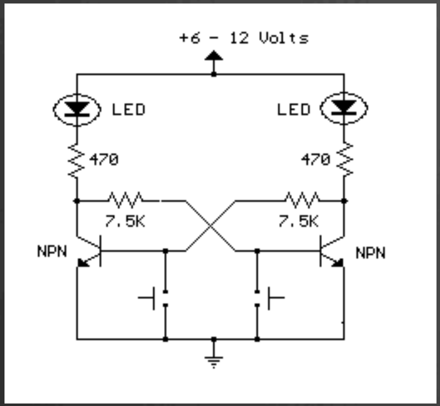
Computing with adversarial noise

Aram Harrow (UW -> MIT)
Matt Hastings (Duke/MSR)
Anup Rao (UW)

The origins of determinism



Theorem [von Neumann]: There exists a constant *p>0* such that for any circuit C there exists a circuit C' such that

- size(C') ≤ size(C) * poly log(size(C))
- If C' is implemented with noise p at every gate, then it will implement C correctly with probability ≥0.99.

Theorem [Shor '95]: Same is true for quantum computers

Assumptions of FTC

1. parallel operations, i.e. $\Omega(n)$ operations per time step. (Assuming that NOP is noisy.)

2. irreversibility / ability to cool bits
   i.e. $\Omega(n)$ fresh '0' bits per time step

3. ~~i.i.d. noise model~~

   decaying correlations:
   e.g. Pr(bits flipped in set S) $\leq$ exp($-c$ $|S|$)

Q: How far can we relax this assumption?

Adversarial noise

- Computation model is deterministic circuits, with constant fan-in and fan-out.
- Number of bits (n) is either static or dynamic.
- In every time step, the adversary chooses $\leq np$ bits to flip.
- Input and outputs are assumed to be encoded in a linear-distance ECC.
- Goals:
  - memory
  - computation

Existing constructions fail

- von Neumann FT (with any code) cannot store $\omega(1/p)$ bits.
- Bits arranged in k-D fails for any constant k.
  More generally, if gates are constrained to be local w.r.t. any easy-to-partition graph, then memory is impossible.

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

Results

Goal	Noise rate	Size overhead
memory	constant	constant
computation	constant	$\exp(o(s))$
computation	$(1/\log(s))^{O(1/\delta)}$	$s^{1+\delta}$
quantum memory	constant	conjectured impossible

The 1->n repetition code

000000000000000000000 or 111111111111111111111

Repeatedly:
Partition into random blocks of size 3 and majority vote

000000000000000000000

001100110111000000100

001 100 110 111 000 000 100

000000111111000000000

Claim:
Starting with an $e \leq 0.1$ fraction of errors, we replace this with $e' \leq 0.9\, e + O(p)$

Computation with the repetition code

If $p \sim 1/k$, then k bits can be encoded.

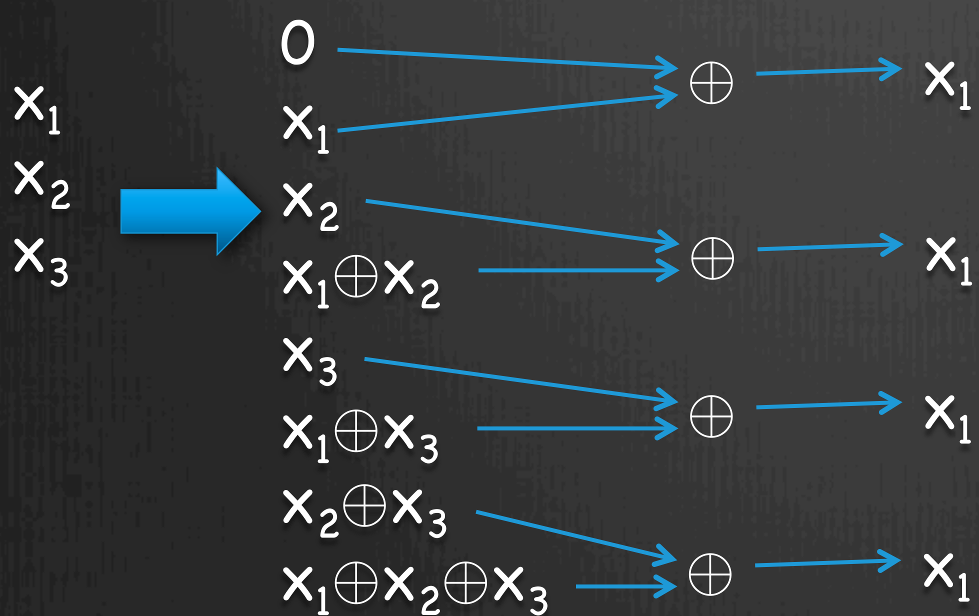
Gates can be performed transversally.
Example:
$(a^n, b^n) \rightarrow (a^n, (a \oplus b)^n)$

The Hadamard code

Definition: k -> $n=2^k$ bits

$$x \in \mathbb{F}_2^k \mapsto (a \cdot x)_{a \in \mathbb{F}_2^k}$$

Example:



Parallel extraction:
Can obtain O(n) copies of any bit.

To replace x_1 with $f(x_2,x_3)$:

1. Extract x_1, x_2, x_3 into repetition codes
2. Compute $x_1 \oplus f(x_2, x_3)$ transversally
3. XOR this with the appropriate locations in the code

Locally correctable codes

Definition:

Given a codeword corrupted in a $\leq \delta$ fraction of positions, there is a randomized method to recover any coordinate of the original codeword, using q queries and giving a wrong answer with probability $\leq \rho$.

Theorem: Any systematic LCC can be used to make a circuit FT against adversarial noise.
Conversely, in any scheme capable of protecting arbitrary circuits against adversarial noise, the input encoding is a LDC.

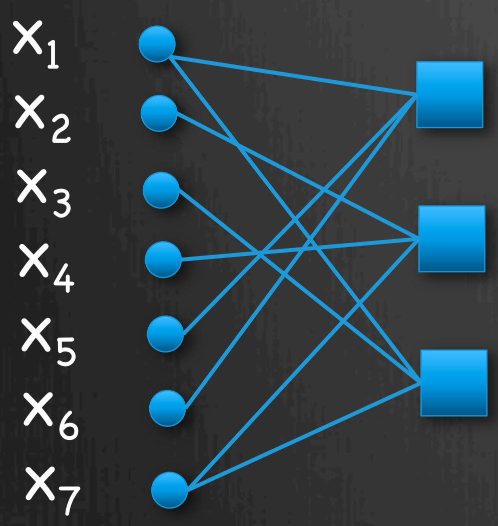
Parameters:

q, δ, ρ constant: k bits into $\exp(o(k))$ bits

$q = \log^{c+1}(k)$, δ, ρ constant: k bits into $k^{1+1/c+o(1)}$ bits

FT memory

<u>LDPC codes</u>



$x_1 \oplus x_5 \oplus x_6 = 0$

$x_2 \oplus x_4 \oplus x_7 = 0$

$x_1 \oplus x_3 \oplus x_7 = 0$

<u>Iterative decoding</u>

Flip variables who have a majority of unsatisfied neighbors.

<u>Key properties</u>

1. Can perform one round of decoding in constant depth.
2. Maps error rate e to $e' \leq 0.9\, e + O(p)$.
3. Good code (i.e. $k/n = \Omega(1)$)

Quantum computing?

Quantum states
n qubits described by a unit vector in $\mathbb{C}^{2^n}$

Immediate difficulties

1. No repetition code:
   - $|\psi\rangle \mapsto |\psi\rangle \otimes |\psi\rangle \otimes |\psi\rangle$ is unphysical
   - no analogue of majority-voting correction
   - cannot correct small errors

2. Parallel extraction impossible / no LDCs

Stabilizer (linear) QECCs

$$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad X = \begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix} \quad iY = \begin{pmatrix} 0 & 1 \\ 0 & -1 \end{pmatrix} \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

<u>n-qubit Pauli matrices</u>

$P_1 \otimes \ldots \otimes P_n$, for $P_1, \ldots, P_n \in \{I, X, iY, Z\}$

<u>stabilizer code</u>

Given commuting Pauli matrices $\{s_1, \ldots, s_{n-k}\}$ define the code space $V = \{|\psi\rangle : s_i|\psi\rangle = |\psi\rangle$ for $i=1,\ldots,n-k\}$

rate = (log dim V)/n = k/n

<u>distance</u>

Let $S = \langle s_1, \ldots, s_{n-k}\rangle$ and $N(S) = \{P : sP=Ps$ for all $s \in S\}$

Distance = min weight of an element of $N(S) \setminus S$

Related quantum open question

- Do there exist stabilizer codes of constant weight and linear distance?
- The only known linear-distance codes have linear-weight generators; the only known constant-weight codes have distance $O(n^{1/2} \log(n))$.
- Homology codes appear unpromising.

Open questions

- QC with adversarial noise?
- Can classical FTC be made more efficient?
- Even against i.i.d. noise, is linear overhead possible?
- Is code deformation of classical codes possible? What if we had a quantum computer?