Flash Memory Summit

# Spatially-Coupled (SC) Codes for Flash Memories

**Homa Esfahanizadeh**, Ahmed Hareedy and Lara Dolecek
hesfahanizadeh@ucla.edu
LORIS Lab, UCLA

08/9/2017

# Presentation Outline

- **Background and Motivation**
  - Block vs Spatially-Coupled (SC) LDPC Codes
  - Error Floor of SC/LDPC Codes: Problematic Objects

- **A Three-Step Approach to Construct NB SC Codes for Flash Memories**
  a. Optimum Partitioning of the Underlying Block Code
  b. Circulant Power Optimization
  c. Edge Weight Optimization

- **Simulation Results**

- **Conclusions and Future Work**

# Presentation Outline

- **Background and Motivation**
  - Block vs Spatially-Coupled (SC) LDPC Codes
  - Error Floor of SC/LDPC Codes: Problematic Objects

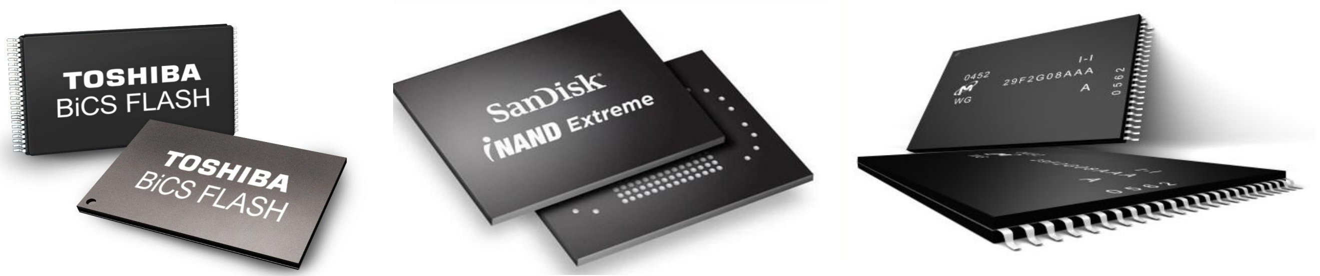
- **A Three-Step Approach to Construct NB SC Codes for Flash Memories**
  a. Optimum Partitioning of the Underlying Block Code
  b. Circulant Power Optimization
  c. Edge Weight Optimization

- **Simulation Results**

- **Conclusions and Future Work**

# High Performance Error Correcting Codes for Flash

- Modern dense Flash memory devices operate at a very low error rates.




- High performance ECCs with outstanding error floor performance are required
- SC/LDPC codes are graph-based codes with capacity approaching performance.
- Our designed SC codes outperform state-of-the-art SC codes over Flash channels.

# Spatially-Coupled (SC) Code as a Chain of Block Codes

- An SC code with memory *m* and coupling length L is formed as follows:
  - Partitioning a block code *H* into (m+1) component matrices.
  - Coupling L copies of the component matrices together to make a chain of coupled block codes.

$$H_0 \quad H_1 \cdots\cdots \quad H_m$$

$$\sum_{i=0}^{m} H_i = H$$

$$\begin{array}{cc} & \begin{array}{cccc} 1 & 2 & \cdots\cdots & L \end{array} \\ H_{SC} = & \begin{bmatrix} H_0 & 0 & & 0 \\ H_1 & H_0 & & \vdots \\ \vdots & H_1 & \ddots & \\ H_m & \vdots & \ddots & H_0 \\ 0 & H_m & \ddots & H_1 \\ \vdots & 0 & & \vdots \\ & \vdots & & H_m \end{bmatrix} \end{array}$$

# Importance of Finite Length Analysis of SC Codes

- SC codes are shown to have an excellent performance in the regime of extremely long block lengths when averaged over many codes [Urbanke et al. 13].
- The asymptotic analysis is based on assumptions e.g., cycle-free that do not apply to the practical finite length set-up.

# Importance of Finite Length Analysis of SC Codes

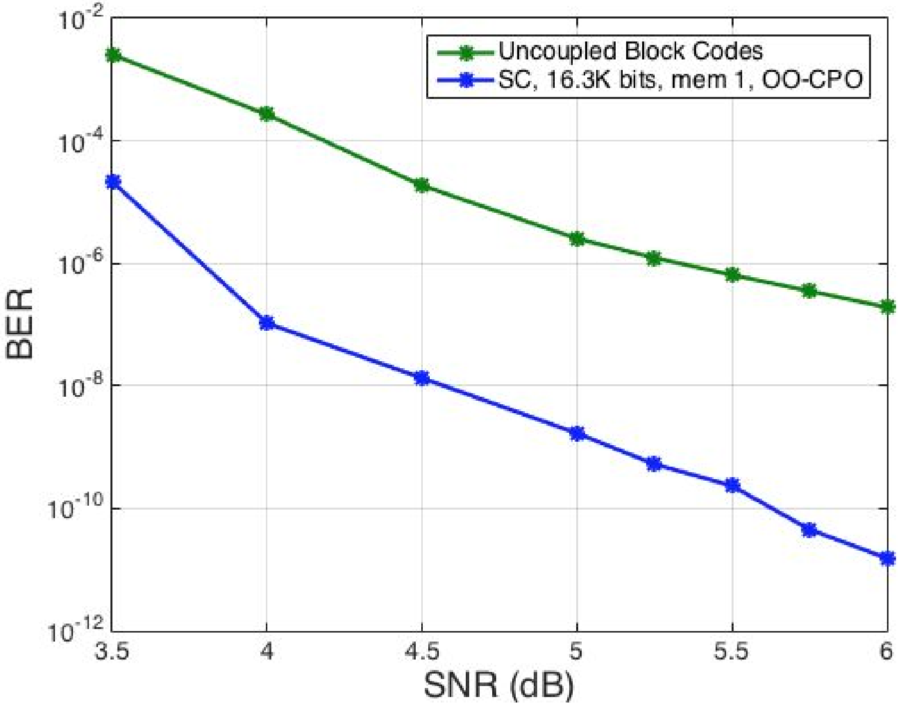
- SC codes are shown to have an excellent performance in the regime of extremely long block lengths when averaged over many codes [Urbanke et al. 13].
- The asymptotic analysis is based on assumptions e.g., cycle-free that do not apply to the practical finite length set-up.
- We propose a novel combinatorial approach for constructing SC codes with outstanding finite length performance [Esfahanizadeh et al. 17, Hareedy et al. 17].

# Importance of Finite Length Analysis of SC Codes

- SC codes are shown to have an excellent performance in the regime of extremely long block lengths when averaged over many codes [Urbanke et al. 13].
- The asymptotic analysis is based on assumptions e.g., cycle-free that do not apply to the practical finite length set-up.
- We propose a novel combinatorial approach for constructing SC codes with outstanding finite length performance [Esfahanizadeh et al. 17, Hareedy et al. 17].
- Our new method outperforms existing methods for constructing SC codes on a variety of practical channels including AWGN and Flash channels.

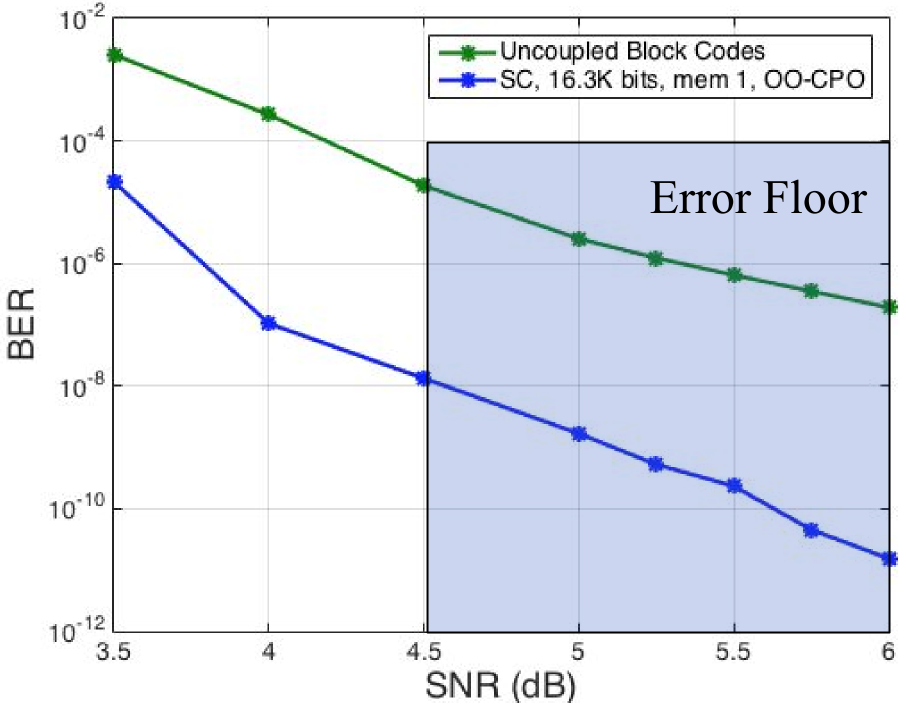
# SC Codes Outperform Block Codes for AWGN Channels

- Our well-designed SC code with length 16K bits and rate 0.818, compared to the uncoupled block codes.



# Error Floor of Graph Codes

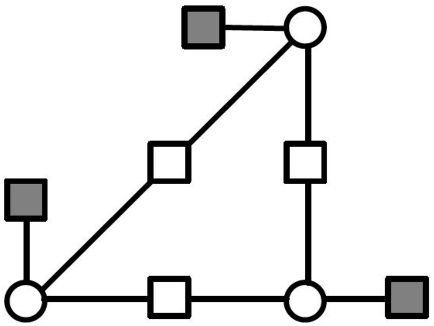
- For AWGN channels, absorbing sets (ASs) [Dolecek et al. 10] are the reason behind the error floor.



# Error Floor of Graph Codes: Problematic Objects

- **An <u>(a, b) absorbing set (AS):</u>**
  - 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
  - each variable node is connected to more satisfied than unsatisfied check nodes.

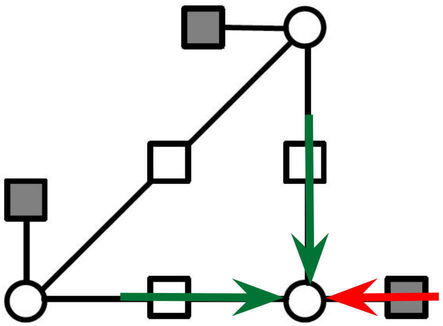
Example: (3,3) AS



# Error Floor of Graph Codes: Problematic Objects

- **An <u>(a, b) absorbing set (AS):</u>**
  - 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
  - each variable node is connected to more satisfied than unsatisfied check nodes.

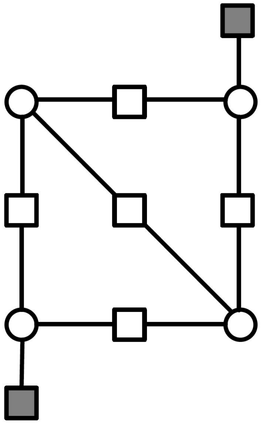
Example: (3,3) AS



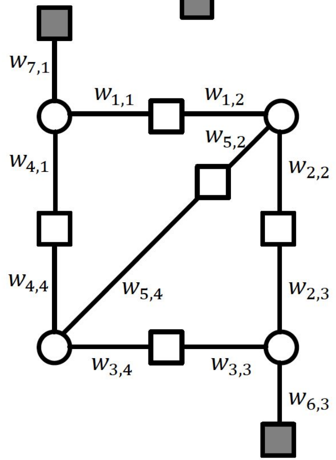
When transmission errors happen on the three variable nodes, each variable node receives more satisfied messages (green) than unsatisfied messages (red).

# Binary vs. Non-binary Absorbing Sets

- **Binary absorbing sets**
  - For Binary graph-based codes
  - Described in terms of topological conditions only



- **Non-binary absorbing sets**
  - For non-binary graph-based codes
  - The values of the variable nodes also matter.
  - $w_{i,j}$ are non-zero elements in Galois Field.

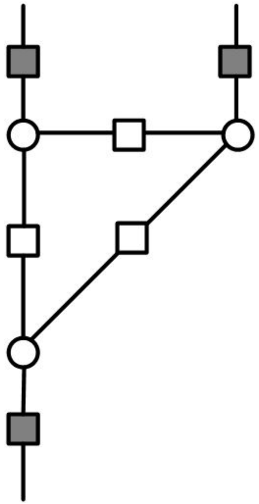


# Objects of Interest for the AWGN Channel

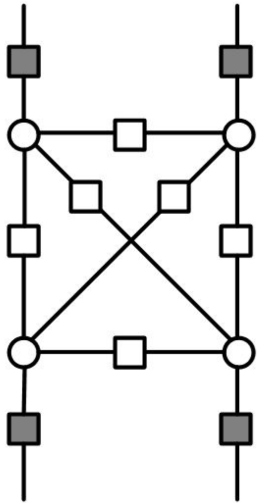
- For symmetric channels (like AWGN), the dominant objects are the elementary ASs.
- Elementary AS: Each satisfied check node is of degree 2 and each unsatisfied check node is of degree 1.

- Examples:
  - (3,3) AS, $\gamma$=3.
  - (4,4) AS, $\gamma$=4.

  Where $\gamma$ is the column weight of the code.



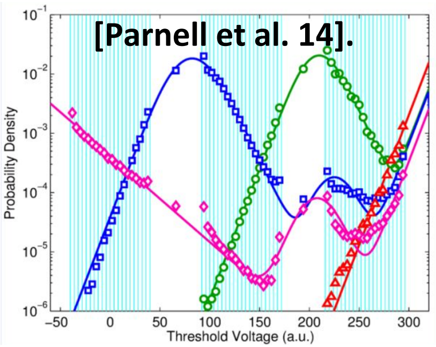
(3,3) AS

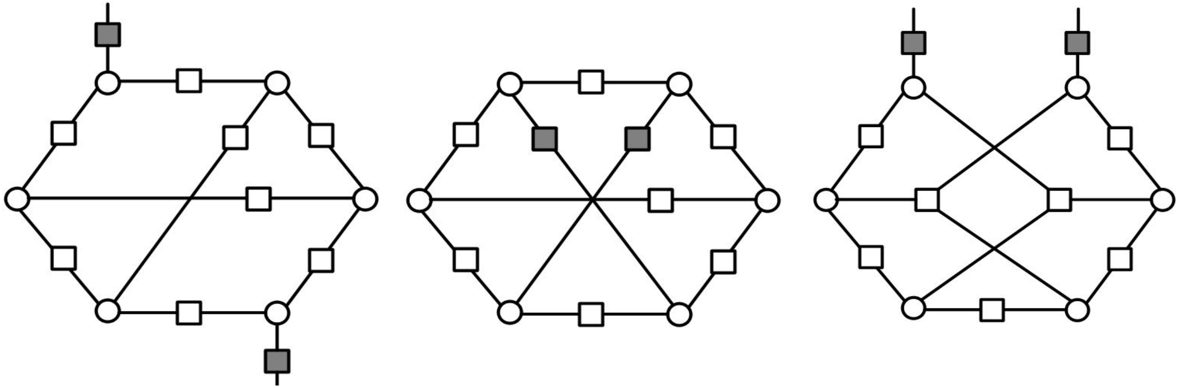


(4,4) AS

# New Combinatorial Objects Are Needed for Flash

- Asymmetry in the channel (e.g., in Flash) can result in:
  - ASs with unsatisfied check nodes having degree > 1.
  - 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, 2) non-elementary AS ($\gamma$=3).



Appropriate non-binary edge weights are assumed for these configurations.

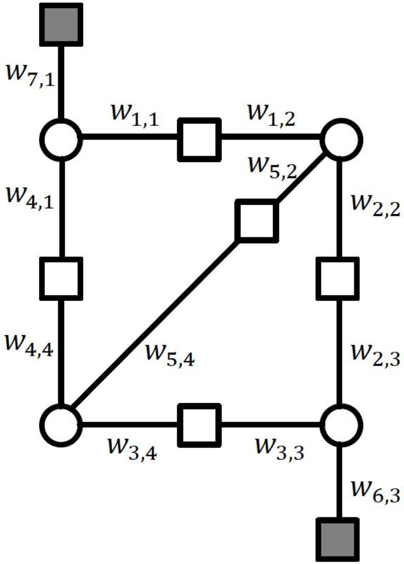
# Error Floor of Graph Codes for Flash

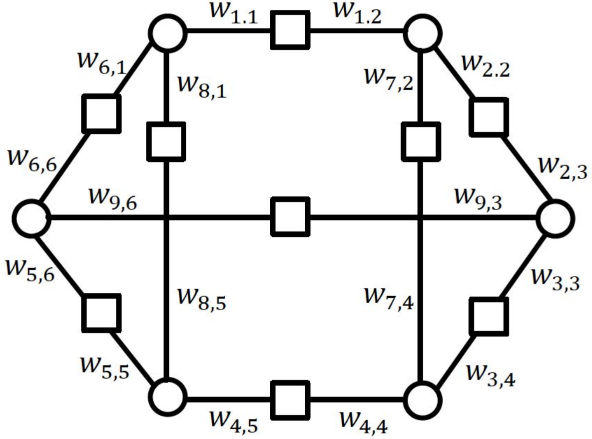
- General absorbing set (GAS)
  - Unsatisfied check nodes of degree $\geq 1$.
  - Satisfied check nodes of degree $\geq 2$.

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

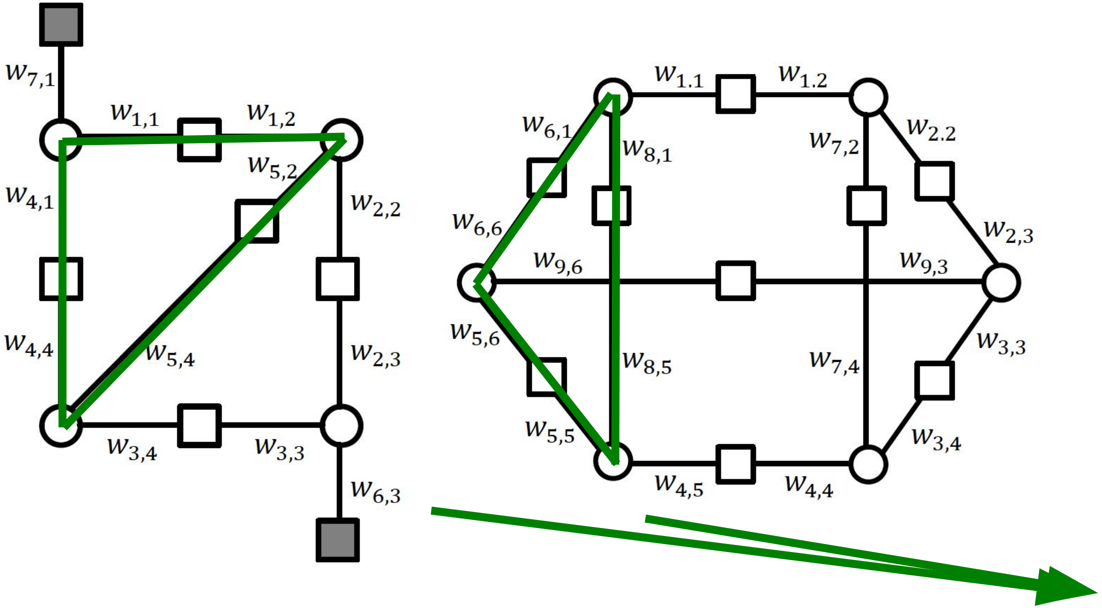
- A GAS of type two (GAST) is a GAS that:
  - The number of degree 2 check nodes is higher than the number of degree $> 2$ check nodes.
  - Unsatisfied check nodes are of degree 1 or degree 2.

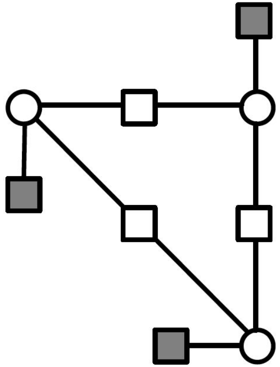
# Two Dominant GASTs for NB codes with γ= 3 over Flash



# Problematic Objects Share a Common Denominator



We avoid the <u>common denominator</u> in the Tanner graph of SC/LDPC codes, and eliminate as many as possible of remaining problematic GASTs using edge weight optimization.

# Presentation Outline

- **Background and Motivation**
  - Block vs Spatially-Coupled (SC) LDPC Codes
  - Error Floor of SC/LDPC Codes: Problematic Objects

- **A Three-Step Approach to Construct NB SC Codes for Flash Memories**
  a. Optimum Partitioning of the Underlying Block Code
  b. Circulant Power Optimization
  c. Edge Weight Optimization

- **Simulation Results**

- **Conclusions and Future Work**

# Circulant Based SC Code Design for Flash

- We have recently developed a three stage framework to design circulant-based SC codes with outstanding performance:
  - **Optimal Overlap partitioning:**
  - **Circulant Power Optimization:**
  - **Weight Consistency Matrix framework [Hareedy et al. 16]:**

# Circulant Based SC Code Design for Flash, Step 1

- We have recently developed a three stage framework to design circulant-based SC codes with outstanding performance:
    - **Optimal Overlap partitioning:**
      A novel partitioning mechanism which acts on the protograph of the SC code to produce optimal partitioning corresponding to the smallest number of detrimental objects.
    - **Circulant Power Optimization:**
    - **Weight Consistency Matrix framework [Hareedy et al. 16]:**

# Circulant Based SC Code Design for Flash, Step 2

- We have recently developed a three stage framework to design circulant-based SC codes with outstanding performance:
  - **Optimal Overlap partitioning:**
  - **Circulant Power Optimization:**
    This step reduces the number of problematic objects after power lifting.
  - **Weight Consistency Matrix framework [Hareedy et al. 16]:**

# Circulant Based SC Code Design for Flash, Step 3

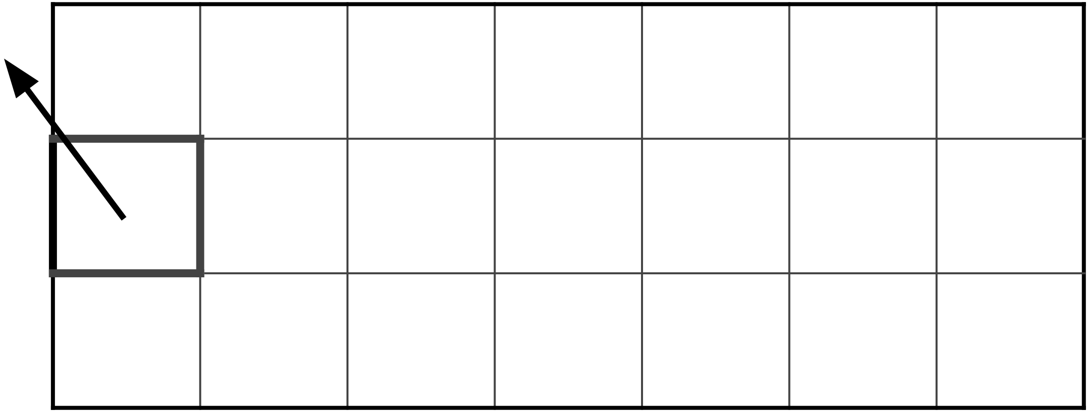
- We have recently developed a three stage framework to design circulant-based SC codes with outstanding performance:
  - **Optimal Overlap partitioning:**
  - **Circulant Power Optimization:**
  - **Weight Consistency Matrix framework [Hareedy et al. 16]:**
    It eliminates as many as possible of the remaining problematic objects by edge weight assignments.

# STEP 1: Optimum Overlap (OO) Partitioning

- **Optimal Overlap partitioning:**
  A novel partitioning mechanism which acts on the protograph of the SC code to produce <u>optimal partitioning</u> corresponding to the smallest number of detrimental objects.

Circulants in the underlying block codes

| | | | | |
|---|---|---|---|---|
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 |



# STEP 1: Optimum Overlap (OO) Partitioning

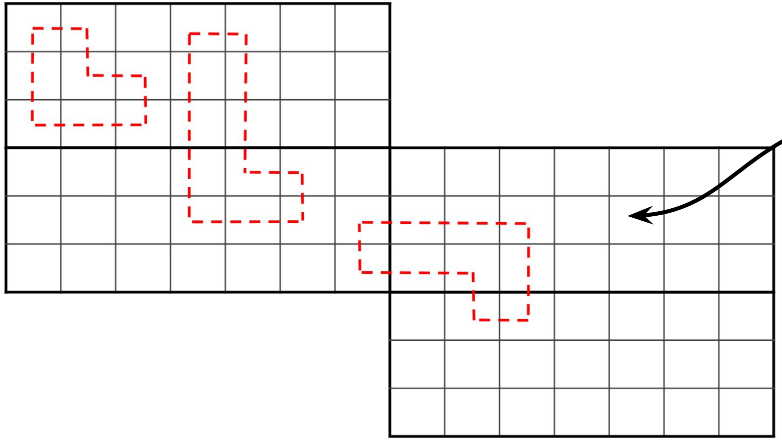
- **Optimal Overlap partitioning:**
  A novel partitioning mechanism which acts on the protograph of the SC code to produce optimal partitioning corresponding to the smallest number of detrimental objects.

(green/0) $\longrightarrow H_0$ (blue/1) $\longrightarrow H_1$

| | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 | 1 | 1 |

$m = 1$

# STEP 1: Optimum Overlap (OO) Partitioning

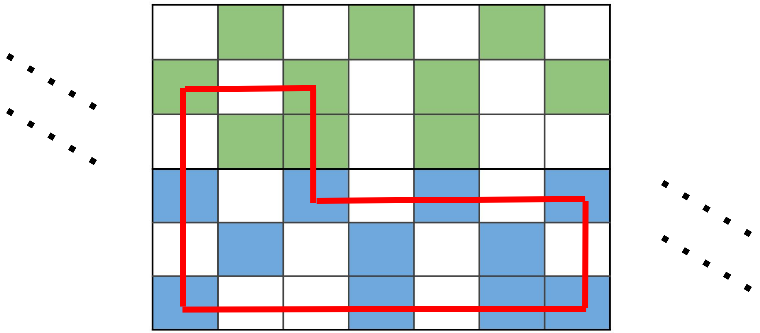


$$H_{SC} = \begin{bmatrix} H_0 & 0 & & 0 \\ H_1 & H_0 & & \vdots \\ \vdots & H_1 & \ddots & \\ H_m & \vdots & \ddots & H_0 \\ 0 & H_m & & H_1 \\ \vdots & 0 & & \vdots \\ & \vdots & & H_m \end{bmatrix}$$

- We developed a combinatorial approach that finds the optimum partitioning which results the minimum number of cycles 6, or *(3, 3)* AS, in the protograph SC code [Esfahanizadeh et al. 17, Hareedy et al. 17].
- Our combinatorial approach is based on optimizing <u>seven overlap parameters</u> that determine the number of cycles for SC codes with memory 1.

# STEP 2: Circulant Power Optimization (CPO)

- **Power lifting:** Each 1 in the protograph SC code is replaced with a circulant matrix, and each 0 is replaced with a zero matrix.



Green: non-zero circulants in $H_0$

Blue: non-zero circulants in $H_1$

- Each cycle in the protograph SC code results in **either *p* or *0* cycles** in the SC code after power lifting, depending on whether a condition on the power of circulants involved in that cycle holds or not.
- We have proposed a heuristic approach that adjusts the circulant powers to break those circulant power conditions.

# STEP 3: Edge Weight Optimization for NB SC Codes

- **Edge Weight Optimization:**
  - By optimum overlap (OO) partitioning and circulant power optimization (CPO), the number of instances of the common denominator in the Tanner graph of SC code is minimized.
  - Then, by performing edge weight optimization, as many as possible of remaining GASTs are removed.
  - We established a set of necessary algebraic conditions such that when we break at least one of them, the GAST is removed.
    Our WCM framework [Hareedy et al. 16]

# STEP 3: Algorithm for Edge Weight 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).
- **We also tested our framework on Cai Flash channel [Cai et al. 13].**

# Presentation Outline

- **Background and Motivation**
  - Block vs Spatially-Coupled (SC) LDPC Codes
  - Error Floor of SC/LDPC Codes: Problematic Objects

- **A Three-Step Approach to Construct NB SC Codes for Flash Memories**
  a. Optimum Partitioning of the Underlying Block Code
  b. Circulant Power Optimization
  c. Edge Weight Optimization

- **Simulation Results**

- **Conclusions and Future Work**

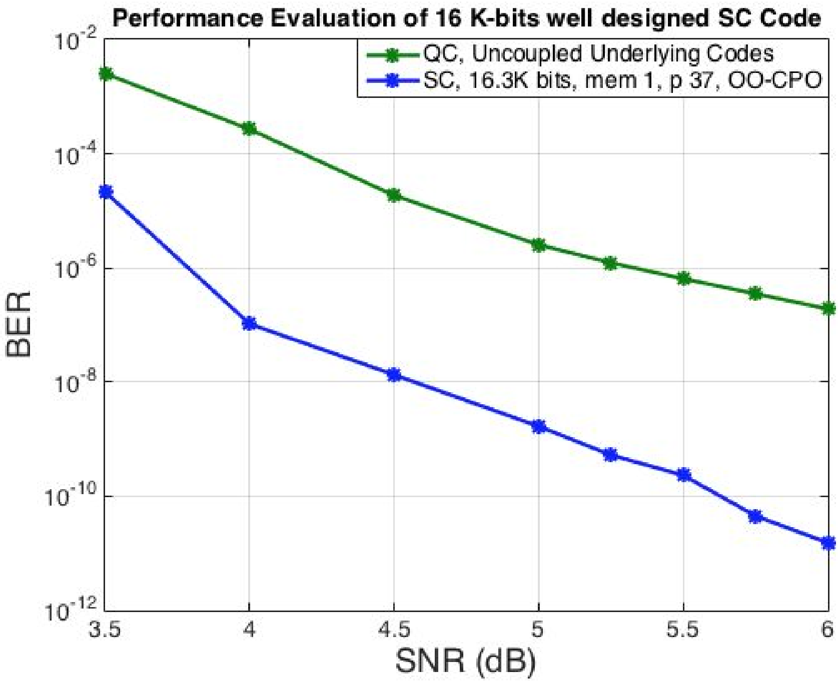
# Comparing the Number of (3, 3) ASs for B-SC Codes

- Consider the underlying block code is a circulant-based code with $\gamma=3$. We construct SC codes with L=30, m=1, and using **different schemes**:

| | **Number of *(3, 3)* ASs** | | | |
|---|---|---|---|---|
| **Code Length** | 1.4 K | 3.6 K | 5 K | 8.6 K |
| Uncoupled Case | 8820 | 36300 | 60840 | 138720 |
| Cutting Vector [Mitchell et al. 14] | 3290 | 14872 | 25233 | 59024 |
| OO-CPO | 203 | 2596 | 5356 | 14960 |

# SC Codes Designed for AWGN Channel

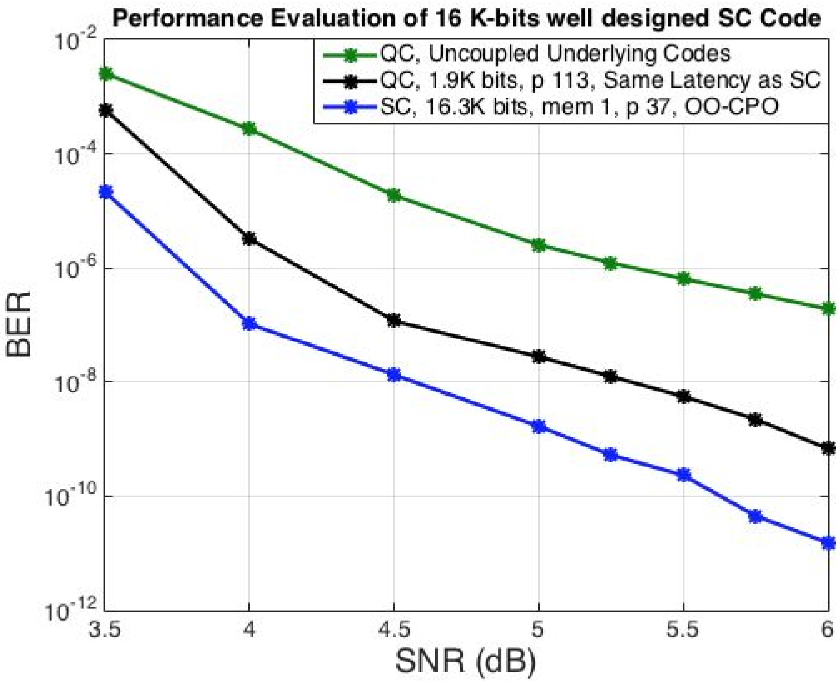
- Simulation results over the AWGN channel for binary SC codes with length 16K and rate 0.818, compared to arbitrarily designed SC code and block codes:




**Uncoupled Block Codes** vs. **SC Code**

# SC Codes Designed for AWGN Channel

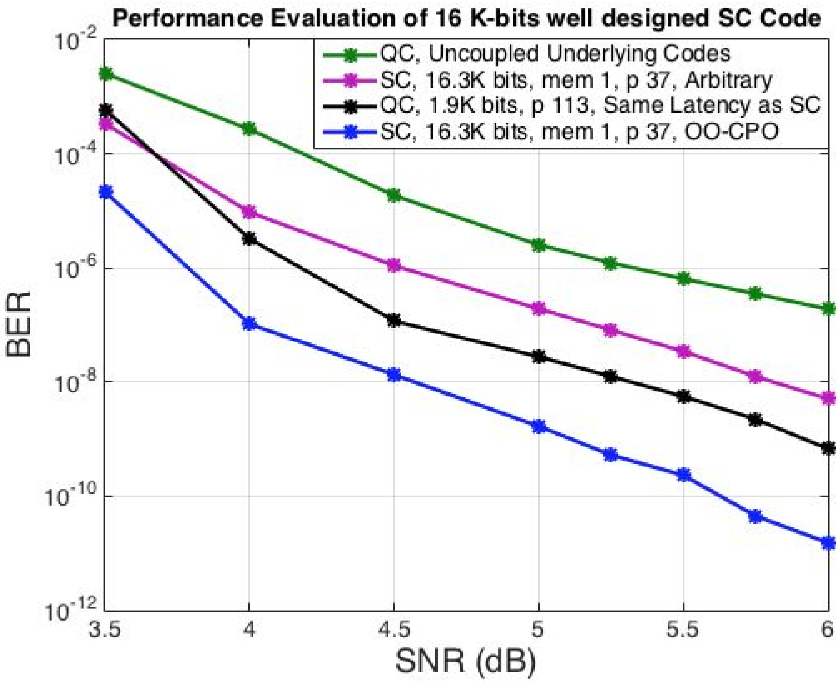
- Simulation results over the AWGN channel for binary SC codes with length 16K and rate 0.818, compared to arbitrarily designed SC code and block codes:




**SC Code** vs. **Block Code with Same Decoding Latency**

# SC Codes Designed for AWGN Channel

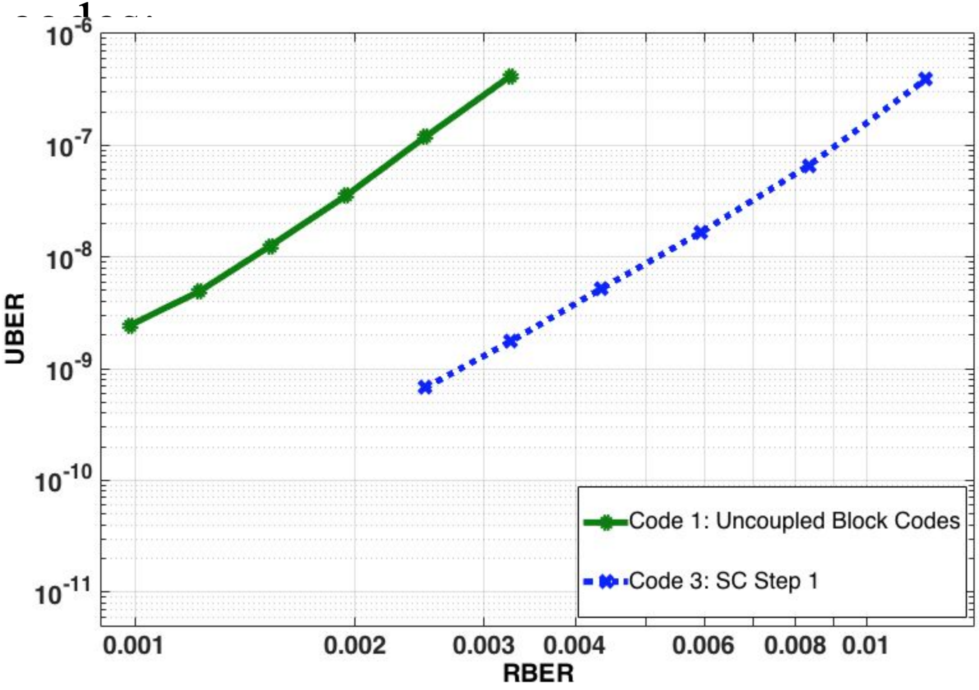
- Simulation results over the AWGN channel for binary SC codes with length 16K and rate 0.818, compared to arbitrarily designed SC code and block codes:




**Arbitrarily** vs.
**Well-Designed SC Code**

# SC Codes Designed for Flash Channel

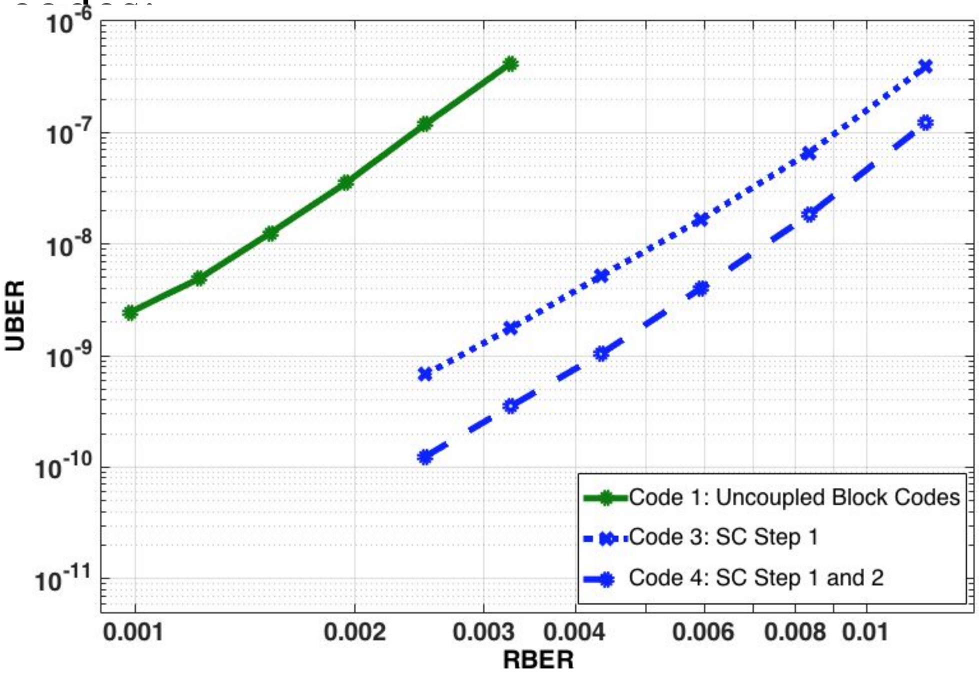
- Simulation results over the Flash channel for non-binary SC codes with length 5.7K and rate 0.815, compared to arbitrarily designed SC code and block codes:




**Uncoupled Block Codes** vs. **SC Code Step 1**

# SC Codes Designed for Flash Channel

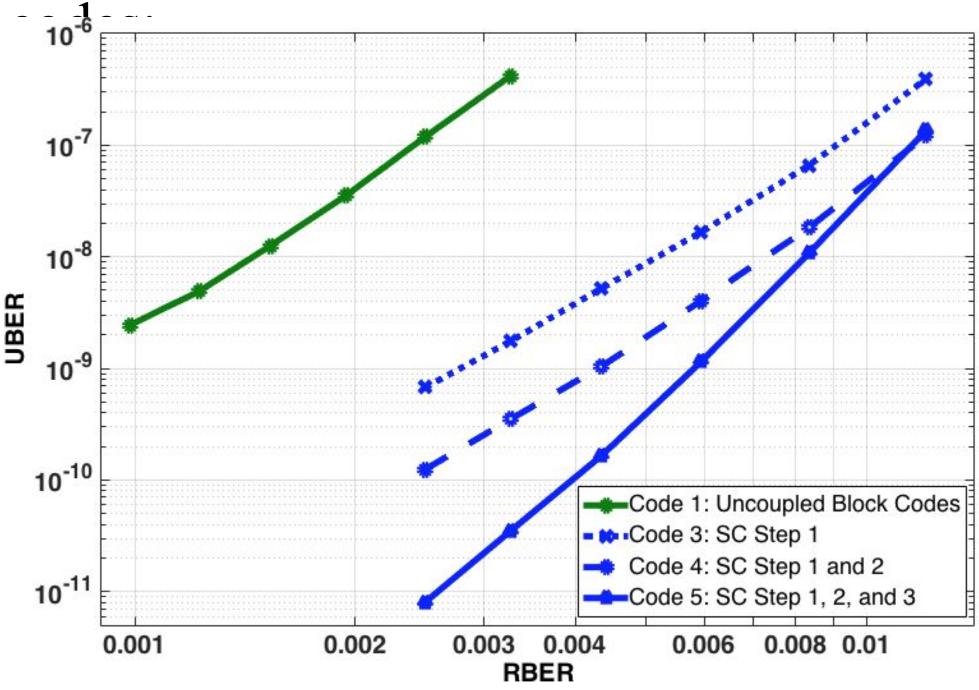
- Simulation results over the Flash channel for non-binary SC codes with length 5.7K and rate 0.815, compared to arbitrarily designed SC code and block codes:




**Uncoupled Block Codes** vs. **SC Code Step 1, 2**

# SC Codes Designed for Flash Channel

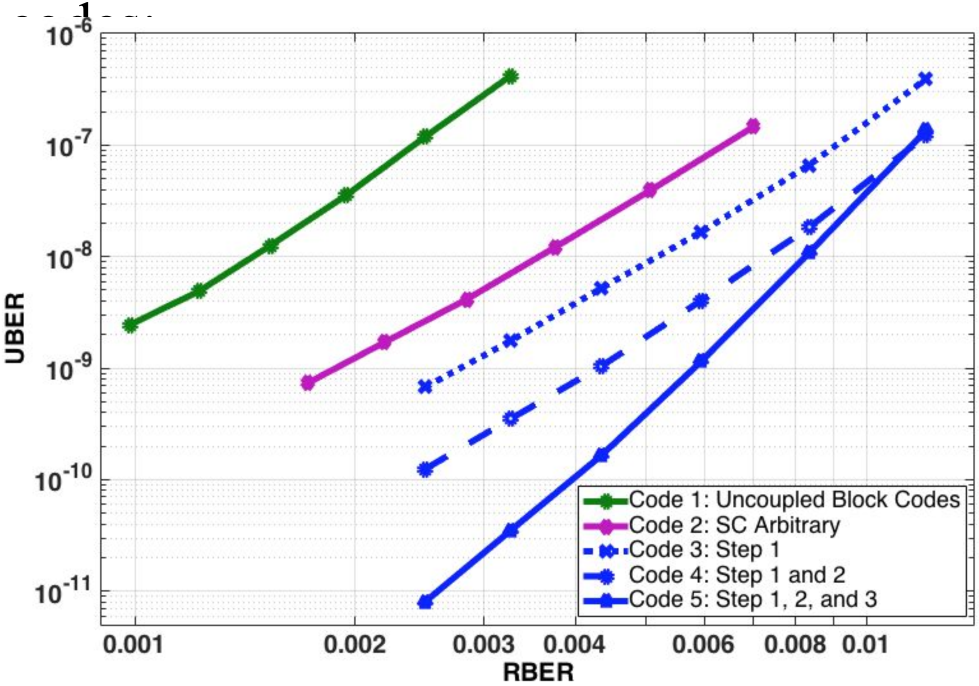
- Simulation results over the Flash channel for non-binary SC codes with length 5.7K and rate 0.815, compared to arbitrarily designed SC code and block codes:




**Uncoupled Block Codes** vs. **SC Code Step 1, 2, 3**

# SC Codes Designed for Flash Channel

- Simulation results over the Flash channel for non-binary SC codes with length 5.7K and rate 0.815, compared to arbitrarily designed SC code and block codes:




**Arbitrarily** vs.
**Well-Designed SC Code**

# Presentation Outline

- **Background and Motivation**
  - Block vs Spatially-Coupled (SC) LDPC Codes
  - Error Floor of SC/LDPC Codes: Problematic Objects

- **A Three-Step Approach to Construct NB SC Codes for Flash Memories**
  a. Optimum Partitioning of the Underlying Block Code
  b. Circulant Power Optimization
  c. Edge Weight Optimization

- **Simulation Results**

- **Conclusions and Future Work**

# Conclusions and Future Work

- We developed a three-stage approach for the design of high performance non-binary SC codes optimized for practical Flash channels.
- We aim at minimizing the number of detrimental general absorbing sets of type two (GASTs) in the graph of the designed SC code.
- Simulation results reveal that SC codes designed using our approach outperform state-of-the-art SC codes when used over Flash channels.
- Future work includes:
  - Developing SC code design techniques for higher memories and column weights.

# References


- [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.
- [Dolecek 10] L. Dolecek *et al.*, “Analysis of absorbing sets and fully absorbing sets of array-based LDPC codes,” *IEEE Trans. Inform. Theory*, 2010.
- [Bazarsky 13] A. Bazarsky *et al.*, “Design of non-binary quasi-cyclic LDPC codes by ACE optimization,” in *Proc. IEEE ITW*, 2013.
- [Esfahanizadeh 17] H. Esfahanizadeh et al., “A Novel Combinatorial Framework to Construct Spatially-Coupled Codes: Minimum Overlap Partitioning”, in *Proc. IEEE ISIT*, 2017.
- [Hareedy 17] A. Hareedy et al., “High Performance Non-Binary Spatially-Coupled Codes for Flash Memories” in *Proc. IEEE ITW*, 2017.
- [Hareedy 16] A. Hareedy *et al.*, “A General Non-Binary LDPC Code Optimization Framework Suitable for Dense Flash Memory and Magnetic Storage," JSAC, 2016.
- [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.
- [Mitchell 14] D. G. M. Mitchell et al., “Absorbing set characterization of array-based spatially coupled LDPC codes,” in *Proc. ISIT*, 2014.

# Thank You