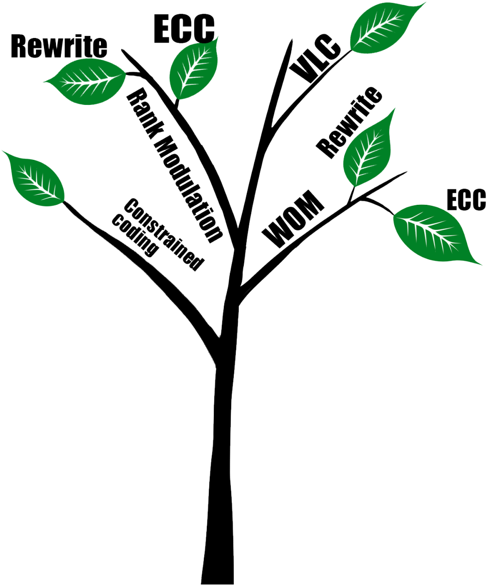
Making Error Correcting Codes Work for Flash Memory

Part III: New Coding Methods

Anxiao (Andrew) Jiang

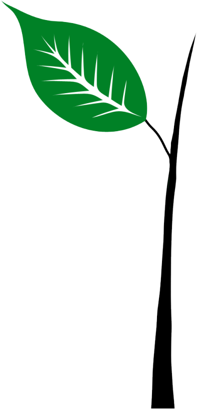
Department of Computer Science and Engineering
Texas A&M University

Tutorial at Flash Memory Summit, August 4, 2014

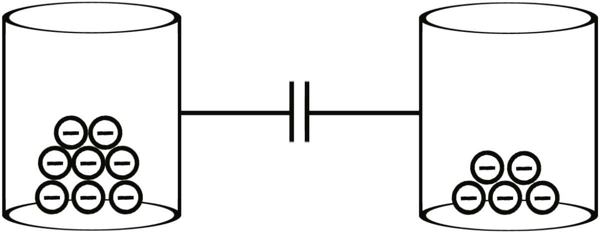


Acknowledgment to: Wenyi Zhu (for artistic illustrations)

Constrained coding



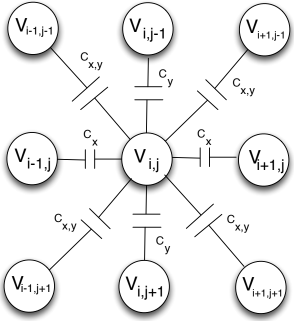
Constrained Coding



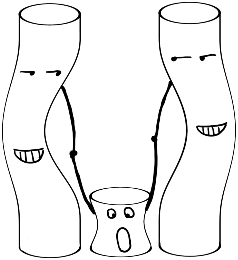
Constrained coding for inter-cell interference

Inter-cell interference in flash memory:
The V_{th} shift of middle cell caused by shifting of neighboring cells is

$$\begin{aligned}\Delta V_{i,j} = \;& C_x(\Delta V_{i-1,j} + \Delta V_{i+1,j}) + C_y(\Delta V_{i,j-1} + \Delta V_{i,j+1}) \\ & + C_{x,y}(\Delta V_{i-1,j-1} + \Delta V_{i+1,j-1} + \Delta V_{i-1,j+1} + \Delta V_{i+1,j+1})\end{aligned}$$



One constraint to set for q-level cells: The difference between adjacent levels cannot be too large.
A concrete example: Avoid $(q-1)0(q-1)$ pattern for adjacent cell levels.



Minghai Qin, Eitan Yaakobi, and Paul Siegel, "Constrained codes that mitigate intercell interference in read/write cycles for flash memories," in JSAC Special Issue, May 2014.

Further reading

- F. Sala and L. Dolecek, "Constrained rank modulation schemes," in ITW 2013.
- K. A. S. Immink, "Coding schemes for multi-level channels with unknown gain and/or offset," in ISIT 2013.
- K. A. S. Immink and J. H. Weber, "Minimum Pearson distance detection for multi-level channels with gain and/or offset mismatch," draft 2014.
- A. Berman and Y. Birk, "Error correction scheme for constrained inter-cell coupling in flash memory," NVMW 2011.
- G. Dong, S. Li and T. Zhang, "Using data post-compensation and pre-distortion to tolerate cell-to-cell interference in MLC NAND flash memory," in *IEEE Trans. Circuits and Systems I*, 2010.
- E. Ordentlich, G. Ribeiro, R. M. Roth, G. Seroussi, and P. O. Vontobel, "Coding for limiting current in memristor crossbar memories," NVMW 2011.
- Y. Cassuto, S. Kvatinsky and E. Yaakobi, "Sneak-path constraints in memristor crossbar arrays," in ISIT 2013.


WOM
Constrained coding


Rewrite
Constrained coding
WOM

Rewriting Data in Flash Memories [1][2][3]

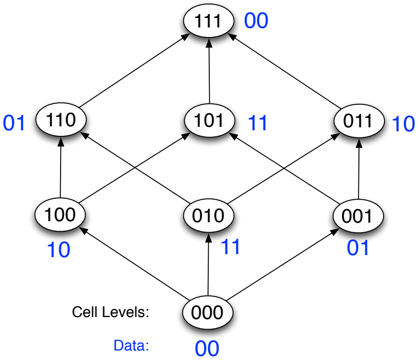
Basic concepts:

- Rewriting: Change the value of the stored data.
- Requirement: The cell levels can only increase, not decrease, in order to avoid block erasures.
- Objective: Maximize the number of times the data are rewritten, or maximize the summation of the code rates over the multiple rewrites.

Papers in ISIT 2007:
[1] A. Jiang, V. Bohossian and J. Bruck, "Floating codes for joint information storage in write asymmetric memories," in Proc. ISIT, pp. 1166-1170, 2007.
[2] V. Bohossian, A. Jiang and J. Bruck, "Buffer coding for asymmetric multi-level memory," in Proc. ISIT, pp. 1186-1190, 2007.
[3] A. Jiang, "On the generalization of error-correcting WOM codes," in Proc. ISIT, pp. 1391-1395, 2007.

Write Once Memory (WOM) [1]

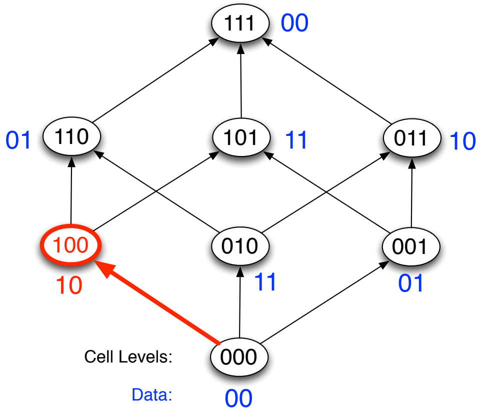
Example: Store 2 bits in 3 SLCs. Write the 2-bit data twice.



[1] R. L. Rivest and A. Shamir, "How to reuse a 'write-once' memory," in Information and Control, vol. 55, pp. 1-19, 1982.

Write Once Memory (WOM)

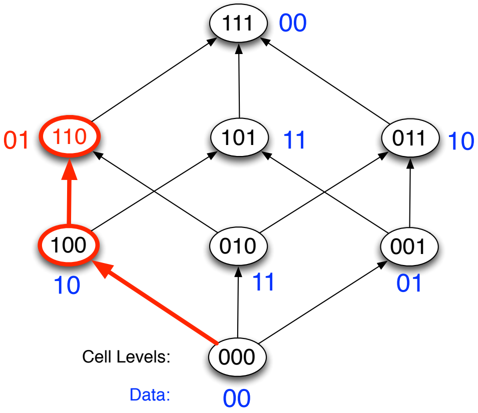
Example: Store 2 bits in 3 SLCs. Write the 2-bit data twice.



1st write: 10

Write Once Memory (WOM)

Example: Store 2 bits in 3 SLCs. Write the 2-bit data twice.

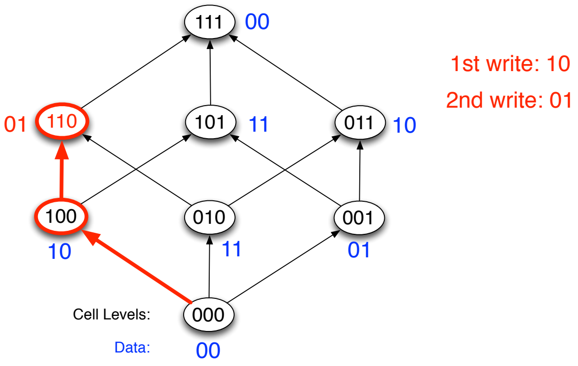


1st write: 10

2nd write: 01

Write Once Memory (WOM)

Example: Store 2 bits in 3 SLCs. Write the 2-bit data twice.



Sum rate: $\frac{2}{3} + \frac{2}{3} = 1.33$

Capacity of WOM [1][2]

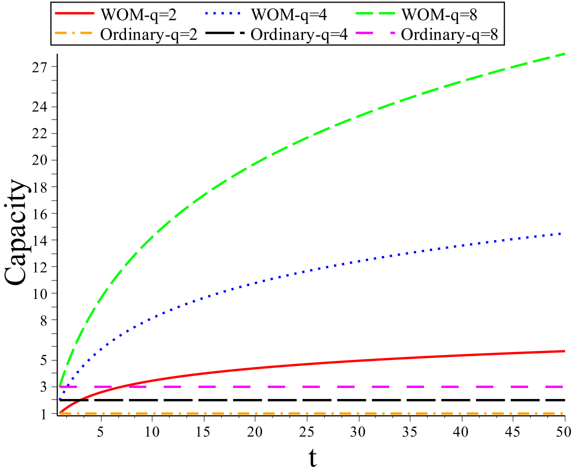
For WOM of q-level cells and t rewrites, the capacity (maximum achievable sum rate) is

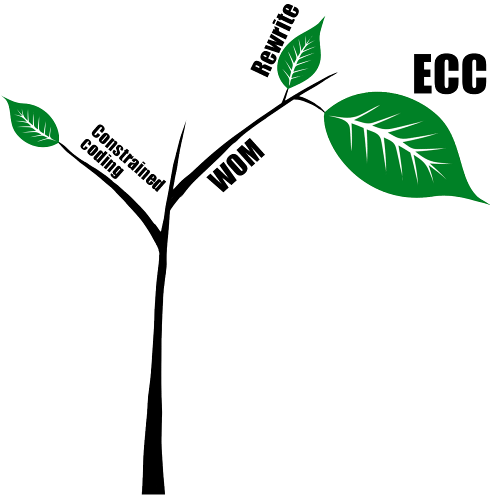
$$\log_2 \binom{t+q-1}{q-1}.$$

bits per cell.

[1] C. Heegard, On the capacity of permanent memory, in *IEEE Trans. Information Theory*, vol. IT-31, pp. 34-42, 1985.
[2] F. Fu and A. J. Han Vinck, On the capacity of generalized write-once memory with state transitions described by an arbitrary directed acyclic graph, in *IEEE Trans. Information Theory*, vol. 45, no. 1, pp. 308-313, 1999.

Capacity of WOM



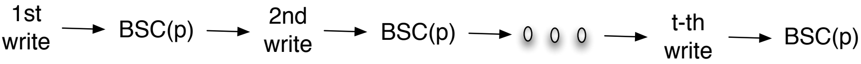

Rewrite
ECC
Constrained coding
WOM

For **Rewriting** to be used in flash memories, it is **CRITICAL** to combine it with **Error-Correcting Codes**.

A joint coding scheme for rewriting and error correction, which can correct a substantial number of errors and supports any number of rewrites.

A. Jiang, Yue Li, Eyal En Gad, Michael Langberg, and Jehoshua Bruck, "Joint rewriting and error correction in write-once memories," in ISIT 2013.

Model of rewriting and noise:



Lower bound to achievable sum-rate (for WOM):

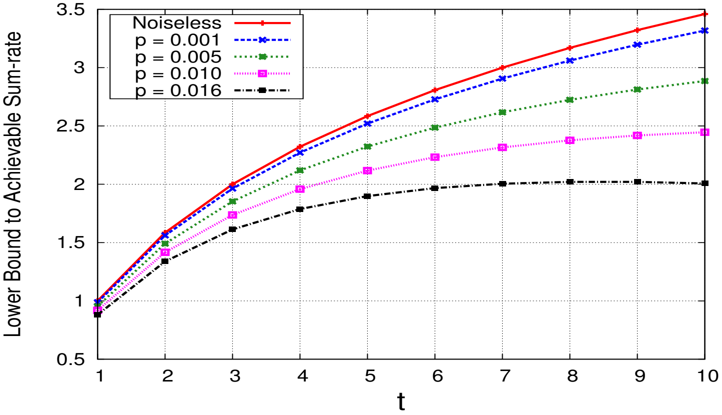
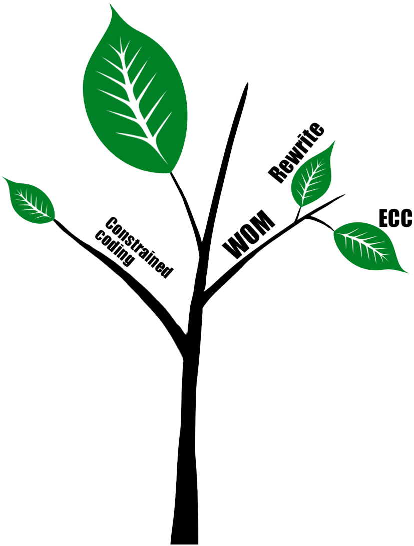


Fig. 6. Lower bound to achievable sum-rates for different error probability p.

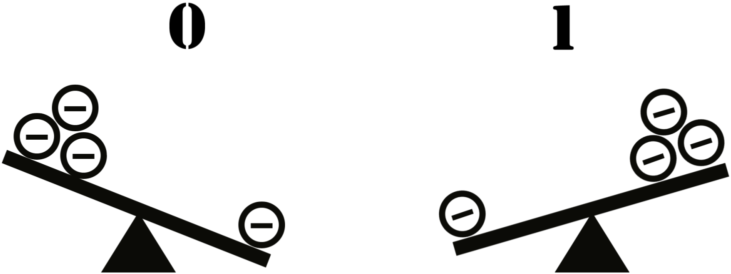
Further reading

- G. D. Cohen, P. Godlewski, and F. Merkx, "Linear binary code for write-once memories," in *IEEE Trans. Information Theory*, vol. IT-32, pp. 697-700, 1986.
- Y. Wu, Low complexity codes for writing write-once memory twice, 2010.
- E. Yaakobi, S. Kayser, P. H. Siegel, A. Vardy and J. K. Wolf, Codes for write-once memories, 2012.
- R. Gabrys, E. Yaakobi, L. Dolecek, P. H. Siegel, A. Vardy and J. Wolf, Non-binary WOM-codes for multilevel flash memories, 2011.
- A. Shpilka, Capacity achieving two-write WOM codes, 2012.
- E. Yaakobi and A. Shpilka, High sum-rate three-write and non-binary WOM codes, 2012.
- A. Shpilka, Capacity achieving multiwrite WOM codes, 2012.

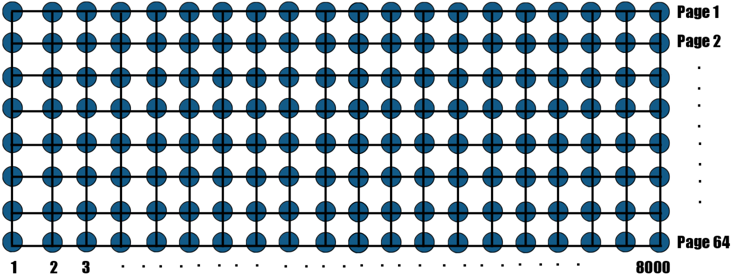
Rank Modulation



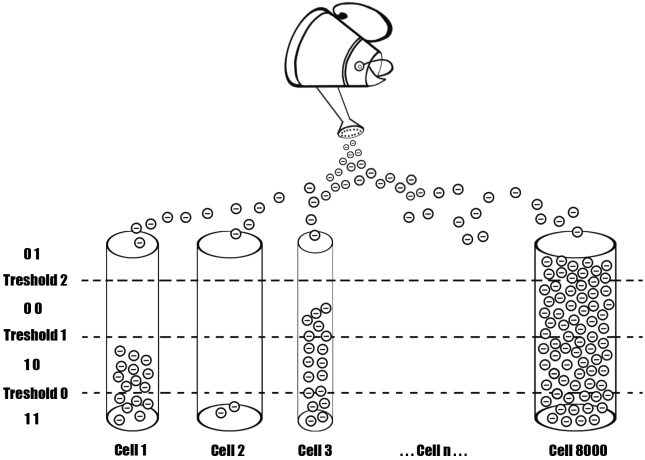
Rank Modulation



Parallel cell programming for MLC



Challenges of parallel cell programming for MLC



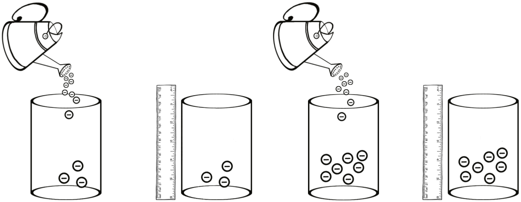
Muti-level cell (MLC): Parallel programming, common thresholds, heterogeneous cells, random process of charge injection, over-injection of charge, disturbs and inter-cell interference, block erasure, difficulty in adjusting threshold voltages, very careful repeated charge injection and measuring.

Challenges of parallel cell programming for MLC

Dilemma among:

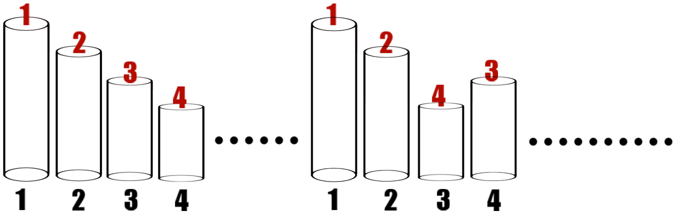
- Capacity
- Speed
- Reliability and endurance

Due to: Inflexibility in adjusting cell levels.



Definition (Rank Modulation)

Use the relative order of cell levels to represent data.

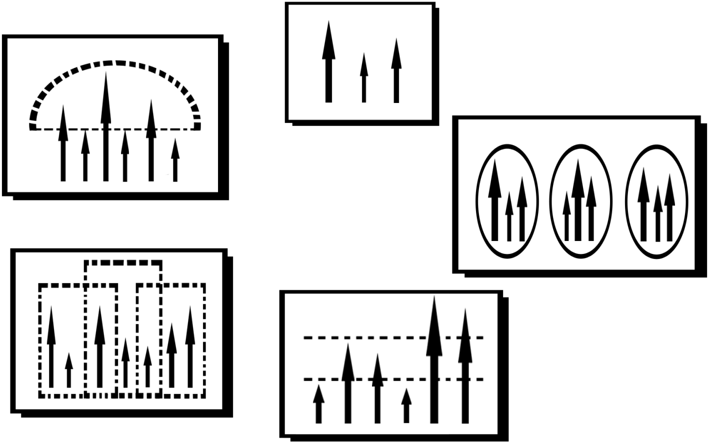


A. Jiang, R. Mateescu, M. Schwartz and J. Bruck, "Rank modulation for flash memories," in ISIT 2008.

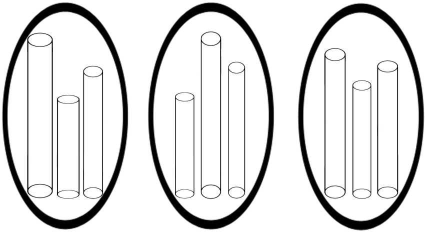
Some advantages of rank modulation:

1. Flexibility in adjusting relative cells levels, even though we can only increase cell levels;
2. Tolerance for charge leakage / cell level drifting;
3. Enable memory scrubbing without block erasure.

2. Extended models of rank modulation



- Extension: Rank modulation with multiple permutations

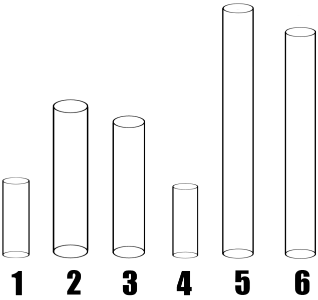


Some advantages: (1) Enable the building of long codes; (2) Cells in different permutations can have very close cell levels.

F. Zhang, H. Pfister and A. Jiang, "LDPC codes for rank modulation in flash memories," in ISIT 2010.

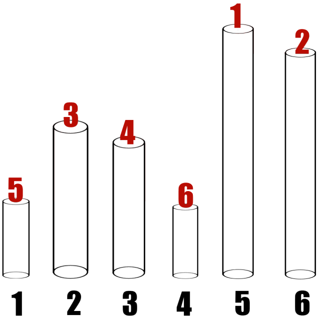
- Extension: Rank modulation with multi-set permutation

Example: A group of $n = 6$ cells

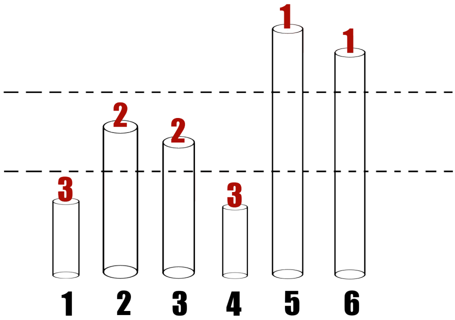


Some advantages: Similar to multiple permutations, but more suitable if cells can be programmed accurately.

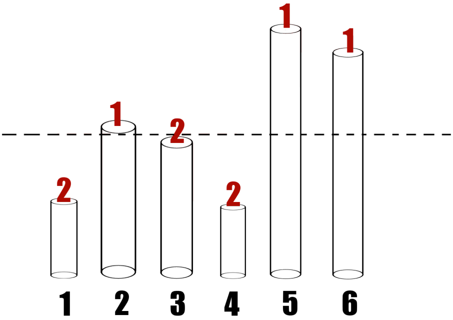
Example: Every rank has one cell



Example: Every rank has two cells



Example: Every rank has three cells



- Extension: Bounded rank modulation

  Z. Wang, A. Jiang and J. Bruck, "On the capacity of bounded rank modulation for flash memories," in ISIT 2009.

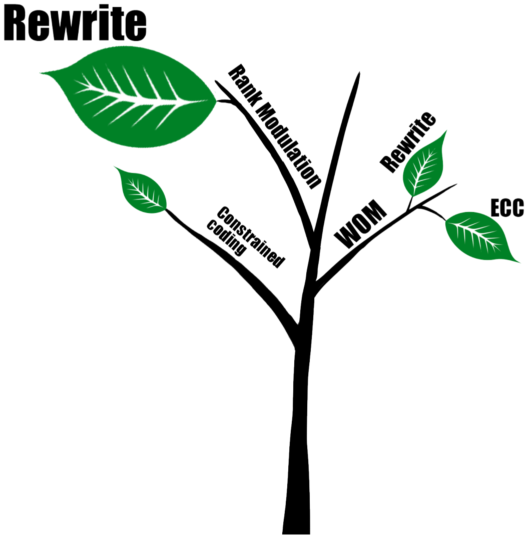
- Extension: Local rank modulation

  M. Schwartz, "Constant-weight Gray codes for local rank modulation," in ISIT 2010.

- Extension: Partial rank modulation:

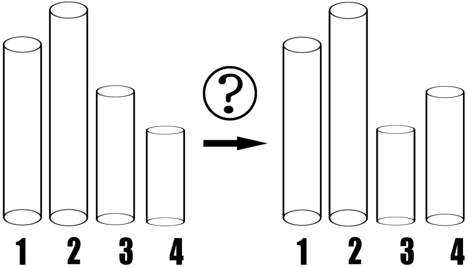
  Z. Wang and J. Bruck, "Partial rank modulation for flash memories," in ISIT 2010.

Some advantages: Faster read, and/or enabling long codewords.

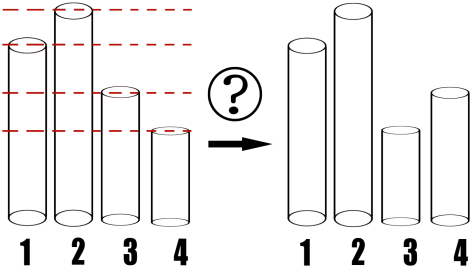

Rewrite
Rank Modulation
Constrained coding
WOM
Rewrite
ECC

Definition (Rewrite)

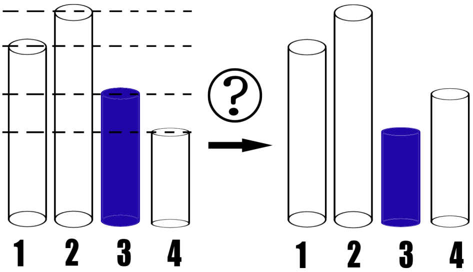
Change data by changing the permutation – by moving cell levels up.



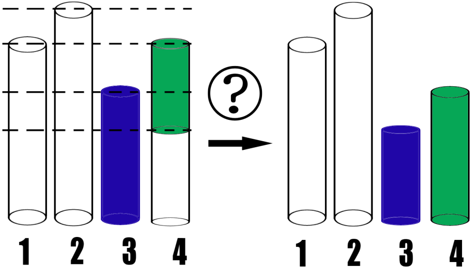
Virtual levels to help us estimate rewriting cost (increase in cell levels).



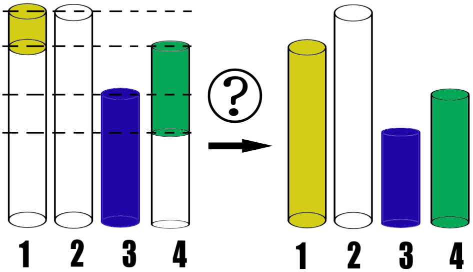
Get the permutation right from low to high.



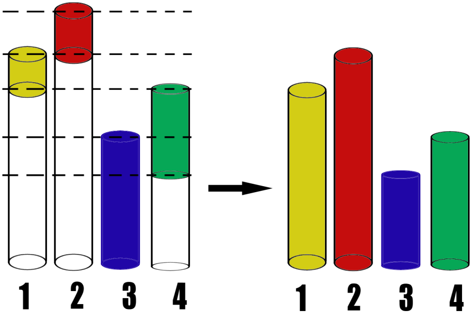
Get the permutation right from low to high.



Get the permutation right from low to high.

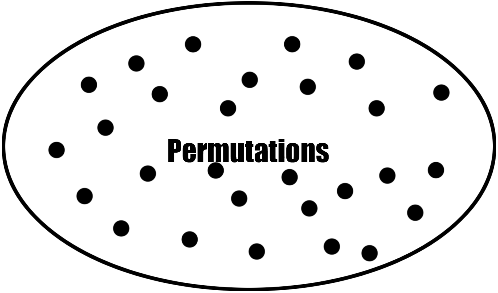


Rewriting cost: **1**.

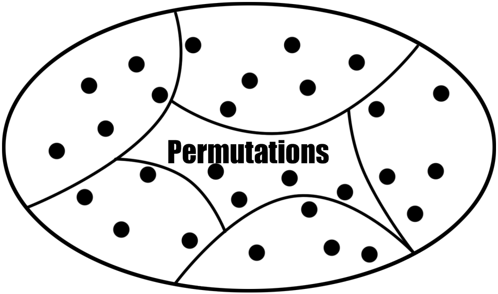


Code construction for rewriting

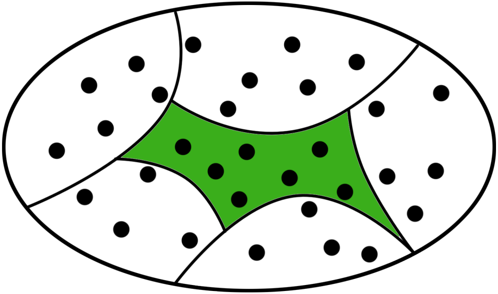
Consider: Store data of k values in n cells.



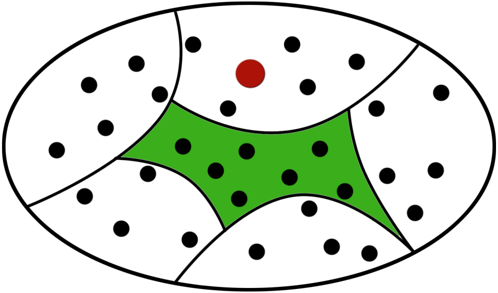
Every subset of permutations represents one value of the data.



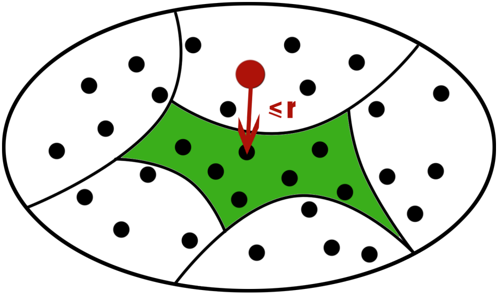
Consider one such subset, which represents one particular data value.



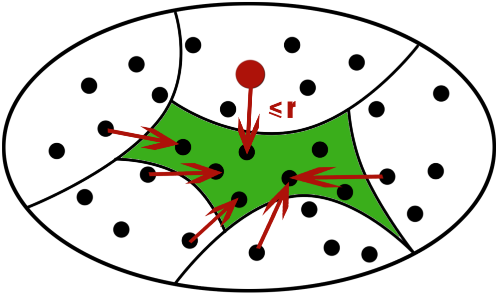
Say the red dot is the current state of the n cells. We want to change the data to the value represented by the green subset $\cdots$



Bound the rewriting cost by r.



The green subset needs to be a *dominating set* of incoming covering radius r.



We show an optimal code as an example.
Parameters: $n = 4$ cells, $k = 6$ data values, rewriting cost $r = 1$.

1 2 3 4	2 3 4 1	3 4 1 2	4 1 2 3
1 2 4 3	2 4 3 1	4 3 1 2	3 1 2 4
1 3 2 4	3 2 4 1	2 4 1 3	4 1 3 2
1 3 4 2	3 4 2 1	4 2 1 3	2 1 3 4
1 4 2 3	4 2 3 1	2 3 1 4	3 1 4 2
1 4 3 2	4 3 2 1	3 2 1 4	2 1 4 3

E. En Gad, A. Jiang and J. Bruck, "Compressed encoding for rank modulation," in ISIT 2011.

1 2 3 4	2 3 4 1	3 4 1 2	4 1 2 3	Subgroup<(1,2,3,4)>
1 2 4 3	2 4 3 1	4 3 1 2	3 1 2 4	Coset
1 3 2 4	3 2 4 1	2 4 1 3	4 1 3 2	Coset
1 3 4 2	3 4 2 1	4 2 1 3	2 1 3 4	Coset
1 4 2 3	4 2 3 1	2 3 1 4	3 1 4 2	Coset
1 4 3 2	4 3 2 1	3 2 1 4	2 1 4 3	Coset

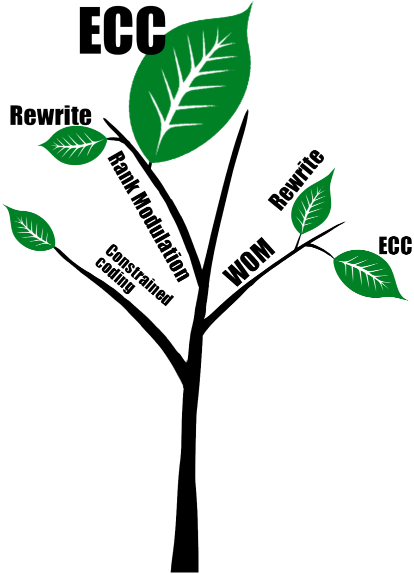
Every row (subgroup) is a dominating set of radius 1.

1 2 3 4	2 3 4 1	3 4 1 2	4 1 2 3	Subgroup<(1,2,3,4)>
1 2 4 3	2 4 3 1	4 3 1 2	3 1 2 4	Coset
1 3 2 4	3 2 4 1	2 4 1 3	4 1 3 2	Coset
1 3 4 2	3 4 2 1	4 2 1 3	2 1 3 4	Coset
1 4 2 3	4 2 3 1	2 3 1 4	3 1 4 2	Coset
1 4 3 2	4 3 2 1	3 2 1 4	2 1 4 3	Coset

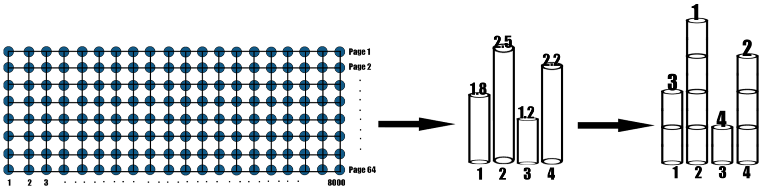
Every row (subgroup) is a dominating set of radius 1.

1 2 3 4	2 3 4 1	3 4 1 2	4 1 2 3	Subgroup<(1,2,3,4)>
1 2 4 3	2 4 3 1	4 3 1 2	3 1 2 4	Coset
1 3 2 4	3 2 4 1	2 4 1 3	4 1 3 2	Coset
1 3 4 2	3 4 2 1	4 2 1 3	2 1 3 4	Coset
1 4 2 3	4 2 3 1	2 3 1 4	3 1 4 2	Coset
1 4 3 2	4 3 2 1	3 2 1 4	2 1 4 3	Coset

So we can map the 6 cosets to 6 data values. The code has a bounded rewriting cost of 1.

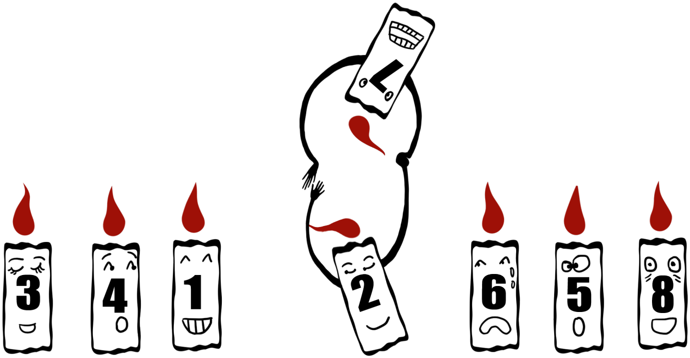

ECC
Rewrite
Rank Modulation
Rewrite
ECC
WOM
Constrained coding

1. Model errors: Noise modeling, and error quantization.



2. Design ECC.

Kendall-τ distance



Definition (Kendall-*tau* distance)

The number of adjacent transpositions to change one permutation into another. (The distance is symmetric.)

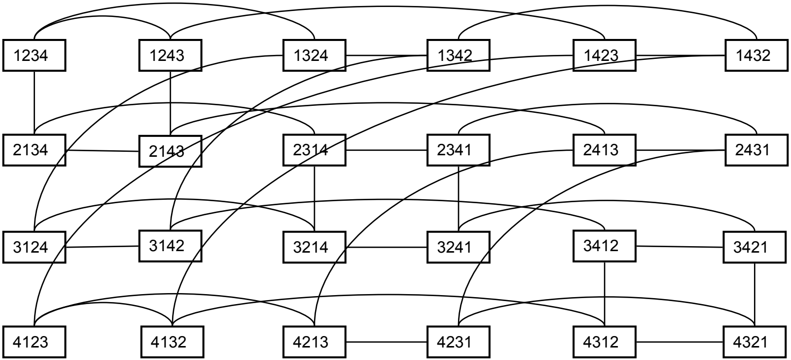
Example

For permutations $\alpha = [2, 1, 3, 4]$ and $\beta = [2, 3, 4, 1]$, the Kendall-τ distance $d_\tau(\alpha, \beta) = 2$ because $[2, 1, 3, 4] \to [2, \mathbf{3}, \mathbf{1}, 4]] \to [2, 3, \mathbf{4}, \mathbf{1}]$.

We can define an adjacency graph for permutations based on Kendall-τ distance.

Example

Permutations S_n with $n = 4$.

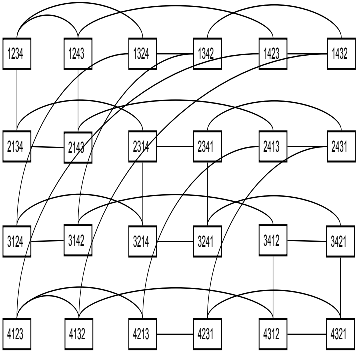


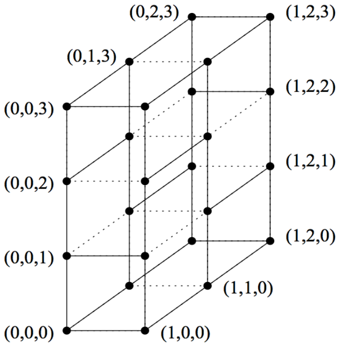
An technique for ECC construction: Embedding

Other techniques: Interleaving (product of sub-codes), modular (for limited-magnitude errors), etc.

