In [1]:
```
from qiskit import *
from qiskit.visualization import plot_histogram
IBMQ.load_account()
import numpy as np
%matplotlib inline
import math as m
import scipy as sci
```

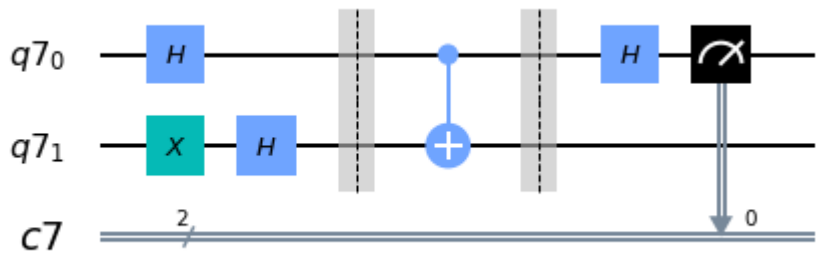
In [12]:
```
#initialising qunantum and classical registers
q=QuantumRegister(2)
c=ClassicalRegister(2)
Deutsch=QuantumCircuit(q,c)
#setting up input registers and output qubit
Deutsch.h(q[0])
Deutsch.x(q[1])
Deutsch.h(q[1])
Deutsch.barrier()
#setting up Functions
b=int(input())
print('b=',b)
if(b==1):          # If value of b=1 then function is balanced as it will appl
y CNOT gate which will give us output as 1
    Deutsch.cx(q[0],q[1])
if(b==0):         # If value of b=0 then function is constant as it will apply
X gate which will give us output as 0
    Deutsch.x(q[0])
Deutsch.barrier()
Deutsch.h(q[0])
Deutsch.measure(0,0)

backend = BasicAer.get_backend('qasm_simulator')
shots = 1024
qubit=list(execute(Deutsch,backend=backend,shots=shots).result().get_counts(De
utsch))[0][1]
print('qubit',qubit)
if(qubit=='0'):
    print('f is constant')
else:
    print('f is balanced')

Deutsch.draw(output='mpl')
```

```
1
b= 1
qubit 1
f is balanced
```

Out[12]:



In [ ]:

In [ ]: