

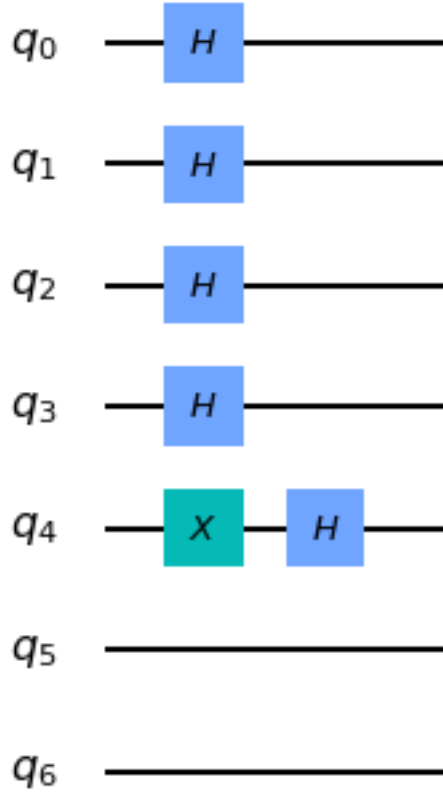
deutsch-josza

August 29, 2020

```
[73]: def dj_oracle(b_str):  
    oracle_qc = QuantumCircuit(len(b_str)+1)  
    for qubit in range(len(b_str)):  
        if b_str[qubit] == '1':  
            oracle_qc.x(qubit)  
  
    for qubit in range(len(b_str)):  
        oracle_qc.cx(qubit, len(b_str))  
  
    for qubit in range(len(b_str)):  
        if b_str[qubit] == '1':  
            oracle_qc.x(qubit)  
  
    return oracle_qc
```

```
[93]: alice_start = QuantumCircuit(4+1+2, name="test")  
alice_start.x(4)  
alice_start.h(4)  
for qubit in range(4):  
    alice_start.h(qubit)  
# oracle_alice = dj_oracle('0101')  
# alice_start.append(tto.circuit, range(4+1+2))  
alice_start.draw(output='mpl')
```

[93]:



```
[60]: alice_bob_tele_qr = QuantumRegister(4+1+4+4+1)
      c11,c12,c21,c22,c31,c32,c41,c42,c_m = ClassicalRegister(1),
      ↪ ClassicalRegister(1), ClassicalRegister(1),
      ↪ ClassicalRegister(1), ClassicalRegister(1), ClassicalRegister(1),
      ↪ ClassicalRegister(1), ClassicalRegister(4)
      alice_bob_tele =
      ↪ QuantumCircuit(alice_bob_tele_qr,c11,c12,c21,c22,c31,c32,c41,c42,c_m)
      alice_bob_tele.draw()
```

[60]:

$q3_0$ —

$q3_1$ —

$q3_2$ —

$q3_3$ —

$q3_4$ —

$q3_5$ —

$q3_6$ —

$q3_7$ —

$q3_8$ —

$q3_9$ —

$q3_{10}$ —

$q3_{11}$ —

$q3_{12}$ —

$q3_{13}$ —

$c0$ ==

$c1$ ==

$c2$ ==

$c3$ ==

$c4$ ==

$c5$ ==

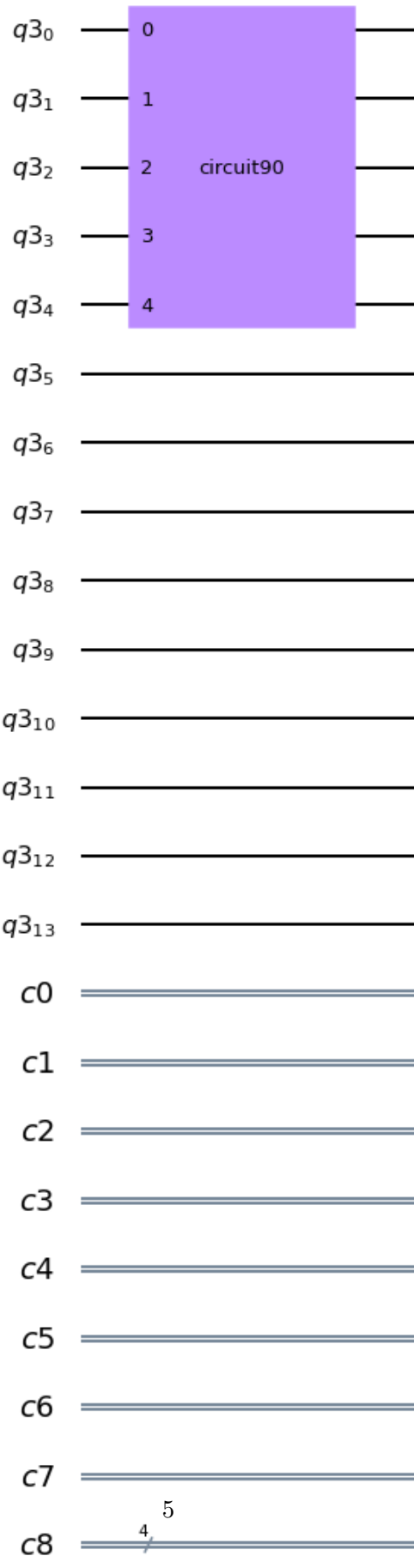
$c6$ ==

$c7$ ==

$c8$ ==⁴

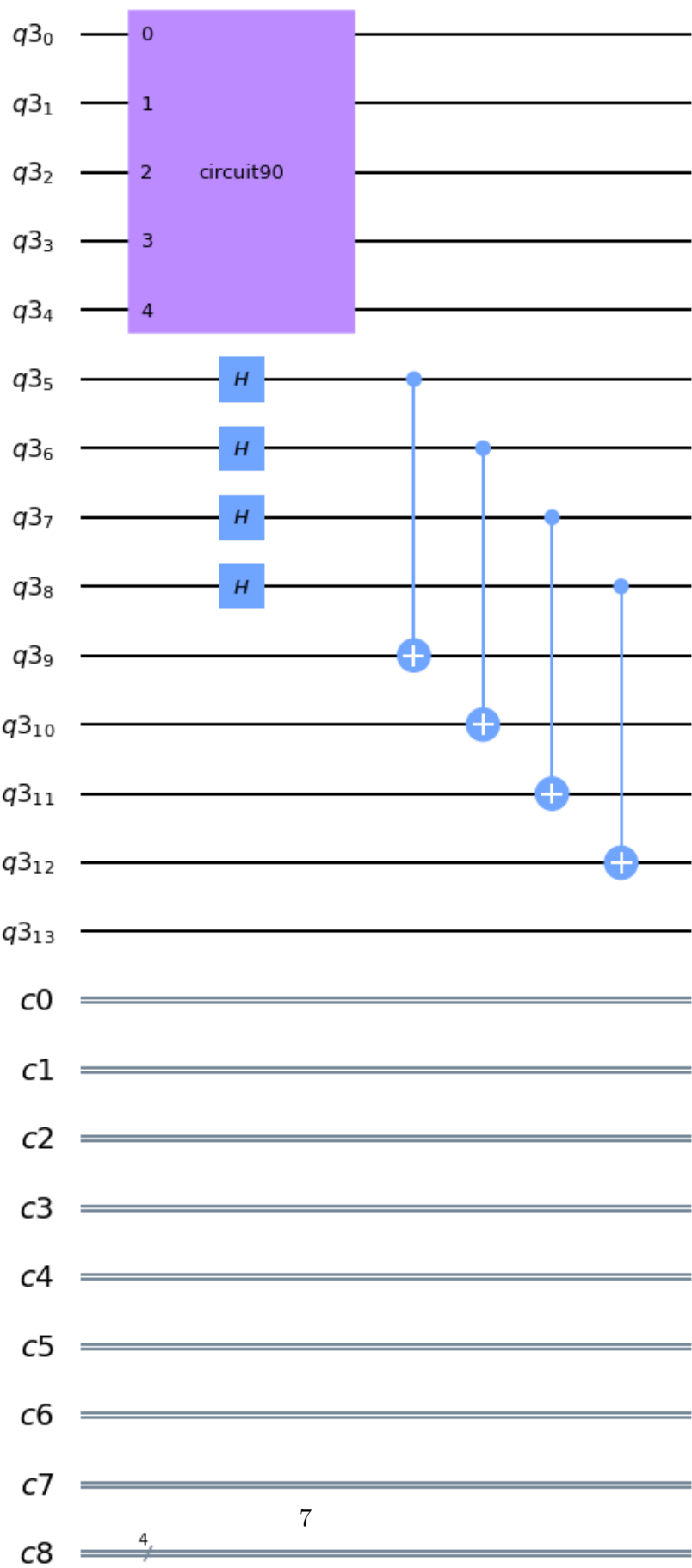
```
[61]: alice_bob_tele.append(alice_start,range(4+1))  
      alice_bob_tele.draw()
```

[61]:



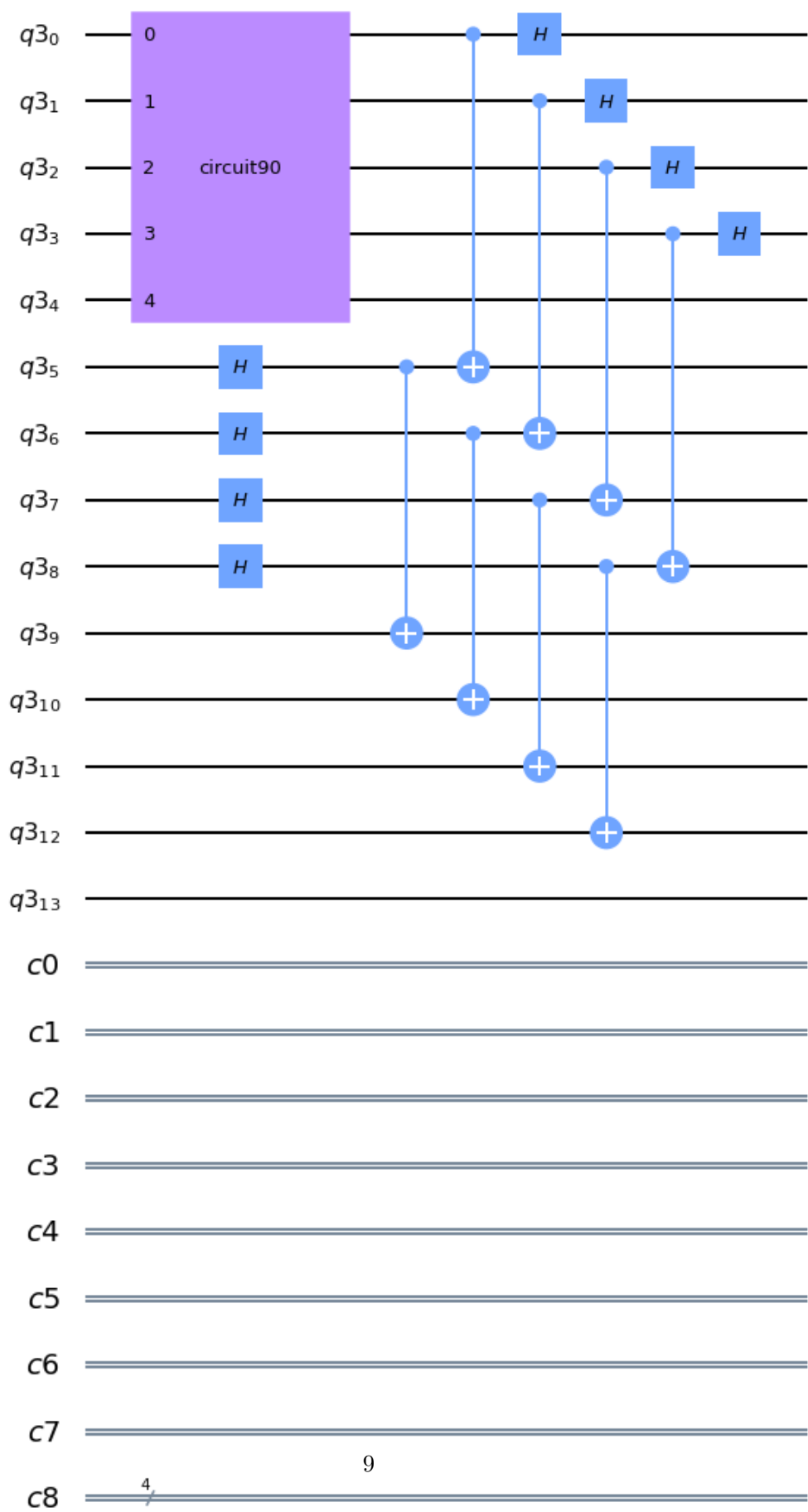
```
[62]: def create_bell_pair(qc, a, b):  
      qc.h(a) # Put qubit a into state |+>  
      qc.cx(a,b) # CNOT with a as control and b as target  
      create_bell_pair(alice_bob_tele, 5, 9)  
      create_bell_pair(alice_bob_tele, 6, 10)  
      create_bell_pair(alice_bob_tele, 7, 11)  
      create_bell_pair(alice_bob_tele, 8, 12)  
      alice_bob_tele.draw()
```

[62]:



```
[63]: def alice_gates(qc, psi, a):  
      qc.cx(psi, a)  
      qc.h(psi)  
      alice_gates(alice_bob_tele, 0, 5)  
      alice_gates(alice_bob_tele, 1, 6)  
      alice_gates(alice_bob_tele, 2, 7)  
      alice_gates(alice_bob_tele, 3, 8)  
      alice_bob_tele.draw()
```

