


Examples of Qubit Ordering with Qiskit

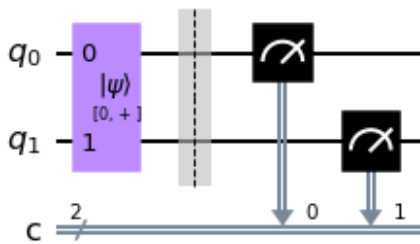
Claudio Fontana
University of Palermo
Palermo, Italy
claudio.fontana@unipa.it 

Initialize two quantum registers to $|0+\rangle$

Example #1:

```
qc = QuantumCircuit(2,2)
#initialize to  $|0+\rangle$ 
qc.initialize('0+',qc.qubits)
qc.barrier()
qc.measure(0,0)
qc.measure(1,1)
print("Circuit")
qc.draw(output="mpl")
```

Circuit

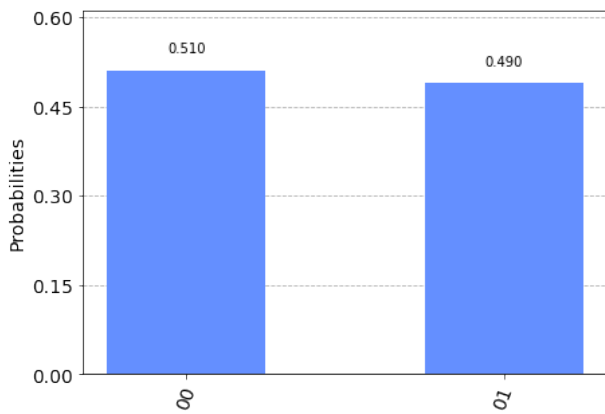


```
backend = BasicAer.get_backend("qasm_simulator")
shots=100;
job = execute(qc, backend, shots=shots)
counts=job.result().get_counts()
print("Counting measurement over",shots,"shots")
print(counts)
print("Histogram")
plot_histogram(counts)
```

Counting measurement over 100 shots

```
{'00': 51, '01': 49}
```

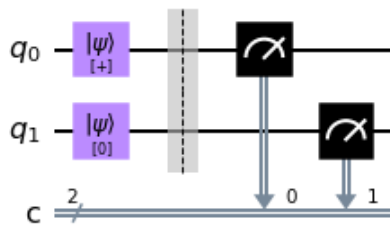
Histogram



Example #2:

```
qc = QuantumCircuit(2,2)
#initialize to  $|0+\rangle$ 
qc.initialize('+',0) # according to the little-endian convention on qubit ordering
qc.initialize('0',1) # according to the little-endian convention on qubit ordering
qc.barrier()
qc.measure(0,0)
qc.measure(1,1)
print("Circuit")
qc.draw(output="mpl")
```

Circuit

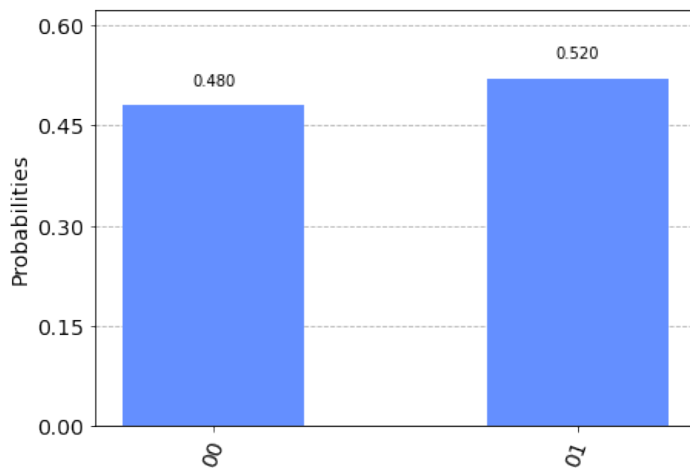


```
backend = BasicAer.get_backend("qasm_simulator")
shots=100;
job = execute(qc, backend, shots=shots)
counts=job.result().get_counts()
print("Counting measurement over",shots,"shots")
print(counts)
print("Histogram")
plot_histogram(counts)
```

Counting measurement over 100 shots

```
{'00': 48, '01': 52}
```

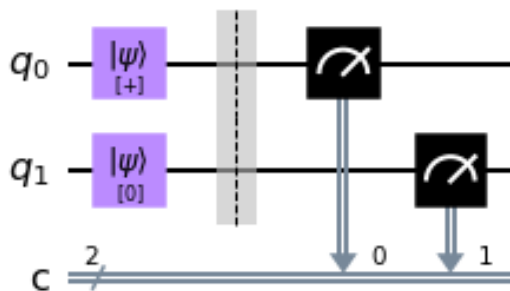
Histogram



Example #3:

```
qc = QuantumCircuit(2,2)
string='0+'
[qc.initialize(val,pos) for pos,val in enumerate(string[::-1])] #according to the little-endian convention on qubit ordering
qc.barrier()
qc.measure(0,0)
qc.measure(1,1)
print("Circuit")
qc.draw(output="mpl")
```

Circuit



```
from qiskit.visualization import plot_histogram
# Get a backend to run on
backend = BasicAer.get_backend("qasm_simulator")
# Execute the circuit on the backend and get the measurement results
shots=100;
job = execute(qc, backend, shots=shots)
result = job.result()

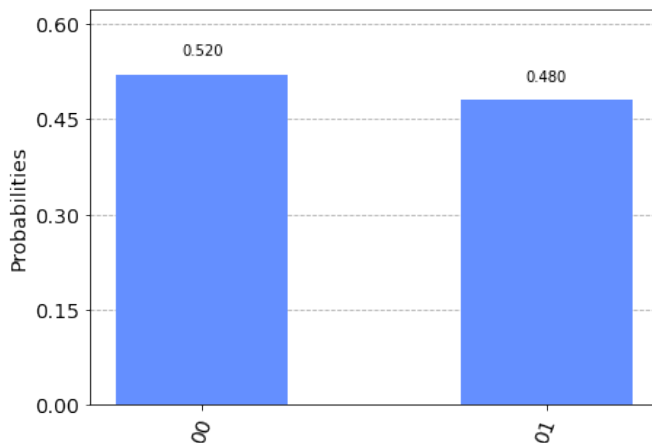
# Print the measurement results
counts=result.get_counts()
print("Counting measurement over",shots,"shots")
print(counts)
```

```
#Plot Histogram
print("Histogram")
plot_histogram(counts)
```

Counting measurement over 100 shots

```
{'00': 52, '01': 48}
```

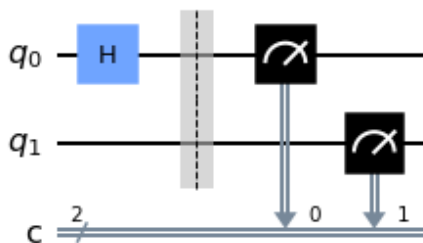
Histogram



Example #4:

```
qc = QuantumCircuit(2,2)
#initialize to |0+>
qc.h(0) # according to the little-endian convention on qubit ordering
qc.barrier()
qc.measure(0,0)
qc.measure(1,1)
print("Circuit")
qc.draw(output="mpl")
```

Circuit



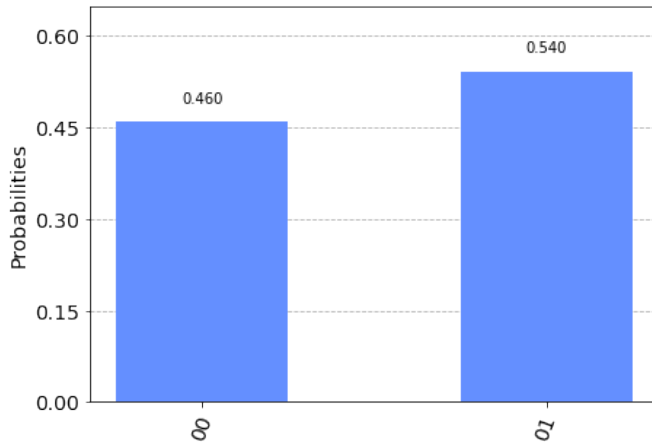
```
from qiskit.visualization import plot_histogram
backend = BasicAer.get_backend("qasm_simulator")
shots=100;
job = execute(qc, backend, shots=shots)
counts=job.result().get_counts()
print("Counting measurement over",shots,"shots")
print(counts)
```

```
print("Histogram")
plot_histogram(counts)
```

Counting measurement over 100 shots

```
{'00': 46, '01': 54}
```

Histogram



Example #5. Initialize two quantum registers to $|01\rangle$:

```
qc = QuantumCircuit(2,2)
bits=[0,1] #i.e. |01>
qc.x([pos for pos,val in enumerate(bits[:-1]) if val==1]) #according to the little-endian convention on qubit ordering
qc.barrier()
qc.measure(0,0)
qc.measure(1,1)
backend = BasicAer.get_backend("qasm_simulator")
job = execute(qc, backend=backend, shots=1, memory=True)
memArray=job.result().get_memory()
print("results:",memArray)
print("Circuit")
qc.draw(output="mpl")
```

```
results: ['01']
```

Circuit

