sparse codes from quantum circuits

arXiv:1411.3334

Dave Bacon
Steve Flammia
Aram Harrow
Jonathan Shi

Coogee
23 Jan 2015

QECC

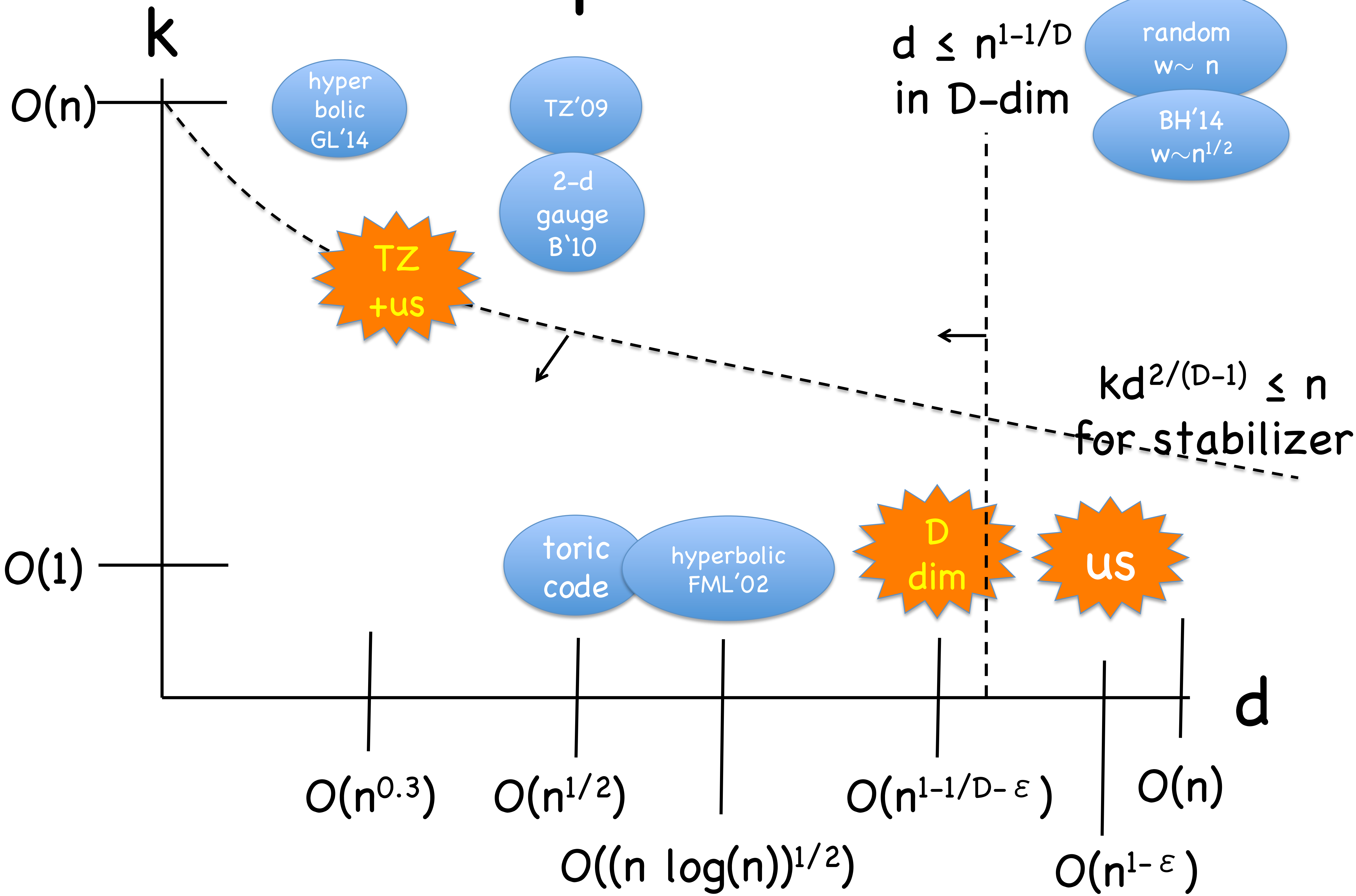
[n,k,d] code: encode k logical qubits in n physical qubits and correct errors on <d/2 positions.

[n,k,d,w]: ...using a decoding procedure that requires measurements of ≤w qubits at a time.

w=O(1) "LDPC" (low-density parity check)
Classically, possible with $k, d = \Omega(n)$.

WWSD principle → qLDPC

qLDPC?


k
O(n)
O(1)
hyper
bolic
GL'14
TZ'09
2-d
gauge
B'10
TZ
+us
d ≤ n^{1-1/D}
in D-dim
random
w∼ n
BH'14
w∼n^{1/2}
kd^{2/(D-1)} ≤ n
for stabilizer
toric
code
hyperbolic
FML'02
D
dim
us
d
O(n^{0.3})
O(n^{1/2})
O((n log(n))^{1/2})
O(n^{1-1/D-ε})
O(n)
O(n^{1-ε})

main results

More general theorem: Given an $[n,k,d]$ stabilizer code with a size-S Fault-Tolerant Error-Detecting Circuit we can construct an $[n'=O(S), k, d, w'=O(1)]$ subsystem code.

↓

Main Theorem: Given an $[n,k,d]$ stabilizer code with stabilizer weights $w_1, \ldots, w_{n-k}$, we can construct an $[n', k, d, w'=O(1)]$ subsystem code with $n' = O(n + \Sigma_i w_i)$.

↓

Subsystem codes exist with k=1, w=O(1), $d \sim n^{1-\frac{c}{\sqrt{\log n}}}$

↑

Also needed: New F-T E-D circuit for measuring a weight-w stabilizer using $O(w)$ gates.

stabilizer codes

- S = subgroup of $\pm\{I, X, XZ, Z\}^n$
- codespace V = $\{|\psi\rangle : s|\psi\rangle = |\psi\rangle$ for all $s \in S\}$
- Paulis anticommuting with some $s \in S$ are detected
- logical operators commute with all of S

3-bit repetition code

$S = \langle ZZI, IZZ\rangle = \langle I\otimes Z\otimes Z, Z\otimes Z\otimes I\rangle$

$V = \text{span}\{|000\rangle, |111\rangle\}$

logical operators $\langle XXX, ZII\rangle$

4-qubit code, distance 2

stabilizer generators		logical qubit 1		logical qubit 2	
XX	ZZ	ZI	XX	IZ	I I
XX	ZZ	ZI	I I	IZ	XX

subsystem/gauge codes

- Replace some logical qubits with "gauge" qubits:
  - Like logical qubits: Commute with stabilizers and errors. Contents can be arbitrary for logical code states.
  - Like stabilizer qubits: Don't care about preserving. Can (and should) measure during decoding.
- Advantages: sparsity, simpler decoding, (sometimes) better thresholds

