multi controlled classicQ

May 25, 2022

[220]:
```
import time
import numpy as np
from qiskit import*
```

[221]:
```
start = time.time()
Control_Qubits = [0,1,2,3,4,5,6,7,8,9,10,11,12,13]
Target_Qubits = [15]
Ancilia_Qubits = [14]
```

[222]:
```
t = []

N = len(Control_Qubits)+2
```

[223]:
```
def Toffoli(control_1, control_2, target):

    return [control_1, control_2, target]
```

[224]:
```
Gates = []
i = 0
while len(Control_Qubits) > 1:

    if i == 0 :

        Gates.append(Toffoli(Control_Qubits[0], Control_Qubits[1],␣
 ↪Ancilia_Qubits[0]))
        Ancilia_Qubits.append(Control_Qubits[0])
        Ancilia_Qubits.append(Control_Qubits[1])
        Control_Qubits.remove(Control_Qubits[0])
        Control_Qubits.remove(Control_Qubits[0])
        t.append(Ancilia_Qubits[0]) # result is in original ancilia.


    elif i == 1:

        Gates.append(Toffoli(t[0], Control_Qubits[0], Ancilia_Qubits[1]))
        Ancilia_Qubits.append(Control_Qubits[0])
        Control_Qubits.remove(Control_Qubits[0])
        t = []
        t.append(Ancilia_Qubits[1])

    else:

        Gates.append(Toffoli(t[0], Control_Qubits[0], Ancilia_Qubits[i]))
        Ancilia_Qubits.append(Control_Qubits[0])
        Control_Qubits.remove(Control_Qubits[0])
        t = []
        t.append(Ancilia_Qubits[i])

    i = i + 1

# final toffoli.
c1 = Gates[-1][-1]
c2 = Control_Qubits[-1]
t = Target_Qubits[-1]
Gates.append(Toffoli(c1, c2, t))
```

[225]:

```
#qc = QuantumCircuit(N)
#for i in Gates:
    #if Gates.index(i) == 0 or Gates.index(i) == len(Gates)-1:
        #pass
    #else:
        #qc.x(i[2])
    #qc.ccx(i[0],i[1],i[2])
#qc.draw('mpl')
```

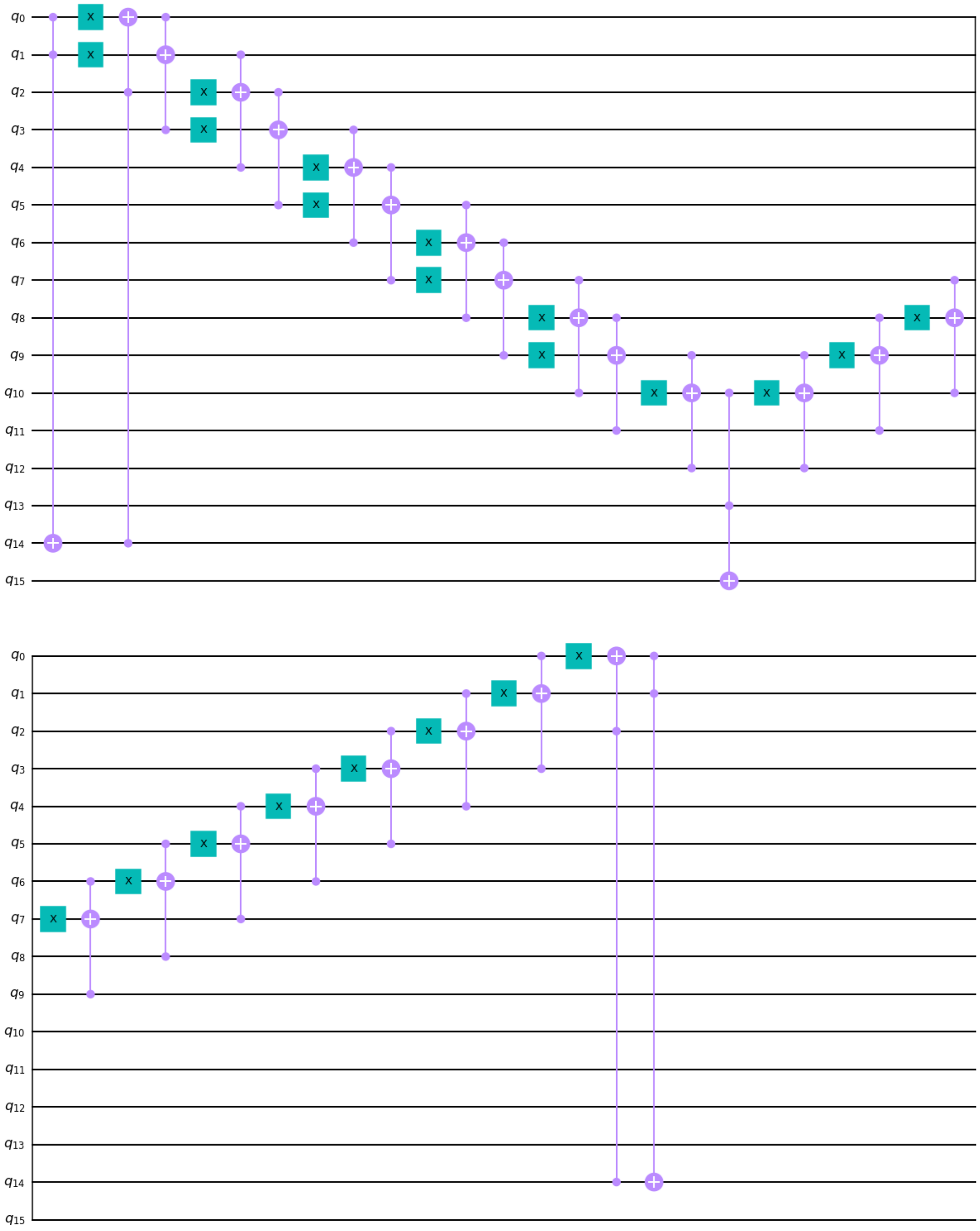
[226]:

```
# first part of the circuit.
qc = QuantumCircuit(N)
for i in range(len(Gates)):
    if i == 0 or i == len(Gates)-1:
        pass
    else:
        qc.x(Gates[i][2])
    qc.ccx(Gates[i][0],Gates[i][1],Gates[i][2])

# second part of the circuit.
for i in range(len(Gates)-2,-1,-1):
    if i == 0 or i == len(Gates)-1:
        pass
    else:
        qc.x(Gates[i][2])
    qc.ccx(Gates[i][0],Gates[i][1],Gates[i][2])
print('time taken', time.time() - start,'seconds')
qc.draw('mpl')
```

```
time taken 0.03775930404663086 seconds
```

[226]:


q_0
q_1
q_2
q_3
q_4
q_5
q_6
q_7
q_8
q_9
q_{10}
q_{11}
q_{12}
q_{13}
q_{14}
q_{15}
X