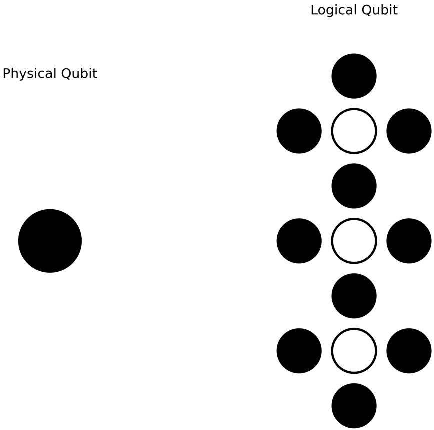
Surface Codes

Quantum Simulation - February 13th 2025

Michele Banfi 869294

Physical qubits vs Logical qubits



Introduction

In the surface code, **physical qubits are entangled together** using a sequence of physical qubit CNOT operations, with subsequent measurements of the entangled states providing a means for error correction and error detection. A set of physical qubits entangled in this way is used to define a *logical* qubit

This means that as long as errors can be detected promptly, they can be undone in classical software. [1]

Combining $\hat{X}$ and $\hat{Z}$ can detect errors, but

$$\left[\hat{X}, \hat{Z}\right] \neq 0 \tag{1}$$

Stabilizers

$\hat{Z}_a\hat{Z}_b$	$\hat{X}_a\hat{X}_b$	$\lvert\psi\rangle$
+1	−1	$\frac{\lvert gg\rangle + \lvert ee\rangle}{\sqrt{2}}$
+1	−1	$\frac{\lvert gg\rangle - \lvert ee\rangle}{\sqrt{2}}$
−1	+1	$\frac{\lvert ge\rangle + \lvert eg\rangle}{\sqrt{2}}$
−1	−1	$\frac{\lvert ge\rangle + \lvert eg\rangle}{\sqrt{2}}$

Table 1: Eigenstates of the two-qubit operators $\hat{Z}_a\hat{Z}_b$ and $\hat{X}_a\hat{X}_b$. The four eigenstates are the Bell states for this system.

$$\begin{aligned}
\left[\hat{X}_a\hat{X}_b, \hat{Z}_a\hat{Z}_b\right] &= \left(\hat{X}_a\hat{X}_b\right)\left(\hat{Z}_a\hat{Z}_b\right) - \left(\hat{Z}_a\hat{Z}_b\right)\left(\hat{X}_a\hat{X}_b\right) \\
&= \hat{X}_a\hat{Z}_a\hat{X}_b\hat{Z}_b - \hat{Z}_a\hat{X}_a\hat{Z}_b\hat{X}_b \\
&= \left(-\hat{Z}_a\hat{X}_a\right)\left(-\hat{Z}_b\hat{X}_b\right) - \hat{Z}_a\hat{X}_a\hat{Z}_b\hat{X}_b = 0
\end{aligned} \tag{2}$$

The surface code

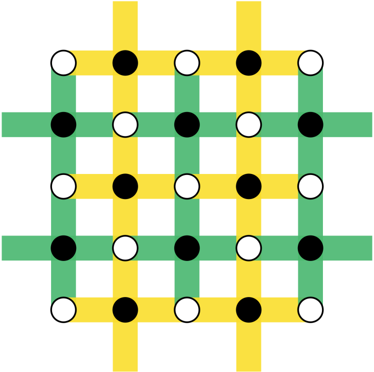


Figure 1: A surface code were black circles represents stabilizers and white circles data qubits

The surface code (ii)

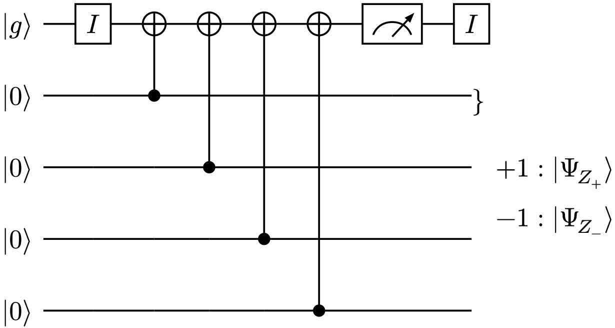
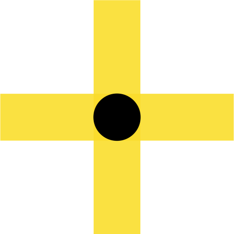


Figure 2: Measure-Z stabilizer

The surface code (iii)



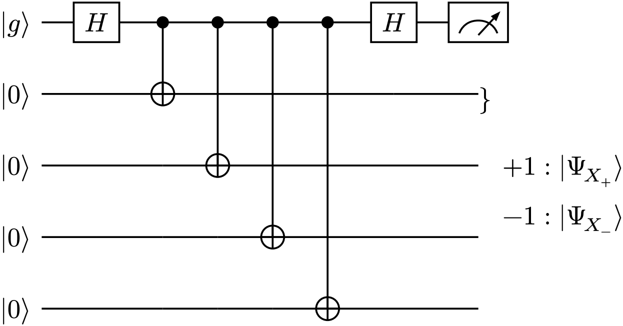
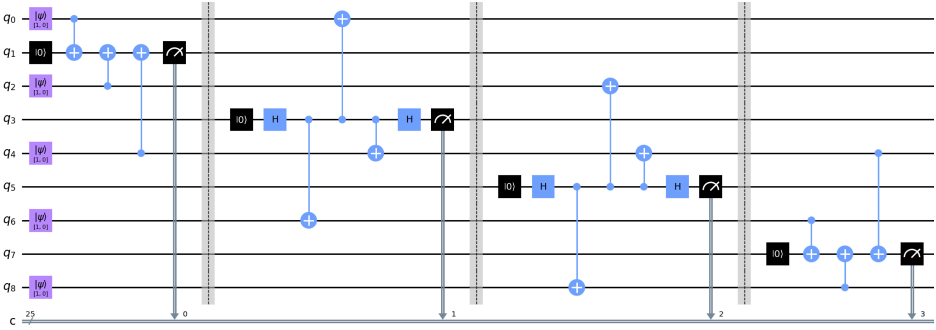


Figure 3: Measure-X stabilizer

The surface code (iv)



Quiescent state of the surface code

$$\hat{Z}_a \hat{Z}_b \hat{Z}_c \hat{Z}_d \left|\psi\right\rangle = Z_{abcd} \left|\psi\right\rangle \tag{3}$$

with eigenvalues $Z_{abcd} = \pm 1$

Stabilizer codes operate from the state $|\psi\rangle$, we call this the **quiescent state**. The quiescent state $|\psi\rangle$ is randomly selected by completing one full surface code cycle, which is the sequence starting, for example, with all data and measurement qubits in their ground states $|g\rangle$.

Single-Qubit errors

Once selected, a quiescent state remains unchanged except when disturbed by errors, for example, erroneous single-qubit $\hat{X}$ bit-flip or $\hat{Z}$ phase-flip operations.

e.g.

$$\hat{I}_a + \varepsilon \hat{Z}_a \tag{4}$$

operating on the data qubit a; $|\varepsilon| \ll 1$ is a small number equal to the probability amplitude for the $\hat{Z}$ phase flip. The error transform:

$$|\psi\rangle \to |\psi'\rangle = \left(\hat{I}_a + \varepsilon \hat{Z}_a\right)|\psi\rangle \tag{5}$$

In the next cycle $|\psi'\rangle$ will be projected to an eigenstate of all $\hat{X}_a \hat{X}_b \hat{X}_c \hat{X}_d$ and $\hat{Z}_a \hat{Z}_b \hat{Z}_c \hat{Z}_d$. Either projecting back to:

- $|\psi\rangle$
- $\hat{Z}_a\ |\psi\rangle$ with probability $|\varepsilon|^2$

Logical operators

The set of surface code stabilizers is actually not always complete, so the array can have additional degrees of freedom.

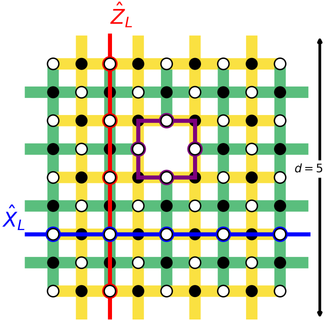


Figure 4: A surface code representing different logical operators

Logical operators (ii)

The two unconstrained degrees of freedom indicate that **this small array might serve as a (single) logical qubit**; we still need to define the logical operators that manipulate these degrees of freedom.

If we pair up operations on pairs of data qubits, we can create multiqubit operator products that commute with the stabilizers; this is how we build the logical operators, which we term $\hat{X}_L$ and $\hat{Z}_L$.

Consider, however, two simultaneous $\hat{X}$ operations on two data qubits a and b that both neighbor one measure-Z qubit. Then we have:

$$\begin{aligned} \hat{Z}_a\hat{Z}_b\hat{Z}_c\hat{Z}_d\left(\hat{X}_a\hat{X}_b \left|\psi\right\rangle\right) &= (-1)^2\hat{X}_a\hat{X}_b\left(\hat{Z}_a\hat{Z}_b\hat{Z}_c\hat{Z}_d \left|\psi\right\rangle\right) \\ &= Z_{abcd}\left(\hat{X}_a\hat{X}_b \left|\psi\right\rangle\right) \end{aligned} \tag{6}$$

Logical operators (iii)

Remembering Pauli operators:

$$\hat{X}\hat{Z} = -\hat{Z}\hat{X} \tag{7}$$

also $\hat{X}_L$ and $\hat{Z}_L$ satisfy the **anticommutation relation**:

$$\begin{aligned}
\hat{X}_L\hat{Z}_L &= \left(\hat{X}_1\hat{X}_2\hat{X}_3\hat{X}_4\hat{X}_5\right)\left(\hat{Z}_6\hat{Z}_7\hat{Z}_3\hat{Z}_8\hat{Z}_9\right) \\
&= \hat{X}_3\hat{Z}_3\left(\hat{X}_1\hat{X}_2\hat{X}_4\hat{X}_5\right)\left(\hat{Z}_6\hat{Z}_7\hat{Z}_8\hat{Z}_9\right) \\
&= -\hat{Z}_3\hat{X}_3\left(\hat{Z}_6\hat{Z}_7\hat{Z}_8\hat{Z}_9\right)\left(\hat{X}_1\hat{X}_2\hat{X}_4\hat{X}_5\right) \\
&= -\hat{Z}_L\hat{X}_L,
\end{aligned} \tag{8}$$

Physical error detection

The errors we consider here are on the physical qubits, either the data or the measure qubits. We can compare the errors of the stabilizers from one round to another, e.g.:

```
round 0:  0 0 1 0 1 1 0 1 0 0
round 1:  1 0 1 1 0 0 0 1 1 1
round 2:  0 0 1 1 0 0 0 1 0 0
round 3:  1 0 1 1 0 0 1 1 0 0
```

From one round to another we can see the bit flipping of physical qubits. The first round represent the quiescent state of the surface code. So the errors from one round to another are:

```
 0 to 1:  x . . x x x . . x x
 1 to 2:  x . . . . . . . x x
 2 to 3:  x . . . . . x . . .
```

Physical error detection (ii)

Individual errors of these types are, of course, the most likely, but concatenated errors can also occur, for example, two, three, or more adjacent data qubits suffering $\hat{X}$ errors in one surface code cycle, creating what are called error chains.

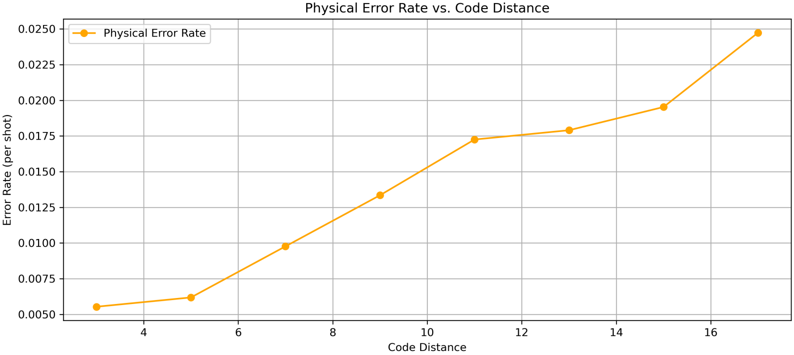


Figure 5: Physical error increase as code distance increase

Logical error detection

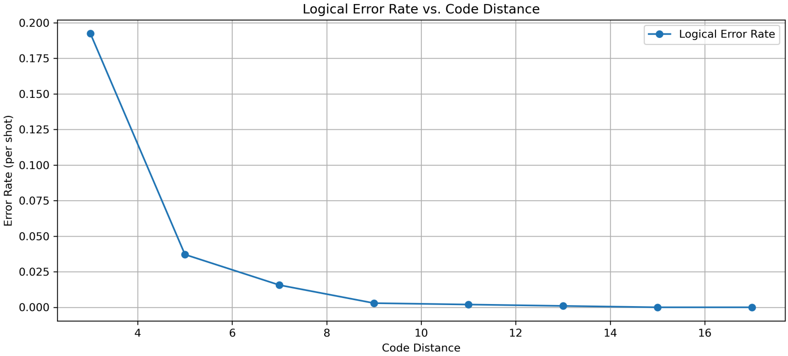
After detecting physical errors, we want to understand if those errors could create also a logical error, which occur when more than $d/2$ data qubits flips in the logical operator. To do so, we use the **Minimum Weight Perfect Matching** (MWPM) algorithm.

The algorithm create a weighted graph in which each node represents a syndrome and edges connect nodes that could be endpoints of a single error chain. The edge weights quantify the “cost” or unlikelihood of an error chain connecting two defects. The matching algorithm then selects a set of edges (a perfect matching) that minimizes the total weight, corresponding to the most likely error configuration that explains the observed syndromes.

Logical error detection (ii)

Figure 6: A graph representing syndromes before applying the MWPM

Logical error detection (iii)



Future works

- Better weight mapping
- Extend the project to create logical qubits
- Multi-qubit gates, CNOTs

Bibliography

[1] A. G. Fowler, M. Mariantoni, J. M. Martinis, and A. N. Cleland, "Surface codes: Towards practical large-scale quantum computation," *Phys. Rev. A*, vol. 86, no. 3, p. 32324, Sep. 2012, doi: 10.1103/PhysRevA.86.032324.