Optimized Graph-Based Codes For Modern Flash Memories

Homa Esfahanizadeh
Joint work with Ahmed Hareedy and Lara Dolecek
LORIS Lab
Electrical Engineering Department, UCLA

10/08/2016

Presentation Outline

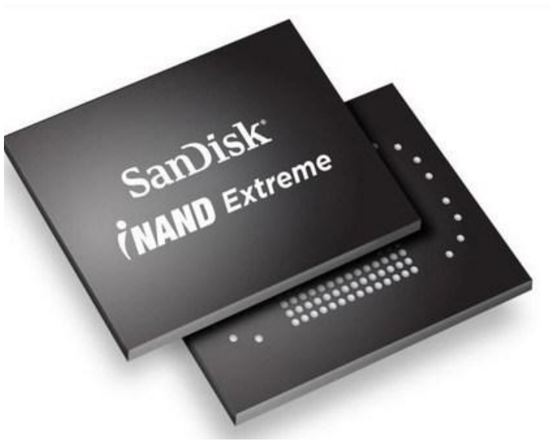
- **Background and motivation**
- **Non-binary LDPC code optimization for flash memory**
  - New combinatorial objects
  - The optimization framework
  - Simulation results for practical Flash
- **Analysis and design of spatially-coupled LDPC codes**
  - Block vs spatially-coupled LDPC codes
  - The optimization framework
  - Simulation results for AWGN channel
- **Conclusions and future work**

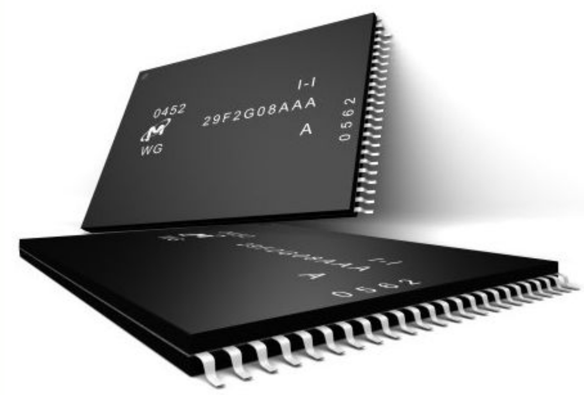
Presentation Outline

- **Background and motivation**
- **Non-binary LDPC code optimization for flash memory**
  - New combinatorial objects
  - The optimization framework
  - Simulation results for practical Flash
- **Analysis and design of spatially-coupled LDPC codes**
  - Block vs spatially-coupled LDPC codes
  - The optimization framework
  - Simulation results for AWGN channel
- **Conclusions and future work**

Graph-Based Codes with Low Error Rate

- **Modern flash memories operate at very low frame error rates. That is why strong ECC schemes are needed.**





- **Low-density parity-check (LDPC) codes are graph-based codes that have capacity approaching performance.**

❖ **Non-binary LDPC (NB-LDPC) codes**
❖ **Spatially-coupled (SC) LDPC codes**

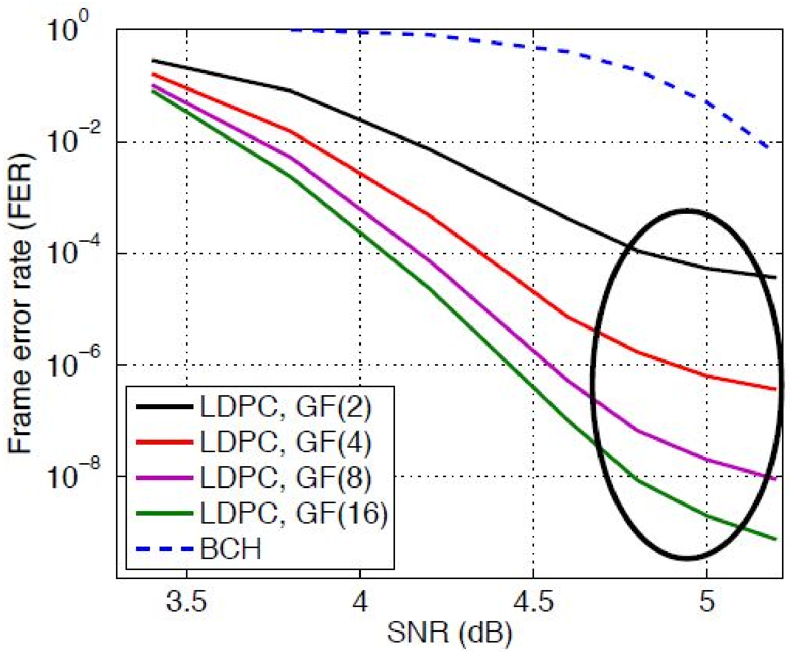
Non-Binary vs Binary LDPC Codes

- **Why non-binary?**
  - Grouping bits into symbols over GF(q) decreases the probability of decoding failure.
  - Increasing the Galois Field size q results in better performance.

Block length ≈ 1000 bits

Rate ≈ 0.9

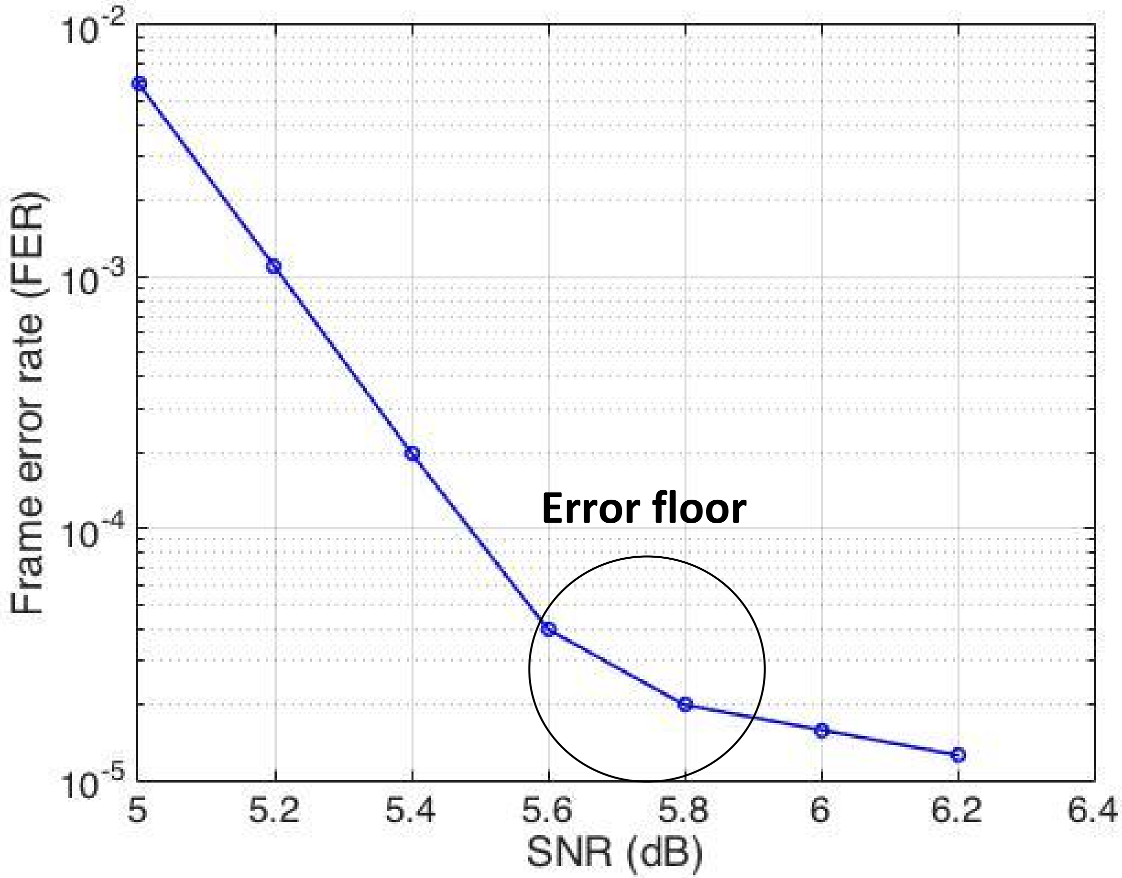
Column weight = 4



- **Disadvantage: Decoding complexity increases.**

Error Floor of LDPC Codes: Absorbing Sets

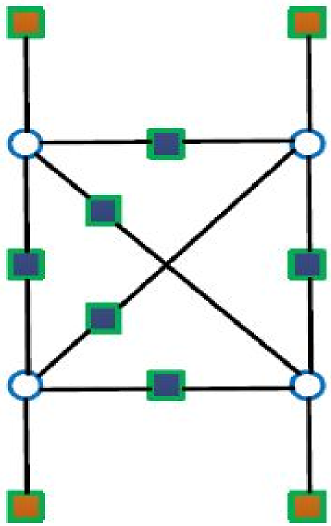
- **For the AWGN channels, absorbing sets (ASs) [Dolecek 10] are the reason behind error floor.**



Error Floor of LDPC Codes: Absorbing Sets

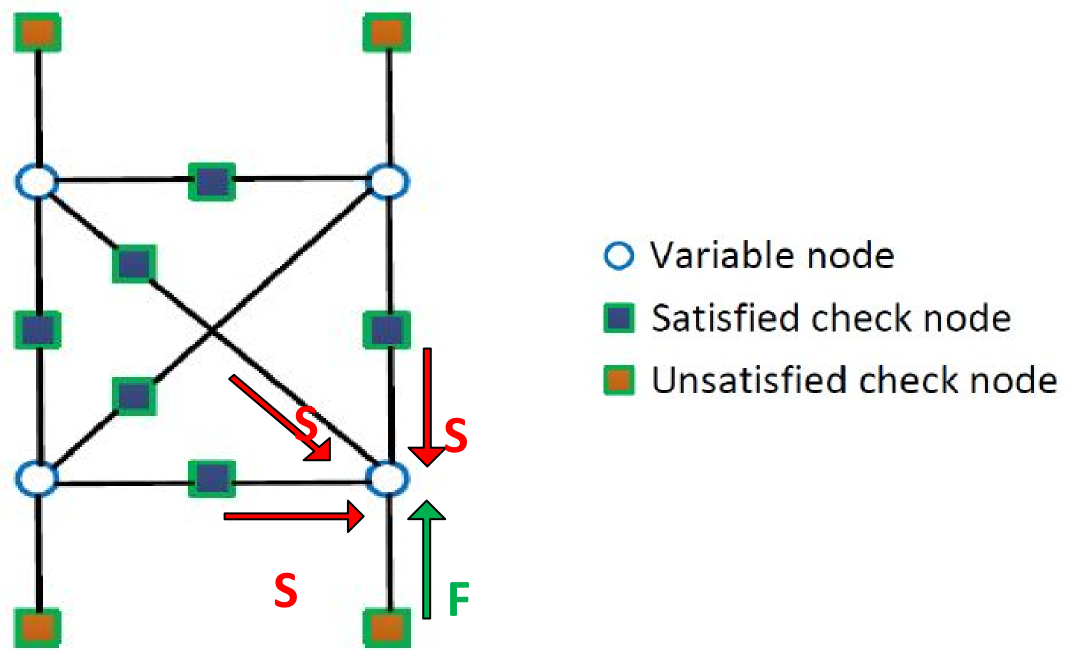
- **An (*a*, *b*) absorbing set:**
  - is a subgraph of the Tanner graph.
  - *a* is the number of variable nodes in the configuration.
  - *b* is the number of unsatisfied check nodes connected to the configuration.
  - each variable node is connected to more satisfied than unsatisfied check nodes.

- **Example: (*4*, *4*) absorbing set:**



Why Absorbing Sets Are Problematic?

– Consider (4, 4) binary absorbing set,
– When errors happen on four variable nodes, each variable node receives more satisfied messages than unsatisfied ones!

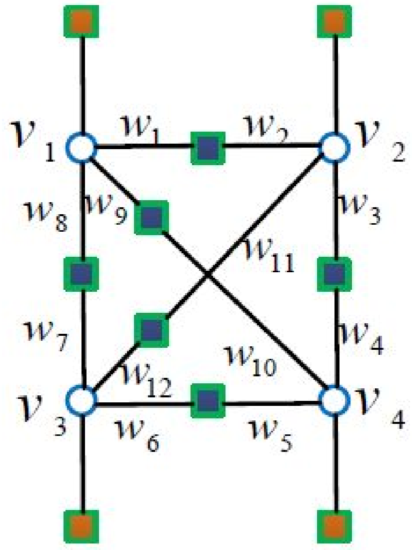


Binary vs. Non-Binary Absorbing Sets

- **Binary AS --> Binary LDPC codes**
- **Non-binary AS --> Non-binary LDPC codes**

- **Binary absorbing sets are described in terms of topological conditions only. For non-binary absorbing sets,**
  - The values of the variable nodes matter.
  - The topological conditions alone are not enough; weight conditions have to be added [Amiri 14].

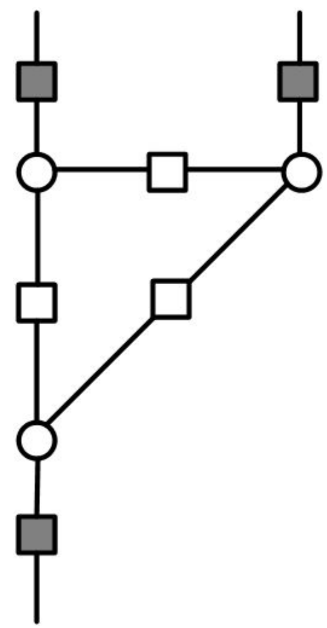
$$\prod_{i=1}^{t} w_{2i-1} = \prod_{i=1}^{t} w_{2i} \text{ over GF}(q)$$



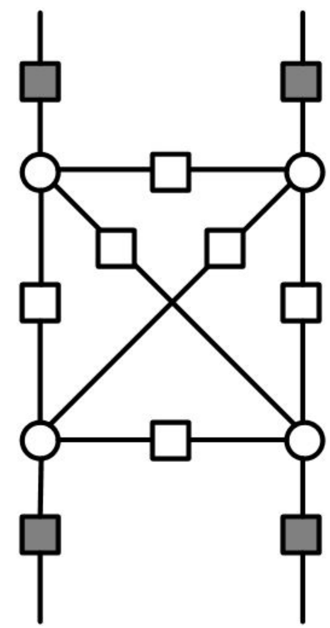
Objects of Interest for the AWGN Channel

- **For symmetric channels (like AWGN), the dominant objects have been the elementary ASs.**
- **Elementary AS: Each satisfied check node is of degree 2 and each unsatisfied check node is of degree 1.**

- **Examples:**
  - (3, 3) AS, γ=3.
  - (4, 4) AS, γ=4.
  - Where γ is the column weight of the code.



(3, 3) AS



(4, 4) AS

Presentation Outline

- **Background and motivation**

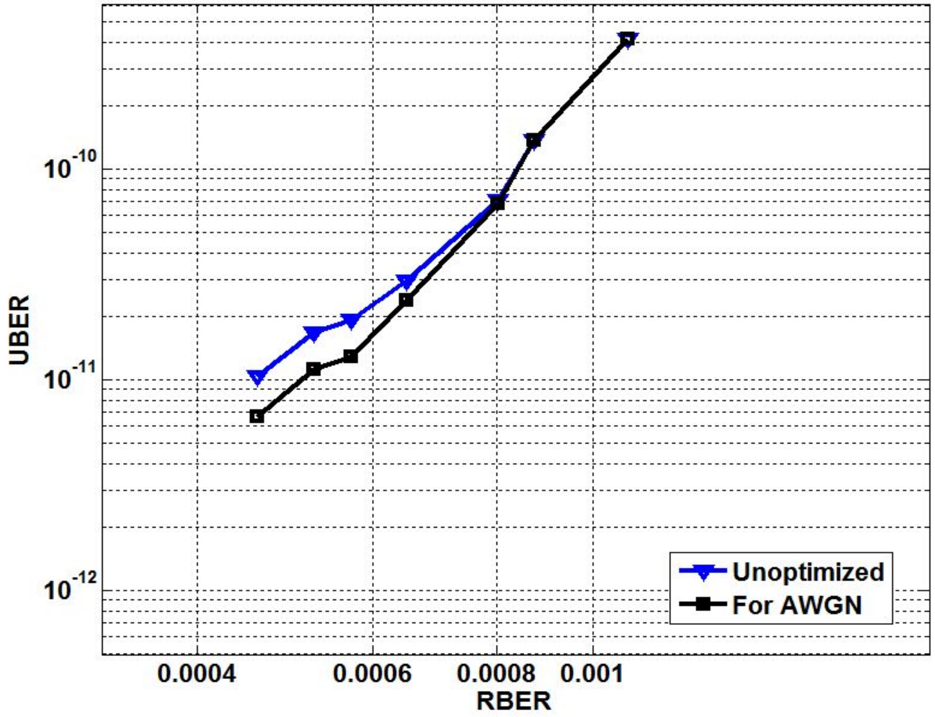
- **Non-binary LDPC code optimization for flash memory**
  - New combinatorial objects
  - The optimization framework
  - Simulation results for practical Flash

- **Analysis and design of spatially-coupled LDPC codes**
  - Block vs spatially-coupled LDPC codes
  - The optimization framework
  - Simulation results for AWGN channel

- **Conclusions and future work**

AWGN Techniques Don't Work for Flash Memories

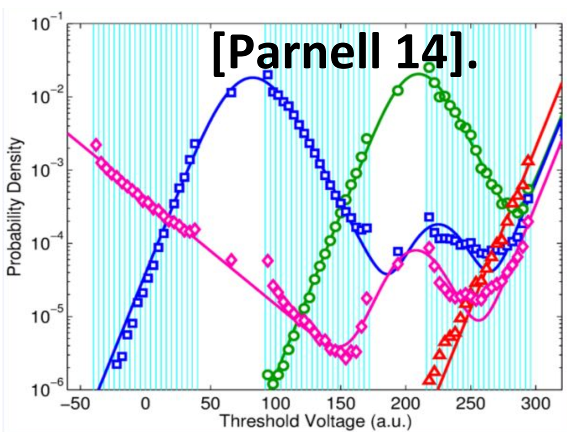
- **Using optimization with respect to *elementary objects*, this is the gain we can reach on an asymmetric Flash channel (NLM Flash channel [Parnell 14]):**

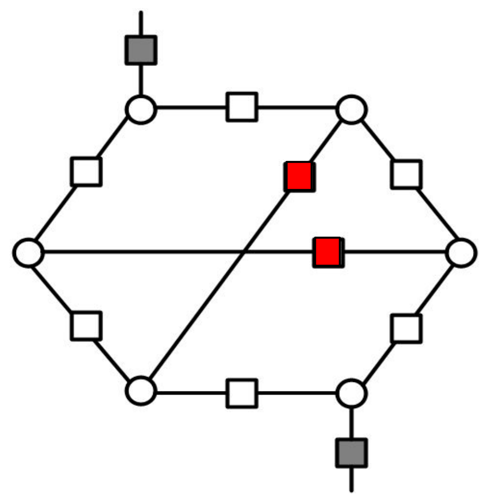


- **Can we do better than techniques that are developed for the AWGN channel?**

The Answer is Yes!

- **Asymmetry in the channel (e.g., in Flash) can result in:**
  - NB ASs with unsatisfied check nodes having degree > 1.
  - NB ASs with satisfied check nodes having degree > 2.
- **Such dominant objects are non-elementary!**
- **This is mainly because of the high VN error magnitudes.**
- **Example: (6, 4) non-elementary NB AS (γ=3).**





- **(6, 2), (6, 3), and (6, 4) are all problematic because of asymmetry.**

We Need New Definitions/Objects

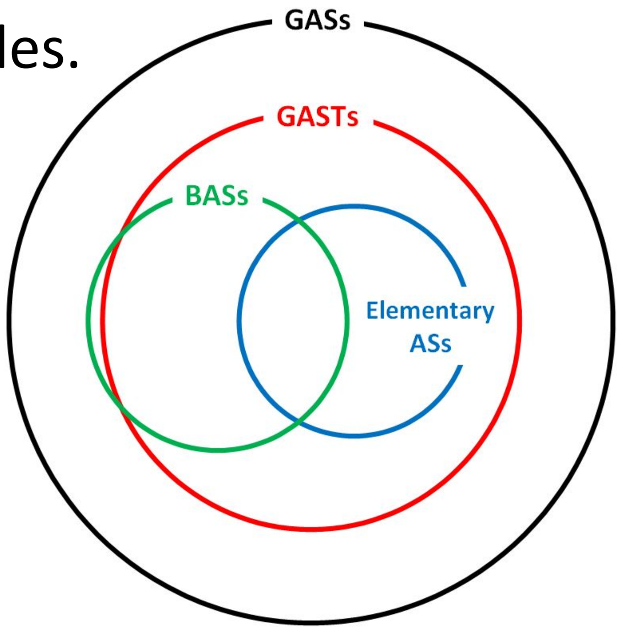
- **General absorbing set (GAS)**
  - We can have unsatisfied check nodes of degree > 1.
  - We can have satisfied check nodes of degree > 2.

- **Because unsatisfied check nodes of degree > 2 are less likely to happen, we introduce GAST.**

- **A GAS of type two (GAST) is a GAS that:**
  - The number of degree 2 check nodes is higher than the number of degree > 2 check nodes.
  - Unsatisfied check nodes are of degree 1 or degree 2.

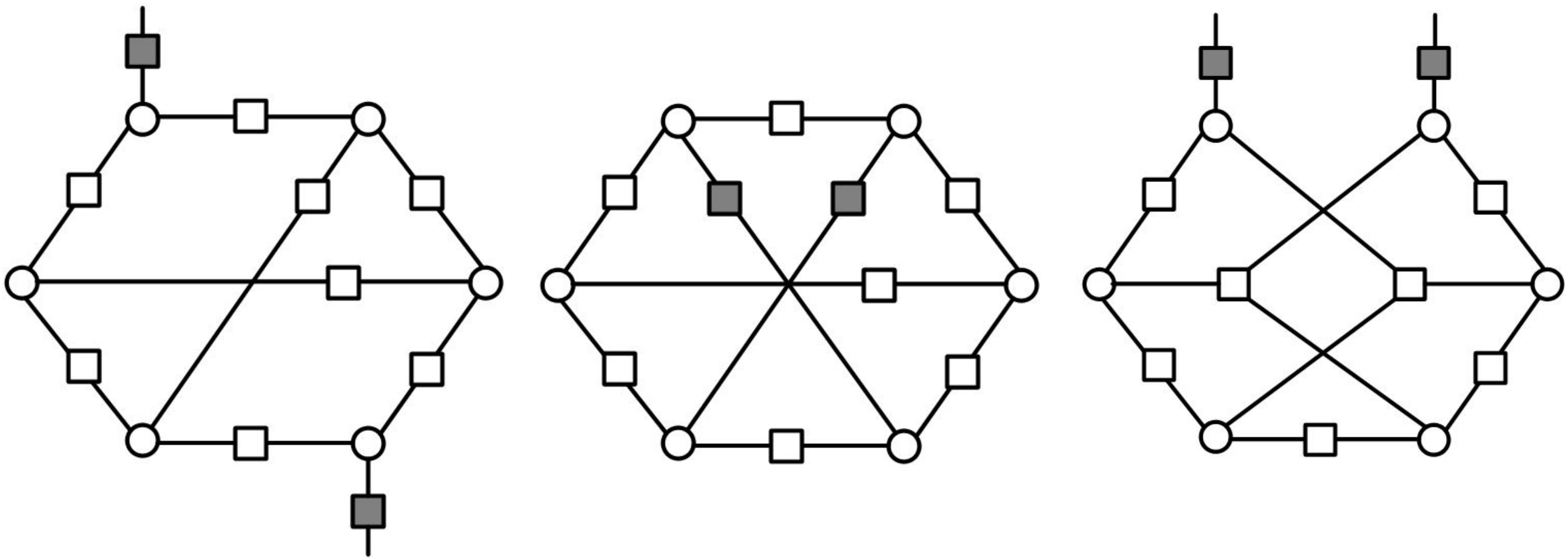
We Need New Definitions/Objects

- **Topological description of the new objects (GASTs):**
  The tuple (a, b, d_1, d_2, d_3) is an NB AS that:
  - a is the number of variable nodes in the set.
  - b is the number of unsatisfied check nodes.
  - d_1 is the number of degree 1 check nodes.
  - d_2 is the number of degree 2 check nodes.
  - d_3 is the number of degree > 2 check nodes.



- **GASTs are more general than any previously introduced type of absorbing sets**

Examples of GASTs



Config (a) Config (b) Config (c)

- Configuration (a) is now a **(6, 2, 2, 8, 0) GAST**.
- Configuration (b) is now a **(6, 2, 0, 9, 0) GAST**.
- Configuration (c) is now a **(6, 2, 2, 5, 2) GAST**.

How to Remove a GAST

- **Objective of removal:**
  - The code structure and properties are preserved.
  - Manipulating the edge weights such that problematic GASTs are completely removed (not converted into other GASTs).

- **We established a set of necessary algebraic conditions such that when we break one of them, the GAST is removed.**

  Our WCM framework [Hareedy 16]

Algorithm: NB-LDPC Code Optimization

- **Input: Tanner graph. Output: Optimized Tanner graph.**

1. Identify the set (*G*) of problematic GASTs.
2. For each candidate, extract its subgraph from the Tanner graph of the code.
3. Determine the set of necessary conditions of that GAST.
4. For each condition in that set:

   Break the condition via the edge weight manipulation.
5. If the GAST removal is successful, reflect the edge weight changes in the Tanner graph of the code.
6. This process continues until all GASTs in *G* are removed or no more GASTs can be removed.

Applications of Our Framework

- **The normal-Laplace mixture (NLM) Flash channel [Parnell 14]:**
  – Accurately models the voltage threshold distribution of sub-20nm MLC (2-bit) NAND Flash memory.
  – Takes into account various sources of error due to wear-out effects (e.g., programming errors).
  – We are using 3 reads (hard decision).

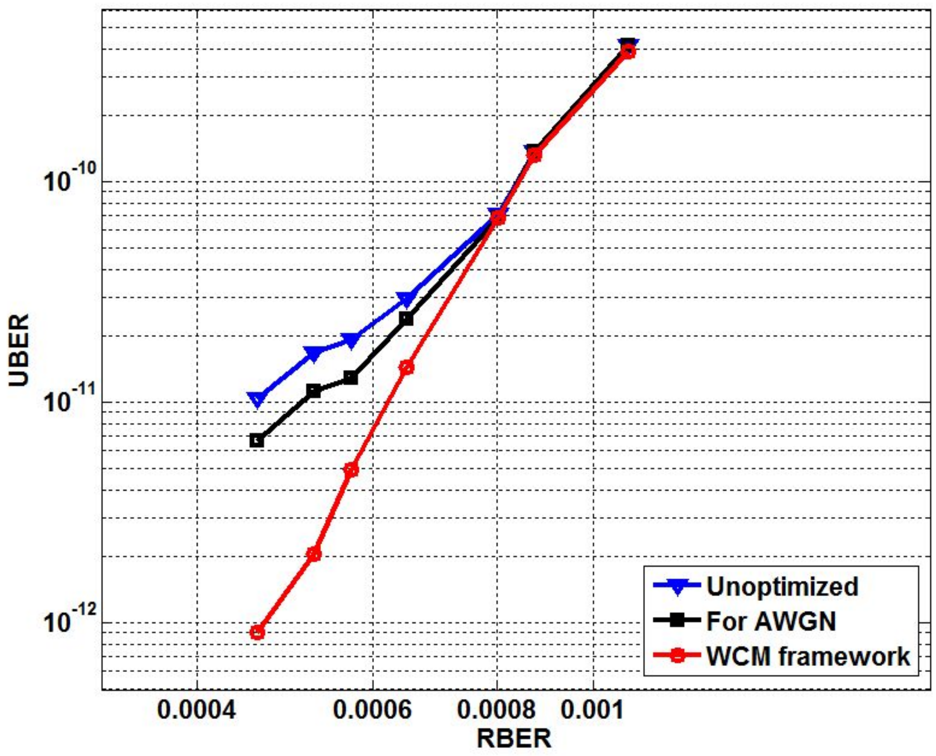
- **Moreover, we test our framework on Cai Flash channel [Cai 13].**

- **Unoptimized codes are designed according to [Bazarsky 13].**

NB-QC-LDPC Codes with Column Weight 3

- **NB-QC-LDPC code: Block-length = 3996 bits, GF(4), rate ≈ 0.89.**

Type	Unopt	For AWGN	WCM Framework
(4, 2, 2, 5, 0)	45	0	0
(4, 3, 2, 5, 0)	15	23	0
(4, 4, 4, 4, 0)	3	0	0
(6, 0, 0, 9, 0)	12	1	0
(6, 1, 0, 9, 0)	7	13	0
(6, 2, 0, 9, 0)	3	5	1
(6, 2, 2, 5, 2)	2	2	0
(7, 1, 0, 10, 1)	4	4	0
Other	9	13	8

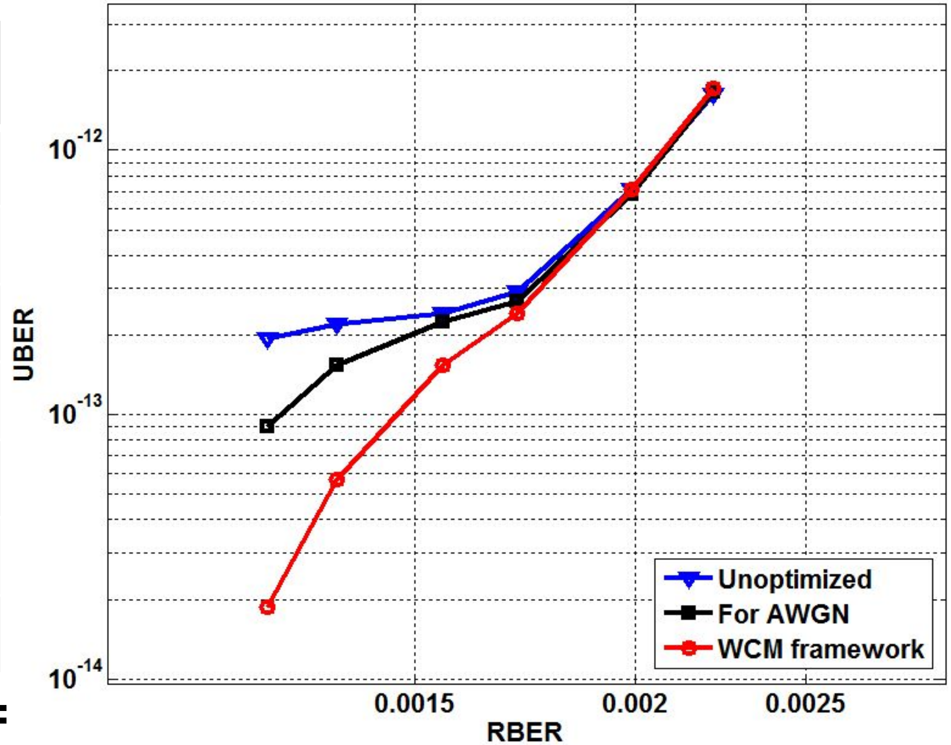


- **Tables are extracted at RBER = 4.60e-4**
  - UBER (unoptimized) = 1.04e-11, UBER (optimized) = 9.04e-13.

NB-QC-LDPC Codes with Column Weight 4

- **NB-QC-LDPC code: Block-length = 3280 bits, GF(4), rate ≈ 0.80.**

Type	Unopt	For AWGN	WCM Framework
(4, 4, 4, 6, 0)	47	2	0
(6, 2, 2, 11, 0)	11	0	0
(6, 4, 2, 11, 0)	14	14	0
(6, 4, 4, 7, 2)	6	6	0
(8, 3, 2, 15, 0)	6	6	1
(8, 4, 4, 14, 0)	5	1	0
Other	11	18	9



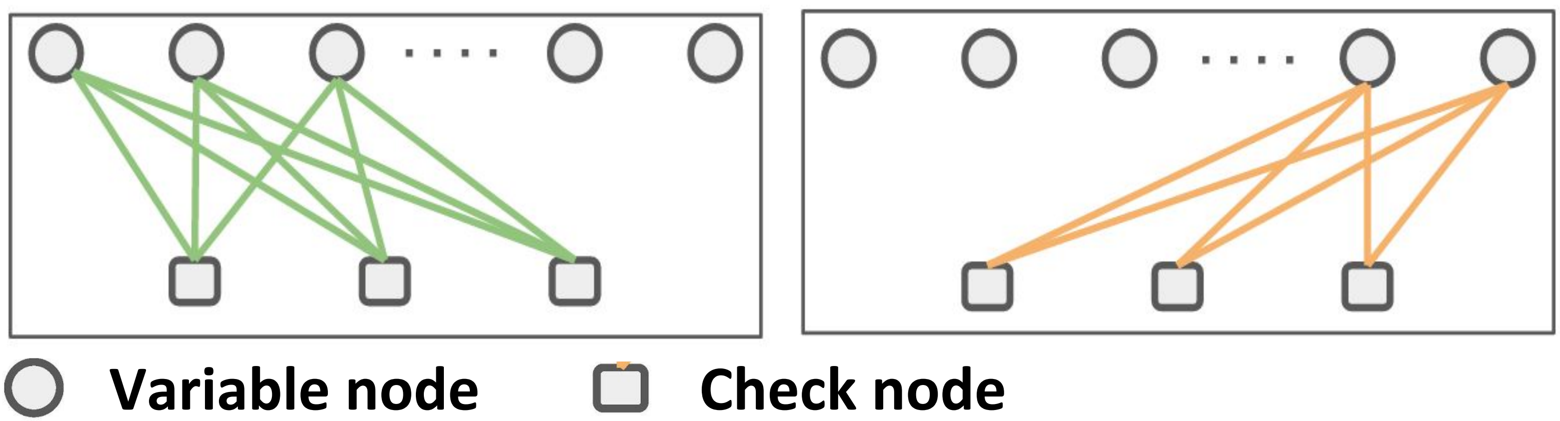
- **Tables are extracted at RBER :**
  – UBER (unoptimized) = 1.93e-13, UBER (optimized) = 1.86e-14.
- **Optimizing using WCM gives > 1 order of magnitude gain.**
- **Optimizing for AWGN (elementary) does not help.**

Presentation Outline

- **Background and motivation**
- **Non-binary LDPC code optimization for flash memory**
  - New combinatorial objects
  - The optimization framework
  - Simulation results for practical Flash
- **Analysis and design of spatially-coupled LDPC codes**
  - Block vs spatially-coupled LDPC codes
  - The optimization framework
  - Simulation results for AWGN channel
- **Conclusions and future work**

SC Code Construction: Partitioning and Concatenation

- **A spatially-coupled code is a chain of coupled block LDPC codes.**
- **A cutting vector partitions the parity-check matrix of the underlying block code to two sub-matrices**



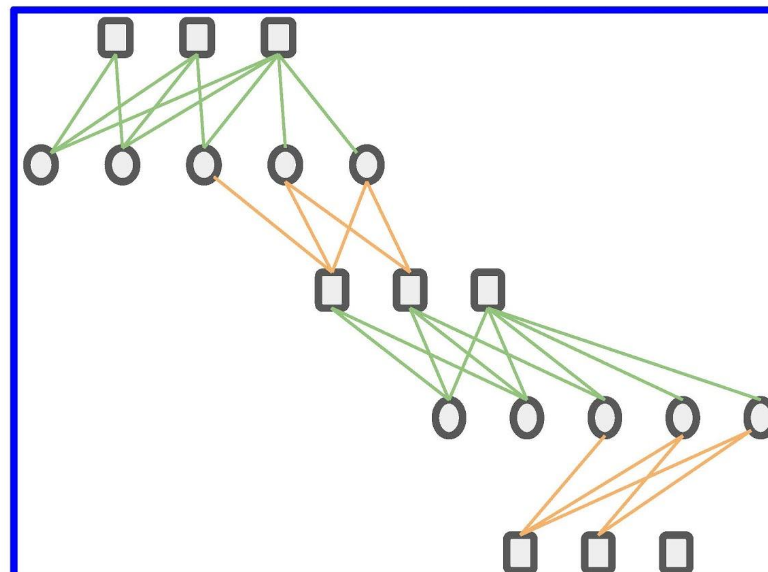
SC Code Construction: Partitioning and Concatenation

- **An spatially-coupled code is formed by coupling replicas of the partitioned sub-matrices together.**

$H_{SC}=$

H0	0	0	0	0	0
H1	H0	0	0	0	0
0	H1	H0	0	0	0
0	0	H1	⋮	0	0
0	0	0		H0	0
0	0	0	0	H1	H0
0	0	0	0	0	H1

Parity-check matrix of an spatially-coupled (SC) code



Tanner graph corresponding to the sub-matrix with blue borders

Importance of Finite length Analysis of SC Codes

- **Shown to have excellent performance in the regime of extremely long block lengths when averaged over many codes.**

- **Many recent papers are on this asymptotic limit:**

- **[Costello 14] [Urbanke 13] [Lentmaier 15], among others.**

- **Our research:**
  Finite-length performance of spatially-coupled codes

SC-LDPC Code Optimization

- **We derived a compact mathematical technique for the enumeration of problematic objects of spatially-coupled codes.**
- **This technique uses the special structure of SC codes to simplify the enumeration of detrimental objects.**
- **We proposed an optimization framework to find the optimal cutting vector that results minimum number of dominant absorbing sets [Amiri 14].**

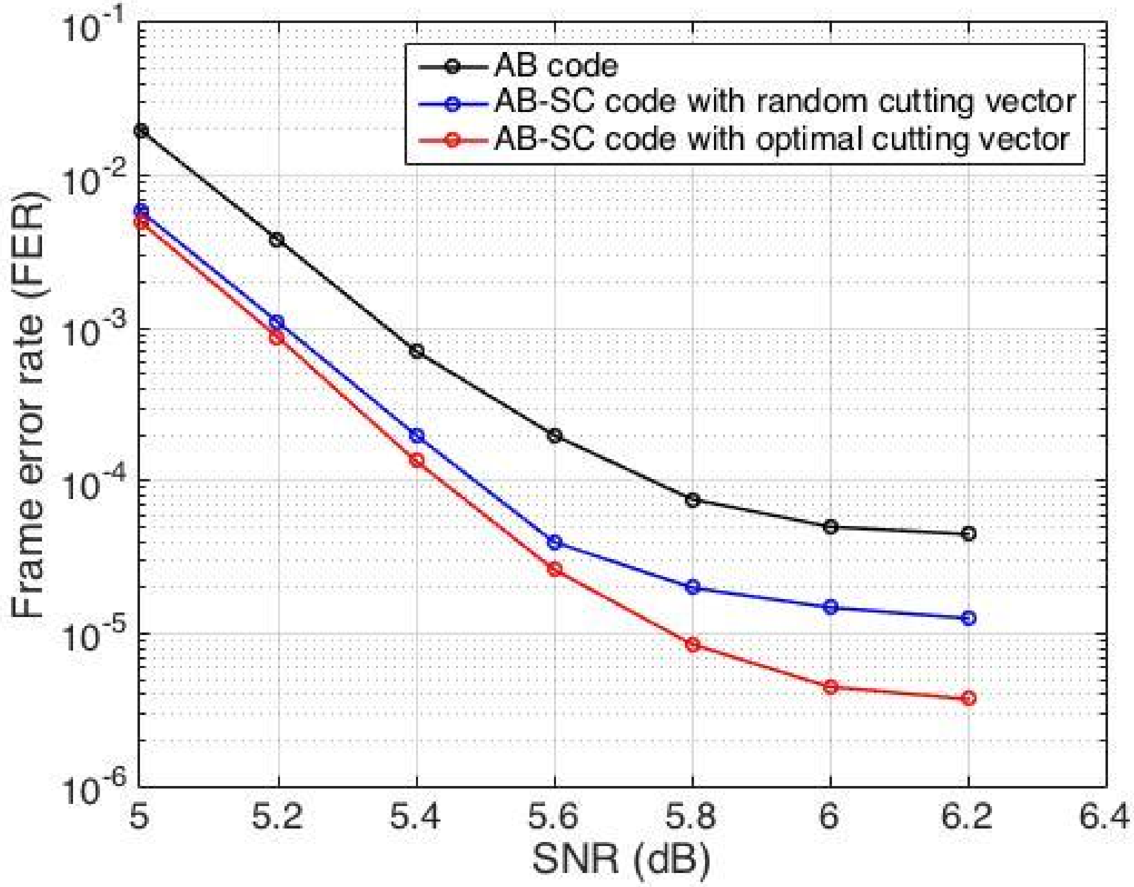
The Best and Worst Cutting Vector Analysis

Circulant size	Coupling length	The best cutting vector	The worst cutting vector
17	50	[4,8,13] (Equi-partition) number of (3,3) = 99144	[17,17,17] (No coupling) number of (3,3) = 231200
23	100	[5,11,17] (Equi-partition) number of (3,3) = 512302	[23,23,23] (No coupling) number of (3,3) = 1163800
31	100	[7,15,23] (Equi-partition) number of (3,3) = 1288608	[31,31,31] (No coupling) number of (3,3) = 2883000

- "Equi-partition" minimizes the number of (3,3) ASs.
- Due to partitioning the block code, SC codes have fewer detrimental ASs than block-based counterparts.

