Quantum Error Correction using the Surface Code

Kwanak Mountain Noru Jumping

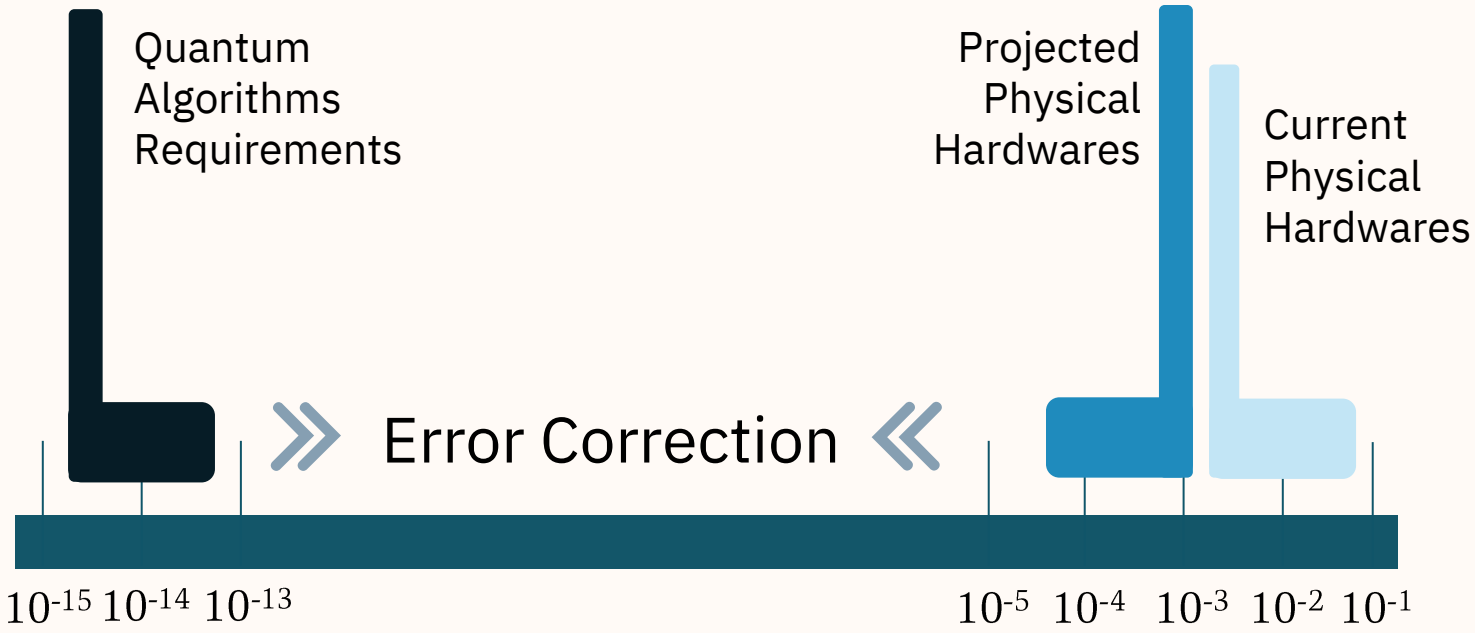
Jaewon Jung[1], Giwon Song[2], Shinyoung Hwang[3, 1], Minwoo Kim[1, 4]
1) Department of Physics & Astronomy, Seoul National University
2) Department of Electric & Computer Engineering, Seoul National University
3) Department of Material Science & Engineering, Seoul National University
4) Department of Computer Science & Engineering, Seoul National University

Contents

01

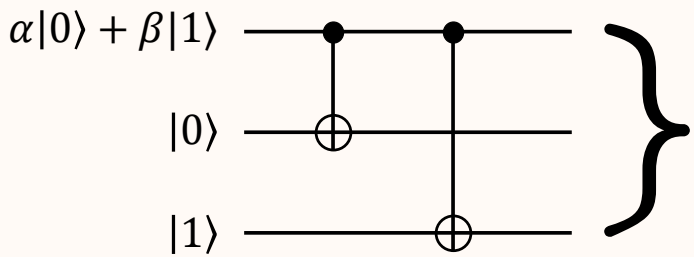
What is the Surface code?

Need for error correction



Error correction: Repetition Code

Define multiple **physical states** of equivalence class to represent a certain **Logical state**



Physical state	Logical state
$\alpha\|000\rangle + \beta\|111\rangle$	$\alpha\|0\rangle_L + \beta\|1\rangle_L$
$\alpha\|001\rangle + \beta\|110\rangle$	robustness against bit-flip error
$\alpha\|011\rangle + \beta\|100\rangle$	$\alpha\|1\rangle_L + \beta\|0\rangle_L$

equivalence class

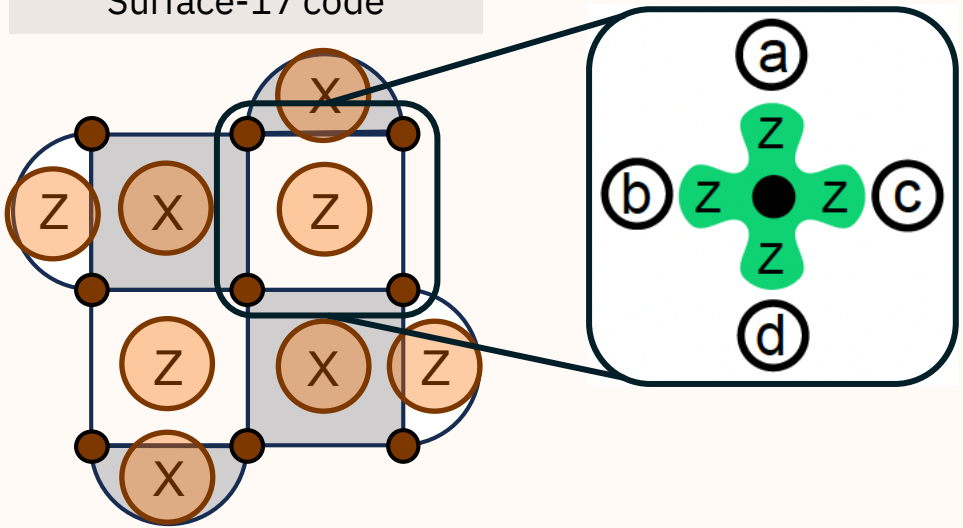
$$|0\rangle_L : \{|000\rangle, |001\rangle, |010\rangle, |100\rangle\}$$

$$|1\rangle_L : \{|111\rangle, |110\rangle, |101\rangle, |011\rangle\}$$

Error correction: Surface Code

Use stabilizer qubits for **non-destructive parity check**

Surface-17 code

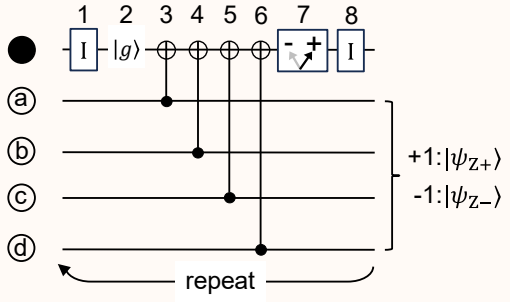


data qubits

stabilizer qubits

Z stabilizer qubit

Surface code cycle



Stabilizes $Z_a Z_b Z_c Z_d$

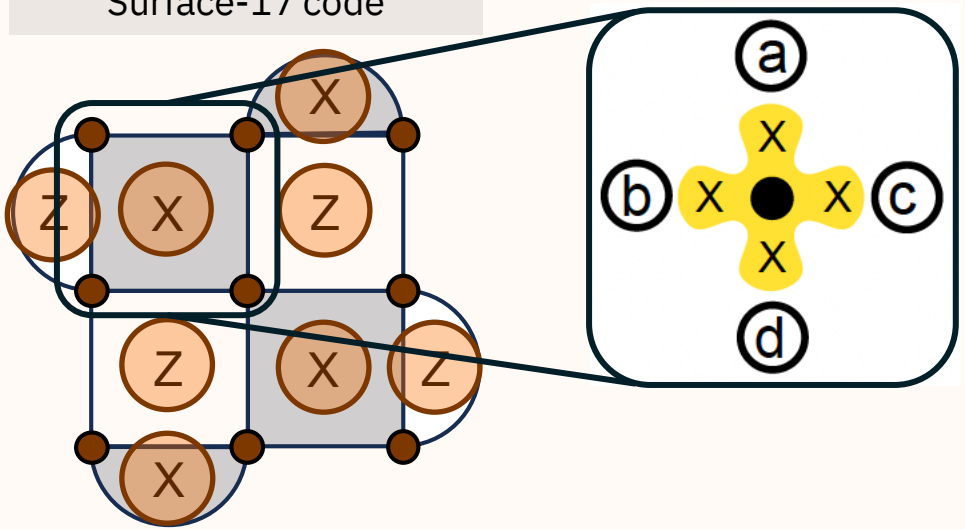
→ Measures parity of the surrounding qubits

Plaquette syndrome

Error correction: Surface Code

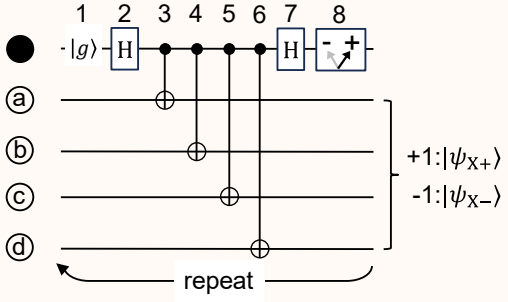
Use stabilizer qubits for **non-destructive parity check**

Surface-17 code



X stabilizer qubit

Surface code cycle



Stabilizes $X_a X_b X_c X_d$

→ Measures parity of the surrounding qubits

Vertex syndrome

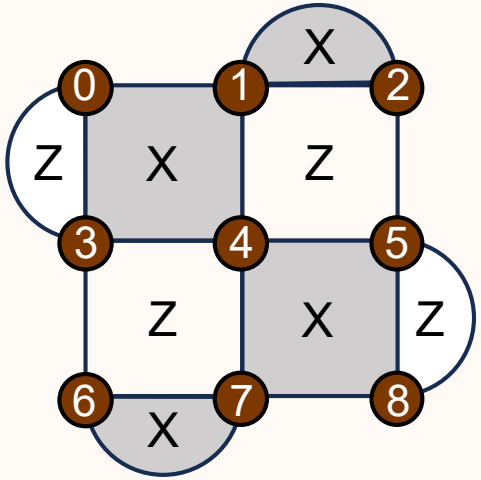
data qubits

stabilizer qubits

Error correction: Surface Code

Use stabilizer qubits for **non-destructive parity check**

Surface-17 code



data qubits

Stabilizer Group

Vertex	Plaquette
X_1X_2	Z_0Z_3
$X_0X_1X_3X_4$	$Z_1Z_2Z_4Z_5$
$X_4X_5X_7X_8$	$Z_3Z_4Z_6Z_7$
X_6X_7	Z_5Z_8

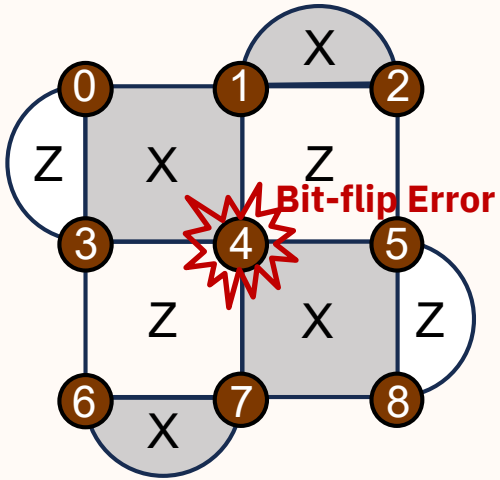
Bit-filp errors

Phase-flip errors

Error correction: Surface Code

Use stabilizer qubits for **non-destructive parity check**

Surface-17 code



data qubits

Vertex Syndromes (Relative)

$X_1X_2 \quad +1$

$X_0X_1X_3X_4 \quad -1$

$X_4X_5X_7X_8 \quad -1$

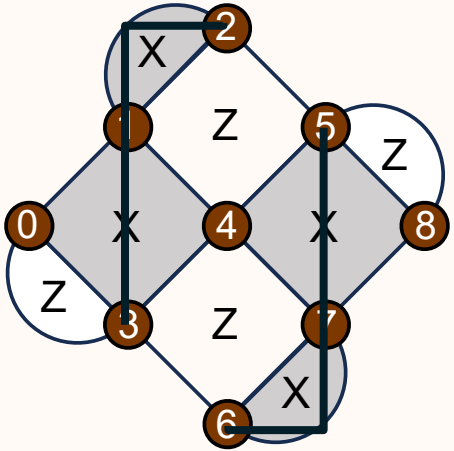
Bit flip on 4

$X_6X_7 \quad +1$

Error correction: Surface Code

Logical States

Surface-17 code



data qubits

States used for encoding: Quiescent States

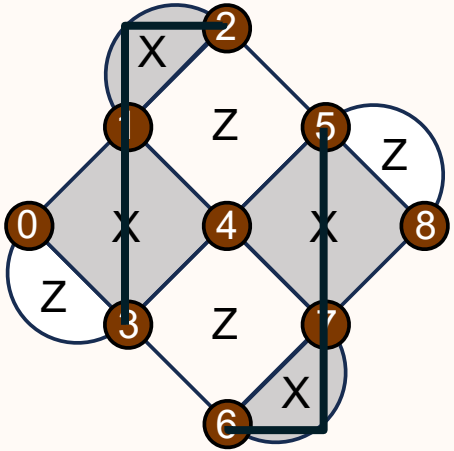
Mutual +1 eigenstates of the stabilizers

Remains the same on syndrome measurements
-> Mutual eigenstates of ZZZZ and XXXX

Error correction: Surface Code

Logical States

Surface-17 code



data qubits

States used for encoding: Quiescent States

Mutual +1 eigenstates of the stabilizers

How to get $|0\rangle_L$?

1. Initialize to any state $(e.g. |0\rangle^{\otimes 9})$
2. Measure syndromes
3. Go through decoding algorithm

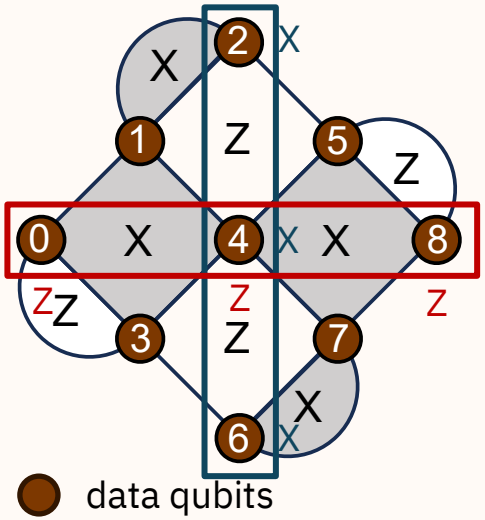
→ Sets the eigenvalues of the stabilizers to +1

→ Becomes the **desired** $|0\rangle_L$

Error correction: Surface Code

Logical Operations

Surface-17 code



Logical X Operation

: swap Z parities of the horizontal lines

→ X gate on a vertical line that goes across the data qubits

Logical Z Operation

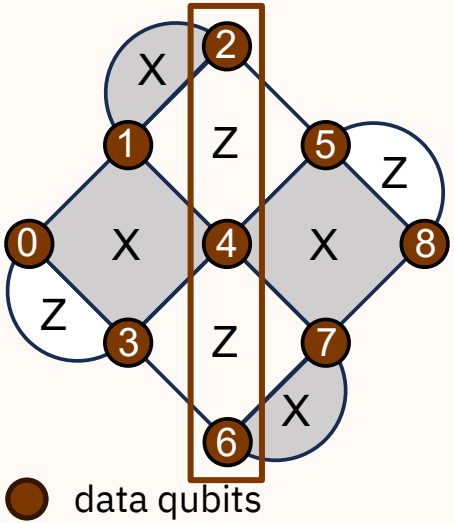
: swap X parities of the vertical lines

→ Z gate on a horizontal line that goes across the data qubits

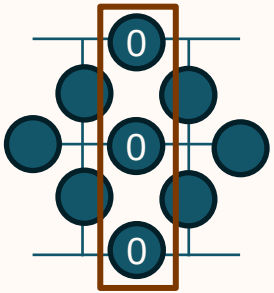
Error correction: Surface Code

Logical Readout: check **Z parity** of the data qubits along a vertical line

Surface-17 code

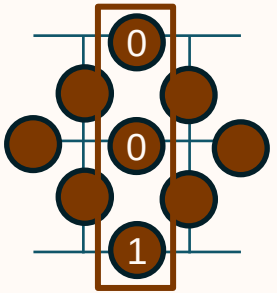


$|0\rangle_L$



Even

$|1\rangle_L$



Odd

02

Implementation of Surface code

Surface Code: Modularization

Initialization

Define QuantumCircuit parameters

Connect the qubits according to the surface code geometry

Support method for drawing

Initialize the state to logical zero

Surface Code: Modularization

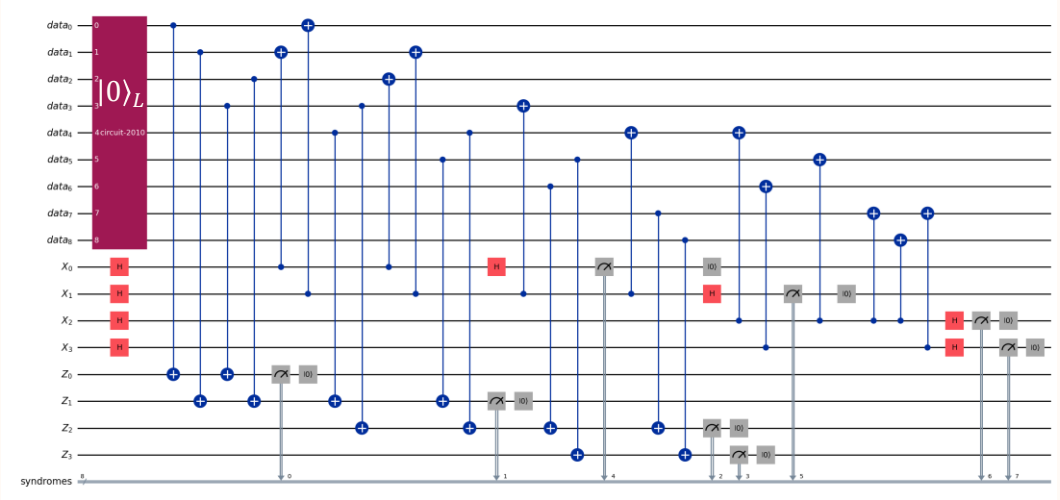
Initialization

Initialize the state to logical zero

$|0\rangle^{\otimes 9}$ ➡ Syndrome measurement & Decoding ➡ $|0\rangle_L$

Repeat 3 times

Verification



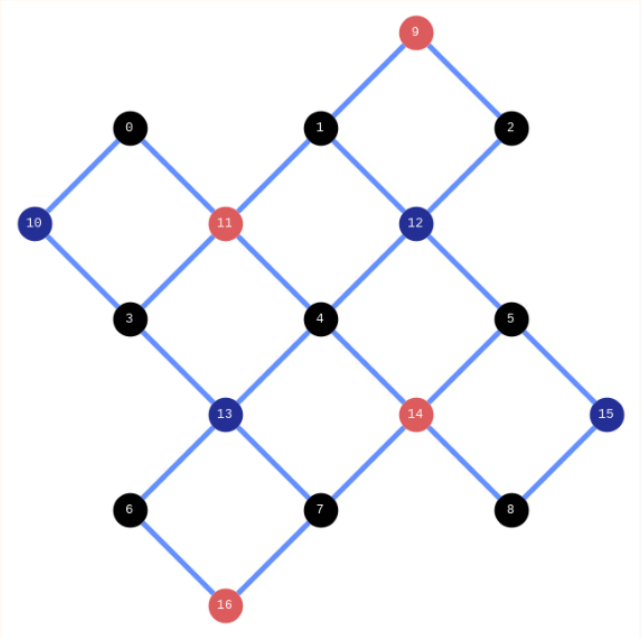
+1 Eigenstate

$|00000000\rangle$

Surface Code: Modularization

Initialization

Surface-17 code, generated with Qiskit



- Data qubit
- X syndrome qubit
- Z syndrome qubit

Surface Code: Modularization

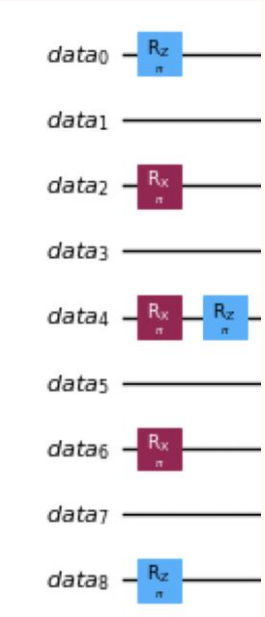
Logical Operations

Define methods for logical operations

Directly apply operations on the QuantumCircuit

Surface Code: Modularization

Logical Operations



Logical X, Z gates, generated with Qiskit

Surface Code: Modularization

Decoding

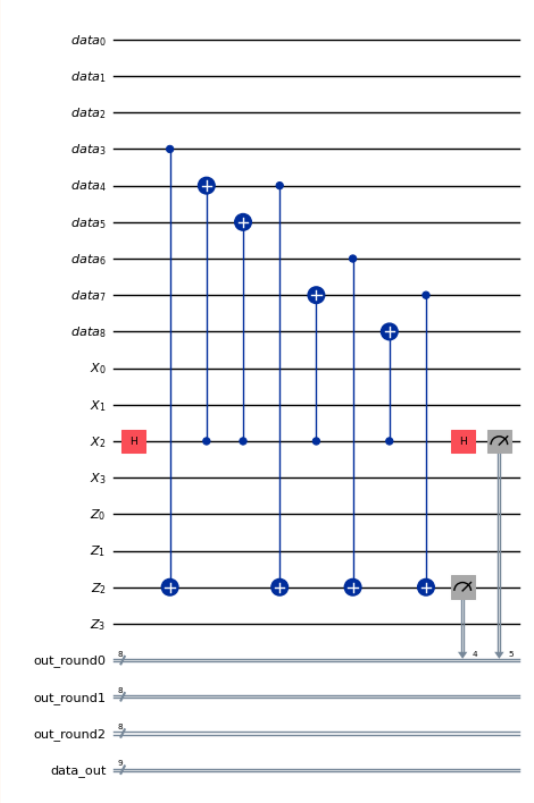
Measure vertex and plaquette error syndromes

Based on the syndromes, implement the **decoding algorithm**

Reset the stabilizer qubits

Surface Code: Modularization

Decoding



Syndrome measuring circuit, generated with Qiskit

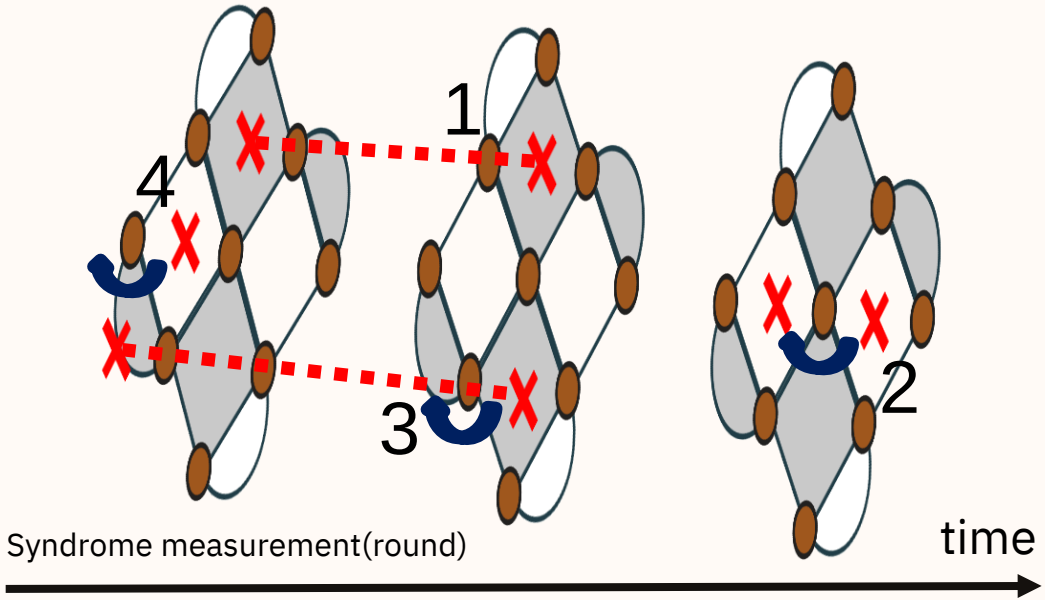
03

Decoding of the Surface code

Surface Code: Decoding Algorithm

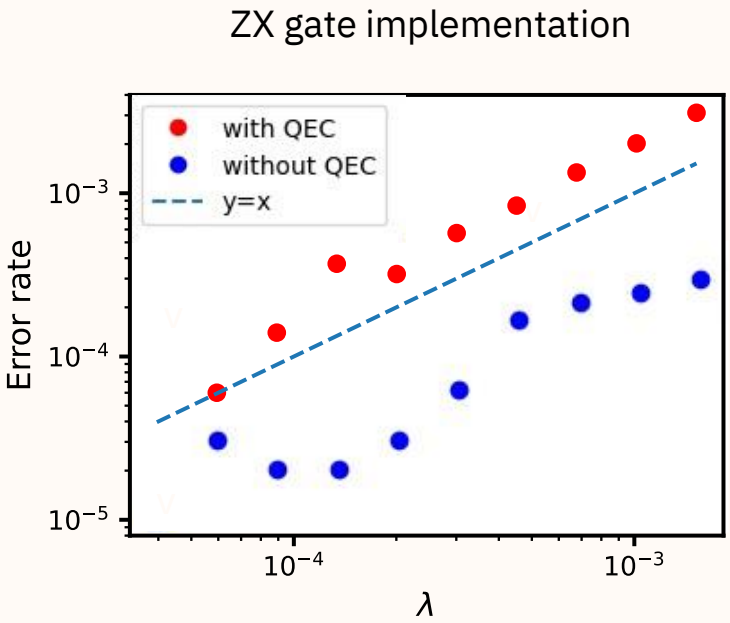
"Look-up table decoder"

Yu Tomita, Krysta M. Svore, "*Low-distance surface codes under realistic quantum noise*", PRA 90, 062320, 2014



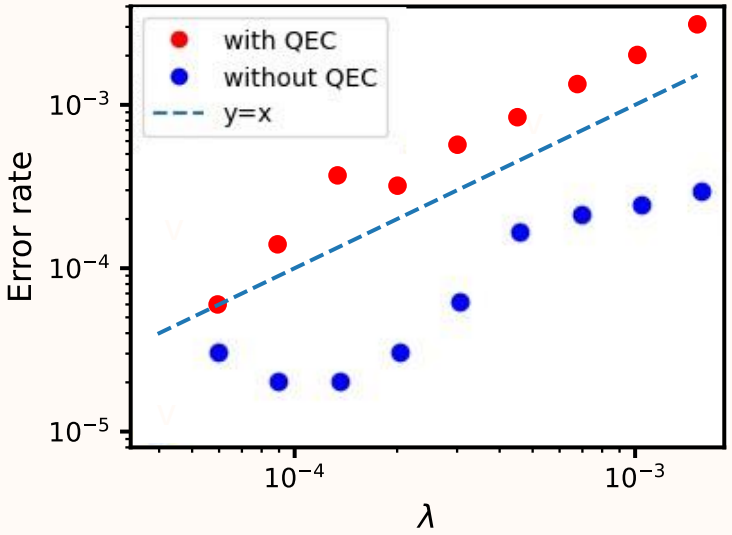
1. Subsequent error
→ measurement error
2. Simultaneous adjacent error
→ flip the data qubit between them
3. Subsequent adjacent error
→ flip the data qubit between them
4. Single error
→ flip the qubit at the boundary

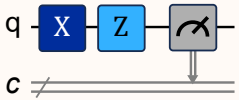
Surface Code: Result



Surface Code: Result

ZX gate implementation

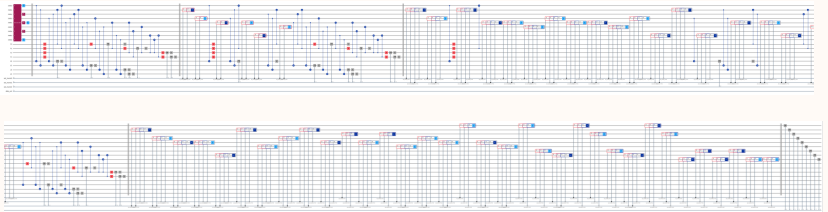




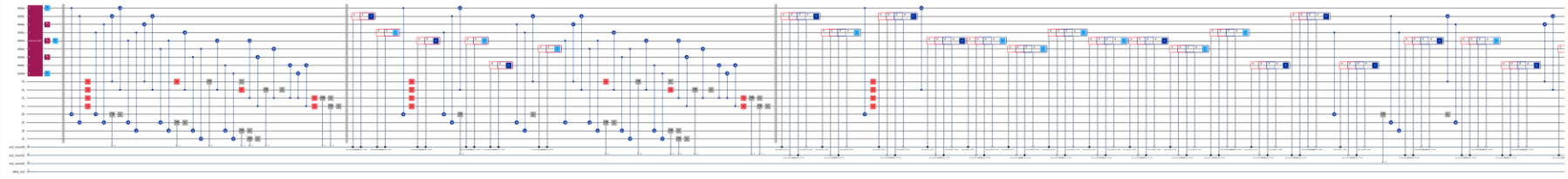
Physical Implementation

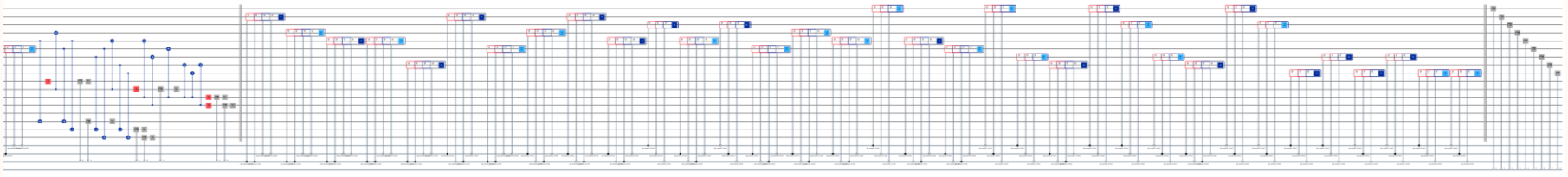
VS

Surface Code Implementation



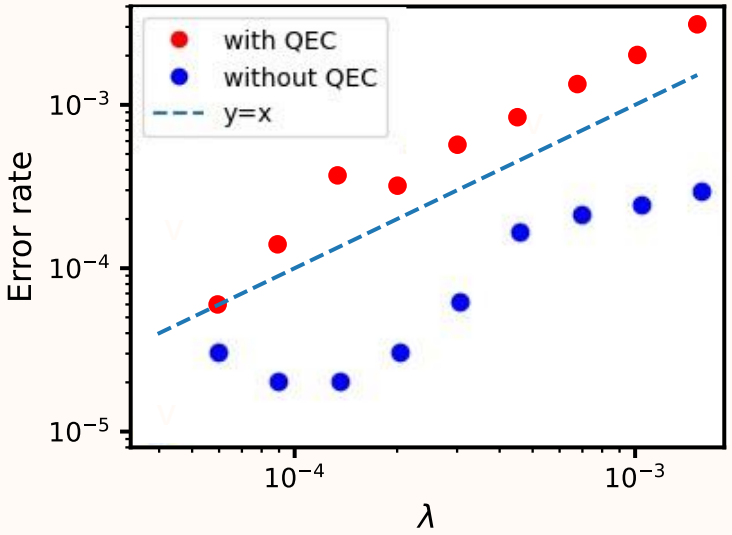
Surface Code: Result

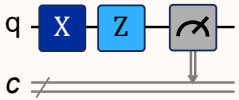




Surface Code: Result

ZX gate implementation

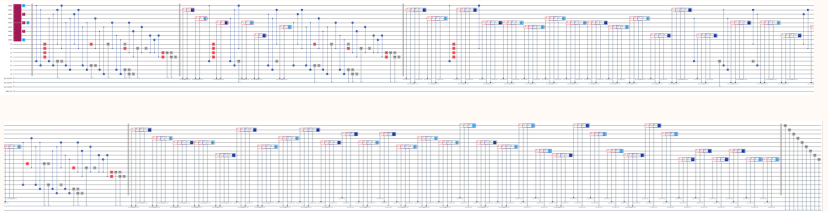




Physical Implementation

VS

Surface Code Implementation

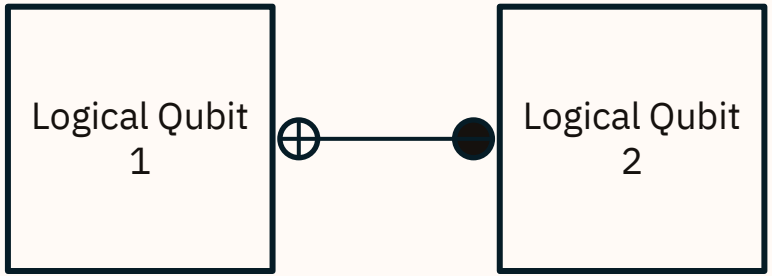


04

Logical Two-qubit Gates on Surface Code

Two Qubit Gate: Merge & Split

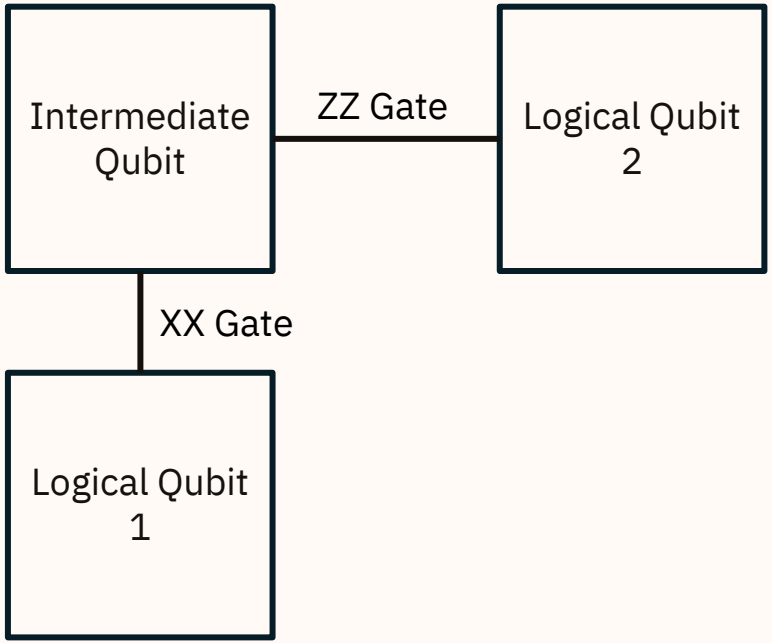
CNOT Gate implementation



No direct CNOT implementation

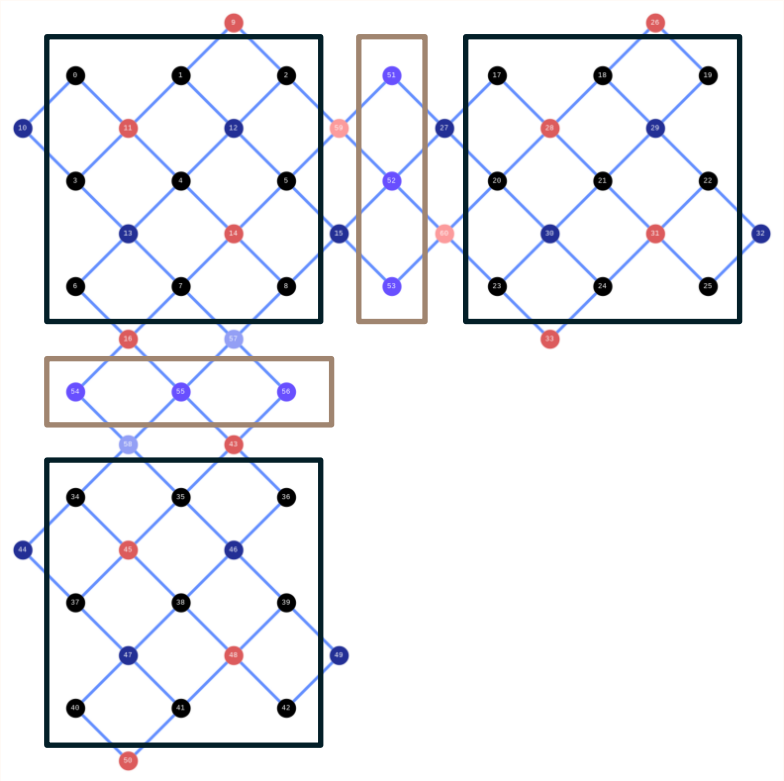
Two Qubit Gate: Merge & Split

CNOT Gate implementation



Two Qubit Gate: Merge & Split

CNOT Gate implementation



Kwanak Mountain Noru Jumping

Jaewon Jung[1], Giwon Song[2], Shinyoung Hwang[3, 1], Minwoo Kim[1, 4]

1) Department of Physics & Astronomy, Seoul National University
2) Department of Electric & Computer Engineering, Seoul National University
3) Department of Material Science & Engineering, Seoul National University
4) Department of Computer Science & Engineering, Seoul National University

Thank you