QIC 710 Lecture 24 12/2/10 How to build a quantum computer: Fault tolerance & threshold theorems

Resource requirements for Shor's factoring algorithm:

# of bits ($\log_2 N$)	gates	qubits
n	$\lesssim 72n^3$	$\lesssim 5n$
1024	$\sim 2^{36}$	~ 5000

Beckman et al. quant-ph/9602016

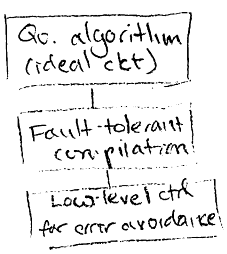
Errors add up, and a single error can destroy the whole computation

$\Rightarrow$ m-gate circuit needs "error rate" $< \frac{1}{m}$.

But for most proposed implementations error rates $< 1\%$ or 0.1% per gate are infeasible $\Rightarrow$ only 100 to 1000 gates possible

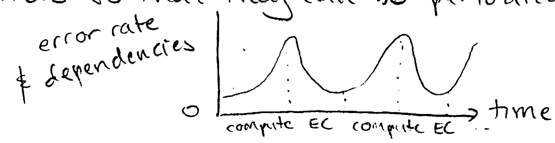
- the exception being topological quantum computation
  - fractional qu. Hall effect [0707.1889]
  - topological insulators [1002.3895, 1003.2856]
  - 1D wires on superconductor... [1006.4395, ...]

What to do?



1. Engineer/optimize physical systems for their particular noise characteristics
   - fabrication, pulse shaping, dynamical decoupling, decoherence-free subspaces, ...
2. But for a scalable system, need higher-level techniques that can handle more generic noise
   - understanding the capabilities & limitations of these techniques can also direct physical design
     - importance of parallel control
     - relative benignness of heralded noise, measurement & preparation noise

Fault tolerance: compute on encoded data, minimizing the spread of errors so that they can be periodically recovered from



Threshold theorems:

1. Fix a particular local noise model
   - e.g., every gate is followed by independent depolarizing noise (randomizing the involved qubits) at rate ε
2. Argue:

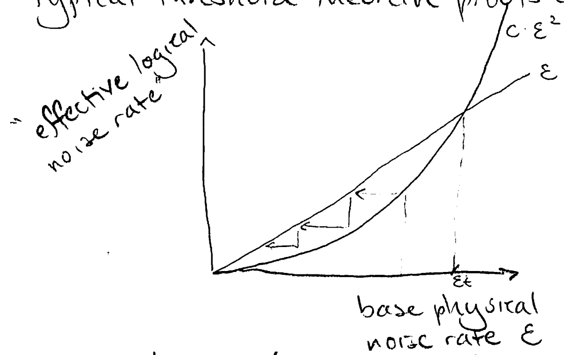
There exists a threshold $\varepsilon_t > 0$ such that if $\varepsilon < \varepsilon_t$ then an ideal quantum circuit with T gates, N qubits, depth D can be simulated by an ε-noisy quantum circuit with

$T \cdot (\log T)^{O(1)}$ gates
$D \cdot (\log T)^{O(1)}$ depth
$N \cdot (\log T)^{O(1)}$ qubits.

In particular, arbitrarily long computations are possible with poly-logarithmic overhead provided the noise is below the threshold.

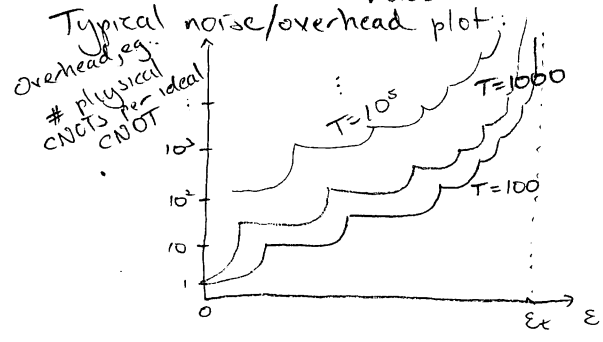
Want high threshold, low overhead (+ other constraints), but these goals conflict.

Typical threshold theorem proofs are by code concatenation



level k	effective ε	overhead
0	ε	1
1	$c\varepsilon^2$	m
2	$c(c\varepsilon^2)^2$	m^2
$\vdots$		
k	$\frac{1}{c}(c\varepsilon)^{2^k}$	m^k

Typical noise/overhead plot:



Typical thresholds (depend on many assumptions):

	proven	estimated
stochastic noise	10^{-4} - 10^{-3}	10^{-3} - 10^{-2}
general noise	10^{-6}	10^{-3} ?

Fault-tolerant computation with Shor's $[\![9,1,3]\!]$ code

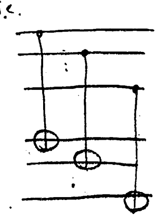
$$|\bar{0}\rangle = (|000\rangle + |111\rangle)(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)$$
$$|\bar{1}\rangle = (|000\rangle - |111\rangle)(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)$$
} not self dual

$X^{\otimes 9} = \bar{Z}$ (or just $X_1 \otimes X_2 \otimes X_3 \otimes \mathbb{1}_{4..9}$)

$Z^{\otimes 9} = \bar{X}$ (or just $Z_1 \otimes Z_4 \otimes Z_7$)

(implement w/ 9 one-qubit Hamiltonians, not one 9-qubit $\mathcal{H}$).

i.e.



(all in parallel)

$\text{CNOT}^{\otimes 9}$: $|\bar{0}\bar{0}\rangle \longrightarrow |\bar{0}\bar{0}\rangle$

$$|\bar{0}\bar{1}\rangle \longrightarrow \left[\text{CNOT}^{\otimes 3}\big((|000\rangle + |111\rangle)(|000\rangle - |111\rangle)\big)\right]^{\otimes 3}$$
$$= \begin{pmatrix} |000,000\rangle - |000,111\rangle \\ + |111,111\rangle - |111,000\rangle \end{pmatrix}^{\otimes 3} = (|000\rangle - |111\rangle)^{\otimes 3} \otimes (|000\rangle - |111\rangle)^{\otimes 3}$$
$$= |\bar{1}\bar{1}\rangle$$
$$|\bar{1}\bar{0}\rangle \longrightarrow |\bar{1}\bar{0}\rangle$$
$$|\bar{1}\bar{1}\rangle \longrightarrow |\bar{0}\bar{1}\rangle$$

$\therefore$ $\text{CNOT}^{\otimes 9}$ implements a logical CNOT in the opposite direction!

[note: Gottesman-Knill stabilizer formalism makes this much easier!]

- although our control is coordinated, it still uses only local gates ("transversal gates", implemented with local Hamiltonians)

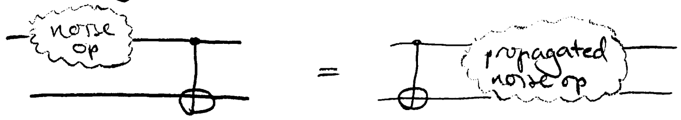
  $\therefore$ control & environmental errors are still local

"fault tolerant encoded operation" = an implementation that does not allow errors to spread within a code block

- still need fault-tolerant preparation, measurement, Toffoli, and <u>error-correction</u> procedures (next time)

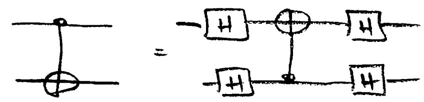
Error propagation

Generically

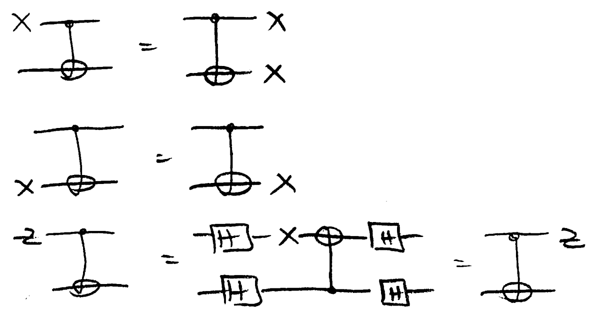


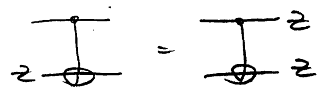
But we can do better...

Recall [hw 2 #2]



Errors in the Pauli expansion are either X, Z or X and Z (Y)

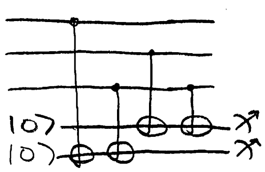




similarly.

• X error correction for Shor's code

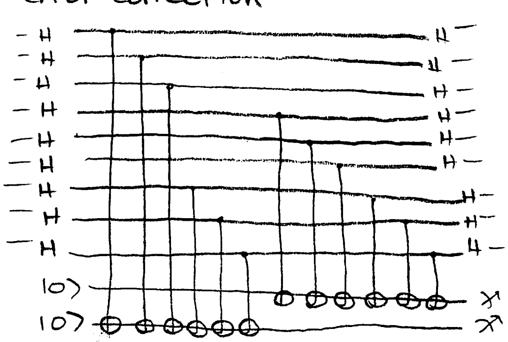
on each block of 3:



↑ apply appropriate correction

this reads off the parities $Z\otimes I\otimes Z$ and $I\otimes Z\otimes Z$, copying X errors down

• Z error correction



any single Z is turned into X by the Hadamards $-Z-H- = -H-X-$

then copied downward to one or both ancillas so its block of 3 can be located

<u>NOT</u> fault tolerant: a Z error on an ancilla can spread backwards, causing <u>multiple</u> X errors on the data.

Alternative method:

1. Observe that transversal X measurements implement a logical Z measurement, & gives any Z errors, while transversal Z meas. implements logical X meas. & gives the X errors.
2. ∴ Prepare encoded $|0\rangle$, apply CNOTs down into it, copying Xs down, Zs up — but with no logical effect! Then measure this ancilla.
3. Of course the previous preparation circuit won't work for preparing $|0\rangle$ fault tolerantly. (A single error might cause you to prepare $|1\rangle$ instead!) So prepare two copies & check one against the other, making sure that no errors are detected before interacting the ancilla with the data.