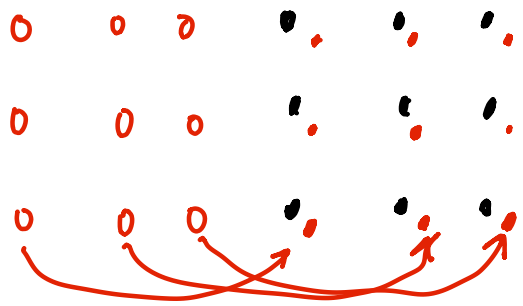
So far we saw how to make CSS codes w/ transversal logical gates, but...

Problem? transversal CNOT btwn 2 copies of 2d code...



need to "align" 2d architectures?

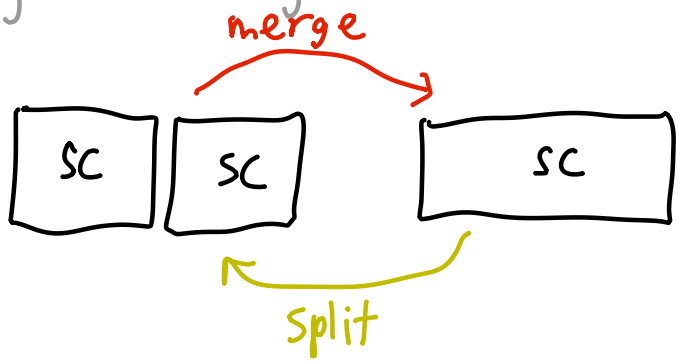
maybe OK for some architectures but hard for others (e.g. SC circuits)

This motivates alternate method for logical CNOT that preserves planarity in a 2d architecture...

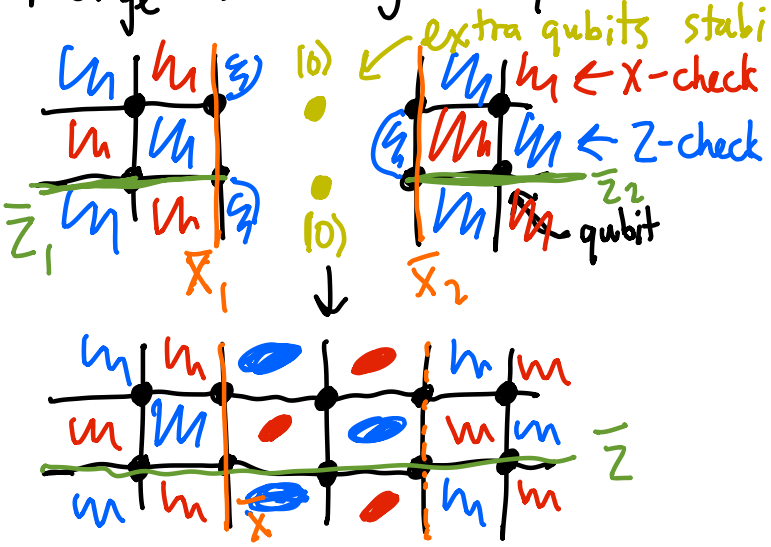
Def: lattice surgery = non-transversal method of applying CNOT in 2d planar codes

Focus on how to apply CNOT b/c we saw this entangling gate can also be used for magic state injection/distillation...

Essential steps:



Merge 2 rough bdy: (smooth merge very analogous...)



The merge takes a pair of logicals into one, as is visible...

- New Z-checks automatically work
- Measure new X-checks ⇒ Measure $\bar{X}_1\bar{X}_2$

What have we done at the level of wave functions?
Assume for now there were no errors to start...

bar denotes logical

$$|\psi_{before}\rangle = [\alpha_1|\bar{0}\rangle + \beta_1|\bar{1}\rangle] \otimes |0\rangle_{merge} \otimes [\alpha_2|\bar{0}_2\rangle + \beta_2|\bar{1}_2\rangle]$$

If $\eta = \prod$ [measurement outcome] then:

new X-check

$$|\psi_{after}\rangle \propto \begin{cases} (\alpha_1\alpha_2+\beta_1\beta_2)\dfrac{|00\rangle+|11\rangle}{\sqrt{2}} + (\alpha_1\beta_2+\alpha_2\beta_1)\dfrac{|01\rangle+|10\rangle}{\sqrt{2}} & \eta=+1 \\ (\alpha_1\alpha_2-\beta_1\beta_2)\dfrac{|00\rangle-|11\rangle}{\sqrt{2}} + (\alpha_1\beta_2-\alpha_2\beta_1)\dfrac{|01\rangle-|10\rangle}{\sqrt{2}} & \eta=-1 \end{cases}$$

up to normalization from Born's rule

new code has $|\bar{0}'\rangle = \dfrac{|00\rangle+|11\rangle}{\sqrt{2}}$ $\quad |\bar{1}'\rangle = \dfrac{|01\rangle+|10\rangle}{\sqrt{2}}$

since $\bar{X}_1\bar{X}_2$ is a new stabilizer!

alternatively: $E|0_L\rangle = \dfrac{|00\rangle-|11\rangle}{\sqrt{2}}$ $\quad E|1_L\rangle = \dfrac{|01\rangle-|10\rangle}{\sqrt{2}}$

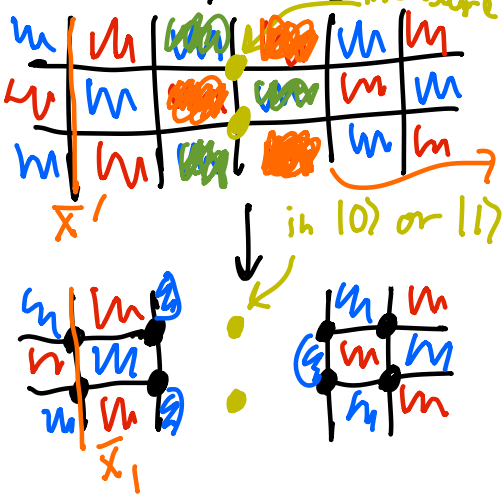
↳ track $E \sim \bar{Z}_1$ by Pauli frame tracking

fault-tolerance ⇒ d rounds of syndrome extraction / correction (tracking)

So lattice merge removed logical ⇒ collapse $|\psi\rangle$

⇒ FT measurement of $\bar{X}_1\bar{X}_2$

Lattice splitting reverses procedure:



destroy ● X-stabilizers by Z-measurement

preserve ● Z-stab

$+1 = Z_aZ_b\,Z_cZ_d \Rightarrow Z_aZ_b = \eta_c\eta_d$

not measured · measured

(fix error / frame track)

Start in logical state $|\psi_{before}\rangle = \alpha|0_L\rangle + \beta|1_L\rangle$

$\bar{X}_2 = \bar{X}_1 = \bar{X}'$ unaffected by measurements

in codespace

$\Rightarrow$ final state stabilized by $\bar{X}_1\bar{X}_2$ up to frame tracking

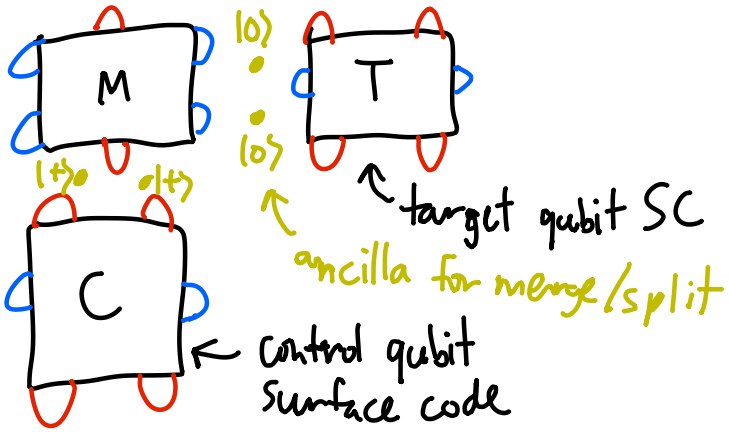
$\Rightarrow |\psi_{after}\rangle = \frac{\alpha+\beta}{\sqrt{2}}|+_1 +_2\rangle + \frac{\alpha-\beta}{\sqrt{2}}|-_1 -_2\rangle.$

Information not destroyed during split, but new entangled logical qubits have been made.

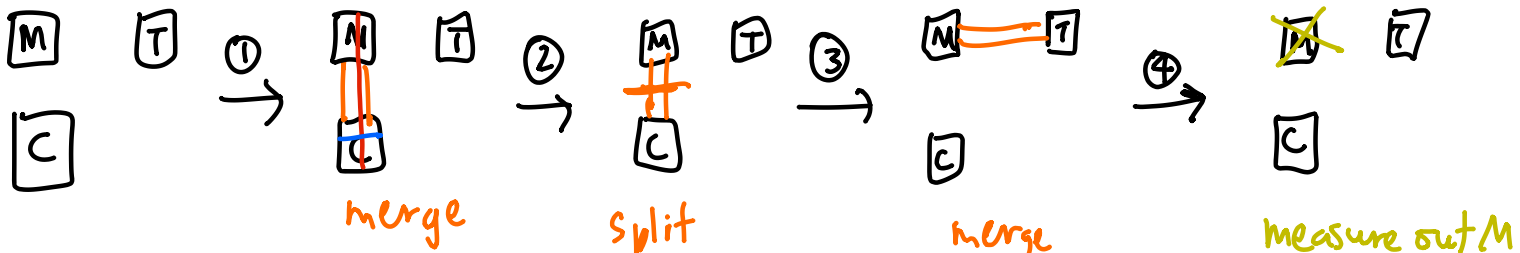
Let's see how these 2 operations can be used to perform FT CNOT

Prepare 3 patches of SC to implement FT $\overline{CNOT}_{C\to T}$

Prepare M surface code in logical $|+\rangle$ state [stabilized by $\bar{X}$]



Implement CNOT by:



See why it works by tracking stabilizers & logicals...

①: stabilizer $\bar{X}_M$ becomes $\bar{Z}_C\bar{Z}_M \cdot \eta_1$ ← measurement outcome during merge.

C-logicals: $\bar{X}_C \to \bar{X}_{CM}$ (illustrated) $\bar{Z}_C \to \bar{Z}_C$

② stabilizer still $\bar{Z}_C \bar{Z}_M \eta_1$; C-logicals $\bar{X}_C \cdot \bar{X}_M$ & $\bar{Z}_C$

T-logicals $\bar{X}_T$ & $\bar{Z}_T = \bar{Z}_T \bar{Z}_M \bar{Z}_C \eta_1$

③ new stabilizer: $\bar{X}_M \bar{X}_T \eta_3$

X-logicals: $(\bar{X}_C \bar{X}_M, \bar{X}_T) = (\bar{X}_C \bar{X}_T \eta_3, \bar{X}_T)$

Z-logicals: $(\bar{Z}_C, \bar{Z}_T \bar{Z}_M \bar{Z}_C \eta_1)$ ← chose this representative to commute w/ new stabilizer!

④ transversal Z measure in region M: $\bar{Z}_M \to \eta_4 \in \{\pm 1\}$

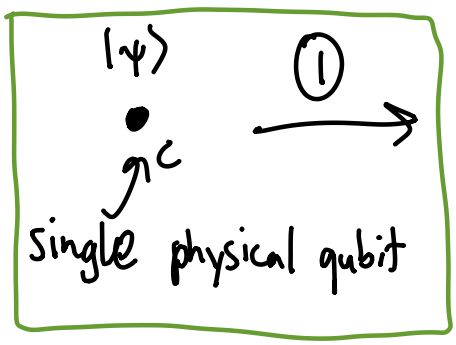
final logical data has implemented CNOT (up to frame tracking)

$(\bar{X}_C, \bar{X}_T) \to (\bar{X}_C \bar{X}_T \eta_3, \bar{X}_T)$ $\quad (\bar{Z}_C, \bar{Z}_T) \to (\bar{Z}_C, \bar{Z}_T \bar{Z}_C \cdot \eta_1 \eta_4)$

If we can combine lattice surgery CNOT w/ single qubit logical gates ⇒ FT computation (universal!)

