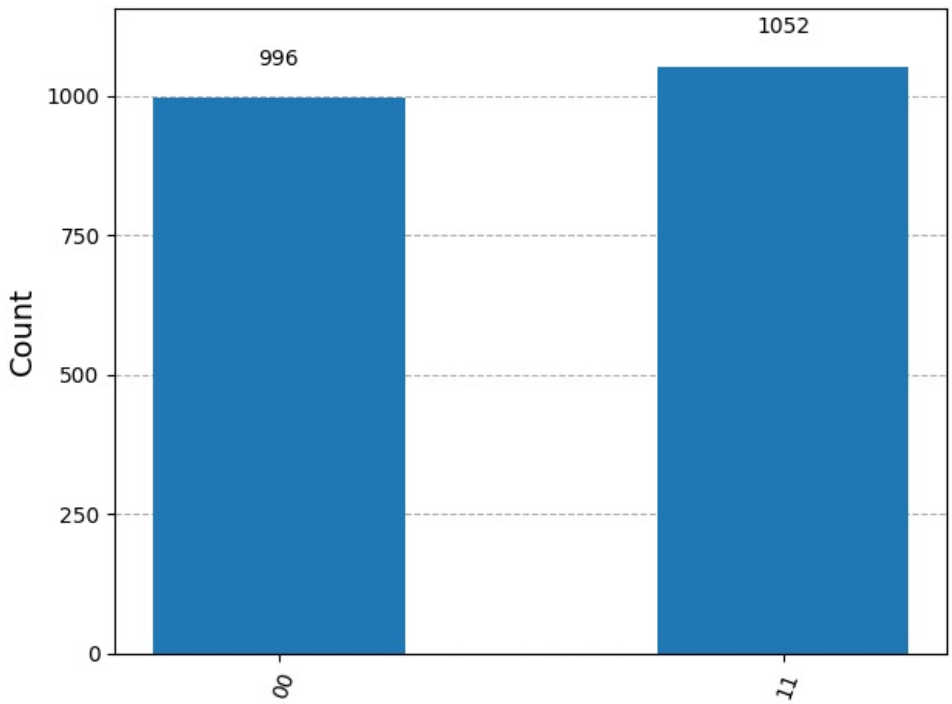
In [4]:
```
gates = [
    ("h", 0),
    ("cx", (0, 1)),
    ("measure", (0, 0)),
    ("measure", (1, 1))
]

circuit, results, counts = run_simple_circuit(
    nx=nx,
    shots=shots,
    backend=backend,
    gate_list=gates
)

circuit.draw(output="mpl")
```

Out[4]:



In [5]:
```
plot_histogram(counts)
```

Out[5]:

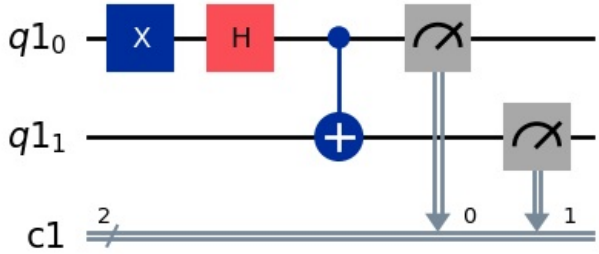


Task 2

In [6]:
```
gates = [
    ("x", 0),
    ("h", 0),
    ("cx", (0, 1)),
    ("measure", (0, 0)),
    ("measure", (1, 1))
]

circuit, results, counts = run_simple_circuit(
    nx=nx,
    shots=shots,
```

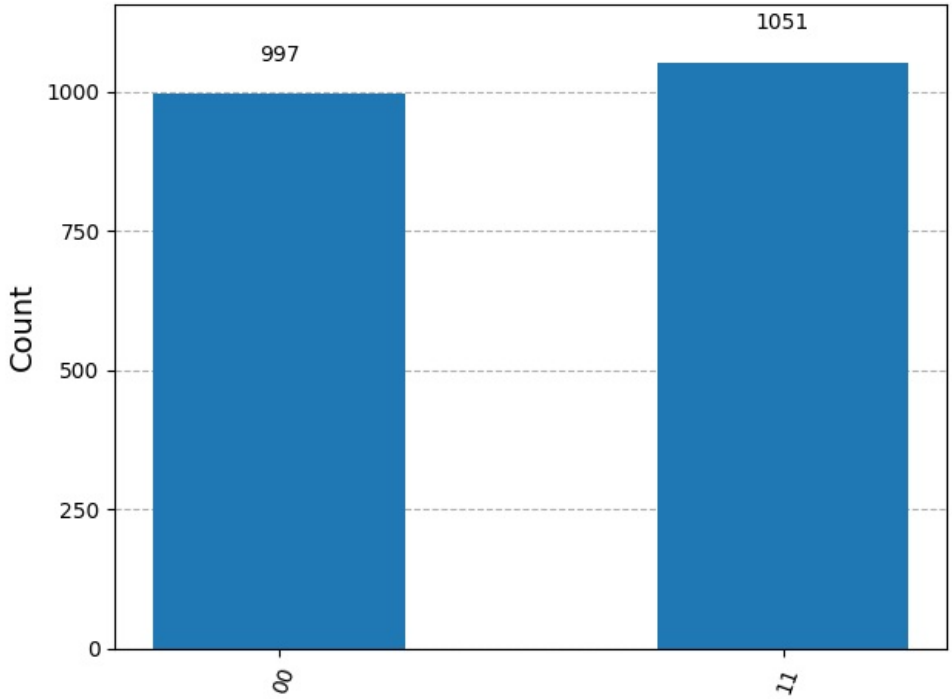
```
    backend=backend,
    gate_list=gates
)

circuit.draw(output="mpl")
```

Out[6]:



In [7]:
```
plot_histogram(counts)
```

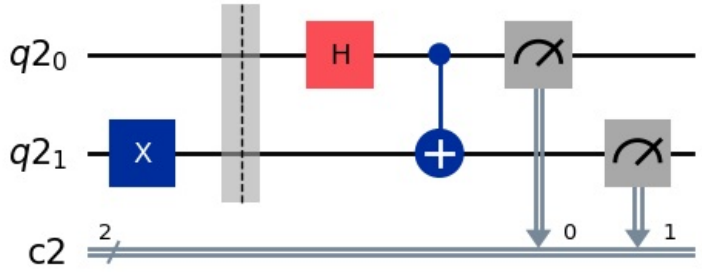
Out[7]:



Task 3

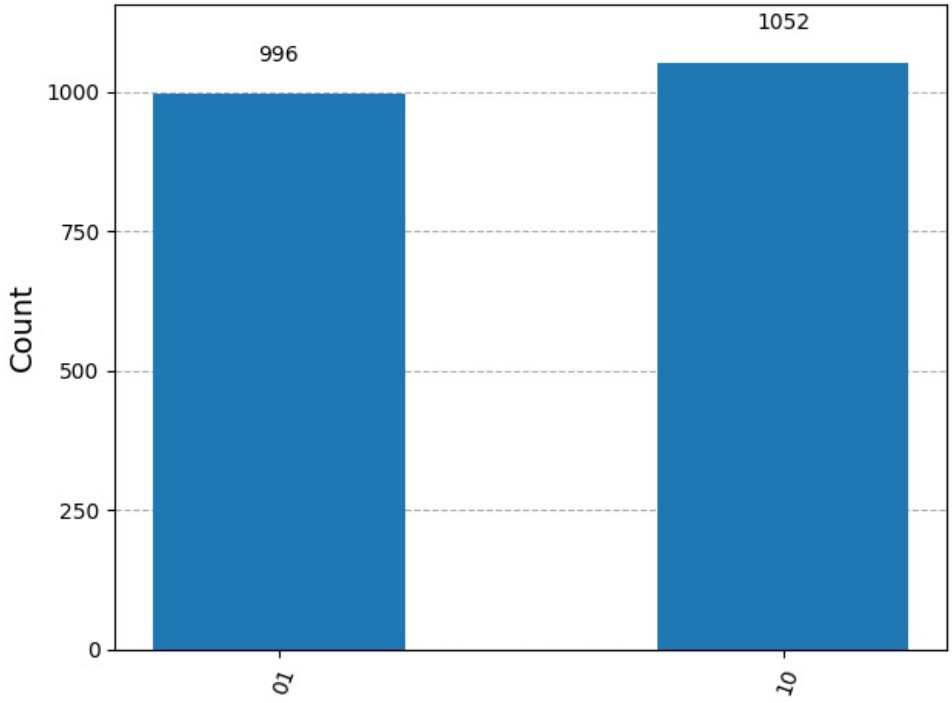
In [8]:
```
gates = [
    ("x", 1),
    ("barrier", (0, 1)),
    ("h", 0),
    ("cx", (0, 1)),
    ("measure", (0, 0)),
    ("measure", (1, 1))
]

circuit, results, counts = run_simple_circuit(
    nx=nx,
    shots=shots,
    backend=backend,
    gate_list=gates
)

circuit.draw(output="mpl")
```

Out[8]:



In [9]:
```
plot_histogram(counts)
```

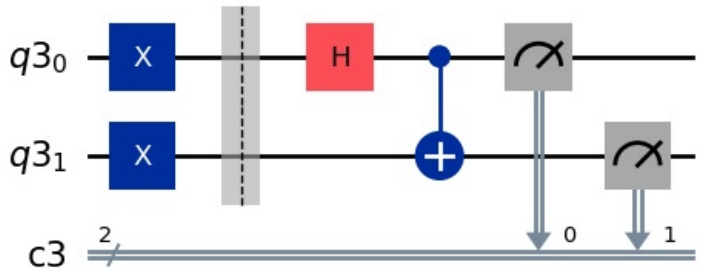
Out[9]:



Task 4

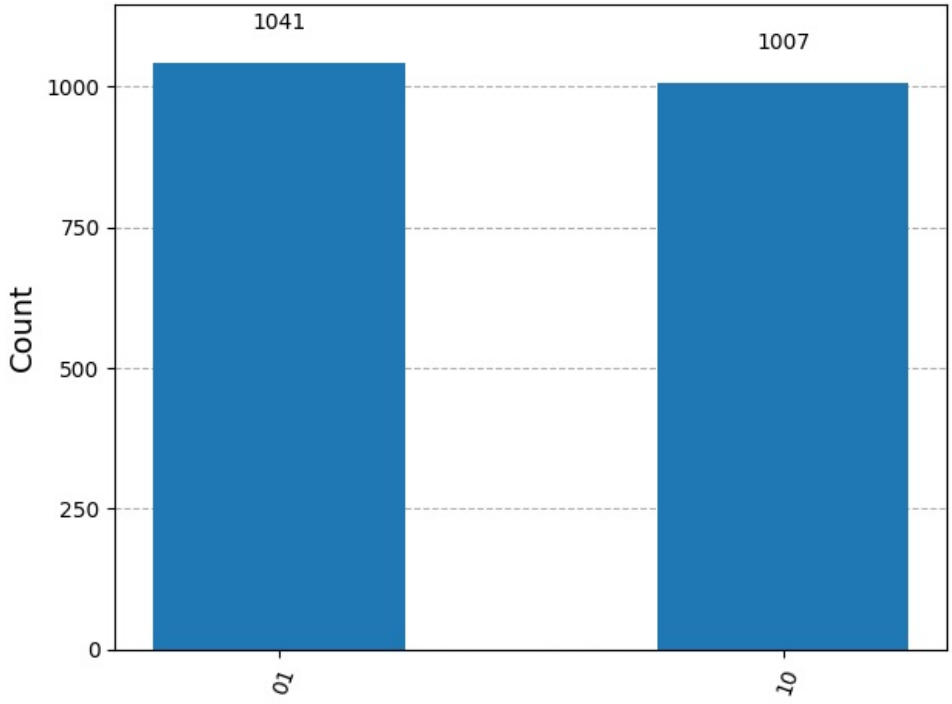
In [10]:
```
gates = [
    ("x", 0),
    ("x", 1),
    ("barrier", (0, 1)),
    ("h", 0),
    ("cx", (0, 1)),
    ("measure", (0, 0)),
    ("measure", (1, 1))
]

circuit, results, counts = run_simple_circuit(
    nx=nx,
    shots=shots,
    backend=backend,
    gate_list=gates
)

circuit.draw(output="mpl")
```

Out[10]:



In [11]:
```
plot_histogram(counts)
```

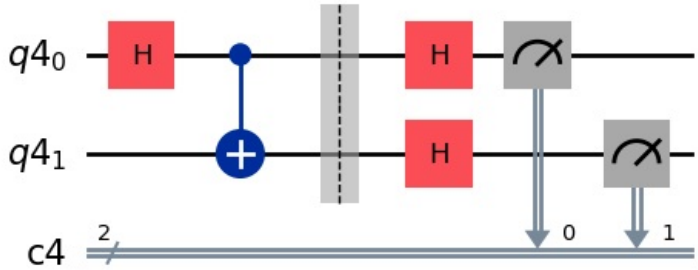
Out[11]:



Task 5

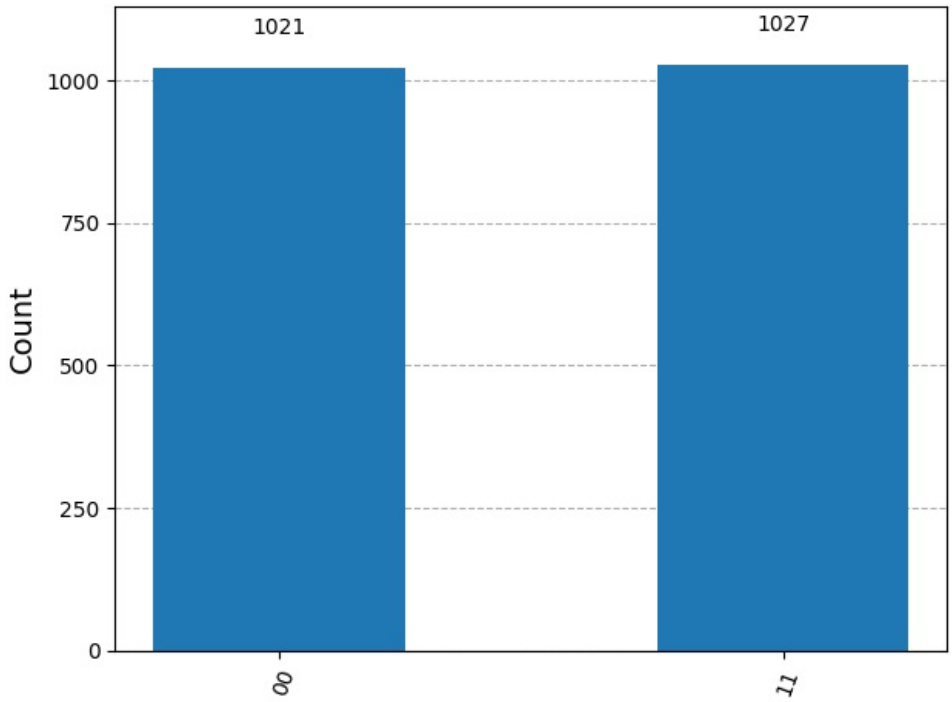
In [12]:
```
gates = [
    ("h", 0),
    ("cx", (0, 1))
]

circuit, results, counts = run_simple_circuit(
    nx=nx,
    shots=shots,
    backend=backend,
    gate_list=gates,
    measurement="XX"
)

circuit.draw(output="mpl")
```

Out[12]:



In [13]:
```
plot_histogram(counts)
```

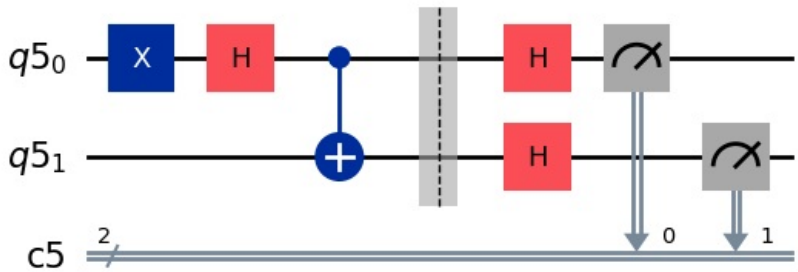
Out[13]:



In [14]:

```
gates = [
    ("x", 0),
    ("h", 0),
    ("cx", (0, 1))
]

circuit, results, counts = run_simple_circuit(
    nx=nx,
    shots=shots,
    backend=backend,
    gate_list=gates,
    measurement="XX"
)

circuit.draw(output="mpl")
```

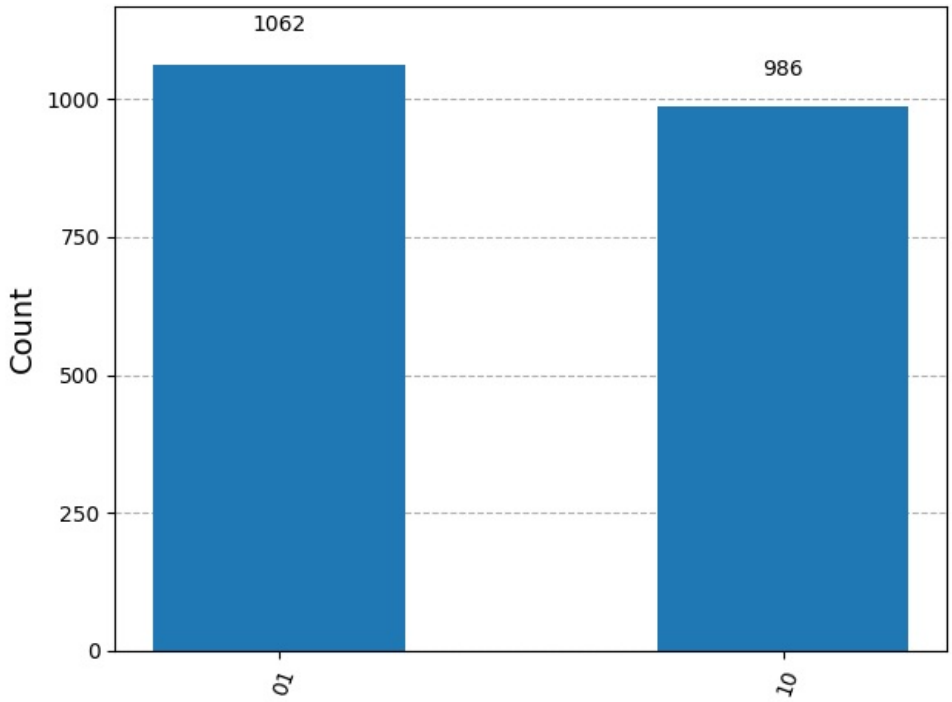
Out[14]:



In [15]:

```
plot_histogram(counts)
```

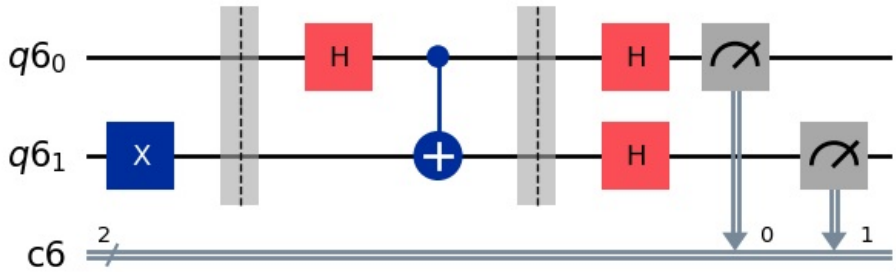
Out[15]:



In [16]:

```
gates = [
    ("x", 1),
    ("barrier", (0, 1)),
    ("h", 0),
    ("cx", (0, 1))
]

circuit, results, counts = run_simple_circuit(
    nx=nx,
    shots=shots,
    backend=backend,
    gate_list=gates,
    measurement="XX"
)

circuit.draw(output="mpl")
```

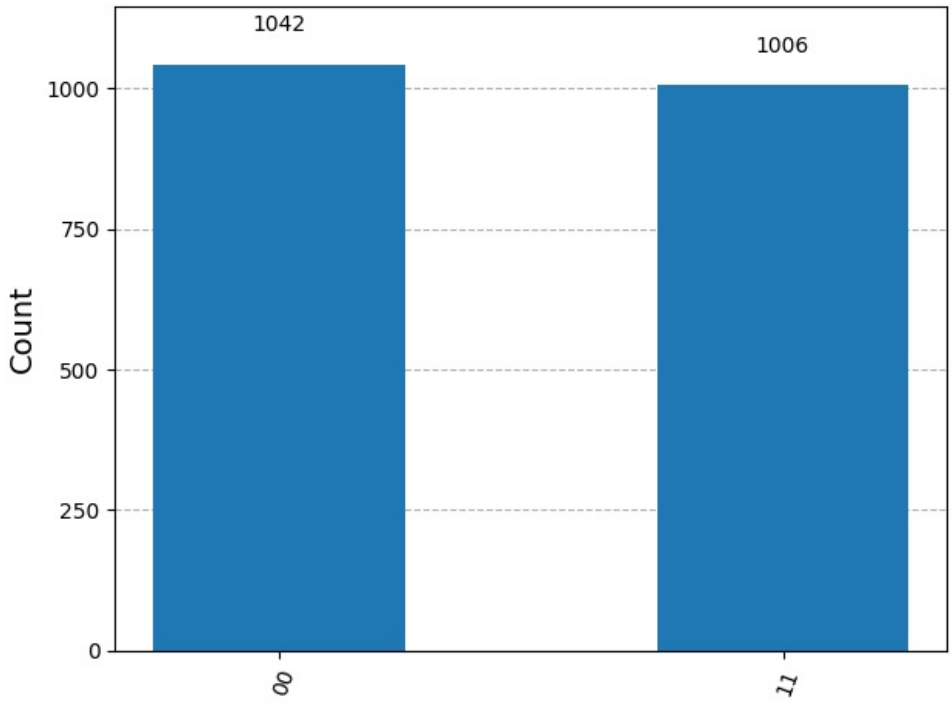
Out[16]:



In [17]:

```
plot_histogram(counts)
```

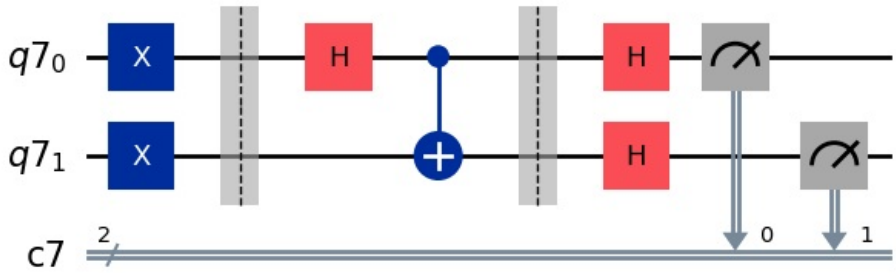
Out[17]:



In [18]:

```
gates = [
    ("x", 0),
    ("x", 1),
    ("barrier", (0, 1)),
    ("h", 0),
    ("cx", (0, 1))
]

circuit, results, counts = run_simple_circuit(
    nx=nx,
    shots=shots,
    backend=backend,
    gate_list=gates,
    measurement="XX"
)

circuit.draw(output="mpl")
```

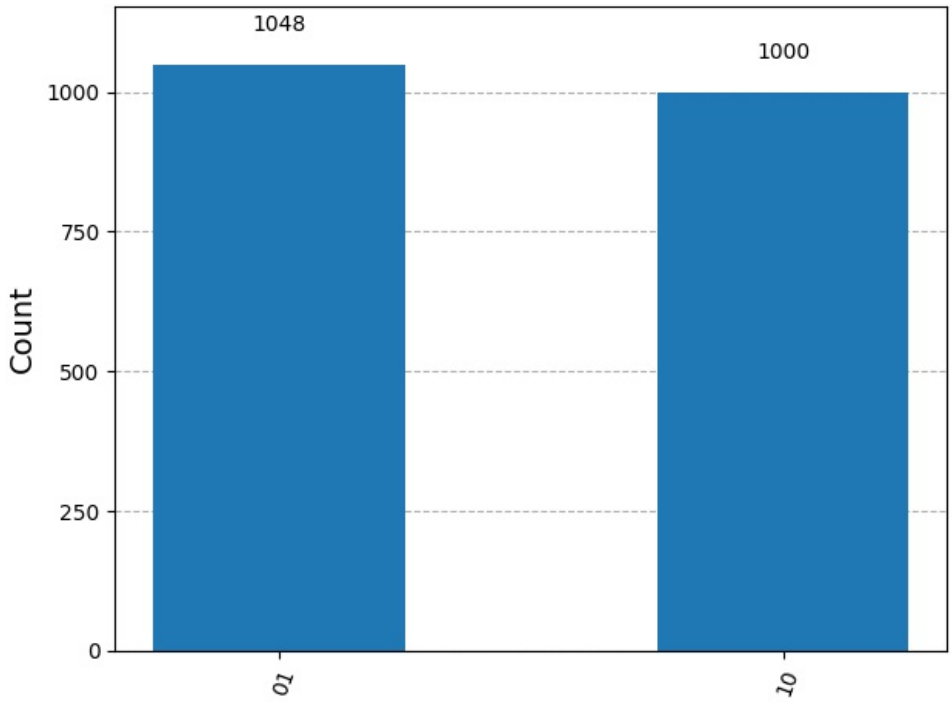
Out[18]:



In [19]:

```
plot_histogram(counts)
```

Out[19]:

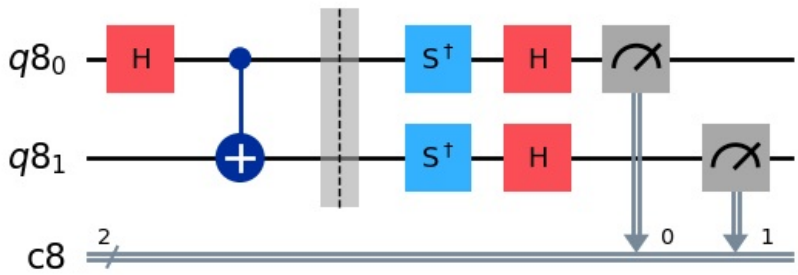


Task 6

In [20]:

```
gates = [
    ("h", 0),
    ("cx", (0, 1)),
]

circuit, results, counts = run_simple_circuit(
    nx=nx,
    shots=shots,
    backend=backend,
    gate_list=gates,
    measurement="YY"
)

circuit.draw(output="mpl")
```

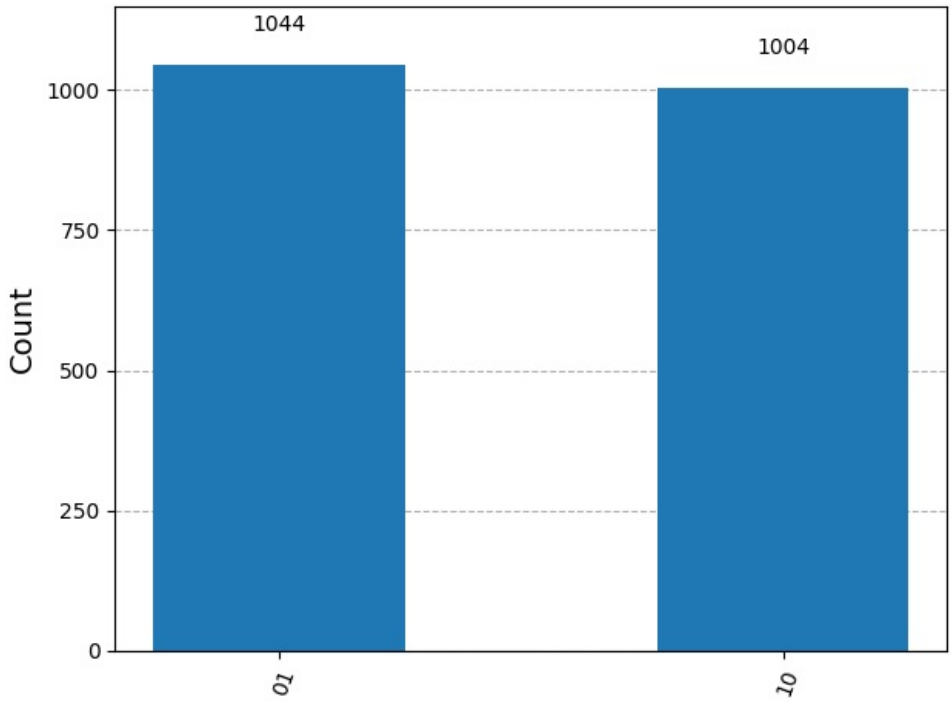
Out[20]:



In [21]:

```
plot_histogram(counts)
```

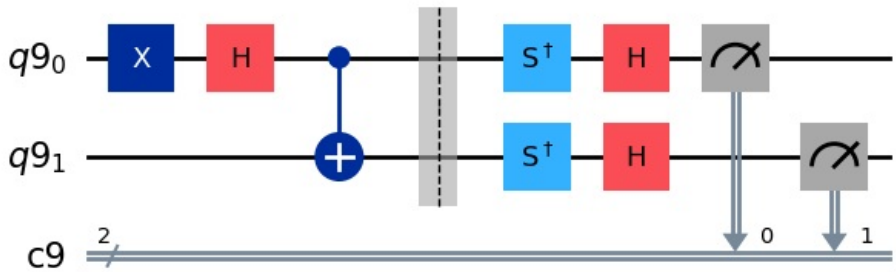
Out[21]:



In [22]:

```
gates = [
    ("x", 0),
    ("h", 0),
    ("cx", (0, 1))
]

circuit, results, counts = run_simple_circuit(
    nx=nx,
    shots=shots,
    backend=backend,
    gate_list=gates,
    measurement="YY"
)

circuit.draw(output="mpl")
```

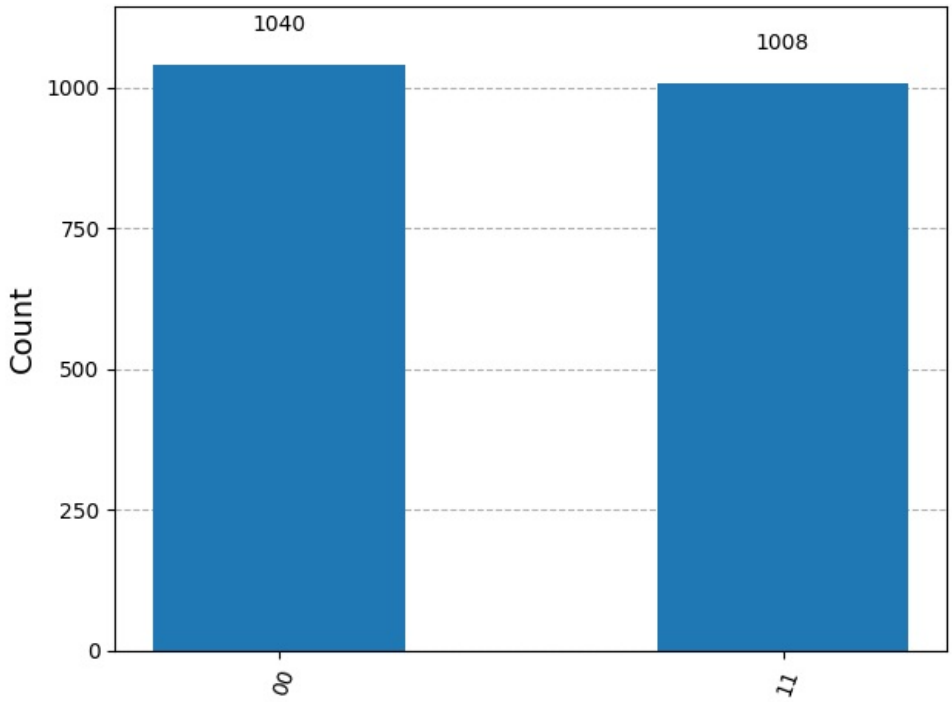
Out[22]:



In [23]:

```
plot_histogram(counts)
```

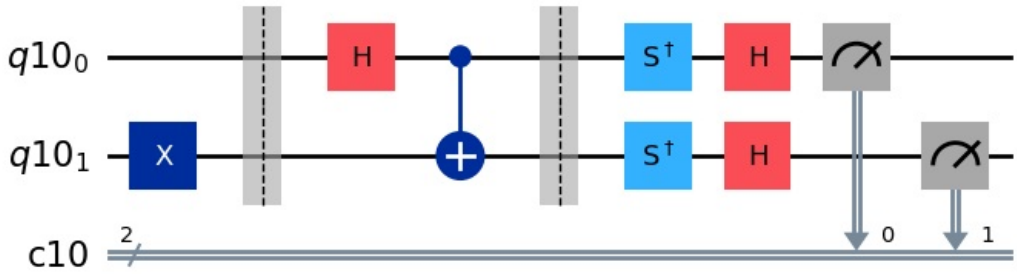
Out[23]:



In [24]:

```
gates = [
    ("x", 1),
    ("barrier", (0, 1)),
    ("h", 0),
    ("cx", (0, 1))
]

circuit, results, counts = run_simple_circuit(
    nx=nx,
    shots=shots,
    backend=backend,
    gate_list=gates,
    measurement="YY"
)

circuit.draw(output="mpl")
```

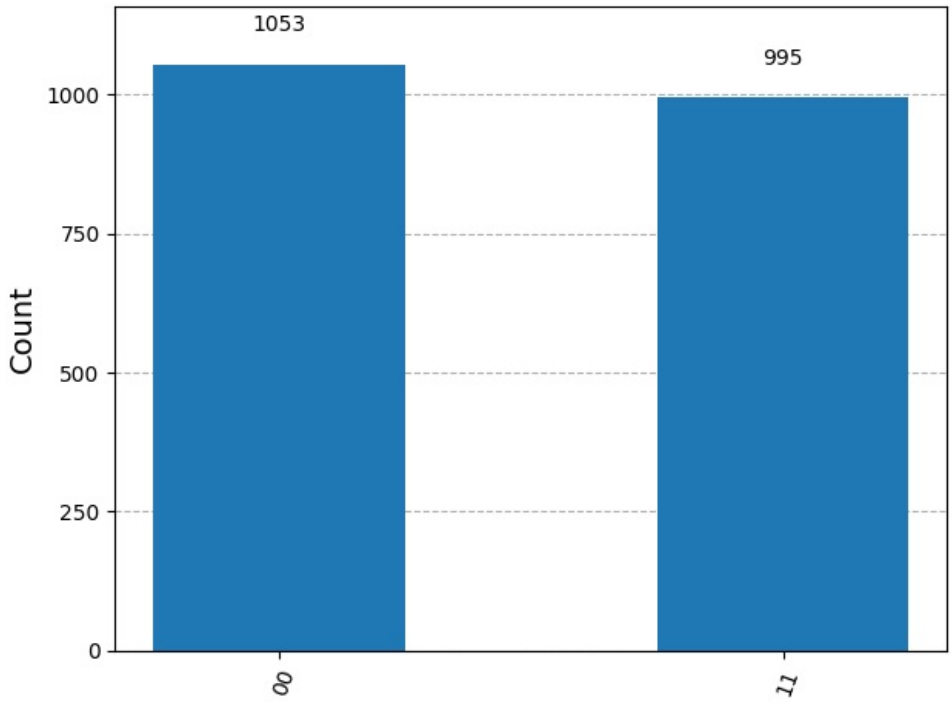
Out[24]:



In [25]:

```
plot_histogram(counts)
```

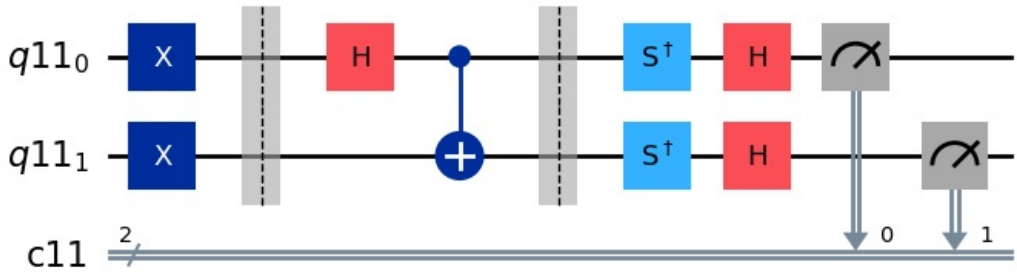
Out[25]:



In [26]:

```
gates = [
    ("x", 0),
    ("x", 1),
    ("barrier", (0, 1)),
    ("h", 0),
    ("cx", (0, 1))
]

circuit, results, counts = run_simple_circuit(
    nx=nx,
    shots=shots,
    backend=backend,
    gate_list=gates,
    measurement="YY"
)

circuit.draw(output="mpl")
```

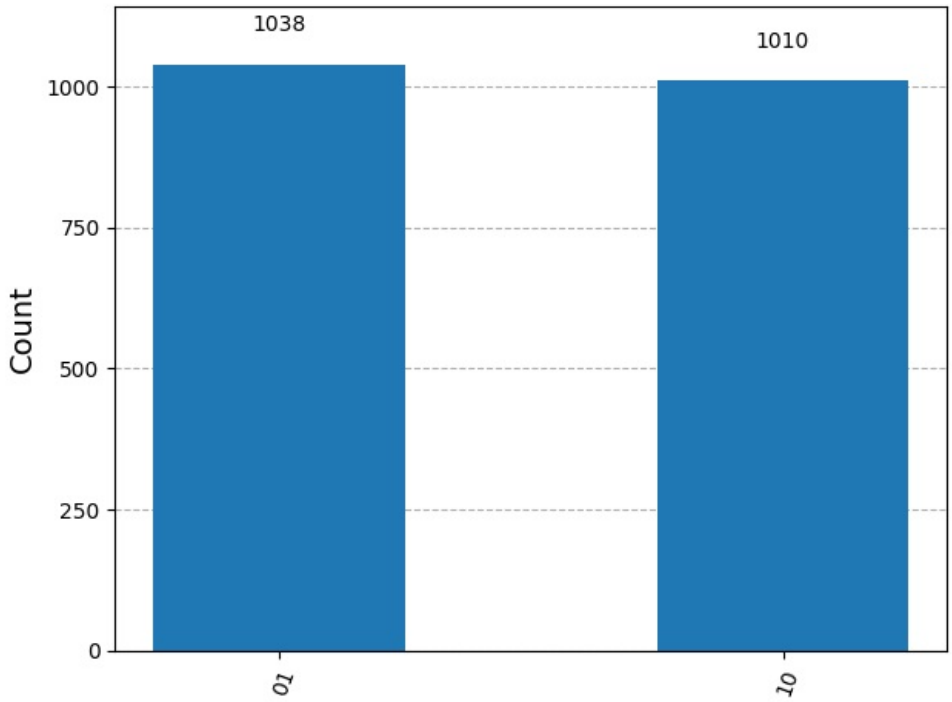
Out[26]:



In [27]:

```
plot_histogram(counts)
```

Out[27]:

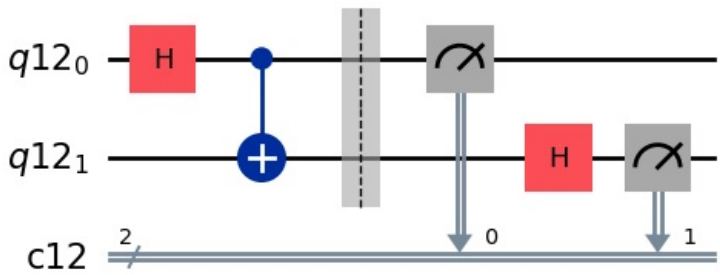


Task 7

In [28]:

```
gates = [
    ("h", 0),
    ("cx", (0, 1))
]

circuit, results, counts = run_simple_circuit(
    nx=nx,
    shots=shots,
    backend=backend,
    gate_list=gates,
    measurement="XZ"
)

circuit.draw(output="mpl")
```

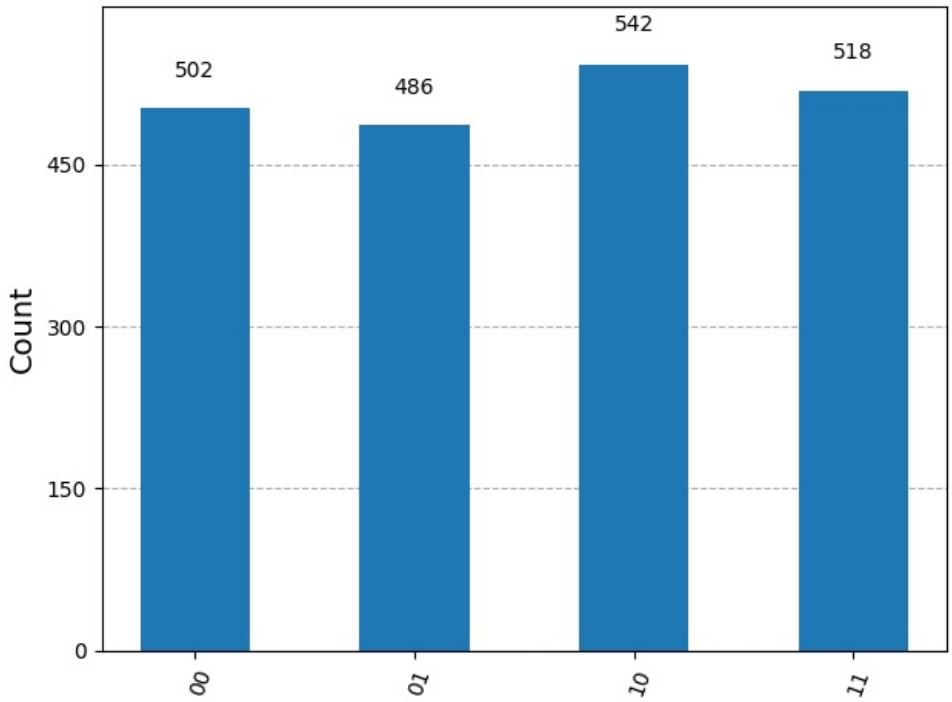
Out[28]:



In [29]:

```
plot_histogram(counts)
```

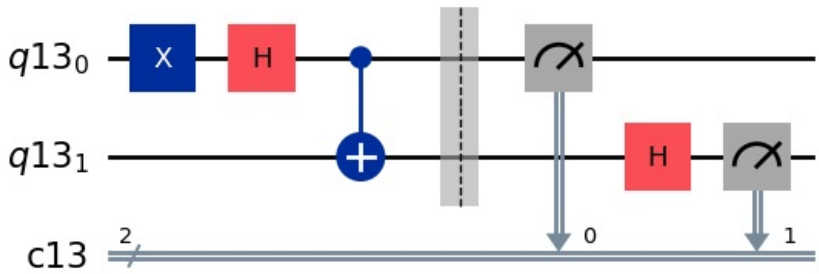
Out[29]:



In [30]:

```
gates = [
    ("x", 0),
    ("h", 0),
    ("cx", (0, 1))
]

circuit, results, counts = run_simple_circuit(
    nx=nx,
    shots=shots,
    backend=backend,
    gate_list=gates,
    measurement="XZ"
)

circuit.draw(output="mpl")
```

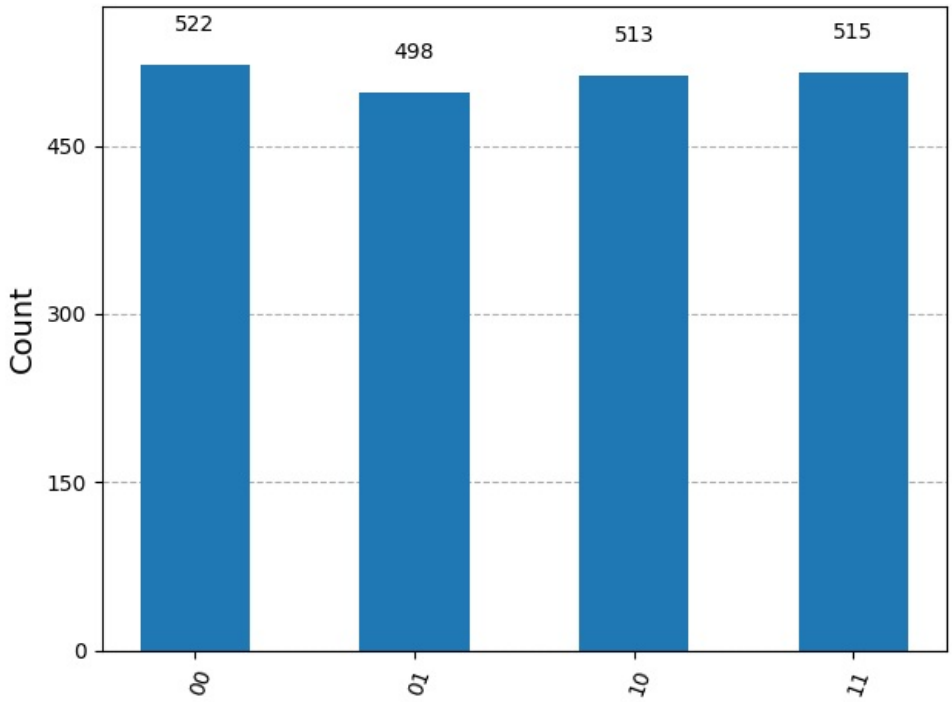
Out[30]:



In [31]:

```
plot_histogram(counts)
```

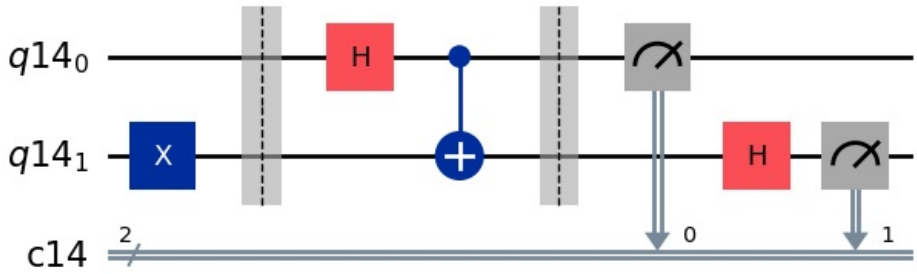
Out[31]:



In [32]:

```
gates = [
    ("x", 1),
    ("barrier", (0, 1)),
    ("h", 0),
    ("cx", (0, 1))
]

circuit, results, counts = run_simple_circuit(
    nx=nx,
    shots=shots,
    backend=backend,
    gate_list=gates,
    measurement="XZ"
)

circuit.draw(output="mpl")
```

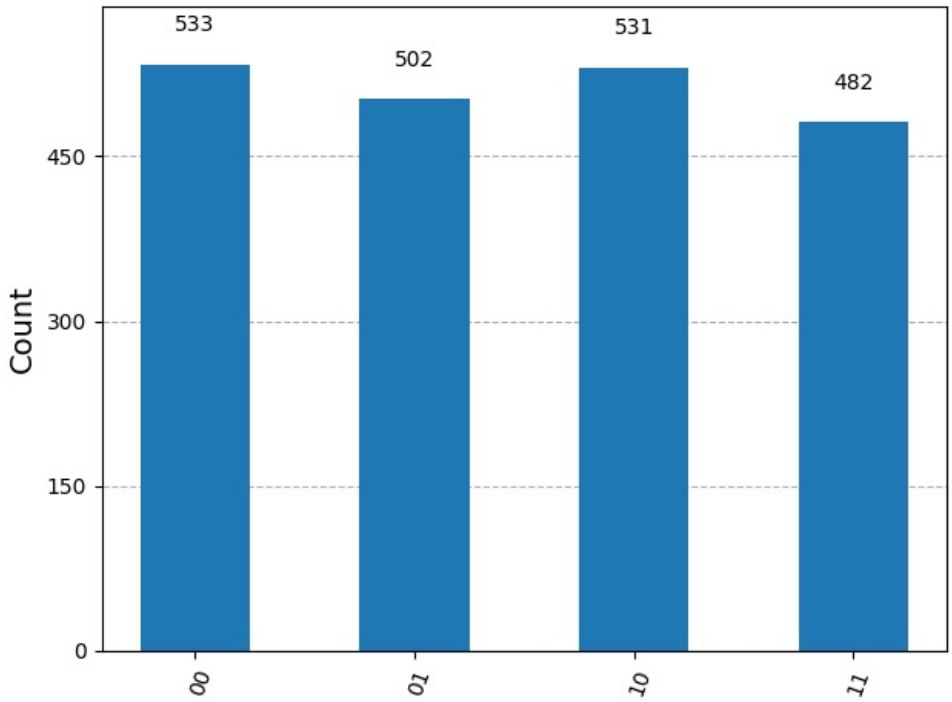
Out[32]:



In [33]:

```
plot_histogram(counts)
```

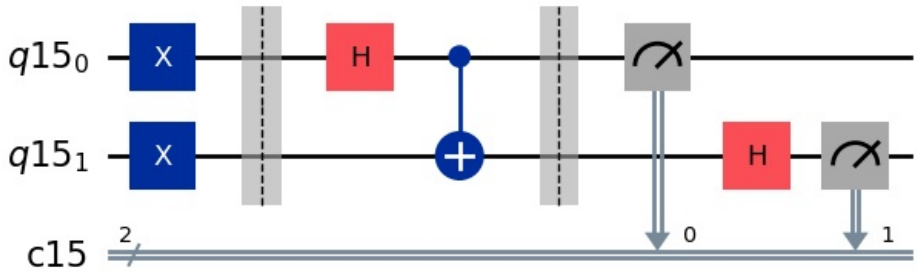
Out[33]:



In [34]:

```
gates = [
    ("x", 0),
    ("x", 1),
    ("barrier", (0, 1)),
    ("h", 0),
    ("cx", (0, 1))
]

circuit, results, counts = run_simple_circuit(
    nx=nx,
    shots=shots,
    backend=backend,
    gate_list=gates,
    measurement="XZ"
)

circuit.draw(output="mpl")
```

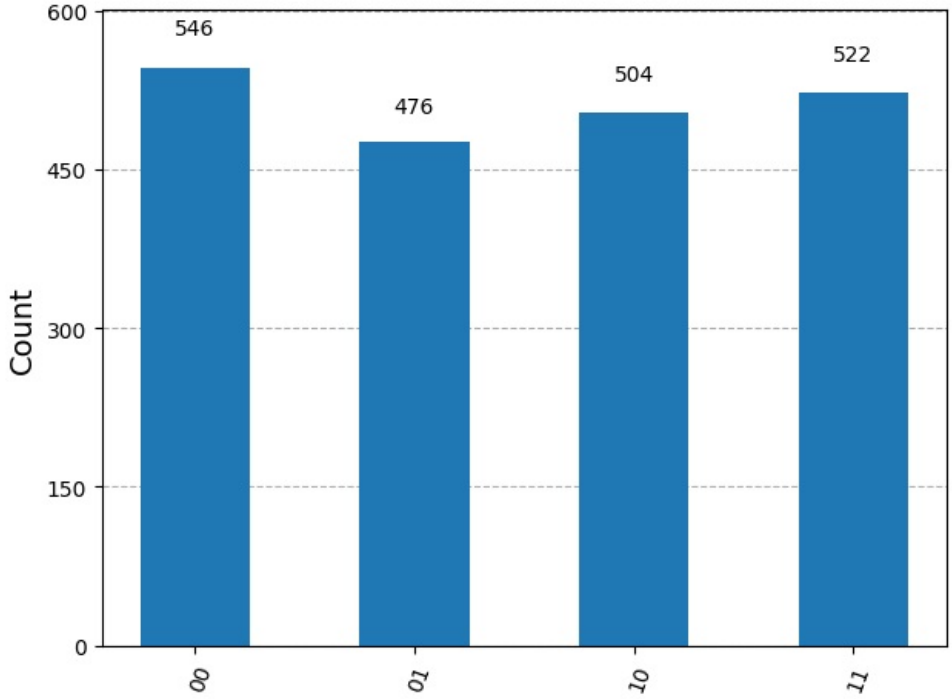
Out[34]:



In [35]:

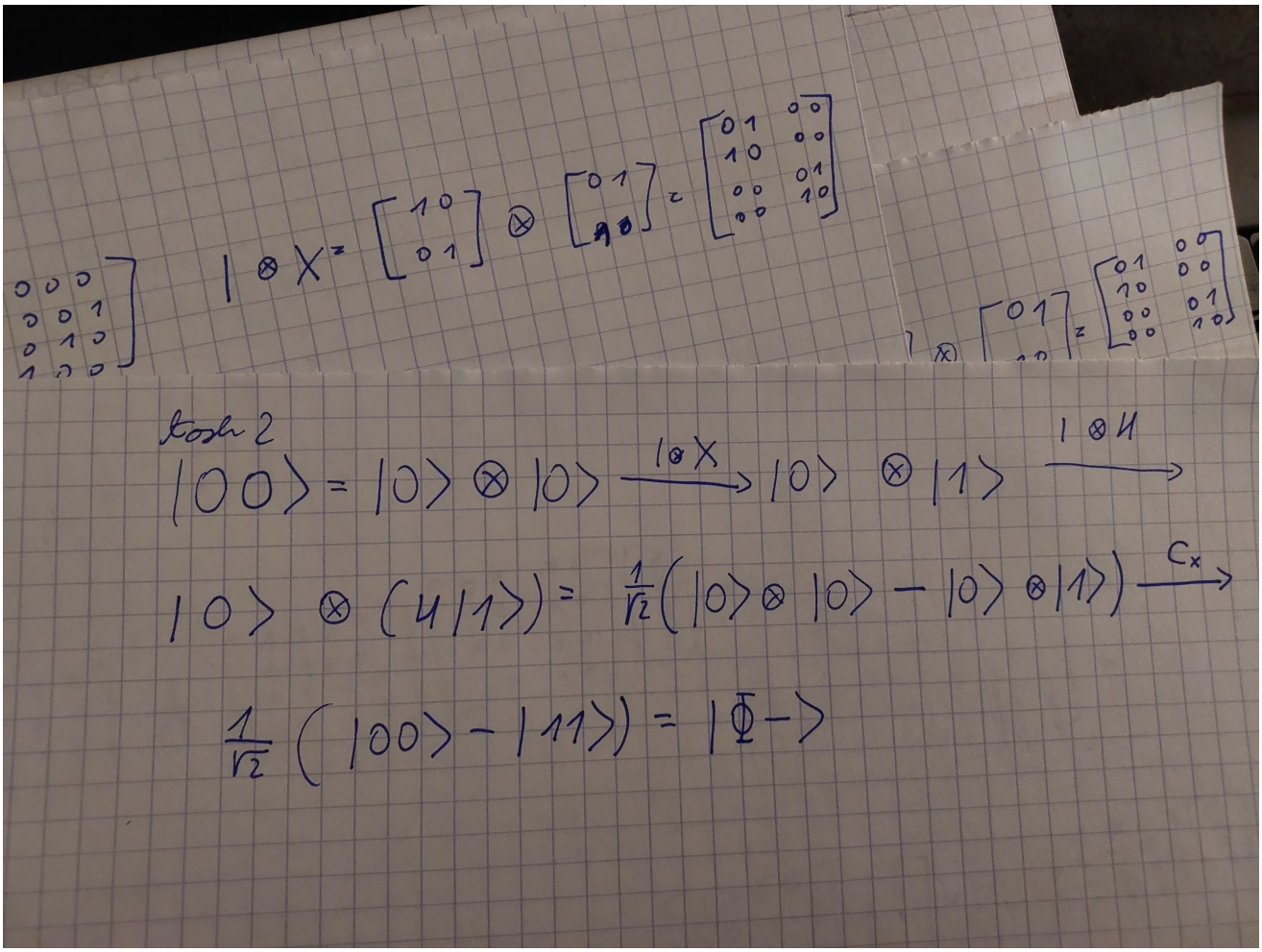
```
plot_histogram(counts)
```

Out[35]:



Calculations

Task 2



task 2

$$C_x = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \qquad I \otimes X = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \otimes \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$I \otimes H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} =$$

$$= \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 1 & 1 \\ 1 & -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 1 \\ -1 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} =$$

$$= \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 0 \\ 0 \\ -1 \end{bmatrix} = \frac{1}{\sqrt{2}} \left(|00\rangle - |11\rangle \right) = |\Phi-\rangle$$

Task 3

$$\frac{1}{\sqrt{2}} \left(|00\rangle - |11\rangle \right) = |\Phi-\rangle$$

task 3

$$|00\rangle = |0\rangle \otimes |0\rangle \xrightarrow{X \otimes I} |1\rangle \otimes |0\rangle \xrightarrow{I \otimes H}$$

$$|1\rangle \otimes \frac{1}{\sqrt{2}} \left(|0\rangle + |1\rangle \right) = \frac{1}{\sqrt{2}} \left(|1\rangle \otimes |0\rangle + |1\rangle \otimes |1\rangle \right) =$$

$$= \frac{1}{\sqrt{2}} \left(|10\rangle + |01\rangle \right) = |\Psi+\rangle$$

task 3

$$C_x = \begin{bmatrix} 1&0&0&0\\ 0&0&0&1\\ 0&0&1&0\\ 0&1&0&0 \end{bmatrix} \qquad X \otimes I = \begin{bmatrix} 0&0&1&0\\ 0&0&0&1\\ 1&0&0&0\\ 0&1&0&0 \end{bmatrix}$$

$$I \otimes H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1&1&0&0\\ 1&-1&0&0\\ 0&0&1&1\\ 0&0&1&-1 \end{bmatrix}$$

$$\begin{bmatrix} 1&0&0&0\\ 0&0&0&1\\ 0&0&1&0\\ 0&1&0&0 \end{bmatrix} \frac{1}{\sqrt{2}} \begin{bmatrix} 1&1&0&0\\ 1&-1&0&0\\ 0&0&1&1\\ 0&0&1&-1 \end{bmatrix} \begin{bmatrix} 0&0&1&0\\ 0&0&0&1\\ 1&0&0&0\\ 0&1&0&0 \end{bmatrix} \begin{bmatrix} 1\\0\\0\\0 \end{bmatrix} =$$

$$= \frac{1}{\sqrt{2}} \begin{bmatrix} 1&1&0&0\\ 0&0&1&-1\\ 0&0&1&1\\ 1&-1&0&0 \end{bmatrix} \begin{bmatrix} 0&0&1&0\\ 0&0&0&1\\ 1&0&0&0\\ 0&1&0&0 \end{bmatrix} \begin{bmatrix} 1\\0\\0\\0 \end{bmatrix} =$$

$$= \frac{1}{\sqrt{2}} \begin{bmatrix} 0&0&1&0\\ 1&-1&0&0\\ 1&1&0&0\\ 0&0&1&-1 \end{bmatrix} \begin{bmatrix} 1\\0\\0\\0 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 0\\1\\1\\0 \end{bmatrix} = \frac{1}{\sqrt{2}} \left(|01\rangle + |10\rangle \right) = |\psi_+\rangle$$

Task 4

task 4

$$|00\rangle = |0\rangle \otimes |0\rangle \xrightarrow{X \otimes X} |1\rangle \otimes |1\rangle \xrightarrow{I \otimes H}$$

$$|1\rangle \otimes \frac{1}{\sqrt{2}} \left(|0\rangle - |1\rangle \right) = \frac{1}{\sqrt{2}} \left(|1\rangle \otimes |0\rangle - |1\rangle \otimes |1\rangle \right)$$

$$\xrightarrow{C_x} \frac{1}{\sqrt{2}} \left(|10\rangle - 01\rangle \right)$$