In [14]:
```
from qiskit import QuantumCircuit, Aer, assemble, execute
from math import pi, sqrt
from qiskit.visualization import plot_bloch_multivector, plot_histogram
import numpy as np
from qiskit_textbook.tools import array_to_latex
```

In [15]:
```
#2.2(3) Calculate the single qubit unitary (U) created by the sequence of gate
s: U = XZH. Use Qiskit's unitary simulator to check your results.
```

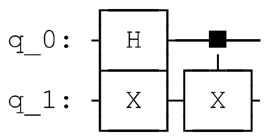
In [19]:
```
qc = QuantumCircuit(1)
qc.h(0)
qc.z(0)
qc.x(0)
display(qc.draw())
# Simulate the unitary
sim = Aer.get_backend('unitary_simulator')
result = execute(qc, sim).result()
unitary = result.get_unitary()
array_to_latex(unitary, pretext="\\text{Circuit = } ")
```

```
q_0: ─┤ H ├─┤ Z ├─┤ X ├─
```

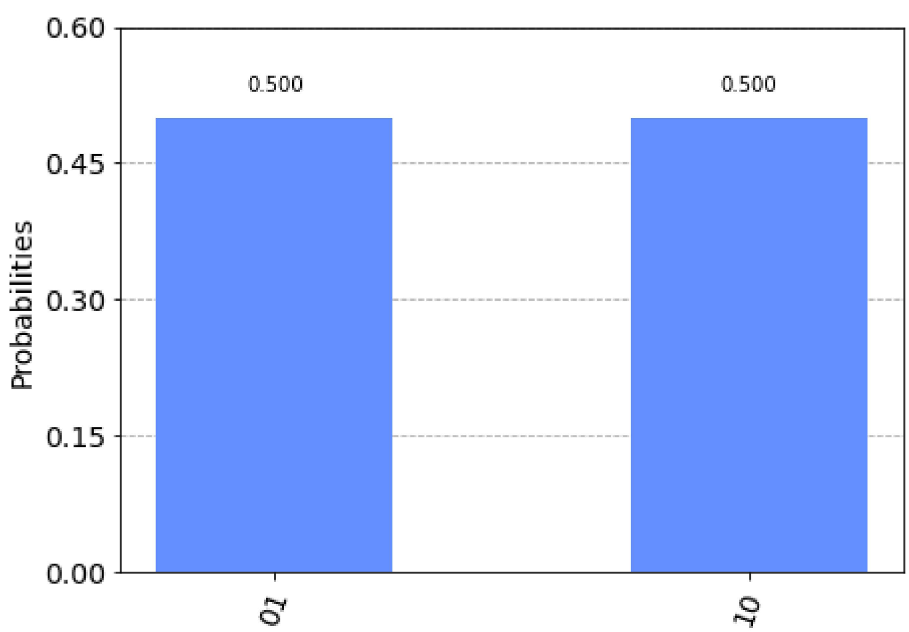
$$\text{Circuit} = \begin{bmatrix} -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}$$

In [18]:
```
#2.2(4) Create a quantum circuit that produces 1/sqrt(2) (|01> + |10>). Use th
e statevector simulator to verify your result.
```

In [20]:
```
circ = QuantumCircuit(2)
circ.h(0)
circ.x(1)
circ.cx(0,1)
display(circ.draw())
#Simulate the circuit
sim = Aer.get_backend('statevector_simulator')
result = execute(circ, sim).result()
counts = result.get_counts(circ)
plot_histogram(counts)
```



Out[20]:



In [21]:
```
#2.2(5) The circuit created above transform the state |00> to 1/sqrt(2) (|01>
 + |10>), calculate the unitary of this circuit using Qiskit's simulator.
```

In [22]:
```
sim = Aer.get_backend('unitary_simulator')
result = execute(circ, sim).result()
unitary = result.get_unitary()
array_to_latex(unitary, pretext="\\text{Circuit = } ")
```

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

In [ ]: