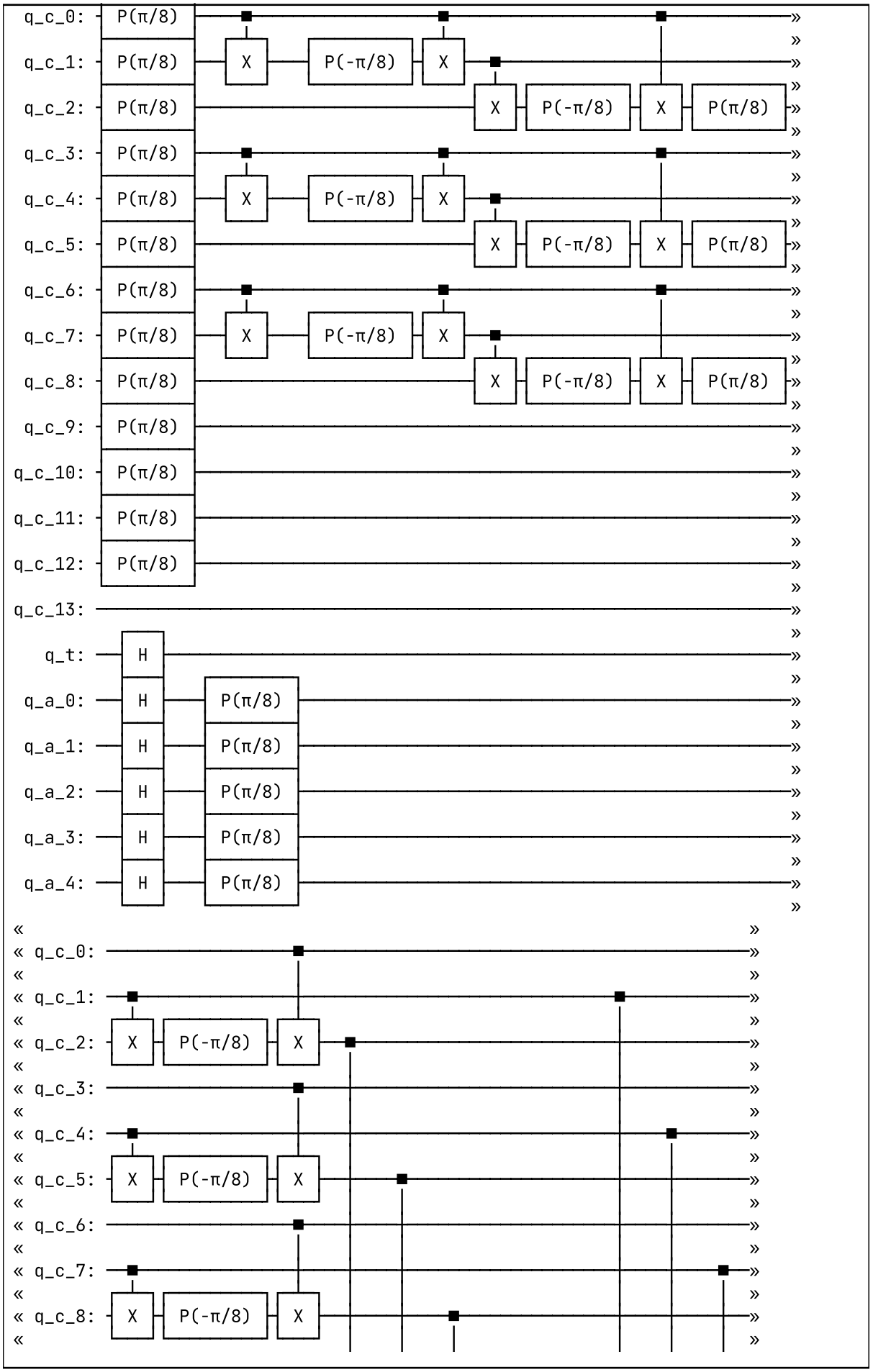
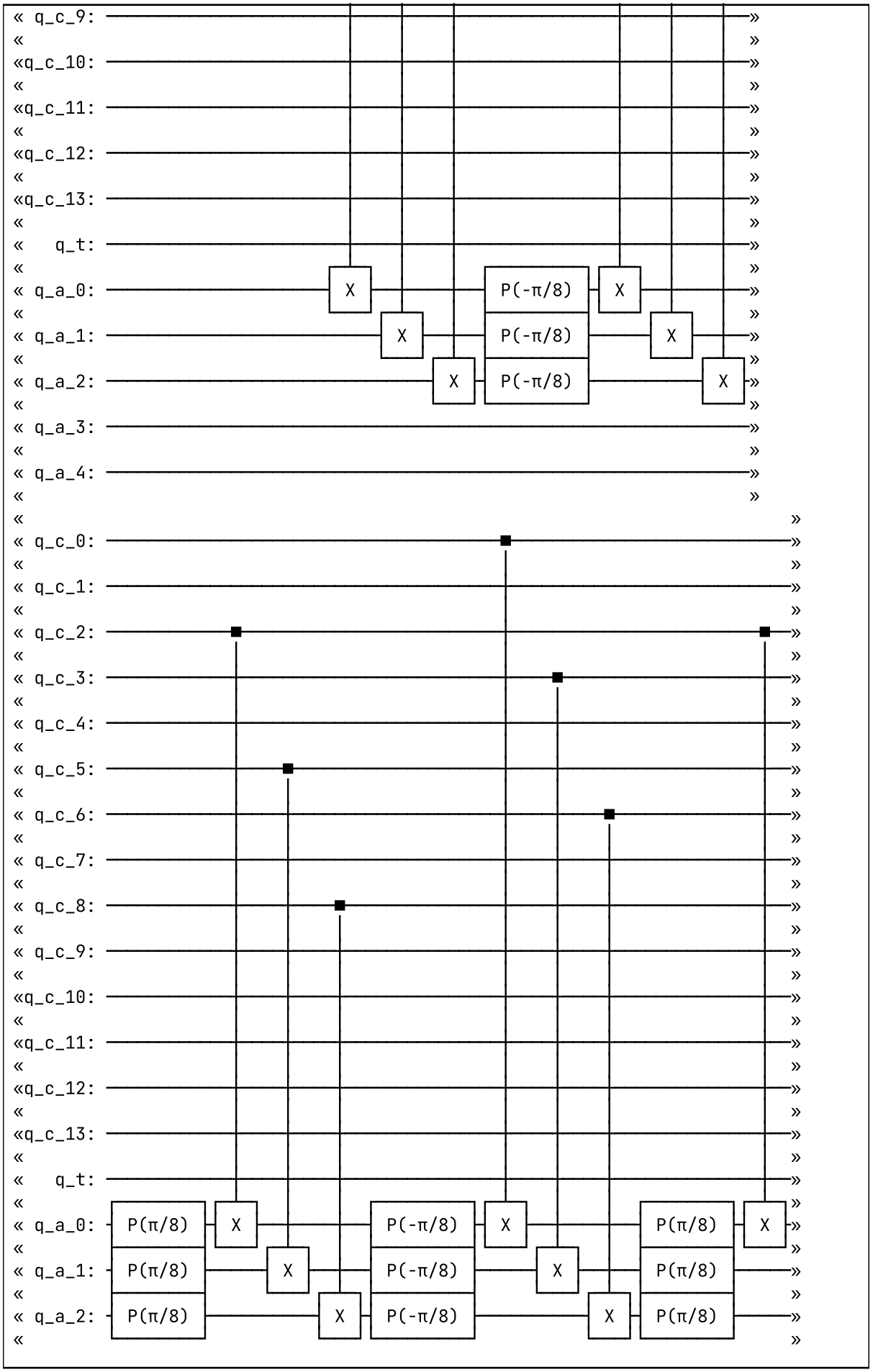
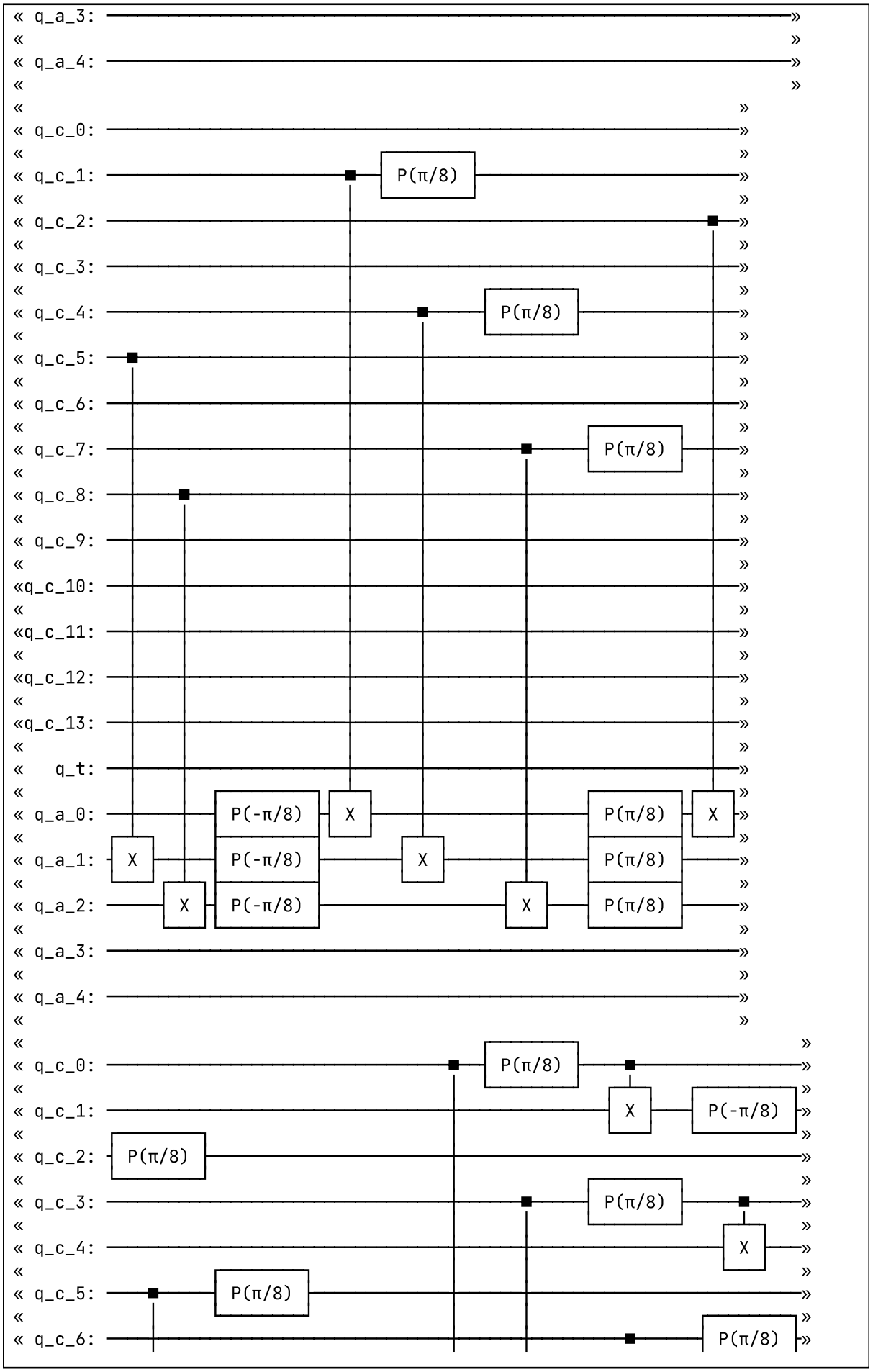
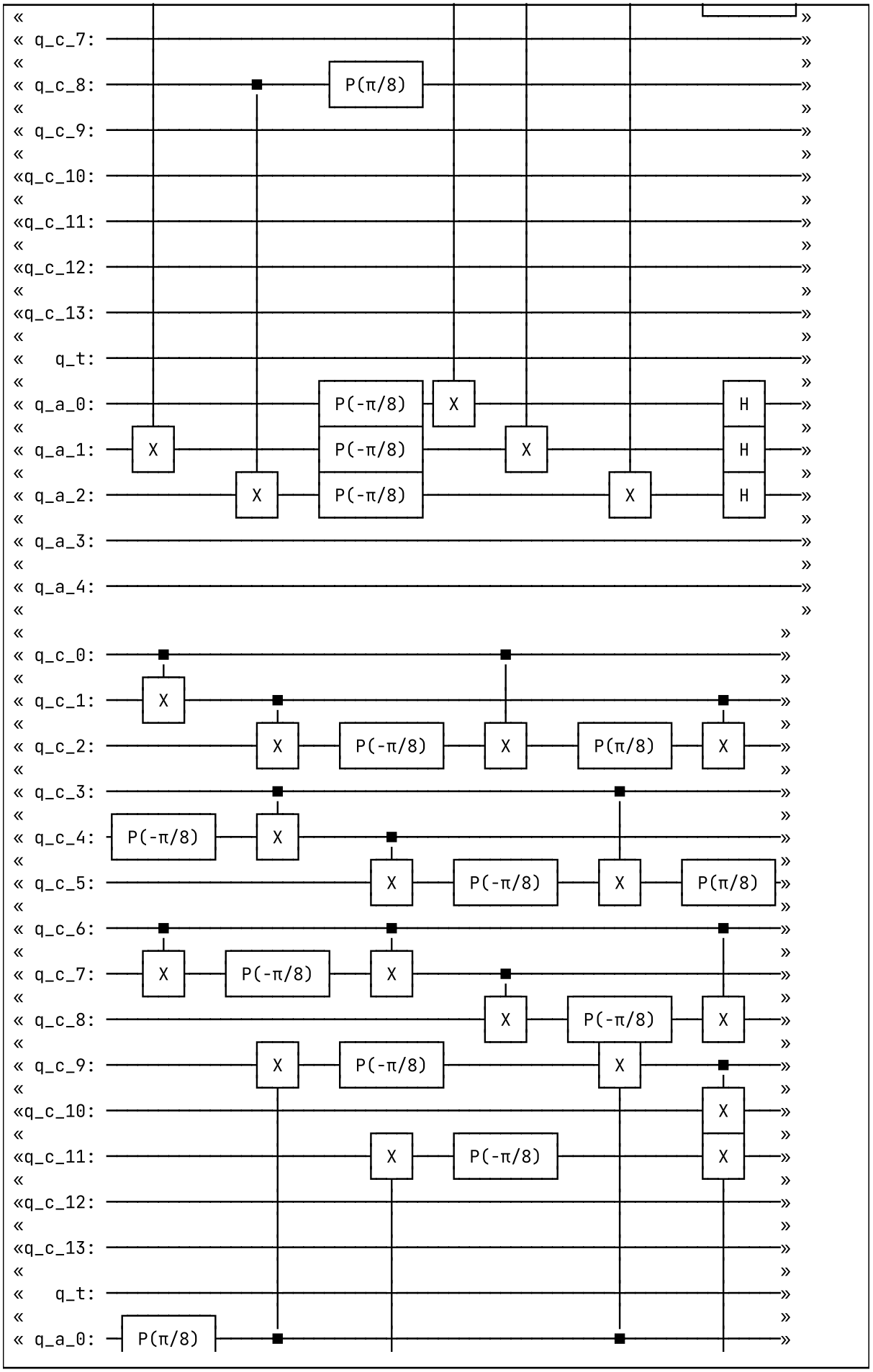
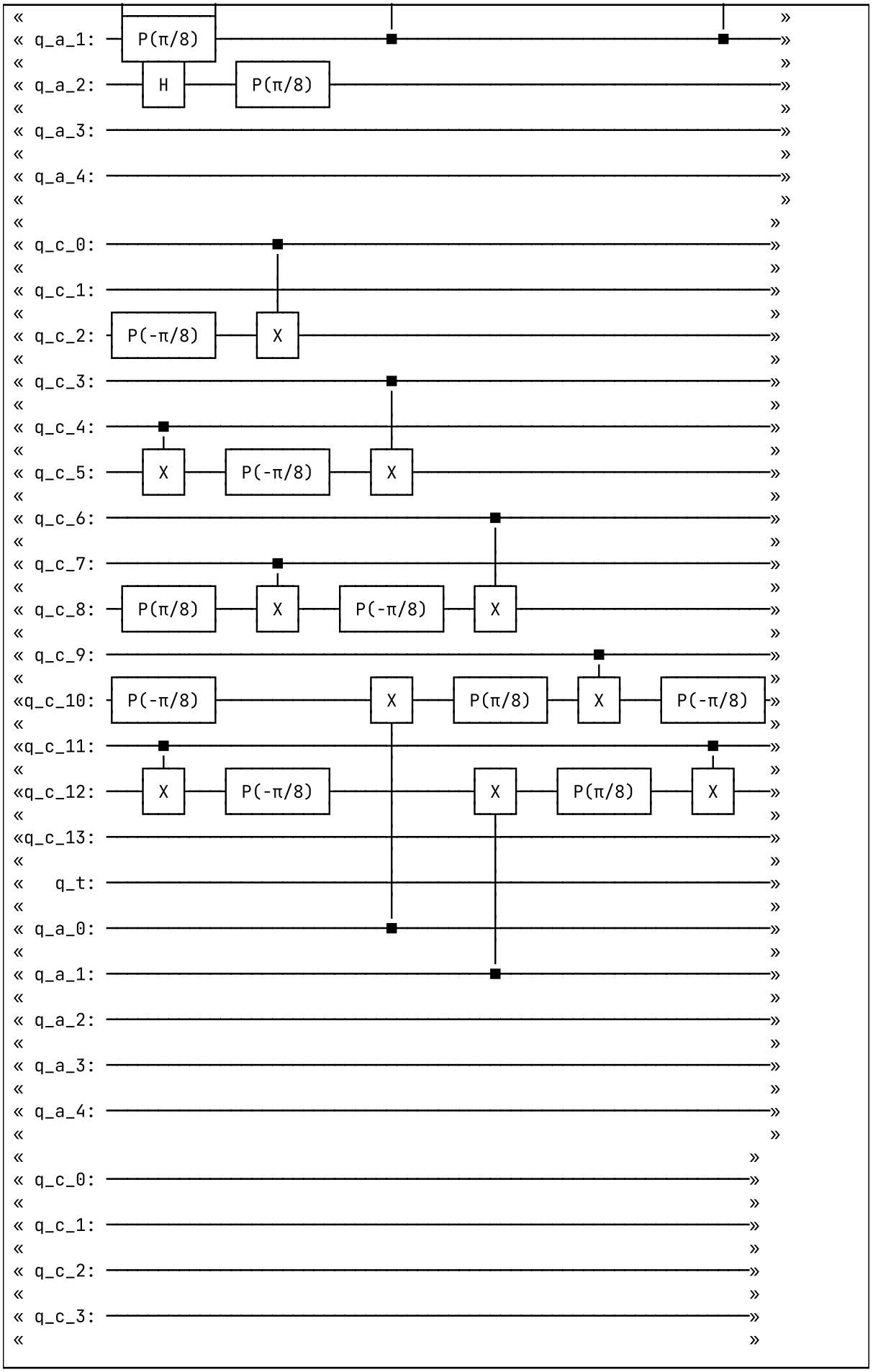
```
C:\PycharmProjects\ClassiqChallenge\venv\Scripts\python.exe C:/PycharmProjects/
ClassiqChallenge/toffoli/tofolli_competition.py
------------------------------------------------------------
|            The CLASSIQ Coding Competition                |
|                   The Challenge :                        |
|      DECOMPOSING A MULTI-CONTROLLED TOFFOLI GATE         |
|           Submitted by : Ron Cohen                       |
------------------------------------------------------------

------------------------------------------------------------
|                        Solution:                         |

Init Circuit - 14 Controls, 1 Target, 5 Ancilla

We want to maximize the usage of ancilla
First we use the first 3 ancilla with C3X gate
which cost only 27 depth (gray code):
C3X c0,c1,c2 -> a0
C3X c3,c4,c5 -> a1
C3X c6,c7,c8 -> a2

Than again C3X which is only 27 depth from those (2 ancilla)
+(left 4 controls) to the remaining ancilla:
C3X a0,c9 ,c10 -> a3
C3X a1,c11,c12 -> a4

Than C2X which is only 11 depth from those 2 ancilla to target:
C2X a3,a4 -> t

Now only need to return ancilla with performing again in
reverse order:
C3X a1,c11,c12 -> a4
C3X a0,c9 ,c10 -> a3
And (order here doesn't matter):
C3X c6,c7,c8 -> a2
C3X c3,c4,c5 -> a1
C3X c0,c1,c2 -> a0
|__________________________________________________________|

------------------------------------------------------------
|                      Compiling to QASM :                 |
QASM file: attached toffoli.qasm
|__________________________________________________________|

------------------------------------------------------------
|                Depth Results from QASM file:             |
 Total depth in QASM is:  99

 With only gates of :
 OrderedDict([('p', 150), ('cx', 146), ('h', 22), ('t', 4), ('tdg', 3)])

Depth calculation : # 2*depth(C3X)*2 + depth(C2X) - Overlaps  =
                    # 2*(   27   )*2 + (   11   ) - (0+1+9+10)=
                    # 99
|__________________________________________________________|

------------------------------------------------------------
|                         Circuit:                         |
        ┌─────────┐                                                »
```

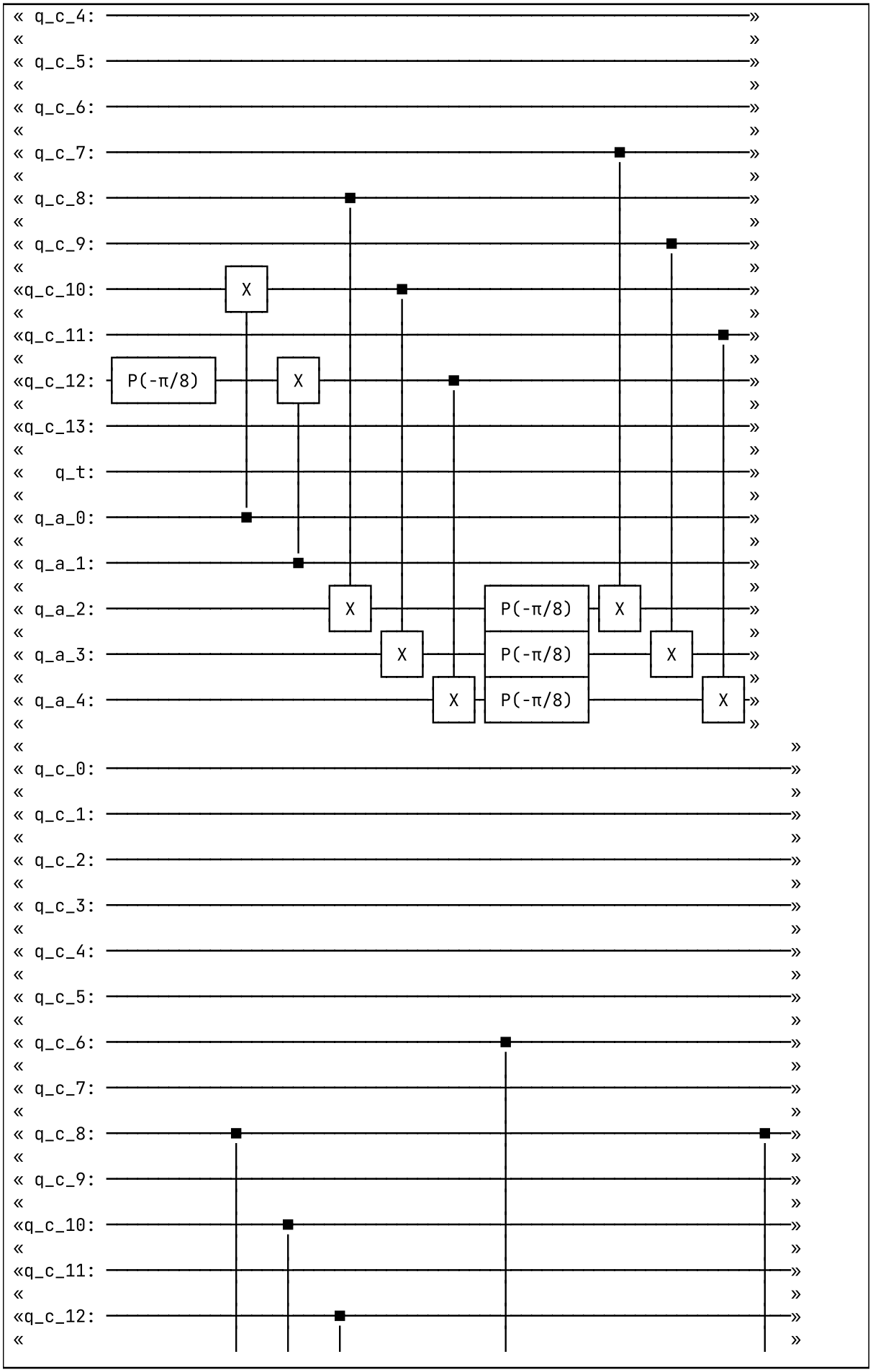

q_c_0:
q_c_1:
q_c_2:
q_c_3:
q_c_4:
q_c_5:
q_c_6:
q_c_7:
q_c_8:
q_c_9:
q_c_10:
q_c_11:
q_c_12:
q_c_13:
q_t:
q_a_0:
q_a_1:
q_a_2:
q_a_3:
q_a_4:
P(π/8)
X
P(-π/8)
H

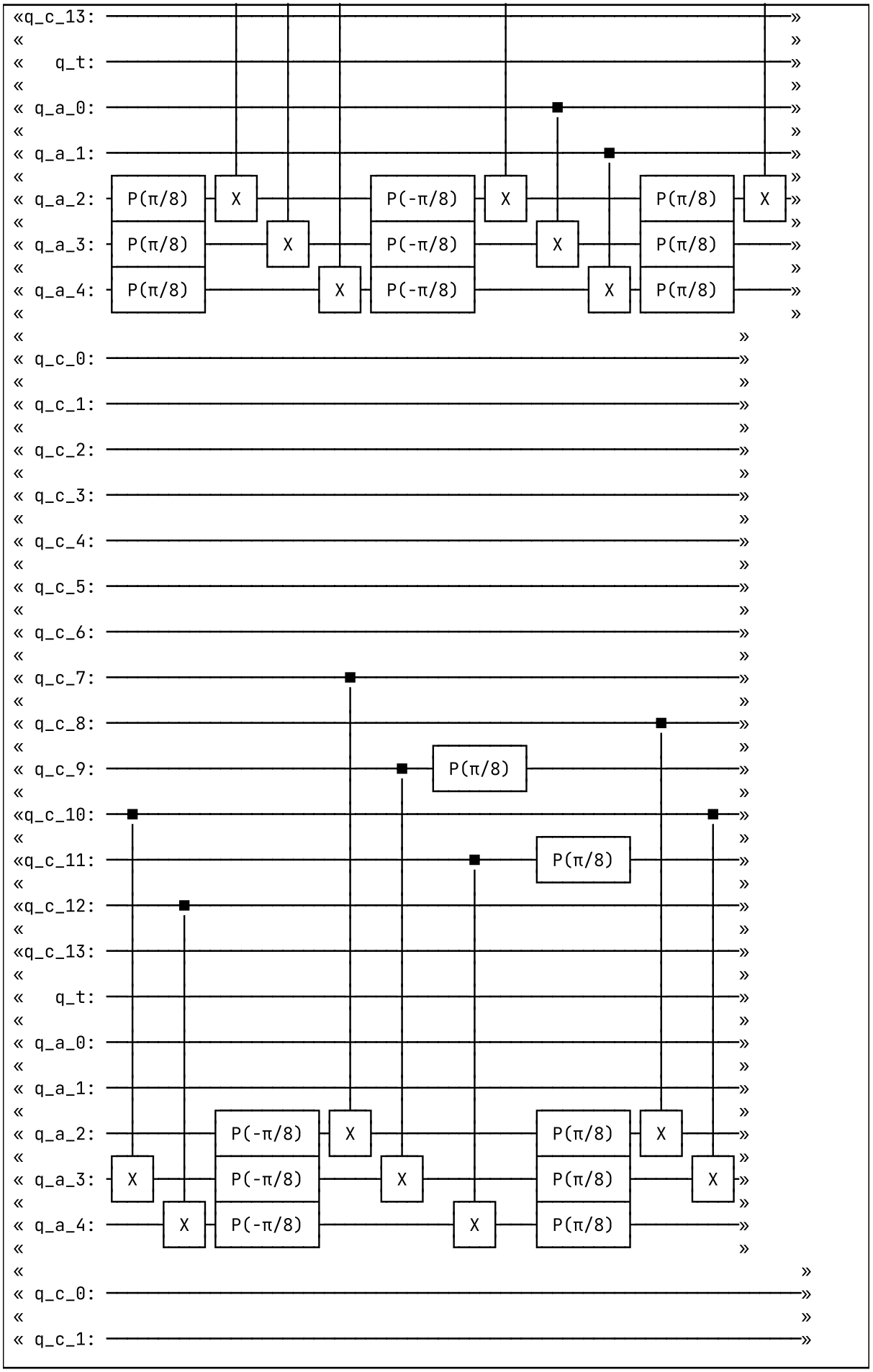

« q_c_9:
«
«q_c_10:
«
«q_c_11:
«
«q_c_12:
«
«q_c_13:
«
« q_t:
«
« q_a_0:
«
« q_a_1:
«
« q_a_2:
«
« q_a_3:
«
« q_a_4:
«
X
X
X
P(-π/8)
P(-π/8)
P(-π/8)
X
X
X
« q_c_0:
« q_c_1:
« q_c_2:
« q_c_3:
« q_c_4:
« q_c_5:
« q_c_6:
« q_c_7:
« q_c_8:
« q_c_9:
«q_c_10:
«q_c_11:
«q_c_12:
«q_c_13:
« q_t:
« q_a_0:
« q_a_1:
« q_a_2:
P(π/8)
P(π/8)
P(π/8)
X
X
X
P(-π/8)
P(-π/8)
P(-π/8)
X
X
X
P(π/8)
P(π/8)
P(π/8)
X
X
X

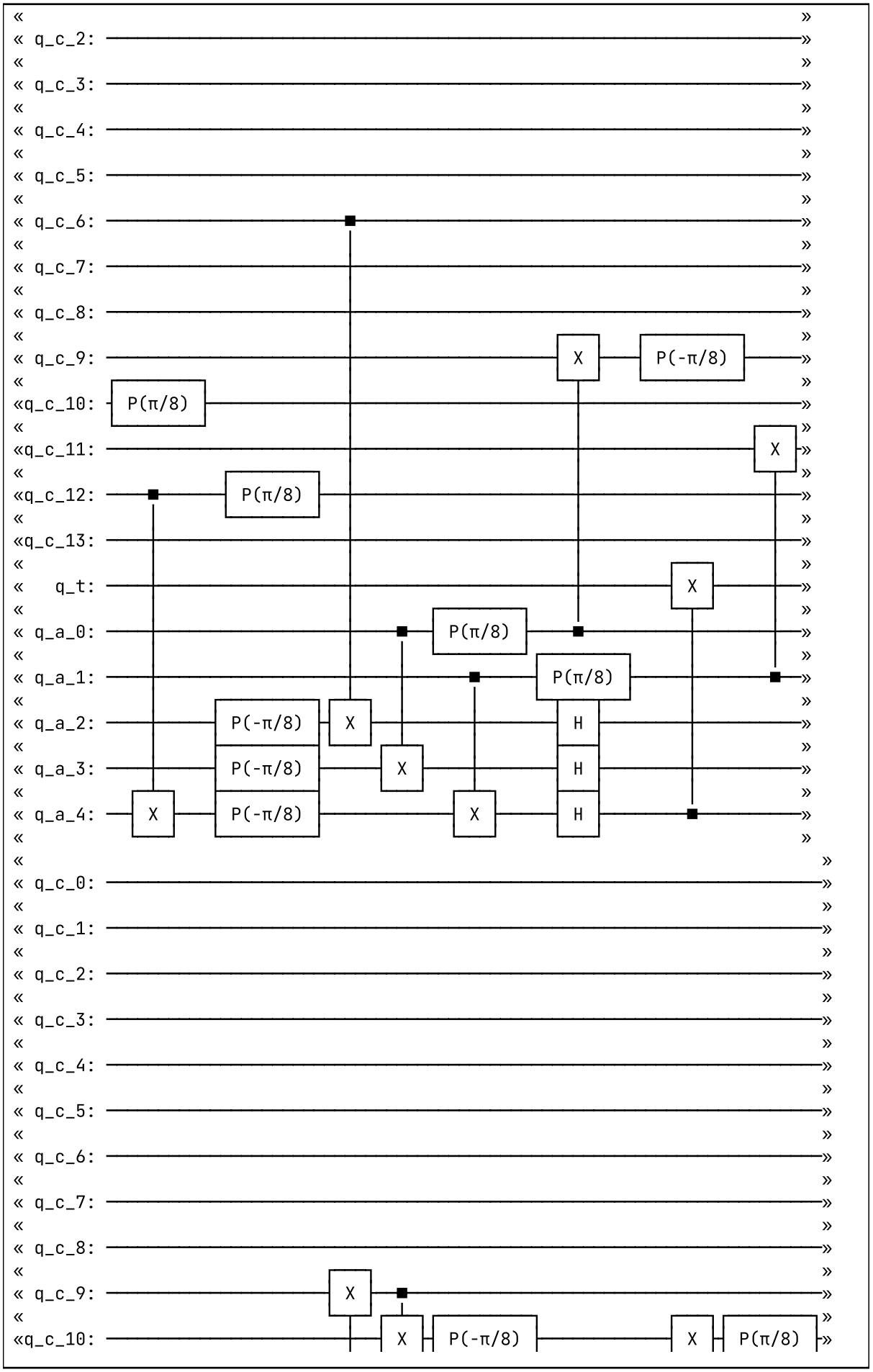

« q_a_3:
« q_a_4:
« q_c_0:
« q_c_1:
« q_c_2:
« q_c_3:
« q_c_4:
« q_c_5:
« q_c_6:
« q_c_7:
« q_c_8:
« q_c_9:
«q_c_10:
«q_c_11:
«q_c_12:
«q_c_13:
« q_t:
« q_a_0:
« q_a_1:
« q_a_2:
« q_a_3:
« q_a_4:
P(π/8)
P(π/8)
P(π/8)
P(-π/8)
P(-π/8)
P(-π/8)
X
X
X
X
X
X
P(π/8)
P(π/8)
P(π/8)
P(π/8)
P(π/8)
P(π/8)
P(π/8)
X
P(-π/8)
X
P(π/8)
P(π/8)
X
P(π/8)

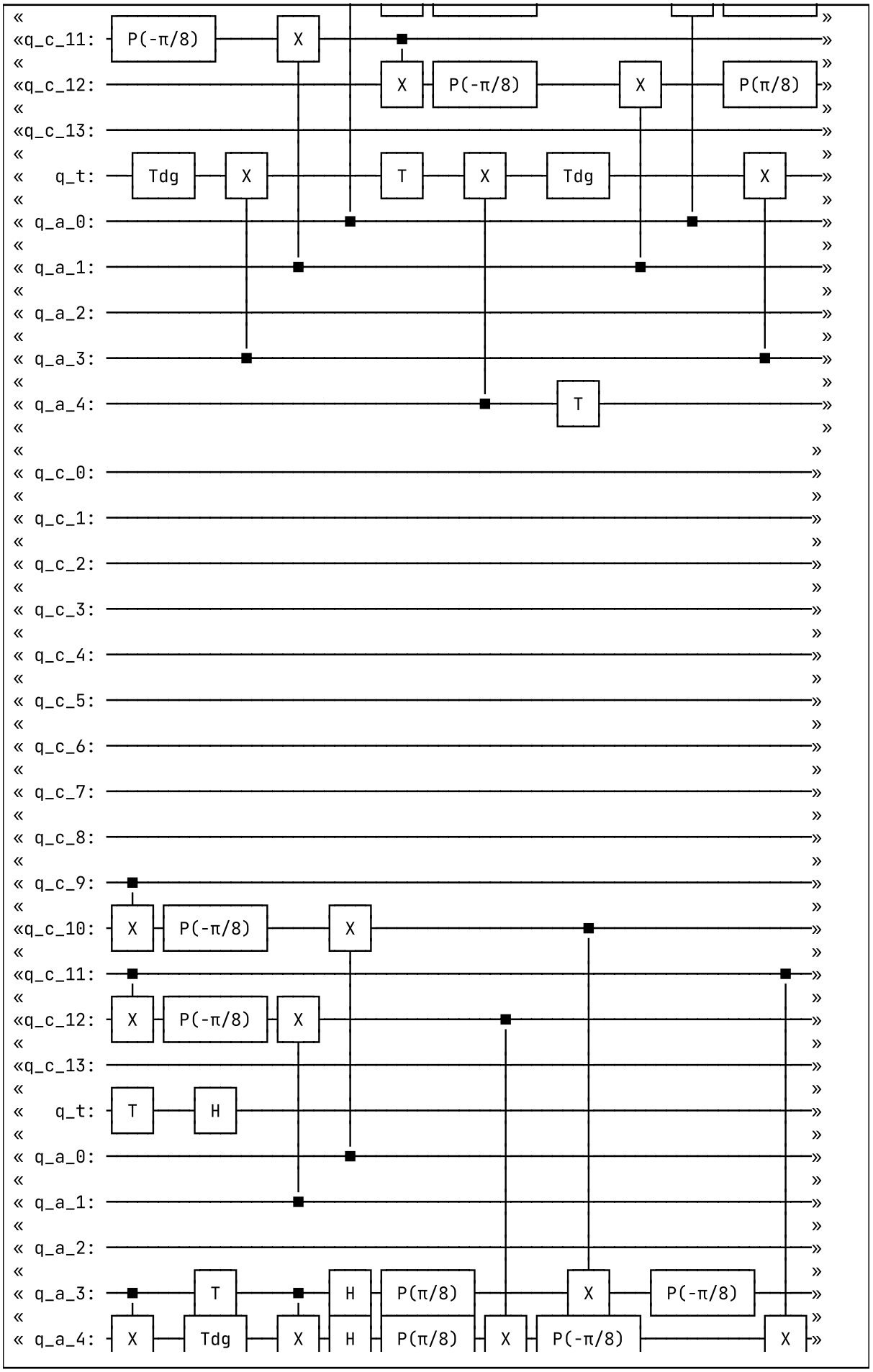

q_c_7:
q_c_8:
P(π/8)
q_c_9:
q_c_10:
q_c_11:
q_c_12:
q_c_13:
q_t:
q_a_0:
P(-π/8)
X
H
q_a_1:
P(-π/8)
X
H
q_a_2:
X
P(-π/8)
X
H
q_a_3:
q_a_4:
q_c_0:
q_c_1:
X
q_c_2:
X
P(-π/8)
X
P(π/8)
X
q_c_3:
q_c_4:
P(-π/8)
X
q_c_5:
X
P(-π/8)
X
P(π/8)
q_c_6:
q_c_7:
X
P(-π/8)
X
q_c_8:
X
P(-π/8)
X
q_c_9:
X
P(-π/8)
X
q_c_10:
X
q_c_11:
X
P(-π/8)
X
q_c_12:
q_c_13:
q_t:
q_a_0:
P(π/8)

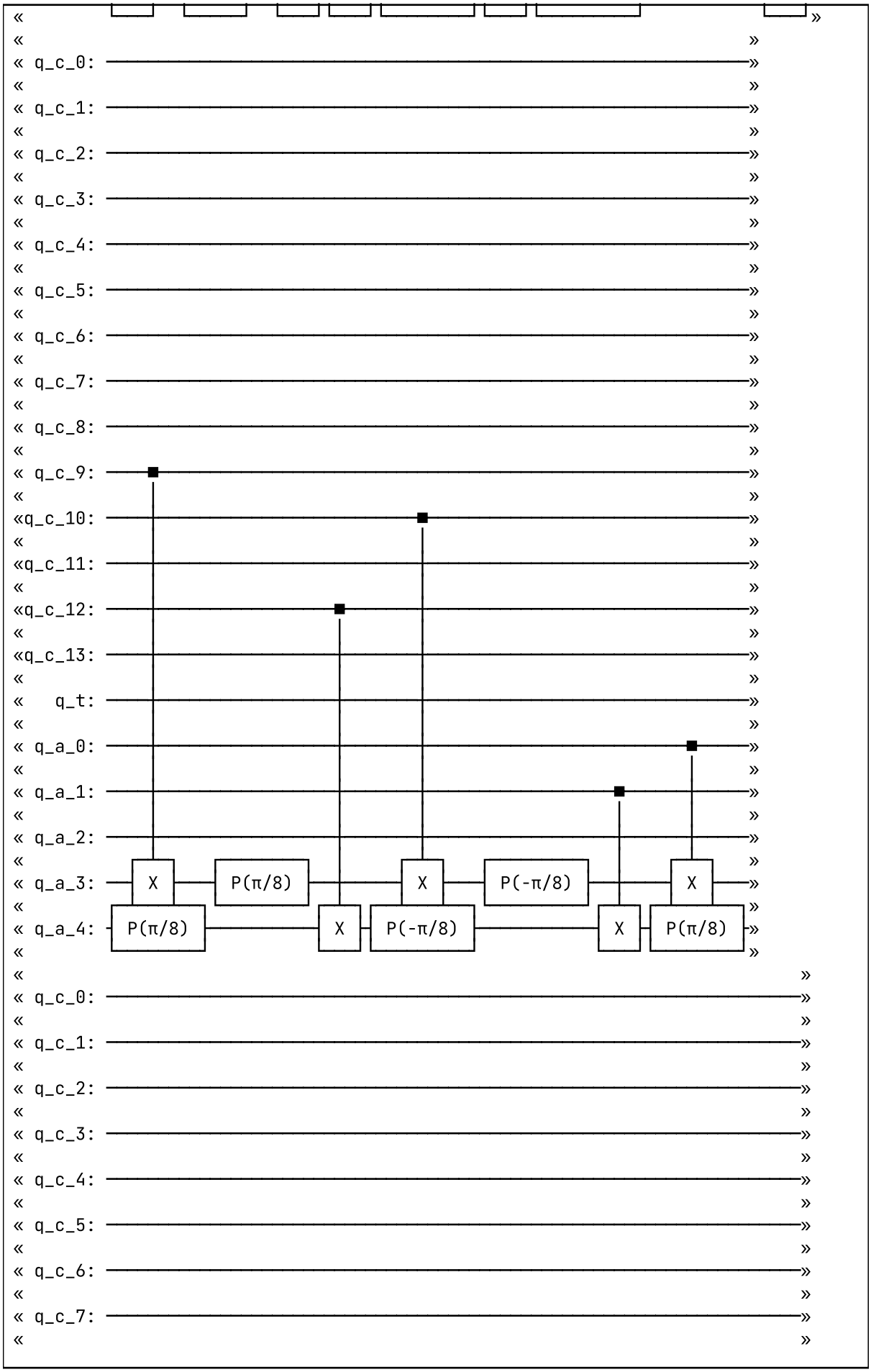

q_a_1:
P(π/8)
q_a_2:
H
P(π/8)
q_a_3:
q_a_4:
q_c_0:
q_c_1:
q_c_2:
P(-π/8)
X
q_c_3:
q_c_4:
q_c_5:
X
P(-π/8)
X
q_c_6:
q_c_7:
q_c_8:
P(π/8)
X
P(-π/8)
X
q_c_9:
q_c_10:
P(-π/8)
X
P(π/8)
X
P(-π/8)
q_c_11:
q_c_12:
X
P(-π/8)
X
P(π/8)
X
q_c_13:
q_t:
q_a_0:
q_a_1:
q_a_2:
q_a_3:
q_a_4:
q_c_0:
q_c_1:
q_c_2:
q_c_3:

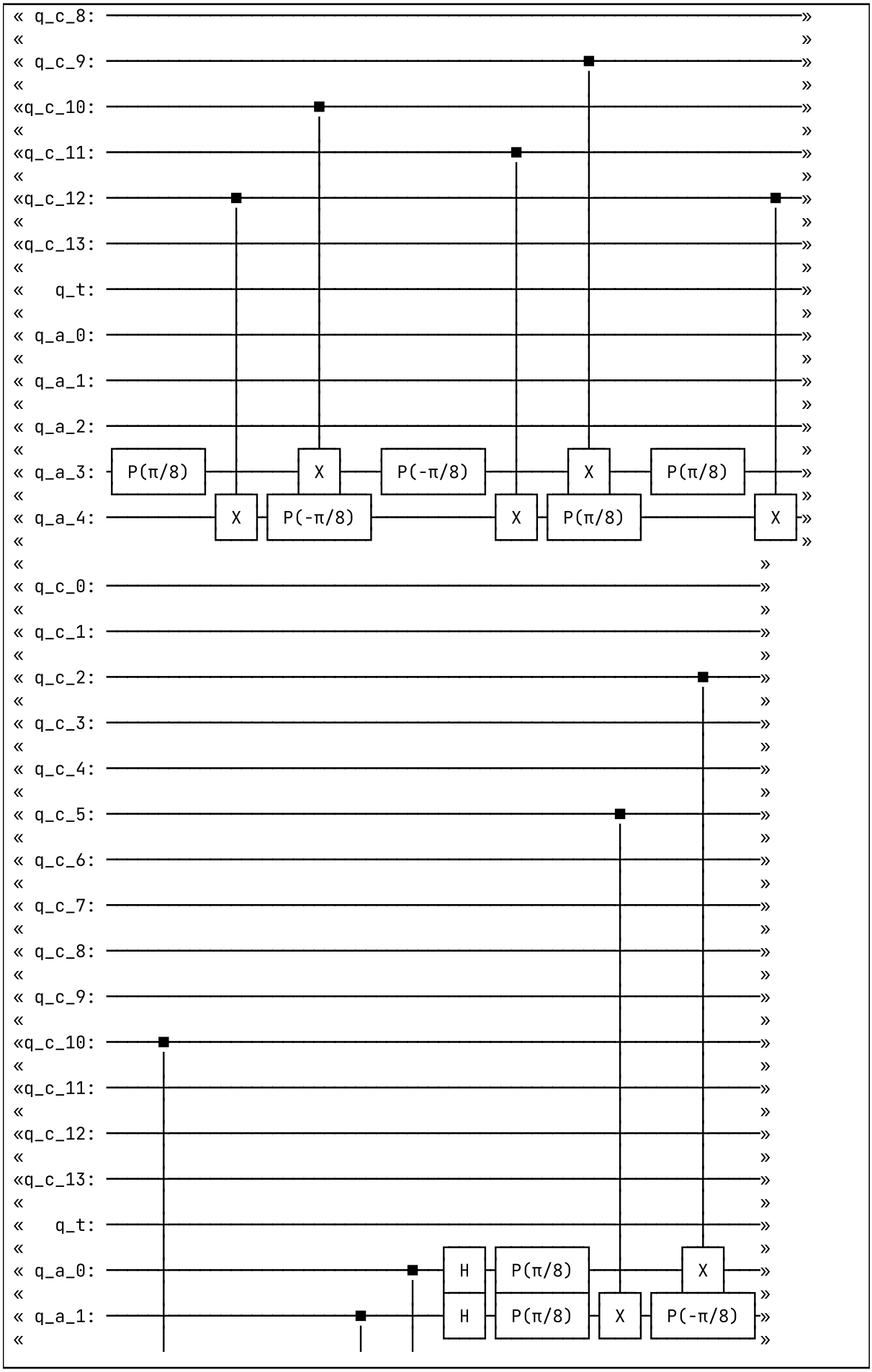

« q_c_4:
« q_c_5:
« q_c_6:
« q_c_7:
« q_c_8:
« q_c_9:
«q_c_10:
«q_c_11:
«q_c_12:
«q_c_13:
« q_t:
« q_a_0:
« q_a_1:
« q_a_2:
« q_a_3:
« q_a_4:
X
P(-π/8)
X
X
X
X
P(-π/8)
P(-π/8)
P(-π/8)
X
X
X
« q_c_0:
« q_c_1:
« q_c_2:
« q_c_3:
« q_c_4:
« q_c_5:
« q_c_6:
« q_c_7:
« q_c_8:
« q_c_9:
«q_c_10:
«q_c_11:
«q_c_12:

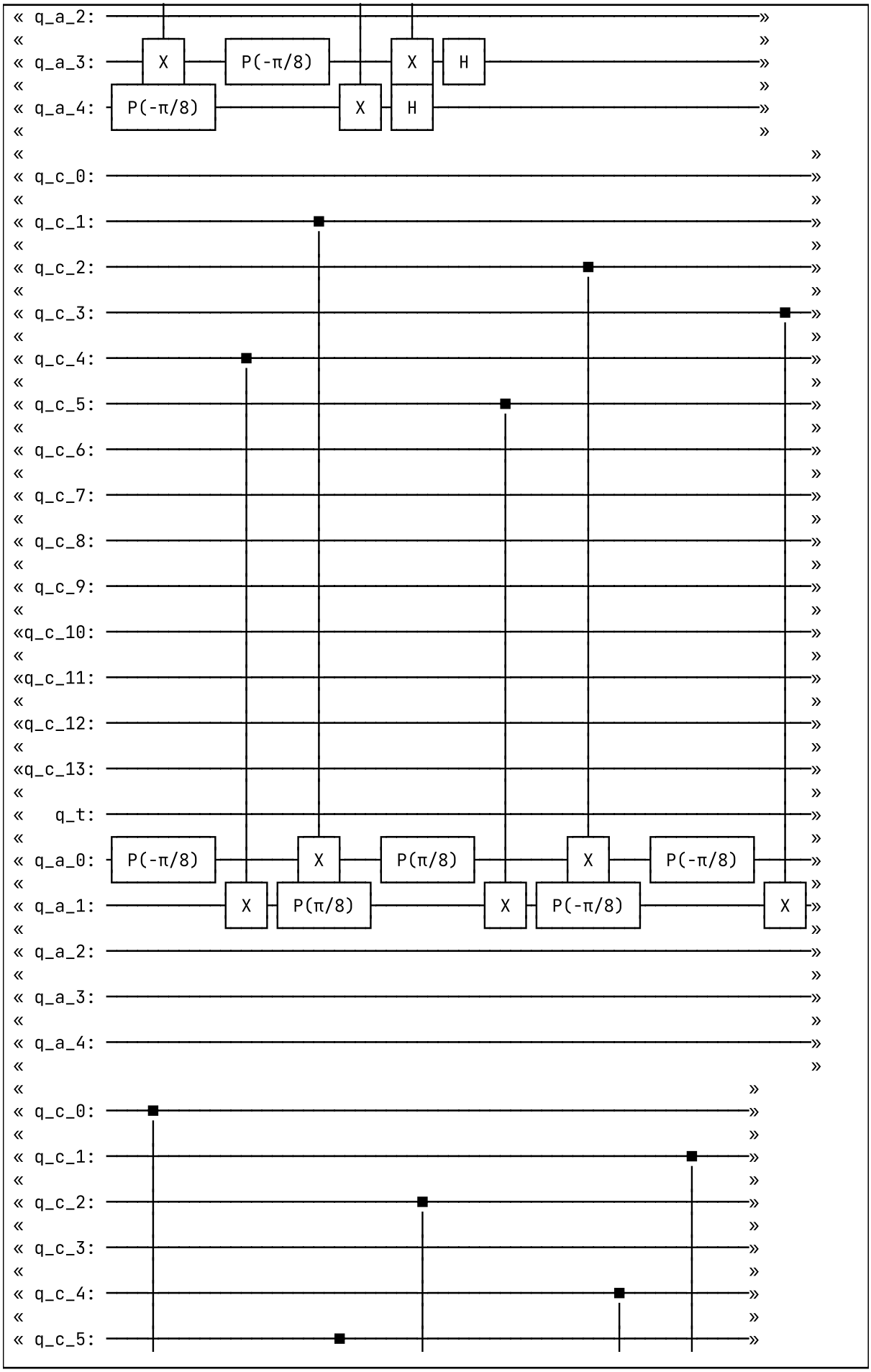

«q_c_13:
«
«   q_t:
«
« q_a_0:
«
« q_a_1:
«
« q_a_2:
«
« q_a_3:
«
« q_a_4:
«
P(π/8)
X
P(-π/8)
«
« q_c_0:
«
« q_c_1:
«
« q_c_2:
«
« q_c_3:
«
« q_c_4:
«
« q_c_5:
«
« q_c_6:
«
« q_c_7:
«
« q_c_8:
«
« q_c_9:
«
«q_c_10:
«
«q_c_11:
«
«q_c_12:
«
«q_c_13:
«
«   q_t:
«
« q_a_0:
«
« q_a_1:
«
« q_a_2:
«
« q_a_3:
«
« q_a_4:
«
«
« q_c_0:
«
« q_c_1:

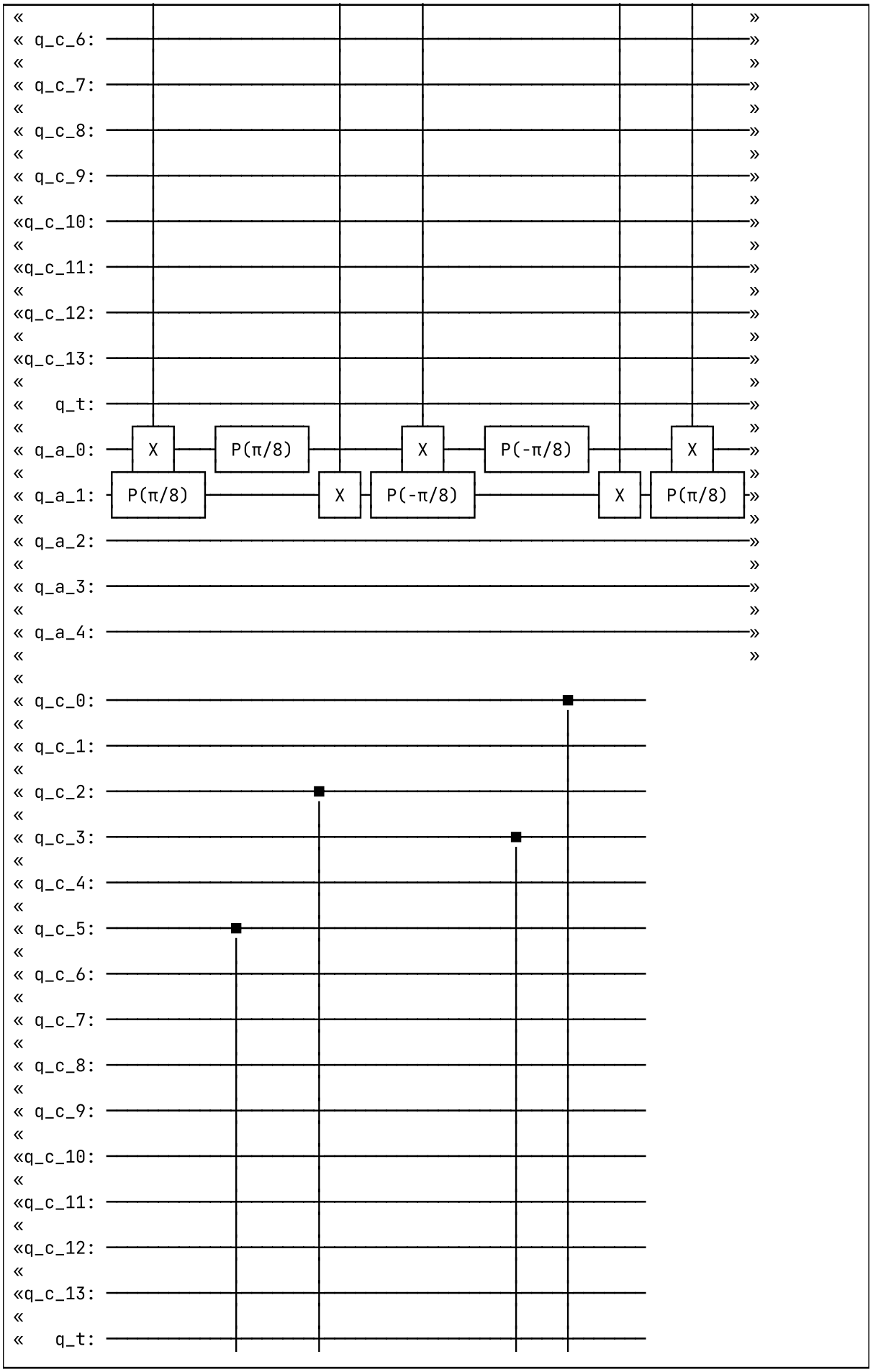

q_c_2:
q_c_3:
q_c_4:
q_c_5:
q_c_6:
q_c_7:
q_c_8:
q_c_9:
q_c_10:
q_c_11:
q_c_12:
q_c_13:
q_t:
q_a_0:
q_a_1:
q_a_2:
q_a_3:
q_a_4:
q_c_0:
q_c_1:
q_c_2:
q_c_3:
q_c_4:
q_c_5:
q_c_6:
q_c_7:
q_c_8:
q_c_9:
q_c_10:
X
P(-π/8)
P(π/8)
X
X
P(π/8)
P(π/8)
P(-π/8)
X
H
P(-π/8)
X
H
X
P(-π/8)
X
H
X
X
P(-π/8)
X
P(π/8)

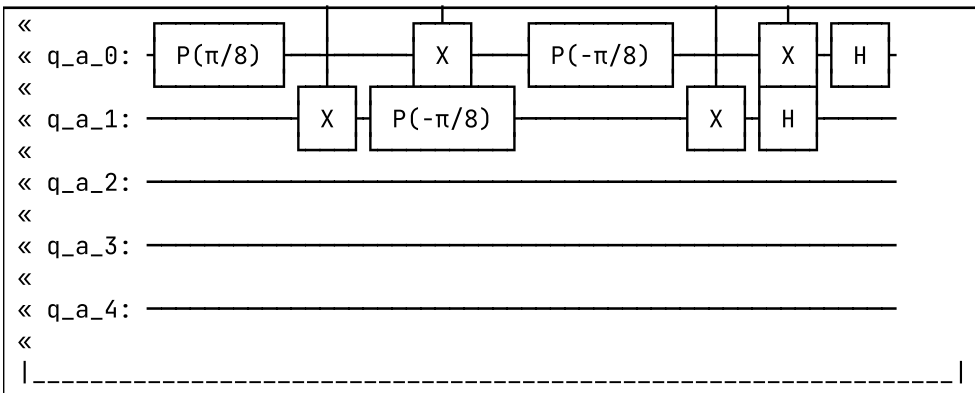

q_c_11:
P(-π/8)
X
q_c_12:
X
P(-π/8)
X
P(π/8)
q_c_13:
q_t:
Tdg
X
T
X
Tdg
X
q_a_0:
q_a_1:
q_a_2:
q_a_3:
q_a_4:
T
q_c_0:
q_c_1:
q_c_2:
q_c_3:
q_c_4:
q_c_5:
q_c_6:
q_c_7:
q_c_8:
q_c_9:
q_c_10:
X
P(-π/8)
X
q_c_11:
q_c_12:
X
P(-π/8)
X
q_c_13:
q_t:
T
H
q_a_0:
q_a_1:
q_a_2:
q_a_3:
T
H
P(π/8)
X
P(-π/8)
q_a_4:
X
Tdg
X
H
P(π/8)
X
P(-π/8)
X


q_c_0:
q_c_1:
q_c_2:
q_c_3:
q_c_4:
q_c_5:
q_c_6:
q_c_7:
q_c_8:
q_c_9:
q_c_10:
q_c_11:
q_c_12:
q_c_13:
q_t:
q_a_0:
q_a_1:
q_a_2:
q_a_3:
q_a_4:
X
P(π/8)
X
P(-π/8)
X
P(π/8)
X
P(-π/8)
X
P(π/8)


« q_c_8:
« q_c_9:
«q_c_10:
«q_c_11:
«q_c_12:
«q_c_13:
« q_t:
« q_a_0:
« q_a_1:
« q_a_2:
« q_a_3:
« q_a_4:
P(π/8)
X
P(-π/8)
X
P(π/8)
X
P(-π/8)
X
P(π/8)
X
« q_c_0:
« q_c_1:
« q_c_2:
« q_c_3:
« q_c_4:
« q_c_5:
« q_c_6:
« q_c_7:
« q_c_8:
« q_c_9:
«q_c_10:
«q_c_11:
«q_c_12:
«q_c_13:
« q_t:
« q_a_0:
« q_a_1:
H
H
P(π/8)
P(π/8)
X
X
P(-π/8)


« q_a_2:
« q_a_3:
« q_a_4:
X
P(-π/8)
X
H
P(-π/8)
X
H
« q_c_0:
« q_c_1:
« q_c_2:
« q_c_3:
« q_c_4:
« q_c_5:
« q_c_6:
« q_c_7:
« q_c_8:
« q_c_9:
«q_c_10:
«q_c_11:
«q_c_12:
«q_c_13:
« q_t:
« q_a_0:
« q_a_1:
« q_a_2:
« q_a_3:
« q_a_4:
P(-π/8)
X
P(π/8)
X
P(π/8)
X
P(-π/8)
X
P(-π/8)
X
« q_c_0:
« q_c_1:
« q_c_2:
« q_c_3:
« q_c_4:
« q_c_5:


q_c_6:
q_c_7:
q_c_8:
q_c_9:
q_c_10:
q_c_11:
q_c_12:
q_c_13:
q_t:
q_a_0:
q_a_1:
q_a_2:
q_a_3:
q_a_4:
X
P(π/8)
X
P(-π/8)
X
P(π/8)
P(π/8)
X
P(-π/8)
X
P(π/8)
q_c_0:
q_c_1:
q_c_2:
q_c_3:
q_c_4:
q_c_5:
q_c_6:
q_c_7:
q_c_8:
q_c_9:
q_c_10:
q_c_11:
q_c_12:
q_c_13:
q_t:




```
Process finished with exit code 0
```