Qiskit Test 1

October 14, 2019

[1]:
```
from qiskit import *
```

[2]:
```
qr = QuantumRegister(2)
```

[3]:
```
cr = ClassicalRegister(2)
```

[4]:
```
circuit = QuantumCircuit(qr, cr)
```

[5]:
```
%matplotlib inline
```

[6]:
```
circuit.draw()
```

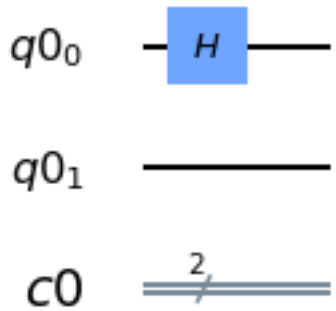
[6]:
```
<qiskit.visualization.text.TextDrawing at 0x7fae5e0b7240>
```

[7]:
```
circuit.h(qr[0])
```

[7]:
```
<qiskit.circuit.instructionset.InstructionSet at 0x7fae5e0bf3c8>
```

[8]:
```
circuit.draw(output='mpl')
```

[8]:

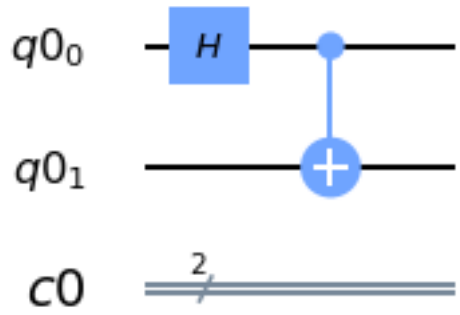


[9]:
```
circuit.cx(qr[0], qr[1])
```

[9]: <qiskit.circuit.instructionset.InstructionSet at 0x7fae53b7cf98>

[10]:
```
circuit.draw(output='mpl')
```

[10]:

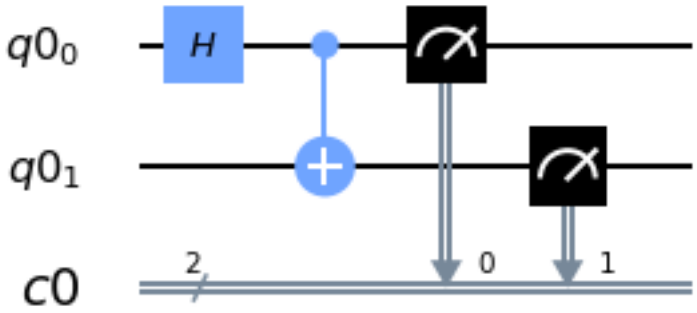


[11]:
```
circuit.measure(qr, cr)
```

[11]: <qiskit.circuit.instructionset.InstructionSet at 0x7fae53b06278>

[12]:
```
circuit.draw(output='mpl')
```

[12]:



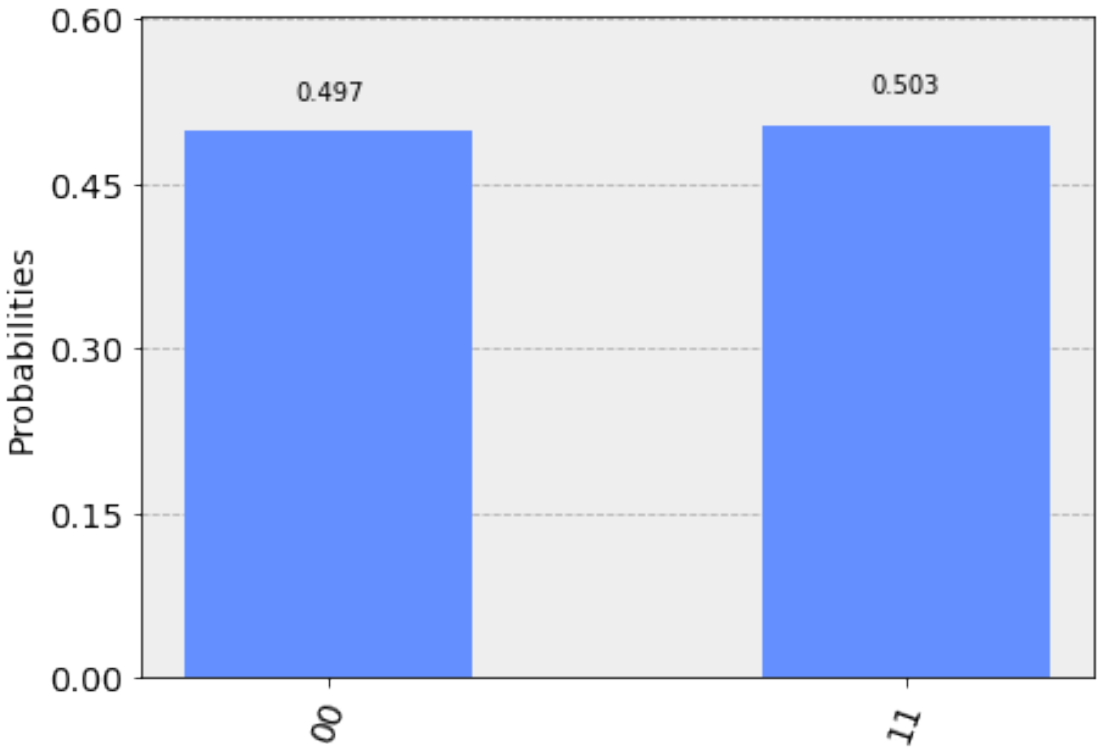
[13]:
```
sim = Aer.get_backend('qasm_simulator')
```

[15]:
```
result = execute(circuit, backend = sim).result()
```

[16]:
```
from qiskit.tools.visualization import plot_histogram
```

[17]:
```
plot_histogram(result.get_counts(circuit))
```

[17]:



[18]:
```
IBMQ.load_account()
```

[18]:
```
<AccountProvider for IBMQ(hub='ibm-q', group='open', project='main')>
```

[19]:
```
provider = IBMQ.get_provider('ibm-q')
```

[20]:
```
qcomp = provider.get_backend('ibmq_16_melbourne')
```

[21]:
```
job = execute(circuit, backend = qcomp)
```

[22]:
```
from qiskit.tools.monitor import job_monitor
```

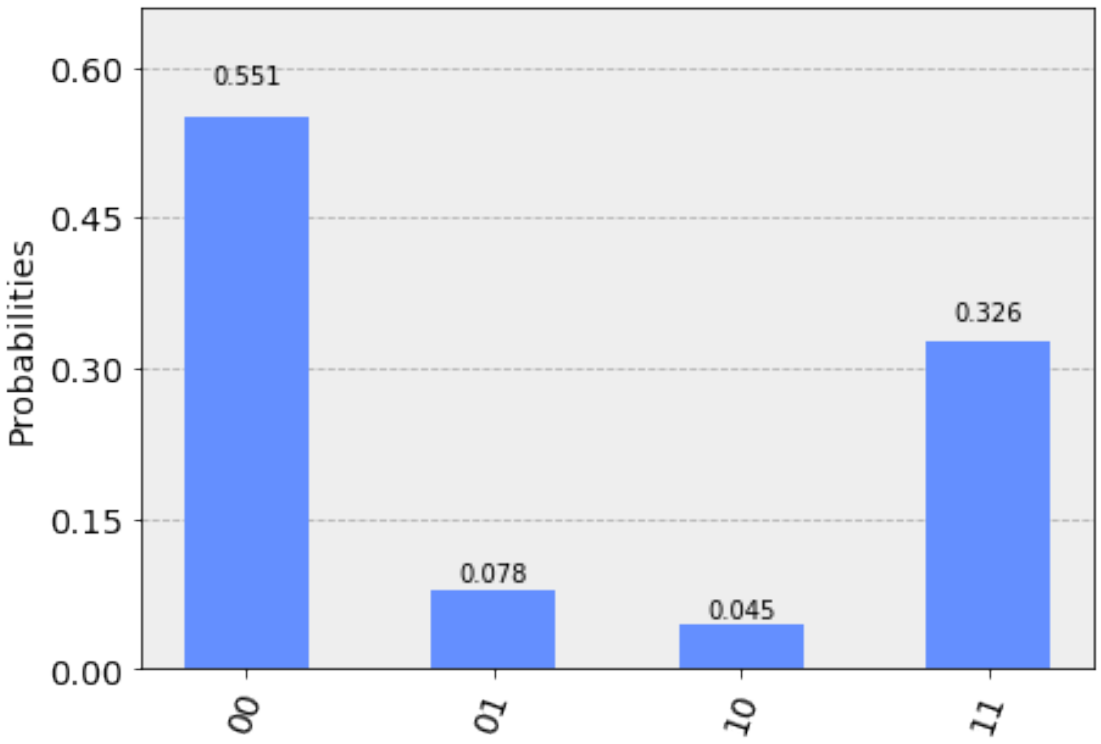
[23]:
```
job_monitor(job)
```

```
Job Status: job has successfully run
```

[24]:
```
result = job.result()
```

[25]:
```
plot_histogram(result.get_counts(circuit))
```

[25]:


0.60
0.45
0.30
0.15
0.00
Probabilities
0.551
0.078
0.045
0.326
00
01
10
11

[ ]: