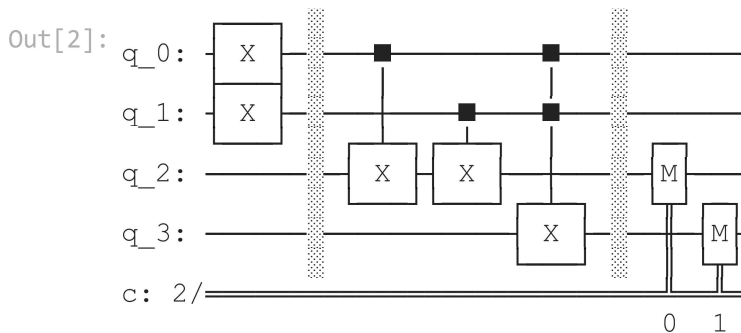


```
In [1]: import qiskit
        from qiskit import *
        from qiskit.visualization import plot_histogram
```

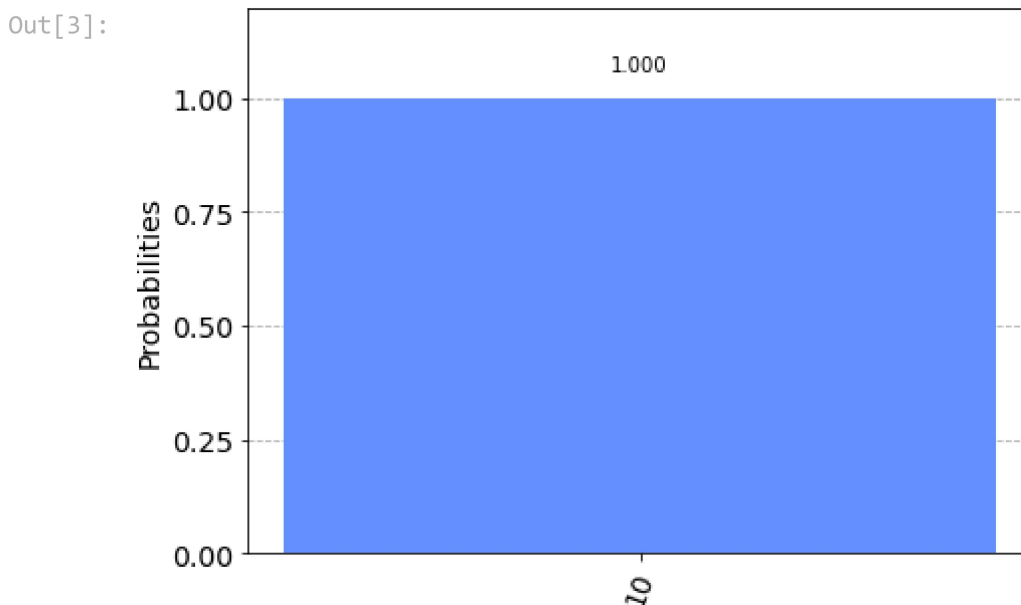
```
In [2]: # ADD 1+1

        qc = QuantumCircuit(4,2)
        qc.x(0)
        qc.x(1)
        qc.barrier()
        qc.cx(0,2)
        qc.cx(1,2)
        qc.ccx(0,1,3)
        qc.barrier()
        qc.measure(2,0)
        qc.measure(3,1)
        qc.draw()
```



```
In [3]: sim = Aer.get_backend('qasm_simulator')
        job = execute(qc, sim, shots = 1000)
        result = job.result()
        counts = result.get_counts(qc)
        plot_histogram(counts)

        # Output of 1+1 is 10
```



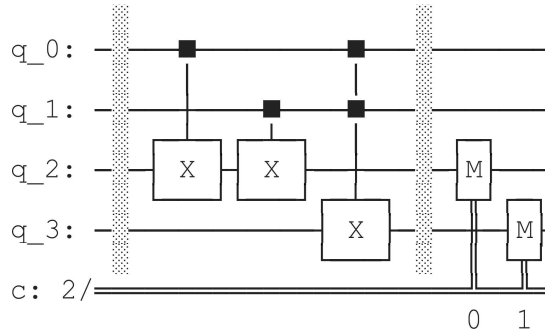
```
In [4]: # ADD 0+0
```

```

qc = QuantumCircuit(4,2)
qc.barrier()
qc.cx(0,2)
qc.cx(1,2)
qc.ccx(0,1,3)
qc.barrier()
qc.measure(2,0)
qc.measure(3,1)
qc.draw()

```

Out[4]:



In [5]:

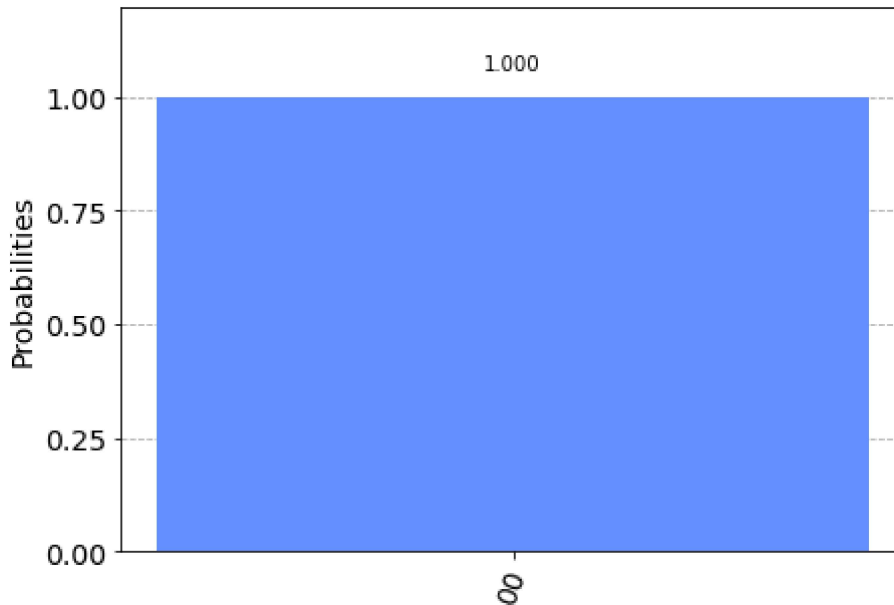
```

sim = Aer.get_backend('qasm_simulator')
job = execute(qc, sim, shots = 1000)
result = job.result()
counts = result.get_counts(qc)
plot_histogram(counts)

# Output of 0+0 is 00

```

Out[5]:



In [6]:

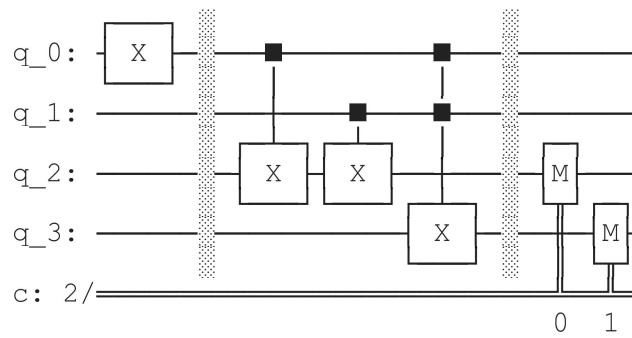
```

# ADD 1+0

qc = QuantumCircuit(4,2)
qc.x(0)
qc.barrier()
qc.cx(0,2)
qc.cx(1,2)
qc.ccx(0,1,3)
qc.barrier()
qc.measure(2,0)
qc.measure(3,1)
qc.draw()

```

Out[6]:

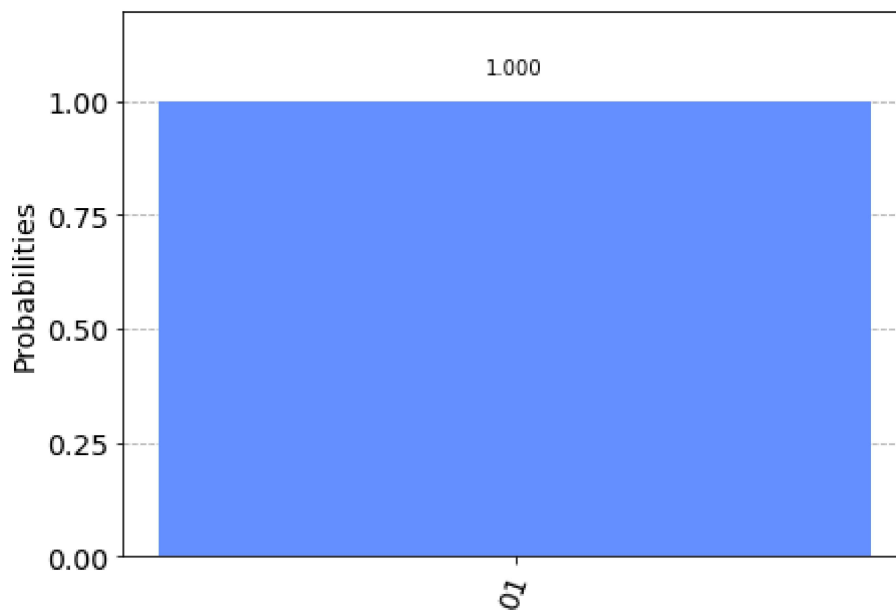


In [7]:

```
sim = Aer.get_backend('qasm_simulator')
job = execute(qc, sim, shots = 1000)
result = job.result()
counts = result.get_counts(qc)
plot_histogram(counts)

# Output of 1+0 is 01
```

Out[7]:

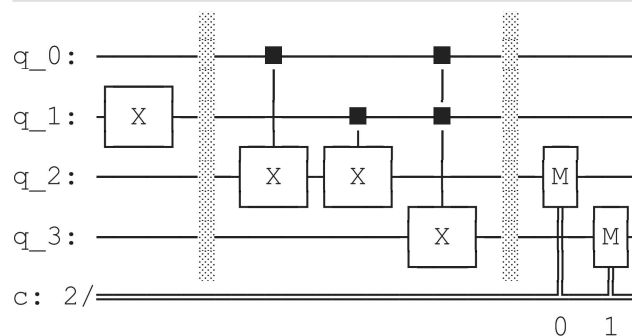


In [8]:

```
# ADD 0+1

qc = QuantumCircuit(4,2)
qc.x(1)
qc.barrier()
qc.cx(0,2)
qc.cx(1,2)
qc.ccx(0,1,3)
qc.barrier()
qc.measure(2,0)
qc.measure(3,1)
qc.draw()
```

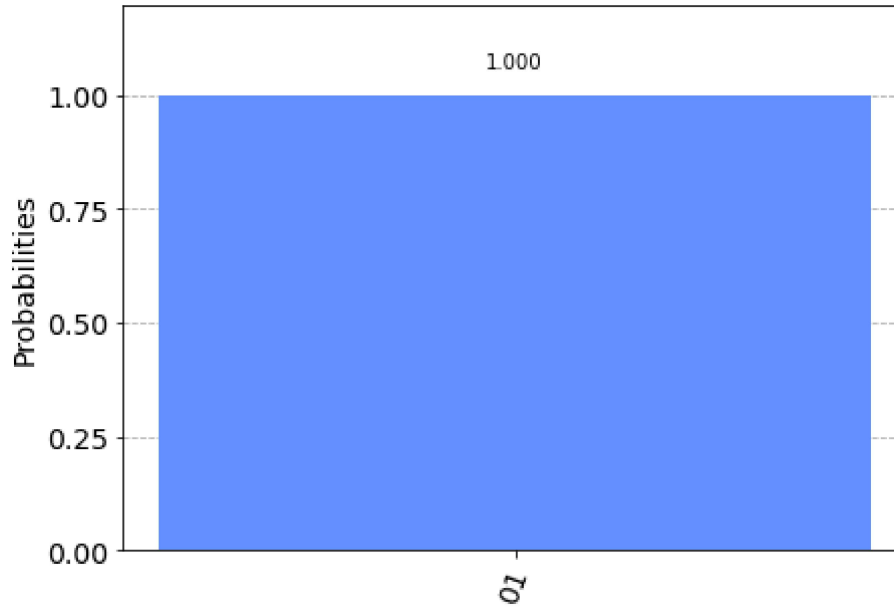
Out[8]:



```
In [9]: sim = Aer.get_backend('qasm_simulator')
job = execute(qc, sim, shots = 1000)
result = job.result()
counts = result.get_counts(qc)
plot_histogram(counts)

# Output of 0+1 is 01
```

Out[9]:



In []: