In [1]:
```
import qiskit as qk
from numpy import pi as pi
from qiskit.tools.monitor import job_monitor
```

```
C:\ProgramData\Anaconda3\lib\site-packages\marshmallow\schema.py:3
64: ChangedInMarshmallow3Warning: strict=False is not recommended.
In marshmallow 3.0, schemas will always be strict. See https://mar
shmallow.readthedocs.io/en/latest/upgrading.html#schemas-are-alway
s-strict
  ChangedInMarshmallow3Warning
```

In [2]:
```
qk.IBMQ.enable_account('10a18d47e83ae9af5bf42c51c6899da81e1f23d5349
a6233bcab1eb72f141e13d391a77f79970dad2777f1b52a6a8559b90ad33005ddbe
5ac433bb7819e35d1b')
```

In [3]:
```
q = qk.QuantumRegister(5, 'q')
c = qk.ClassicalRegister(5, 'c')
qft_circ = qk.QuantumCircuit(q, c)
```

In [4]:
```
def initialize_circuit(init='00000'):
    init_circ = qk.QuantumCircuit(q, c)
    for idx, value in enumerate(init):
        if int(value):
            init_circ.x(q[idx])
    return init_circ
```

In [5]:
```
init_circ = initialize_circuit('10110')
```

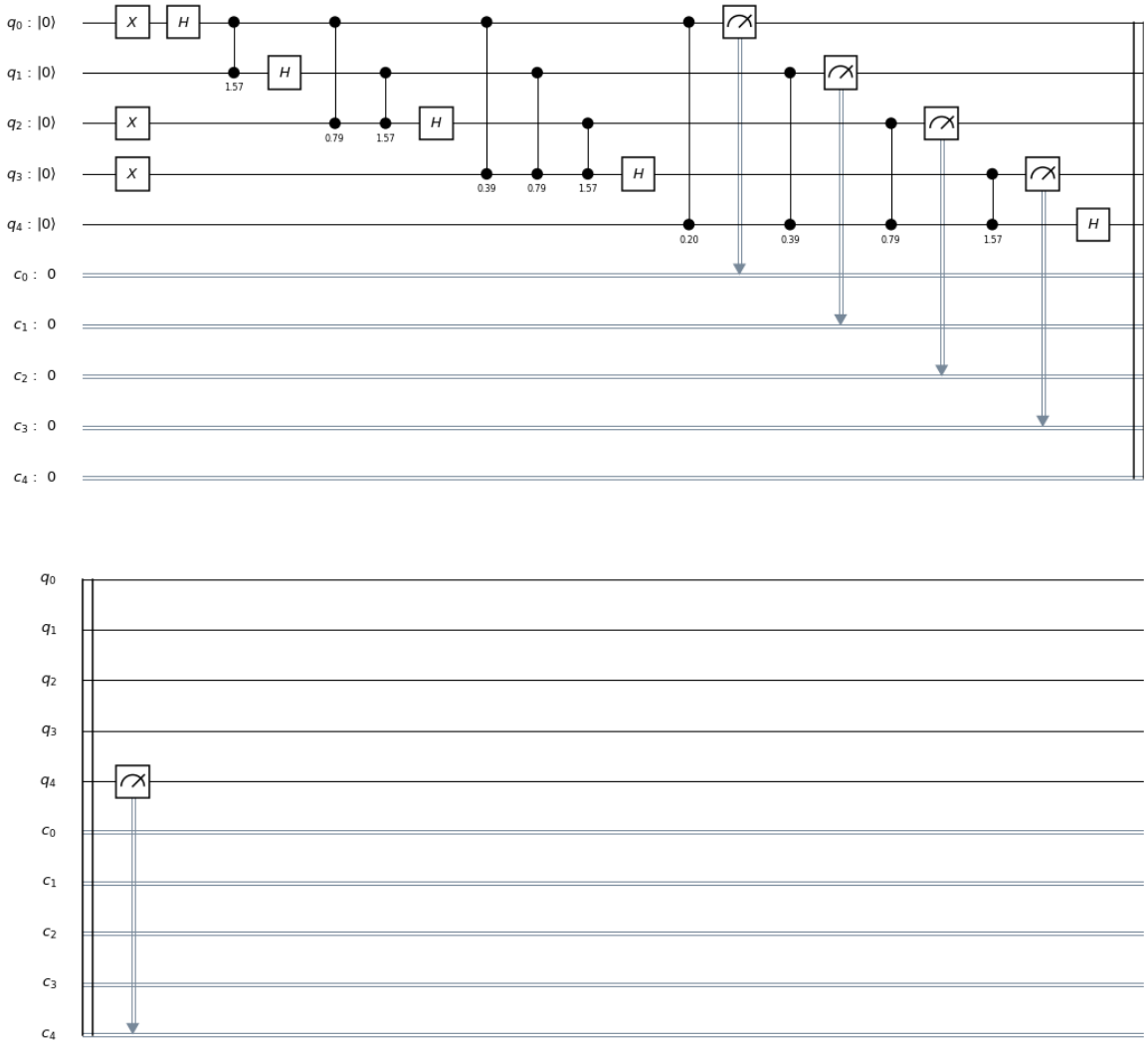
In [6]:
```
for qubit in range(5):
    qft_circ.h(q[qubit])
    for n in range(1, 5-qubit):
        qft_circ.cu1(pi/(2**n), q[qubit+n], q[qubit])
qft_circ.measure(q, c)
```

Out[6]:
```
<qiskit.circuit.instructionset.InstructionSet at 0x24699243240>
```

In [7]:
```
qft_circ = init_circ + qft_circ
```

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

Out[8]:



In [9]:
```
qk.execute(qft_circ, qk.BasicAer.get_backend('qasm_simulator'), sho
ts=1024).result().get_counts()
```

```
C:\ProgramData\Anaconda3\lib\site-packages\marshmallow\schema.py:3
64: ChangedInMarshmallow3Warning: strict=False is not recommended.
In marshmallow 3.0, schemas will always be strict. See https://mar
shmallow.readthedocs.io/en/latest/upgrading.html#schemas-are-alway
s-strict
  ChangedInMarshmallow3Warning
```

Out[9]:
```
{'00000': 42,
 '00001': 29,
 '00010': 28,
 '00011': 36,
 '00100': 46,
 '00101': 28,
 '00110': 28,
 '00111': 39,
 '01000': 38,
 '01001': 30,
 '01010': 36,
 '01011': 30,
 '01100': 26,
 '01101': 38,
 '01110': 28,
 '01111': 31,
 '10000': 32,
 '10001': 26,
 '10010': 33,
 '10011': 34,
 '10100': 32,
 '10101': 21,
 '10110': 29,
 '10111': 32,
 '11000': 37,
 '11001': 33,
 '11010': 34,
 '11011': 24,
 '11100': 23,
 '11101': 29,
 '11110': 36,
 '11111': 36}
```

In [10]:
```
real_backend = qk.providers.ibmq.least_busy(qk.IBMQ.backends(simula
tor=False, operational=True))
```

In [11]:
```
real_backend.name()
```

Out[11]:
```
'ibmq_16_melbourne'
```

In [ ]:
```
job = qk.execute(qft_circ, backend=real_backend, shots=1024)
```

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

In [ ]:
```
job.result().get_counts()
```

In [12]:
```
builtin_qft = initialize_circuit('10110')
```

In [13]:
```
# built-in
from qiskit.tools.qi.qi import qft
qft(builtin_qft, q, 5)
```

In [14]:
```
builtin_qft.draw(output='mpl')
```

Out[14]: