2017-05-30 Fault tolerance with few qubits (talk)

Friday, May 19, 2017 11:19 AM

FAULT-TOLERANT QUANTUM COMPUTATION WITH FEW QUBITS

Ben Reichardt
USC

with Rui Chao

Quantum error correction with only two extra qubits

Rui Chao, Ben W. Reichardt https://arxiv.org/abs/1705.02329

(Submitted on 5 May 2017)

Noise rates in quantum computing experiments have dropped dramatically, but reliable qubits remain precious. Fault-tolerance schemes with minimal qubit overhead are therefore essential. We introduce fault-tolerant error-correction procedures that use only two ancilla qubits. The procedures are based on adding "flags" to catch the faults that can lead to correlated errors on the data. They work for various distance-three codes.
In particular, our scheme allows one to test the [[5,1,3]] code, the smallest error-correcting code, using only seven qubits total. Our techniques also apply to the [[7,1,3]] and [[15,7,3]] Hamming codes, thus allowing to protect seven encoded qubits on a device with only 17 physical qubits.

Fault-tolerant quantum computation with few qubits

Rui Chao, Ben W. Reichardt https://arxiv.org/abs/1705.05365

(Submitted on 15 May 2017)

Reliable qubits are difficult to engineer, but standard fault-tolerance schemes use seven or more physical qubits to encode each logical qubit, with still more qubits required for error correction. The large overhead makes it hard to experiment with fault-tolerance schemes with multiple encoded qubits.
The 15-qubit Hamming code protects seven encoded qubits to distance three. We give fault-tolerant procedures for applying arbitrary Clifford operations on these encoded qubits, using only two extra qubits, 17 total. In particular, individual encoded qubits within the code block can be targeted. Fault-tolerant universal computation is possible with four extra qubits, 19 total. The procedures could enable testing more sophisticated protected circuits in small-scale quantum devices.
Our main technique is to use gadgets to protect gates against correlated faults. We also take advantage of special code symmetries, and use pieceable fault tolerance.

Summary: More efficient fault-tolerance

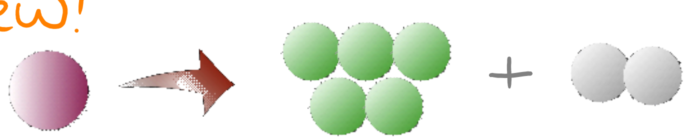
Previous methods:

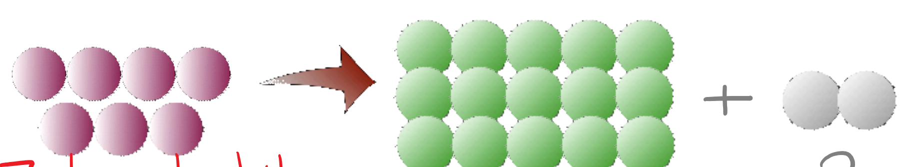
logical qubit — 5 physical qubits for the code + 5 for error correction [Shor '96]

5 + 4 [DiVincenzo-Aliferis '07]

5 + 3 [Yoder-Kim '16]

New!





7 logical qubits — 15 + 2

physical qubits for the code

for error correction and COMPUTATION (Clifford)

+4 = 19 for universality

QUBITS ARE NOISY!

Typical noise rates: 10^{-2} to 10^{-4} error per gate

Operation	Current duration	Current infidelity	Anticipated duration	Anticipated Infidelity
Single-qubit gates	$5\mu s$	$5\cdot10^{-5}$	$1\mu s$	$1\cdot10^{-5}$
Entangling (2 qubits)	$40\mu s$	$1\cdot10^{-2}$	$15\mu s$	$2\cdot10^{-4}$
Entangling (5 qubits)	$60\mu s$	$5\cdot10^{-2}$	$15\mu s$	$1\cdot10^{-3}$
Dual species entangling (2 qubits)	$60\ \mu s$	$3\cdot10^{-2}$	$15\ \mu s$	$4\cdot10^{-4}$
Dual species entangling (3 qubits)	$80\ \mu s$	$5\cdot10^{-2}$	$15\ \mu s$	$6\cdot10^{-4}$
Dual species entangling (5 qubits)	-	-	$15\ \mu s$	$2\cdot10^{-3}$
Measurement	$400\mu s$	$1\cdot10^{-3}$	$30\mu s$	$1\cdot10^{-4}$
Re-cooling	$400\mu s$	$\bar{n}<0.1$	$100\mu s$	$\bar{n}<0.1$
Qubit reset	$50\mu s$	$5\cdot10^{-3}$	$10\mu s$	$5\cdot10^{-3}$ *

Assessing the progress of trapped-ion processors towards fault-tolerant quantum computation

A. Bermudez, X. Xu, R. Nigmatullin, J. O'Gorman, V. Negnevitsky, P. Schindler, T. Monz, U. G. Poschinger, C. Hempel, J. Home, F. Schmidt-Kaler, M. Biercuk, R. Blatt, S. Benjamin, M. Müller

https://arxiv.org/abs/1705.02771

Shor's algorithm

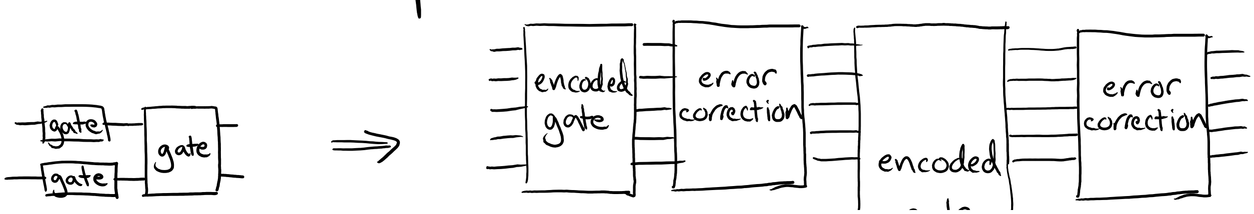
factors a 1024-bit numbers

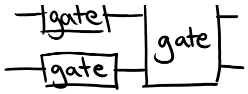
using 10^{11} gates on 5000 qubits

$\Rightarrow$ need error $< 10^{-11}$ per gate

FAULT TOLERANCE IS AMAZING!

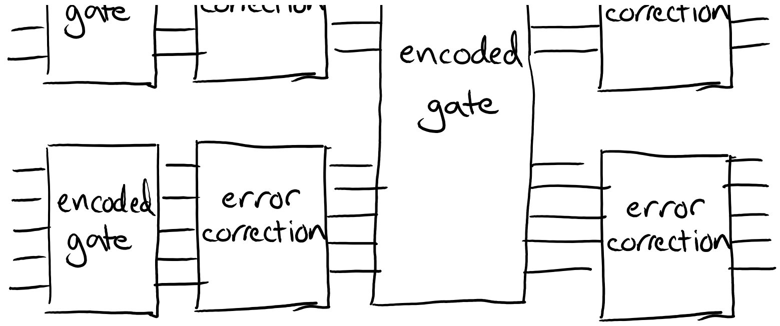
- Noise is digital (X, Y, Z)
- Error-correcting codes exist
- We can compute with them!





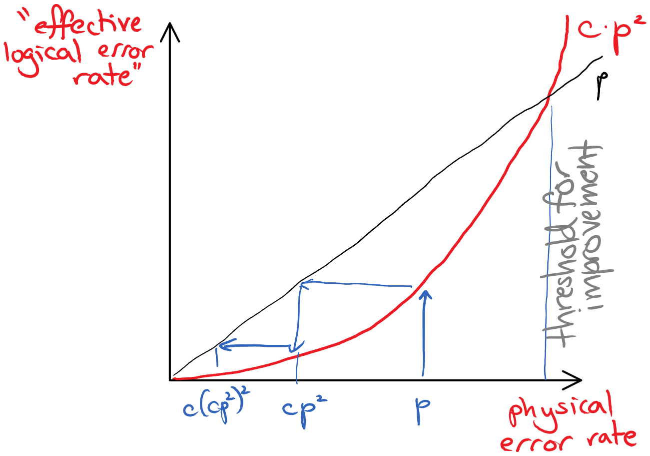
ideal circuit

$\Rightarrow$



encoded circuit

Distance 3 code $\Rightarrow$ 1 error okay, 2 is bad



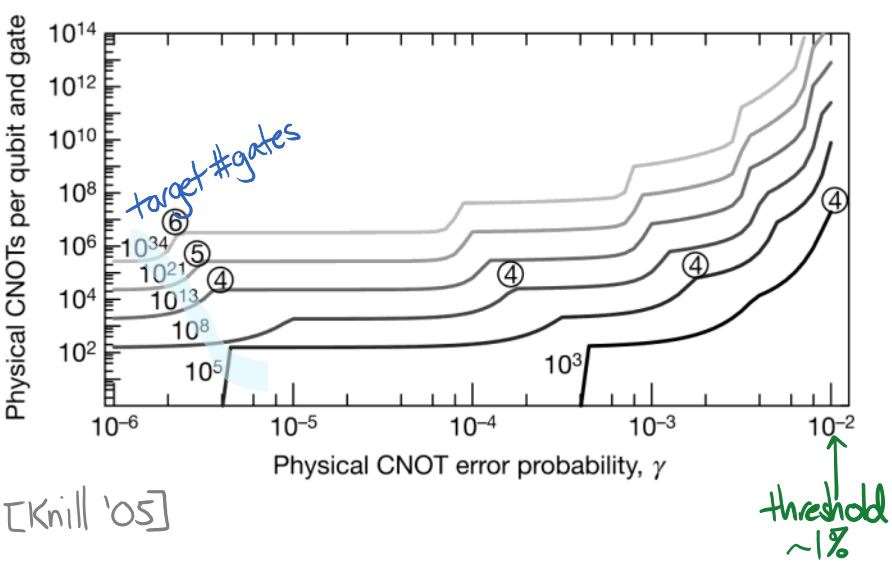
Concatenate scheme for arbitrary reliability

levels of coding	error rate	overhead
0	p	1
1	cp^2	5
2	c^3p^4	25
3	c^7p^8	125
⋮		
k	$c^{2^k-1}p^{2^k}$	5^k

$p^{2^k} = p^*$ $\Rightarrow k \approx \log\log p^*$

$5^k \approx \log p^*$

Fault tolerance has HIGH OVERHEAD



[Knill '05]

Goal: Implement fault-tolerant error correction and computation on small quantum devices

- to test/demonstrate the theory
- to assess FT schemes' performance in real error models
- to adapt FT schemes to real noise

Quantum codes

$$\alpha|0\rangle + \beta|1\rangle$$

$$\alpha|000\rangle + \beta|111\rangle$$

distance 3 for X errors

$$\xrightarrow{X_3} \alpha|001\rangle + \beta|110\rangle$$

$$\xrightarrow{Z_3} \alpha|000\rangle - \beta|111\rangle$$

$$\alpha\begin{pmatrix}|0000\rangle \\ +|1111\rangle\end{pmatrix} + \beta\begin{pmatrix}|0011\rangle \\ +|1100\rangle\end{pmatrix}$$

$d=2$ for X, Y, Z errors

any single error maps to orthogonal subspace

$\Rightarrow$ can be detected (not corrected)

Common codes

	Qubits	Distance
• ~~Shor's code~~ Bacon-Shor	9	3 → can correct one error
• Steane code	7	3
• Golay code	23	7 can correct

Shor's code: repetition on dual repetition

000/111 +++/−−−

Steane code: self-dual CSS

IIIXXXX   IIIZZZZ

IXXIIXX   IZZIIZZ

XIXIXIX   ZIZIZIZ

- Surface codes
  good planar embedding

3 errors

Many other codes

n #qubits	d distance	
5	3	(10 total)
31 RM	7	
47 QR	11	
49 Steane^2	9	
79 QR	15	
103	19	

n/k	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1	1	1																													
2	2	1	1																												
3	2	1	1	1																											
4	2	2	2	1	1																										
5	3	3	2	1	1	1																									
6	4	3	2	2	2	1	1																								
7	3	3	2	2	2	1	1	1																							
8	4	3	3	3	2	2	2	1	1																						
9	4	3	3	3	2	2	2	1	1	1																					
10	4	4	4	3	3	2	2	2	2	1	1																				
11	5	5	4	3	3	3	2	2	2	1	1	1																			
12	6	5	4	4	4	3	3	2	2	2	2	1	1																		
13	5	5	4	4	4	3	3	3	2	2	2	1	1	1																	
14	6	5	5	4-5	4	4	4	3	3	2	2	2	2	1	1																
15	6	5	5	5	4	4	4	3	3	3	2	2	2	1	1	1															
16	6	6	6	5	5	4-5	4	4	3	3	3	2	2	2	2	1	1														
17	7	7	6	5-6	5	4-5	4-5	4	4	4	3	3	2	2	2	1	1	1													
18	8	7	6	5-6	5-6	5	5	4	4	4	3	3	2	2	2	2	2	1	1												
19	7	7	6	5-6	5-6	5-6	5	4-5	4	4	3-4	3	3	2	2	2	2	1	1	1											
20	8	7	6-7	6-7	6	5-6	5-6	4-5	4-5	4	4	3-4	3	3	2	2	2	2	2	1	1										
21	8	7	6-7	6-7	6-7	6	5-6	5-6	4-5	4-5	4	4	3-4	3	3	3	2	2	2	1	1	1									
22	8	7-8	6-8	6-7	6-7	6-7	5-6	5-6	5-6	4-5	4-5	4	4	3-4	3	3	2	2	2	2	2	1	1								
23	8-9	7-9	7-8	6-8	6-7	6-7	5-7	5-6	5-6	4-6	4-5	4-5	4	4	3-4	3	3	2	2	2	2	1	1	1							
24	8-10	8-9	7-8	7-8	6-8	6-7	6-7	5-7	5-6	5-6	5-6	4-5	4-5	4	4	3-4	3	3	2	2	2	2	2	1	1						
25	8-9	9	7-8	7-8	7-8	7-8	6-7	5-7	5-7	5-6	5-6	4-6	4-5	4-5	4	4	3-4	3	3	2	2	2	2	1	1	1					
26	8-10	9	8-9	8-9	8	7-8	6-8	6-8	6-7	5-7	5-6	5-6	5-6	4-5	4-5	4	4	3-4	3	3	2	2	2	2	2	1	1				
27	9-10	9	9	9	8-9	7-8	6-8	6-8	6-8	6-7	5-7	5-6	5-6	5	4-5	4-5	4	4	3-4	3	3	2	2	2	2	1	1	1			
28	10	10	10	9	8-9	7-9	6-8	6-8	6-8	6-8	6-7	6-7	6	5-6	5-6	4-5	4	4	4	3-4	3	3	2	2	2	2	2	1	1		
29	11	11	10	9-10	8-9	7-9	7-9	6-8	6-8	6-8	6-7	6-7	6	5-6	5-6	4-5	4-5	4	4	4	3-4	3	3	2	2	2	2	1	1	1	
30	12	11	10	9-10	8-10	8-9	7-9	7-9	7-8	6-8	6-8	6-7	6-7	5-6	5-6	5-6	5	4-5	4	4	4	3-4	3	3	2	2	2	2	2	1	1
n/k	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30

http://codetables.de

- more efficient to encode multiple qubits per block

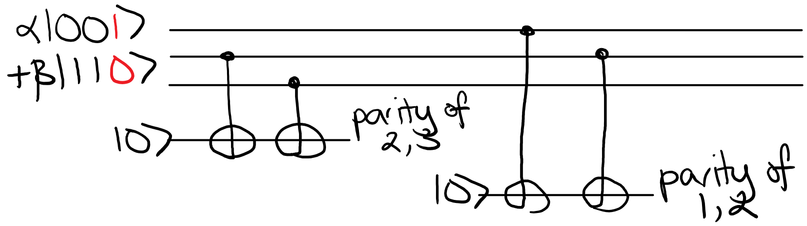
	n	k #logical qubits	d
	8	3	3
	10	4	3
	11	5	3
	⋮	⋮	
Hamming codes	15	7	3
	31	21	3
	63	51	3
	127	113	3
BCH codes	31	11	5
	63	27	7
	127	29	15

Quantum error correction

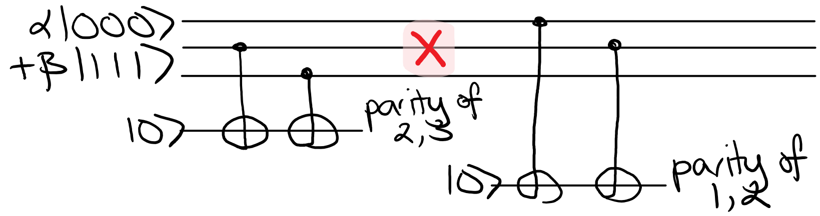
$$\alpha|000\rangle + \beta|111\rangle \xrightarrow[\text{noise}]{X_3} \alpha|001\rangle + \beta|110\rangle$$

Correction

Measuring the qubits, you'll find the error...
but also collapse the state!
Instead, measure the parities ("stabilizers")

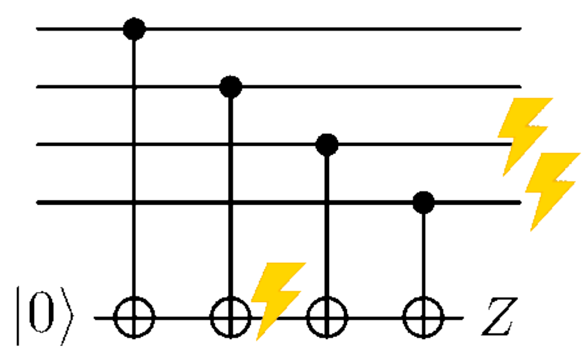


Watch out for faults in error correction!



Possible solution: If a syndrome is nontrivial,
measure them all again before making the correction.

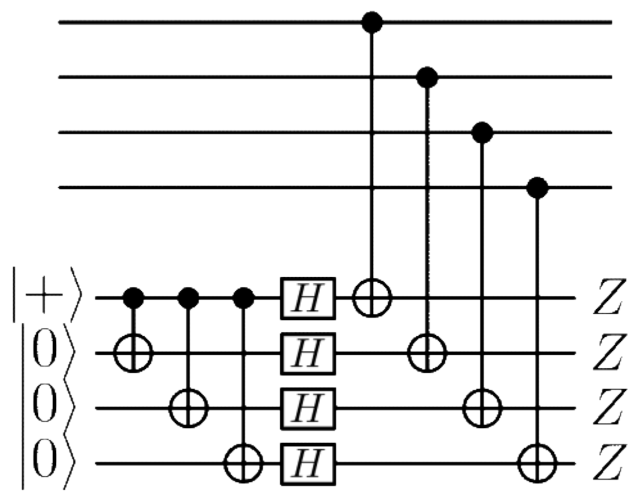
Bigger problem:
Errors can spread

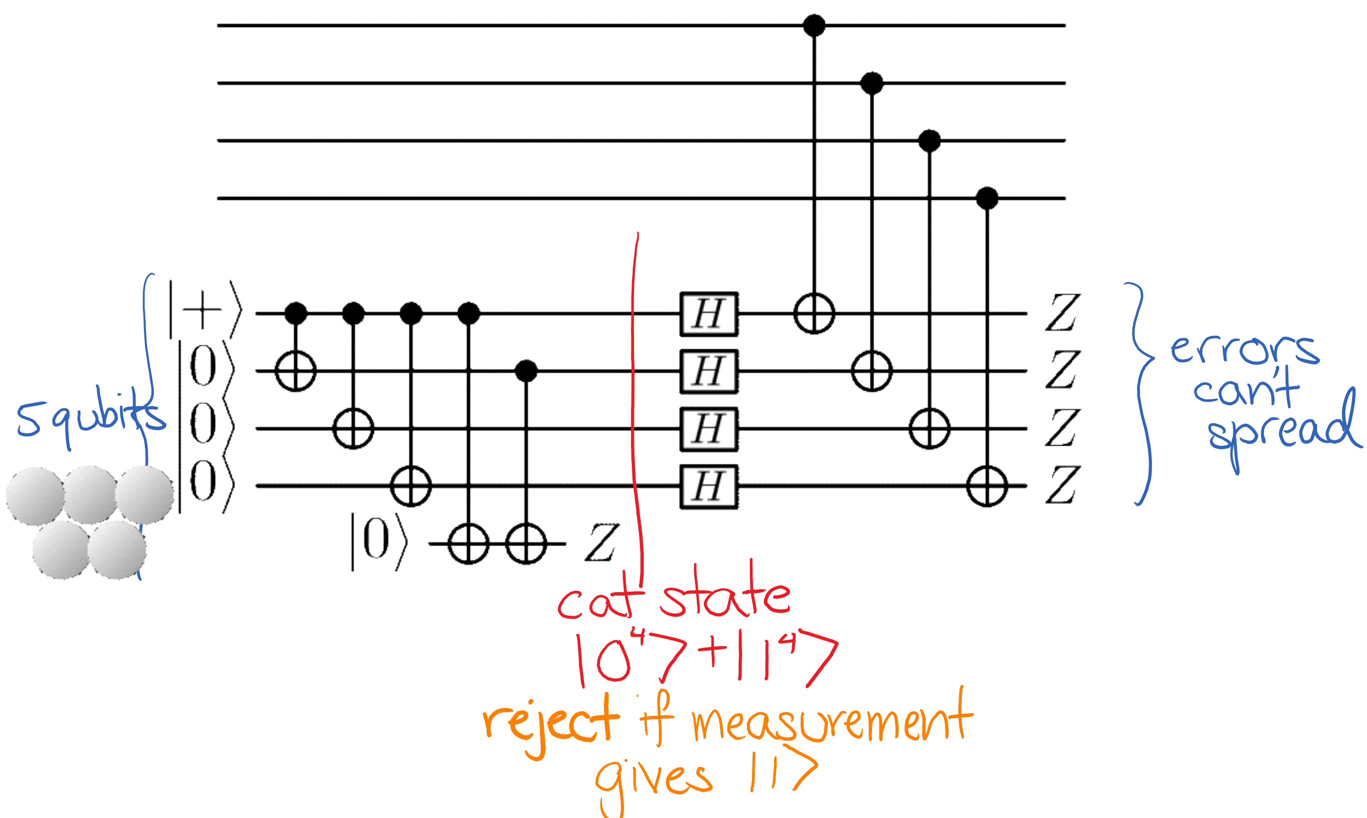


Previous approaches to avoid spreading errors

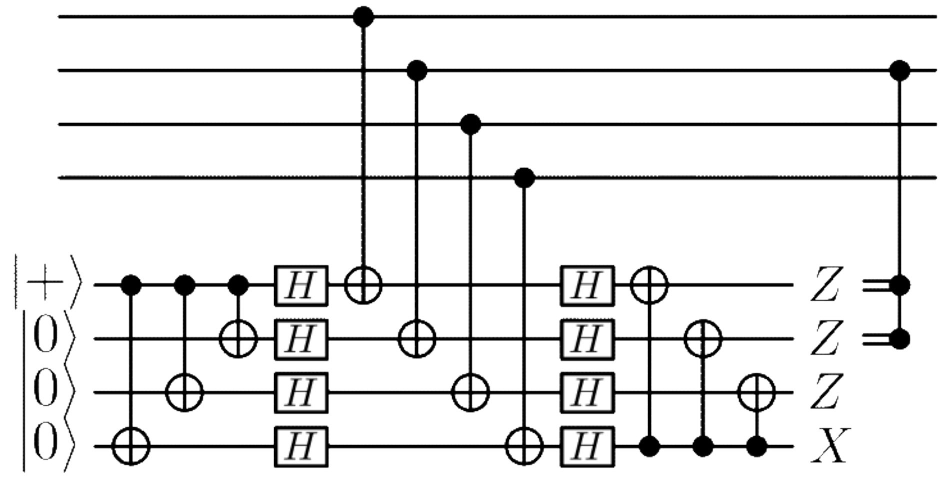
1. Shor
2. DiVincenzo-Aliferis
3. Stephens-Yoder-Kim

① Shor

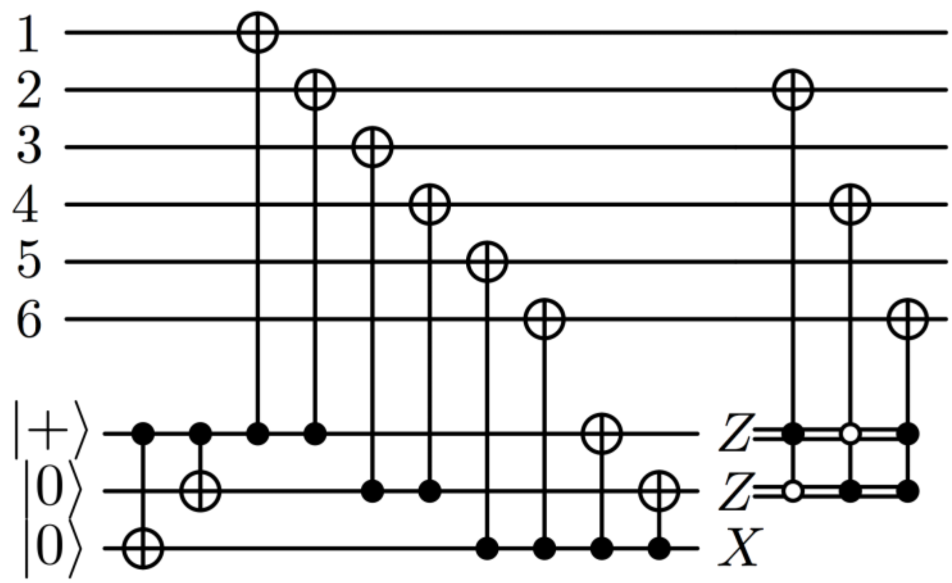


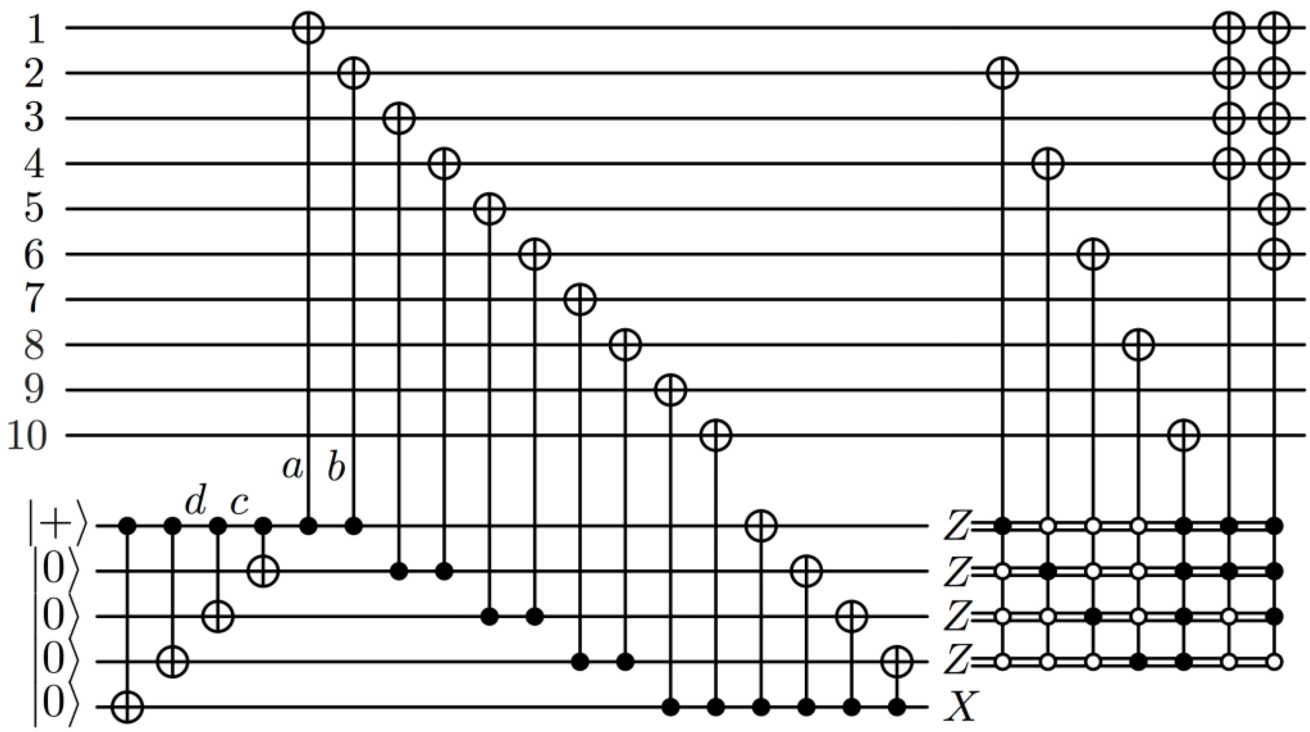


② DiVincenzo-Aliferis



③ Stephens-Yoder-Kim





Ancilla qubits required for FT syndrome extraction

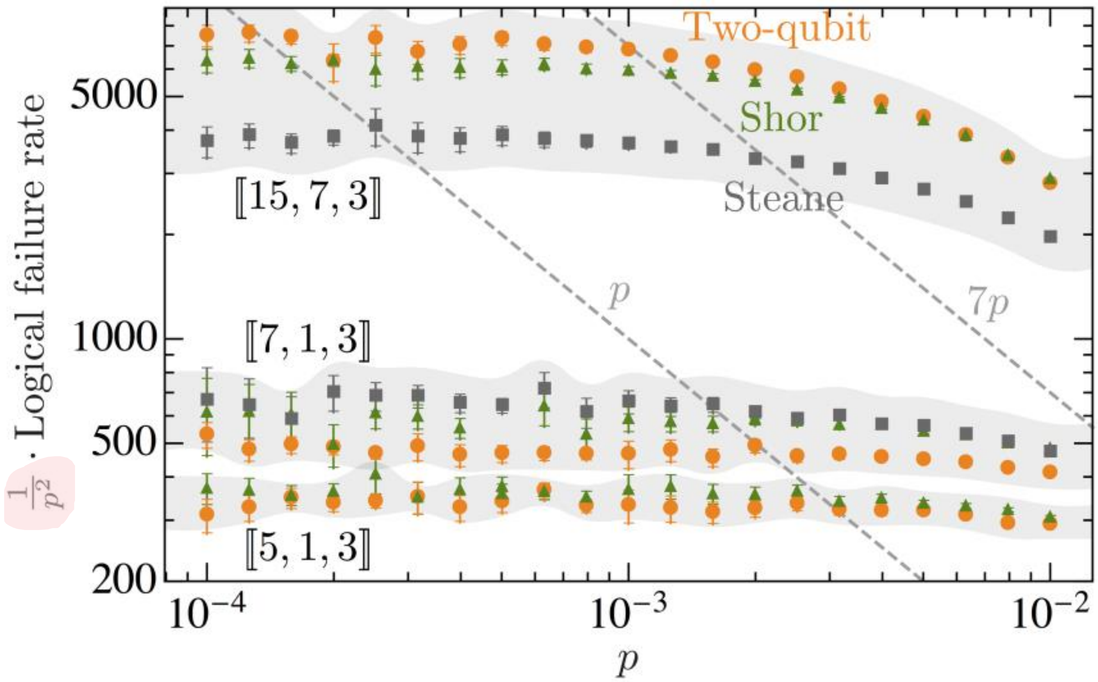
	Shor	DiVincenzo-Aliferis	Stephens-Yoder-Kim	d=3 New method
ω = stabilizer weight	$\omega+1$	ω	$\max\{3, \lceil \omega/2 \rceil\}$	2

*for all codes we have tried

Code	Ancilla qubits required for Shor cat state	Decoded half cat	Flagged
$[\![5,1,3]\!]$	5	3	2
$\diamond$ $[\![7,1,3]\!]$	5	3	2 [8]
$[\![9,1,3]\!]$	1	–	–
$[\![8,3,3]\!]$	7	3	2
$[\![10,4,3]\!]$	9	4	2
$[\![11,5,3]\!]$	9	4	2
$\diamond$ $[\![15,7,3]\!]$	9	4	2
$\diamond$ $[\![31,21,3]\!]$	17	8	2
$\diamond$ $[\![2^r-1, 2^r-1-2r, 3]\!]$	$2^{r-1}+1$	2^{r-2}	2

Simulation

(depolarizing noise
No geometry, No rest error)

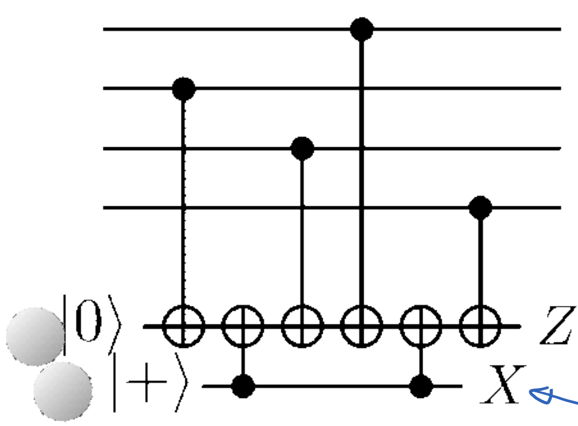


Main problem: Errors can spread.

Previous approaches: Try to avoid this

Our Main idea: Catch the errors that can spread.

Our 2-qubit method



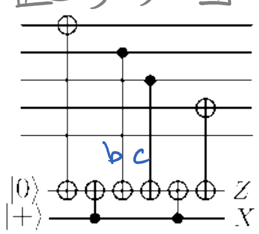
the order of gates can matter
— the bad errors must be distinguishable

parity goes here

this detects faults that can spread to weight 2 on data

Real examples: This works for many codes.

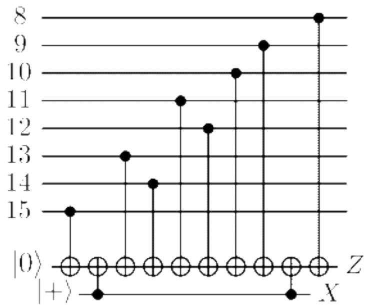
[[5,1,3]]:



(b) failure	Data error
IZ	$IIZXI$
XZ	$IXZXI$
YZ	$IYZXI$
ZZ	$IZZXI$

(c) failure	Data error
IZ	$IIIXI$
XZ	$IIXXI$
YZ	$IIYXI$
ZZ	$IIZXI$

[[15,7,3]]:



stabilizers:

0 0 0 0 0 0 0 1 1 1 1 1 1 1 1
0 0 0 1 1 1 1 0 0 0 0 1 1 1 1
0 1 1 0 0 1 1 0 0 1 1 0 0 1 1
1 0 1 0 1 0 1 0 1 0 1 0 1 0 1

possible correlated errors:

$\mathbf{1}, Z_8, Z_{\{8,9\}}, Z_{\{8,9,10\}}, Z_{\{8,9,10,12\}}, Z_{\{8,9,10,11,12\}}, Z_{\{8,9,10,11,12,14\}}, Z_{\{8,9,10,11,12,13,14\}}$

Fault-tolerant computation

Previous approach for computation

It's much easier when each block encodes only one qubit.

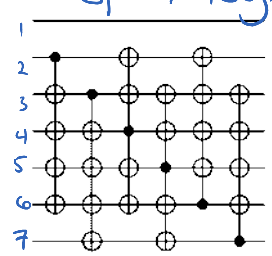
[Gottesman '97]: - Teleport one logical qubit into its own block
- Work there
- Teleport it back

[Harrington, R. '12]
[Grassl, Roetteler '13]

Use code's permutation symmetries

$$\sigma_1 = (1,2,3)(4,14,10)(5,12,9)(6,13,11)(7,15,8)$$
$$\sigma_2 = (1,10,5,2,12)(3,6,4,8,9)(7,14,13,11,15)$$
$$\sigma_3 = (1,10,15,3,8,13)(4,6)(5,12,11)(7,14,9)$$

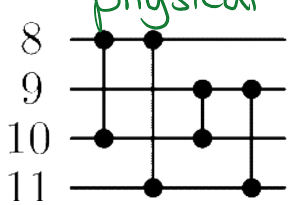
} generate all even permutations of 7 logical qubits



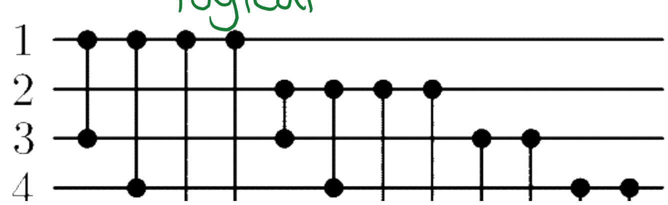
Our approach for computation

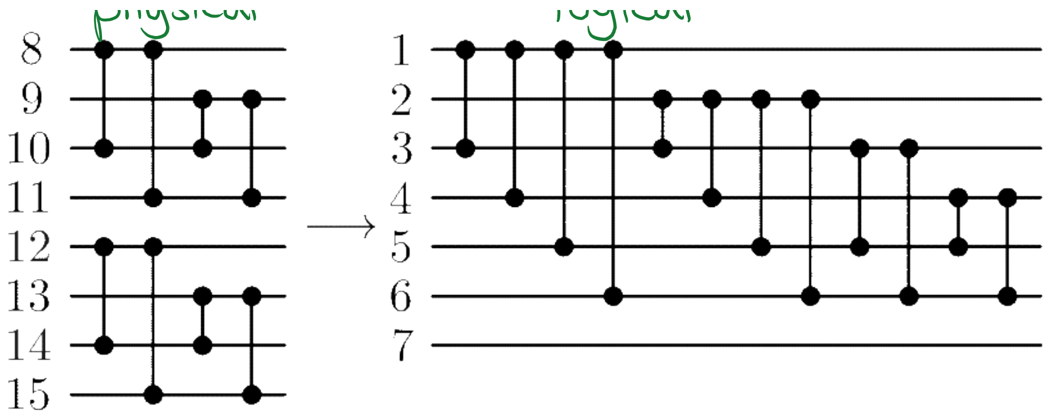
Operate within the block:

physical



logical

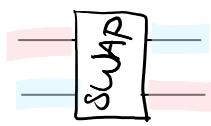




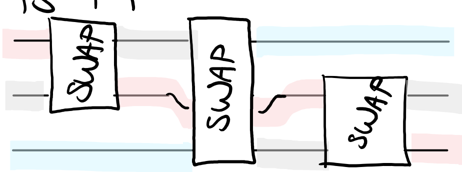
Not fault tolerant!
one gate failure can create wt-2 or 3 error

Trick: Gadgets to catch correlated failures

Gottesman '00:
SWAP is not fault tolerant (in some architectures)

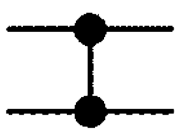


this is FT:



Moral: Extra qubits can avoid correlated errors

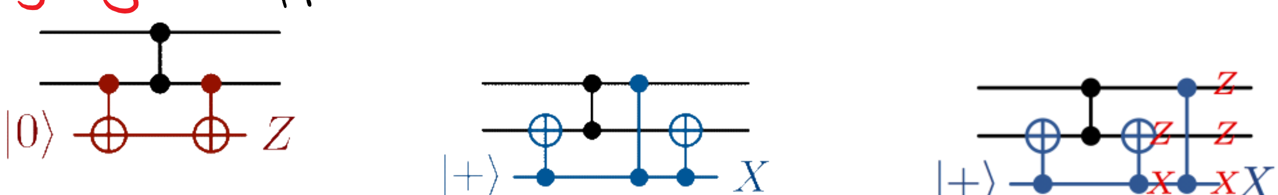
CZ gate gadgets

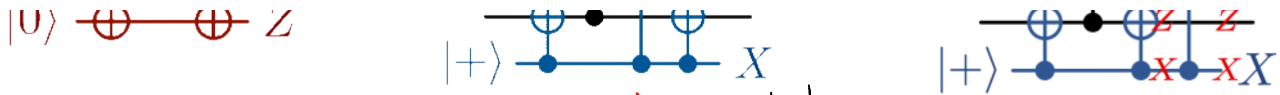


Ways it can fail:
IX, IY, IZ, XI, YI, ZI ← 1-qubit failure after gate
XZ, YZ, ZX, ZY ← before gate

XX, XY, YX, YY, ZZ ← true 2-qubit failure

X gadget: applies CZ, catches XX, XY, YX, YY

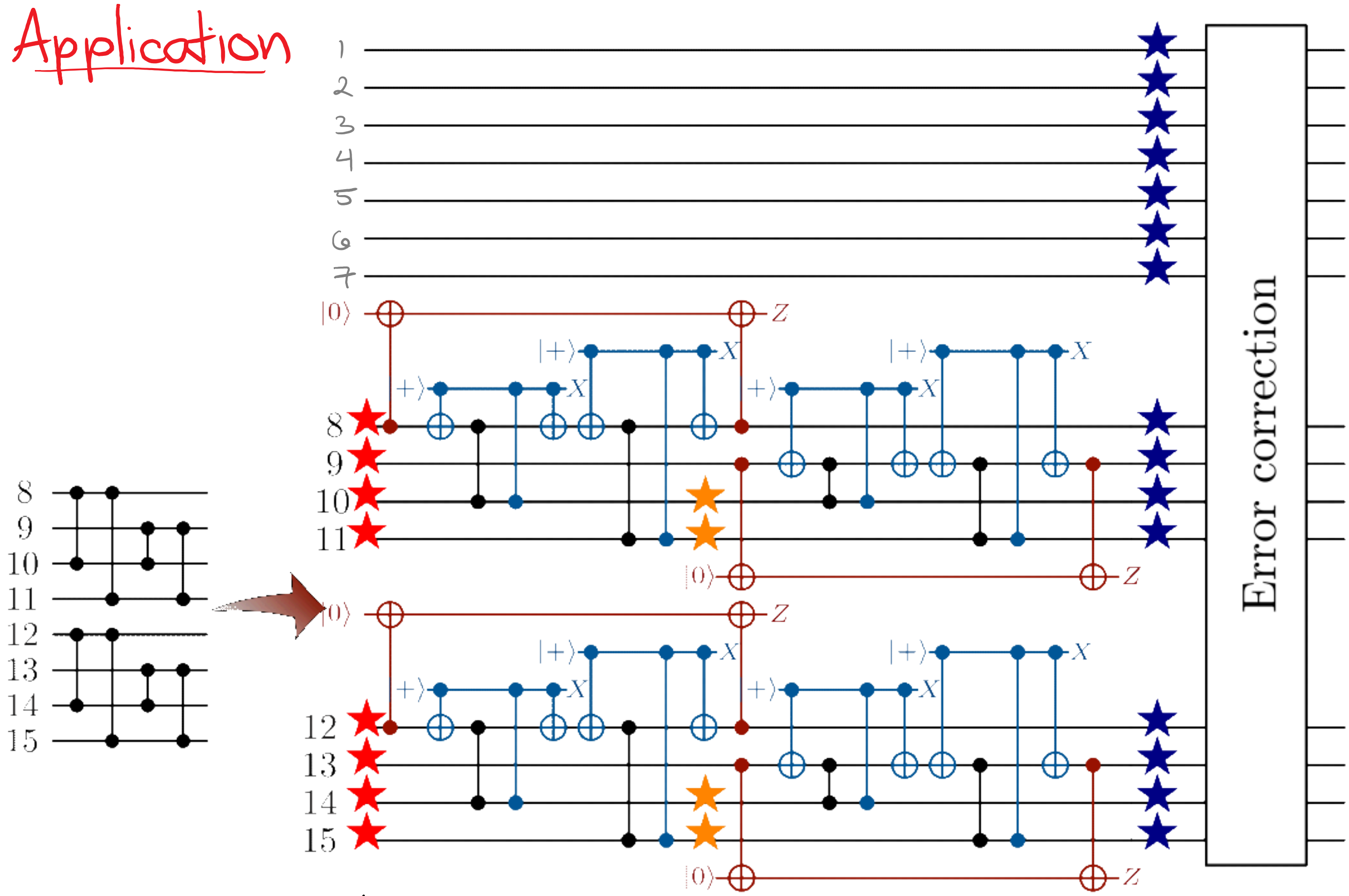




Z gadget: catches ZZ, YX

Combined gadget: catches all true 2-qubit failures

Application



Claim: • 0 failures ⇒ correct effect

• 1 failure, detected by a gadget
⇒ possible errors distinguishable

• 1 failure, not detected (★ ★ ★)
⇒ errors still correctable!

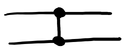
Another trick is needed to get the full Clifford group

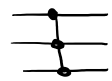
Universal computation with CCZs

① Round-robin CCZs
② CCZ gadget to catch correlated faults

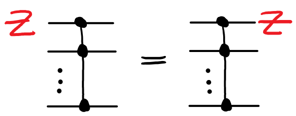
Controlled Z gates

$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$ $\quad CZ = \begin{pmatrix} 1 & & & 0 \\ & 1 & & \\ & & 1 & \\ 0 & & & -1 \end{pmatrix}$ $\quad CCZ = \begin{pmatrix} 1 & & & & & & & 0 \\ & 1 & & & & & & \\ & & 1 & & & & & \\ & & & 1 & & & & \\ & & & & 1 & & & \\ & & & & & 1 & & \\ & & & & & & 1 & \\ 0 & & & & & & & -1 \end{pmatrix}$ $\cdots$

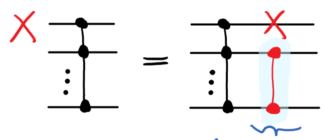




Error propagation



Z errors commute



diagonal ⇒ linear comb. of Zs

X errors copy to Zs

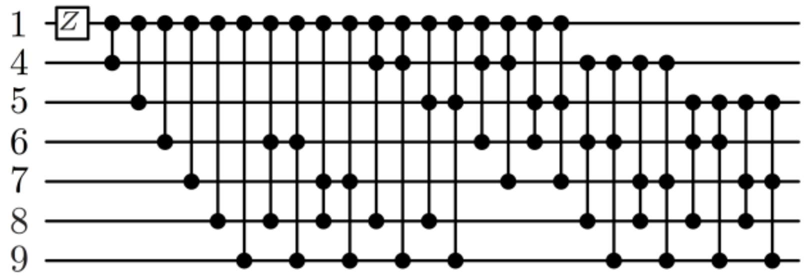
Round-robin CZs

Claim: [Jones, Yoder-Kim]
For any code (CSS or not), if

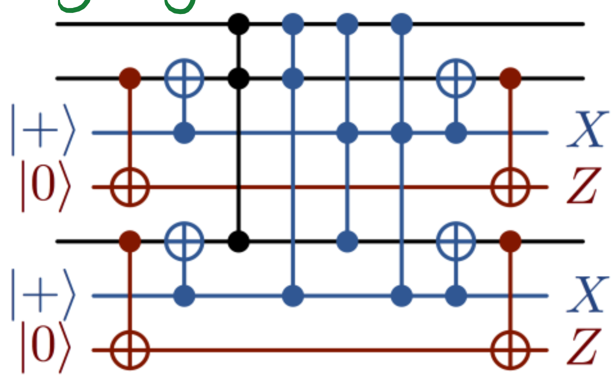
$$\underset{\text{logical}}{\bar{Z}_{I_1}} = \underset{\text{physical}}{Z_{J_1}}, \ldots, \bar{Z}_{I_k} = Z_{J_k}$$

then round-robin $C^{(k-1)}Z$ gates on
$$J_1 \times J_2 \times \cdots \times J_k$$
implements on the codespace logical round-robin $C^{(k-1)}Z$ gates on
$$I_1 \times \cdots \times I_k .$$

round-robin CCZs
$\{1,4,5\} \times \{1,6,7\} \times \{1,8,9\}$



CCZ gadget to catch correlated faults



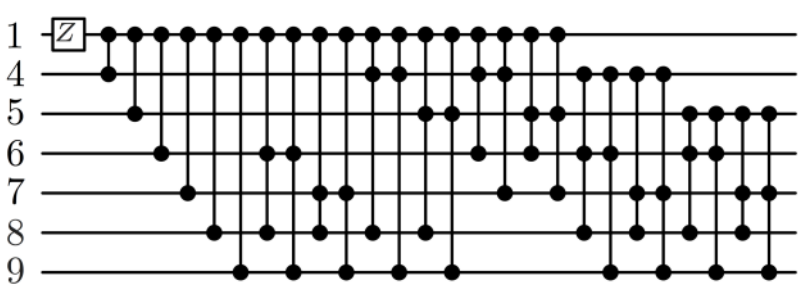
$C^{(k-1)}Z$ gates on codes

Observe: When you apply one or more $C^{(k-1)}Z$ gates to a CSS code, the result might not even be a stabilizer code.

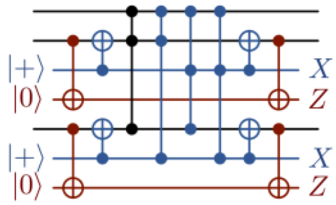
But Z stabilizers are unchanged.

⇒ Can use them to correct X errors.

round-robin CCZs
$\{1,4,5\} \times \{1,6,7\} \times \{1,8,9\}$



1. Use gadget for every CZ or CCZ
2. Correct X errors between gadgets



Analysis

1. A gadget is triggered. If a gadget is triggered, then any Pauli errors can be present on its output data qubits. It is straightforward to check mechanically that for each CZ gate in (15), all four possible X errors, II, IX, XI and XX, have distinct Z syndromes, and so can be corrected immediately in the subsequent X error correction, before the errors can spread. By symmetry, the four possible Z errors have distinct syndromes. These errors commute through (15) and are fixed by the final Z error correction.

Similar considerations hold for each CCZ gate: the possible X and Z error components have distinct syndromes, so an error's X component can be corrected immediately and the Z component corrected at the end.

2. No gadgets are triggered. If there is a single failure in a CZ or CCZ gadget, but the gadget is not triggered, then the error leaving the gadget is a linear combination of the same Paulis that could result from a one-qubit X, Y or Z fault before or after the gadget.

If the error has no X component, then as a weight-one Z error it commutes to the end of (15), at which point Z error correction fixes it.

If the error has X component of weight one, then the Z component can be a permutation of any of III, IIZ, IZZ, ZZZ on the three involved qubits (or of II, IZ, ZZ for a CZ gadget). As we have already argued, these Z errors have distinct X syndromes. The X error correction immediately following the gadget will catch and correct the error's X component, keeping it from spreading. The final Z error correction, alerted to the X failure, will correct the error's Z component.

Summary: More efficient fault-tolerance

Previous methods:

logical qubit

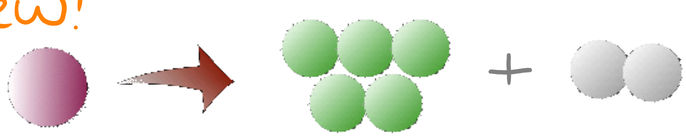
5 physical qubits for the code

+ 5 for error correction [Shor '96]

5 + 4 [DiVincenzo-Aliferis '07]

5 + 3 [Yoder-Kim '16]

New!



7 logical qubits

15 physical qubits for the code

+ 2 for error correction and COMPUTATION (Clifford)

+4 = 19 for universality

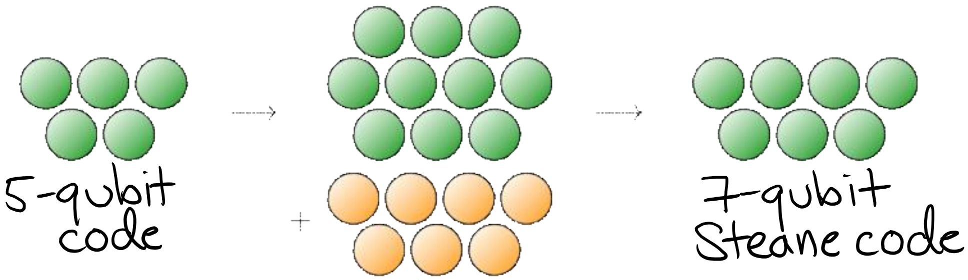
Another application:
Code conversion

Fault-tolerant quantum error correction code conversion

Charles D. Hill, Austin G. Fowler, David S. Wang, Lloyd C. L. Hollenberg

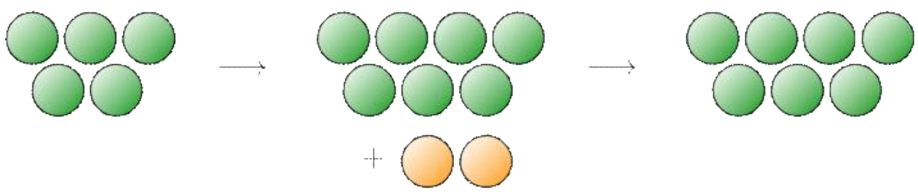
(Submitted on 12 Dec 2011) https://arxiv.org/abs/1112.2417

In this paper we demonstrate how data encoded in a five-qubit quantum error correction code can be converted, fault-tolerantly, into a seven-qubit Steane code. This is achieved by progressing through a series of codes, each of which fault-tolerantly corrects at least one error. Throughout the conversion the encoded qubit remains protected. We found, through computational search, that the method used to convert between codes given in this paper is optimal.



10 qubit codes
+7 for error correction

New: Using CZ gadgets and flagged EC for the intermediate codes



7+2=9 qubits
are enough