Theorem

The adjacency graph for permutations is a subgraph of an $(n-1)$-dimensional array, whose size is $2 \times 3 \times \cdots \times n$.





A. Jiang, M. Schwartz and J. Bruck, Error-correcting codes for rank modulation, in ISIT 2009.

Construction (One-Error-Correcting Rank Modulation Code)

Let $C_1, C_2 \subseteq S_n$ denote two rank modulation codes constructed as follows. Let $A \in S_n$ be a general permutation whose inversion vector is $(x_1, x_2, \cdots, x_{n-1})$. Then A is a codeword in C_1 iff the following equation is satisfied:

$$\sum_{i=1}^{n-1} i x_i \equiv 0 \pmod{2n-1}$$

A is a codeword in C_2 iff the following equation is satisfied:

$$\sum_{i=1}^{n-2} i x_i + (n-1) \cdot (-x_{n-1}) \equiv 0 \pmod{2n-1}$$

Between C_1 and C_2, choose the code with more codewords as the final output.

For the above code, it can be proved that:

- The code can correct one Kendall error.
- The size of the code is at least $\frac{(n-1)!}{2}$.
- The size of the code is at least half of optimal.

Codes correcting more Kendall errors are constructed based on embedding.

First, consider codes of the following form:

- Let $m \geq n-1$ and let $h_1, \cdots, h_{n-1}$ be a set of integers, where $0 < h_i < m$ for $i = 1, \cdots, n-1$. Define the code as follows:

$$\mathcal{C} = \{(x_1, x_2, \cdots, x_{n-1}) \mid \sum_{i=1}^{n-1} h_i x_i \equiv 0 \mod m\}$$

[1] A. Barg and A. Mazumdar, "Codes in Permutations and Error Correction for Rank Modulation," in *Proc. IEEE International Symposium on Information Theory (ISIT)*, pp. 854–858, June 2010.

Capacity with Kendall-τ distance

Let the number of cells $n \to \infty$. Consider capacity.

Theorem (Capacity of Rank Modulation ECC with $n \to \infty$)

Let $A(n, d)$ be the maximum number of permutations in S_n with minimum Kendall-tau distance d. We call

$$C(d) = \lim_{n\to\infty} \frac{\ln A(n,d)}{\ln n!}$$

the capacity of rank modulation ECC of Kendall-tau distance d. Then,

$$C(d) = \begin{cases} 1 & \text{if } d = O(n) \\ 1-\epsilon & \text{if } d = \Theta(n^{1+\epsilon}),\ 0 < \epsilon < 1 \\ 0 & \text{if } d = \Theta(n^2) \end{cases}$$

[1] A. Barg and A. Mazumdar, "Codes in Permutations and Error Correction for Rank Modulation," in *Proc. IEEE International Symposium on Information Theory (ISIT)*, pp. 854–858, June 2010.

Further reading

- A. Mazumdar, A. Barg and G. Zemor, "Constructions of Rank Modulation Codes," in *Proc. IEEE International Symposium on Information Theory (ISIT)*, 2011.
- A. Jiang, M. Schwartz and J. Bruck, "Correcting charge-constrained errors in the rank modulation scheme," in *IEEE Trans. Information Theory*, May 2010.
- F. Farnoud, V. Skachek, and O. Milenkovic, "Error-correction in flash memories via codes in the Ulam metric," in *IEEE Trans. Information Theory*, May 2013.
- I. Tamo and M. Schwartz, "Correcting limited-magnitude errors in the rank-modulation scheme," in *IEEE Trans. Information Theory*, June 2010.
- Fan Zhang, Henry Pfister, and A. Jiang, "LDPC codes for rank modulation in flash memories," in ISIT 2010.
- R. Heymann, J. H. Weber, T. G. Swart, and H. C. Ferreira, "Concatenated permutation block codes based on set partitioning for substitution and deletion error-control," in ITW 2013.

Further reading

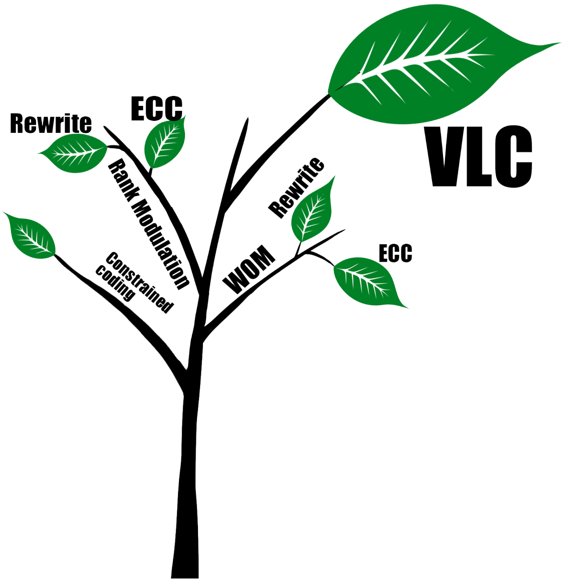
- H. Zhou, A. Jiang and J. Bruck, "Systematic error-correcting codes for rank modulation," in ISIT 2012.
- H. Zhou, A. Jiang and J. Bruck, "Error-correcting schemes with dynamic thresholds in nonvolatile memories," in ISIT 2011.
- E. En Gad, A. Jiang and J. Bruck, "Trade-offs between instantaneous and total capacity in multi-cell flash memories," in ISIT 2012.
- Q. Li, "Compressed rank modulation," in Allerton 2012.
- F. Sala, R. Gabrys, and L. Dolecek, "Dynamic threshold schemes for multi-level non-volatile memories," in *IEEE Trans. Communications*, July 2013.
- S. Buzaglo, E. Yaakobi, J. Bruck, and T. Etzion, "Error-correcting codes for multipermutations," in ISIT 2013.
- F. Farnoud and O. Milenkovic, "Multipermutation codes in the Ulam Metric for nonvolatile memories," in JSAC special issue on Communication Methodologies for the Next-generation Storage Systems, May 2014.

