Qsam simulator

July 4, 2021

[1]:
```
import numpy as np
# Importing standard Qiskit libraries
from qiskit import QuantumCircuit, transpile, Aer, IBMQ
from qiskit.tools.jupyter import *
from qiskit.visualization import *
from ibm_quantum_widgets import *

# Loading your IBM Quantum account(s)
provider = IBMQ.load_account()
```

[2]:
```
# Build
#------

# Create a Quantum Circuit acting on the q register
circuit = QuantumCircuit(2, 2)

# Add a H gate on qubit 0
circuit.h(0)

# Add a CX (CNOT) gate on control qubit 0 and target qubit 1
circuit.cx(0, 1)

# Map the quantum measurement to the classical bits
circuit.measure([0,1], [0,1])

# END
```

[2]:
```
<qiskit.circuit.instructionset.InstructionSet at 0x7faf280a4250>
```

[6]:
```
# Execute
#--------
from qiskit import execute
# Use Aer's qasm_simulator
simulator = Aer.get_backend('qasm_simulator')

# Execute the circuit on the qasm simulator
job = execute(circuit, simulator, shots=1000)
```

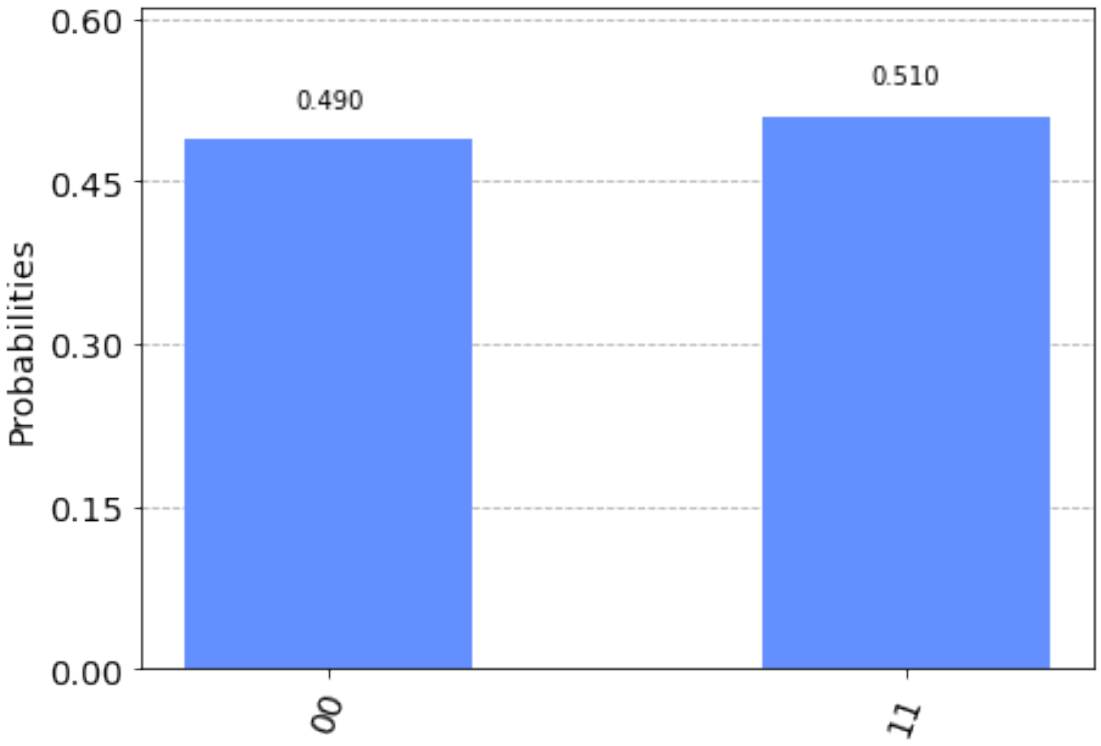
```
# Grab results from the job
result = job.result()

# Return counts
counts = result.get_counts(circuit)
print("\nTotal count for 00 and 11 are:",counts)

# END
```

```
Total count for 00 and 11 are: {'00': 490, '11': 510}
```

[8]:
```
# Visualize
#----------

# Import draw_circuit, then use it to draw the circuit
from ibm_quantum_widgets import draw_circuit
draw_circuit(circuit)

# Analyze
#--------

# Plot a histogram
plot_histogram(counts)

# END
```

```
CircuitComposer(circuit=<qiskit.circuit.quantumcircuit.QuantumCircuit object at 0x7fae9c87ce20
```

[8]:



[ ]:
```
#Program executed by Bhadale IT in IBM Quantum Lab ( https://www.bhadaleit.com)
```