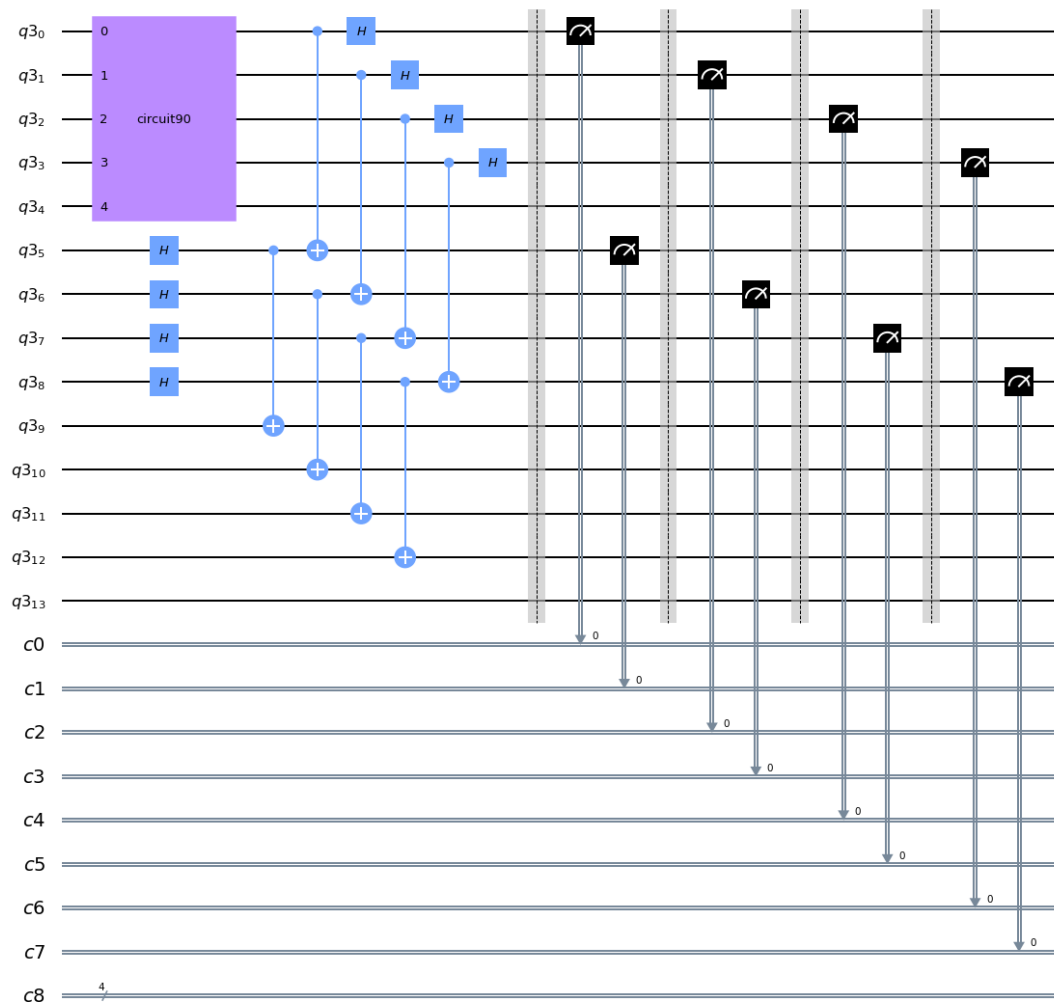
[63]:



```
[64]: def measure_and_send(qc, psi, a, c1, c2):
    qc.barrier()
    qc.measure(psi,c1)
    qc.measure(a,c2)
measure_and_send(alice_bob_tele,0,5,0,1)
measure_and_send(alice_bob_tele,1,6,2,3)
measure_and_send(alice_bob_tele,2,7,4,5)
measure_and_send(alice_bob_tele,3,8,6,7)
```

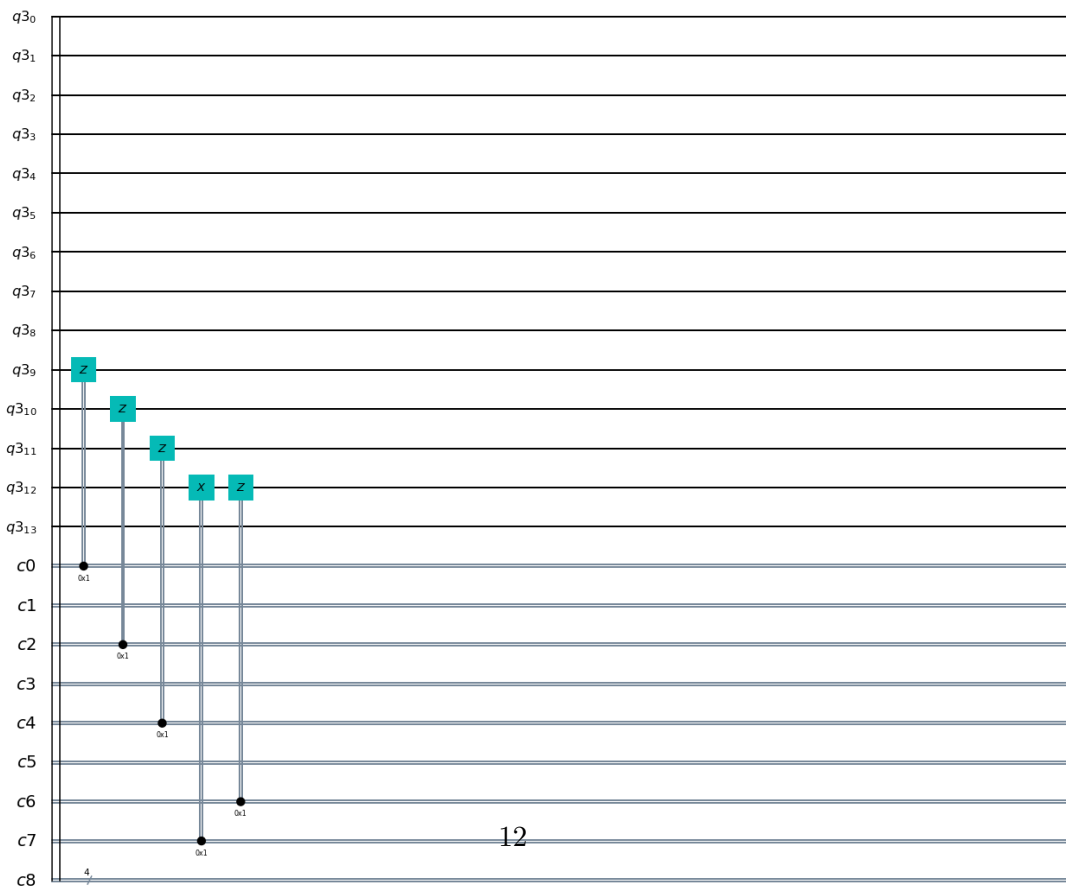
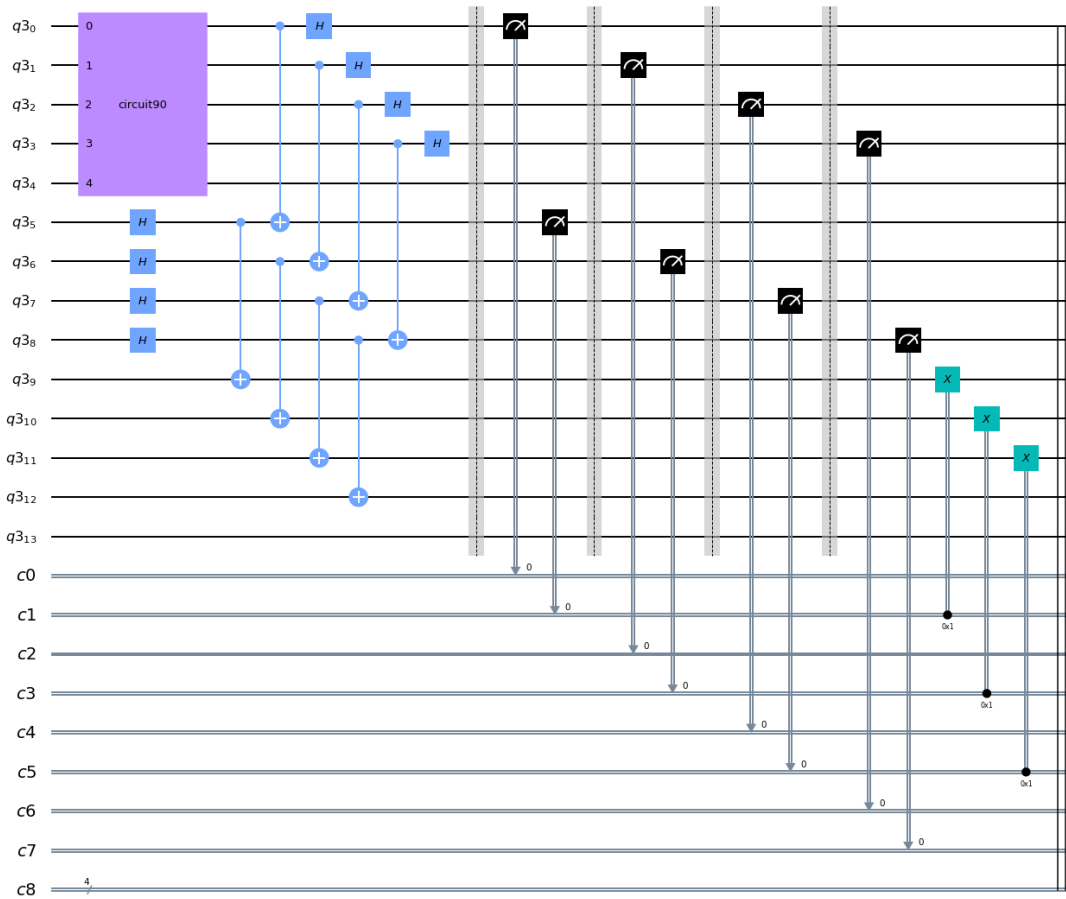
```
[65]: alice_bob_tele.draw()
```

[65]:



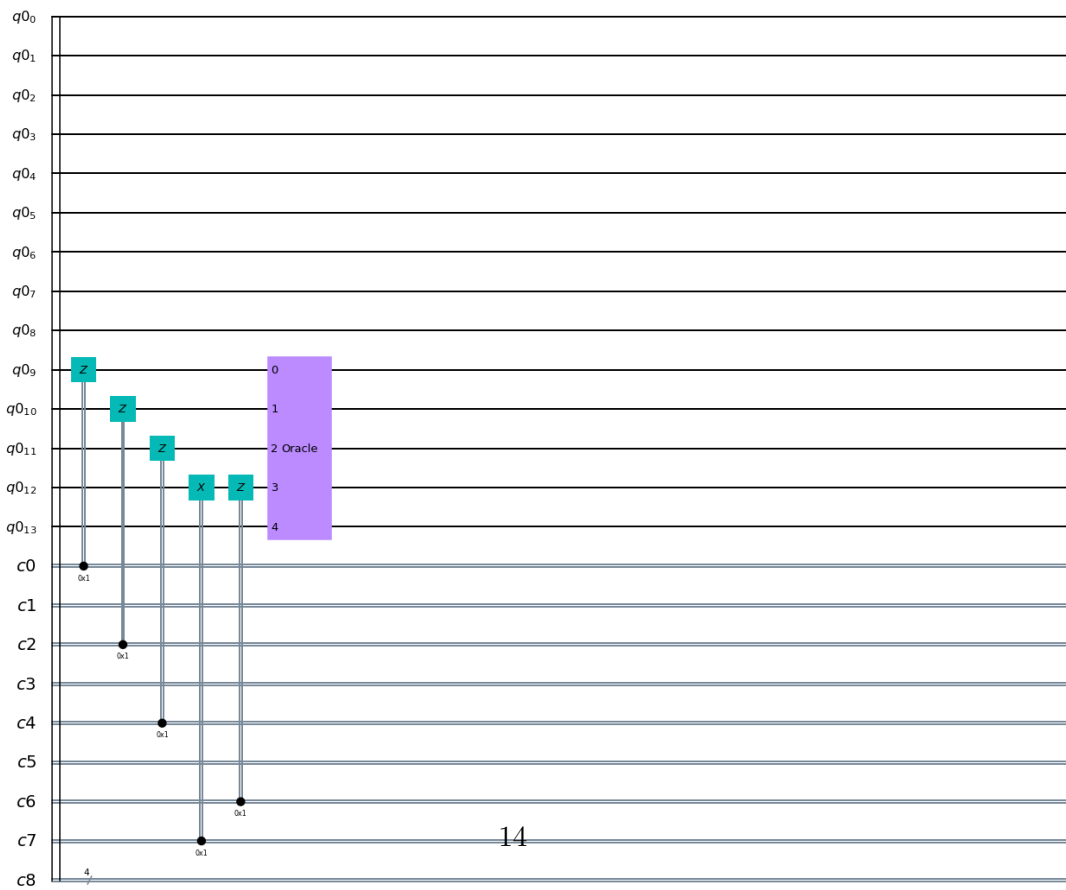
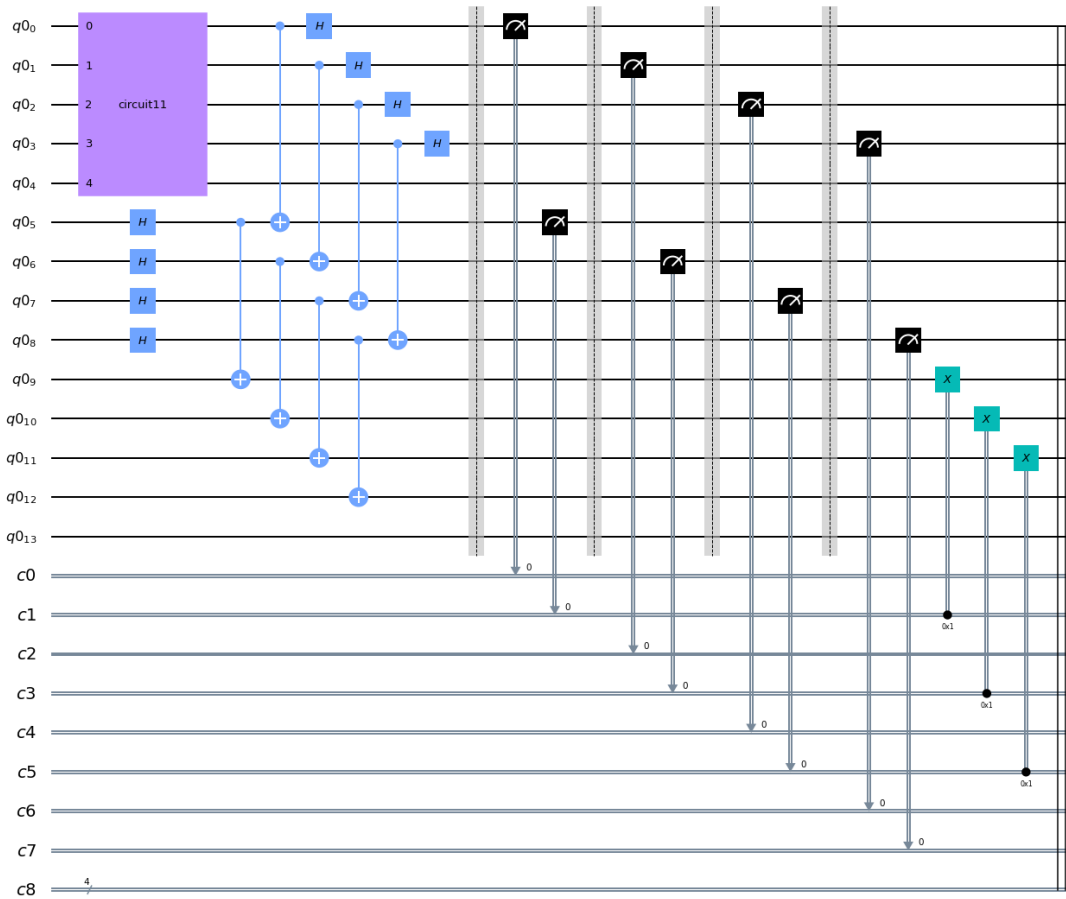
```
[66]: def bob_gates(qc, qubit, crz, crx):  
      qc.x(qubit).c_if(crx, 1) # Apply gates if the registers  
      qc.z(qubit).c_if(crz, 1) # are in the state '1'  
      bob_gates(alice_bob_tele, 9, c11, c12)  
      bob_gates(alice_bob_tele, 10, c21, c22)  
      bob_gates(alice_bob_tele, 11, c31, c32)  
      bob_gates(alice_bob_tele, 12, c41, c42)  
      alice_bob_tele.draw()
```

[66]:



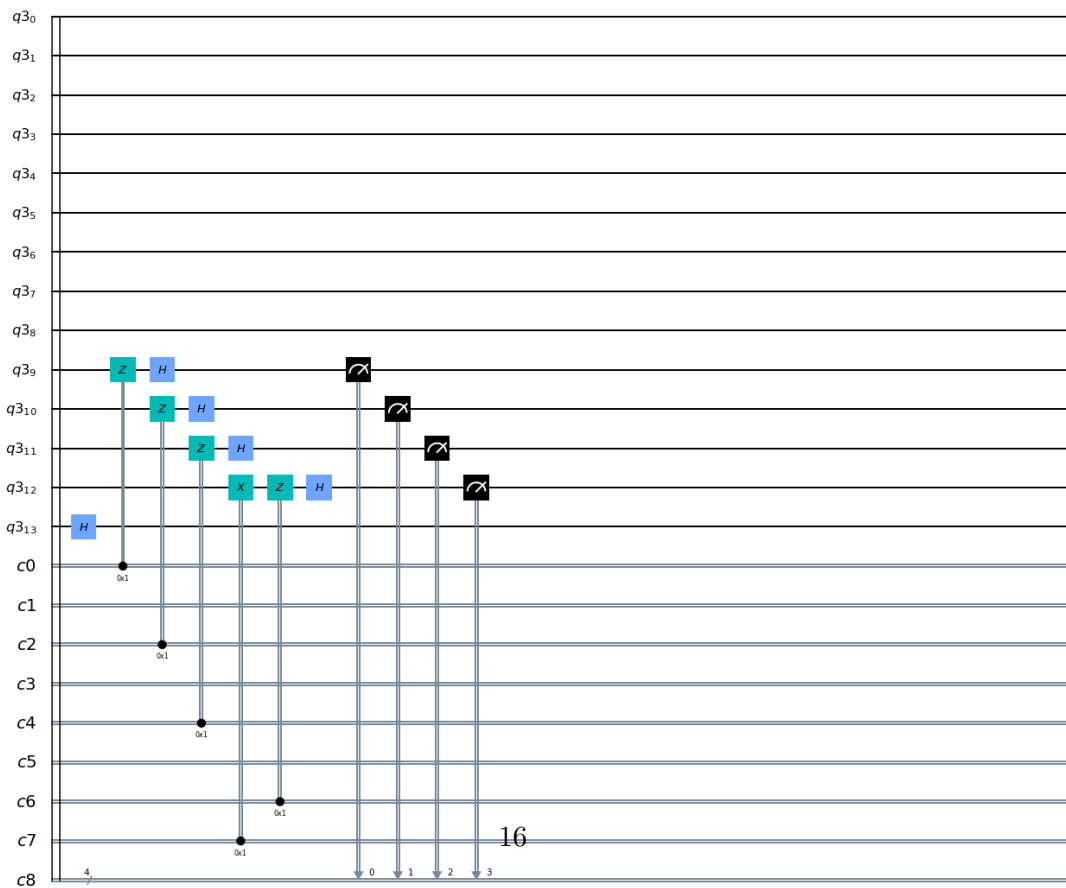
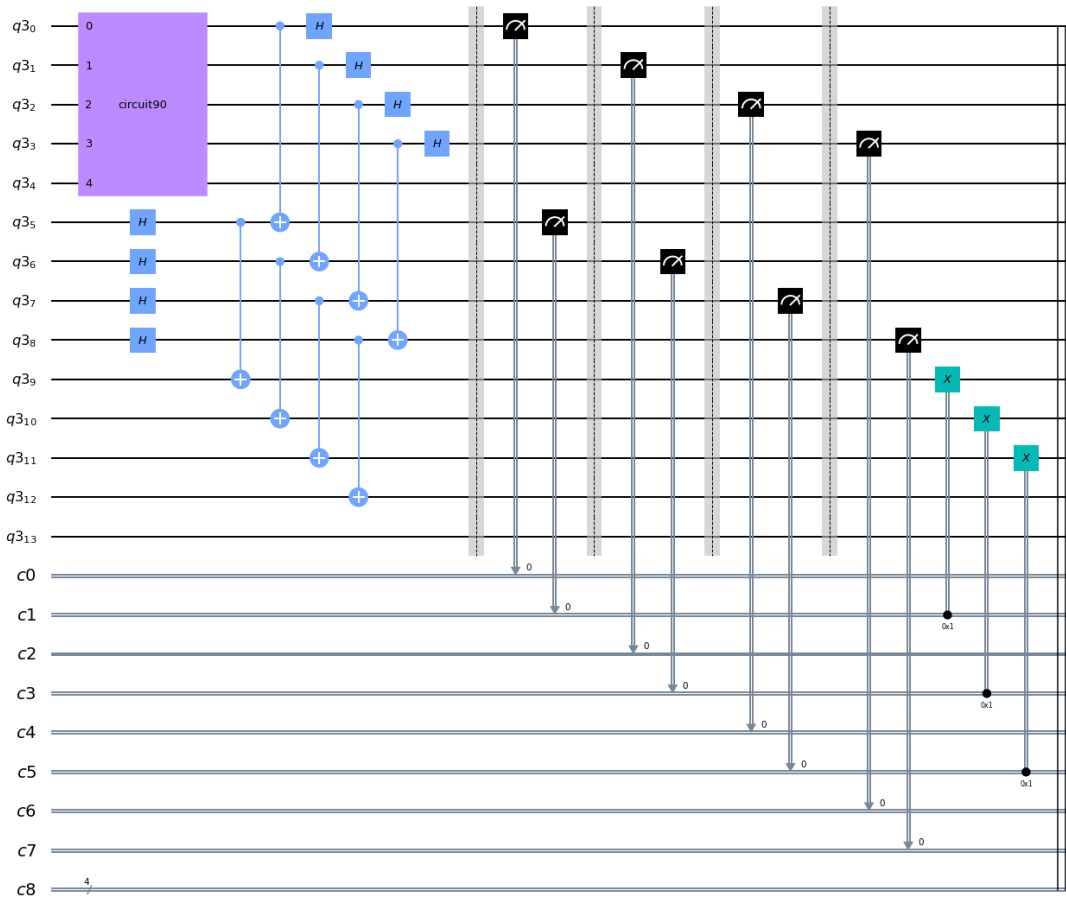
```
[16]: bob_oracle = dj_oracle('0101')  
      alice_bob_tele.append(bob_oracle, [9,10,11,12,13])  
      alice_bob_tele.draw()
```

[16]:



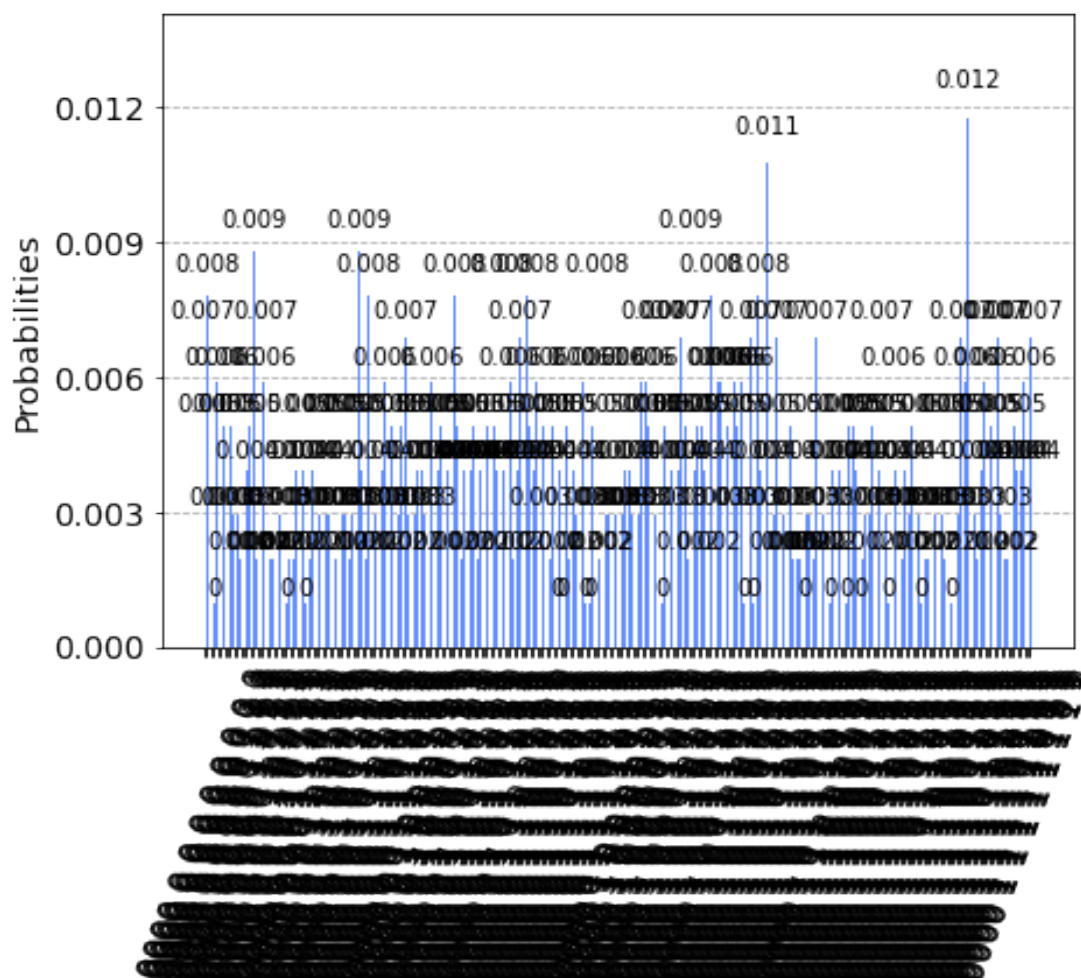
```
[67]: for qubit in [9,10,11,12,13]:  
        alice_bob_tele.h(qubit)  
  
alice_bob_tele.measure(9, 8)  
alice_bob_tele.measure(10, 9)  
alice_bob_tele.measure(11, 10)  
alice_bob_tele.measure(12, 11)  
  
alice_bob_tele.draw()
```

[67]:




```
[108]: backend = BasicAer.get_backend('qasm_simulator')
shots = 1024
results = execute(alice_bob_tele, backend=backend, shots=1024).result()
answer = results.get_counts()
plot_histogram(answer)
```

[108]:



```
[ ]: plot_histogram(answer)
```