4-qubit code, distance 2

stabilizer generators.		logical qubit.		gauge qubit.	
XX	ZZ	ZI	XX	ZZ	XI
XX	ZZ	ZI	I I	I I	XI

structure of subsystem codes

Gauge group $G \leq \pm\{I, X, XZ, Z\}^n$.
Center is stabilizer group: $S \cong Z(G)/\{\pm 1\}$
Normalizer is logical group: $L \cong N(G)/S$

Paulis

X_1	Z_1
X_2	Z_2
...	...
X_n	Z_n

<u>4-qubit code</u>

gauge generators

XI	IX	ZZ	I I
XI	IX	I I	ZZ

stabilizer subgroup generated by

XX	ZZ
XX	ZZ

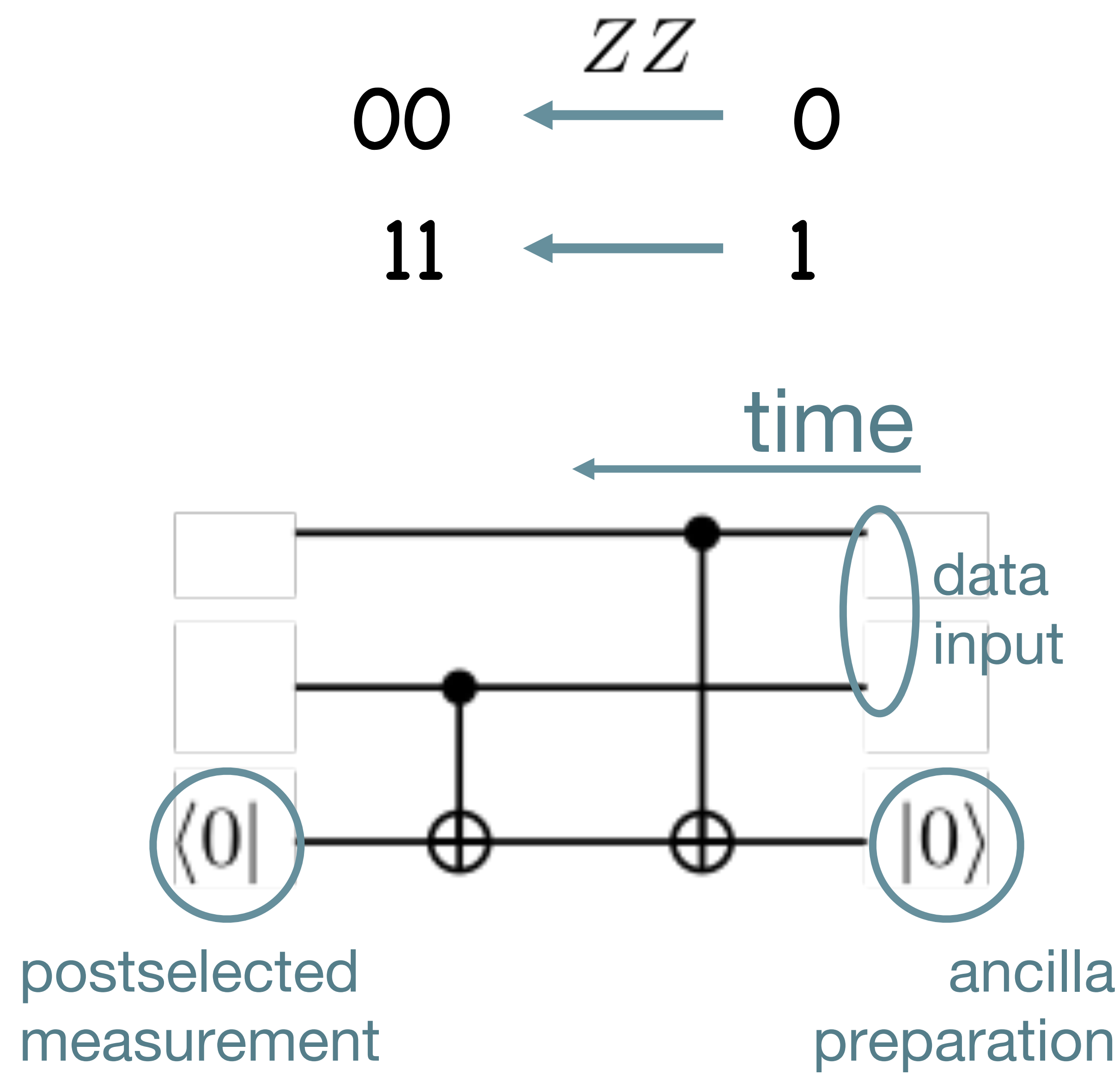
logical group generated by

ZI	XX
ZI	I I

stabilizers	errors
logical X operators	logical Z operators
gauge X operators	gauge Z operators

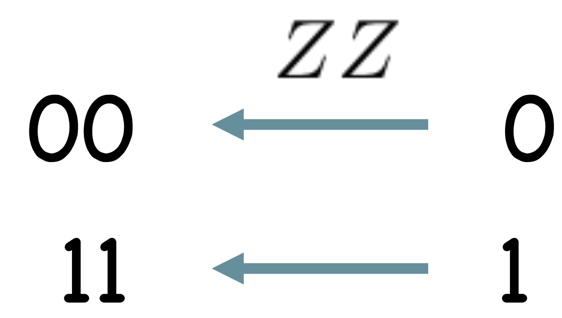
From Codes to Circuits to Codes Again...

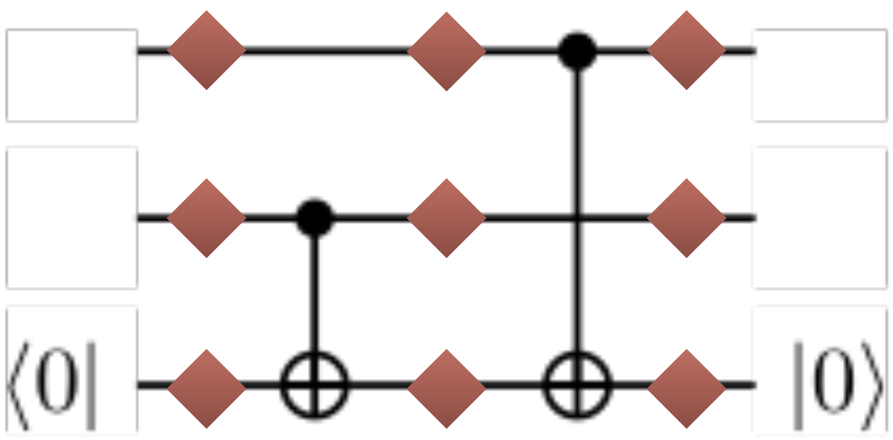
- Begin with a stabilizer code of your choice
- Write a quantum circuit for measuring the stabilizers of this code.



From Codes to Circuits to Codes Again...

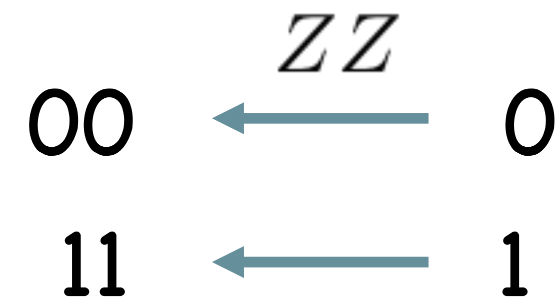
- Begin with a stabilizer code of your choice
- Write a quantum circuit for measuring the stabilizers of this code.
- Turn the circuit elements into input/ output qubits

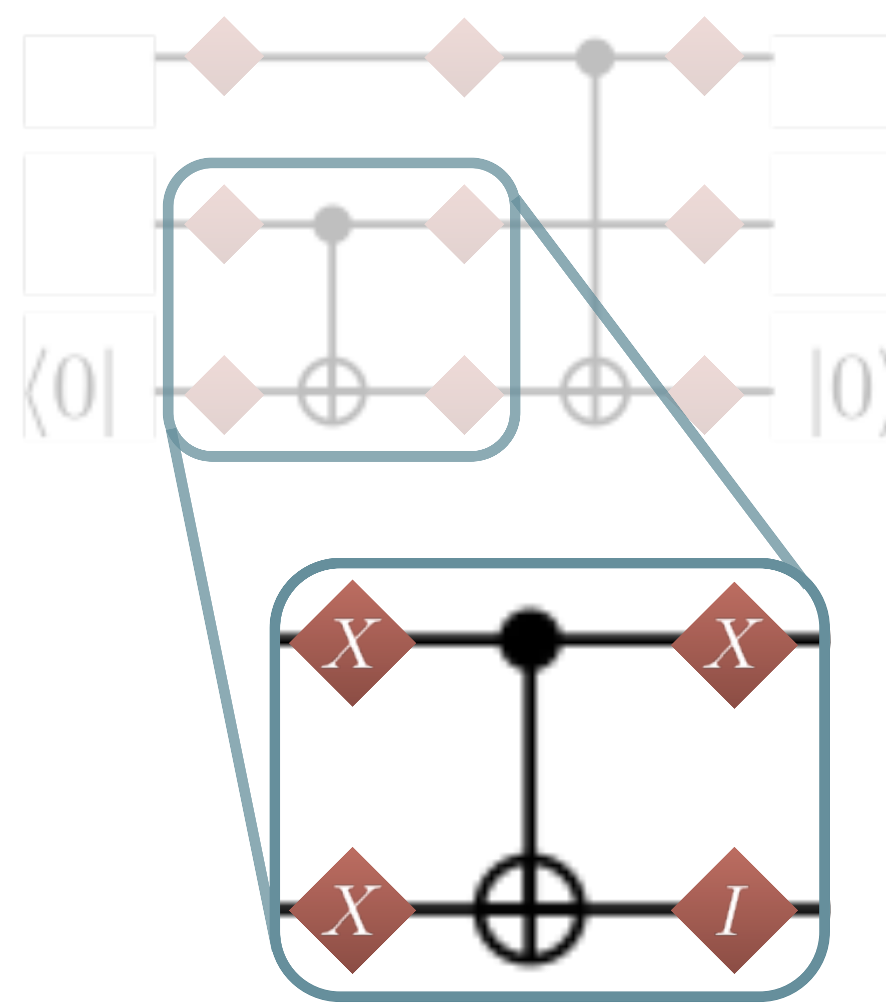




From Codes to Circuits to Codes Again...

- Begin with a stabilizer code of your choice
- Write a quantum circuit for measuring the stabilizers of this code.
- Turn the circuit elements into input/output qubits
- Add gauge generators via Pauli circuit identities.





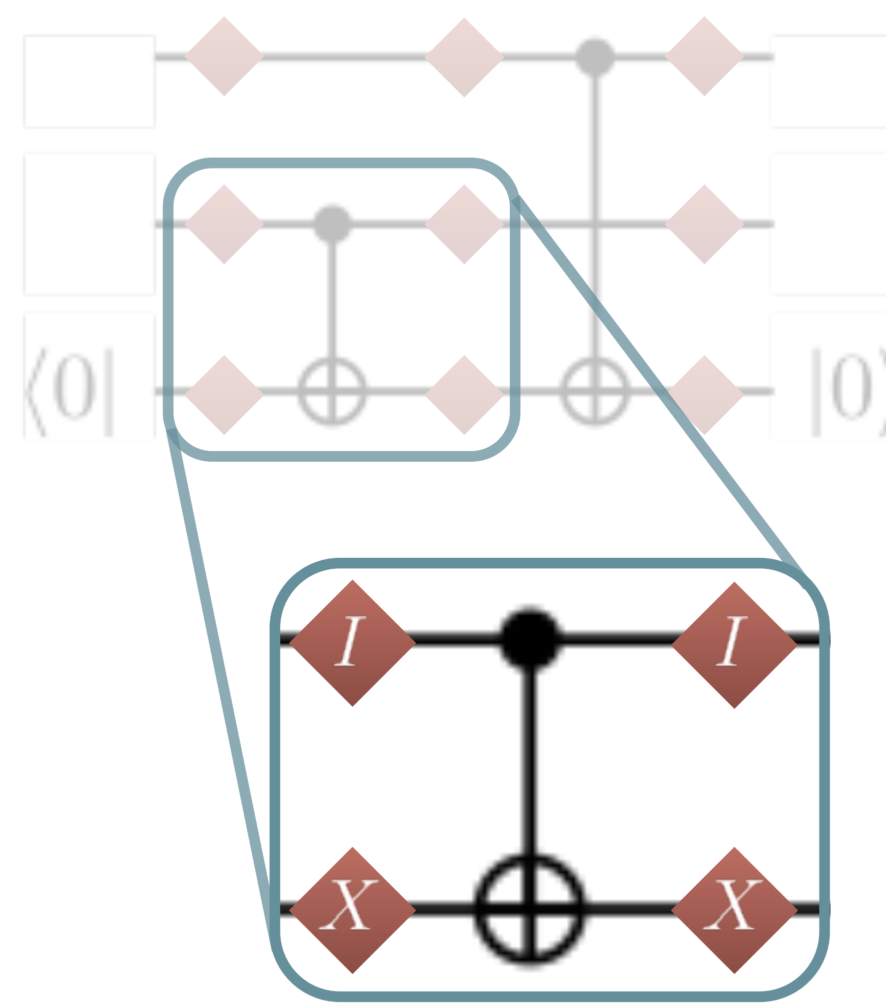
From Codes to Circuits to Codes Again...

- Begin with a stabilizer code of your choice
- Write a quantum circuit for measuring the stabilizers of this code.
- Turn the circuit elements into input/output qubits
- Add gauge generators via Pauli circuit identities.

ZZ

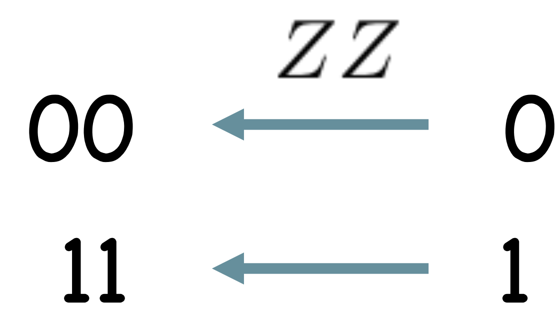
00 ← 0

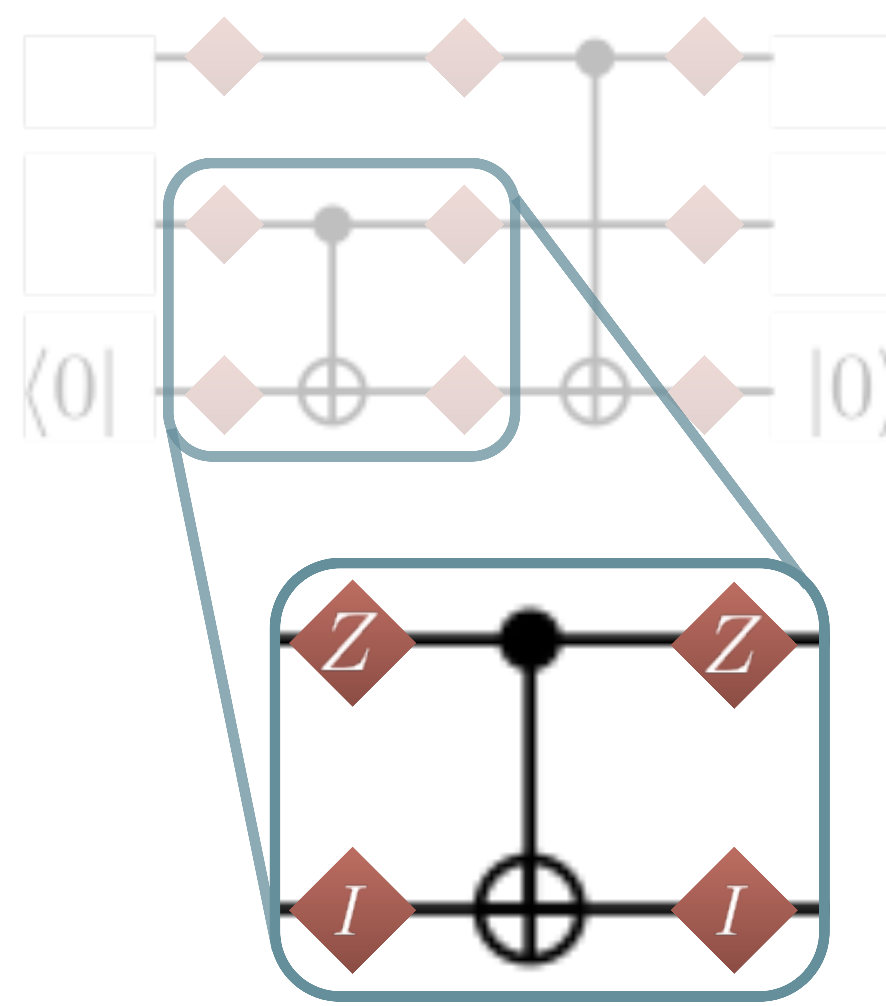
11 ← 1



From Codes to Circuits to Codes Again...

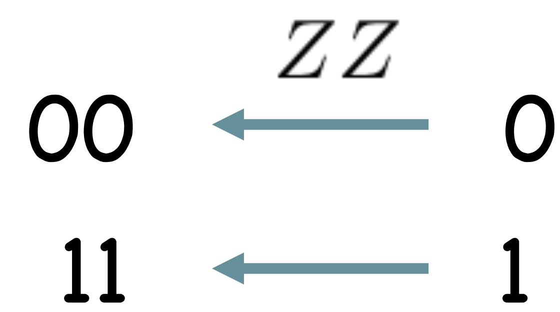
- Begin with a stabilizer code of your choice
- Write a quantum circuit for measuring the stabilizers of this code.
- Turn the circuit elements into input/output qubits
- Add gauge generators via Pauli circuit identities.

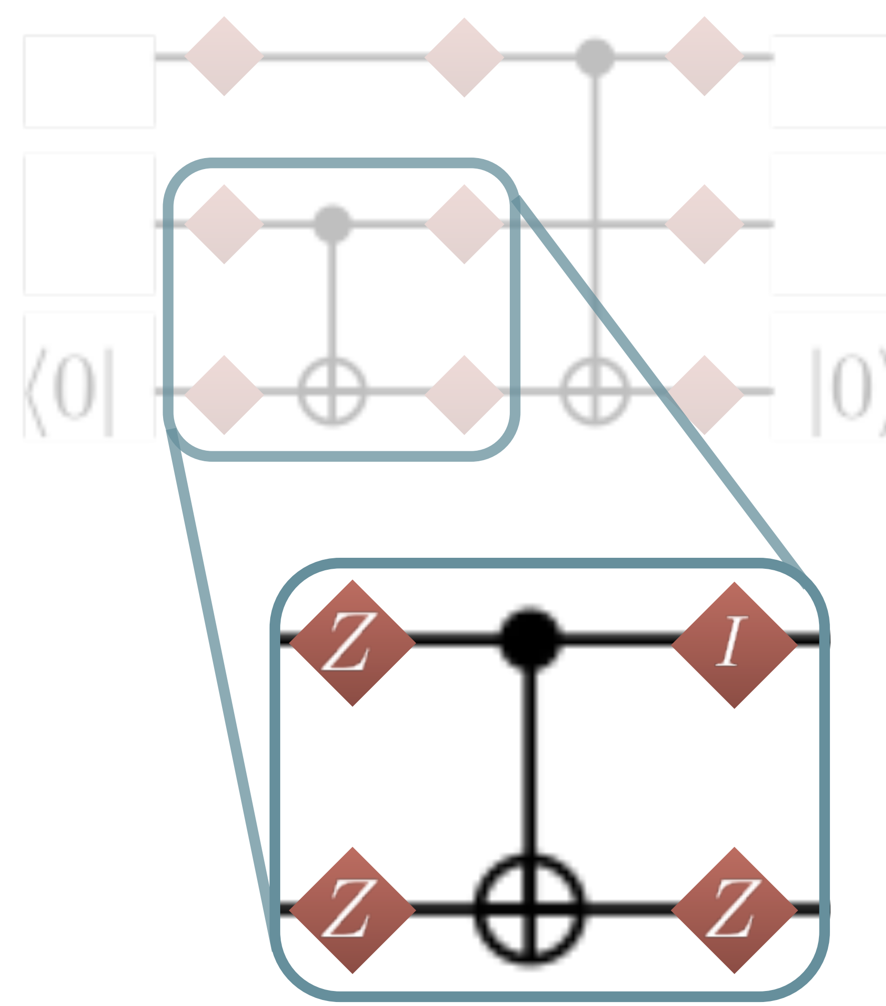




From Codes to Circuits to Codes Again...

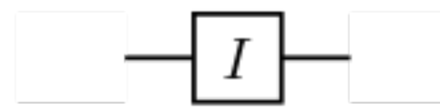
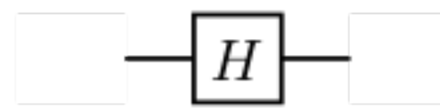
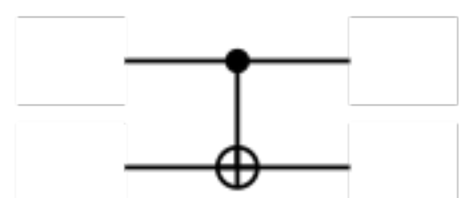
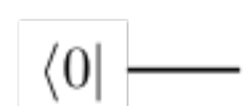
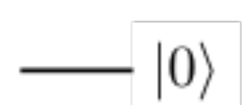
- Begin with a stabilizer code of your choice
- Write a quantum circuit for measuring the stabilizers of this code.
- Turn the circuit elements into input/output qubits
- Add gauge generators via Pauli circuit identities.





From Codes to Circuits to Codes Again...

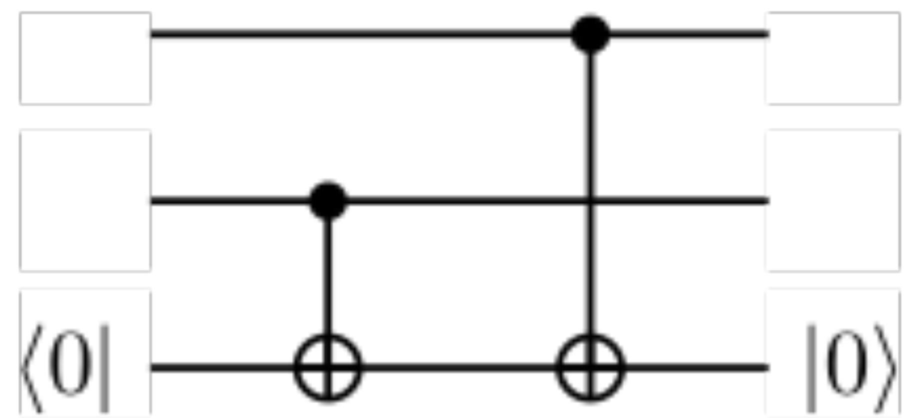
- Begin with a stabilizer code of your choice
- Write a quantum circuit for measuring the stabilizers of this code.
- Turn the circuit elements into input/output qubits
- Add gauge generators via Pauli circuit identities
- This defines the code

Circuit element	Gauge generators
	XX, ZZ
	ZX, XZ
	YX, ZZ
	$\begin{matrix}XX\\XI\end{matrix}, \begin{matrix}II\\XX\end{matrix}, \begin{matrix}ZZ\\II\end{matrix}, \begin{matrix}ZI\\ZZ\end{matrix}$
	Z
	Z

Bravyi 2011 does something similar with “generalized Bacon-Shor” codes

Properties of this Construction

- Circuits as linear operators preserving the code space

$V =$ 

$$V = |00\rangle\langle 00| + |11\rangle\langle 11|$$

$$\mathcal{C} = \mathrm{span}\big(\{|00\rangle, |11\rangle\}\big)$$

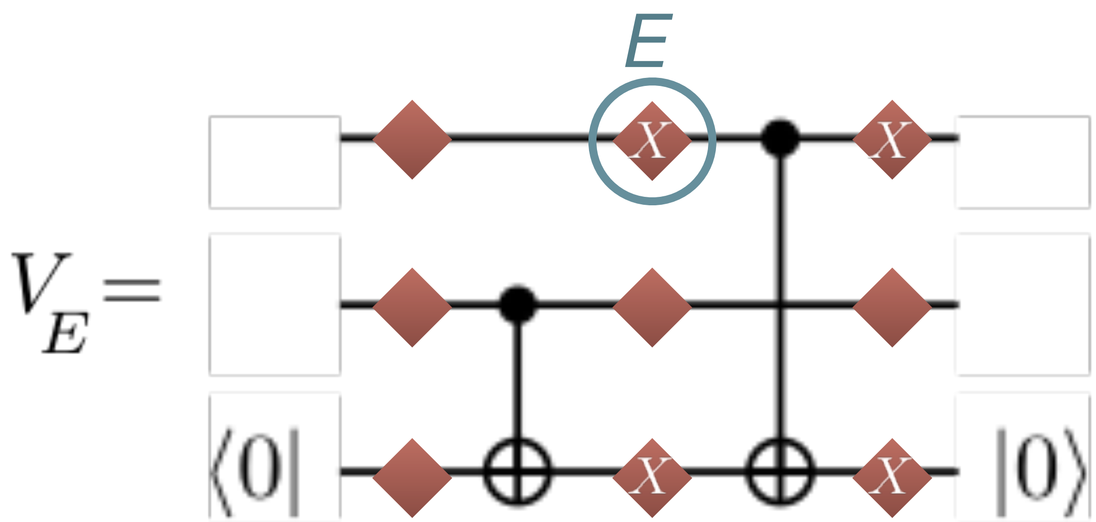
V is an *error-detecting circuit*

General condition:

V is E-D iff $V^\dagger V = \Pi_{\mathcal{C}}$

Properties of this Construction

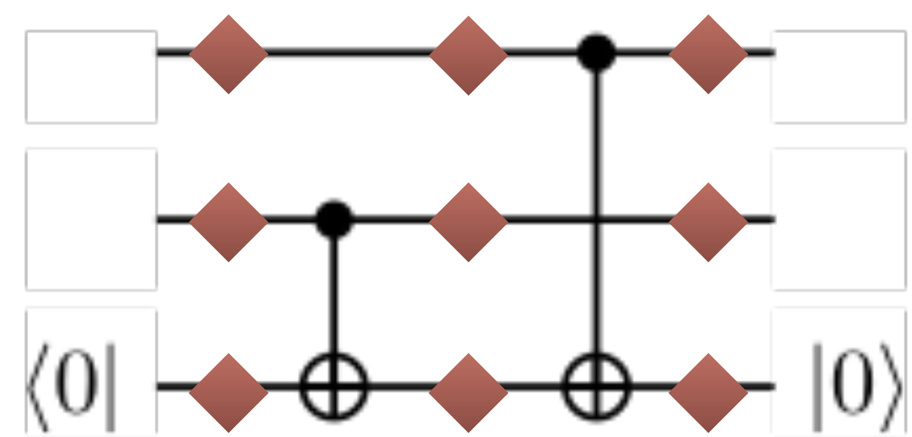
- Circuits as linear operators preserving the code space
- Gauge equivalence of errors: $V_E = \pm V_{GE}$



Apply gauge operators...

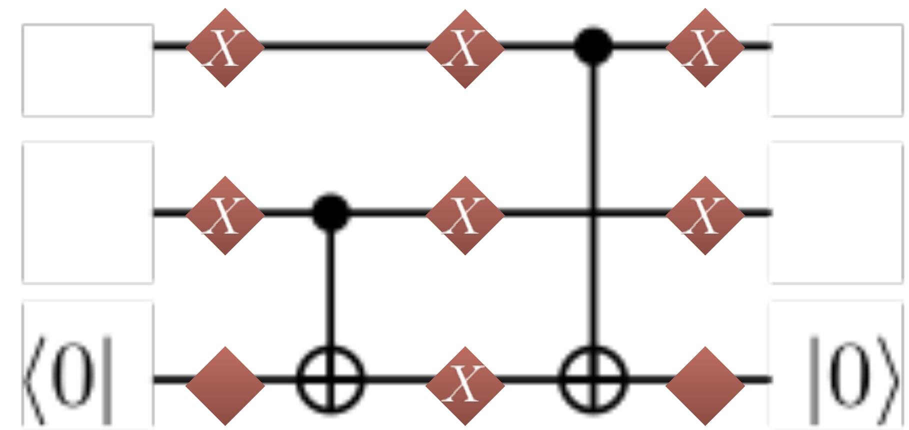
Properties of this Construction

- Circuits as linear operators preserving the code space
- Gauge equivalence of errors: $V_E = \pm V_{GE}$
- Squeegee lemma: using gauge operations, we can localize errors to the initial data qubits



Stabilizer and Logical Operators

- Spackling: like squeegee, but you leave a residue
- Spackling of logical operators gives the new logical operators
- Spackling of stabilizers on the inputs and ancillas are the new stabilizers
- Everything else is gauge or detectable error
- ...what about distance?



$$L_X = \begin{matrix} X & X & X \\ X & X & X \\ I & X & I \end{matrix} \quad L_Z = \begin{matrix} Z & Z & Z \\ I & I & I \\ I & I & I \end{matrix}$$

$$S = \begin{matrix} Z & Z & Z \\ Z & Z & Z \\ I & I & I \end{matrix} \quad S_a = \begin{matrix} Z & Z & I \\ Z & I & I \\ Z & Z & Z \end{matrix}$$

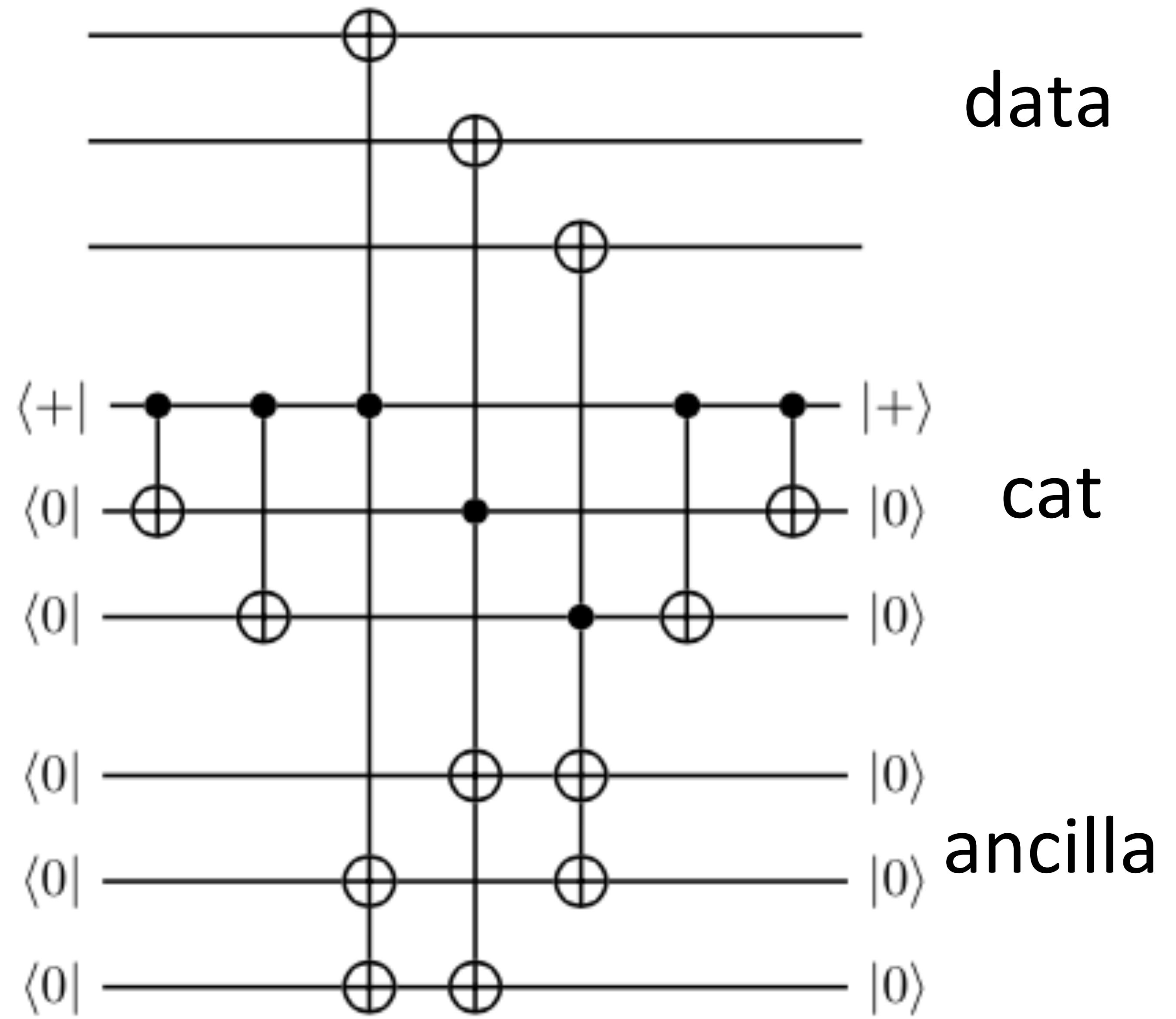
*even/odd effect means that circuits wires must have odd length

Code Distance and Fault Tolerance

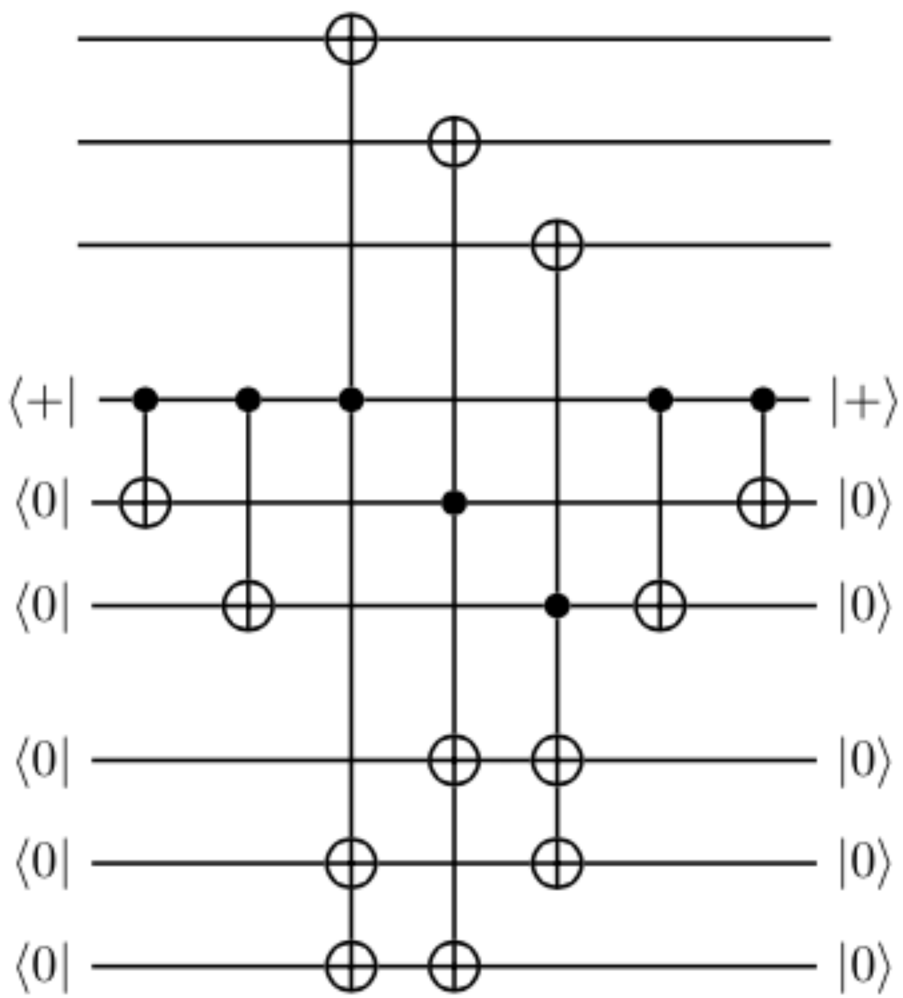
- For most error-detecting circuits, the new code is uninteresting (i.e. has bad distance).
- Theorem: If we use a fault-tolerant circuit then we preserve the code distance
- Fault tolerance definition: for every error pattern E, either $V_E = 0$ or there exists E' on inputs s.t. $V E' = V_E$ and $|E'| \leq |E|$.
- Idiosyncratic constraints:
  - Circuit must be Clifford (so no majority vote)
  - No classical feedback or post-processing allowed
  - However, we only need to detect errors

Fault-Tolerant Gadgets

- Use modified Shor/ DiVincenzo cat states
- Build a cat, and postselect ...not fault tolerant
- Redeem this idea by coupling to expanders
- constant-degree expanders exist with sufficient edge expansion to make this fault tolerant



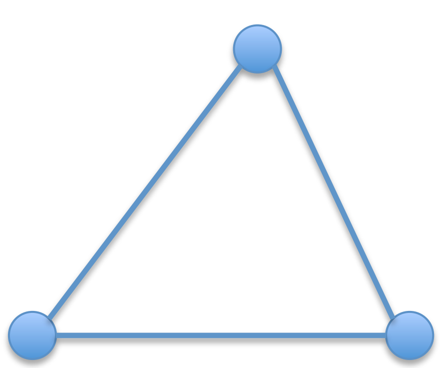
expander gadgets



data qubits ≅ {1, ..., n}

cat qubits ≅ V, |V|=n

ancilla qubits ≅ E



- Recipe: multiple-CNOT from each v to corresponding data qubit and all incident edges.
- Requirement: Edge expansion ≥ 1 means X errors on cat qubits cause more errors on ancillas.
- Corresponds to classical ECC with “energy barrier”.

Wake Up!

Theorem 1. *Given any $[n_0, k_0, d_0]$ quantum stabilizer code with stabilizer generators of weight $w_1, \ldots, w_{n_0-k_0}$, there is an associated $[n, k, d]$ quantum subsystem code whose gauge generators have weight $O(1)$ and where $k = k_0$, $d = d_0$, and $n = O(n_0 + \sum_i w_i)$. This mapping is constructive given the stabilizer generators of the base code.*

- Created sparse subsystem codes with the same k and d parameters as the base code
- Used fault-tolerant circuits in a new way, via expanders
- Extra ancillas are required according to the circuit size

Almost "Good" Sparse Subsystem Codes

- Start with an $[n_0, 1, d_0]$ random stabilizer code
  (so that $d_0 = O(n_0)$ with high probability)
- Concatenate this m times to get an $[n_0^m, 1, d_0^m]$ code
- Stabilizers: n_0^j of weight $\leq n_0^{m-j+1}$.
  Total weight $m \cdot n_0^{m+1}$
- Apply Theorem 1 with $m = (\log n)^{1/2}$

Sparse subsystem codes exist with
$d = O(n^{1-\varepsilon})$ and $\varepsilon = O(1/\sqrt{\log n})$.

Best previous distance for sparse codes was
$d = O(\sqrt{n \log n})$ by Freedman, Meyer, Luo 2002

*Thank you Sergei Bravyi!

Spatially Local Subsystem Codes Without Strings

- Take the circuit construction from the previous result
- Using SWAP gates and wires, spread the circuit over the vertices of a cubic lattice in D dimensions
- Let $n=L^D$ be the total number of qubits

Local subsystem codes exist with
$d = O(L^{D-1-\varepsilon})$ and $\varepsilon = O(1/\sqrt{\log n})$.

Compared to Known Bounds

- Local subsystem codes in D dimensions
  $d \leq O(L^{D-1})$
  - Our code: $d = \Omega(L^{D-1-\varepsilon})$
- Best known local stabilizer codes: $d = O(L^{D/2})$

- Local commuting projector codes
  $kd^{2/(D-1)} \leq O(n)$
  - Our codes: $kd^{2/(D-1)} = \Omega(n)$
    (use the hypergraph product codes and our main theorem)

Bravyi & Terhal 2009; Bravyi, Poulin, Terhal 2010; Tillich & Zémor 2009

* $\varepsilon = O((\log n)^{-1/2})$

Conclusion & Open Questions

- Showed a generic way to turn stabilizer codes into sparse subsystem codes
- New connection between quantum error correction & fault-tolerant quantum circuits
- What are the limits for sparse *stabilizer* codes?
- Self-correcting memory from the gauge Hamiltonian?
- Efficient, fault-tolerant decoding for these codes?
- Improve the rate? (Bravyi & Hastings 2013)
- Extend these results to allow for subsystem codes?
- Holography? ???
- See *arxiv:1411.3334* for more details!

The Best Sparse Codes

Code	k	d	Subsystem?	Decoder?
Z_2-systolic codes (Freedman, Meyer, Luo 2002)	O(1)	$O(\sqrt{n} \log n)$		
4D Hyperbolic (Hastings 2013)	$O(n)$	$O(\log n)$		
4D Arithmetic Hyperbolic (Guth & Lubotzky 2013)	$O(n)$	$O(n^{0.3})$		
Hypergraph Product (Tillich & Zémor 2009)	$O(n)$	$O(n^{0.5})$		
BFHS 2014 (this talk)*	O(1)	$O(n^{1-\varepsilon})$	yes	
Homological Product† (Bravyi & Hastings 2013)	$O(n)$	$O(n)$		

*subsystem code, $\varepsilon = O(1/\sqrt{\log n})$;
†sparsity $s = O(\sqrt{n})$; .

The Best (Euclidean) Local Codes

$n=L^D$

Code	D	k	d	Subsystem?	Decoder?
Toric Code (Kitaev 1996)	≥2	O(1)	$O(\sqrt{n})$		
Generalized Bacon-Shor (Bravyi 2011)	2	$O(L)$	$O(L)$	yes	
Welded Code (Michnicki 2012)	3	1	$O(L^{4/3})$		
Embedded Fractal (Brell 2014)	3'ish	$O(n)$	$O(n^{0.5})$		
Gauge Color Codes (Bombin 2013)	3	$O(n)$	$O(n)$	yes	
Gauge Color Codes (Bombin 2013)	3	$O(n)$	$O(n)$	yes	
BFHS 2014 (this talk)*	≥2	O(1)	$O(L^{D-1-\varepsilon})$	yes	

*subsystem code, $\varepsilon = O(1/\sqrt{\log n})$;
†sparsity $s = O(\sqrt{n})$; .

Local Subsystem Codes Without Strings

- Specialize to *D*=3
- Sparse subsystem code on a lattice with $[L^3, O(1), L^{2-\varepsilon}]$
- No strings, either for bare or dressed logical operators
  - cf. Bombin's gauge color codes
- ...on the other hand it's a *subsystem* code
- How does this compare to other candidate self-correcting quantum memories?

Comparing Candidate Self-Correcting Memories

Code	Self-correcting?	Comments
3D Bacon-Shor (Bacon 2005)	no	No threshold, so no self-correction (Pastawski *et al*. 2009)
Welded Code (Michnicki 2014)	no	See Brown *et al*. 2014 review article for discussion
Cubic Code (Haah 2011)	marginal	poly(L) memory lifetime for $L< e^{\beta/3}$ (Bravyi & Haah 2013)
Embedded Fractal Product Codes (Brell 2014)	maybe	very large ground-state degeneracy?
Gauge Color Codes (Bombin 2013)	???	Does have a threshold, also has string-like dressed operators
This talk (BFHS 2014)	???	No strings, concatenated codes have a threshold

Not depicted: Codes with long-range couplings (e.g. several works by the Loss group) or Hamma *et al.* 2009
See the talk by Olivier Landon-Cardinal on Friday for more discussion of these types of codes.

Challenges with Gauge Hamiltonians

- Gauge Hamiltonians are sometimes gapped:
  (Kitaev 2005; Brell *et al.* 2011; Bravyi *et al.* 2013)
- ...but sometimes not:
  (Bacon 2005; Dorier, Becca, & Mila 2005)
- The simplest example of our code (a wire) reduces to Kitaev's quantum wire, which is gapped as long as the couplings aren't equal in magnitude
- Our codes are a vast generalization of Kitaev's wire to arbitrary circuits!
- This undoubtedly has a rich phase diagram... might there be a gapped self-correcting phase, or something more?

Kitaev 2001; Lieb, Schultz, & Mattis 1961