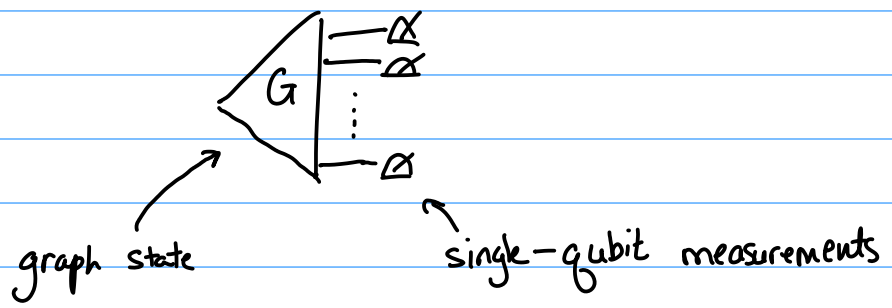
Measurement-based quantum computing (MBQC)

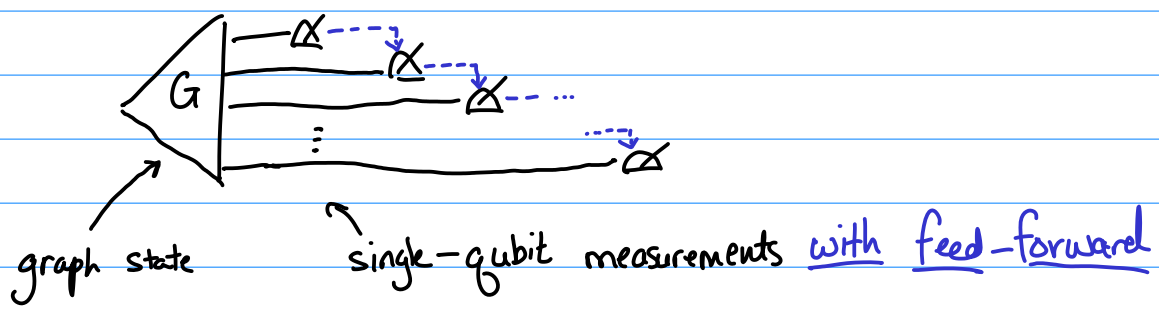
:= QC where measurements make up most of the computation.

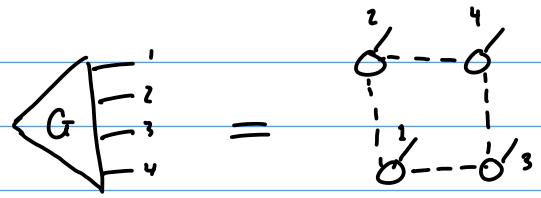
The "code" of MBQC is a measurement pattern:
:= measurement choices + classical control (feed-forward)

Several models:
- (gate teleportation)
- one-way model *
- hypergraph MBQC
- fault tolerant QC
  - lattice surgery (*)
  - topological FTQC
- ...

One-way model of MBQC (Raussendorf/Briegel 2001)

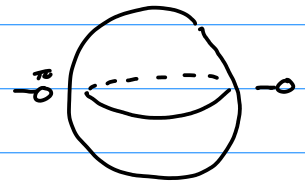




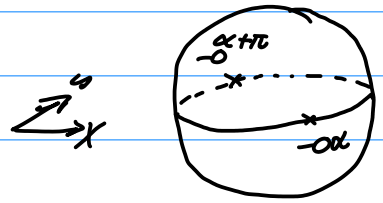


SINGLE-QUBIT MEASUREMENTS:

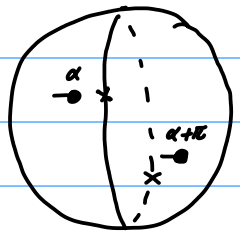
X-measurement: $\left\{ \overset{k\pi}{\circ} \right\}_{k=0,1}$



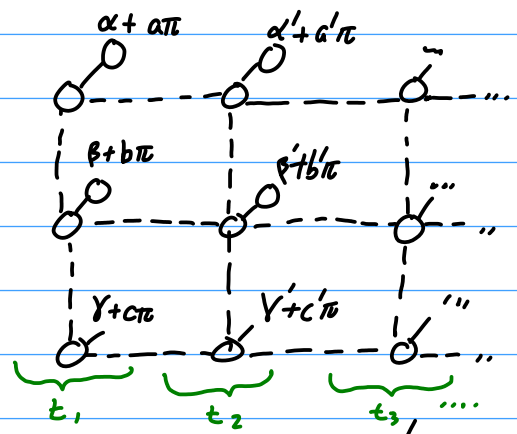
More generally: <u>XY-plane</u> measurements: $\left\{ \overset{\alpha+k\pi}{\circ} \right\}_{k=0,1}$



Similarly, <u>YZ-plane</u> measurements: $\left\{ \overset{\alpha+k\pi}{\bullet} \right\}_{k=0,1}$



(Z-measurements $\Rightarrow \alpha = 0$)

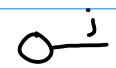


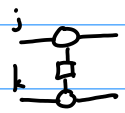
Feed-forward : $\alpha' = \alpha'(a,b,c)$ ← fn of (earlier) measurement outcomes. (a.k.a. signals)

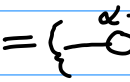
$\beta' = \beta'(a,b,c)$

.....

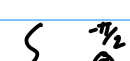
<u>Def</u> A <u>measurement pattern</u> for the one-way model consists of a sequence of instructions:

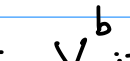
* $N_j :=$  prepare a <u>new</u> qubit in $|+\rangle$

* $E_{jk} :=$  <u>e</u>ntangle qubits j & k

* $M_j^{\alpha} := \{$  $\}_{s_j \in 0,1}$ <u>m</u>easure qubit j in XY plane
  * store result in <u>signal</u> $s_j \in \{0,1\}$
  * $\alpha = \alpha(s_{k_1}, s_{k_2}, \dots)$

* $M_j^{YZ,\alpha} := \{$  $\}_{s_j \in 0,1}$ " " " " YZ plane "

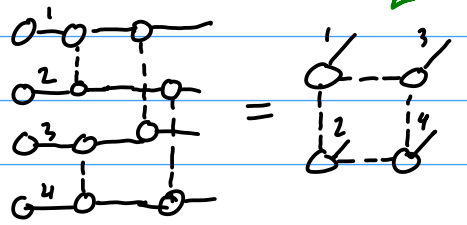
* $M_j^{XZ,\alpha} := \{$  $\}_{s_j \in 0,1}$ " " " - XZ plane "

* $Z_j^b :=$  , $X_j^b :=$ perform Pauli corrections, where
  * $b = b(s_{k_1}, s_{k_2}, \dots)$

nodes edges

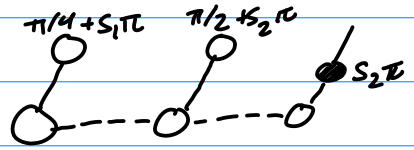
$$P := N_1 ; N_2 ; N_3 ; N_4 ; E_{12} ; E_{34} ; E_{13} ; E_{24}$$

graph state



feed forward

$$Q := N_1 ; N_2 ; N_3 ; E_{12} ; E_{23} ; M_1^{\pi/4} ; M_2^{\pi/2} ; X_3^{s_2}$$



<u>Def</u> A measurement pattern is:

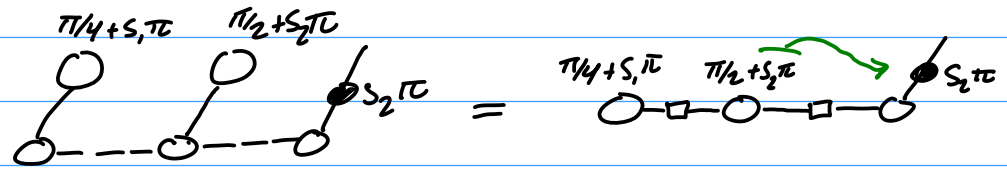
* <u>runnable</u> if all angles / corrections are fns of past measurement outcomes.

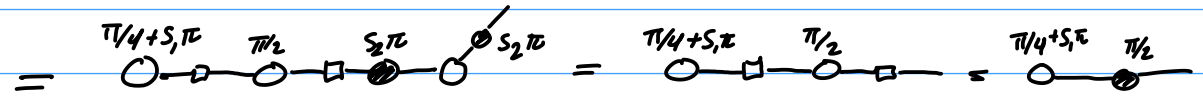
$$M_j^{\alpha} ; \ldots\ldots ; Z_k^{s_j} \quad \text{OK}$$

$$Z_k^{s_j} ; \ldots\ldots ; M_j^{\alpha} \quad \text{BAD}$$

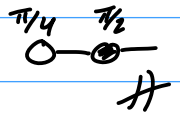
* <u>deterministic</u> if all choices of measurement outcomes give the same map (up to scalars)

