Tight bounds on computing error-correcting codes by bounded-depth circuits with arbitrary gates

Emanuele Viola
Northeastern University

Anna Gál
Kristoffer Hansen
Michal Koucký
Pavel Pudlák

Spring 2012

Error-correcting codes

- Asymptotically good code over $\{0,1\}$: $C \subseteq \{0,1\}^n$

  rate $\Omega(1)$: $|C| = 2^k$, $k = \Omega(n)$

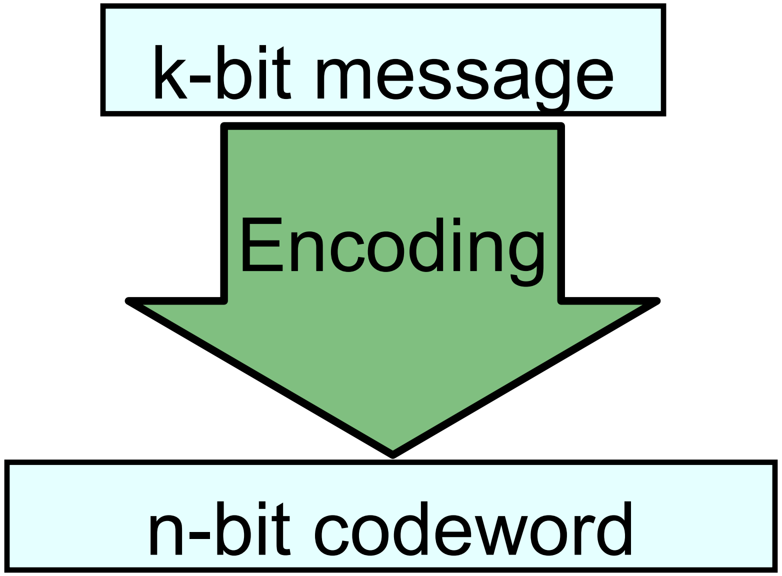
  distance $\Omega(n)$: $\forall\, x \neq y \in C$, x and y differ in $\Omega(n)$ bits

- Useful in communication, combinatorics, hashing, …

- Especially useful if efficiently encodable / decodable

Encoding circuit

- This work:
  complexity of encoding



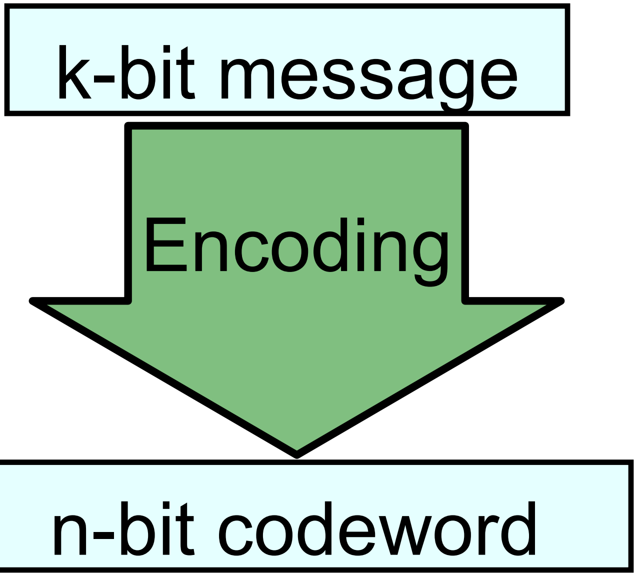
- Since k = Θ(n), measure complexity in terms of n

Previous work

- [Furst Saxe Sipser, …]

  Encoding by AC^0 circuits

  ⇨ size exponential in $n^{\Theta(1)}$



- [Bazzi Mitter 05]

  Encoding by O(n)-time branching programs

  ⇨ space $\Theta(n)$

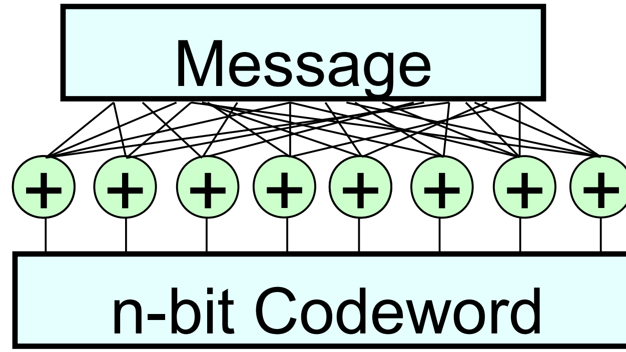
- Rest of this talk: Circuits with arbitrary gates

Previous work

- **Depth 1 Wires $\Theta(n^2)$**

  Unbounded fan-in

  Linear codes

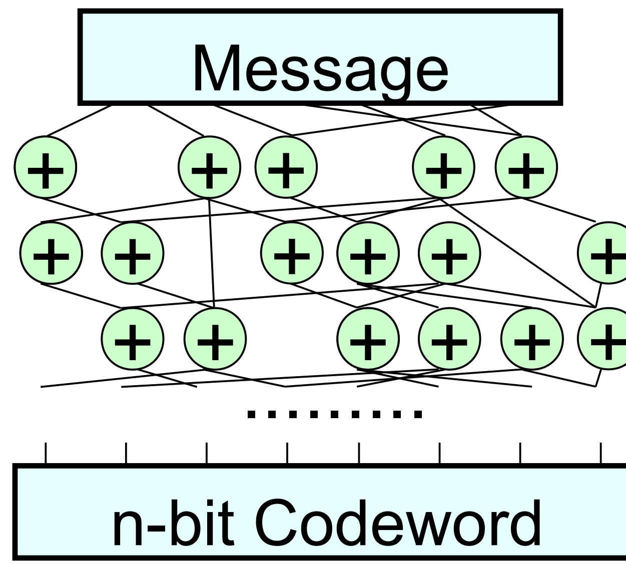


- **Depth $O(\log n)$ Wires $\Theta(n)$**

  Fan-in 2

  [Gelfand Dobrushin Pinsker 73]

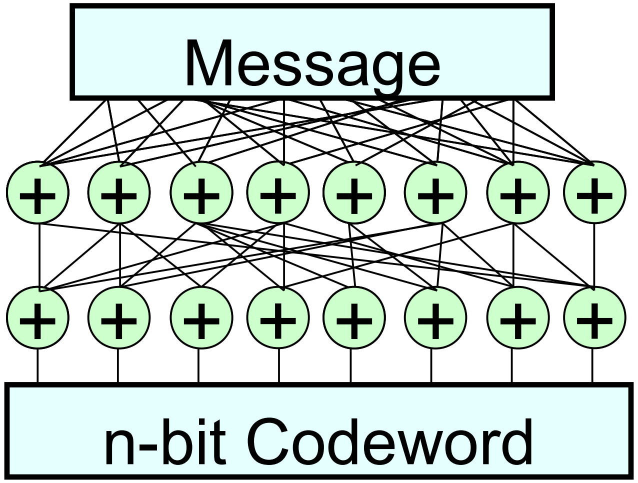
  [Spielman 95]



- Question: How many wires for depth 2?

Our results

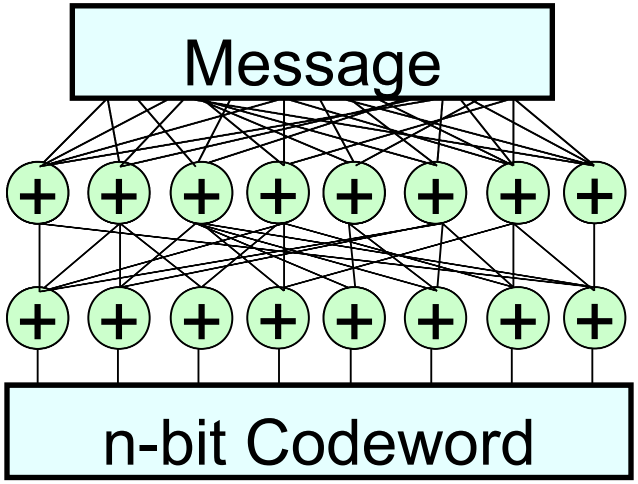
Depth	Wires
2	$n \cdot \Theta\left(\frac{\log n}{\log \log n}\right)^2$
d > 2	$n \cdot \Theta(\lambda_d(n))$



- λ inverse Ackermann: $\lambda_3(n)=\log \log n$, $\lambda_4(n)=\log^* n$, ...
- This talk: Focus on depth 2

Our results, upper bound

Depth	Wires
2	$n \cdot O\left(\frac{\log n}{\log \log n}\right)^2$



- Construction uses XOR gates only

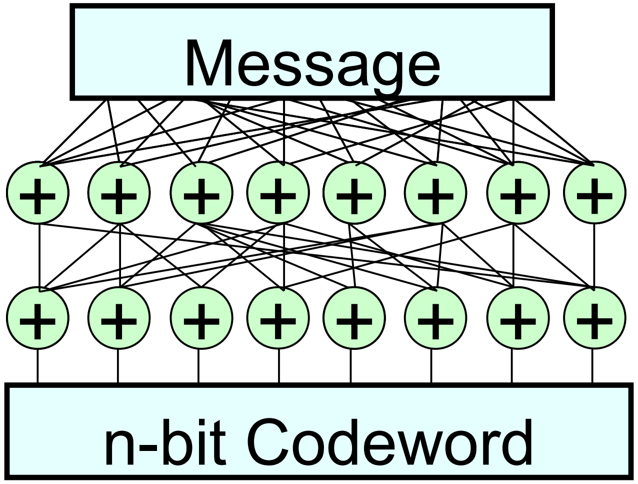
  ⇨ ∃ good code whose (dense) generator matrix

  $M = S_1 S_2$, where $S_1 S_2$ are sparse matrixes

- Not explicit

Our results, lower bound

Depth	Wires
2	$n \cdot \Omega\left(\frac{\log n}{\log \log n}\right)^2$



- ∃ explicit, linear good codes

⇩

Lower bound improves previous depth-2 bounds for explicit linear maps: $\Omega(n \log^{1.5} n)$ [Pudlák Rödl 94]

- Lower bounds hold for any gates

Rest of talk

- Techniques
  - upper bounds
  - lower bounds
- Results for hash functions

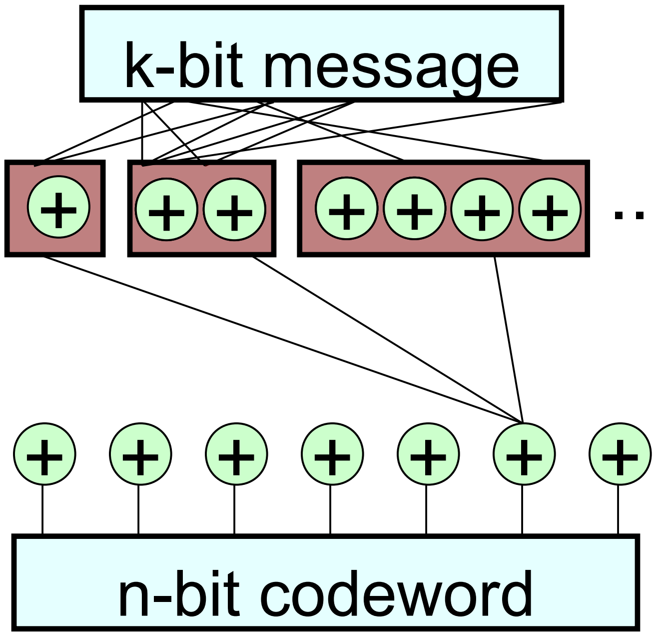
Probabilistic construction

Layer of log n blocks

∀ message ∃ balanced block

Output bit:

XOR one random bit per block



- i-th block balanced for message weight $w = \Theta(n/2^i)$
  Can do with wires $(n/w) \log \binom{n}{w} < n\, i$

- Total wires $= \Sigma_{i < \log n} (n\, i) + n \log n = O(n \log^2 n)$

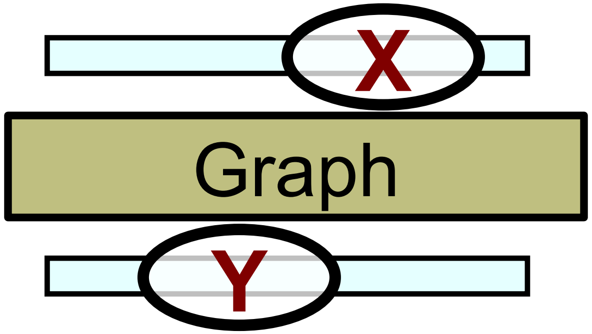
Techniques for lower bounds

- [Spielman]
  Encoding circuit graph reminds super-concentrator

- We revisit connection

- Then adapt super-concentrator lower bounds
  [Valiant] [Pippenger]
  [Dolev Dwork Pippenger Wigderson] [Pudlák]
  [Alon Pudlák] [Radhakrishnan Ta-Shma]

Super-concentrators



Disjoint paths X → Y

- Original super-concentrator: **∀ X, ∀ Y**
  [Valiant]

- Encoding circuit: **∀ X, random Y**
  [This work]

- Relaxed super-concentrator: **random X, random Y**
  [Dolev Dwork Pippenger Wigderson] [Pudlák]

Encoding vs. super-concentrator size

Depth	Original	Encoding	Relaxed
2	$n \cdot \Theta\left(\frac{\log^2 n}{\log \log n}\right)$	$n \cdot \Theta\left(\frac{\log n}{\log \log n}\right)^2$	$n \cdot \Theta(\log n)$
d > 2	$n \cdot \Theta(\lambda_d(n))$	$n \cdot \Theta(\lambda_d(n))$	$n \cdot \Theta(\lambda_d(n))$

- λ inverse Ackermann: $\lambda_3(n) = \log \log n$, $\lambda_4(n) = \log^* n$, ...
- Same size for every depth, except 2

Hash functions

- Goal: Compute hash $f : \{0,1\}^n \times \{0,1\}^{O(n)} \to \{0,1\}^n$

  $\forall x \neq y$, $(f(x,U), f(y,U))$ uniform

- We obtain similar results for hashing as for encoding, with factor-2 depth loss in upper bounds

- Depth-d encoding ⇨ depth-2d hashing

  [Ishai Kushilevitz Ostrovsky Sahai 08]

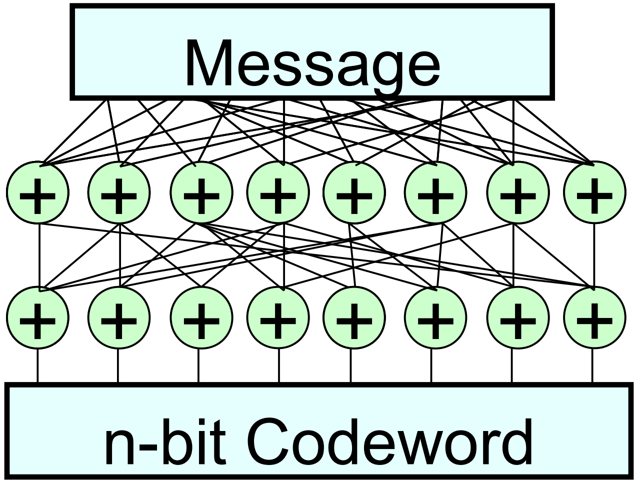
- Depth-d encoding ⇦ depth-d hashing

  [Miltersen 98]

Summary

- Complexity of circuit encoding message in good code

Depth	Wires
2	$n \cdot \Theta\left(\frac{\log n}{\log \log n}\right)^2$
d > 2	$n \cdot \Theta(\lambda_d(n))$



- Similar bounds for hash functions
- Revisit encoding circuit vs. super-concentrators
- Open: Explicit, tight depth of hashing, decoding