Physical constraint-aware CNOT quantum circuit synthesis & optimization

Zhu, Cheng, Zhu, Chen, Guan , 2022

Abstract

1. What are the physical constraints for noisy intermediate-scale quantum (NISQ) devices?

   Nearest-neighbour (NN) interaction and gate error

   Def: A two-qubit gate can only be performed on adjacent qubits.

   ↓

   quantum circuit execution success rate

   Def: Various noises in the quantum system impact the precision of qc on the hardware

2. What are the physical constraints of the CNOT circuit w.r.t mapping and optimization?

   ① NN interaction

   ② gate error rate.

3. What is the deliverable of this paper?

   They presented a physical constraint-aware CNOT quantum circuit synthesis and optimization approach.

   synthesis:
   ① weighted Steiner tree to account for CNOT gate errors between adjacent qubits of the NISQ device
   ② The minimum Steiner tree algorithm

   optimization: matrix-transformation-based optimization approach

4. What are the experimental results?

   They showed that on average, the proposed NN-synthesis enhanced the circuit fidelity by:

   - Success rate ↗ 382.86%
   - # gates ↘ 58.3%

I. Introduction

1.2 What is the purpose of such mapping?

For the logical quantum circuit to be executed in the physical quantum hardw

there is a success rate.

1.1 What is the synthesis and mapping of quantum circuits?

To automatically construct a quantum circuit with a given quantum function on the target quantum computer architecture. ⇨ Find a representation of a quantum algorithm. [logical level].

2. Why study the synthesis of the CNOT quantum circuit?

(1) Universal quantum circuits can be built using CNOT gates and different single-qubit gates. A CNOT circuit only consists of CNOT gates.

(2) the synthesis of CNOT circuits helps optimize the quantum circuits.

Improve

both # gates and the fidelity.

3. What kinds of architecture do we focus on?

superconducting qubit-based quantum computing devices.

II. Preliminaries

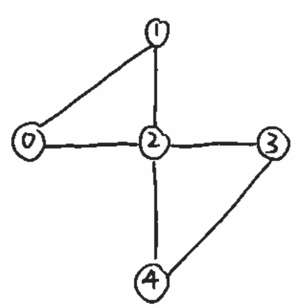
1. Logical qubits: the qubits that participate in a quantum algorithm. A quantum algorithm is represented by a logical quantum circuit. Each logical qubit is represented by q_i, $i \in N$.

2. Physical qubits: the qubits on a quantum computer. Each physical qubit is represented by Q_i, $i \in N$.

3. Initial mapping: one-to-one relationships between the logical and physical qubits.

In this context, we ignore the layer of QEC.

4. IBMQ_YORKTOWN coupling graph:



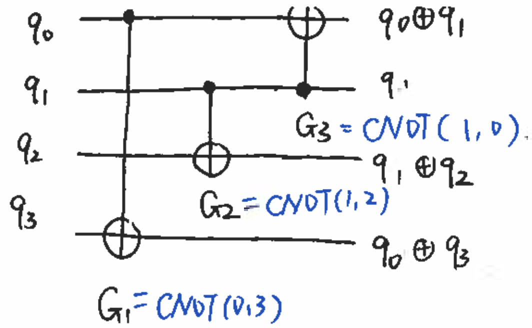
(1) What does each vertex mean?

Each vertex corresponds to a physical qubit register on the hardware.

(2) What does each edge mean?

It denotes that the qubits connect to it can interact directly. Only when the CNOT gate on the logical quantum circuit is placed on the physically adjacent physical qubits can it be successfully implemented.

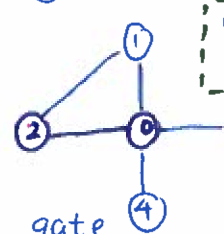
Example 1



(1) Suppose we perform the mapping

$q_i - Q_i \quad 0 \le i \le 4$.