Q: runnable? ✓ deterministic ✗

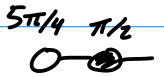




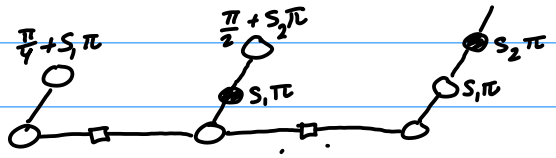
$s_1 = 0 \implies *$ =

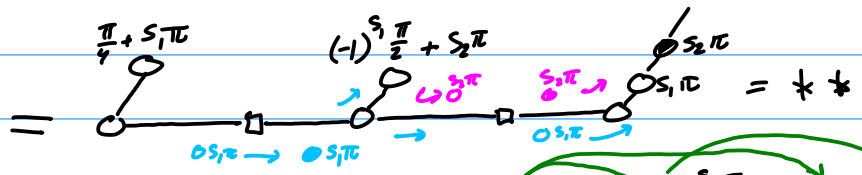


$s_1 = 1 \implies *$ =



$$Q' := N_1; N_2; N_3; E_{12}; E_{23}; M_1^{\frac{\pi}{4}}; X_2^{s_1}; Z_3^{s_1}; X_3^{s_2}$$





$$Q' = Q'' := N_1; N_2; N_3; E_{12}; E_{23}; M_1^{\frac{\pi}{4}}; M_2^{(-1)^{s_1}\frac{\pi}{2}}; Z_3^{s_1}; X_3^{s_2}$$

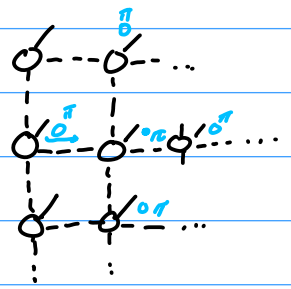
Q'' : runnable? ✓ deterministic? ✓

$s_1, s_2 \in 0, 1 \implies **$ =

Question Can I alway "push" errors forward in time?

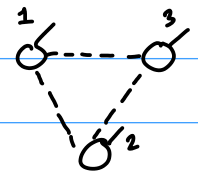
Answer: It depends on the graph.

✓



$t_1 < t_2 < t_3 \dots$

✗



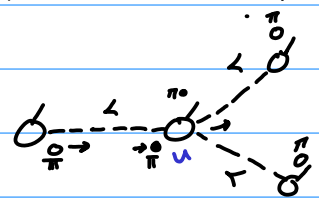
there is no time ordering for qubits $\{1,2,3\}$ that works.

CLUSTER STATE
(:= graph state shaped like a square lattice)

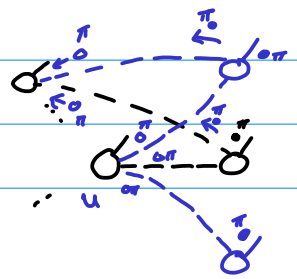
Q: how can we classify which graph states "work"?

IDEA: 1. Fix a time-ordering $\prec$: $\begin{cases} \text{past}(u) := \{v \mid v \prec u\} \\ \text{future}(u) := \{v \mid u \prec v\} \end{cases}$

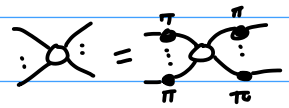
2. push errors from u into future(u) (without messing up past(u))

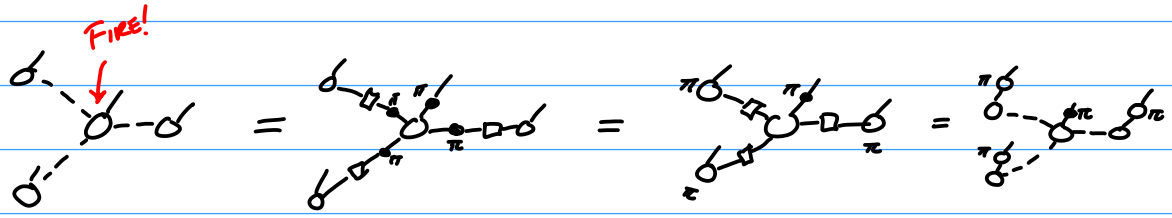


✓



✓

Equivalently, think about "firing" a spider with 

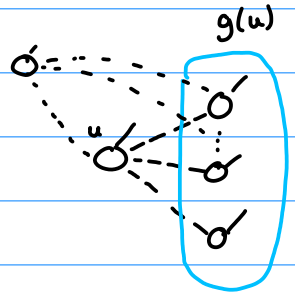


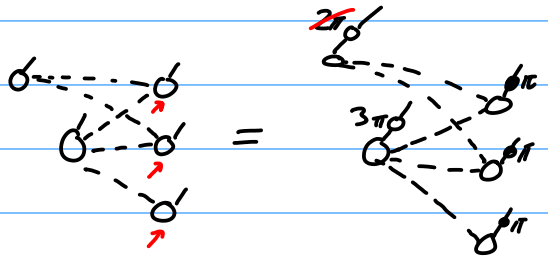
The game: for each u, find a set $g(u)$ that is:

(i) in the future of u

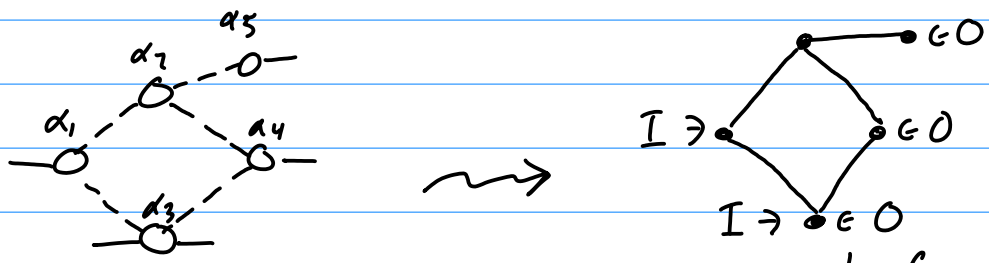
(ii) connected to u an <u>odd</u> number of times

(iii) connected to the past of u an <u>even</u> number of times





Def An open graph is a graph G with a set of inputs $I_G \subseteq V_G$ and outputs $O_G \subseteq V_G$.



graph-like ZX-diag

open graph G

Def An open graph has generalised flow (gflow) if there exists a partial order $\preceq$ on V_G and a function $g: V_G \setminus O_G \to \mathcal{P}(V_G \setminus I_G)$ such that $\forall u$:

(i) $g(u) \subseteq \text{future}(u)$

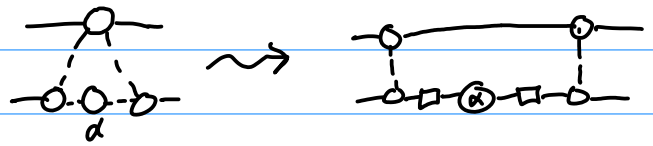
(ii) $g(u)$ connects to u an odd # of times

(iii) $\forall v \in V_G \setminus O_G$. if $v \neq u$, $v \notin \text{future}(u)$ then $g(u)$ connects to v an even # of times.

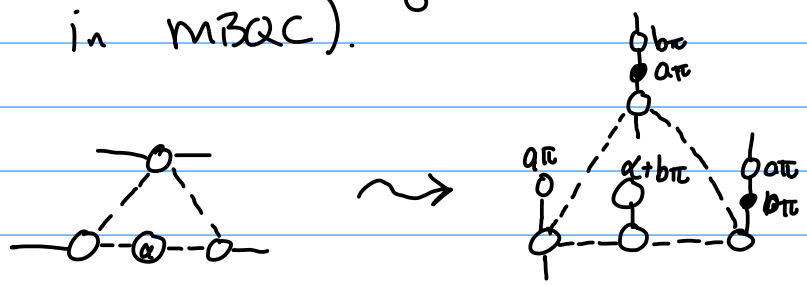
THM (Determinism) For any graph-like ZX-diagram D with gflow, there exists a runnable, deterministic pattern P that implements it.

$\Rightarrow$ There are at least 2 ways that a ZX-diagram can be "run" on a quantum computer:

1. If it can be transformed into a circuit.



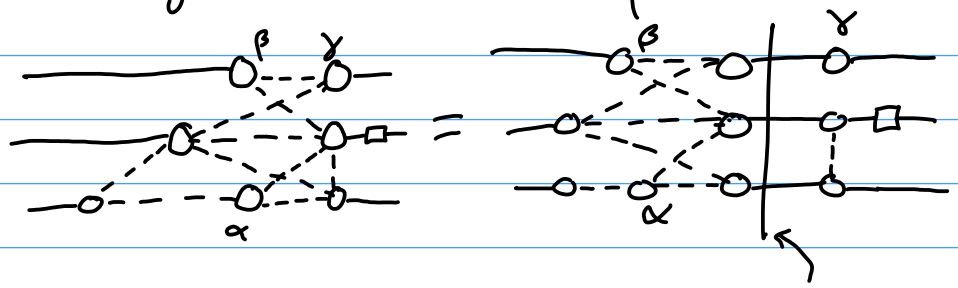
2. If it has gflow (hence can be implemented in MBQC).



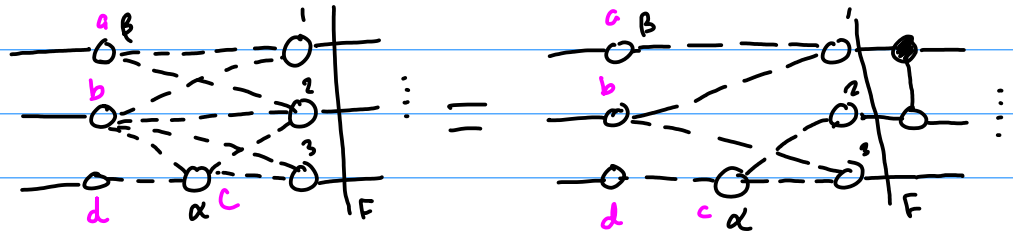
Now: 2. $\Rightarrow$ 1. (circuit extraction)

ALGORITHM (CIRCUIT EXTRACTION)

1. unfuse gates as much as possible:



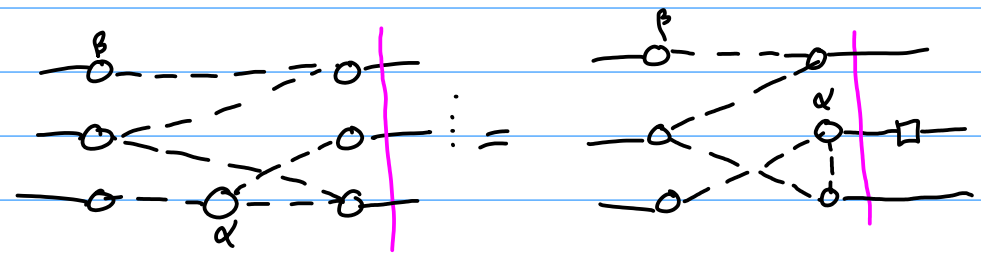
2. use CNOTs to do row operations until we get an "extractible" spider (= unit-vector row)

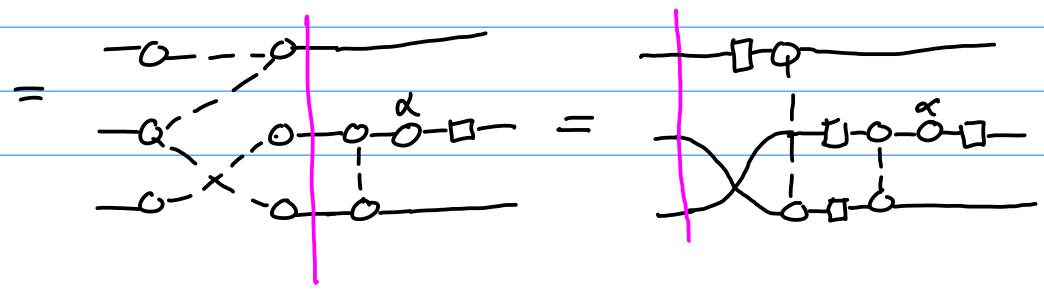


$$\begin{array}{c|cccc} & a & b & c & d \\ \hline 1 & 1 & 1 & 0 & 0 \\ 2 & 1 & 1 & 1 & 0 \\ 3 & 0 & 1 & 1 & 0 \end{array} \xrightarrow{R_2 = R_2 + R_1} \begin{array}{cccc} a & b & c & d \\ \hline 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \end{array}$$

extractible

3. Repeat 1 & 2 until nothing is left of the frontier.

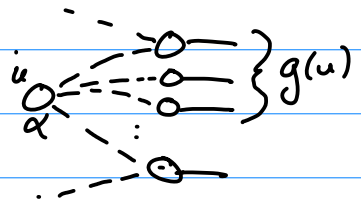




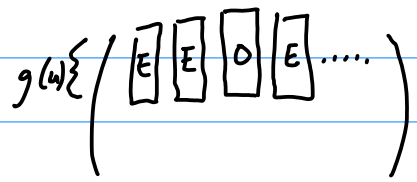
Thm If a ZX-diagram has gflow, CIRCUIT EXTRACTION terminates with a quantum circuit.

Pf Step 1 never adds spiders to the left of the frontier, so s.t.s. Step 2 always removes a spider.

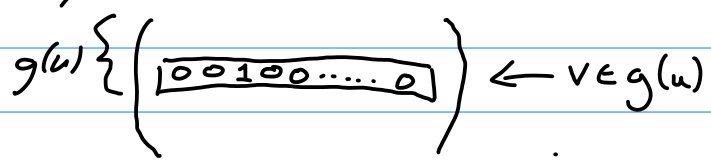
Take a maximal non-output u, w.r.t $\prec$.
Then $g(u) \subseteq \text{future}(u)$ must be all outputs:



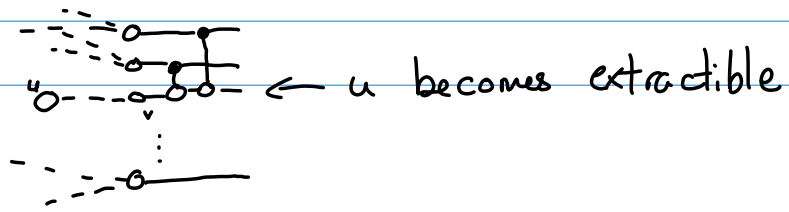
By gflow, the only node connected an odd # of times to $g(u)$ is u.



If we add all the rows to a single row, then we get:

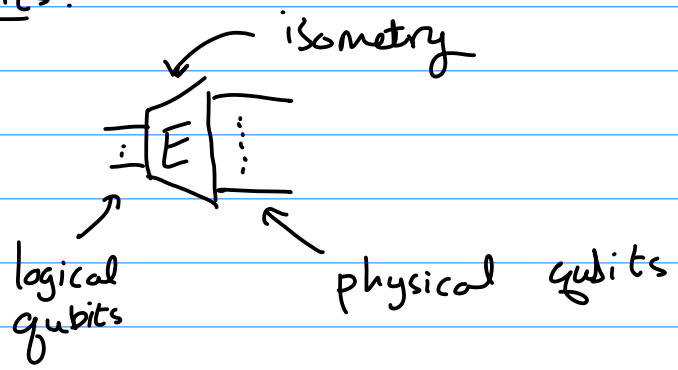


So, doing CNOTs ctrl'ed on a single $v \in g(u)$ to all other $v' \in g(u)$ gives:



Extract v make u an output. The result still has gflow and there is one fewer spider left of the frontier. ∎

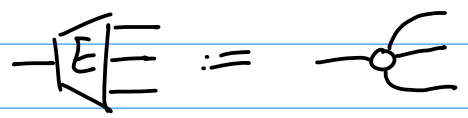
Quantum error correction works by encoding some logical qubits into a space of (more) physical qubits.



Q: Why?

A: Because some errors can be detected and/or corrected using quantum measurements without destroying the logical state.

Ex. The GHZ code:

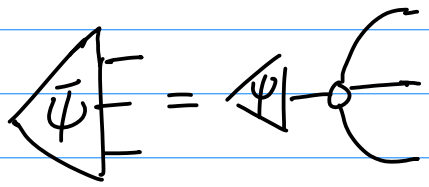


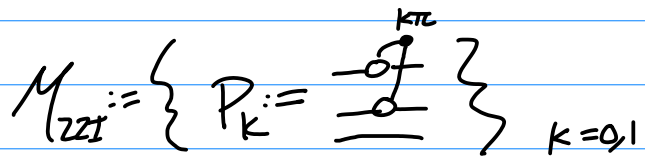
$$\underset{2D}{\mathbb{C}^2 = \mathrm{span}\{|0\rangle, |1\rangle\}} \xmapsto{E} \underset{2D}{\mathrm{span}\{|000\rangle, |111\rangle\}} \leq \underset{8D}{(\mathbb{C}^2)^{\otimes 3}}$$

$$|\bar{0}\rangle := |000\rangle, \quad |\bar{1}\rangle := |111\rangle$$

MORE GENERALLY: $|\bar{\psi}\rangle := E|\psi\rangle$.

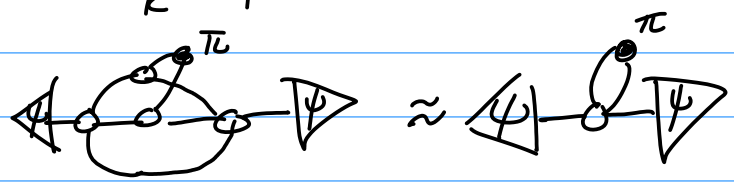
Suppose I measure ZZI:



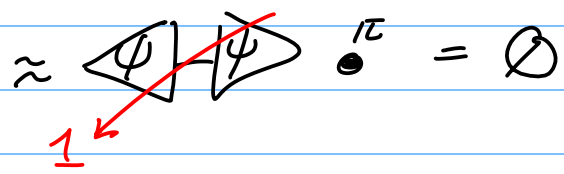


$$\mathrm{Prob}(1 \,|\, |\bar{\psi}\rangle) = \langle\bar{\psi}| P_k |\bar{\psi}\rangle$$

$=$

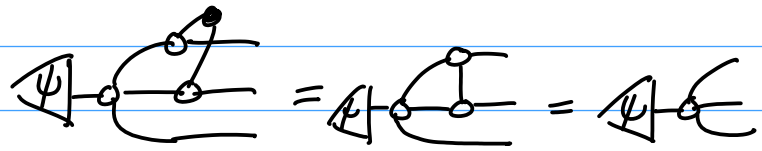


$\approx$



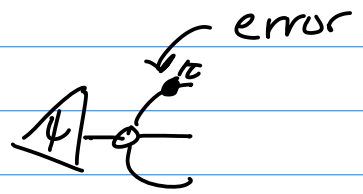
$$\Rightarrow \mathrm{Prob}(0 \,|\, |\bar{\psi}\rangle) = 1.$$

Also:

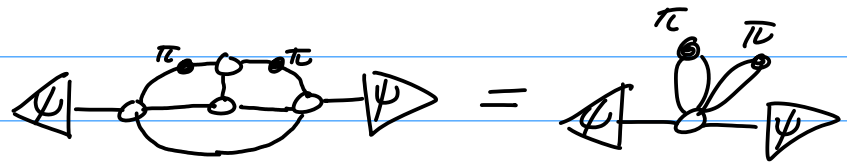
$P_0|\bar{\psi}\rangle =$  $= |\bar{\psi}\rangle$

$\Rightarrow$ measuring ZZI does not disturb $|\bar{\psi}\rangle$.

