

In [7]:

```
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()
```

ibmqfactory.load_account:WARNING:2022-02-12 21:03:38,026: Credentials are already in use. The existing account in the session will be replaced.

In [13]:

```
from qiskit import*
q=QuantumRegister(6,'q')
M=ClassicalRegister(3,'c')
S_f=QuantumCircuit(q,M)
for i in range(3):
    S_f.h(q[i])

S_f.x(q[1])
S_f.cx(q[1],q[3])

S_f.x(q[0])
S_f.cx(q[0],q[1])
S_f.cx(q[1],q[2])
S_f.cx(q[2],q[4])

S_f.cx(q[1],q[2])

S_f.cx(q[0],q[1])

S_f.x(q[0])

S_f.cx(q[0],q[2])
S_f.cx(q[2],q[5])

S_f.cx(q[0],q[2])

for i in range(3):
    S_f.h(q[i])
    S_f.measure(q[i],M[i])
#Executing the circuit in simulator.
backend = Aer.get_backend('qasm_simulator')
qjob = execute(S_f,backend,shots=10)
counts= qjob.result().get_counts()
print(counts)

S_f.draw()
```

```
{'010': 1, '000': 3, '111': 6}
```

Out[13]:

