Non-binary LDPC Codes:
The Next Frontier in ECC for Flash

Prof. Lara Dolecek
LORIS Lab and CoDESS Center
Electrical Engineering Department, UCLA

FMS : 08/07/2014

The Case for Non-binary LDPC Codes

- **LDPC codes outperform commonly used BCH codes in Flash**

The Case for Non-binary LDPC Codes

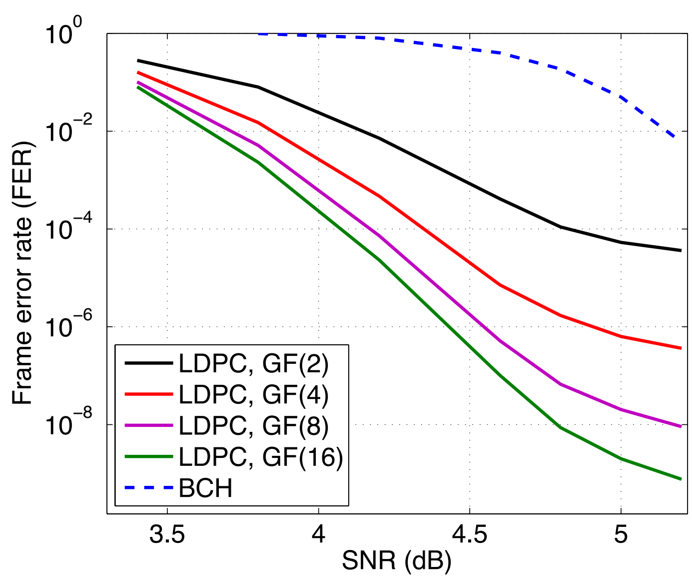
- LDPC codes outperform commonly used BCH codes in Flash
- The constructions thus far have primarily focused on binary LDPC codes

The Case for Non-binary LDPC Codes

- LDPC codes outperform commonly used BCH codes in Flash
- The constructions thus far have primarily focused on binary LDPC codes
- Non-binary LDPC codes offer substantial performance improvement over binary LDPC codes

The Case for Non-binary LDPC Codes

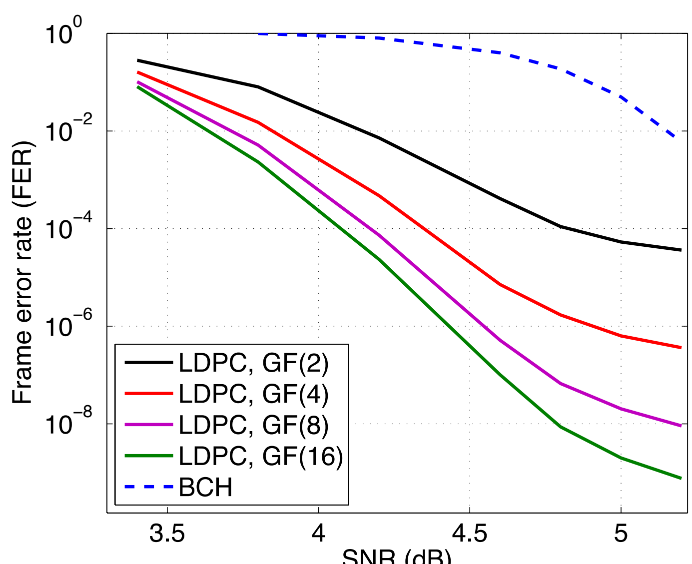
- LDPC codes outperform commonly used BCH codes in Flash
- The constructions thus far have primarily focused on binary LDPC codes
- Non-binary LDPC codes offer substantial performance improvement over binary LDPC codes
- A representative example:



- Code parameters:
  – Code length = 1000 bits,
  – Code rate = 0.9,
  – Column weight = 4

The Case for Non-binary LDPC Codes

- **LDPC codes outperform commonly used BCH codes in Flash**
- **The constructions thus far have primarily focused on binary LDPC codes**
- **Non-binary LDPC codes offer substantial performance improvement over binary LDPC codes**
- **A representative example:**



- **Code parameters:**
  - **Code length = 1000 bits,**
  - **Code rate = 0.9,**
  - **Column weight = 4**

For multilevel flash, well-designed non-binary LDPC codes show a lot of promise

LDPC Codes

- Low-density parity-check (LDPC) codes are a class of channel codes described by sparse parity check matrices, or equivalently by sparse bipartite graphs

- Well known capacity achieving performance

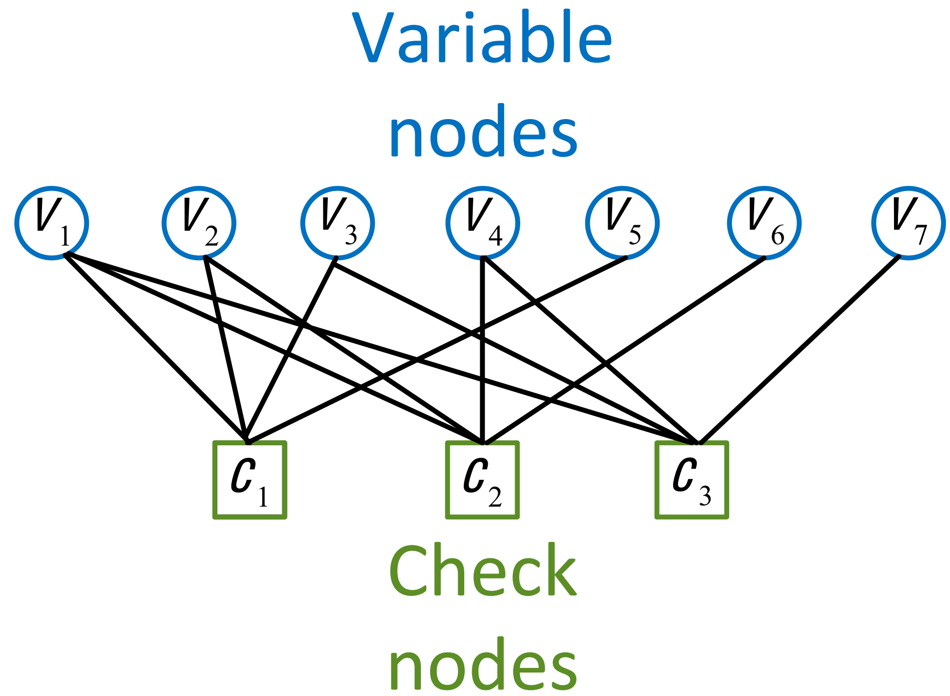
- Graph representation allows for design of practical iterative algorithms

Binary LDPC Codes

- **An example:**

Binary code

$$
H = \begin{array}{c}
\begin{array}{ccccccc} v_1 & v_2 & v_3 & v_4 & v_5 & v_6 & v_7 \end{array} \\
\begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}
\end{array}
\begin{array}{c} \\ c_1 \\ c_2 \\ c_3 \end{array}
$$



- **Parity check c_1 tells us that $v_1+v_2+v_3+v_5 = 0 \bmod 2$.**

Symbol-Based Representation

- Instead of representing values in bits, represent values in symbols
- Each symbol represents a prescribed number of bits
- Symbols take values in a Galois Field of size q (GF(q)), where q is a power of 2.

- Example: GF(4)={0,1,2,3}

+	0	1	2	3
0	0	1	2	3
1	1	0	3	2
2	2	3	0	1
3	3	2	1	0

x	0	1	2	3
0	0	0	0	0
1	0	1	2	3
2	0	2	3	1
3	0	3	1	2

Non-binary LDPC Codes

- **Variable nodes take values in GF(q), and edges of the bipartite graph and weighted by non-zero elements of GF(q)**
- **An example:**

Non-binary code over GF(4)

$$
H = \begin{array}{c} \\ \\ \end{array}
\begin{array}{cc}
\begin{array}{ccccccc} V_1 & V_2 & V_3 & V_4 & V_5 & V_6 & V_7 \end{array} & \\
\begin{bmatrix} 2 & 1 & 1 & 0 & 3 & 0 & 0 \\ 1 & 2 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 3 & 1 & 0 & 0 & 2 \end{bmatrix} & \begin{array}{c} c_1 \\ c_2 \\ c_3 \end{array}
\end{array}
$$

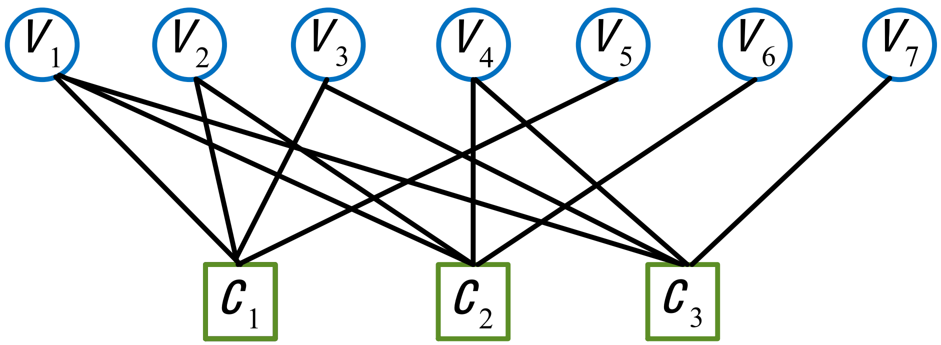
Non-binary LDPC Codes

- Variable nodes take values in GF(q), and edges of the bipartite graph and weighted by non-zero elements of GF(q)
- An example:

Non-binary code over GF(4)

$$
\begin{array}{cc}
 & \begin{matrix} V_1 & V_2 & V_3 & V_4 & V_5 & V_6 & V_7 \end{matrix} \\
H = & \begin{bmatrix} 2 & 1 & 1 & 0 & 3 & 0 & 0 \\ 1 & 2 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 3 & 1 & 0 & 0 & 2 \end{bmatrix} \begin{matrix} C_1 \\ C_2 \\ C_3 \end{matrix}
\end{array}
$$

Variable nodes



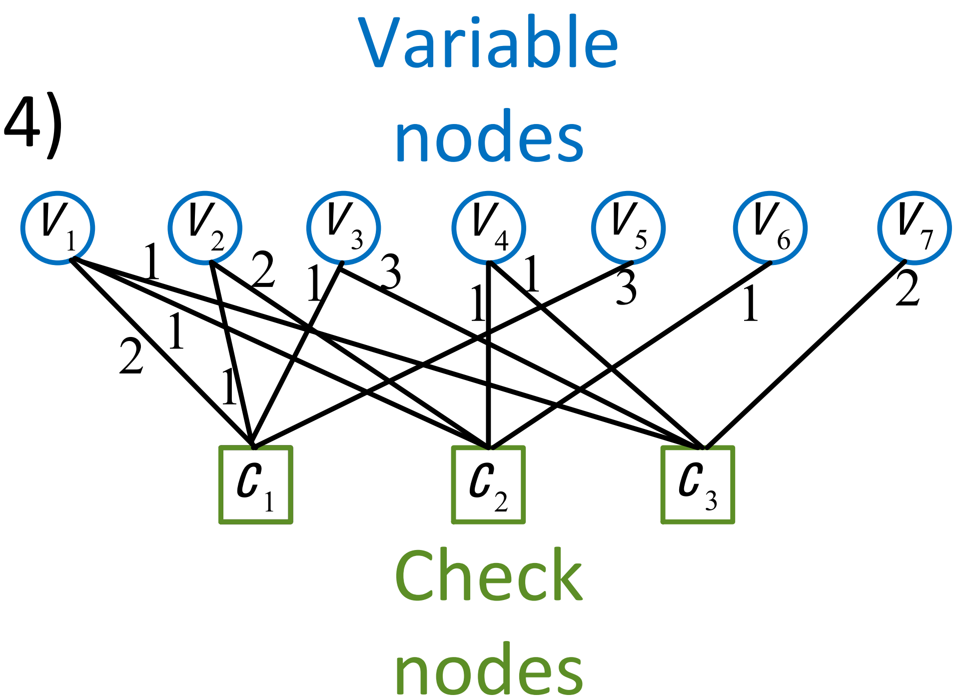
Check nodes

Non-binary LDPC Codes

- Variable nodes take values in GF(q), and edges of the bipartite graph and weighted by non-zero elements of GF(q)
- An example:

Non-binary code over GF(4)

$$
\begin{array}{c} \\ H = \end{array}
\begin{array}{cc}
\begin{array}{ccccccc} V_1 & V_2 & V_3 & V_4 & V_5 & V_6 & V_7 \end{array} & \\
\begin{bmatrix} 2 & 1 & 1 & 0 & 3 & 0 & 0 \\ 1 & 2 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 3 & 1 & 0 & 0 & 2 \end{bmatrix} & \begin{array}{c} c_1 \\ c_2 \\ c_3 \end{array}
\end{array}
$$

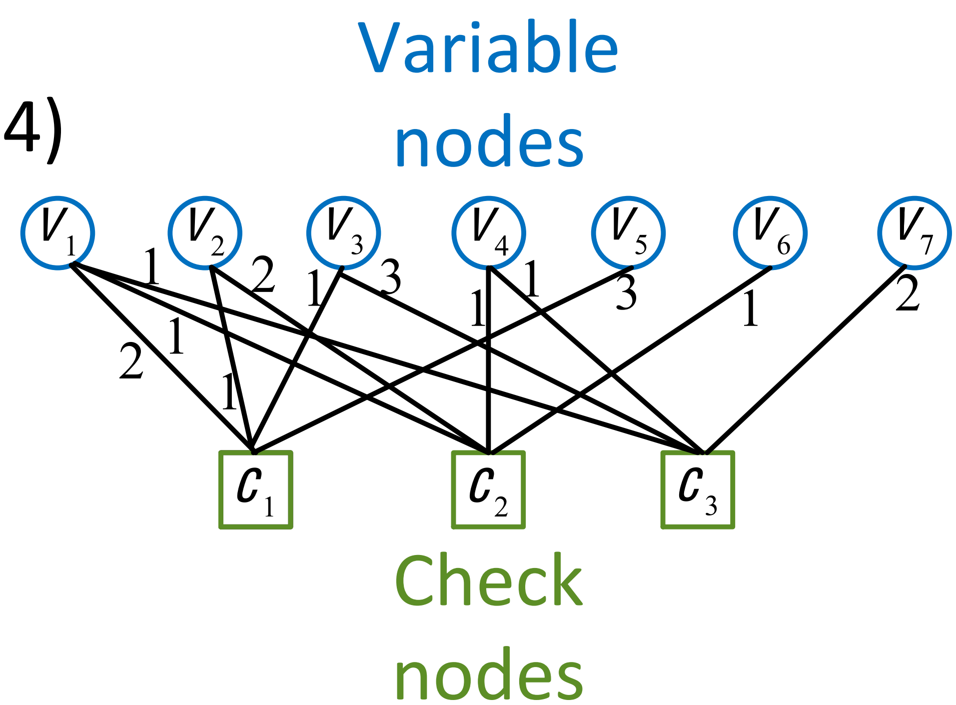


Non-binary LDPC Codes

- **Variable nodes take values in GF(q), and edges of the bipartite graph and weighted by non-zero elements of GF(q)**
- **An example:**

Non-binary code over GF(4)

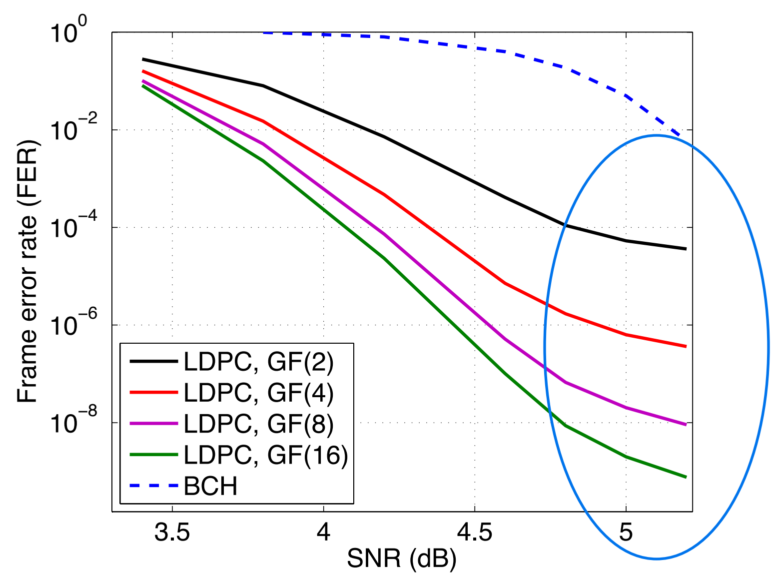
$$
\begin{array}{cc}
 & \begin{array}{ccccccc} V_1 & V_2 & V_3 & V_4 & V_5 & V_6 & V_7 \end{array} \\
H = & \begin{bmatrix} 2 & 1 & 1 & 0 & 3 & 0 & 0 \\ 1 & 2 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 3 & 1 & 0 & 0 & 2 \end{bmatrix} \begin{array}{c} c_1 \\ c_2 \\ c_3 \end{array}
\end{array}
$$



- **Parity check c_1 tells us that $2v_1+v_2+v_3+3v_5 = 0$ in GF(4).**

Error Floor

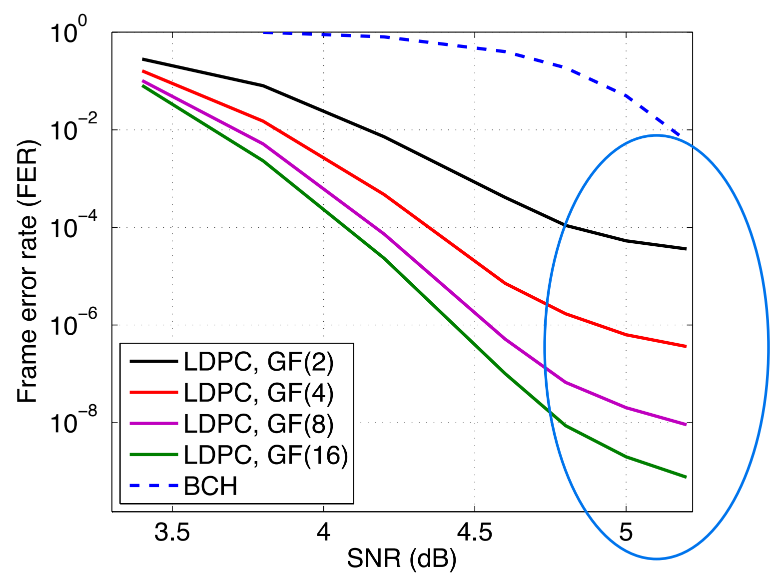
- **Error floor: a change in the slope of FER vs. SNR curve in high SNR region.**
- **In flash applications, the elimination of the error floor is critical**



Error Floor

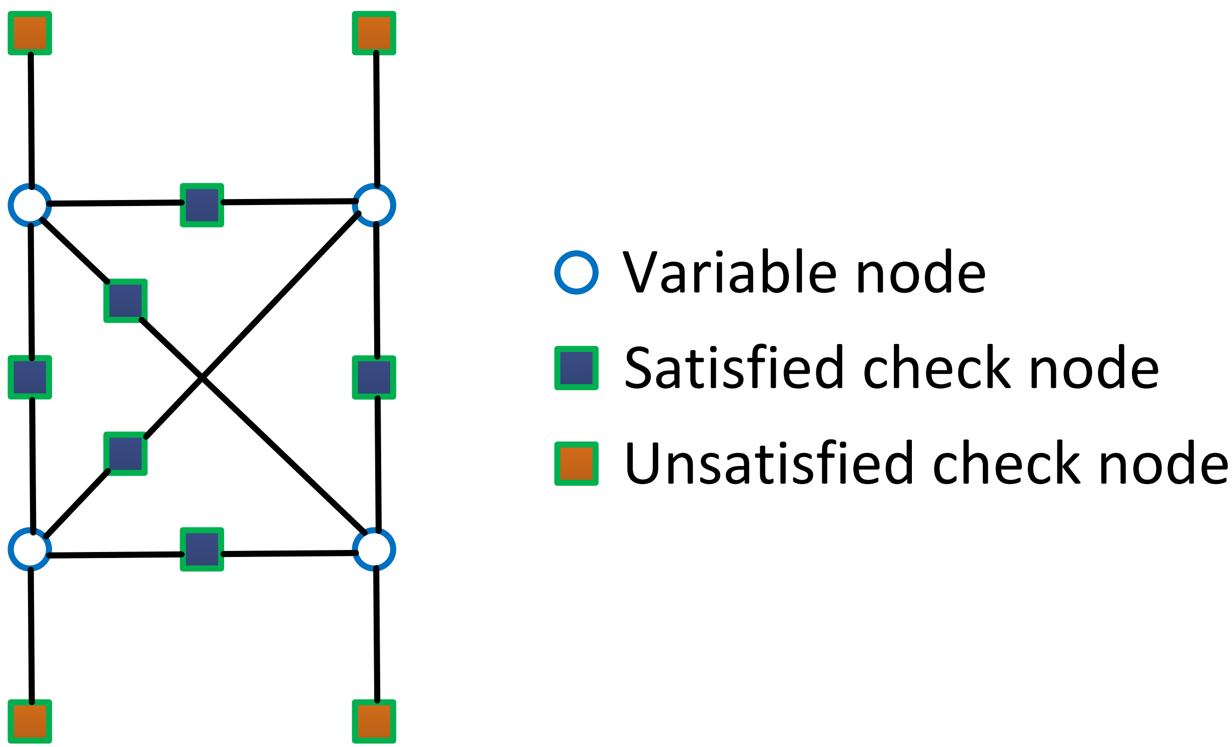
- **Error floor: a change in the slope of FER vs. SNR curve in high SNR region.**
- **In flash applications, the elimination of the error floor is critical**

- **In the error floor regime, certain codewords called absorbing sets dominate the performance**



Binary Absorbing Sets (for Binary LDPC codes)

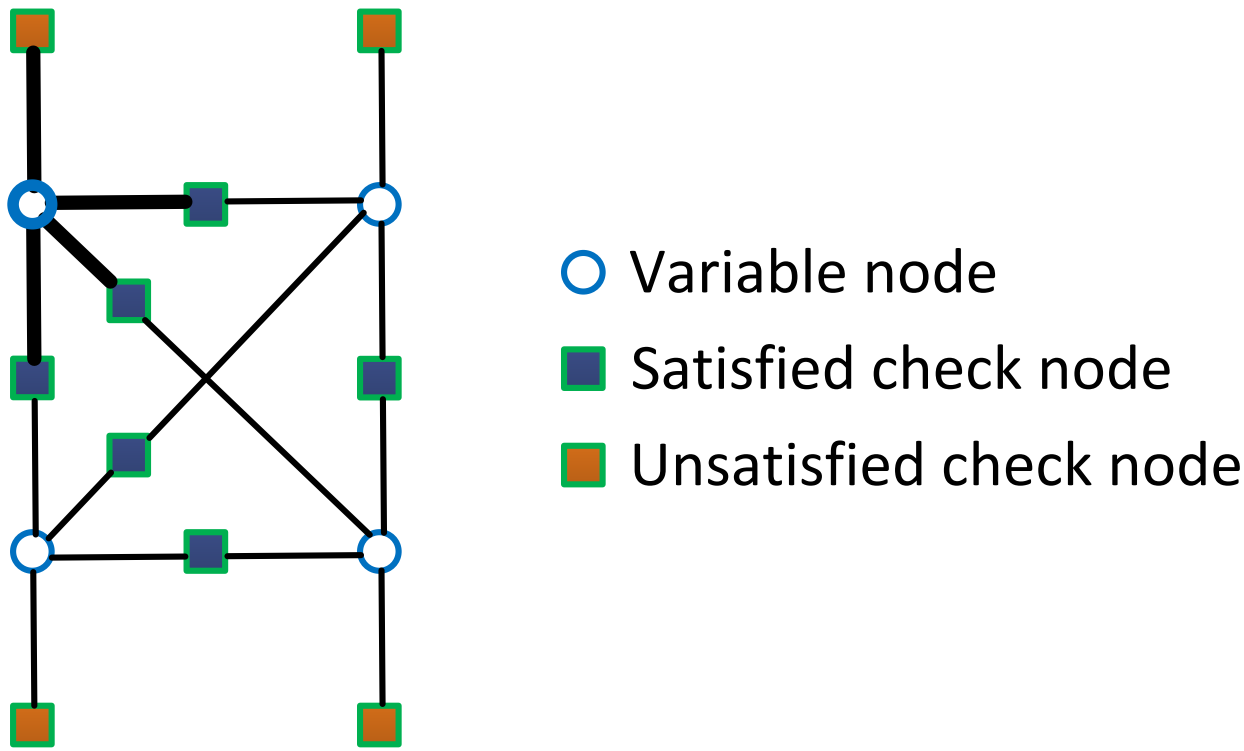
- **Example: (4,4) absorbing set**



- **Topological Condition: Each variable node is connected to strictly more satisfied than unsatisfied checks**
- **(a,b) absorbing set has a variable nodes and b unsatisfied checks**

Binary Absorbing Sets

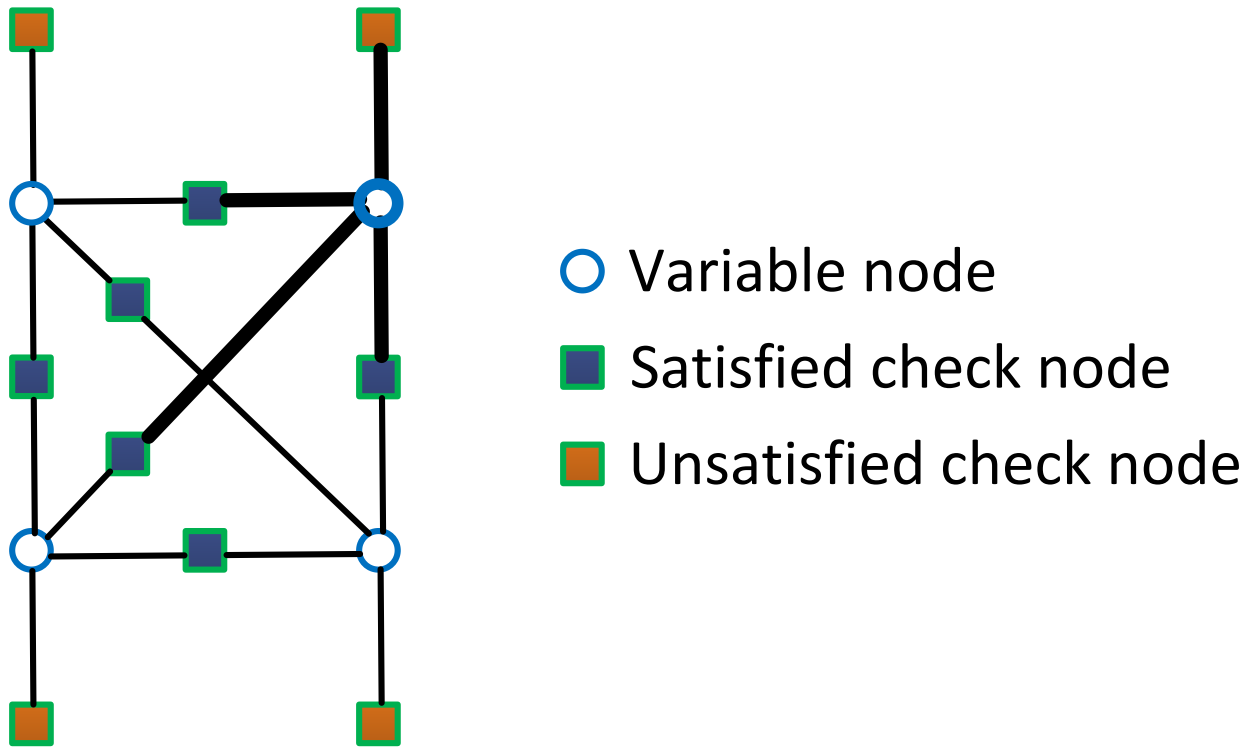
- **Example: (4,4) absorbing set**



- **Topological Condition: Each variable node is connected to strictly more satisfied than unsatisfied checks**
- **(a,b) absorbing set has a variable nodes and b unsatisfied checks**

Binary Absorbing Sets

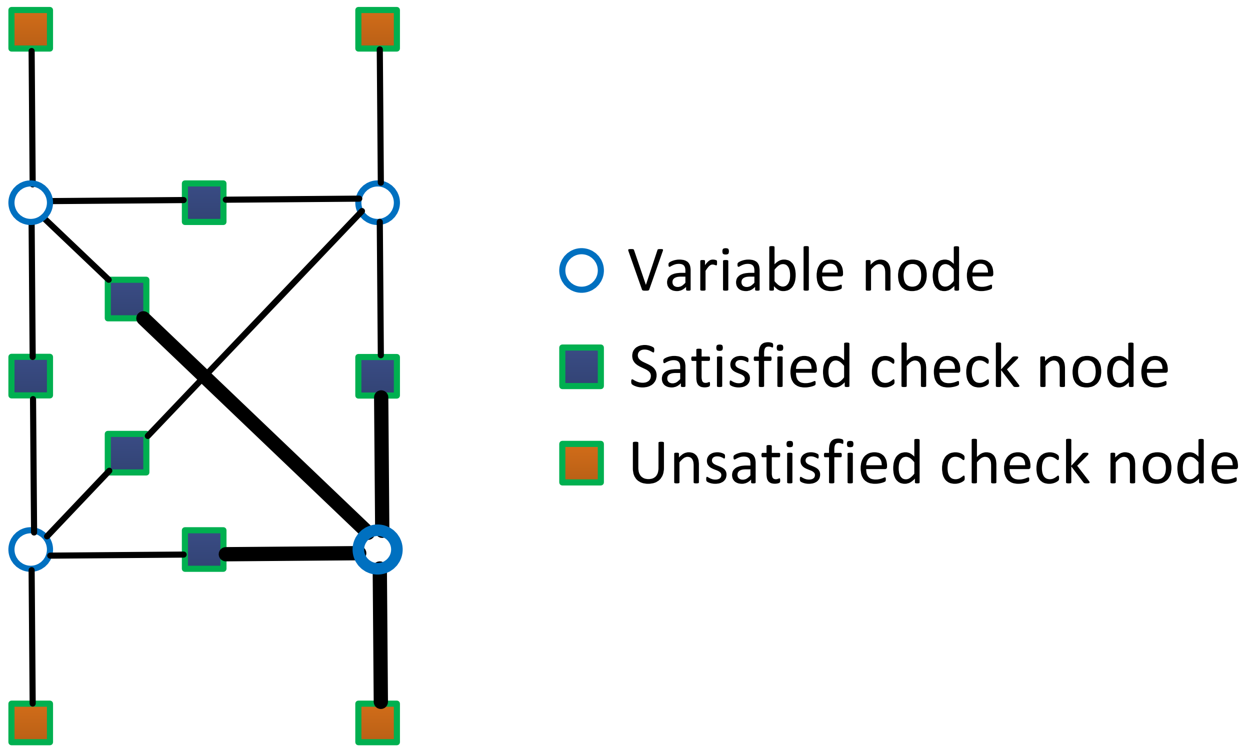
- **Example: (4,4) absorbing set**



- **Topological Condition: Each variable node is connected to strictly more satisfied than unsatisfied checks**
- **(a,b) absorbing set has a variable nodes and b unsatisfied checks**

Binary Absorbing Sets

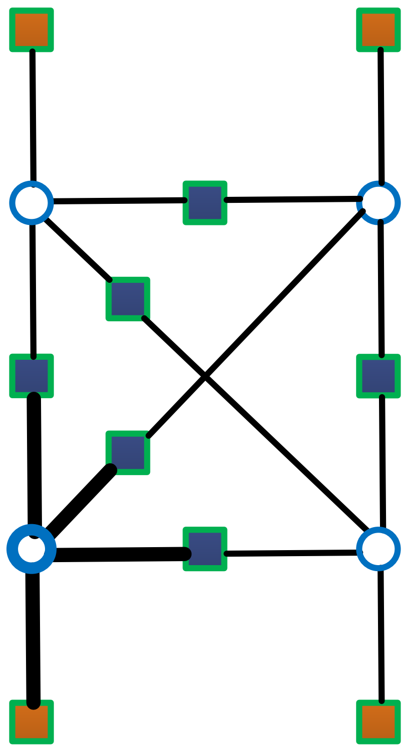
- **Example: (4,4) absorbing set**



- **Topological Condition: Each variable node is connected to strictly more satisfied than unsatisfied checks**
- **(a,b) absorbing set has a variable nodes and b unsatisfied checks**

Binary Absorbing Sets

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



- Variable node
- Satisfied check node
- Unsatisfied check node

- **Topological Condition: Each variable node is connected to strictly more satisfied than unsatisfied checks**
- **(a,b) absorbing set has a variable nodes and b unsatisfied checks**

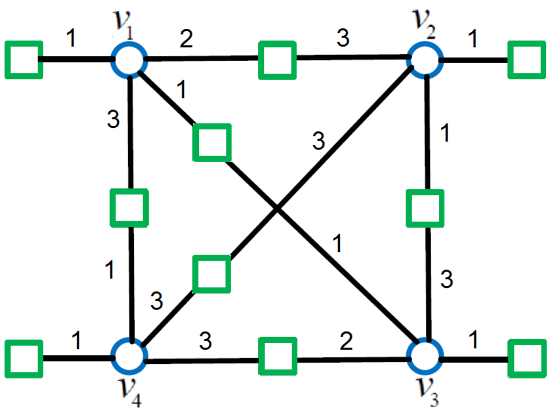
Non-Binary Absorbing Sets

- Certain subgraphs of the Tanner graph called **non-binary absorbing sets** cause the error floor.

Non-Binary Absorbing Sets

- Certain subgraphs of the Tanner graph called **non-binary absorbing sets** cause the error floor.

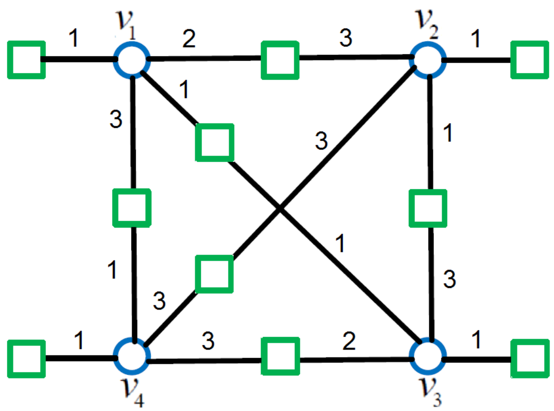
Example: (4,4) NB absorbing set over GF(4)



Non-Binary Absorbing Sets

- Certain subgraphs of the Tanner graph called **non-binary absorbing sets** cause the error floor.

Example: (4,4) NB absorbing set over GF(4)

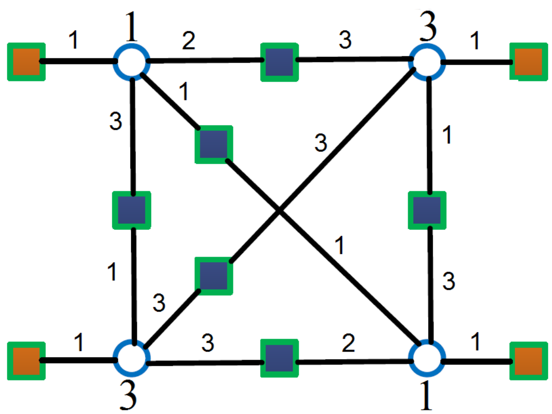


Consider: $(v_1, v_2, v_3, v_4) = (1, 3, 1, 3)$

Non-Binary Absorbing Sets

- Certain subgraphs of the Tanner graph called **non-binary absorbing sets** cause the error floor.

Example: (4,4) NB absorbing set over GF(4)



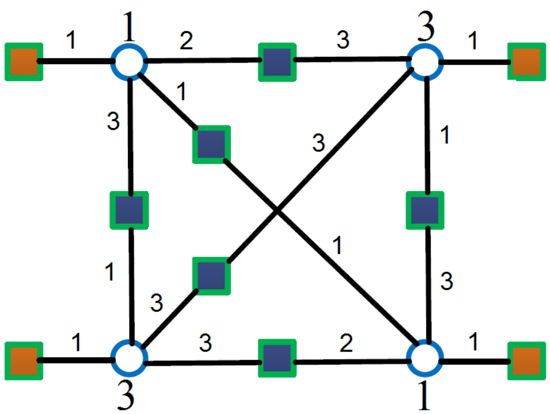
Consider: $(v_1, v_2, v_3, v_4) = (1, 3, 1, 3)$

- Variable node
- Satisfied check node
- Unsatisfied check node

Non-Binary Absorbing Sets

- Certain subgraphs of the Tanner graph called **non-binary absorbing sets** cause the error floor.

Example: (4,4) NB absorbing set over GF(4)



Consider: $(v_1, v_2, v_3, v_4) = (1, 3, 1, 3)$

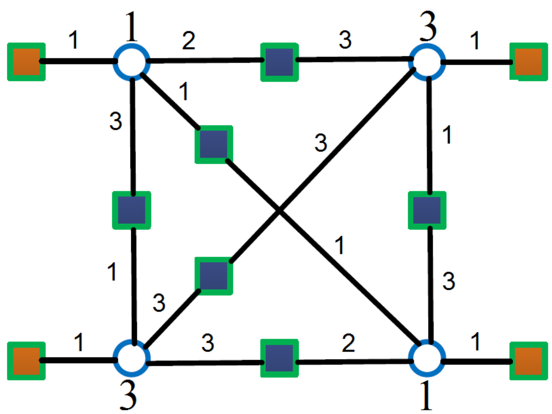
- Variable node
- Satisfied check node
- Unsatisfied check node

- Subgraphs corresponding to NB absorbing sets satisfy **two conditions:**

Non-Binary Absorbing Sets

- Certain subgraphs of the Tanner graph called **non-binary absorbing sets** cause the error floor.

Example: (4,4) NB absorbing set over GF(4)



Consider: $(v_1, v_2, v_3, v_4) = (1\ ,3\ ,1\ ,3\ )$

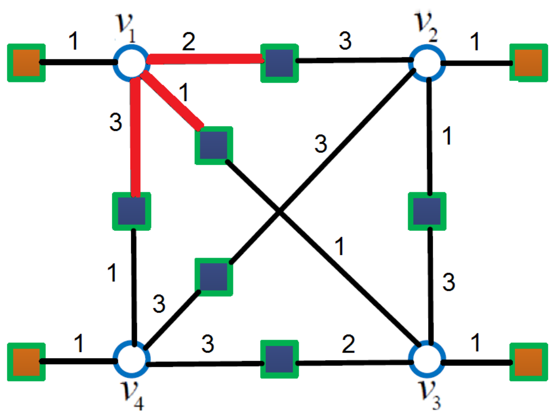
- Variable node
- Satisfied check node
- Unsatisfied check node

- Subgraphs corresponding to NB absorbing sets satisfy **two conditions:**
  - **Topological condition:** Each variable node is connected to **more satisfied check nodes than unsatisfied check nodes**.

Non-Binary Absorbing Sets

- Certain subgraphs of the Tanner graph called **non-binary absorbing sets** cause the error floor.

Example: (4,4) NB absorbing set over GF(4)



Consider: $(v_1, v_2, v_3, v_4) = (1, 3, 1, 3)$

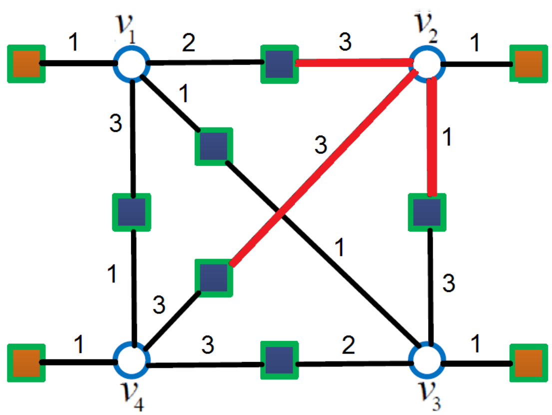
- Variable node
- Satisfied check node
- Unsatisfied check node

- Subgraphs corresponding to NB absorbing sets satisfy **two conditions:**
  - **Topological condition:** Each variable node is connected to **more satisfied check nodes than unsatisfied check nodes**.

Non-Binary Absorbing Sets

- Certain subgraphs of the Tanner graph called **non-binary absorbing sets** cause the error floor.

Example: (4,4) NB absorbing set over GF(4)



Consider: $(v_1, v_2, v_3, v_4) = (1, 3, 1, 3)$

○ Variable node

■ Satisfied check node

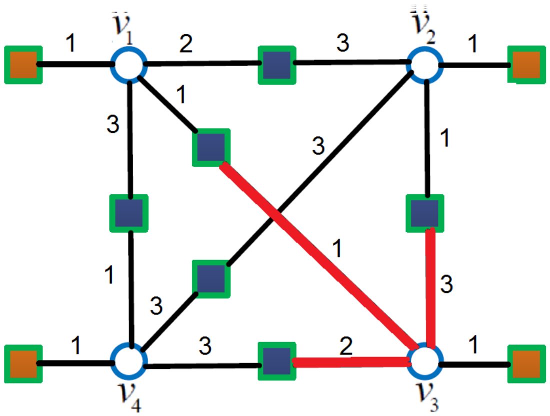
■ Unsatisfied check node

- Subgraphs corresponding to NB absorbing sets satisfy **two conditions:**
  – **Topological condition:** Each variable node is connected to **more satisfied check nodes than unsatisfied check nodes**.

Non-Binary Absorbing Sets

- Certain subgraphs of the Tanner graph called **non-binary absorbing sets** cause the error floor.

Example: (4,4) NB absorbing set over GF(4)



Consider: $(v_1, v_2, v_3, v_4) = (1, 3, 1, 3)$

○ Variable node

■ Satisfied check node

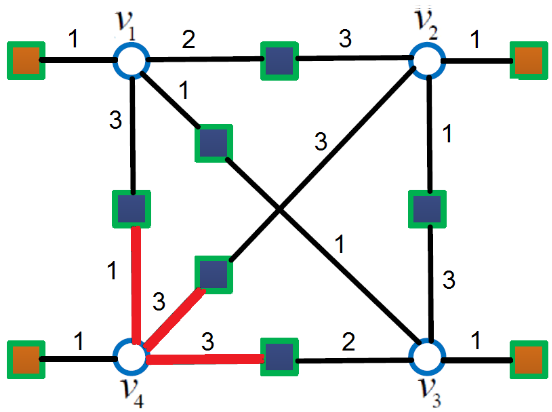
■ Unsatisfied check node

- Subgraphs corresponding to NB absorbing sets satisfy **two conditions:**
  - **Topological condition:** Each variable node is connected to **more satisfied check nodes than unsatisfied check nodes**.

Non-Binary Absorbing Sets

- Certain subgraphs of the Tanner graph called **non-binary absorbing sets** cause the error floor.

Example: (4,4) NB absorbing set over GF(4)



Consider: $(v_1, v_2, v_3, v_4) = (1, 3, 1, 3)$

○ Variable node

■ Satisfied check node

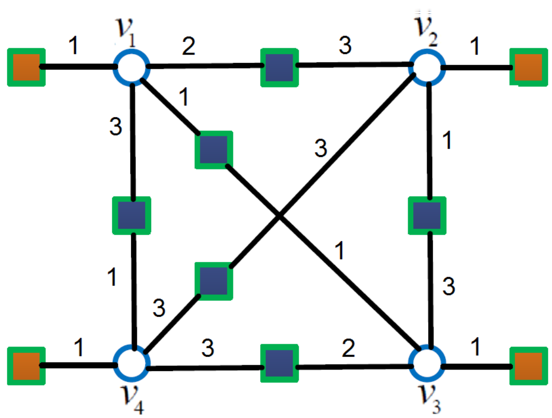
■ Unsatisfied check node

- Subgraphs corresponding to NB absorbing sets satisfy **two conditions:**
  - **Topological condition:** Each variable node is connected to **more satisfied check nodes than unsatisfied check nodes**.

Non-Binary Absorbing Sets

- Certain subgraphs of the Tanner graph called **non-binary absorbing sets** cause the error floor.

Example: (4,4) NB absorbing set over GF(4)



Consider: $(v_1, v_2, v_3, v_4) = (1, 3, 1, 3)$

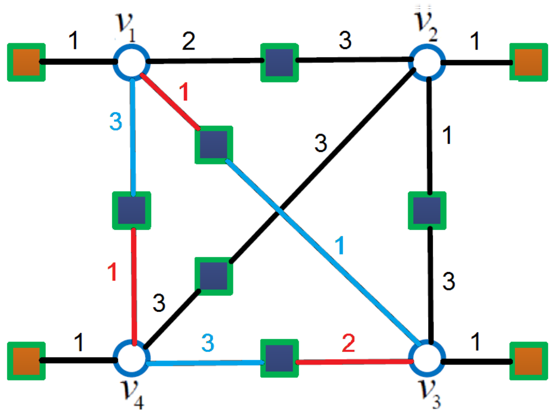
- Variable node
- Satisfied check node
- Unsatisfied check node

- Subgraphs corresponding to NB absorbing sets satisfy **two conditions:**
  - **Topological condition:** Each variable node is connected to **more satisfied check nodes than unsatisfied check nodes**.
  - **Weight condition:** The edges in the fundamental cycles satisfy weight conditions.

Non-Binary Absorbing Sets

- Certain subgraphs of the Tanner graph called **non-binary absorbing sets** cause the error floor.

Example: (4,4) NB absorbing set over GF(4)



Consider: $(v_1, v_2, v_3, v_4) = (1, 3, 1, 3)$

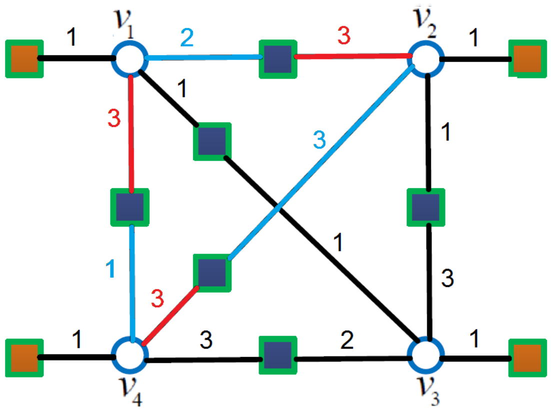
- Variable node
- Satisfied check node
- Unsatisfied check node

- Subgraphs corresponding to NB absorbing sets satisfy **two conditions:**
  - **Topological condition:** Each variable node is connected to **more satisfied check nodes than unsatisfied check nodes**.
  - **Weight condition:** The edges in the fundamental cycles satisfy weight conditions. $1 \times 2 \times 1 = 1 \times 3 \times 3$ over GF(4)

Non-Binary Absorbing Sets

- Certain subgraphs of the Tanner graph called **non-binary absorbing sets** cause the error floor.

Example: (4,4) NB absorbing set over GF(4)



Consider: $(v_1, v_2, v_3, v_4) = (1, 3, 1, 3)$

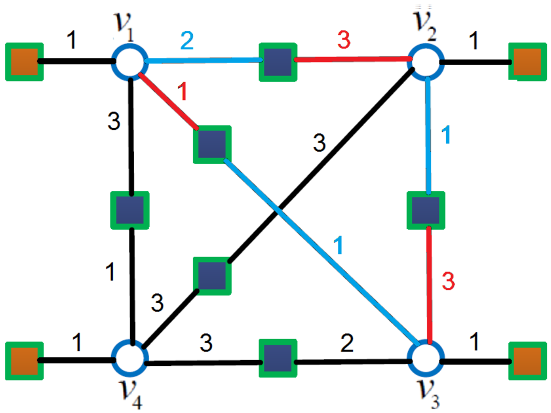
- Variable node
- Satisfied check node
- Unsatisfied check node

- Subgraphs corresponding to NB absorbing sets satisfy **two conditions:**
  - **Topological condition:** Each variable node is connected to **more satisfied check nodes than unsatisfied check nodes**.
  - **Weight condition:** The edges in the fundamental cycles satisfy weight conditions. $1 \times 2 \times 1 = 1 \times 3 \times 3$ over GF(4)
  $3 \times 3 \times 3 = 2 \times 3 \times 1$ over GF(4)

Non-Binary Absorbing Sets

- Certain subgraphs of the Tanner graph called **non-binary absorbing sets** cause the error floor.

Example: (4,4) NB absorbing set over GF(4)



Consider: $(v_1, v_2, v_3, v_4) = (1, 3, 1, 3)$

○ Variable node

■ Satisfied check node

■ Unsatisfied check node

- Subgraphs corresponding to NB absorbing sets satisfy **two conditions:**
  - **Topological condition:** Each variable node is connected to **more satisfied check nodes than unsatisfied check nodes**.
  - **Weight condition:** The edges in the fundamental cycles satisfy weight conditions.

$$1 \times 2 \times 1 = 1 \times 3 \times 3 \quad \text{over GF(4)}$$
$$3 \times 3 \times 3 = 2 \times 3 \times 1 \quad \text{over GF(4)}$$
$$3 \times 3 \times 1 = 2 \times 1 \times 1 \quad \text{over GF(4)}$$

Elementary Absorbing Sets

- **Definition:**
  - A subset of the node set is an elementary absorbing set if all neighboring satisfied checks have degree 2 and all neighboring unsatisfied checks have degree 1.
- **Lemma:**
  - An elementary non-binary absorbing set satisfies the following conditions.
    - (Topological condition) Unlabeled subgraph is an elementary binary absorbing set.
    - (Weight conditions) All of its cycles satisfy

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

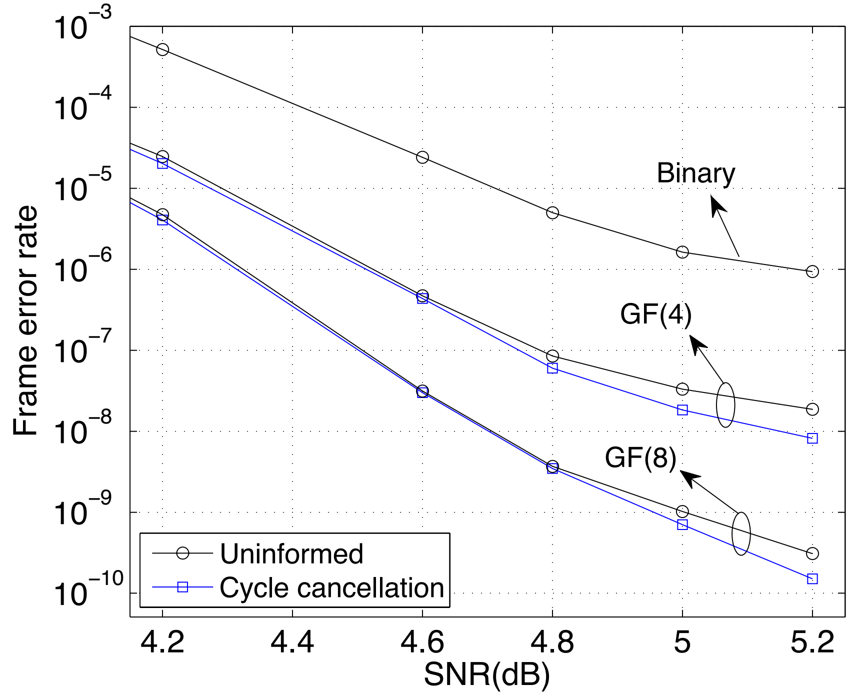
where w_i's, $1 \leq i \leq 2p$, are edge weights and $2p$ is the length of the cycle. Adjacent edges are labeled one after another in a clockwise fashion.

Algorithm to Eliminate Detrimental Absorbing Sets

1. Identify a list (W) of problematic absorbing sets.
2. Find all (a, b) binary elementary absorbing sets in the unlabeled bipartite graph as the candidates for (a, b) non-binary absorbing sets.
3. For each candidate, check if the weight conditions are satisfied.
4. If yes, we change the weight of an edge to another non-zero element of GF(q).
5. This process continues until all (a, b) non-binary absorbing sets in W are eliminated or no more absorbing sets can be eliminated.

Performance Improvement Using the Proposed Algorithm – Regular Codes

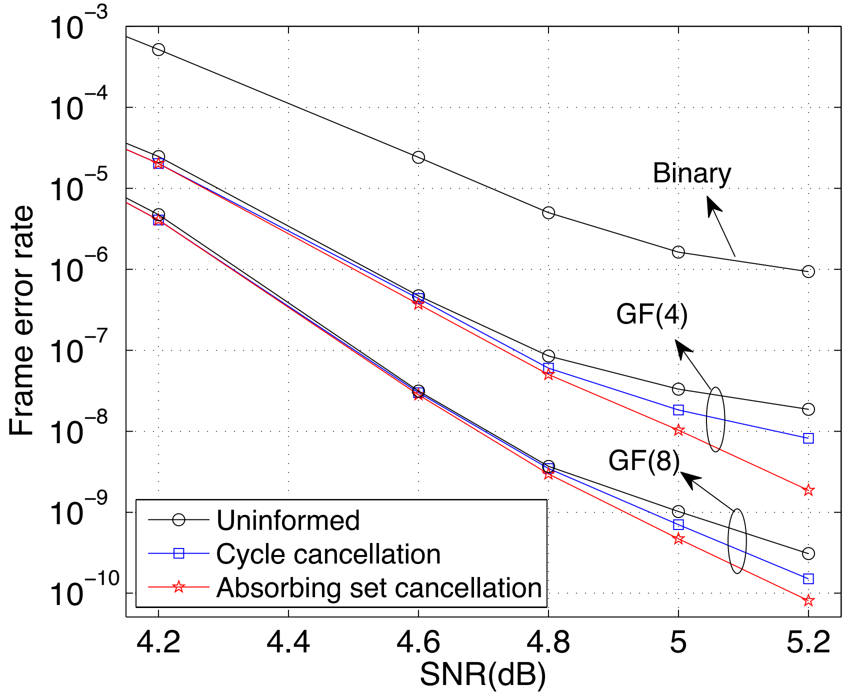
- **Code length ~ 2800 bits, rate ~ 0.87, column-weight 4 and performing the QSPA-FFT decoder.**



[R1] C. Poulliat et al. "Design of regular (2, dc)–LDPC codes over GF(q) using their binary images," IEEE Trans. on Commun., vol. 56, no. 10, pp. 1626-1635, Oct. 2008.

Performance Improvement Using the Proposed Algorithm – Regular Codes

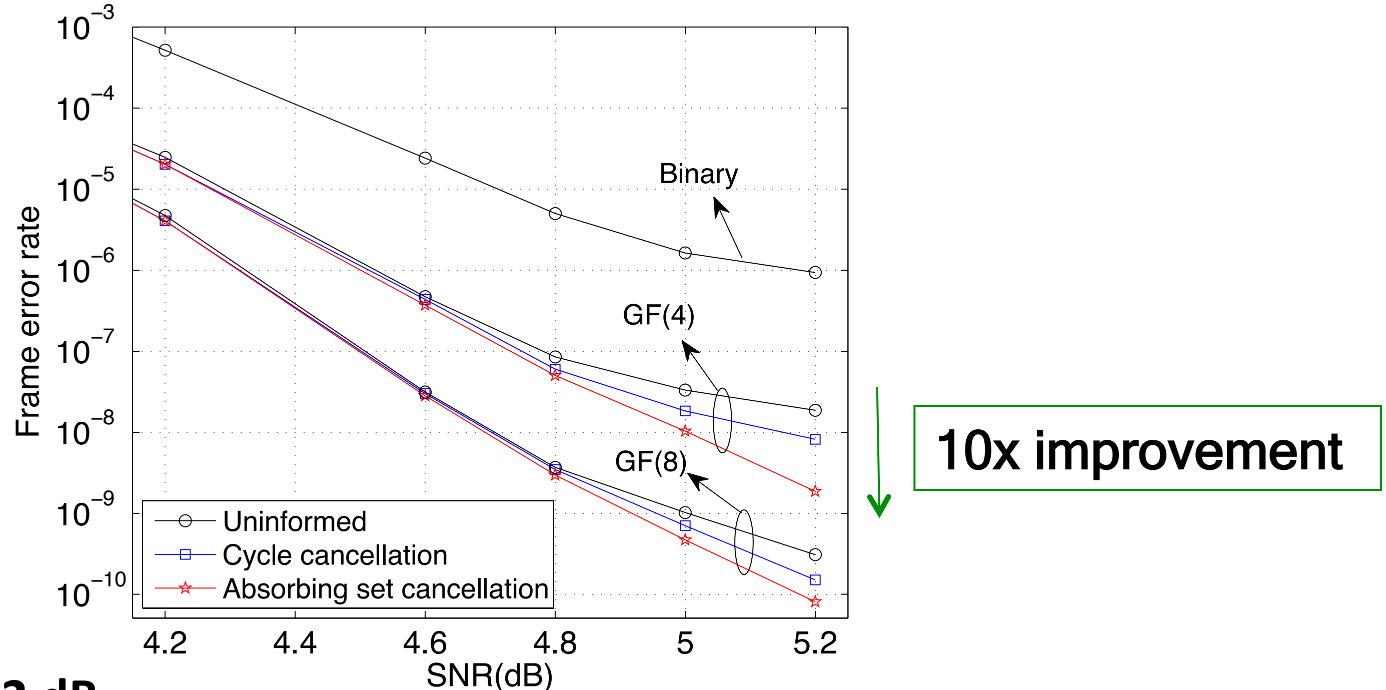
- **Code length ~ 2800 bits, rate ~ 0.87, column-weight 4 and performing the QSPA-FFT decoder.**



[R1] C. Poulliat et al. "Design of regular (2, dc)–LDPC codes over GF(q) using their binary images," IEEE Trans. on Commun., vol. 56, no. 10, pp. 1626-1635, Oct. 2008.

Performance Improvement Using the Proposed Algorithm – Regular Codes

- **Code length ~ 2800 bits, rate ~ 0.87, column-weight 4 and performing the QSPA-FFT decoder.**



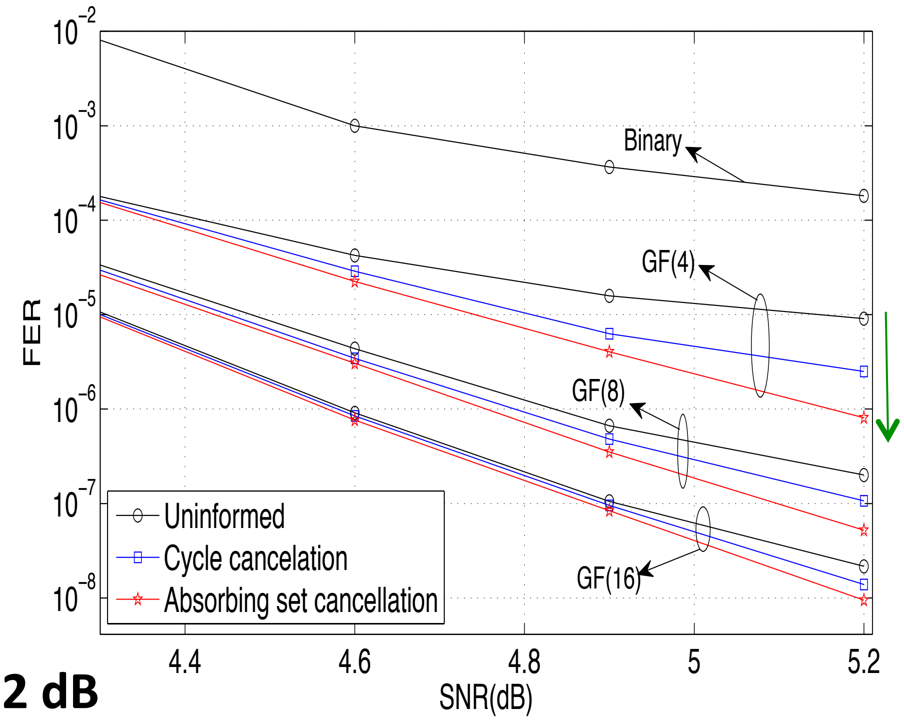
For GF(4) at 5.2 dB

Error Type	(4, 4)	(5, 0)	(5, 2)	(6, 2)	(6, 6)	(7, 4)	(8, 2)	other
Uninformed	32	9	14	7	18	9	9	15
Cycle cancel.	0	0	0	0	12	5	5	12
AS cancel.	0	0	0	0	0	0	0	13

[R1] C. Poulliat et al. "Design of regular (2, dc)–LDPC codes over GF(q) using their binary images," IEEE Trans. on Commun., vol. 56, no. 10, pp. 1626-1635, Oct. 2008.

Performance Improvement Using the Proposed Algorithm – Regular Codes

- **Code length ~ 2800 bits, rate ~ 0.87, column-weight 4 and performing the min-sum decoder.**



10x improvement

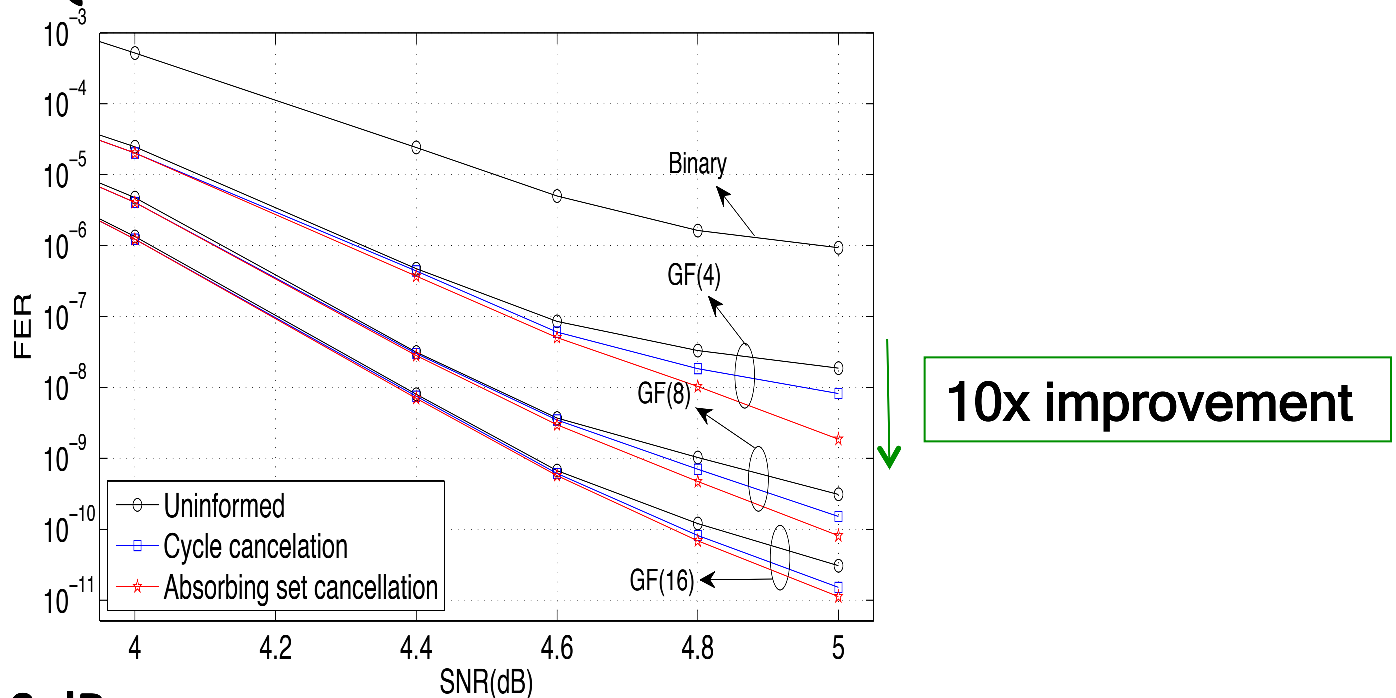
For GF(4) at 5.2 dB

Error Type	(4, 4)	(5, 0)	(5, 2)	(6, 2)	(6, 4)	(6, 6)	(7, 4)	(8, 2)	other
Uninformed	43	9	14	8	8	12	7	6	23
Cycle cancel.	0	0	0	0	0	9	5	4	14
AS cancel.	0	0	0	0	0	0	0	0	15

[R1] C. Poulliat et al. "Design of regular (2, dc)–LDPC codes over GF(q) using their binary images," IEEE Trans. on Commun., vol. 56, no. 10, pp. 1626-1635, Oct. 2008.

Performance Improvement Using the Proposed Algorithm – Irregular Codes

- Code length ~ 2000 bits, rate ~ 0.85, column-weights 4 and 5, and performing the QSPA-FFT decoder.



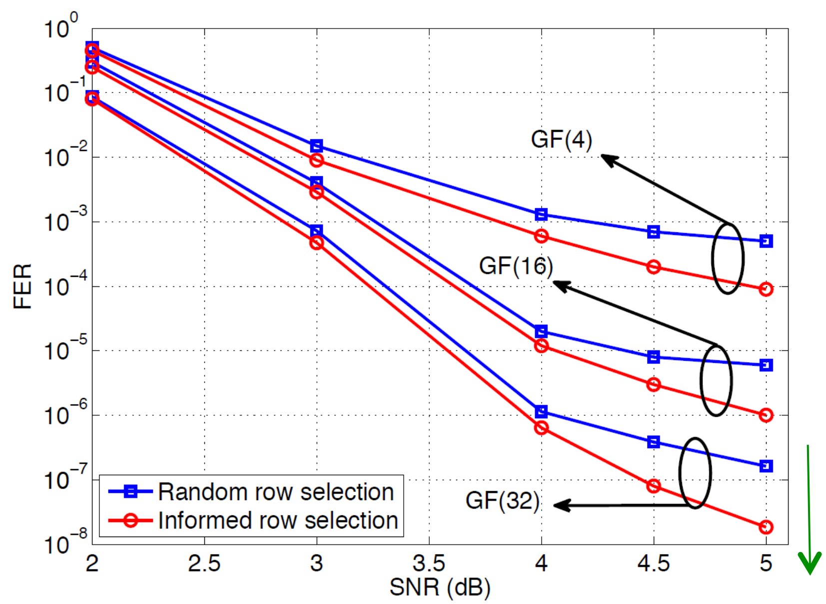
For GF(4) at 5.0 dB

Error Type	size 4 AS	size 5 AS	size 6 AS	size 7 AS	size 8 AS	other
Uninformed	42	31	16	18	8	31
Cycle cancel.	0	0	0	12	5	24
AS cancel.	0	0	0	0	0	25

[R1] C. Poulliat et al. "Design of regular (2, dc)–LDPC codes over GF(q) using their binary images," IEEE Trans. on Commun., vol. 56, no. 10, pp. 1626-1635, Oct. 2008.

Non-binary quasi-cyclic codes without small absorbing sets

- **Quasi-cyclic codes with code length ~1200 bits, rate ~0.8, column weight 5.**



10x improvement

For GF(16) at 5.0 dB

Error Type	$(4,8)$	$(5,9)$	$(6,8)$	$(6,10)$	$(7,9)$	$(7,11)$	$(8,6)$	other
Random	29	0	37	0	22	0	9	0
Informed	0	0	2	5	4	4	8	0

[R2] B. Zhou et al. "High-performance non-binary LDPC codes on Euclidean geometries," IEEE Trans. on Commun., vol. 57, no. 5, pp. 1298-1311, May 2009.

In summary

- **Non-binary LDPC codes offer significant performance improvement over both binary LDPC codes and over BCH codes**

For multilevel flash, well-designed non-binary LDPC codes show a lot of promise

In summary

- **Non-binary LDPC codes offer significant performance improvement over both binary LDPC codes and over BCH codes**

- **Elimination of error floor is critical for high-reliability applications**

- **Non-binary absorbing sets capture decoding errors**
  - We have developed algorithmic and combinatorial techniques for elimination of non-binary absorbing sets
  - Techniques are applicable for a broad range of codes and decoders and offer > 10x improvement over best NB LDPC codes

For multilevel flash, well-designed non-binary LDPC codes show a lot of promise

References

[1] B. Amiri, J. Kliewer, and L. Dolecek, "Analysis and Enumeration of Absorbing Sets for Non-Binary Graph-Based Codes," *IEEE Transactions on Communications*, 2013.

[2] L. Dolecek, D. Divsalar, Y. Sun and B. Amiri, "Non-Binary Protograph-Based LDPC Codes: Enumerators, Analysis, and Designs," *IEEE Transactions on Information Theory*, 2014.

[3] B. Amiri, A. Flores, and L. Dolecek, "Design of Non-Binary Quasi-Cyclic LDPC Codes by Absorbing Set Removal," *IEEE Information Theory Workshop*, 2014.

UCLA Center on Development of Emerging Data Storage Systems (CoDESS)

- CoDESS is founded in 2013.
- Webpage: **http://www.uclacodess.org/**
- Mission:
  - Push the frontiers in emerging data storage systems through integrated research program.
  - Create highly--trained workforce of graduate students and post-doctoral researchers.
- For more information, please email to
  - Prof. Lara Dolecek (dolecek@ee.ucla.edu)
  - Prof. Richard Wesel (wesel@ee.ucla.edu)