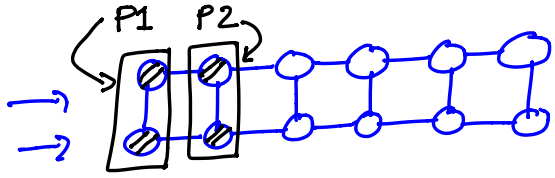
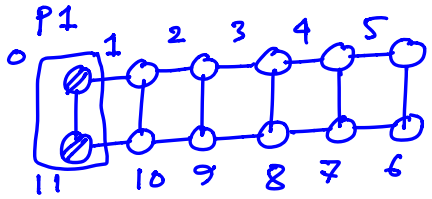


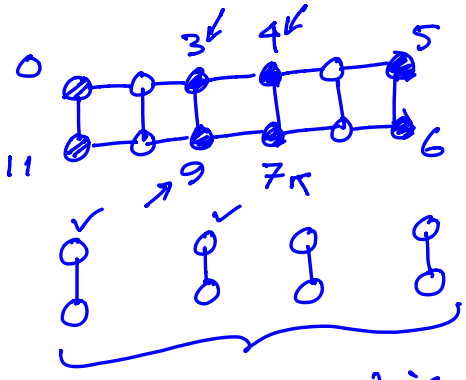


orig ckt.

on run ^4 isomorphic subgraphs

`qc.cx(qr[0], qr[11])`

create 4 copies



iso-morphic subgraph

1. To run py prog. for 4 times.

2. run py prog for 1 time but use "circ list" in execution function

```
qc0. cx(qr[0], qr[11])
 :
qc1. cx(qr[3], qr[9])
 :
qc2. cx(qr[4], qr[7])
qc3. cx(qr[5], qr[6])
```

qc orig ckt.

SHOT = 8192 or 1024

75 times

circ_list = [qc, qc, ...]

Now,

circ_list = [qc0, qc1, qc2, qc3, qc0....]

max allowed 75

72

circ_list = [qc0, qc1, qc2, qc3....]

x 18

1x

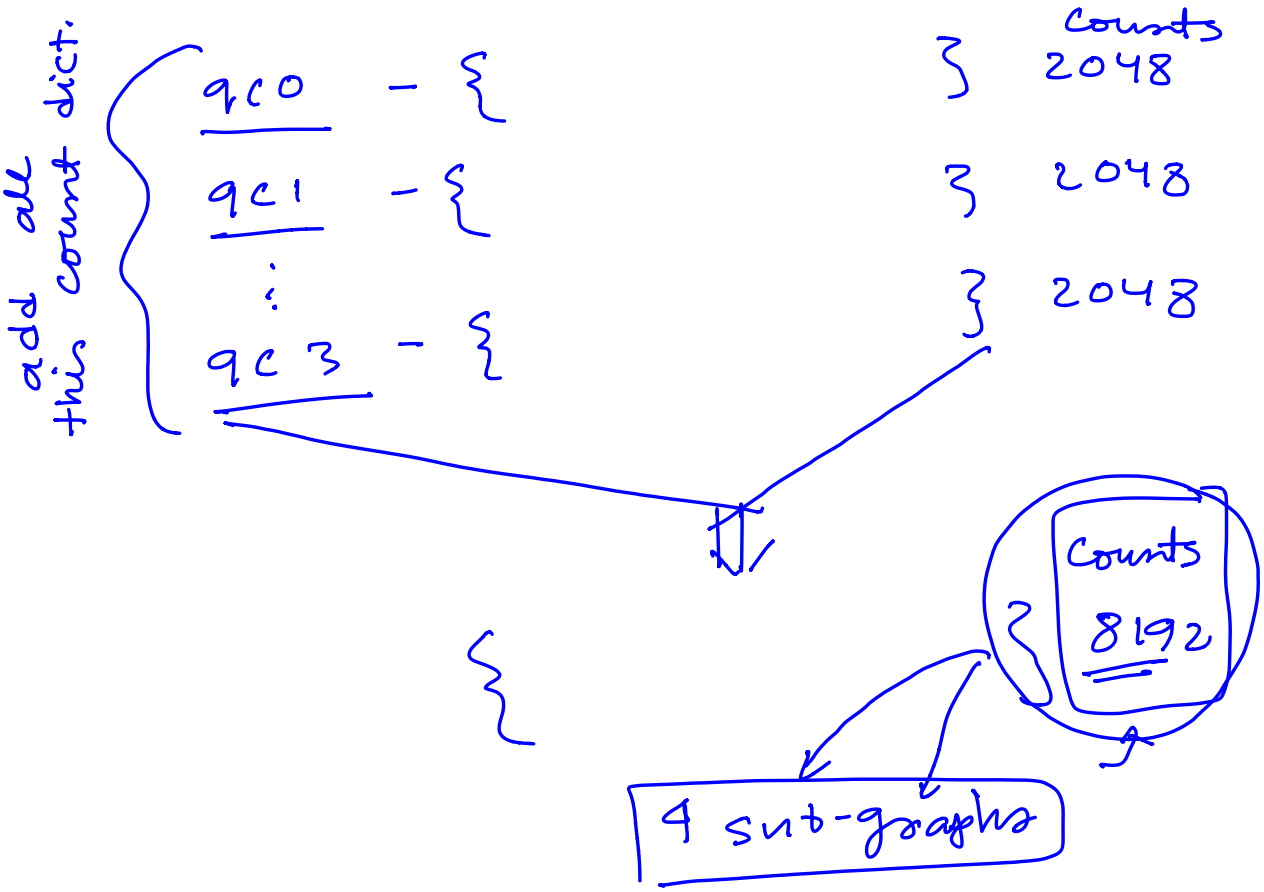
so that the length of circ_list will 72

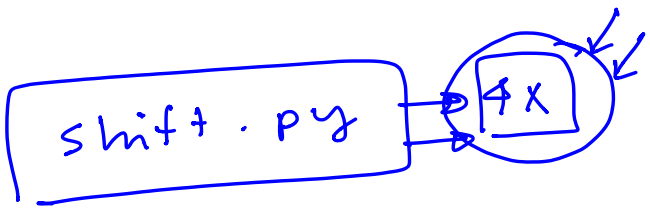
job = execute (circ_list, backend, shots = 2048)

shifting xx.py

8192/4=2048

x4 ?

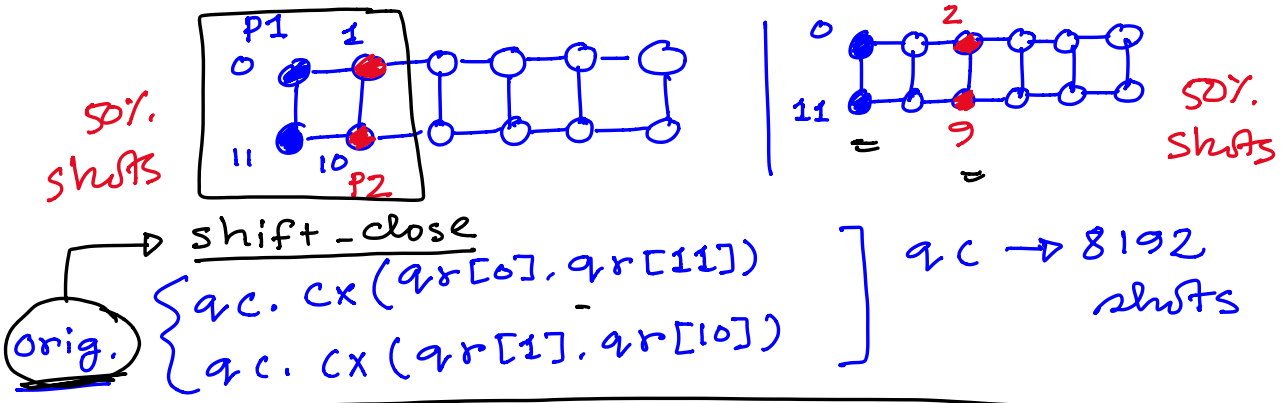


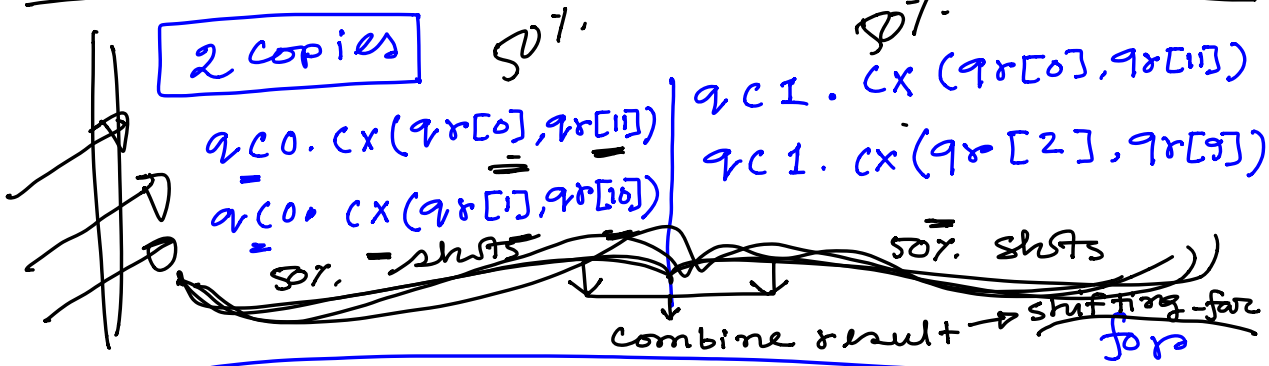


qr[ ], qr[ ]

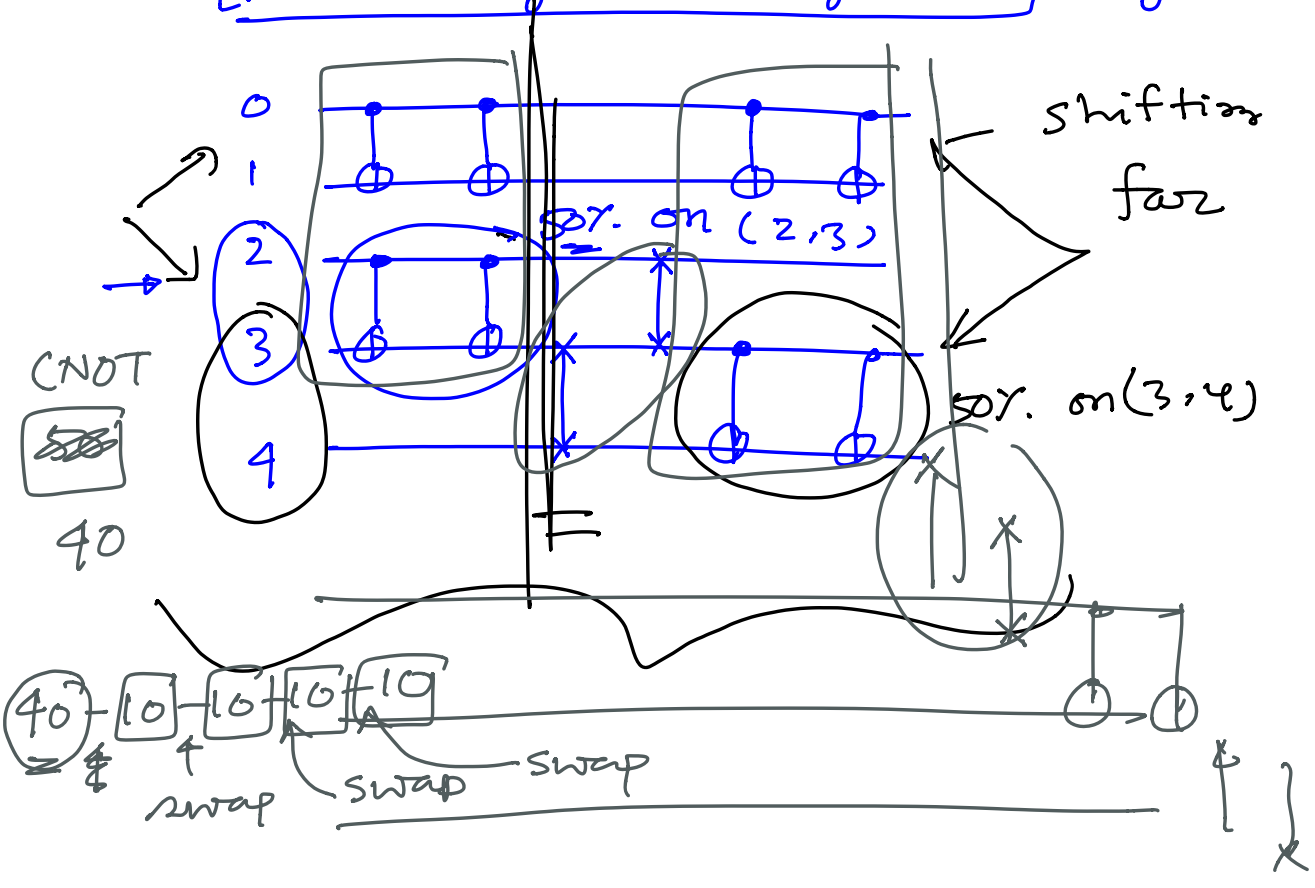
shift = 2048)

execution(





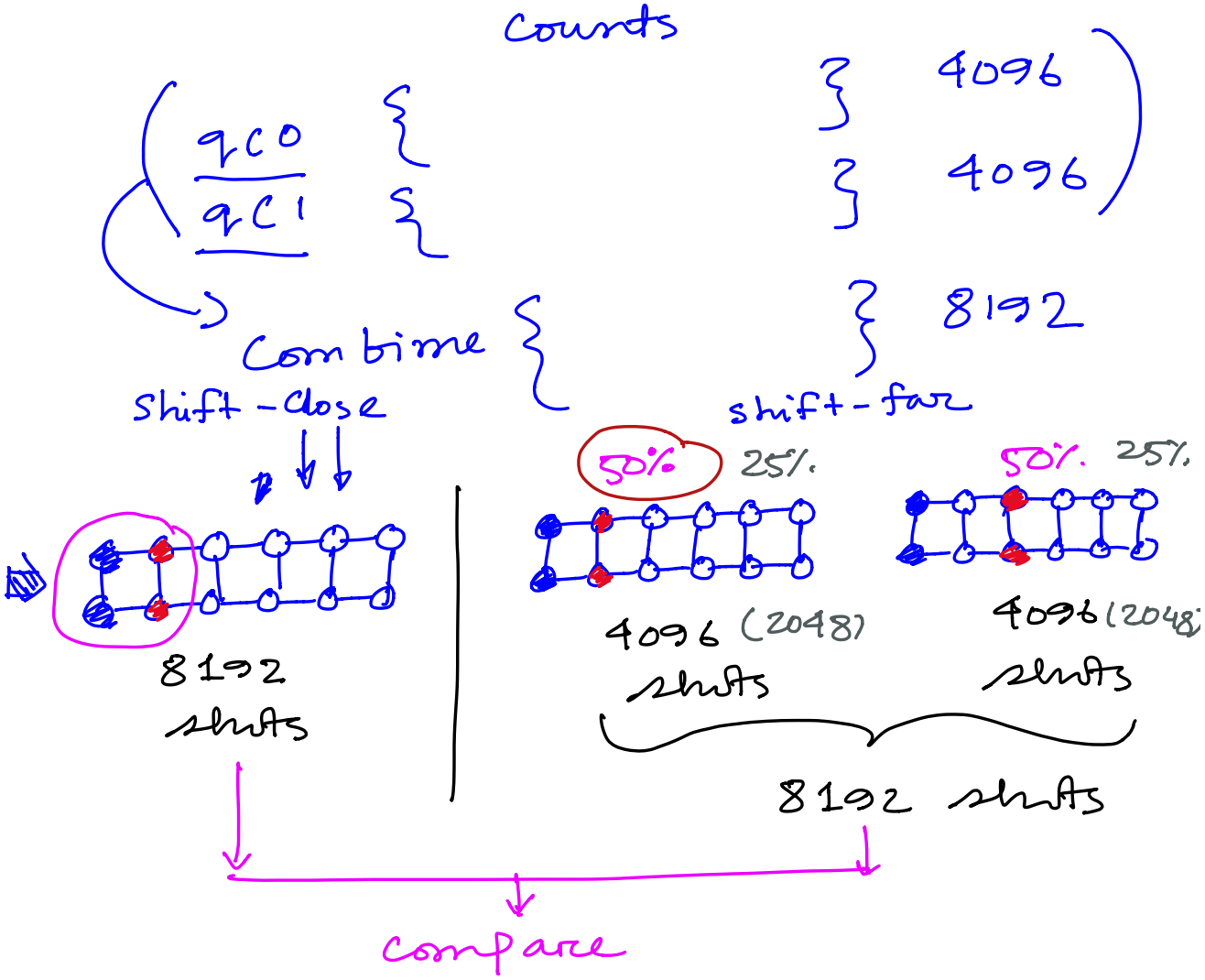
Re-visiting shifting far: Bogota

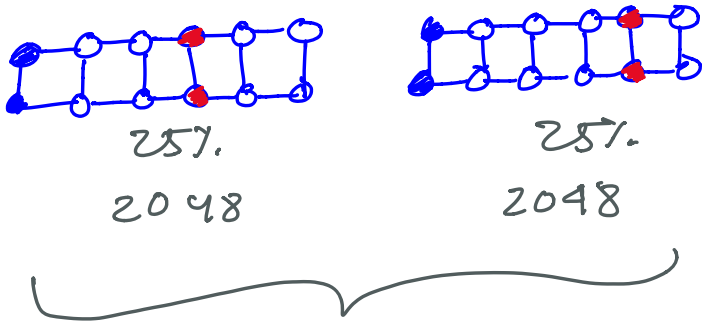


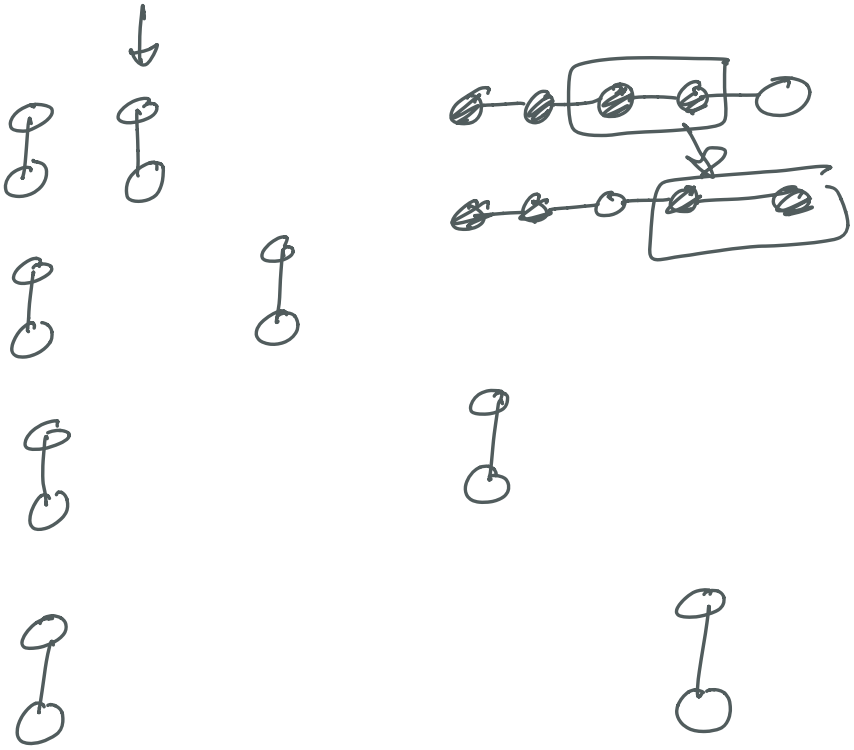
qc0
qc1

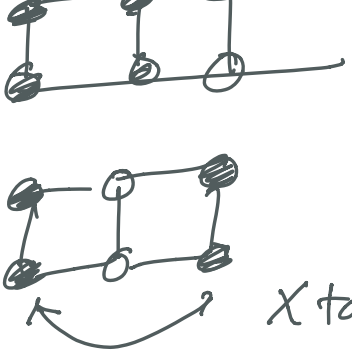
circ_list = [qc0, qc1, ...]
repeated

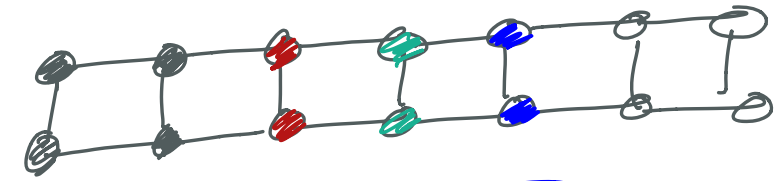
execution = (circ_list, backend, shots = 8192//2)




25%.
2048
25%.
2048




Xtalk max.
Xtalk will be smaller



for
1
2
3

swaps

~~SWAP~~

New

distributing number of SHOTS