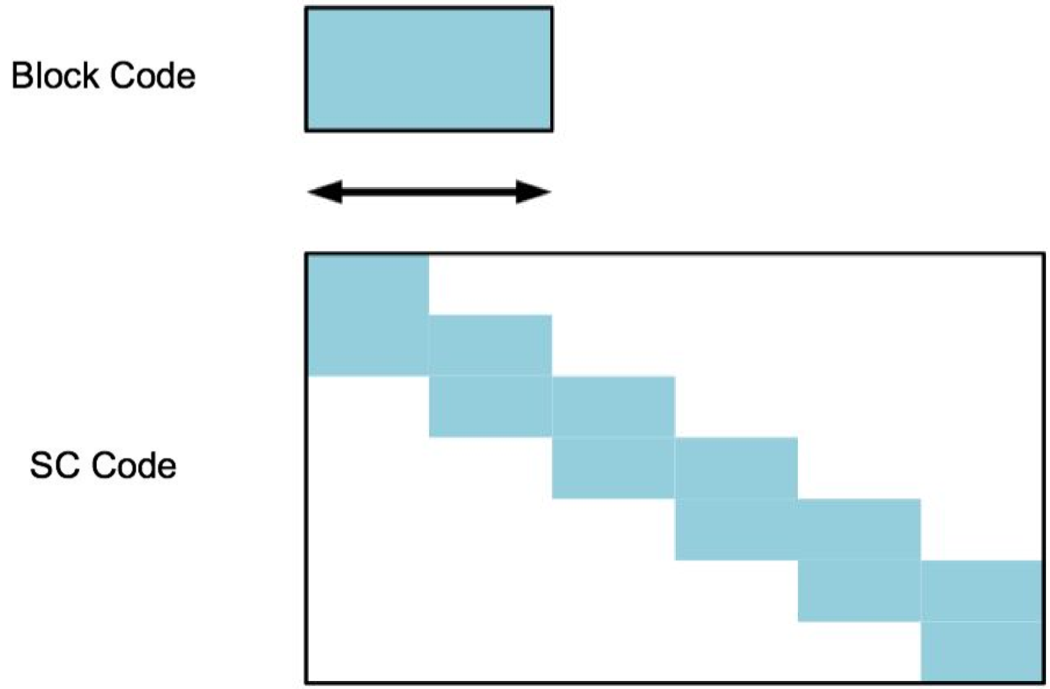
Spatially-coupled codes outperform block codes

- **Binary array-based code with circulant size p=67 and *column weight* 3, and it's derived SC codes with coupling length *L* = 50.**



Block Codes vs. Spatially-Coupled Codes

- **The research of SC codes often motivated by their "superior" performance (threshold saturation phenomenon, etc.)**
- **Block codes and SC codes are compared for fixed constraint lengths (equal decoding latency)?**



Presentation Outline

- **Background and motivation**
- **Non-binary LDPC code optimization for flash memory**
  - New combinatorial objects
  - The optimization framework
  - Simulation results for practical Flash
- **Analysis and design of spatially-coupled LDPC codes**
  - Block vs spatially-coupled LDPC codes
  - The optimization framework
  - Simulation results for AWGN channel
- **Conclusions and future work**

Conclusions and Future work

- **The nature of the errors which dominate the error floor region of NB-LDPC codes in flash memories is different from the AWGN channel.**
  - GASTs are the objects of interest in practical Flash channels.
- **Our framework results at least one order of magnitude performance gain.**
- **We demonstrated that SC codes always have fewer problematic absorbing sets than block codes and that the choice of the cutting vector matters.**
- **Future work includes:**
  - Developing SC code design techniques for flash memories.

References

- [Dolecek 10] L. Dolecek *et al.*, "Analysis of absorbing sets and fully absorbing sets of array-based LDPC codes," *IEEE Trans. Inform. Theory*, 2010.
- [Parnell 14] T. Parnell *et al.*, "Modelling of the threshold voltage distributions of sub-20nm NAND flash memory," in *Proc. IEEE GLOBECOM*, 2014.
- [Cai 13] Y. Cai *et al.*, "Threshold voltage distribution in MLC NAND flash memory: Characterization, analysis, and modeling," in *Proc. DATE*, 2013.
- [Bazarsky 13] A. Bazarsky *et al.*, "Design of non-binary quasi-cyclic LDPC codes by ACE optimization," in *Proc. IEEE ITW*, 2013.
- [Amiri 14] B. Amiri *et al.*, "Analysis and enumeration of absorbing sets for non-binary graph-based codes," *IEEE Trans. Commun.*, 2014.
- [Hareedy 16] A. Hareedy *et al.*, "The weight consistency matrix framework for general non-binary LDPC code optimization: applications in Flash memories," in Proc. *IEEE International Symp. Inform. Theory*, 2016.
- [Costello 14] K. Huang *et al.*, "Performance comparison of non-binary LDPC block and spatially coupled codes," *IEEE International Symp. Inform. Theory*, 2014.
- [Urbanke 13] S. Kudekar *et al.*, " Spatially Coupled Ensembles Universally Achieve Capacity Under Belief Propagation," *IEEE Trans. Inform. Theory*, 2013.
- [Lentmaier 15] D. G. M. Mitchell *et al.*, " Spatially Coupled LDPC Codes Constructed From Protographs," *IEEE Trans. Inform. Theory*, 2015.

Thank You