In [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 *
from qiskit.providers.aer import QasmSimulator

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

In [16]:

```
from qiskit import*
#building an oracle for the function x1x2 xor x2x3 xor x3x1
qr=QuantumRegister(4,'q')
cr=ClassicalRegister(3,'c')
orc=QuantumCircuit(qr,cr)
for i in range(3):
    orc.h(qr[i])
orc.x(qr[3])
orc.h(qr[3])
orc.ccx(qr[1],qr[2],qr[3])
orc.ccx(qr[1],qr[0],qr[3])
orc.ccx(qr[0],qr[2],qr[3])
for i in range(3):
    orc.h(qr[i])
for i in range(3):
    orc.measure(qr[i],cr[i])
# Executing the code in the simulator
backend = Aer.get_backend('qasm_simulator')
qjob = execute(orc,backend,shots=100000)
counts = qjob.result().get_counts()
print(counts)
orc.draw()
```

```
{'001': 24982, '010': 25073, '111': 24825, '100': 25120}
```

Out[16]: