Quantum Error Correction: Part 2
Surface Code

J. Roffe, arXiv:1907.11157v1
Surface Code

• Based stabilizers and repeated measurements in both the Z and X bases

• Qubits classified as "data" or "measurement"

• Requirements:
  • All qubits must allow single-qubit rotations and CNOT between nearest neighbors
  • For Hadamard, must be able to SWAP state with neighbors
  • Measurement in the Z basis
Basic Four-Cycle

A₁ measures $X₁X₂$ (phase)

A₂ measures $Z₁Z₂$ (bit flip)

2 physical qubits (D), 2 stabilizers

$n = 2$, $m = n - k = 2$, which means $k = 0$

Not a useful code!
[[5,1,2]] Surface Code

Stabilizers:
\[ A_1 = X_1X_2X_3 \]
\[ A_2 = Z_1Z_3Z_4 \]
\[ A_3 = Z_2Z_3Z_5 \]
\[ A_4 = X_3X_4X_5 \]

Detecting Errors
Logical X
Perform X along vertical border

Logical Z
Perform Z along horizontal border
[[13, 1, 3]] Error Correcting Code

\[ n = \lambda^2 + (\lambda - 1)^2, \quad k = 1, \quad d = \lambda \]

one logical qubit

**Logical X**

Perform X along vertical border

**Logical Z**

Perform Z along horizontal border
Measurement Qubits

- "measure-Z" qubits (green)
  "measure-X" qubits (yellow)

- measure-Z qubit forces its neighbors (a, b, c, d) into an eigenvalue of $Z_a Z_b Z_c Z_d$

- measure-X qubit forces its neighbors (a, b, c, d) into an eigenvalue of $X_a X_b X_c X_d$
### Stabilizers

<table>
<thead>
<tr>
<th>Eigenvalue</th>
<th>( \hat{Z}_a \hat{Z}_b \hat{Z}_c \hat{Z}_d )</th>
<th>( \hat{X}_a \hat{X}_b \hat{X}_c \hat{X}_d )</th>
</tr>
</thead>
<tbody>
<tr>
<td>+1</td>
<td>(</td>
<td>gggg\rangle) (</td>
</tr>
<tr>
<td></td>
<td>(</td>
<td>gegee\rangle) (</td>
</tr>
<tr>
<td></td>
<td>(</td>
<td>eegge\rangle) (</td>
</tr>
<tr>
<td></td>
<td>(</td>
<td>geegge\rangle) (</td>
</tr>
<tr>
<td></td>
<td>(</td>
<td>eege\rangle) (</td>
</tr>
<tr>
<td>-1</td>
<td>(</td>
<td>gggee\rangle) (</td>
</tr>
<tr>
<td></td>
<td>(</td>
<td>ggege\rangle) (</td>
</tr>
<tr>
<td></td>
<td>(</td>
<td>gege\rangle) (</td>
</tr>
<tr>
<td></td>
<td>(</td>
<td>egege\rangle) (</td>
</tr>
<tr>
<td></td>
<td>(</td>
<td>eege\rangle) (</td>
</tr>
</tbody>
</table>
Surface Code Cycle

\[ +1: |\Psi_{Z+}\rangle \]
\[ -1: |\Psi_{Z-}\rangle \]

\[ +1: |\Psi_{X+}\rangle \]
\[ -1: |\Psi_{X-}\rangle \]
Error Detection

• Error causes changes in the measurement outcomes
• Does not try to correct (e.g. by applying X or Z)
  • Those operations are error-prone
• In software, tracks the errors in a qubit and adjusts subsequent measurement results (classically)
  • Later errors can "undo" the adjustment
Logical Operations

Missing measurement qubits (e.g., on boundaries) introduce additional degrees of freedom. Ex: Diagram has 41 data qubits and only 40 stabilizers.

**Can this array be viewed as a single logical qubit?**

Applying X at ALL of the blue locations will alter the state of the array, but measurement results will remain the same. It has the affect applying a logical X to the logical qubit.

Likewise for Z operator applied along red line.

Large arrays are desired for low logical error rates. (More on this later.) How can we increase the number of logical qubits within an array?
Create "holes" (defects) to generate additional degrees of freedom. Just "turn off" one or more measurement bits.

Figure shows a "single Z-cut" qubit.
Logical Operators

These cuts result in $d = 3$.

Moving further apart increases the distance for $X$ in figure (a), but $Z$ is limited by the size of the cut.

Can create larger holes and space further apart to increase $d$. (E.g., $d=8$, $d=16$)
Software “Gates”

Logical X and Z operators are implemented in **software**

Commute through gates until reach another gate of the same type (cancel each other) or a measurement (correct the outcome)
Moving Qubits Around the 2D Array

• Can move the location of the holes around in the 2D surface code array

• This is helpful for implementing a logical CNOT gate between two logical qubits, by a braiding transformation

Next few slides borrowed from tutorial by Mark Wilde, LSU
a) Wait for current surface code cycle to complete.

b) Then don’t measure $Z$-stabilizer below $Z$-cut, and turn 4-terminal $X$-stabilizers acting on qubit 6 to 3-terminal stabilizers. Measure $X$ of qubit 6

c) Turn on measure-$Z$ qubit of original $Z$-cut hole. Convert $X$ stabilizers back to 4 terminals
Multicell Move Extending One-Cell Move

- Multicell move is performed similarly to one cell move, and can be completed in a single surface code cycle.
CNOT by Braiding (Sketch)

• Recall that CNOT in Heisenberg picture is equivalent to transformation of Pauli operators as

\[ X \otimes I \rightarrow X \otimes X \]
\[ Z \otimes I \rightarrow Z \otimes I \]
\[ I \otimes X \rightarrow I \otimes X \]
\[ I \otimes Z \rightarrow Z \otimes Z \]

• So we need to effect this transformation on logical operators
CNOT by Braiding (Sketch)

Effect of loop move around the X-cut of the upper hole of the second qubit is equivalent to realizing a logical X for the second qubit:

\[ X \otimes I \rightarrow X \otimes X \]
Effect of loop move for $I$ on 1\textsuperscript{st} qubit and $X$ on 2\textsuperscript{nd} has no effect on 2\textsuperscript{nd} qubit:

$I \otimes X \rightarrow I \otimes X$
CNOT by Braiding (Sketch)

• Effect of loop move for $I$ on 1st qubit and $Z$ on 2nd affects 1st qubit:
  
  $I \otimes Z \rightarrow Z \otimes Z$
Hadamard (no details)
S and T Gates

\[ |\psi_L\rangle \rightarrow \hat{S}_L |\psi_L\rangle \]

\[ |Y_L\rangle \rightarrow H |Y_L\rangle \]

\[ |A_L\rangle \rightarrow S_L \left[ \begin{array}{c} \hat{I}_L \\ \hat{X}_L \hat{Z}_L \end{array} \right] \hat{T}_L |\psi_L\rangle \]

\[ |\psi_L\rangle \rightarrow M_Z \]
How many physical qubits?

Increasing $d$ (dimension of array) improves logical error rate.

On the order of **thousands** of physical qubits per logical qubit.