<u>But</u>



$\mathrm{Prob}(0|(X\otimes I\otimes I)|\bar{\psi}\rangle) =$



$= \emptyset$.

$\Rightarrow \mathrm{Prob}(1|(X\otimes I\otimes I)|\bar{\psi}\rangle) = 1$.

So a ZZI measurement can <u>detect</u> the error $X\otimes I\otimes I$.

<u>THM</u> The GHZ code can detect (and correct) any error in the set $\{XII, IXI, IIX\}$.

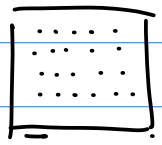
bit-flip errors

Better codes correct more errors (e.g. "phase flips" like ZII..., multi-qubit errors, etc.)

Q: How can I <u>compute</u> with encoded states.

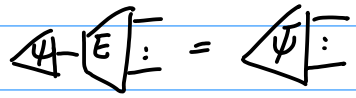
A: FTQC!

<u>SCHEME</u>: LATTICE SURGERY.

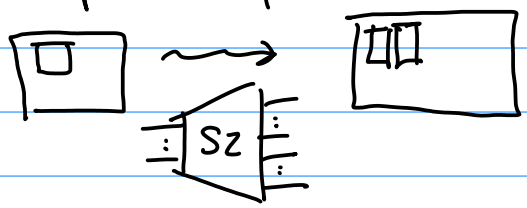
<u>IDEA</u>: • Use a grid of qubits 

• encode 1 qubit as a "patch"

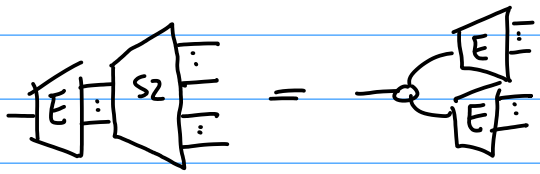
• implement operations to:

— prepare 1Q states: 

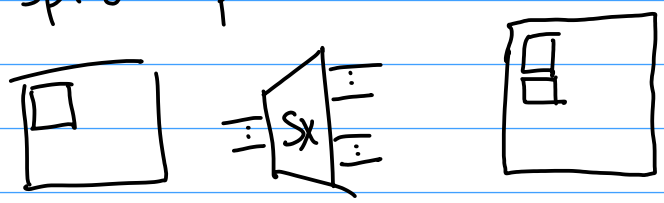
— "Z-split" patches:

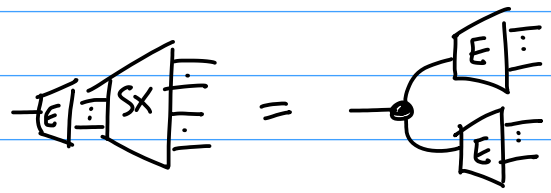


where:

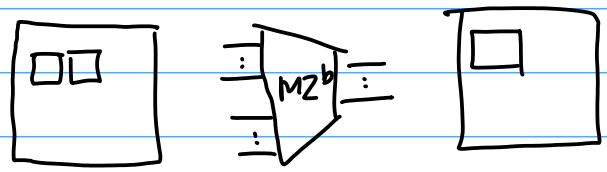


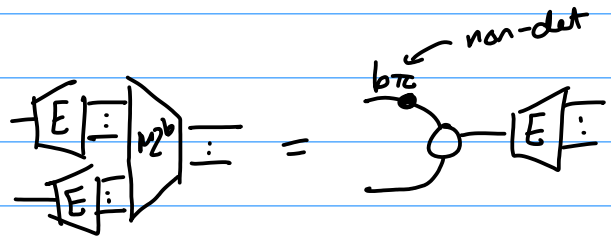
— "X-split" patches



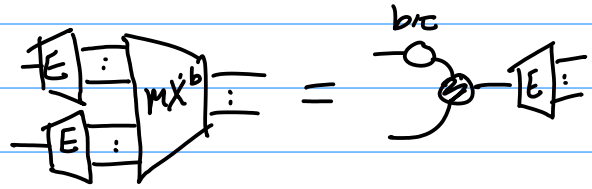


— "Z-merge"

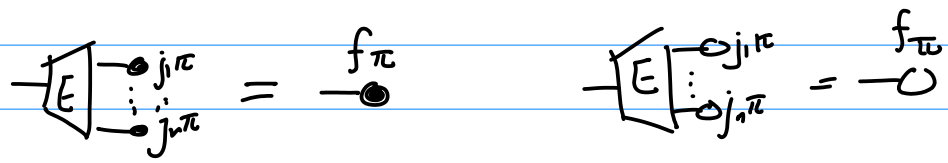




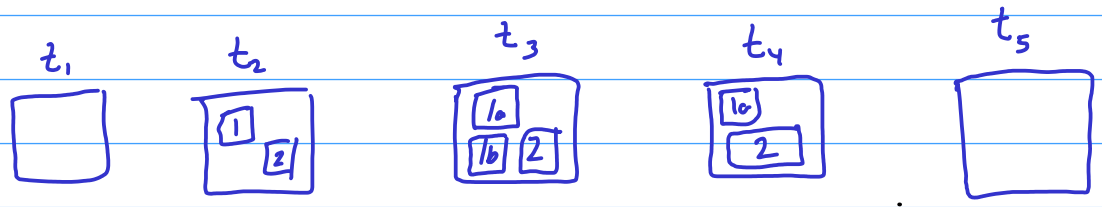
— "X-merge"

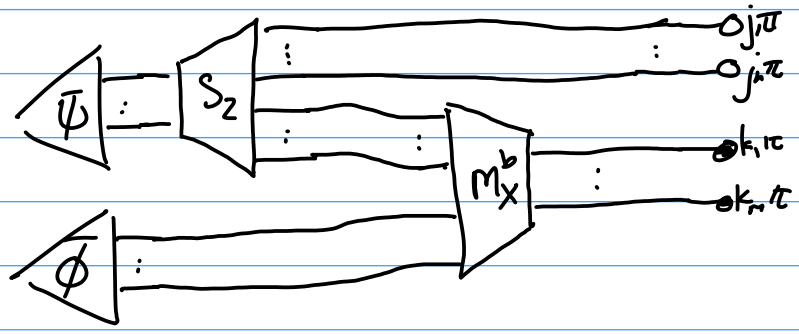


— Pauli measurements



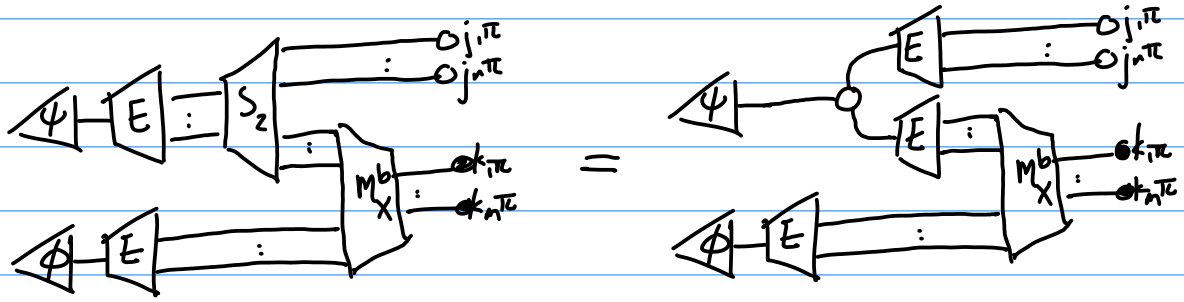
Encoded computation:



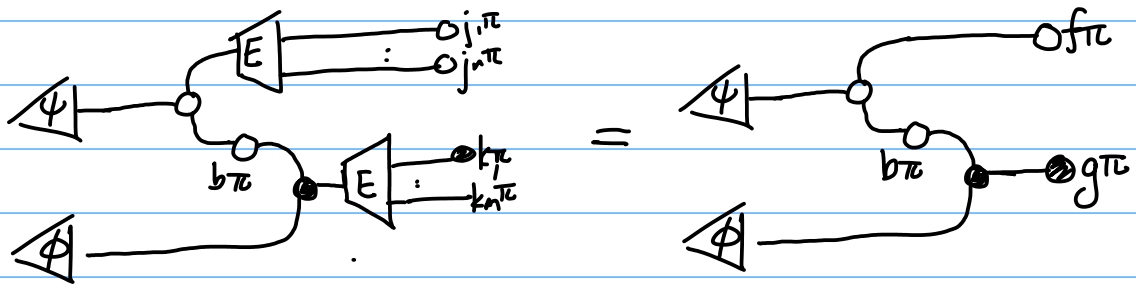


$= *$

$*=$



$=$



$=$