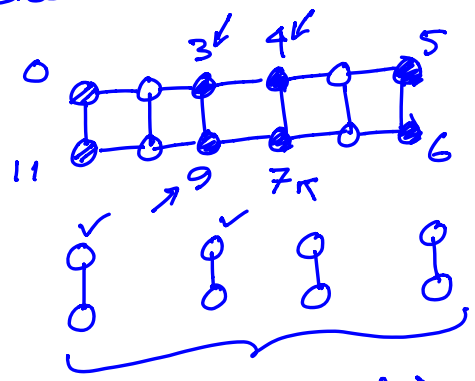


orig ckt.

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

create 4 copies



iso-morphic subgraph

on
run 4 isomorphic
subgraphs

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])`
 \vdots
`qc1.cx(qr[3], qr[9])`
 \vdots
`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,

ow,


↓
max allowed (75) 72

$$\text{circ_list} = [\underbrace{q_{c0}, q_{c1}, q_{c2}, q_{c3}, \dots}_{\times 18}]$$

1x ✓

so that
the length of circ_list will 72

```
job = execute(circ-int, backend,  
shots = 2048)
```

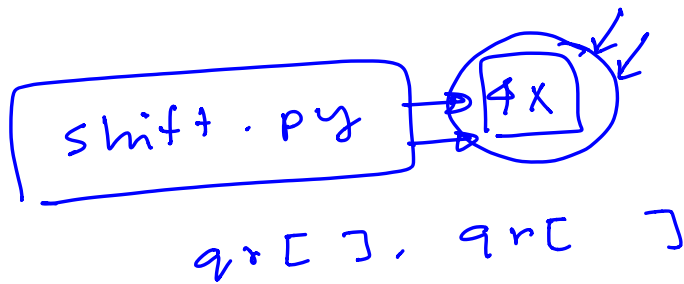
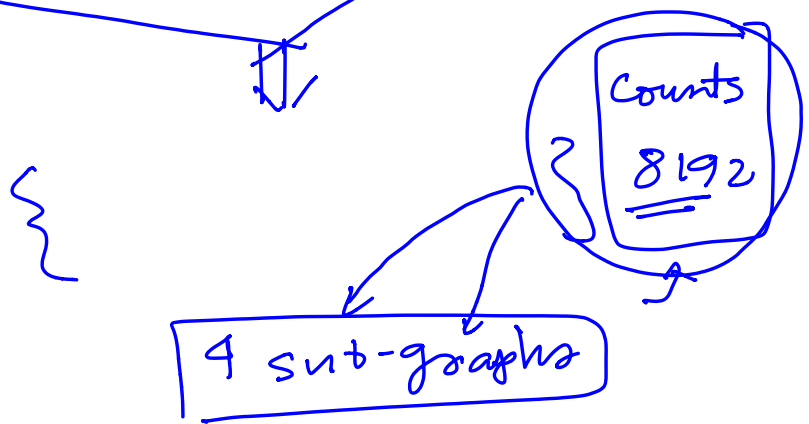
shifting xx.py

$$8192/4 = 2048$$

x^4 ?

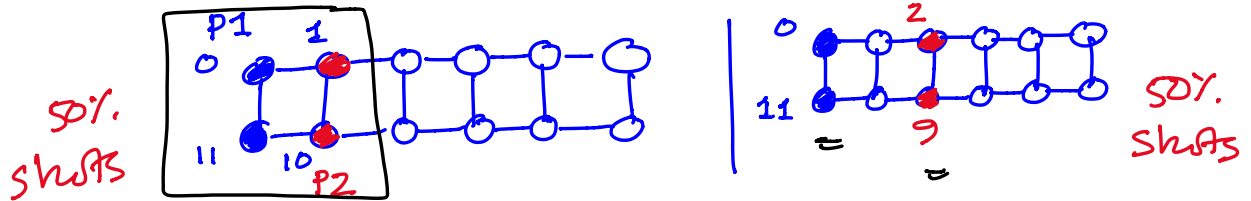
add all
this count dict.

<u>qc0</u> - {	}	Counts 2048
<u>qc1</u> - {	}	2048
:	}	2048
<u>qc3</u> - {	}	

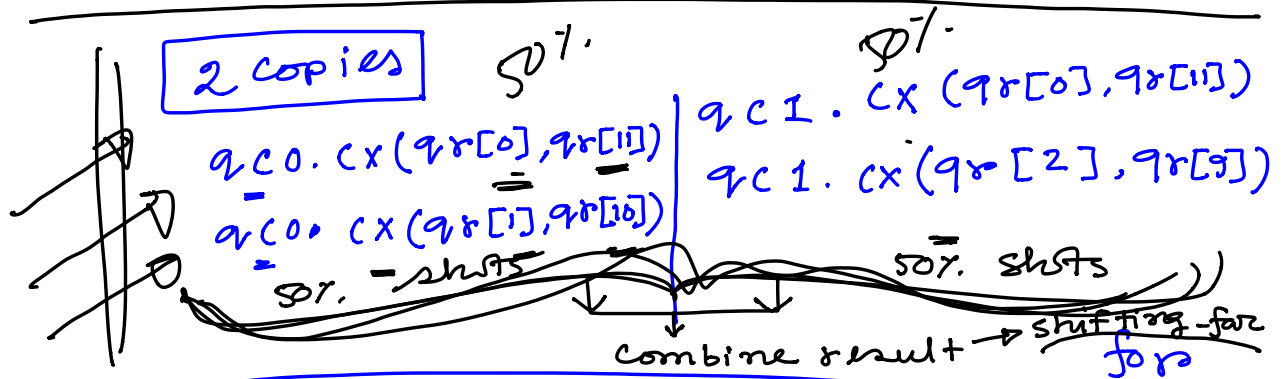


exeeution(

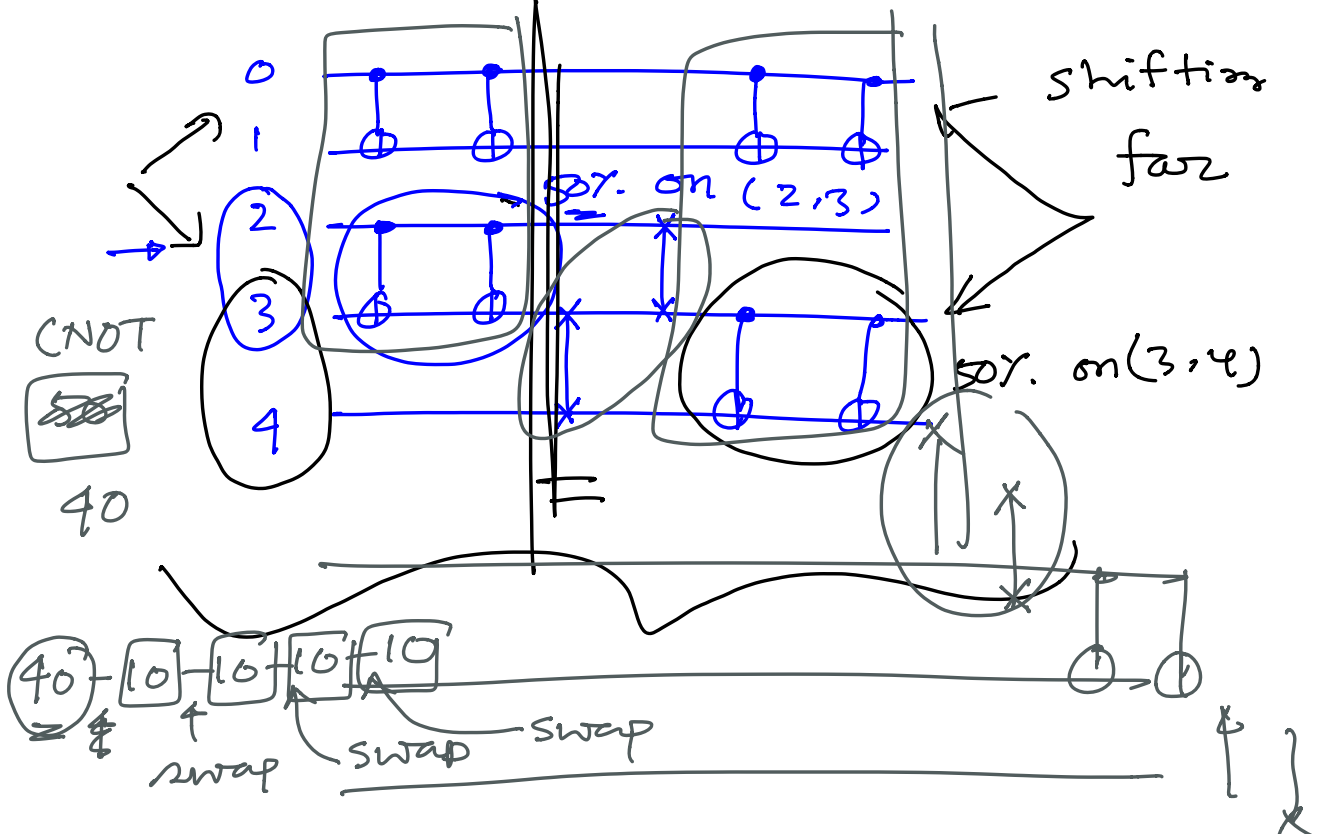
shift=2048)



orig. $\xrightarrow{\text{shift_close}}$ $\left\{ \begin{array}{l} qc.cx(qr[0], qr[11]) \\ qc.cx(qr[1], qr[10]) \end{array} \right\} \rightarrow qc \rightarrow 8192 \text{ shifts}$



Re-visiting shifting far: Bogota



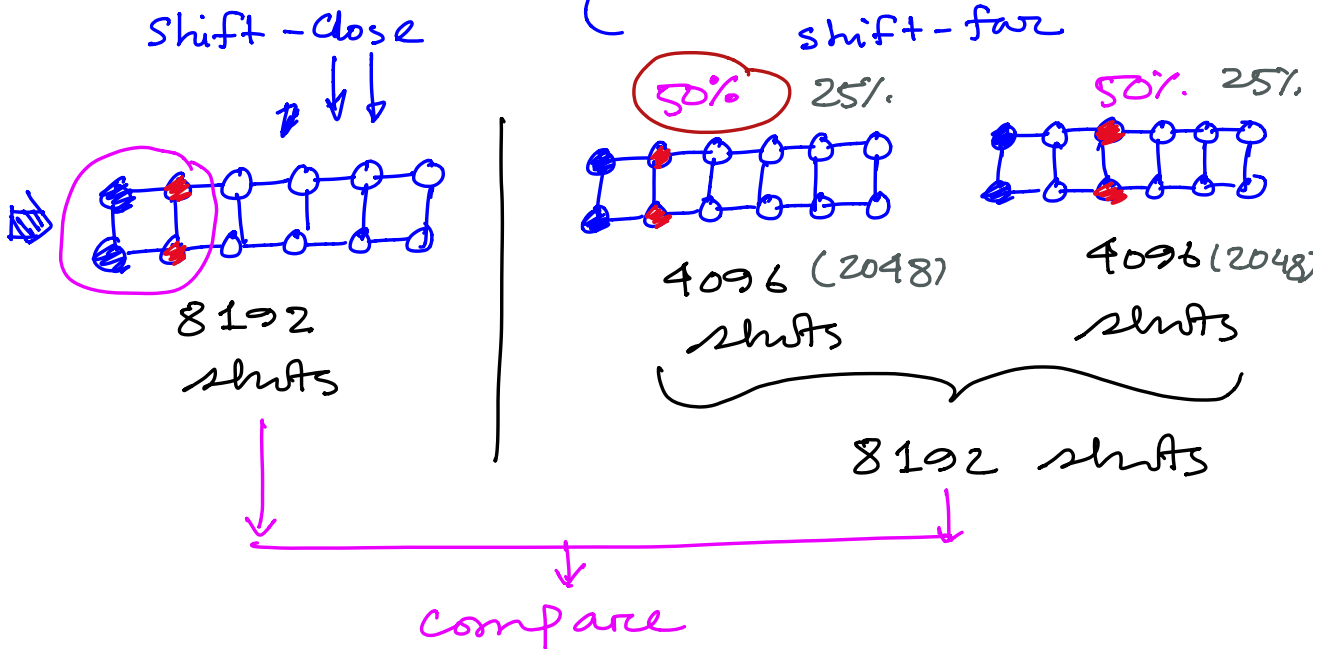
qco
qci

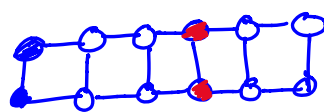
circ-list = [qco, qci,]
repeated

execution = (circ-list, backend,
shifts = 8192/2)

counts

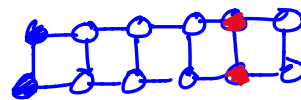
$\left(\begin{array}{l} \{ \text{qco} \\ \{ \text{qci} \} \end{array} \right) \left\{ \begin{array}{l} 4096 \\ 4096 \end{array} \right\}$
 Comb time $\left\{ \begin{array}{l} 8192 \end{array} \right\}$





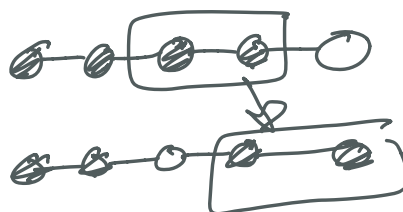
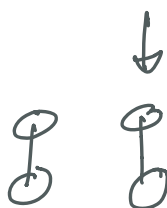
25%

2048

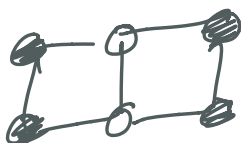
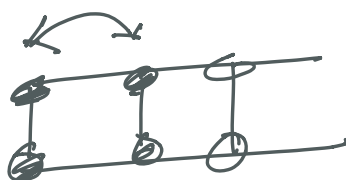


25%

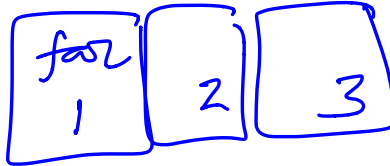
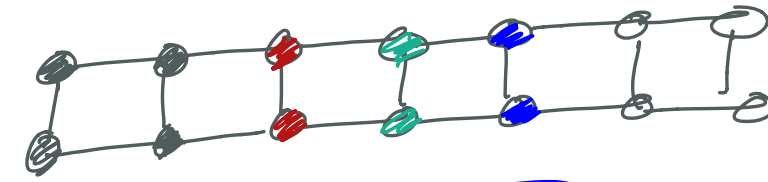
2048



$x_{talk} \max.$



x_{talk} will be smaller



swaps

New ~~SWAP~~

→ distributing number
of
SHOTS