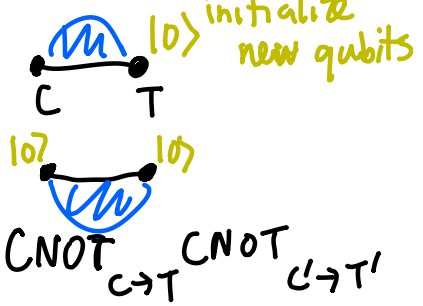
(Magic) state injection ⇒ single-qubit gate by distillation + state injection?

Iterate analogous procedure to above to grow logical surface code...



NOT fault-tolerant

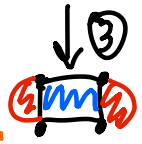
↳ distillation needed



②

measure X-check, correct/frame track

③



$|0_L\rangle$

rough merge

$|\psi_L\rangle$

etc. ←

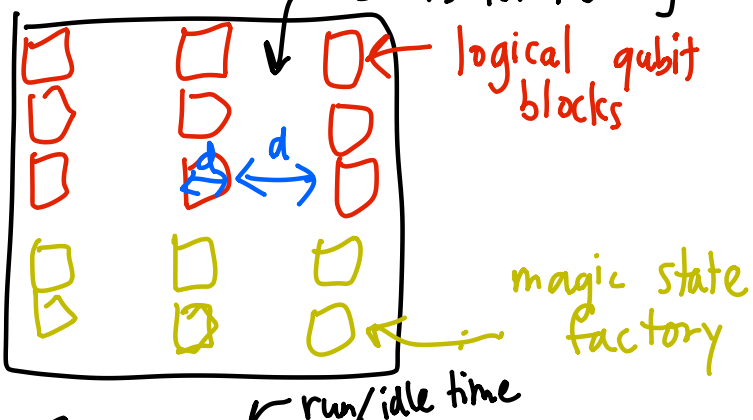
If error rate sufficiently small, error limited by state injection
$\Rightarrow$ FT via distillation.

There are also some tricks to implement H in planar patches as well but we won't go through them here.

That's all! We now have ingredients to build a useful quantum computer. Let's talk just a bit about how to estimate needs for FT computation... These are certainly not optimized but give you a schematic/asymptotic sense

Goal: computation w/ k logical qubits
logical circuit has M T gates
physical error rate p
desired logical gate/distillation error rate ε

How big of a surface code needed?



estimate:

$$\varepsilon \sim (100p)^{\frac{d+1}{2}}$$

$$\Rightarrow d \sim 2\frac{\log(1/\varepsilon)}{\log(100p)} \sim 2\log\frac{1}{\varepsilon}$$

$\varepsilon[M + nt] \ll 1$ makes computation safe...

Take $\varepsilon \sim 10^{-15}$, $M \sim 10^8$, $p \sim 10^{-3}$ $\Rightarrow$ $d \sim 29$

If $k \sim 10^3$, estimate $n \sim k \cdot d^2 \cdot (1.01) \sim 2 \times 10^6$

(d^2: patch size; 1.01: routing + factory space-time ancilla)

need 2 layers of 15-to-1 distillation: $35(35p^3)^3 \ll \varepsilon$ ← neglects circuit level noise...

$\Rightarrow$ lowest level factory runs 225M $\sim 2\times10^{10}$ times

↳ but first level can be done w/ smaller SC patch...

so only need 15M $\sim 10^9$ larger-block distillations

If $O(1)$ γk T's can be distilled in parallel:

factory runs $10^9/\gamma k \sim 10^6$ times [× depth of distillation circuit]

depth of distillation circuit: $d \times O(10)$

FT QEC → d; $O(10)$ ← details depend on p.

$\Rightarrow$ runtime of $\sim 3\times10^8$ cycles

if cycle takes $\delta t \sim 10\ \mu s$ (fast!) $\Rightarrow$ $\sim$ 1 hr compute w/ $\sim 2\times10^6$ phys. qubit

Pretty sad, but the ability to break RSA is sufficient to make this field very well funded. And more modern LDPC codes may drastically reduce these costs... particularly in physical qubit count. So the estimate above intended to give order of magnitude estimate, rather than precise counts for RSA (which rapidly improve anyway...)