(2) G_1 is not an NN CNOT gate because physical qubits 0 and 3 are not adjacent. This means G_1 cannot be successfully implemented.

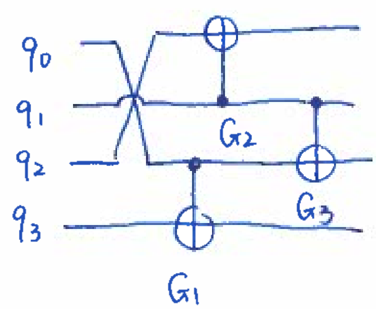


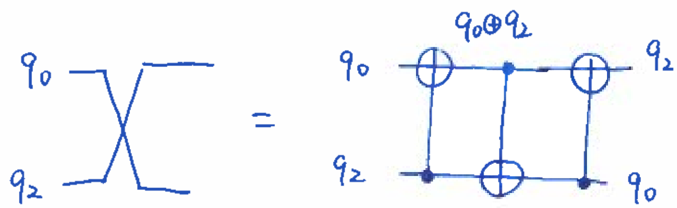
$q_0 - Q_2$
$q_2 - Q_0$

(3) How to convert G_1 to an NN CNOT gate?

Approach 1: Alter the initial mapping procedure

Approach 2: Insert the SWAP gate in front of the G_1 gate





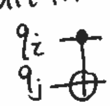
5. Boolean matrix / Parity matrix: $n \times n$ invertible matrix over $\mathbb{F}_2$

(1) The ith row represents the output parity on the ith wire.

(2) The jth column represents the jth input qubit.

$$\begin{array}{c|cccc} & q_0 & q_1 & q_2 & q_3 \\ \hline q_0' & 1 & 1 & 0 & 0 \\ q_1' & 0 & 1 & 0 & 0 \\ q_2' & 0 & 1 & 1 & 0 \\ q_3' & 1 & 0 & 0 & 1 \end{array} = M'$$

For the circuit in **Example 1**, write its parity matrix



$$I = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ 0 & 0 & \ddots & \vdots \\ \vdots & \vdots & & \\ 0 & 0 & \cdots & 1 \end{bmatrix} \xrightarrow[R_j \leftarrow R_i \oplus R_j]{CNOT(i,j)} \begin{array}{c|ccccccc} & 0 & \cdots & i & \cdots & j & \cdots & n-1 \\ \hline 0 & 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & & \vdots & & \vdots \\ i & 0 & \cdots & 1 & \cdots & 0 & \cdots & 0 \\ \vdots & \vdots & & & \ddots & \vdots & & \vdots \\ j & 0 & \cdots & 1 & \cdots & 1 & \cdots & 0 \\ \vdots & \vdots & & \vdots & & \vdots & \ddots & \\ n-1 & 0 & \cdots & 0 & \cdots & 0 & \cdots & 1 \end{array}$$

To generate M' from G_1, G_2, and G_3, perform the inverse product of the Boolean matrix of G_i.

G_3: $CNOT(1,0) \rightarrow M_3 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ G_2: $CNOT(1,2) \rightarrow M_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ G_1: $CNOT(0,3) \rightarrow M_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}$

$M' = M_3 M_2 M_1$

$$M_1 I = M_1 = \begin{bmatrix} 1&0&0&0 \\ 0&1&0&0 \\ 0&0&1&0 \\ 1&0&0&1 \end{bmatrix}$$

$$M_2 M_1 = \begin{bmatrix} 1&0&0&0 \\ 0&1&0&0 \\ 0&1&1&0 \\ 0&0&0&1 \end{bmatrix} \begin{bmatrix} 1&0&0&0 \\ 0&1&0&0 \\ 0&0&1&0 \\ 1&0&0&1 \end{bmatrix} = \begin{bmatrix} 1&0&0&0 \\ 0&1&0&0 \\ 0&1&1&0 \\ 1&0&0&1 \end{bmatrix}$$

$$M_3 M_2 M_1 = \begin{bmatrix} 1&1&0&0 \\ 0&1&0&0 \\ 0&0&1&0 \\ 0&0&0&1 \end{bmatrix} \begin{bmatrix} 1&0&0&0 \\ 0&1&0&0 \\ 0&1&1&0 \\ 1&0&0&1 \end{bmatrix} = \begin{bmatrix} 1&1&0&0 \\ 0&1&0&0 \\ 0&1&1&0 \\ 1&0&0&1 \end{bmatrix} = M'.$$

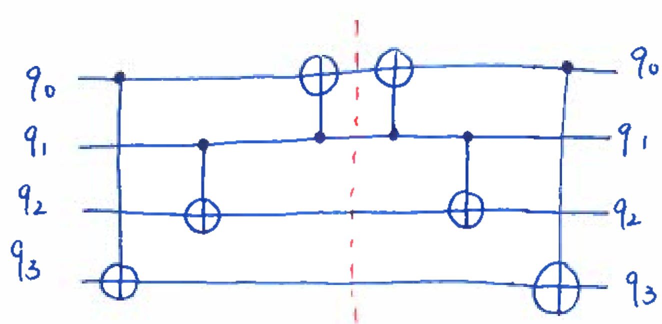
checked

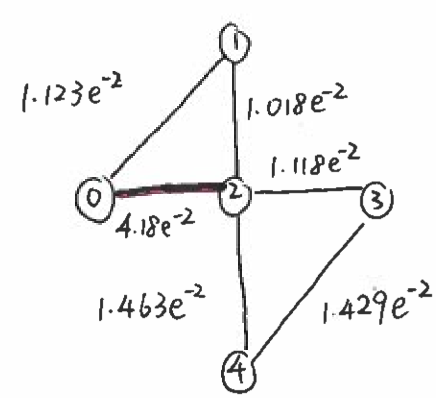
$$M_3 M' = \begin{bmatrix} 1&1&0&0 \\ 0&1&0&0 \\ 0&0&1&0 \\ 0&0&0&1 \end{bmatrix} \begin{bmatrix} 1&1&0&0 \\ 0&1&0&0 \\ 0&1&1&0 \\ 1&0&0&1 \end{bmatrix} = \begin{bmatrix} 1&0&0&0 \\ 0&1&0&0 \\ 0&1&1&0 \\ 1&0&0&1 \end{bmatrix} = M_2 M_1$$

$$M_2(M_2 M_1) = \begin{bmatrix} 1&0&0&0 \\ 0&1&0&0 \\ 0&1&1&0 \\ 0&0&0&1 \end{bmatrix} \begin{bmatrix} 1&0&0&0 \\ 0&1&0&0 \\ 0&1&1&0 \\ 1&0&0&1 \end{bmatrix} = \begin{bmatrix} 1&0&0&0 \\ 0&1&0&0 \\ 0&0&1&0 \\ 1&0&0&1 \end{bmatrix} = M_1$$

$$M_1 M_1 = \begin{bmatrix} 1&0&0&0 \\ 0&1&0&0 \\ 0&0&1&0 \\ 1&0&0&1 \end{bmatrix} \begin{bmatrix} 1&0&0&0 \\ 0&1&0&0 \\ 0&0&1&0 \\ 1&0&0&0 \end{bmatrix} = \begin{bmatrix} 1&0&0&0 \\ 0&1&0&0 \\ 0&0&1&0 \\ 0&0&0&1 \end{bmatrix} = I$$

$M_1 M_2 M_3 M' = I$





6. What is CNOT error?

It is the error rate of operations between two adjacent qubits.

Also known as the interaction error. This error is of order of magnitude more than a single-qubit operation error.

III Gate error-aware CNOT circuit NN synthesis

Def 1 On a quantum device, a CNOT gate is applied to the physical qubit Q_i and Q_j

$CNOT(Q_i, Q_j)$

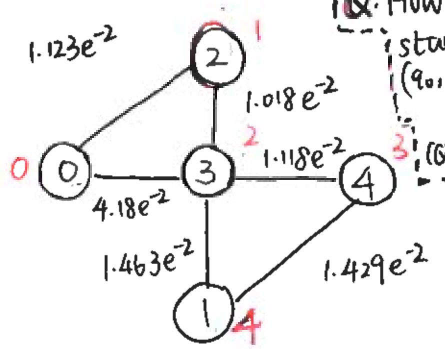
$$Set(Q_i, Q_j) = Q_i \oplus Q_j,\ Q_i = 1,\ Q_j = 0 \rightarrow Q_j' = 1$$

$$Del(Q_i, Q_j) = Q_i \oplus Q_j,\ Q_i = 1,\ Q_j = 1 \rightarrow Q_j' = 0.$$

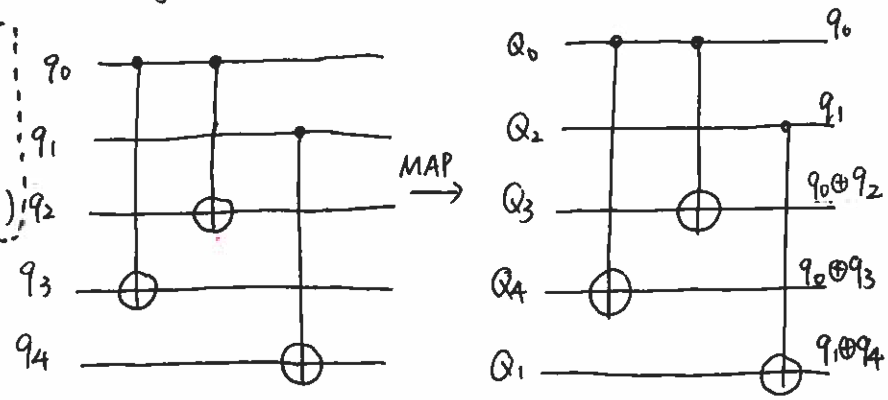
Def 2 The CNOT error is used as the edge weights of the architecture diagram $G = (V, E)$. The minimum spanning tree is defined as a spanning tree that covers a set of required nodes $V_n \subseteq V$ and has the $\boxed{\text{lowest sum of edge weights}}$. Steiner nodes are the additional added vertices $V_s \subseteq V$ and $V_n \cap V_s = \phi$. The tree is denoted as $ST(V_0, V_n, V_s)$, where V_0 is the root node.

Q: Why this is not the multiplication of error rate?
A: $\prod \varepsilon_i \propto \sum \varepsilon_i$? And $\sum \varepsilon_i$ corresponds to existing algorithm?

(1) The initial mapping. q_i logical, Q_j physical, $0 \le i, j \le 4$



Q: How to understand $(q_0, q_1, q_2, q_3, q_4)$
$\downarrow$
$(Q_0, Q_2, Q_3, Q_4, Q_1)$?



CNOT logic quantum circuit

$$\begin{array}{c|c}
 & \begin{array}{ccccc} q_0 & q_1 & q_2 & q_3 & q_4 \end{array} \\
\begin{array}{c} Q_0 \\ Q_2 \\ Q_3 \\ Q_4 \\ Q_1 \end{array} & \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{bmatrix} \\
 & \begin{array}{ccccc} C_1 & C_2 & C_3 & C_4 & C_5 \end{array}
\end{array}$$

The initial Boolean matrix

For C_1

$$\begin{array}{c} Q_0 \\ Q_2 \\ Q_3 \\ Q_4 \\ Q_1 \end{array} \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

0 — 3 — 4

CX_{34} CX_{03}

(2) Generate a weighted Steiner tree based on the topology above and the CNOT error rate Below we try to reconstruct P and D using a naive traversal approach, because Algorithm 1 is unreadable.

Step 1: Fill in the first row of P and D.

$$P = \begin{array}{c|ccccc} & Q_0 & Q_2 & Q_3 & Q_4 & Q_1 \\ \hline Q_0 & [\ ] & [(0,2)] & \begin{bmatrix}(0,2)\\(2,3)\end{bmatrix} & \begin{bmatrix}(0,2)\\(2,3)\\(3,4)\end{bmatrix} & \begin{bmatrix}(0,2)\\(2,3)\\(3,1)\end{bmatrix} \\ \vdots & & & & & \end{array}$$

$$D = \begin{array}{c|ccccc} & Q_0 & Q_2 & Q_3 & Q_4 & Q_1 \\ \hline Q_0 & 0 & 1.123e^{-2} & 2.141e^{-2} & 3.259e^{-2} & 3.604e^{-2} \\ \vdots & & & & & \end{array}$$

Path	Error rate	
Q_0-Q_2	0.01	✓
$Q_0-Q_3-Q_2$	0.049	

1.123×10^{-2}
$= 1.123e^{-2}$ ✓

Path	Error rate	
Q_0-Q_3	0.04	
$Q_0-Q_2-Q_3$	0.019	✓

2.141×10^{-2}
$=(1.123+1.018)\times10^{-2}$
$= 2.141e^{-2}$

Path	Error rate
$Q_0-Q_3-Q_4$	$0.04+0.01=0.05$
$Q_0-Q_2-Q_3-Q_4$	$0.01+0.009+0.01=0.029$ ✓
$Q_0-Q_3-Q_1-Q_4$	$0.04+0.014+0.013=0.06$
$Q_0-Q_2-Q_3-Q_1-Q_4$	X

3.259×10^{-2}
$= 1.123+1.018+1.118$
$= 3.259e^{-2}$

Path	Error rate	
$Q_0-Q_3-Q_1$	$0.04+0.014=0.054$	
$Q_0-Q_3-Q_4-Q_1$	X	
$Q_0-Q_2-Q_3-Q_1$	$0.01+0.009+0.014=0.033$	✓ $\rightarrow 3.604e^{-2}$
$Q_0-Q_2-Q_3-Q_4-Q_1$	X	

$= 1.123+1.018+1.463 = 3.604e^{-2}$

* Since P and D are symmetric, we only need to fill in the upper triangle part of both matrices. Also, we don't need to worry about diagonals because they are set to 1.

Step 2: Fill in the second row of the upper triangular component of P and D.

$$P = \begin{array}{c|ccc} & Q_3 & Q_4 & Q_1 \\ \hline \vdots & \vdots & \vdots & \vdots \\ Q_2 & [(2,3)] & \begin{bmatrix}(2,3)\\(2,4)\end{bmatrix} & \begin{bmatrix}(2,3)\\(3,1)\end{bmatrix} \\ \vdots & & & \end{array}$$

$$D = \begin{array}{c|ccc} & Q_3 & Q_4 & Q_1 \\ \hline \vdots & \vdots & \vdots & \vdots \\ Q_2 & 1.108e^{-2} & 2.136e^{-2} & 2.481e^{-2} \\ \vdots & & & \end{array}$$

$= 1.018+1.118 = 2.136e^{-2}$ $\qquad = 1.018+1.463 = 2.481e^{-2}$

Path	Error rate	
Q_2-Q_3	0.009	✓ $\rightarrow 1.108e^{-2}$
$Q_2-Q_0-Q_3$	$0.01+0.04=0.05$	

Path	Error rate	
$Q_2-Q_3-Q_4$	0.019	✓ $\rightarrow 2.136e^{-2}$
$Q_2-Q_3-Q_1-Q_4$	0.036	

we use the previous knowledge to exclude: $Q_2-Q_0-Q_3\cdots$

Path	Error rate	
$Q_2-Q_3-Q_1$	0.023	✓ $\rightarrow 2.481e^{-2}$
$Q_2-Q_3-Q_4-Q_1$	0.032	

Step 3: Fill in the third row of the upper triangular component of P and D

$$P = \begin{array}{c|cc} & \cdots\ Q_4 & Q_1 \\ \vdots & \vdots & \vdots \\ Q_3 & [(3,4)] & [(3,1)] \\ \vdots & & \end{array} \qquad D = \begin{array}{c|cc} & \cdots\ Q_4 & Q_1 \\ \vdots & \vdots & \\ Q_3 & 1.118e^{-2} & 1.463e^{-2} \\ \vdots & & \end{array}$$

Path	Error rate
$Q_3 - Q_4$	$0.01 \nearrow 1.118e^{-2}$ ✓
$Q_3 - Q_1 - Q_4$	$0.014 + 0.013 = 0.027$

Path	Error rate
$Q_3 - Q_1$	$0.014 \nearrow 1.463e^{-2}$ ✓
$Q_3 - Q_4 - Q_1$	0.023

Step 4: Fill in the fourth row of the upper triangular component of P and D

$$P = \begin{array}{c|c} & \cdots\ Q_1 \\ \vdots & \vdots \\ Q_4 & [(4,1)] \\ \vdots & \end{array} \qquad D = \begin{array}{c|c} & \cdots\ Q_1 \\ \vdots & \vdots \\ Q_4 & 1.429e^{-2} \\ \vdots & \end{array}$$

Path	Error rate
$Q_4 - Q_1$	$0.013 \nearrow 1.429e^{-2}$ ✓
$Q_4 - Q_3 - Q_1$	0.024

Q: How is the last pagraph on page 8 associated to the above naive analysis?

Def 3 The overall error of the NN circuit is defined as <u>the sum of the products of the</u> CNOT gates applied on each weighted Steiner tree during the error-aware CNOT circuit NN synthesis process.

Q: What does this mean? An example?

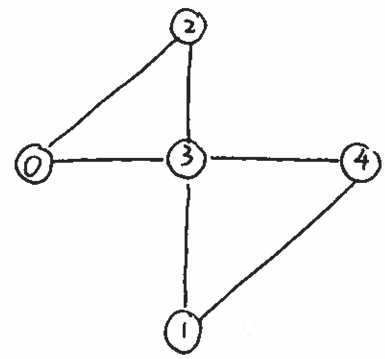
Next we use P, D, and M (shown on the next page) to perform the gate error-aware CNOT circuit NN synthesis

Example:

$$M = \begin{array}{c|ccccc} & C_1 & C_2 & C_3 & C_4 & C_5 \\ \hline Q_0 & 1 & 0 & 0 & 0 & 0 \\ Q_2 & 0 & 1 & 0 & 0 & 0 \\ Q_3 & 1 & 0 & 1 & 0 & 0 \\ Q_4 & 1 & 0 & 0 & 1 & 0 \\ Q_1 & 0 & 1 & 0 & 0 & 1 \end{array}$$

$$P = \begin{array}{c|ccccc} & Q_0 & Q_2 & Q_3 & Q_4 & Q_1 \\ \hline Q_0 & [\ ] & [(0,2)] & [(0,2),(2,3)] & [(0,2),(2,3),(3,4)] & [(0,2),(2,3),(3,1)] \\ Q_2 & & [\ ] & [(2,3)] & [(2,3),(3,4)] & [(2,3),(3,1)] \\ Q_3 & & & [\ ] & [(3,4)] & [(3,1)] \\ Q_4 & & & & [\ ] & [(4,1)] \\ Q_1 & & & & & [\ ] \end{array}$$

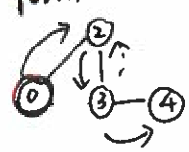
$$D = \begin{array}{c|ccccc} & Q_0 & Q_2 & Q_3 & Q_4 & Q_1 \\ \hline Q_0 & 0 & 1.123e^{-2} & 2.141e^{-2} & 3.259e^{-2} & 3.604e^{-2} \\ Q_2 & & 0 & 1.018e^{-2} & 2.136e^{-2} & 2.481e^{-2} \\ Q_3 & & & 0 & 1.118e^{-2} & 1.463e^{-2} \\ Q_4 & & & & 0 & 1.429e^{-2} \\ Q_1 & & & & & 0 \end{array}$$



< Stage 1: Reduce M to an upper triangular form.

Step 1: Reduce the first column.

Consider $C_1 = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} \begin{matrix} Q_0 \\ \\ Q_3 \\ Q_4 \\ \\ \end{matrix}$



$\varepsilon_1 \quad \varepsilon_2 \quad \varepsilon_3 \quad \varepsilon_4$

$CX_{32}\ CX_{34}\ CX_{23}\ CX_{02}$

Q: How are errors accummulated?

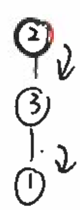
A: $1 - \underbrace{(1-\varepsilon_1)(1-\varepsilon_2)(1-\varepsilon_3)(1-\varepsilon_4)}_{\text{when the circuit has no error}}$

$$\begin{array}{c} Q_0 \\ Q_2 \\ Q_3 \\ Q_4 \\ Q_1 \end{array} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{bmatrix} \xrightarrow[CX_{34}]{CX_{32}} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{bmatrix} \xrightarrow{CX_{23}} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{bmatrix} \xrightarrow{CX_{02}}$$

$$\begin{array}{c} Q_0 \\ Q_2 \\ Q_3 \\ Q_4 \\ Q_1 \end{array} \begin{array}{c} \begin{matrix} & C_2 & & & \end{matrix} \\ \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{bmatrix} \end{array}$$

Step 2: Reduce the second column

$C_2 = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 1 \end{bmatrix} \begin{matrix} \\ Q_2 \\ Q_3 \\ \\ Q_1 \end{matrix}$



$CX_{31}\ CX_{23}$

$$\xrightarrow{CX_{31}} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \xrightarrow{CX_{23}} \begin{array}{c} \begin{matrix} & & C_3 & & \end{matrix} \\ \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \end{array} \begin{array}{c} Q_0 \\ Q_2 \\ Q_3 \\ Q_4 \\ Q_1 \end{array} \xrightarrow{CX_{34}}$$

Step 3: Reduce the third column

$C_3 = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \\ 0 \end{bmatrix} \begin{matrix} \\ \\ Q_3 \\ Q_4 \\ \\ \end{matrix}$

CX_{34}

$$
M' = \begin{array}{c} \begin{array}{ccccc} & & c_3 & & \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \end{array}
$$

Stage 2: Reduce M' to the identity.

Step 1: Reduce the third column

Consider $c_3 = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}$ $\qquad 3 \text{ — } 2$ $\quad CX_{32}$

$M' \to I$.

In total, we have used eight NN CNOTs:

$$
\begin{array}{cccccccc}
CX_{32} & CX_{34} & CX_{23} & CX_{02} & CX_{31} & CX_{23} & CX_{34} & CX_{32} \\
\downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\
\varepsilon_{32} & \varepsilon_{34} & \varepsilon_{23} & \varepsilon_{02} & \varepsilon_{31} & \varepsilon_{23} & \varepsilon_{34} & \varepsilon_{32}
\end{array}
$$

$$
\varepsilon_{total} = 1 - \prod_{(i,j) \in reg} (1 - \varepsilon_{ij}), \text{ where } reg = \{(3,2), (3,4), (2,3), (0,2), (3,1), (2,3), (3,4), (3,2)\}
$$

Next, we read ε_{ij} where $(i,j) \in Reg$ from D and compute ε_{total}.

$$
\varepsilon_{total} = 1 - (1 - 1.018e^{-2})^4 \times (1 - 1.18 \times 10^{-2})^2 \times (1 - 1.123 \times 10^{-2}) \times (1 - 1.413 \times 10^{-2}) \simeq 0.085563
$$