In [3]:
```
from qiskit import*
import matplotlib.pyplot as plt
from qiskit.visualization import plot_histogram
from qiskit.tools.monitor import job_monitor
```

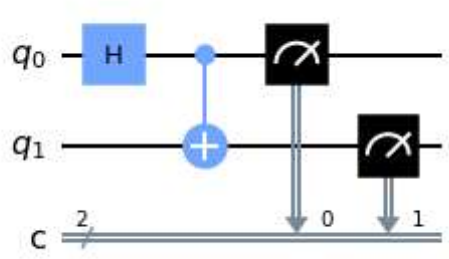
In [5]:
```
#Bell state for two qubit
bell = QuantumCircuit(2,2)
bell.h(0)
bell.cx(0,1)

bell.measure([0,1], [0,1])
```

Out[5]:
```
<qiskit.circuit.instructionset.InstructionSet at 0x7fe1c87de4f0>
```

In [6]:
```
bell.draw()
```

Out[6]:

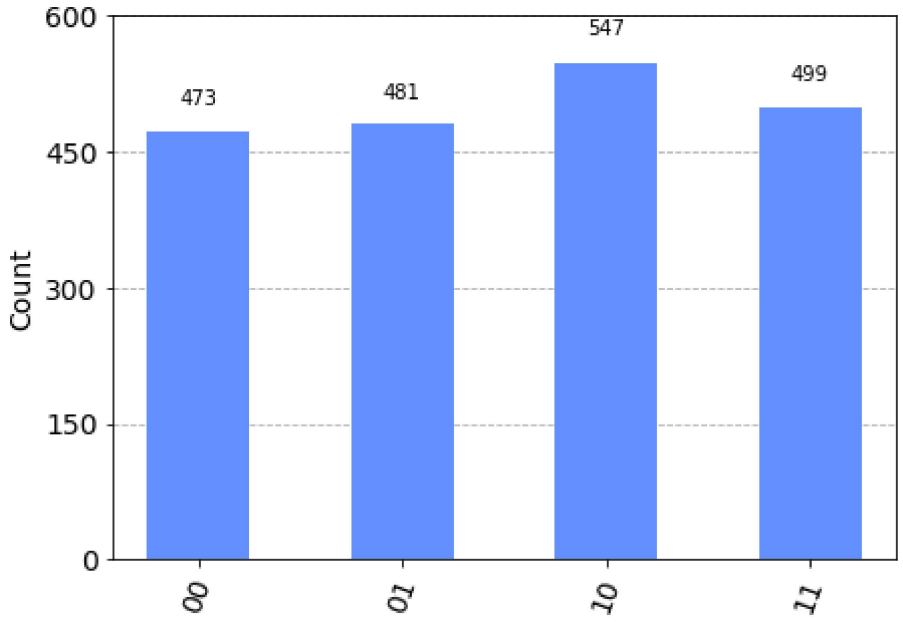


In [8]:
```
#run quantum circuit
backend = BasicAer.get_backend('qasm_simulator')  #will run on
circ = bell.compose(bell)
result = backend.run(transpile(circ, backend),shots=2000).result()
counts = result.get_counts(circ)
print(counts)
```

```
{'11': 499, '00': 473, '01': 481, '10': 547}
```

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

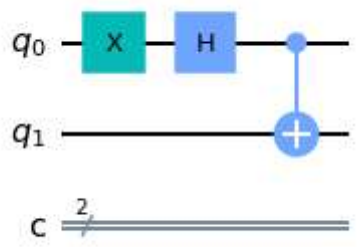
Out[9]:



In [11]:
```
#Bell state-2 for two qubits
bell = QuantumCircuit(2,2)
bell.x(0)
```

```
bell.h(0)
bell.cx(0,1)
bell.draw()
```

Out[11]:



In [12]:
```
bell.measure([0,1],[0,1])
```

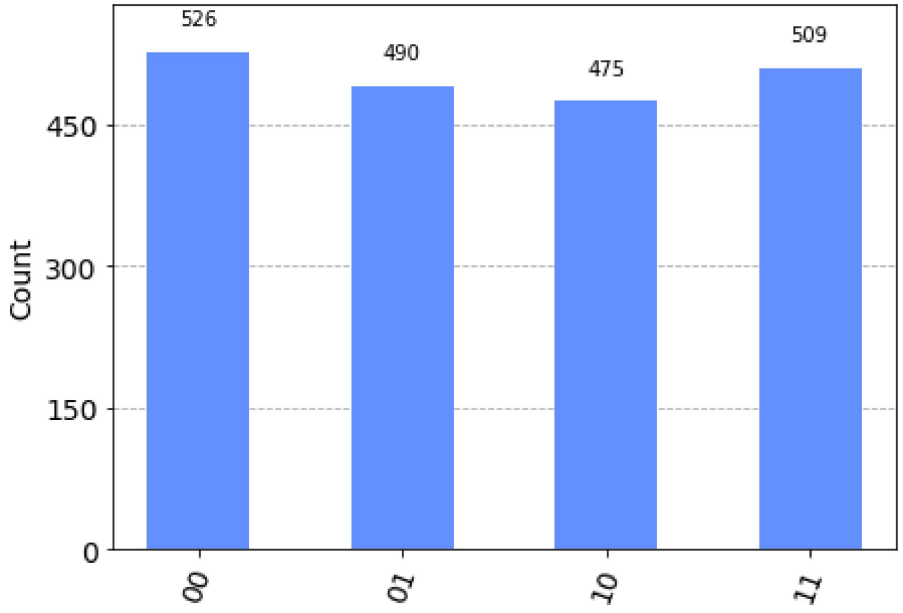
Out[12]:
```
<qiskit.circuit.instructionset.InstructionSet at 0x7fe12c9135e0>
```

In [13]:
```
#exzecute quantum circuit
backend = BasicAer.get_backend('qasm_simulator') #device will run on
circ = bell.compose(bell)
result = backend.run(transpile(circ,backend),shots=2000).result()
counts = result.get_counts(circ)
print(counts)
```

```
{'00': 526, '10': 475, '01': 490, '11': 509}
```

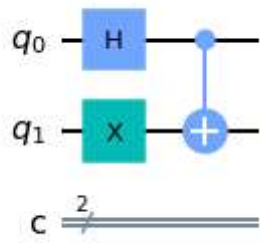
In [16]:
```
plot_histogram(counts)
```

Out[16]:



In [17]:
```
#Bell state-3 quantum circuit
bell = QuantumCircuit(2,2) #2 quantum register and 2 classical register
bell.x(1)
bell.h(0)
bell.cx(0,1)
bell.draw()
```

Out[17]:



In [18]:
```
bell.measure([0,1],[0,1])
```

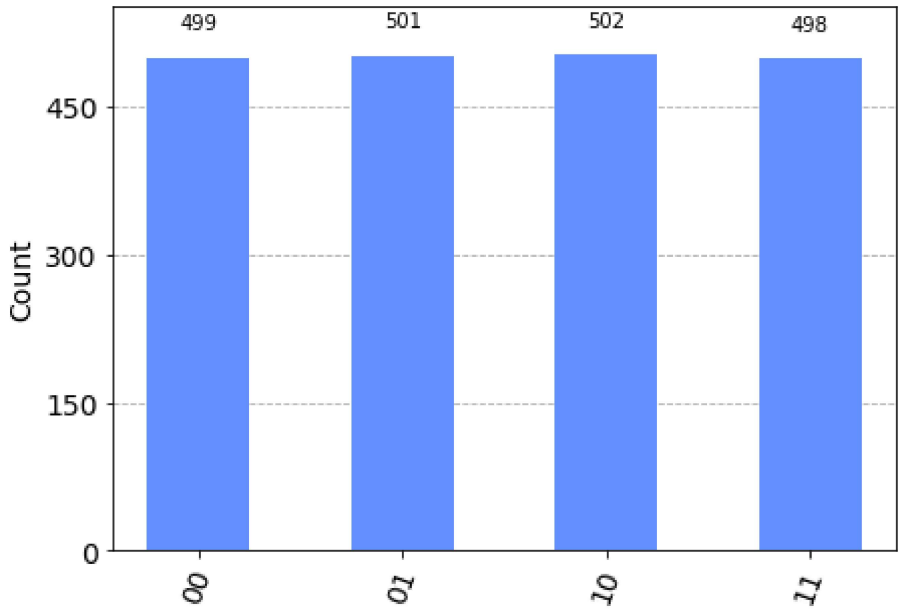
Out[18]:
```
<qiskit.circuit.instructionset.InstructionSet at 0x7fe12c8325e0>
```

In [19]:
```
#execute quantum circuit
backend = BasicAer.get_backend('qasm_simulator') #device will run on
circ = bell.compose(bell)
result = backend.run(transpile(circ,backend),shots=2000).result()
counts = result.get_counts(circ)
print(counts)
```

```
{'01': 501, '10': 502, '11': 498, '00': 499}
```

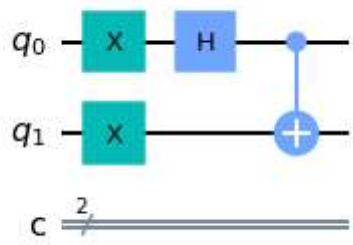
In [20]:
```
plot_histogram(counts)
```

Out[20]:



In [24]:
```
#bell state-4 quantum circuit
bell = QuantumCircuit(2,2) #2 quantum register,2 classical register
bell.x(0)
bell.x(1)
bell.h(0)
bell.cx(0,1)
bell.draw()
```

Out[24]:



In [25]:
```
bell.measure([0,1],[0,1])
```

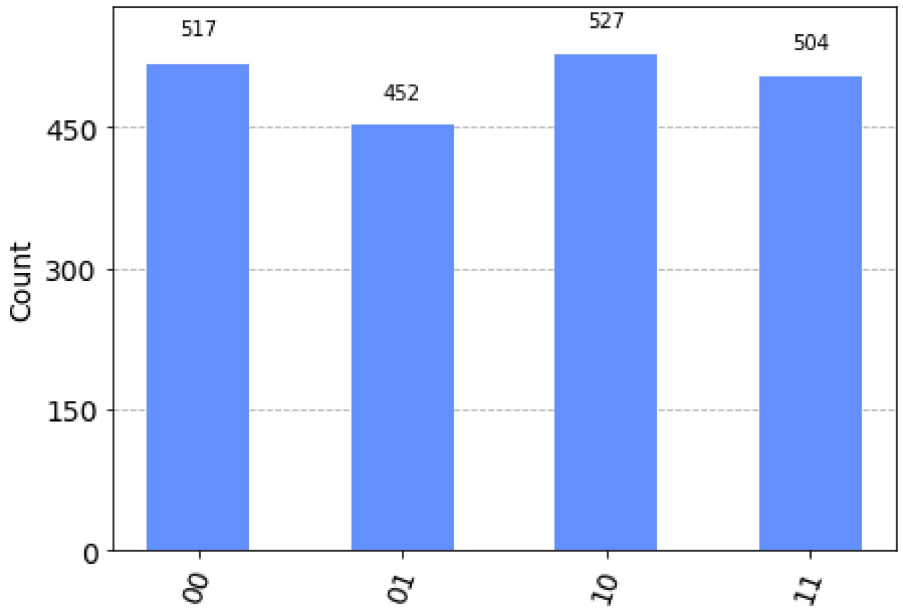
Out[25]:
```
<qiskit.circuit.instructionset.InstructionSet at 0x7fe12c66b1f0>
```

In [27]:
```
#execute quantum circuit
backend = BasicAer.get_backend('qasm_simulator') #device will run on
circ = bell.compose(bell)
result = backend.run(transpile(circ,backend),shots=2000).result()
counts = result.get_counts(circ)
print(counts)
```

```
{'10': 527, '11': 504, '00': 517, '01': 452}
```

In [28]:
```
plot_histogram(counts)
```

Out[28]:



In [ ]: