

```

In [1]: # CNF equation:
# (x1 v x2 v x3)
#   ^ (x2 v x3)
#   ^ (x3 v x4 v x5)
#   ^ (x1 v x3 v x5)
#   ^ (x2 v x4)
#   ^ (¬x1 v x2 v x3 v x4 v x5)
#   ^ (x2 v ¬x3 v x4)
#   ^ (x3 v x4 v ¬x5)
#   ^ (¬x1 v x3 v x4 v x5)
#   ^ (x2 v ¬x4 v x6)
#   ^ (¬x1 v ¬x2 v x3 v x5)
#   ^ (x2 v ¬x3 v ¬x4)
#   ^ (¬x3 v x5)
#   ^ (x1 v x2 v ¬x3 v ¬x4 v ¬x5)
#   ^ (¬x2 v ¬x3 v x4 v ¬x5)

import numpy as np

# Importing standard Qiskit libraries
from qiskit import *
from qiskit.circuit import *
from qiskit.tools.jupyter import *
from qiskit.visualization import *
from ibm_quantum_widgets import *
from qiskit.providers.aer import QasmSimulator
from qiskit.circuit.library.standard_gates import XGate,ZGate,HGate

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

q_reg = QuantumRegister(20, 'q')
c_reg = ClassicalRegister(5, 'c')
circuit = QuantumCircuit(q_reg, c_reg)

# Qslice 0
circuit.append(HGate(), [q_reg[0]])
circuit.append(HGate(), [q_reg[1]])
circuit.append(HGate(), [q_reg[2]])
circuit.append(HGate(), [q_reg[3]])
circuit.append(HGate(), [q_reg[4]])
circuit.barrier(q_reg)
# Qslice 1
circuit.append(XGate(), [q_reg[0]])
circuit.append(XGate(), [q_reg[1]])
circuit.append(XGate(), [q_reg[2]])
# Qslice 2
circuit.append(XGate().control(3), [q_reg[0], q_reg[1], q_reg[2], q_reg[5]])

```

```
# Qslice 3
circuit.append(XGate(), [q_reg[0]])
circuit.append(XGate(), [q_reg[1]])
circuit.append(XGate(), [q_reg[2]])
circuit.append(XGate(), [q_reg[5]])
circuit.barrier(q_reg)
# Qslice 4
circuit.append(XGate(), [q_reg[1]])
circuit.append(XGate(), [q_reg[2]])
# Qslice 5
circuit.append(XGate().control(2), [q_reg[1], q_reg[2], q_reg[6]])
# Qslice 6
circuit.append(XGate(), [q_reg[1]])
circuit.append(XGate(), [q_reg[2]])
circuit.append(XGate(), [q_reg[6]])
circuit.barrier(q_reg)
# Qslice 7
circuit.append(XGate(), [q_reg[2]])
circuit.append(XGate(), [q_reg[3]])
circuit.append(XGate(), [q_reg[4]])
# Qslice 8
circuit.append(XGate().control(3), [q_reg[2], q_reg[3], q_reg[4], q_reg[7]])
# Qslice 9
circuit.append(XGate(), [q_reg[2]])
circuit.append(XGate(), [q_reg[3]])
circuit.append(XGate(), [q_reg[4]])
circuit.append(XGate(), [q_reg[7]])
circuit.barrier(q_reg)
# Qslice 10
circuit.append(XGate(), [q_reg[0]])
circuit.append(XGate(), [q_reg[2]])
circuit.append(XGate(), [q_reg[4]])
# Qslice 11
circuit.append(XGate().control(3), [q_reg[0], q_reg[2], q_reg[4], q_reg[8]])
# Qslice 12
circuit.append(XGate(), [q_reg[0]])
circuit.append(XGate(), [q_reg[2]])
circuit.append(XGate(), [q_reg[4]])
circuit.append(XGate(), [q_reg[8]])
circuit.barrier(q_reg)
# Qslice 13
circuit.append(XGate(), [q_reg[1]])
circuit.append(XGate(), [q_reg[3]])
# Qslice 14
circuit.append(XGate().control(2), [q_reg[1], q_reg[3], q_reg[9]])
# Qslice 15
circuit.append(XGate(), [q_reg[1]])
circuit.append(XGate(), [q_reg[3]])
circuit.append(XGate(), [q_reg[9]])
circuit.barrier(q_reg)
# Qslice 16
circuit.append(XGate(), [q_reg[1]])
```

```
circuit.append(XGate(), [q_reg[2]])
circuit.append(XGate(), [q_reg[3]])
circuit.append(XGate(), [q_reg[4]])
# Qslice 17
circuit.append(XGate().control(5), [q_reg[0], q_reg[1], q_reg[2], q_reg[3], q_reg[4], q_reg[10]])
# Qslice 18
circuit.append(XGate(), [q_reg[1]])
circuit.append(XGate(), [q_reg[2]])
circuit.append(XGate(), [q_reg[3]])
circuit.append(XGate(), [q_reg[4]])
circuit.append(XGate(), [q_reg[10]])
circuit.barrier(q_reg)
# Qslice 19
circuit.append(XGate(), [q_reg[1]])
circuit.append(XGate(), [q_reg[3]])
# Qslice 20
circuit.append(XGate().control(3), [q_reg[1], q_reg[2], q_reg[3], q_reg[11]])
# Qslice 21
circuit.append(XGate(), [q_reg[1]])
circuit.append(XGate(), [q_reg[3]])
circuit.append(XGate(), [q_reg[11]])
circuit.barrier(q_reg)
# Qslice 22
circuit.append(XGate(), [q_reg[2]])
circuit.append(XGate(), [q_reg[3]])
# Qslice 23
circuit.append(XGate().control(3), [q_reg[2], q_reg[3], q_reg[4], q_reg[12]])
# Qslice 24
circuit.append(XGate(), [q_reg[2]])
circuit.append(XGate(), [q_reg[3]])
circuit.append(XGate(), [q_reg[12]])
circuit.barrier(q_reg)
# Qslice 25
circuit.append(XGate(), [q_reg[2]])
circuit.append(XGate(), [q_reg[3]])
circuit.append(XGate(), [q_reg[4]])
# Qslice 26
circuit.append(XGate().control(4), [q_reg[0], q_reg[2], q_reg[3], q_reg[4], q_reg[13]])
# Qslice 27
circuit.append(XGate(), [q_reg[2]])
circuit.append(XGate(), [q_reg[3]])
circuit.append(XGate(), [q_reg[4]])
circuit.append(XGate(), [q_reg[13]])
circuit.barrier(q_reg)
# Qslice 28
circuit.append(XGate(), [q_reg[1]])
circuit.append(XGate(), [q_reg[4]])
# Qslice 29
circuit.append(XGate().control(3), [q_reg[1], q_reg[3], q_reg[4], q_reg[14]])
```

```
# Qslice 30
circuit.append(XGate(), [q_reg[1]])
circuit.append(XGate(), [q_reg[4]])
circuit.append(XGate(), [q_reg[14]])
circuit.barrier(q_reg)
# Qslice 31
circuit.append(XGate(), [q_reg[2]])
circuit.append(XGate(), [q_reg[4]])
# Qslice 32
circuit.append(XGate().control(4), [q_reg[0], q_reg[1], q_reg[2], q_reg[4], q_reg[15]])
# Qslice 33
circuit.append(XGate(), [q_reg[2]])
circuit.append(XGate(), [q_reg[4]])
circuit.append(XGate(), [q_reg[15]])
circuit.barrier(q_reg)
# Qslice 34
circuit.append(XGate(), [q_reg[1]])
# Qslice 35
circuit.append(XGate().control(3), [q_reg[1], q_reg[2], q_reg[3], q_reg[16]])
# Qslice 36
circuit.append(XGate(), [q_reg[1]])
circuit.append(XGate(), [q_reg[16]])
circuit.barrier(q_reg)
# Qslice 37
circuit.append(XGate(), [q_reg[4]])
# Qslice 38
circuit.append(XGate().control(2), [q_reg[2], q_reg[4], q_reg[17]])
# Qslice 39
circuit.append(XGate(), [q_reg[4]])
circuit.append(XGate(), [q_reg[17]])
circuit.barrier(q_reg)
# Qslice 40
circuit.append(XGate(), [q_reg[0]])
circuit.append(XGate(), [q_reg[1]])
# Qslice 41
circuit.append(XGate().control(5), [q_reg[0], q_reg[1], q_reg[2], q_reg[3], q_reg[4], q_reg[18]])
# Qslice 42
circuit.append(XGate(), [q_reg[0]])
circuit.append(XGate(), [q_reg[1]])
circuit.append(XGate(), [q_reg[18]])
circuit.barrier(q_reg)
# Qslice 43
circuit.append(XGate(), [q_reg[3]])
# Qslice 44
circuit.append(XGate().control(4), [q_reg[1], q_reg[2], q_reg[3], q_reg[4], q_reg[19]])
# Qslice 45
circuit.append(XGate(), [q_reg[3]])
circuit.append(XGate(), [q_reg[19]])
circuit.barrier(q_reg)
# Qslice 46
```

```

circuit.append(ZGate().control(14),[q_reg[5], q_reg[6], q_reg[7], q
_reg[8], q_reg[9], q_reg[10], q_reg[11], q_reg[12], q_reg[13], q_re
g[14], q_reg[15], q_reg[16], q_reg[17], q_reg[18], q_reg[19]])
circuit.barrier(q_reg)
# Qslice 45
circuit.append(XGate(),[q_reg[3]])
circuit.append(XGate(),[q_reg[19]])
# Qslice 44
circuit.append(XGate().control(4),[q_reg[1], q_reg[2], q_reg[3], q_
reg[4], q_reg[19]])
# Qslice 43
circuit.append(XGate(),[q_reg[3]])
circuit.barrier(q_reg)
# Qslice 42
circuit.append(XGate(),[q_reg[0]])
circuit.append(XGate(),[q_reg[1]])
circuit.append(XGate(),[q_reg[18]])
# Qslice 41
circuit.append(XGate().control(5),[q_reg[0], q_reg[1], q_reg[2], q_
reg[3], q_reg[4], q_reg[18]])
# Qslice 40
circuit.append(XGate(),[q_reg[0]])
circuit.append(XGate(),[q_reg[1]])
circuit.barrier(q_reg)
# Qslice 39
circuit.append(XGate(),[q_reg[4]])
circuit.append(XGate(),[q_reg[17]])
# Qslice 38
circuit.append(XGate().control(2),[q_reg[2], q_reg[4], q_reg[17]])
# Qslice 37
circuit.append(XGate(),[q_reg[4]])
circuit.barrier(q_reg)
# Qslice 36
circuit.append(XGate(),[q_reg[1]])
circuit.append(XGate(),[q_reg[16]])
# Qslice 35
circuit.append(XGate().control(3),[q_reg[1], q_reg[2], q_reg[3], q
_reg[16]])
# Qslice 34
circuit.append(XGate(),[q_reg[1]])
circuit.barrier(q_reg)
# Qslice 33
circuit.append(XGate(),[q_reg[2]])
circuit.append(XGate(),[q_reg[4]])
circuit.append(XGate(),[q_reg[15]])
# Qslice 32
circuit.append(XGate().control(4),[q_reg[0], q_reg[1], q_reg[2], q_
reg[4], q_reg[15]])
# Qslice 31
circuit.append(XGate(),[q_reg[2]])
circuit.append(XGate(),[q_reg[4]])
circuit.barrier(q_reg)
# Qslice 30
circuit.append(XGate(),[q_reg[1]])

```

```
circuit.append(XGate(), [q_reg[4]])
circuit.append(XGate(), [q_reg[14]])
# Qslice 29
circuit.append(XGate().control(3), [q_reg[1], q_reg[3], q_reg[4], q_reg[14]])
# Qslice 28
circuit.append(XGate(), [q_reg[1]])
circuit.append(XGate(), [q_reg[4]])
circuit.barrier(q_reg)
# Qslice 27
circuit.append(XGate(), [q_reg[2]])
circuit.append(XGate(), [q_reg[3]])
circuit.append(XGate(), [q_reg[4]])
circuit.append(XGate(), [q_reg[13]])
# Qslice 26
circuit.append(XGate().control(4), [q_reg[0], q_reg[2], q_reg[3], q_reg[4], q_reg[13]])
# Qslice 25
circuit.append(XGate(), [q_reg[2]])
circuit.append(XGate(), [q_reg[3]])
circuit.append(XGate(), [q_reg[4]])
circuit.barrier(q_reg)
# Qslice 24
circuit.append(XGate(), [q_reg[2]])
circuit.append(XGate(), [q_reg[3]])
circuit.append(XGate(), [q_reg[12]])
# Qslice 23
circuit.append(XGate().control(3), [q_reg[2], q_reg[3], q_reg[4], q_reg[12]])
# Qslice 22
circuit.append(XGate(), [q_reg[2]])
circuit.append(XGate(), [q_reg[3]])
circuit.barrier(q_reg)
# Qslice 21
circuit.append(XGate(), [q_reg[1]])
circuit.append(XGate(), [q_reg[3]])
circuit.append(XGate(), [q_reg[11]])
# Qslice 20
circuit.append(XGate().control(3), [q_reg[1], q_reg[2], q_reg[3], q_reg[11]])
# Qslice 19
circuit.append(XGate(), [q_reg[1]])
circuit.append(XGate(), [q_reg[3]])
circuit.barrier(q_reg)
# Qslice 18
circuit.append(XGate(), [q_reg[1]])
circuit.append(XGate(), [q_reg[2]])
circuit.append(XGate(), [q_reg[3]])
circuit.append(XGate(), [q_reg[4]])
circuit.append(XGate(), [q_reg[10]])
# Qslice 17
circuit.append(XGate().control(5), [q_reg[0], q_reg[1], q_reg[2], q_reg[3], q_reg[4], q_reg[10]])
# Qslice 16
```

```
circuit.append(XGate(), [q_reg[1]])
circuit.append(XGate(), [q_reg[2]])
circuit.append(XGate(), [q_reg[3]])
circuit.append(XGate(), [q_reg[4]])
circuit.barrier(q_reg)
# Qslice 15
circuit.append(XGate(), [q_reg[1]])
circuit.append(XGate(), [q_reg[3]])
circuit.append(XGate(), [q_reg[9]])
# Qslice 14
circuit.append(XGate().control(2), [q_reg[1], q_reg[3], q_reg[9]])
# Qslice 13
circuit.append(XGate(), [q_reg[1]])
circuit.append(XGate(), [q_reg[3]])
circuit.barrier(q_reg)
# Qslice 12
circuit.append(XGate(), [q_reg[0]])
circuit.append(XGate(), [q_reg[2]])
circuit.append(XGate(), [q_reg[4]])
circuit.append(XGate(), [q_reg[8]])
# Qslice 11
circuit.append(XGate().control(3), [q_reg[0], q_reg[2], q_reg[4], q_reg[8]])
# Qslice 10
circuit.append(XGate(), [q_reg[0]])
circuit.append(XGate(), [q_reg[2]])
circuit.append(XGate(), [q_reg[4]])
circuit.barrier(q_reg)
# Qslice 9
circuit.append(XGate(), [q_reg[2]])
circuit.append(XGate(), [q_reg[3]])
circuit.append(XGate(), [q_reg[4]])
circuit.append(XGate(), [q_reg[7]])
# Qslice 8
circuit.append(XGate().control(3), [q_reg[2], q_reg[3], q_reg[4], q_reg[7]])
# Qslice 7
circuit.append(XGate(), [q_reg[2]])
circuit.append(XGate(), [q_reg[3]])
circuit.append(XGate(), [q_reg[4]])
circuit.barrier(q_reg)
# Qslice 6
circuit.append(XGate(), [q_reg[1]])
circuit.append(XGate(), [q_reg[2]])
circuit.append(XGate(), [q_reg[6]])
# Qslice 5
circuit.append(XGate().control(2), [q_reg[1], q_reg[2], q_reg[6]])
# Qslice 4
circuit.append(XGate(), [q_reg[1]])
circuit.append(XGate(), [q_reg[2]])
circuit.barrier(q_reg)
# Qslice 3
circuit.append(XGate(), [q_reg[0]])
circuit.append(XGate(), [q_reg[1]])
```

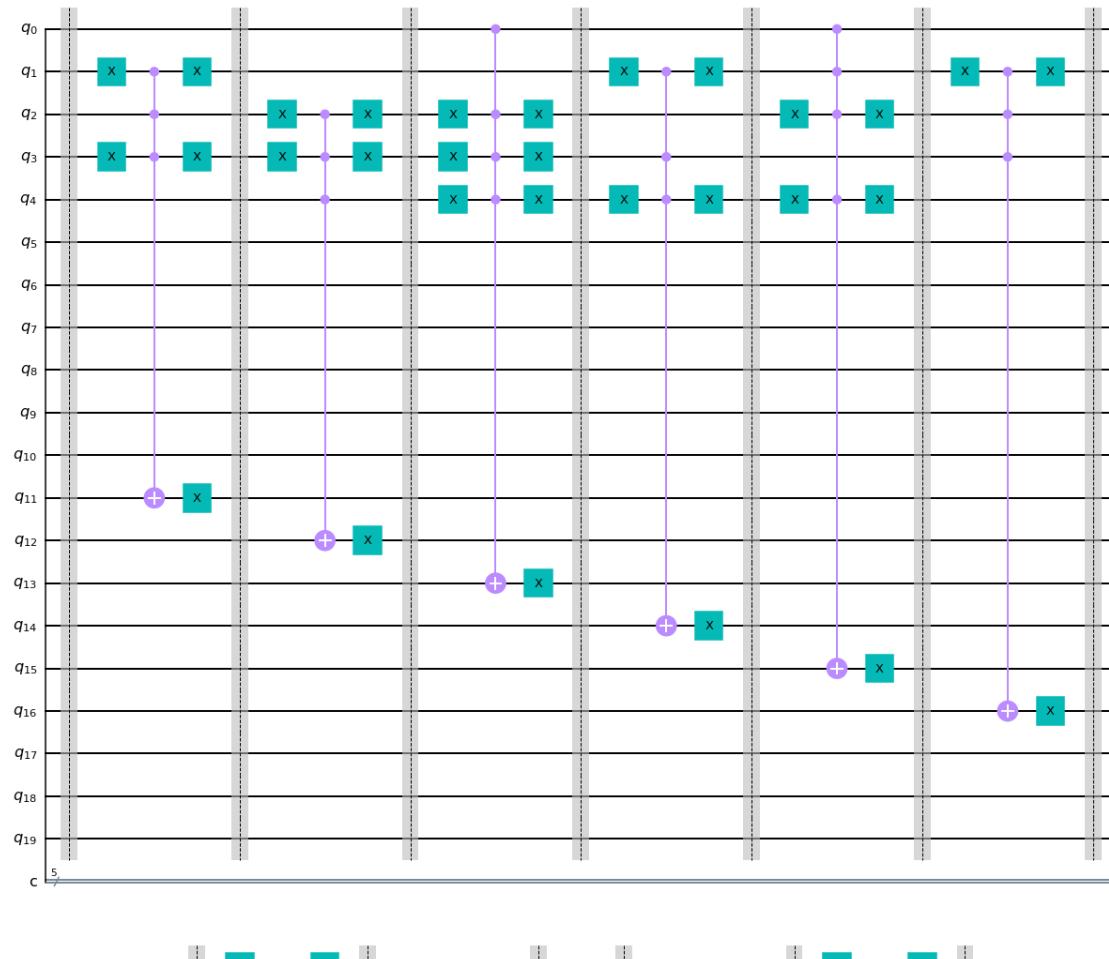
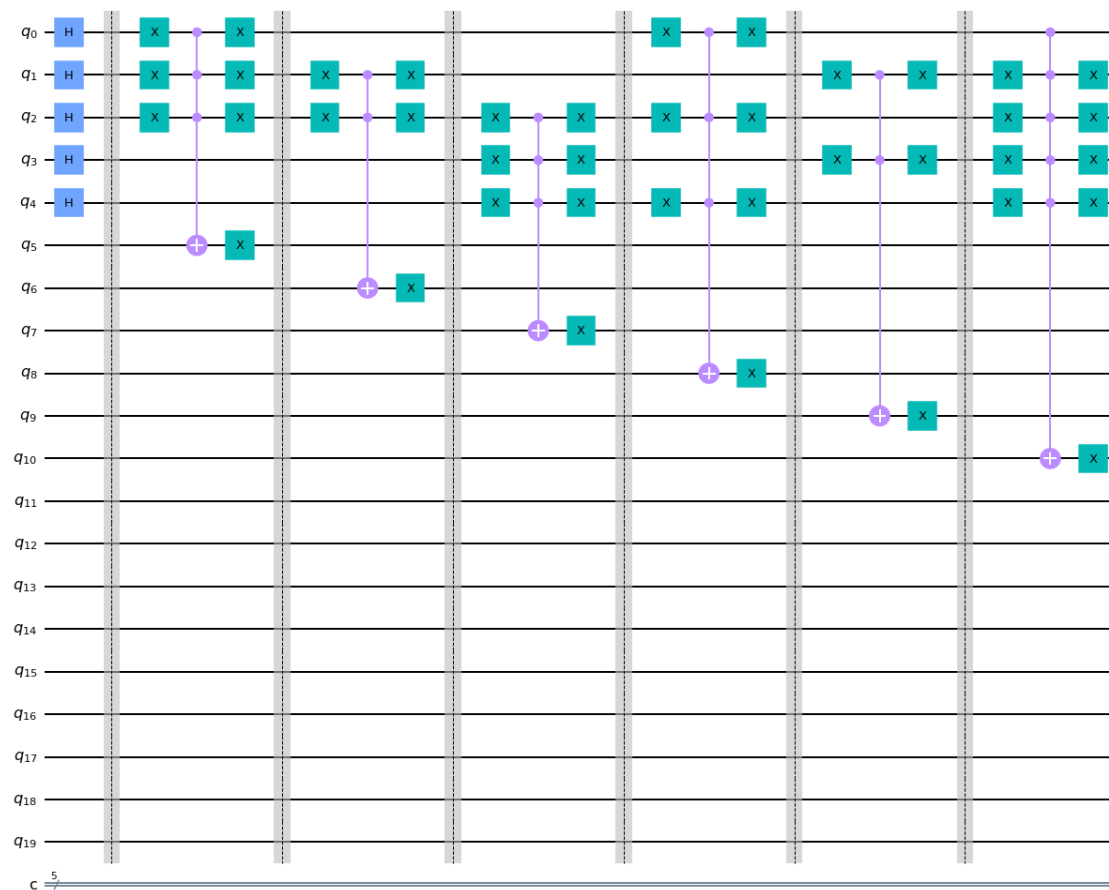
```
circuit.append(XGate(), [q_reg[2]])
circuit.append(XGate(), [q_reg[5]])
# Qslice 2
circuit.append(XGate().control(3), [q_reg[0], q_reg[1], q_reg[2], q_reg[5]])
# Qslice 1
circuit.append(XGate(), [q_reg[0]])
circuit.append(XGate(), [q_reg[1]])
circuit.append(XGate(), [q_reg[2]])
circuit.barrier(q_reg)
# Qslice 47
circuit.append(HGate(), [q_reg[0]])
circuit.append(HGate(), [q_reg[1]])
circuit.append(HGate(), [q_reg[2]])
circuit.append(HGate(), [q_reg[3]])
circuit.append(HGate(), [q_reg[4]])
# Qslice 48
circuit.append(XGate(), [q_reg[0]])
circuit.append(XGate(), [q_reg[1]])
circuit.append(XGate(), [q_reg[2]])
circuit.append(XGate(), [q_reg[3]])
circuit.append(XGate(), [q_reg[4]])
# Qslice 49
circuit.append(ZGate().control(4), [q_reg[0], q_reg[1], q_reg[2], q_reg[3], q_reg[4]])
# Qslice 48
circuit.append(XGate(), [q_reg[0]])
circuit.append(XGate(), [q_reg[1]])
circuit.append(XGate(), [q_reg[2]])
circuit.append(XGate(), [q_reg[3]])
circuit.append(XGate(), [q_reg[4]])
# Qslice 51
circuit.append(HGate(), [q_reg[0]])
circuit.append(HGate(), [q_reg[1]])
circuit.append(HGate(), [q_reg[2]])
circuit.append(HGate(), [q_reg[3]])
circuit.append(HGate(), [q_reg[4]])
circuit.barrier(q_reg)
# Qslice 52
circuit.measure(q_reg[0], c_reg[0])
circuit.measure(q_reg[1], c_reg[1])
circuit.measure(q_reg[2], c_reg[2])
circuit.measure(q_reg[3], c_reg[3])
circuit.measure(q_reg[4], c_reg[4])

circuit.draw('mpl')
```

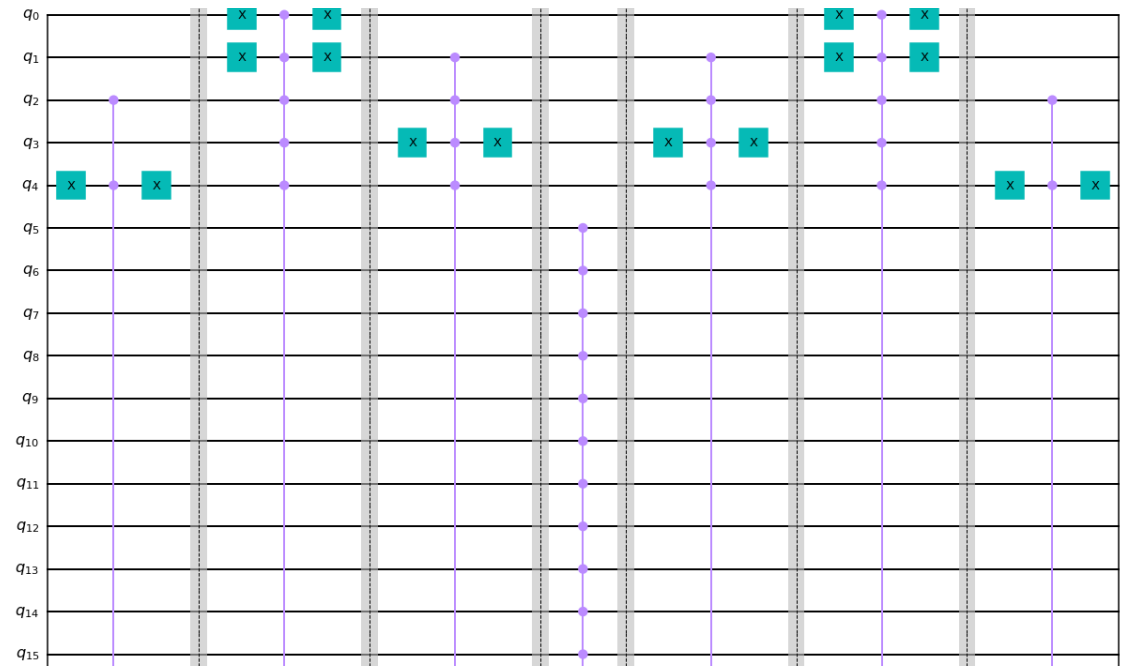


```
<frozen importlib._bootstrap>:219: RuntimeWarning: scipy._lib.messagestream.MessageStream size changed, may indicate binary incompatibility. Expected 56 from C header, got 64 from Py0bject
```

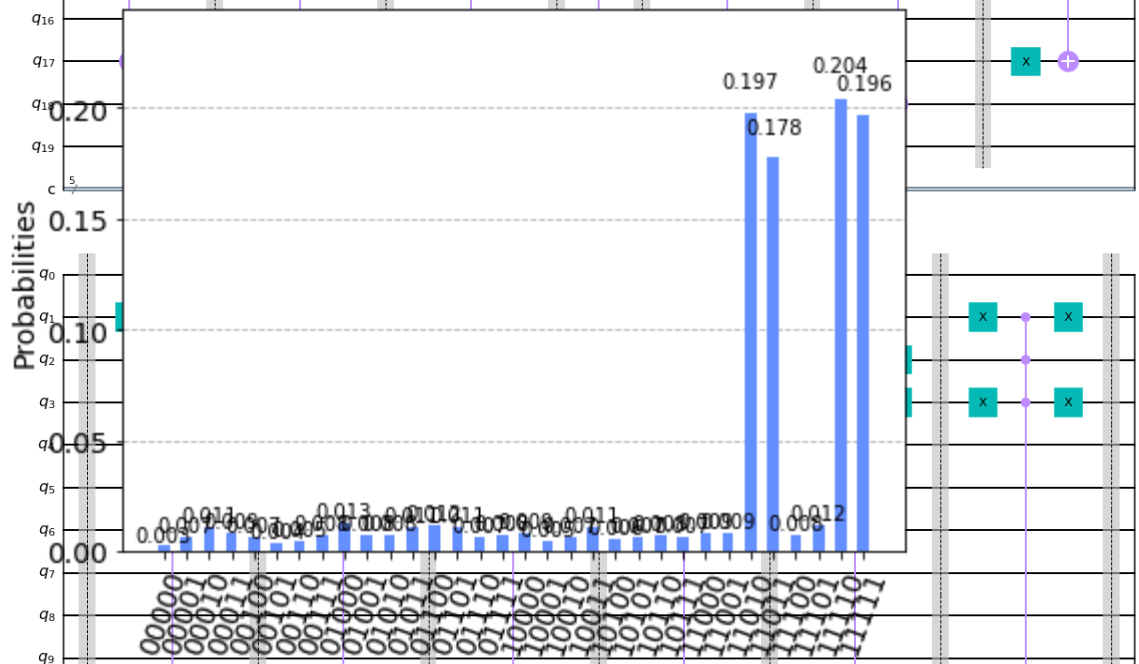
Out[1]:



In [2]:



Out[2]:



In [3]:

```
{'00111': 8, '00110': 5, '11001': 9, '00101': 4, '01001': 8, '01110': 7, '01000': 13, '01101': 11, '00001': 7, '11010': 202, '10101': 7, '10000': 9, '11100': 8, '11110': 209, '00010': 11, '11011': 182, '10110': 8, '11111': 201, '10010': 7, '01100': 12, '10100': 6, '11101': 12, '10011': 11, '01010': 8, '10001': 5, '10111': 7, '11000': 9, '00100': 7, '00011': 9, '01011': 11, '00000': 3, '01111': 8}
```

In []:



