In [1]:

```
# CNF equation:
# (x1 ∨ x2 ∨ ¬x3 ∨ x5)
#  ∧ (x6 ∨ ¬x7)
#  ∧ (x7 ∨ x8 ∨ x9 ∨ ¬x10)
#  ∧ (x1 ∨ x3 ∨ x5 ∨ x7 ∨ x9)
#  ∧ (x2 ∨ x4 ∨ x6 ∨ ¬x8 ∨ x10)
#  ∧ (¬x1 ∨ x2 ∨ x3 ∨ x4 ∨ ¬x5 ∨ x8 ∨ ¬x9 ∨ ¬x10)
#  ∧ (x2 ∨ ¬x3 ∨ x4 ∨ ¬x6 ∨ ¬x8)
#  ∧ (x3 ∨ x4 ∨ ¬x5 ∨ x7 ∨ ¬x8 ∨ x9 ∨ ¬x10)
#  ∧ (¬x1 ∨ x3 ∨ ¬x8 ∨ x9 ∨ x10)
#  ∧ (x2 ∨ ¬x4 ∨ x6 ∨ ¬x7 ∨ x8 ∨ ¬x10)

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(10, '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.append(HGate(),[q_reg[5]])
circuit.append(HGate(),[q_reg[6]])
circuit.append(HGate(),[q_reg[7]])
circuit.append(HGate(),[q_reg[8]])
circuit.append(HGate(),[q_reg[9]])
circuit.barrier(q_reg)
# Qslice 1
circuit.append(XGate(),[q_reg[0]])
circuit.append(XGate(),[q_reg[1]])
circuit.append(XGate(),[q_reg[4]])
# Qslice 2
circuit.append(XGate().control(4),[q_reg[0], q_reg[1], q_reg[2], q_reg[4],  q_re
g[10]])
# Qslice 3
circuit.append(XGate(),[q_reg[0]])
circuit.append(XGate(),[q_reg[1]])
circuit.append(XGate(),[q_reg[4]])
circuit.append(XGate(),[q_reg[10]])
circuit.barrier(q_reg)
```

```
# Qslice 4
circuit.append(XGate(),[q_reg[5]])
# Qslice 5
circuit.append(XGate().control(2),[q_reg[5], q_reg[6],  q_reg[11]])
# Qslice 6
circuit.append(XGate(),[q_reg[5]])
circuit.append(XGate(),[q_reg[11]])
circuit.barrier(q_reg)
# Qslice 7
circuit.append(XGate(),[q_reg[6]])
circuit.append(XGate(),[q_reg[7]])
circuit.append(XGate(),[q_reg[8]])
# Qslice 8
circuit.append(XGate().control(4),[q_reg[6], q_reg[7], q_reg[8], q_reg[9],  q_re
g[12]])
# Qslice 9
circuit.append(XGate(),[q_reg[6]])
circuit.append(XGate(),[q_reg[7]])
circuit.append(XGate(),[q_reg[8]])
circuit.append(XGate(),[q_reg[12]])
circuit.barrier(q_reg)
# Qslice 10
circuit.append(XGate(),[q_reg[0]])
circuit.append(XGate(),[q_reg[2]])
circuit.append(XGate(),[q_reg[4]])
circuit.append(XGate(),[q_reg[6]])
circuit.append(XGate(),[q_reg[8]])
# Qslice 11
circuit.append(XGate().control(5),[q_reg[0], q_reg[2], q_reg[4], q_reg[6], q_reg
[8],  q_reg[13]])
# 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[6]])
circuit.append(XGate(),[q_reg[8]])
circuit.append(XGate(),[q_reg[13]])
circuit.barrier(q_reg)
# Qslice 13
circuit.append(XGate(),[q_reg[1]])
circuit.append(XGate(),[q_reg[3]])
circuit.append(XGate(),[q_reg[5]])
circuit.append(XGate(),[q_reg[9]])
# Qslice 14
circuit.append(XGate().control(5),[q_reg[1], q_reg[3], q_reg[5], q_reg[7], q_reg
[9],  q_reg[14]])
# Qslice 15
circuit.append(XGate(),[q_reg[1]])
circuit.append(XGate(),[q_reg[3]])
circuit.append(XGate(),[q_reg[5]])
circuit.append(XGate(),[q_reg[9]])
circuit.append(XGate(),[q_reg[14]])
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[7]])
# Qslice 17
```

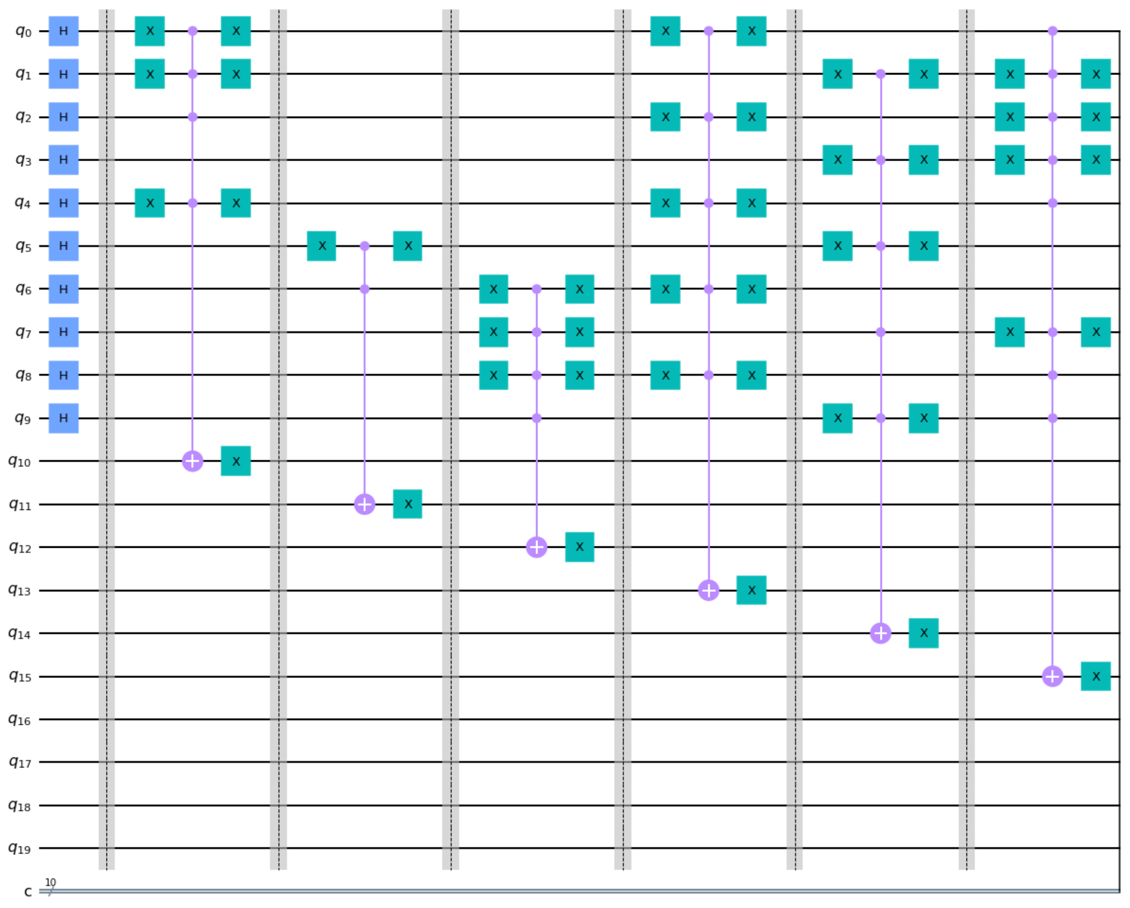
```
circuit.append(XGate().control(8),[q_reg[0], q_reg[1], q_reg[2], q_reg[3], q_reg
[4], q_reg[7], q_reg[8], q_reg[9],  q_reg[15]])
# Qslice 18
circuit.append(XGate(),[q_reg[1]])
circuit.append(XGate(),[q_reg[2]])
circuit.append(XGate(),[q_reg[3]])
circuit.append(XGate(),[q_reg[7]])
circuit.append(XGate(),[q_reg[15]])
circuit.barrier(q_reg)
# Qslice 19
circuit.append(XGate(),[q_reg[1]])
circuit.append(XGate(),[q_reg[3]])
# Qslice 20
circuit.append(XGate().control(5),[q_reg[1], q_reg[2], q_reg[3], q_reg[5], q_reg
[7],  q_reg[16]])
# Qslice 21
circuit.append(XGate(),[q_reg[1]])
circuit.append(XGate(),[q_reg[3]])
circuit.append(XGate(),[q_reg[16]])
circuit.barrier(q_reg)
# Qslice 22
circuit.append(XGate(),[q_reg[2]])
circuit.append(XGate(),[q_reg[3]])
circuit.append(XGate(),[q_reg[6]])
circuit.append(XGate(),[q_reg[8]])
# Qslice 23
circuit.append(XGate().control(7),[q_reg[2], q_reg[3], q_reg[4], q_reg[6], q_reg
[7], q_reg[8], q_reg[9],  q_reg[17]])
# Qslice 24
circuit.append(XGate(),[q_reg[2]])
circuit.append(XGate(),[q_reg[3]])
circuit.append(XGate(),[q_reg[6]])
circuit.append(XGate(),[q_reg[8]])
circuit.append(XGate(),[q_reg[17]])
circuit.barrier(q_reg)
# Qslice 25
circuit.append(XGate(),[q_reg[2]])
circuit.append(XGate(),[q_reg[7]])
circuit.append(XGate(),[q_reg[8]])
circuit.append(XGate(),[q_reg[9]])
# Qslice 26
circuit.append(XGate().control(5),[q_reg[0], q_reg[2], q_reg[7], q_reg[8], q_reg
[9],  q_reg[18]])
# Qslice 27
circuit.append(XGate(),[q_reg[2]])
circuit.append(XGate(),[q_reg[7]])
circuit.append(XGate(),[q_reg[8]])
circuit.append(XGate(),[q_reg[9]])
circuit.append(XGate(),[q_reg[18]])
circuit.barrier(q_reg)
# Qslice 28
circuit.append(XGate(),[q_reg[1]])
circuit.append(XGate(),[q_reg[5]])
circuit.append(XGate(),[q_reg[7]])
# Qslice 29
circuit.append(XGate().control(6),[q_reg[1], q_reg[3], q_reg[5], q_reg[6], q_reg
[7], q_reg[9],  q_reg[19]])
# Qslice 30
circuit.append(XGate(),[q_reg[1]])
circuit.append(XGate(),[q_reg[5]])
circuit.append(XGate(),[q_reg[7]])
circuit.append(XGate(),[q_reg[19]])
circuit.barrier(q_reg)
# Qslice 31
circuit.append(ZGate().control(9),[q_reg[10], q_reg[11], q_reg[12], q_reg[13], q_reg[14], q_reg[15], q_reg[16], q_reg[17], q_reg[18],  q_reg[19]])
circuit.barrier(q_reg)
# Qslice 30
circuit.append(XGate(),[q_reg[1]])
circuit.append(XGate(),[q_reg[5]])
circuit.append(XGate(),[q_reg[7]])
circuit.append(XGate(),[q_reg[19]])
# Qslice 29
circuit.append(XGate().control(6),[q_reg[1], q_reg[3], q_reg[5], q_reg[6], q_reg[7], q_reg[9],  q_reg[19]])
# Qslice 28
circuit.append(XGate(),[q_reg[1]])
circuit.append(XGate(),[q_reg[5]])
circuit.append(XGate(),[q_reg[7]])
circuit.barrier(q_reg)
# Qslice 27
circuit.append(XGate(),[q_reg[2]])
circuit.append(XGate(),[q_reg[7]])
circuit.append(XGate(),[q_reg[8]])
circuit.append(XGate(),[q_reg[9]])
circuit.append(XGate(),[q_reg[18]])
# Qslice 26
circuit.append(XGate().control(5),[q_reg[0], q_reg[2], q_reg[7], q_reg[8], q_reg[9],  q_reg[18]])
# Qslice 25
circuit.append(XGate(),[q_reg[2]])
circuit.append(XGate(),[q_reg[7]])
circuit.append(XGate(),[q_reg[8]])
circuit.append(XGate(),[q_reg[9]])
circuit.barrier(q_reg)
# Qslice 24
circuit.append(XGate(),[q_reg[2]])
circuit.append(XGate(),[q_reg[3]])
circuit.append(XGate(),[q_reg[6]])
circuit.append(XGate(),[q_reg[8]])
circuit.append(XGate(),[q_reg[17]])
# Qslice 23
circuit.append(XGate().control(7),[q_reg[2], q_reg[3], q_reg[4], q_reg[6], q_reg[7], q_reg[8], q_reg[9],  q_reg[17]])
# Qslice 22
circuit.append(XGate(),[q_reg[2]])
circuit.append(XGate(),[q_reg[3]])
circuit.append(XGate(),[q_reg[6]])
circuit.append(XGate(),[q_reg[8]])
circuit.barrier(q_reg)
# Qslice 21
circuit.append(XGate(),[q_reg[1]])
circuit.append(XGate(),[q_reg[3]])
circuit.append(XGate(),[q_reg[16]])
# Qslice 20
circuit.append(XGate().control(5),[q_reg[1], q_reg[2], q_reg[3], q_reg[5], q_reg[7],  q_reg[16]])
# 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[7]])
circuit.append(XGate(),[q_reg[15]])
# Qslice 17
circuit.append(XGate().control(8),[q_reg[0], q_reg[1], q_reg[2], q_reg[3], q_reg
[4], q_reg[7], q_reg[8], q_reg[9],  q_reg[15]])
# 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[7]])
circuit.barrier(q_reg)
# Qslice 15
circuit.append(XGate(),[q_reg[1]])
circuit.append(XGate(),[q_reg[3]])
circuit.append(XGate(),[q_reg[5]])
circuit.append(XGate(),[q_reg[9]])
circuit.append(XGate(),[q_reg[14]])
# Qslice 14
circuit.append(XGate().control(5),[q_reg[1], q_reg[3], q_reg[5], q_reg[7], q_reg
[9],  q_reg[14]])
# Qslice 13
circuit.append(XGate(),[q_reg[1]])
circuit.append(XGate(),[q_reg[3]])
circuit.append(XGate(),[q_reg[5]])
circuit.append(XGate(),[q_reg[9]])
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[6]])
circuit.append(XGate(),[q_reg[8]])
circuit.append(XGate(),[q_reg[13]])
# Qslice 11
circuit.append(XGate().control(5),[q_reg[0], q_reg[2], q_reg[4], q_reg[6], q_reg
[8],  q_reg[13]])
# Qslice 10
circuit.append(XGate(),[q_reg[0]])
circuit.append(XGate(),[q_reg[2]])
circuit.append(XGate(),[q_reg[4]])
circuit.append(XGate(),[q_reg[6]])
circuit.append(XGate(),[q_reg[8]])
circuit.barrier(q_reg)
# Qslice 9
circuit.append(XGate(),[q_reg[6]])
circuit.append(XGate(),[q_reg[7]])
circuit.append(XGate(),[q_reg[8]])
circuit.append(XGate(),[q_reg[12]])
# Qslice 8
circuit.append(XGate().control(4),[q_reg[6], q_reg[7], q_reg[8], q_reg[9],  q_re
g[12]])
# Qslice 7
```

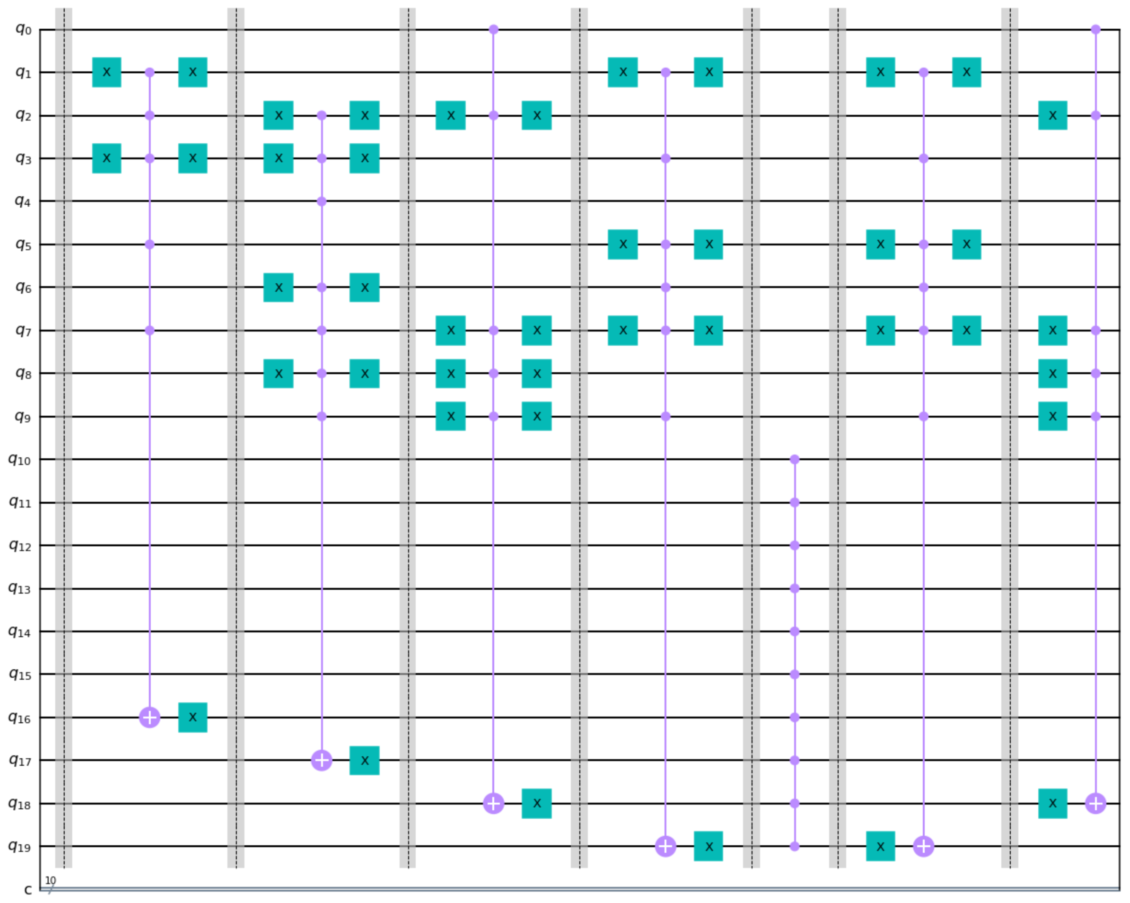
```
circuit.append(XGate(),[q_reg[6]])
circuit.append(XGate(),[q_reg[7]])
circuit.append(XGate(),[q_reg[8]])
circuit.barrier(q_reg)
# Qslice 6
circuit.append(XGate(),[q_reg[5]])
circuit.append(XGate(),[q_reg[11]])
# Qslice 5
circuit.append(XGate().control(2),[q_reg[5], q_reg[6],  q_reg[11]])
# Qslice 4
circuit.append(XGate(),[q_reg[5]])
circuit.barrier(q_reg)
# Qslice 3
circuit.append(XGate(),[q_reg[0]])
circuit.append(XGate(),[q_reg[1]])
circuit.append(XGate(),[q_reg[4]])
circuit.append(XGate(),[q_reg[10]])
# Qslice 2
circuit.append(XGate().control(4),[q_reg[0], q_reg[1], q_reg[2], q_reg[4],  q_re
g[10]])
# Qslice 1
circuit.append(XGate(),[q_reg[0]])
circuit.append(XGate(),[q_reg[1]])
circuit.append(XGate(),[q_reg[4]])
circuit.barrier(q_reg)
# Qslice 32
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.append(HGate(),[q_reg[5]])
circuit.append(HGate(),[q_reg[6]])
circuit.append(HGate(),[q_reg[7]])
circuit.append(HGate(),[q_reg[8]])
circuit.append(HGate(),[q_reg[9]])
# Qslice 33
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]])
circuit.append(XGate(),[q_reg[5]])
circuit.append(XGate(),[q_reg[6]])
circuit.append(XGate(),[q_reg[7]])
circuit.append(XGate(),[q_reg[8]])
circuit.append(XGate(),[q_reg[9]])
# Qslice 34
circuit.append(ZGate().control(9),[q_reg[0], q_reg[1], q_reg[2], q_reg[3], q_reg
[4], q_reg[5], q_reg[6], q_reg[7], q_reg[8],  q_reg[9]])
# Qslice 33
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]])
circuit.append(XGate(),[q_reg[5]])
circuit.append(XGate(),[q_reg[6]])
circuit.append(XGate(),[q_reg[7]])
circuit.append(XGate(),[q_reg[8]])
circuit.append(XGate(),[q_reg[9]])
# Qslice 36
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.append(HGate(),[q_reg[5]])
circuit.append(HGate(),[q_reg[6]])
circuit.append(HGate(),[q_reg[7]])
circuit.append(HGate(),[q_reg[8]])
circuit.append(HGate(),[q_reg[9]])
circuit.barrier(q_reg)
# Qslice 37
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.measure(q_reg[5], c_reg[5])
circuit.measure(q_reg[6], c_reg[6])
circuit.measure(q_reg[7], c_reg[7])
circuit.measure(q_reg[8], c_reg[8])
circuit.measure(q_reg[9], c_reg[9])

circuit.draw('mpl')
```

```
<frozen importlib._bootstrap>:219: RuntimeWarning: scipy._lib.messag
estream.MessageStream size changed, may indicate binary incompatibil
ity. Expected 56 from C header, got 64 from PyObject
```

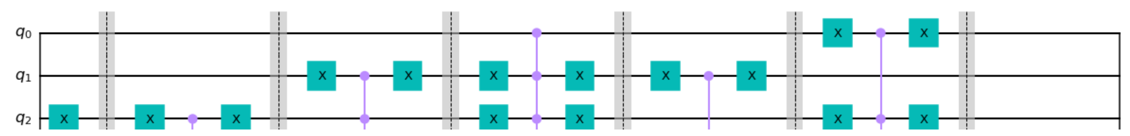
Out[1]:



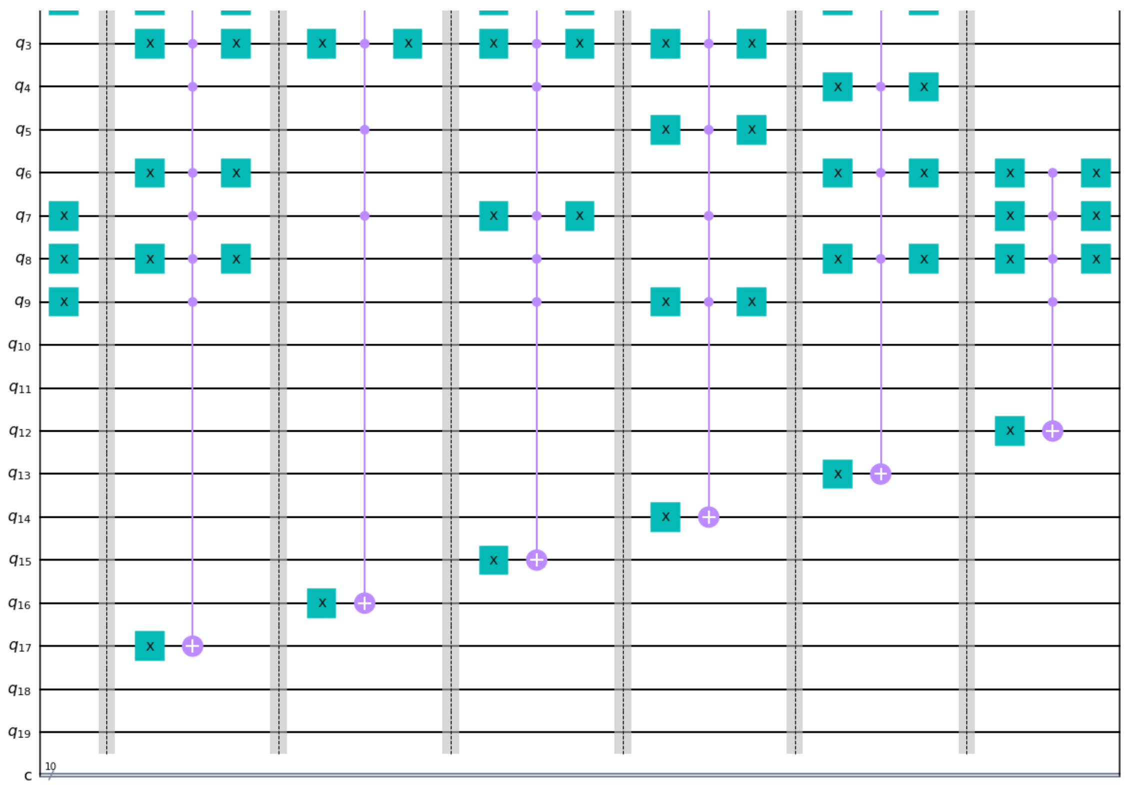
q0
q1
q2
q3
q4
q5
q6
q7
q8
q9
q10
q11
q12
q13
q14
q15
q16
q17
q18
q19
c
10
H
X



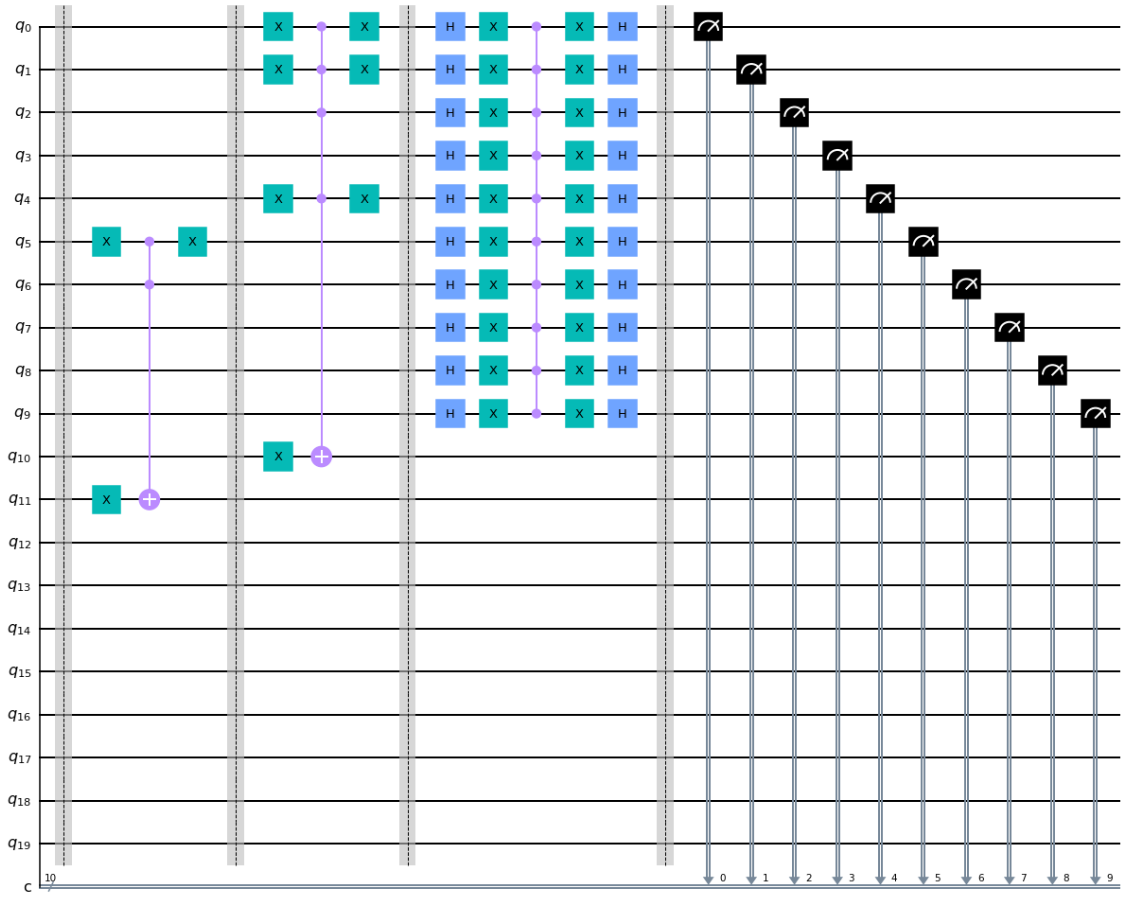
q0
q1
q2
q3
q4
q5
q6
q7
q8
q9
q10
q11
q12
q13
q14
q15
q16
q17
q18
q19
c
10
X



q0
q1
q2
X



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}
q_{16}
q_{17}
q_{18}
q_{19}
c
10
X

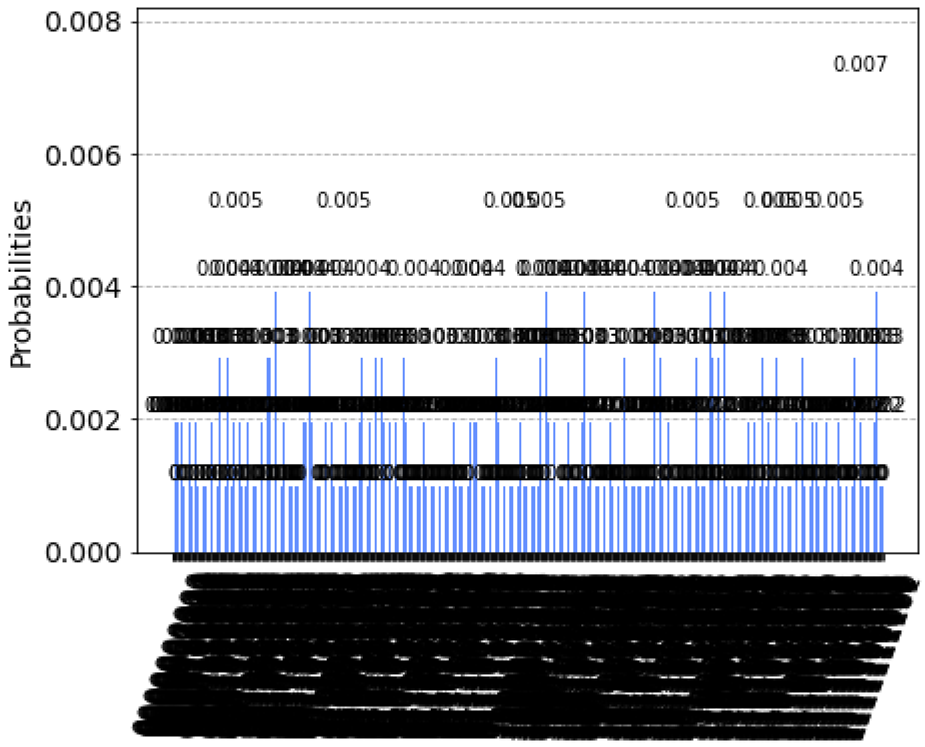


q_0
q_1
q_2
q_3
q_4
q_5
q_6
q_7
q_8
q_9
q_{10}
q_{11}
q_{12}
q_{13}
q_{14}
q_{15}
q_{16}
q_{17}
q_{18}
q_{19}
c
10
X
H
0
1
2
3
4
5
6
7
8
9

In [2]:

```
from qiskit import Aer
from qiskit.compiler import transpile
from qiskit.visualization import plot_histogram

# Use Aer's qasm_simulator
backend_sim = Aer.get_backend('qasm_simulator')

# Execute the circuit on the qasm simulator.
# We've set the number of repeats of the circuit
# to be 1024, which is the default.
job_sim = backend_sim.run(transpile(circuit, backend_sim), shots=1024)

# Grab the results from the job.
result_sim = job_sim.result()
counts = result_sim.get_counts(circuit)
plot_histogram(counts)
```

Out[2]:



In [3]:

```
print(counts)
```

```
{'0101011010': 1, '1010010010': 1, '0110010100': 1, '1101001100': 1,
'1010000000': 1, '1101001111': 1, '1100100000': 1, '0110011100': 1,
'1011101101': 1, '0010011010': 1, '0010110101': 1, '0110100111': 1,
'0100011010': 1, '0110011101': 1, '0111011000': 1, '1111100011': 1,
'1111100110': 1, '1000001010': 1, '1011101110': 1, '1001011111': 1,
'1111111100': 1, '0101001000': 2, '1101000110': 1, '1001101000': 1,
'1100110011': 1, '1011001101': 1, '1000001100': 1, '0011111100': 2,
'0001000010': 1, '1010101010': 1, '0100101110': 1, '0001101000': 1,
'1110110001': 1, '0011001001': 2, '1111111101': 1, '0001001001': 1,
'0100001001': 1, '0111111100': 1, '0001100101': 1, '0001010010': 2,
'0111110100': 1, '0111000100': 1, '0111100110': 1, '0111000111': 2,
'0000100010': 1, '0111110000': 1, '1000000011': 1, '1100101111': 1,
'0110100100': 1, '1101111001': 1, '0111001000': 2, '1010010000': 1,
'0100001101': 1, '0001100100': 1, '1000111100': 1, '0100110011': 2,
'1010100100': 1, '1101101101': 1, '0100110010': 1, '0111011011': 1,
'0111010101': 1, '0010001010': 1, '0001101001': 3, '0011111010': 1,
'1001001001': 1, '0101101101': 1, '0001100001': 2, '1001110101': 1,
'1011110010': 1, '1010000010': 1, '0110111101': 1, '0110010101': 2,
'1101100110': 1, '1001110110': 1, '1101111100': 1, '1011001100': 1,
'0011100111': 2, '1101100000': 1, '0000110010': 1, '1000001000': 2,
'1001110011': 2, '1010110110': 1, '0001001101': 1, '0110110010': 1,
'0000001110': 1, '1000110101': 2, '1001011110': 1, '1100011001': 2,
'1101101110': 1, '1010101101': 1, '0011010111': 1, '0101100001': 1,
'0011110100': 2, '0100110111': 1, '0111011101': 2, '0110101001': 1,
'0001011101': 2, '0010111000': 1, '0000110111': 2, '0101011000': 2,
'1101001011': 2, '1101011010': 1, '1001011100': 4, '0011010100': 2,
'1111111010': 1, '1001011010': 3, '0001011011': 1, '0100100100': 3,
'1011000101': 1, '1010000001': 1, '1010100110': 1, '0000001000': 2,
'0100000101': 2, '0001111101': 1, '0010101110': 1, '0111101101': 1,
'1100010111': 1, '0101110101': 1, '1001001110': 1, '0000001111': 1,
'0011010110': 3, '1010010111': 1, '0111110010': 2, '0111110110': 1,
'1101110001': 1, '0100100000': 2, '1101100111': 1, '1001001010': 2,
'0100101011': 1, '0001111000': 1, '1010000101': 1, '1111001001': 1,
'1110010001': 2, '0011011011': 1, '1000100001': 2, '0100101001': 1,
'1000011110': 4, '0101101100': 1, '0011101000': 1, '1110000010': 1,
'1001010010': 2, '0110001111': 1, '1001110000': 1, '1101010011': 1,
'1100100100': 1, '1111011001': 2, '0001010100': 1, '0111001101': 4,
'1111010000': 1, '1010100011': 1, '0111010011': 1, '1100100001': 1,
'0101011011': 1, '1011000111': 1, '0111000010': 1, '1100110101': 1,
'1110001110': 2, '0000101101': 2, '1110101010': 2, '1001001011': 3,
'1011001110': 2, '1100101001': 1, '0000100100': 2, '0110010001': 1,
'0111010100': 1, '0011010001': 4, '1101001110': 1, '1111101000': 3,
'1000101011': 2, '0110110011': 1, '1111000000': 2, '0101010100': 3,
'1111010100': 1, '0001010011': 2, '0100011110': 1, '0110000100': 1,
'0111001010': 3, '1111010010': 1, '0100100001': 1, '0101101011': 1,
'1000000100': 3, '0011000001': 1, '1101100100': 1, '0011010010': 2,
'1011111010': 3, '0100100110': 4, '1011100000': 1, '0000001011': 2,
'0000000011': 1, '1010111001': 1, '0001001110': 1, '1100001100': 4,
'1000111000': 3, '0011110101': 2, '1001010100': 1, '1011010011': 4,
'0101000011': 3, '1111011101': 7, '1100111001': 2, '0000011010': 1,
'1100001101': 1, '1111010011': 1, '0000000110': 2, '0011001011': 2,
'1010001010': 2, '1001000000': 1, '0001110001': 2, '1010110011': 3,
'1111111000': 1, '1001011000': 2, '0111100000': 1, '0100011000': 1,
'1000101101': 2, '0101110011': 2, '1001010101': 2, '0110011010': 2,
'1010111000': 2, '1111010111': 3, '0011101001': 1, '1010101100': 4,
'0000110100': 2, '1111001000': 1, '1000111110': 1, '1001011101': 1,
'1100000011': 1, '0011001000': 2, '0001010101': 1, '1010001001': 1,
'0001100010': 1, '0000001010': 3, '1101000001': 2, '0111011010': 1,
'0101001101': 2, '0111001111': 2, '0110000000': 2, '0110110100': 2,
'0100000010': 1, '1010000011': 2, '0000010110': 2, '0010001101': 3,
'0010000101': 1, '1100111011': 2, '1110010101': 1, '1110100011': 1,
'1111101110': 1, '1111000111': 1, '1101011011': 3, '1011011110': 1,
'0011110111': 1, '0011111011': 1, '0111110101': 5, '1110110110': 1,
'0101001010': 1, '1101110000': 1, '1000011011': 3, '0010011101': 2,
'1000010111': 4, '1001001000': 1, '0100011111': 1, '1111011011': 2,
'0000111101': 1, '1111100101': 1, '1000100111': 3, '0000101100': 3,
'1010101111': 2, '1001000001': 2, '1101111010': 2, '0101000000': 1,
'1100111010': 1, '1111111001': 2, '0000000001': 1, '1110110101': 2,
'1100111000': 1, '0110001100': 3, '1001000111': 1, '1111000101': 2,
'1101010010': 5, '1010100001': 1, '0101001110': 1, '1011011111': 1,
'0001011010': 5, '0011011111': 2, '1000010011': 1, '0000000000': 2,
'1011000110': 2, '0101010110': 2, '0111101111': 1, '0111101001': 1,
'1110011110': 2, '0111000011': 4, '1001110100': 1, '0010001100': 3,
'0101010101': 2, '0111011001': 1, '1110001001': 1, '1110101011': 3,
'1011010001': 1, '1001000100': 2, '1010001000': 4, '1010100010': 3,
'1011110111': 4, '1101000010': 1, '0101001001': 3, '1100010100': 1,
'1101000100': 3, '1010101001': 1, '0111000101': 2, '1001010111': 1,
'1001000110': 1, '1110111100': 1, '0100110100': 2, '0011010011': 2,
'1100001011': 1, '0110101000': 2, '0101100101': 1, '1001101011': 1,
'1000001111': 1, '1111011111': 1, '1010010100': 1, '0010100001': 1,
'1110000111': 1, '0000110101': 1, '0010110100': 1, '0101111010': 1,
'0111000001': 2, '1000010110': 3, '1001011001': 3, '0001111100': 1,
'1000001011': 1, '1100001110': 1, '1000011000': 1, '0111111011': 1,
'0111101100': 1, '1000011111': 2, '0101011100': 2, '0000011000': 2,
'0111010110': 2, '0100001000': 1, '0001011100': 1, '0001000100': 3,
'0001010001': 3, '1000101111': 2, '0111000110': 2, '0010101001': 4,
'0101110111': 1, '0100011011': 1, '1011010000': 1, '1011110000': 1,
'0010100010': 1, '1111001100': 1, '0000101010': 1, '0011001010': 2,
'1010000100': 2, '0111111101': 1, '1010001110': 2, '0000011110': 2,
'0000000111': 2, '1111100111': 1, '0110000010': 2, '1101010110': 1,
'0010001111': 1, '0111001100': 1, '0011101100': 2, '1011010100': 1,
'0010001000': 1, '0011000110': 1, '1001001100': 2, '1010110000': 3,
'0110101010': 1, '1111010001': 1, '1111110101': 4, '1111001011': 1,
'1000000111': 3, '1011010010': 1, '1100010101': 4, '0110111100': 2,
'1101101111': 2, '1010000110': 3, '1111110010': 2, '0010000000': 2,
'0110011111': 1, '1011010111': 2, '1000000010': 2, '1011011010': 3,
'0011111110': 1, '0110001000': 1, '0011111111': 1, '1000100010': 1,
'0111001110': 3, '0101000110': 2, '1100010001': 3, '1100101110': 2,
'0000111001': 3, '1000000101': 1, '1100000111': 1, '0011110011': 2,
'0000110011': 2, '0100101111': 1, '0101011001': 1, '0101101110': 4,
'0001101011': 2, '1111011100': 2, '1111000110': 2, '1111001010': 2,
'0111111010': 2, '1111011010': 1, '1010100000': 1, '1011110001': 2,
'0101010111': 2, '0000100000': 1, '1011000011': 3, '1101000000': 2,
'1011001000': 4, '1011101100': 2, '1001010011': 4, '0001110000': 2,
'1010101011': 2, '0011001101': 2, '0101101001': 1, '0001011001': 4,
'0010010000': 1, '1111011000': 2, '0011101111': 3, '1101000111': 1,
'1001010110': 1, '0010001001': 3, '0010000001': 2, '0101100000': 3,
'0010101010': 2, '0110111010': 1, '1101010111': 1, '0110101101': 2,
'1001101101': 1, '1101110011': 2, '1111000100': 5, '0011111101': 3,
'1101111101': 3, '0110000011': 1, '1110110011': 1, '1000000000': 3,
'1110111101': 1, '1100111100': 1, '0100000000': 2, '1010010001': 1,
'1011111111': 1, '1101000011': 3, '0100001111': 1, '1111011110': 1,
'0101010010': 2, '1010110111': 1, '0000110001': 1, '0000001001': 1,
'0111100010': 2, '1001101001': 1, '1110011011': 1, '1110101100': 1,
'0001111010': 2, '0011110000': 4, '0101111110': 1, '1001110010': 1,
'1010110101': 1, '1001000010': 4, '1111000001': 2, '1101110101': 1,
'1111110100': 3, '0000110000': 1, '1100010000': 3, '1010011100': 1,
'0101010011': 2, '1001100100': 4, '1000111001': 1, '1110111110': 1,
'1011100110': 1, '0000111000': 2, '1101100011': 3, '0001000111': 1,
'0010111101': 1, '0000100001': 2, '1001101100': 1, '0100010101': 1,
'0111001011': 2, '0111100011': 1, '1100000001': 1, '0001000101': 3,
'0110011110': 1, '0011011001': 3, '0001000011': 1, '1010111011': 1,
'1000101010': 3, '0010000010': 1, '1011000100': 1, '1001000101': 2,
'1100000100': 3, '1001010001': 2, '0010111001': 1, '0001101110': 1,
'1010100101': 1, '0000100011': 2, '0101000111': 3, '0000011011': 2,
'0111010010': 2, '1000110001': 2, '0010010101': 4, '0001111110': 2,
'1110111011': 1, '1011001001': 2, '1110111111': 1, '1011010110': 2,
'1001100110': 1, '1111100001': 2, '1000110110': 1, '1010110001': 2,
'0010101100': 1, '1100110010': 1, '1110000000': 1, '0101011110': 1,
'1001100011': 1, '0011000111': 2, '1000001001': 1, '0011010000': 2,
'0001111001': 1, '1100001001': 1, '0000011001': 3, '1101010100': 2,
'1101001000': 3, '1011001010': 1, '0101001011': 2, '1011100010': 1,
'0010100101': 1, '0101000100': 1, '0001001100': 2, '0111111001': 1,
'1001110111': 1, '0110010011': 1, '0111111110': 2, '1000110011': 2,
'1110101000': 2, '1000111011': 2, '1011010101': 5, '0010111100': 1,
'0011100101': 1, '1011111001': 1, '1000011100': 1, '0111111000': 2,
'0110001101': 1, '0111000000': 2, '0110111000': 1, '1011000001': 2,
'1101010101': 3, '0010101011': 1, '0100000100': 5, '0111010111': 1,
'1010010110': 2, '1010001100': 1, '1011000010': 2, '1101011000': 3,
'0111011110': 3, '1101011001': 3, '0001110011': 1, '1000001101': 1,
'1110001000': 1, '0010110011': 2, '1010011111': 1, '1100000110': 1,
'1011001011': 1, '1101011101': 4, '0011100001': 1, '0111001001': 2,
'0101001100': 3, '0101111000': 1, '0100000111': 2, '1010101000': 2,
'0110110111': 1, '0101100100': 2, '0000101000': 1, '1101010000': 2,
'1110000011': 1, '1000010001': 2, '1101011110': 1, '1110001011': 1,
'0110100101': 3, '1111001101': 1, '0001101100': 2, '0001001011': 4,
'1011011001': 1, '0110110110': 1, '0110110101': 1, '0100111001': 1,
'1010011110': 1, '1000010100': 5, '0101100011': 2, '0101111111': 1,
'0000101001': 3, '0011000011': 4, '0111010000': 1, '0010010100': 1,
'1001111001': 1, '0111100101': 2, '1011000000': 2, '1100101100': 1,
'0011000010': 1, '1111100100': 3, '0100001100': 1, '0101001111': 1,
'1101001010': 3, '1100000000': 1, '0001010111': 2, '1101101100': 5,
'0000110110': 1, '1100011000': 1, '1110100001': 1}
```

In [ ]: