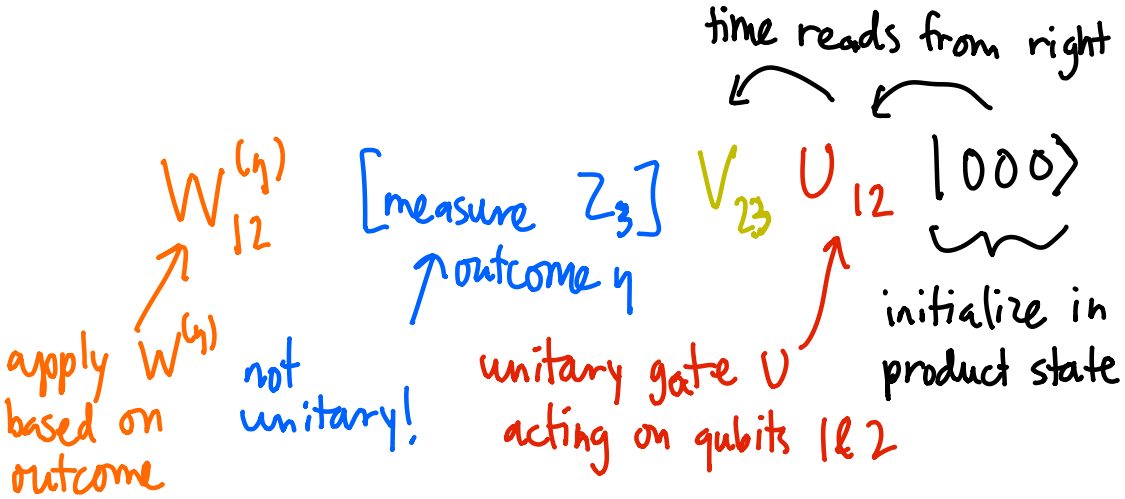
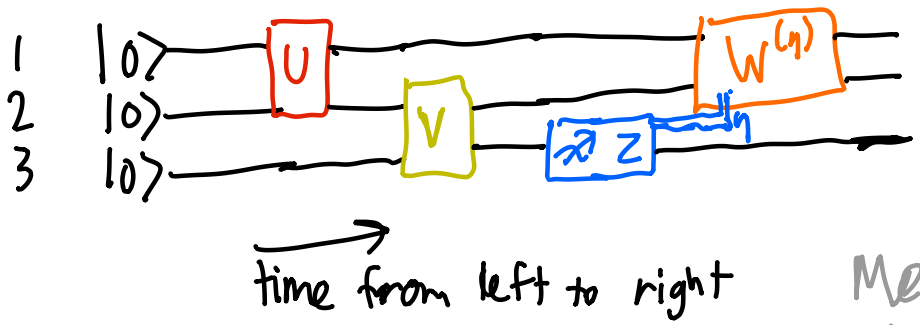
This lecture discusses how to actually measure syndromes for CSS codes in a fault-tolerant manner.

First thing to review is some notation for quantum circuits which will be more useful than writing out sequences of unitaries.

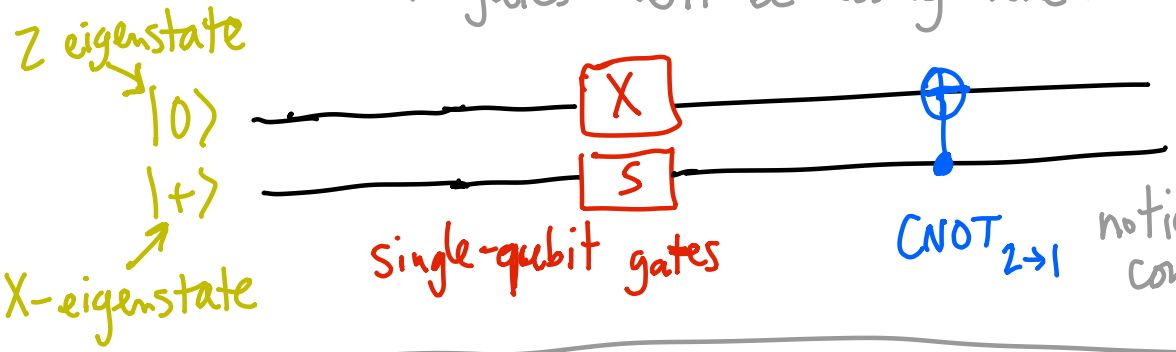




generic box ⇒ generic unitary.
note symbol for measurement

Measurement outcome denoted w/ double line

The most common gates we'll be using are:



---

let's now build the circuits needed for syndrome extraction in a CSS code.

Key issues: ① DO NOT measure $Z_1Z_2Z_3Z_4$ by measuring Z_1, then Z_2, ...

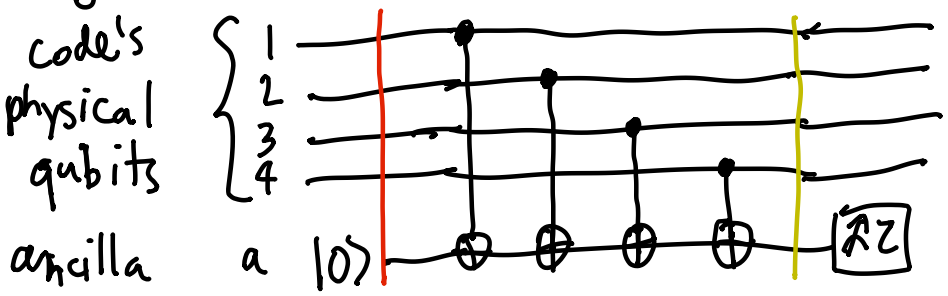
② most experiments can only do 2-qubit gates

Resolve: add ancilla qubits to help w/ syndrome extraction

data qubits: physical qubits of QEC code

Recall: $CNOT_{1\to 2}\begin{pmatrix} X_1 \\ X_2 \\ Z_1 \\ Z_2 \end{pmatrix} CNOT_{1\to 2} = \begin{pmatrix} X_1X_2 \\ X_2 \\ Z_1 \\ Z_1Z_2 \end{pmatrix}$ ($=CNOT^{\dagger}_{1\to 2}$)

Alg (Z-check syndrome extraction): to measure $Z_1Z_2Z_3Z_4$:



Important that we start ancilla in a known state. To see why it works, notice that at initial time, Stabilizer $Z_a = +1$

$CNOT_{1\to a} Z_a CNOT_{1\to a} = Z_1Z_a \Rightarrow$ final stabilizer $Z_1Z_2Z_3Z_4Z_a = +1$

or: $Z_a = Z_1Z_2Z_3Z_4$

Measure $Z_a \Rightarrow$ outcome tells us $Z_1Z_2Z_3Z_4$

Example: Start in $|\psi_0\rangle = \frac{1}{2}\left[\underbrace{|0000\rangle}_{+} + \underbrace{|1111\rangle}_{+} + \underbrace{|0001\rangle}_{-} + \underbrace{|1110\rangle}_{-}\right] \otimes |0\rangle$

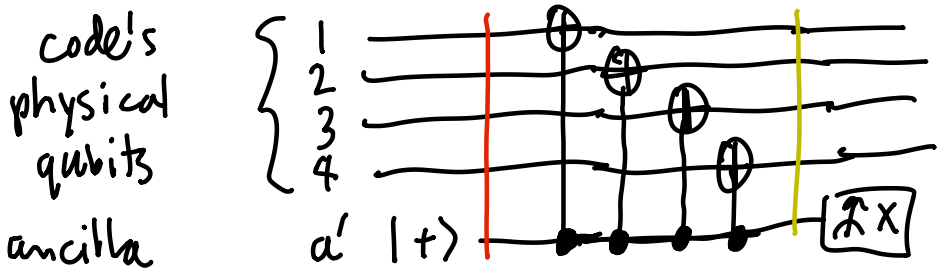
What's final outcome?

$$|\psi_{final}\rangle = \begin{cases} \frac{1}{\sqrt{2}}\left[|0000\rangle + |1111\rangle\right] \otimes |0\rangle & \text{w/ prob. } 1/2 \\ \frac{1}{\sqrt{2}}\left[|0001\rangle + |1110\rangle\right] \otimes |1\rangle & \text{w/ prob } 1/2 \end{cases}$$

$\Rightarrow$ error digitization as we should expect from theory of QEC

Straightforward to generalize this algorithm to measure a Z-check of any size.

Alg (X-check syndrome extraction): to measure $X_1X_2X_3X_4$,



Initial time: stabilizer $X_{a'} = +1$

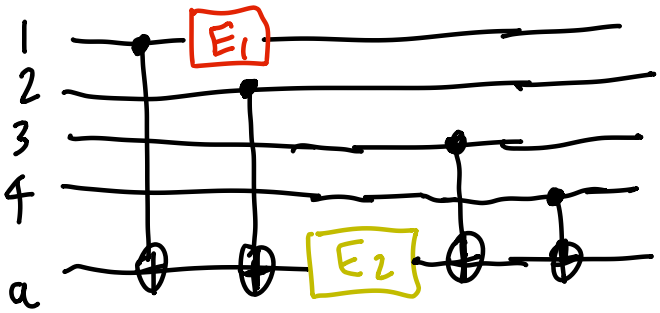
$CNOT_{a'\to 1} X_{a'} CNOT_{a'\to 1} = X_1 X_{a'}$ ⇒ final stabilizer $X_1X_2X_3X_4X_{a'}$

Measure $X_{a'} = X_1X_2X_3X_4$ to collapse code into syndrome eigenspace!

Key thing is that these circuits only used 2-qubit gates... if expt has higher-order gates then those may also be useful.

Now need to discuss fault-tolerance of this protocol...

Focus on Z-syndrome extraction since analogous story for X...



Note: errors depicted here like unitaries but same story applies for more general channels...

error E_1 on physical qubit <u>after</u> CNOT:

$E_1 = Z$ ⇒ "no effect" here (it'll be picked up during X-syndrome extraction...)

$E_1 = X$ ⇒ undetected error!

Nice way to see this is by pushing through E_1 to one end of the syndrome extraction circuit:

$$E_1 CNOT_{1\to a} = CNOT_{1\to a} \cdot (CNOT_{1\to a} E_1 CNOT_{1\to a})$$

↪ $Z_1 \to Z_1$, $X_1 \to X_1X_a$

$\Rightarrow$ X_1 error becomes $X_1 X_a \Rightarrow$ syndrome outcome opposite:

$$X_1 X_2 X_3 X_4 X_a = -1$$

So this error can enhance probability of a faulty syndrome outcome.

error E_2 on ancilla during extraction:

$E_2 = X_a \Rightarrow$ syndrome outcome wrong

$E_2 = Z_a \Rightarrow$ hook error

harmless stabilizer

$$Z_a \underbrace{CNOT_{12\to a}} = CNOT_{12\to a}\left(CNOT_{12\to a}\, Z_a\, CNOT_{12\to a}\right)$$

$$= CNOT_{12\to a}\, Z_1 Z_2\, Z_a$$

hook error

$$= CNOT_{2\to a}\, CNOT_{1\to a}$$

Note: hook error $Z_1 Z_2$ stabilizer-equiv. to $Z_3 Z_4$. check weight

worry: correlated Z-errors (hook) of weight $\leq \lfloor \frac{1}{2} \Delta_C \rfloor$

arise during noisy syndrome extraction

$\Rightarrow$ "engineering" problem: circuit-level fault-tolerance

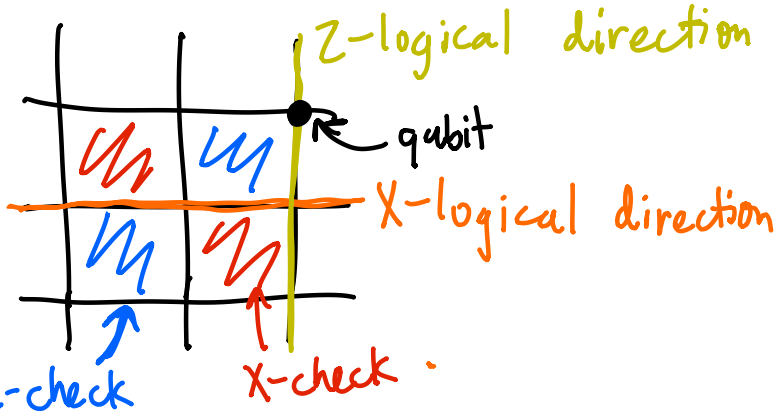
$\Rightarrow$ threshold for correlated errors generated thru these circuits

One generally resorts to large-scale numerical studies using G-K algorithm to efficiently simulate circuit-level noise.

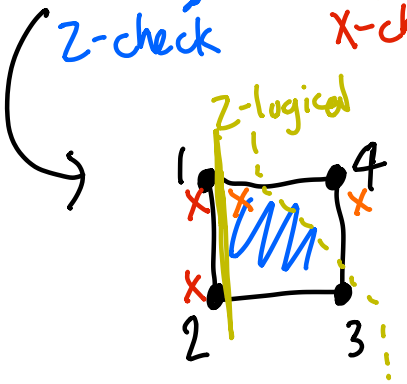
circuit-level noise $\Rightarrow$ threshold at $p \sim 0.01$ (not 0.03) for standard (MWPM) decoder

Indeed let's now see how circuit-level noise complicates syndrome extraction for (rotated) surface code...

Issue #1: hook errors can reduce the effective distance.



Recall the setup of the rotated surface code



Measuring Z-check ⇒ Z-hook error possible btwn first 2 measured qubits (equiv. to last 2)

syndrome extraction $1 \to 2 \to 3 \to 4$ causes hook error aligned w/ logical

$\Rightarrow d_{eff} \approx d/2$

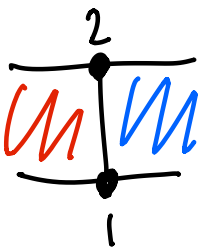
instead choose $1 \to 4 \to 3 \to 2$ so Z-hook error doesn't align w/ any lowest-weight Z-logical

Similar strategy arises for X-syndrome extraction...

Issue #2: CNOT scheduling.

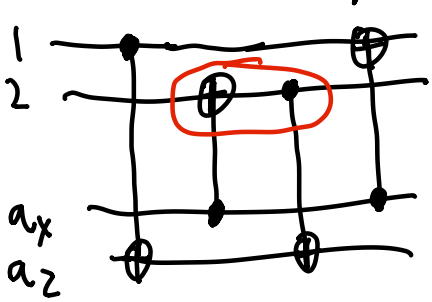
Ideally, weight-4 checks measured in 4 rounds.

⇒ (most) qubits in a CNOT in each layer of circuit.

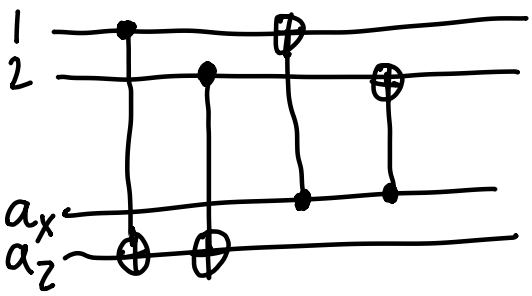


Consider a patch of surface code w/ checks $X_1 X_2 \cdots$ $Z_1 Z_2 \cdots$ checks to measure.

bad circuit layout:



good circuit layout:



We can see the issue by evolving final measurement outcome backward through circuit...

$U_{bad}^\dagger Z_{a_2} U_{bad} = Z_1 Z_2 Z_{a_x} Z_{a_2}$ unwanted!! not a stabilizer of initial state

$U_{good}^\dagger Z_{a_2} U_{good} = Z_1 Z_2 Z_{a_2}$ $=+1$ at start, meaning we measure $Z_1 Z_2$ at the end

in principle, fault-tolerance for generic qLDPC.

Why? - finite rounds of syndrome extraction for all checks
$\Rightarrow$ limits idling errors
- finite-size hook errors w/ 2-qubit gates...

Since one error might duplicate in hook, estimate

$$\mathbb{P}[\text{logical fail}] \sim n\, p_c^{\,d/(\Delta_c/2)}$$ in worst case

So LDPC still important to ensure

$$d \gg \log n \Rightarrow \mathbb{P}[\text{logical fail}] \to 0$$