HelloWorld_qiskit

December 20, 2020

1 Hello World Program for a Quantum Circuit

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

[29]:
```
quantReg = QuantumRegister(2)
```

[30]:
```
clReg = ClassicalRegister(2)
```

[31]:
```
circuit = QuantumCircuit(quantReg, clReg)
```

[32]:
```
circuit.draw()
```

[32]:
```
q7_0:

q7_1:

c1: 2/
```

I am not using *circuit.draw(output = 'mpl')* since I have not yet restarted the kernel with a fresh download of *pylatexenc*. For now I will have to make do with text based diagrams

[33]:
```
circuit.h(quantReg[0])
```

[33]:
```
<qiskit.circuit.instructionset.InstructionSet at 0x23d2c07dfa0>
```

[34]:
```
circuit.cx(quantReg[0], quantReg[1])
```

[34]:
```
<qiskit.circuit.instructionset.InstructionSet at 0x23d2c109ca0>
```

[35]:
```
circuit.draw()
```

[35]:
```
q7_0:  H

q7_1:    X
```

```
c1: 2/
```

[36]:
```
circuit.measure(quantReg, clReg)
```

[36]:
```
<qiskit.circuit.instructionset.InstructionSet at 0x23d2c09daf0>
```

[37]:
```
nativeSim = Aer.get_backend('qasm_simulator')
```

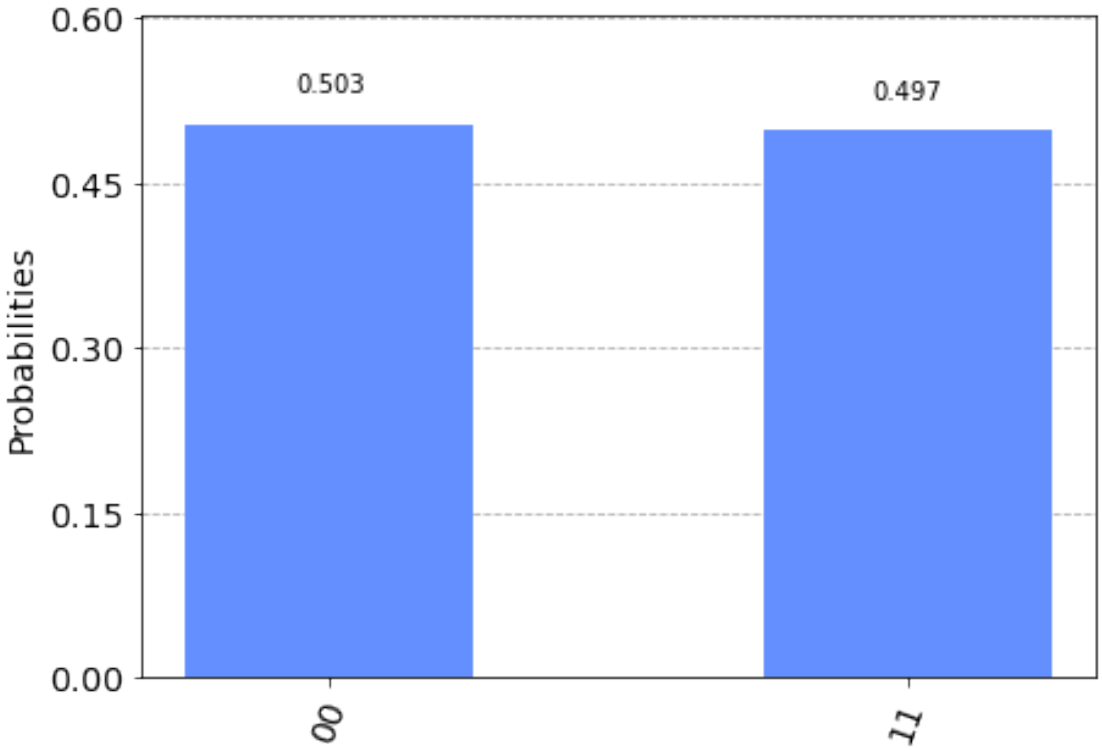
[38]:
```
result = execute(circuit, backend = nativeSim).result()
```

[39]:
```
from qiskit.tools.visualization import plot_histogram
```

2 Result on the Native 'Ideal' Quantum Simulator

[40]:
```
plot_histogram(result.get_counts(circuit))
```

[40]:



[41]:
```
IBMQ.load_account()
```

```
C:\Users\lenovo\anaconda3\lib\site-
packages\qiskit\providers\ibmq\ibmqfactory.py:192: UserWarning: Timestamps in
IBMQ backend properties, jobs, and job results are all now in local time instead
```

```
of UTC.
  warnings.warn('Timestamps in IBMQ backend properties, jobs, and job results '
```

[41]:
```
<AccountProvider for IBMQ(hub='ibm-q', group='open', project='main')>
```

[42]:
```
quantProvider = IBMQ.get_provider()
```

[44]:
```
quantumComputer = quantProvider.get_backend('ibmq_5_yorktown')
```

[47]:
```
job = execute(circuit, backend = quantumComputer)
```

3 Alternate State Vector Result

[50]:
```
from qiskit.tools.visualization import plot_bloch_multivector
```

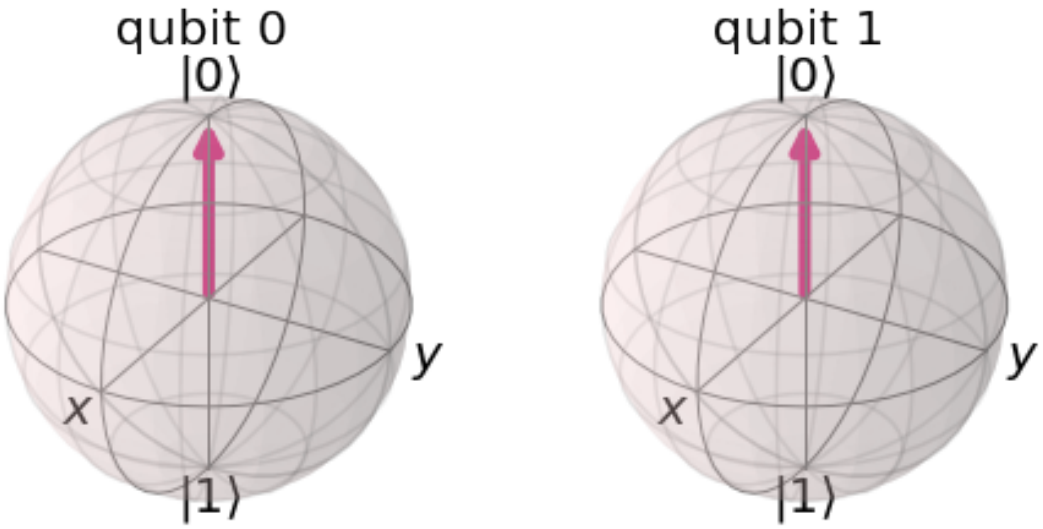
[53]:
```
svSim = Aer.get_backend('statevector_simulator')
```

[54]:
```
svResult = execute(circuit, backend = svSim).result()
```

[55]:
```
stateVector = svResult.get_statevector()
```

[56]:
```
plot_bloch_multivector(stateVector)
```

[56]:



4 Result from an IBMQ Quantum Computer

[45]:
```
from qiskit.tools.monitor import job_monitor
```

[48]:
```
job_monitor(job)
```

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

[49]:
```
plot_histogram(job.result().get_counts(circuit))
```

[49]: