COIMBATORE INSTITUTE OF TECHNOLOGY

CAT-2

REGISTER NO: 1931012

NAME: IVAN HERALD W

CLASS: M.Sc. SOFTWARE SYSTEMS, 4th YEAR

SUBJECT: QUANTUM COMPUTING LABORATORY

SUBJECT CODE: 15MSSL15

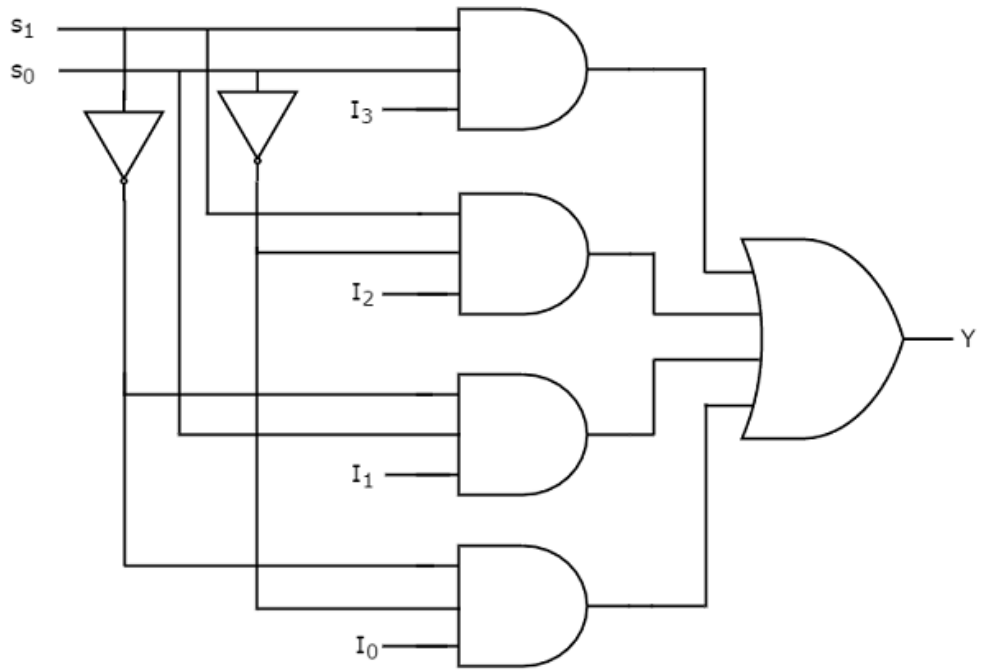
AIM

To Construct a 4*1 Multiplexer that consists of 4 parallel inputs and a single output, that is then connected to a D-Flip Flop as a clock input.

SHORT DESCRIPTION

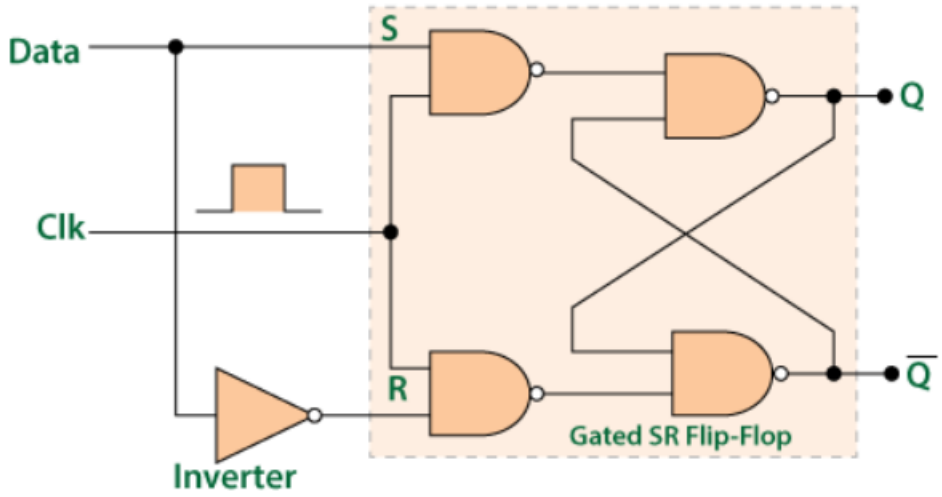
Implementation of this multiplexer and D- Flip Flop is done Using Multiple quantum Gates. The input for this circuit is 4 bits. The following code is a generalized one.

CLASSICAL MULTIPLEXER



Selection Lines		Output
S_1	S_0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

CLASSICAL D FLIP FLOP



Clock	D	Q	Q′
↓ » 0	0	0	1
↑ » 1	0	0	1
↓ » 0	1	0	1
↑ » 1	1	1	0

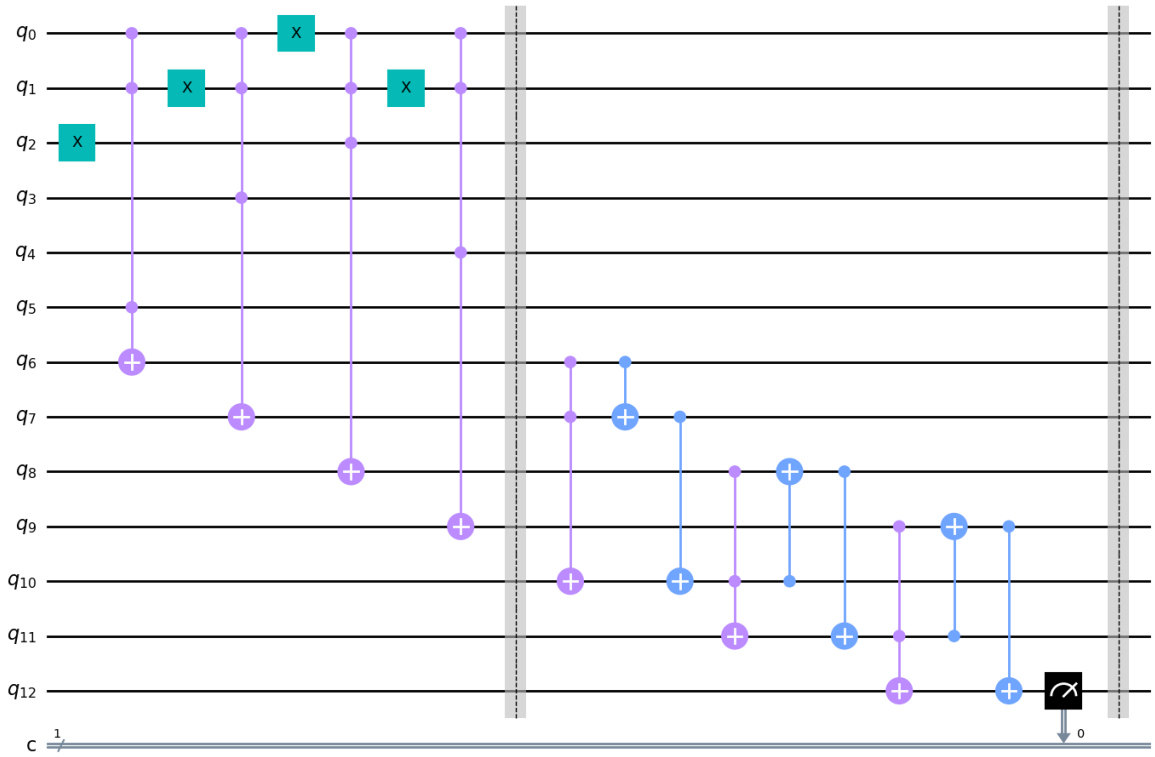
CODE

```
from qiskit import QuantumRegister, QuantumCircuit, Aer, execute
from qiskit.visualization import *
```

MULTIPLEXER

```
qc=QuantumCircuit(13,1)
qc.x(2)

qc.mct([0,1,5],6)

qc.x(1)
qc.mct([0,1,3],7)
```

```
qc.x(0)
qc.mct([0,1,2],8)

qc.x(1)
qc.mct([0,1,4],9)

qc.barrier()
qc.ccx(6,7,10)
qc.cx(6,7)
qc.cx(7,10)

qc.ccx(8,10,11)
qc.cx(10,8)
qc.cx(8,11)

qc.ccx(9,11,12)
qc.cx(11,9)
qc.cx(9,12)
qc.measure(12,0)
qc.barrier()
qc.draw('mpl')
```



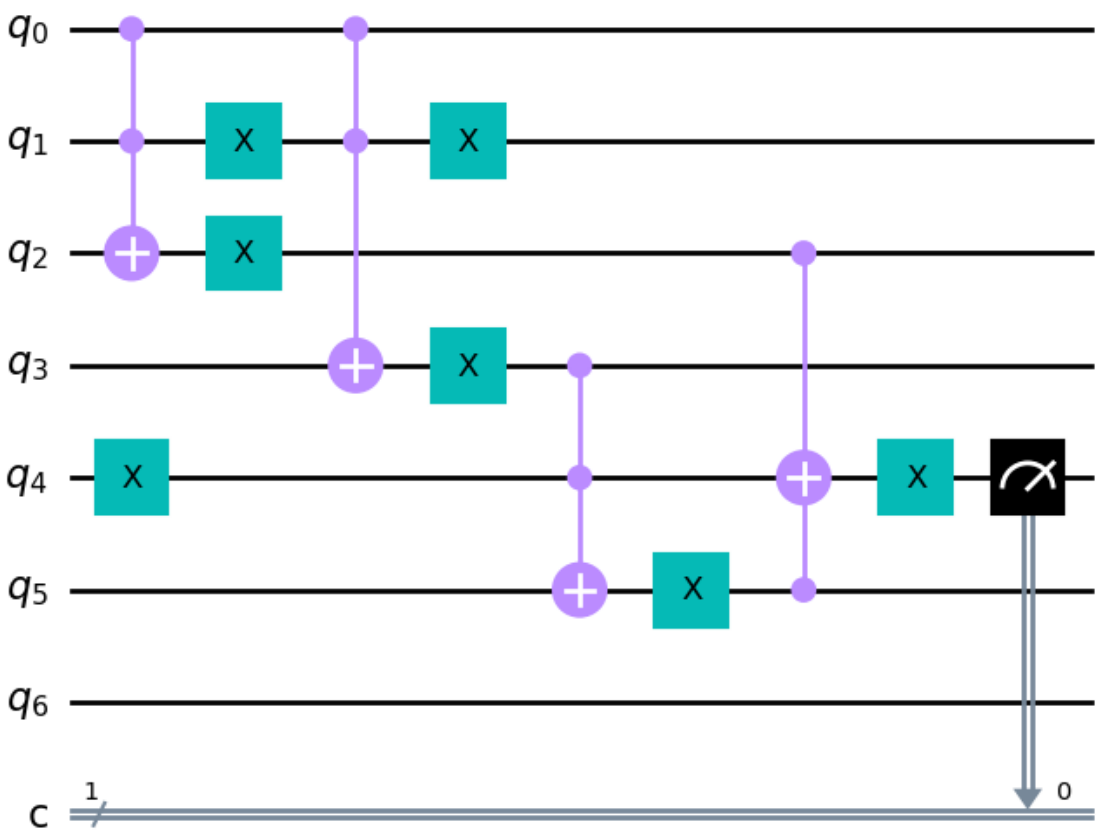

```
backend = Aer.get_backend('qasm_simulator')
job = backend.run(qc, shots=1024, memory=True)
output = job.result().get_memory()[0]
print(output)
```

OUTPUT:1

D Flip Flop

```
qc=QuantumCircuit(7,1)


# qc.x(0)
# qc.x(1)
qc.x(4)


qc.ccx(0,1,2)
qc.x(2)
```

```
qc.x(1)
qc.ccx(0,1,3)
qc.x(3)
qc.x(1)
qc.ccx(3,4,5)
qc.x(5)
qc.ccx(2,5,4)
qc.x(4)
qc.measure(4,0)
qc.draw('mpl')
```




```
backend = Aer.get_backend('qasm_simulator')
job = backend.run(qc, shots=1024, memory=True)
```

```
output = job.result().get_memory()[0]
print(output)
```

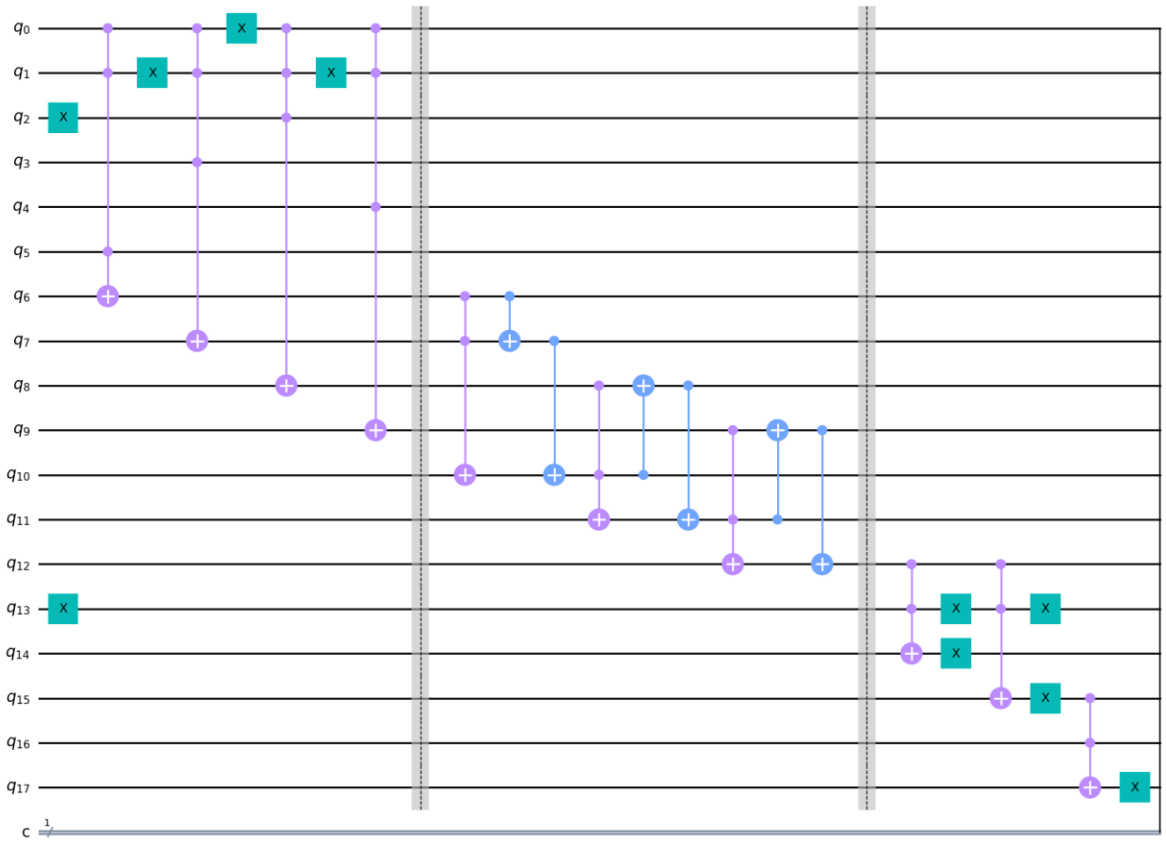
OUTPUT: 0

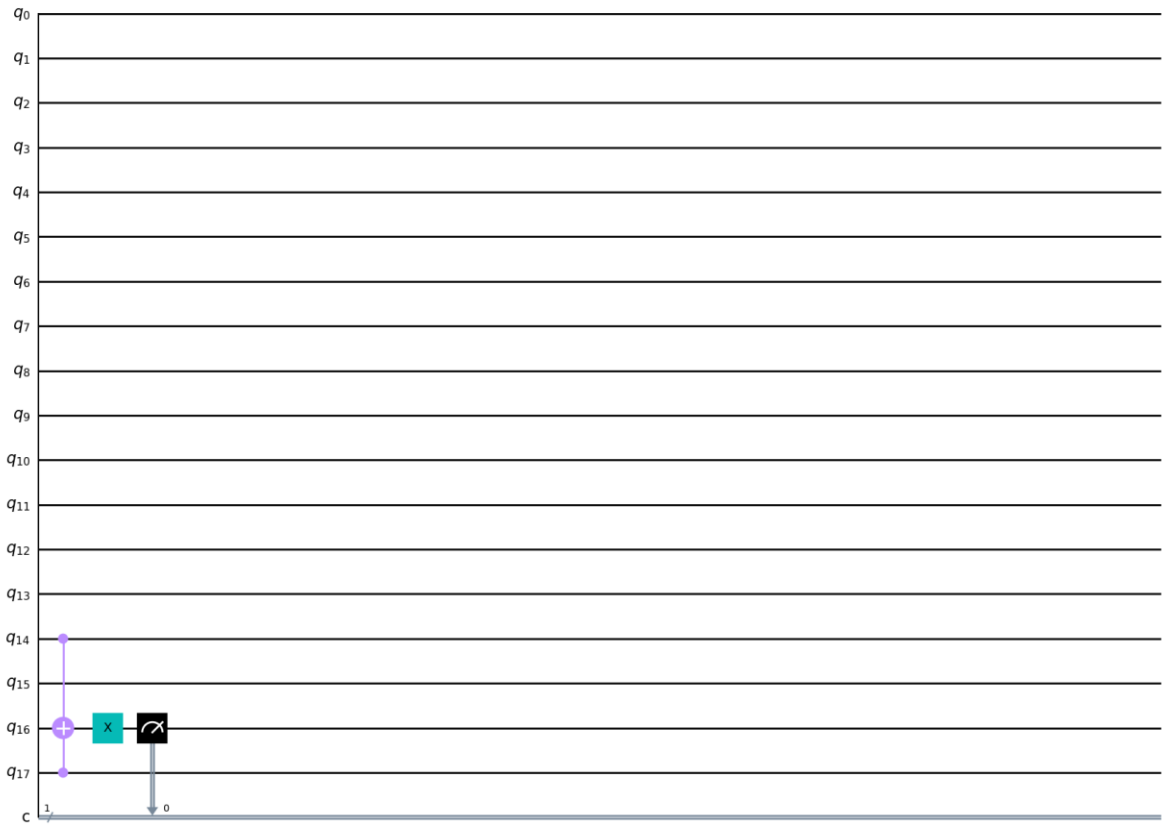
COMPLETE CIRCUIT:

```
qc=QuantumCircuit(18,1)
# s0 - q0
# s1 - q1
# I0 - q2
# I1 - q3
# I2 - q4
# I3 - q5

# S0 , S1 , I3 - q6
# S1' , S0 , I1 - q7
# S1' , S0' , I0 - q8
# S1 , S0' , I2 - q9


# multiplexer output - q12
# D flip flop D      - q13
# Q                  - q16
```

```
# s0 - 0
# s1 - 0
# I0 - 1
# y  - 1  --->multiplexer output

qc.x(2)
qc.x(13)

qc.mct([0,1,5],6)

qc.x(1)
qc.mct([0,1,3],7)

qc.x(0)
qc.mct([0,1,2],8)

qc.x(1)
qc.mct([0,1,4],9)

qc.barrier()
qc.ccx(6,7,10)
qc.cx(6,7)
qc.cx(7,10)
```

```
qc.ccx(8,10,11)
qc.cx(10,8)
qc.cx(8,11)

qc.ccx(9,11,12)
qc.cx(11,9)
qc.cx(9,12)

qc.barrier()



qc.ccx(12,13,14)
qc.x(14)
qc.x(13)
qc.ccx(12,13,15)
qc.x(13)
qc.x(15)
qc.ccx(15,16,17)
qc.x(17)
qc.ccx(14,17,16)
qc.x(16)
qc.measure(16,0)

qc.draw('mpl')
```


q0
q1
q2
q3
q4
q5
q6
q7
q8
q9
q10
q11
q12
q13
q14
q15
q16
q17
c
1
X


q0
q1
q2
q3
q4
q5
q6
q7
q8
q9
q10
q11
q12
q13
q14
q15
q16
q17
c
1
X
0

```
backend = Aer.get_backend('qasm_simulator')
job = backend.run(qc, shots=1024, memory=True)
output = job.result().get_memory()[0]
print(output)
```

Output:1

SAMPLE TESTCASES

INPUT:

S0,S1,I0,I1,I2,13,D,Q

0010000000000100100

OUTPUT:

1

THINGS LEARNED

- To reduce the depth of circuit.
- Tried a new experiment of connecting two different classical circuits and producing a single output.

REFERENCES

- **https://www.tutorialspoint.com/digital_circuits/digital_circuits_multiplexers.htm**
- **https://www.electronicsforu.com/technology-trends/learn-electronics/flip-flop-rs-jk-t-d**
- **https://arxiv.org/abs/1807.02940#:~:text=Distributing%20entangled%20pairs%20is%20a,pair%20of%20remote%20quantum%20memories.**
- **https://www.irjmets.com/uploadedfiles/paper//issue_5_may_2022/22088/final/fin_irjmets1651928209.pdf**