Further reading

- E. En Gad, E. Yaakobi, A. Jiang and J. Bruck, "Rank-modulation rewriting codes for flash memories," in ISIT 2013.
- D. Slepian, "Permutation modulation," in *Proc. IEEE*, Mar. 1965.
- I. F. Blake, G. Cohen, and M. Deza, "Coding with permutations," *Inf. Control*, 1979.
- A. J. Han Vinck and H. C. Ferreira, "Permutation trellis-codes," in ISIT 2001.
- C. J. Colbourn, T. Klove, and A. C. H. Ling, "Permutation arrays for powerline communication and mutually orthogonal Latin squares," in *IEEE Trans. Information Theory*, June 2004.
- I. Tamo and M. Schwartz, "On the labeling problem of permutation group codes under the infinity metric," in *IEEE Trans. Information Theory*, Oct. 2012.

Further Reading

- M. Schwartz and I. Tamo, "Optimal permutation anticodes with the infinity norm via permanents of $(0, 1)$-matrices," in *Journal of Combinatorial Theory, 2011.*
- T. Klove, T. Lin, S. Tsai and W. Tzeng, "Permutation arrays under the Chebyshev distance," in *IEEE Trans. Information Theory*, June 2010.
- Y. Yehezkeally and M. Schwartz, "Snake-in-the-box codes for rank modulation," in *IEEE Trans. Information Theory*, Aug. 2012.
- T. Wadayama and M. Hagiwara, "LP-decodable permutation codes based on linearly constrained permutation matrices," in *IEEE Trans. Information Theory*, Aug. 2012.
- M. Schwartz, "Quasi-cross lattice tilings with applications to flash memory," in *IEEE Trans. Information Theory*, Apr. 2012.
- M. Qin, A. Jiang and P. H. Siegel, "Parallel programming of rank modulation," in ISIT 2013.


VLC
Rewrite
ECC
Rank Modulation
Constrained coding
WOM
Rewrite
ECC

What is the right number of levels?

Performance of SLC, MLC and TLC:

- SLC: 2 levels, endurance of $\sim 10^5$ Program/Erase cycles.
- MLC: 4 levels, endurance of $\sim 10^4$ Program/Erase cycles.
- TLC: 8 levels, endurance of $\sim 10^3$ Program/Erase cycles.

Question: Is there a way to adaptively choose the number of levels, based on the cells' quality and random programming performance?

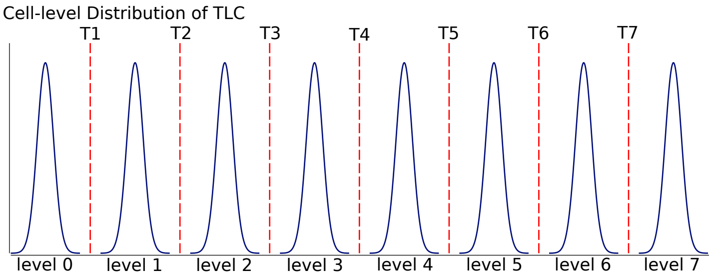
Variable Level Cell (VLC) [1]

Main Idea of VLC:

- Set thresholds dynamically.
- Do not fix the number of levels in advance.

[1] A. Jiang, H. Zhou and J. Bruck, Variable-level cells for nonvolatile memories, in *Proc. ISIT*, pp. 2489-2493, 2011.

Existing Technology: Fixed Thresholds and Levels

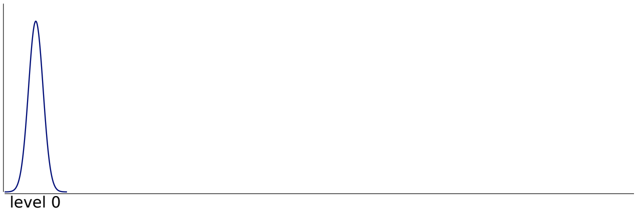


Variable Level Cell (VLC)

Main Idea of VLC:

- Set thresholds dynamically.
- Do not fix the number of levels in advance.

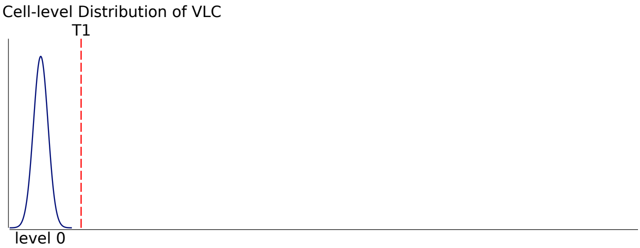
Cell-level Distribution of VLC



Variable Level Cell (VLC)

Main Idea of VLC:

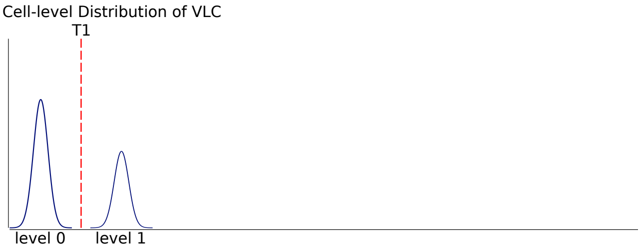
- Set thresholds dynamically.
- Do not fix the number of levels in advance.



Variable Level Cell (VLC)

Main Idea of VLC:

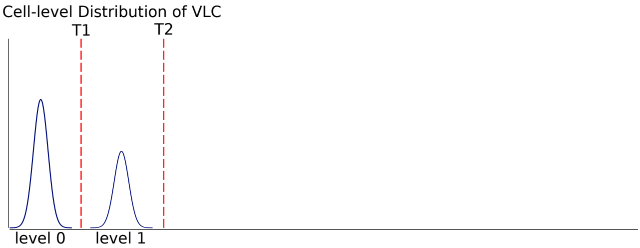
- Set thresholds dynamically.
- Do not fix the number of levels in advance.



Variable Level Cell (VLC)

Main Idea of VLC:

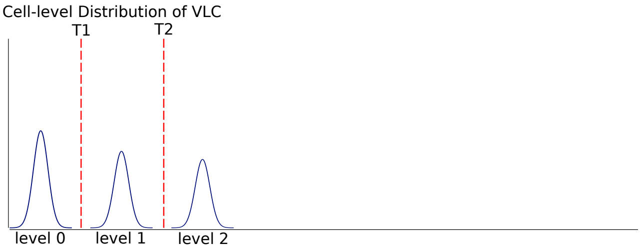
- Set thresholds dynamically.
- Do not fix the number of levels in advance.



Variable Level Cell (VLC)

Main Idea of VLC:

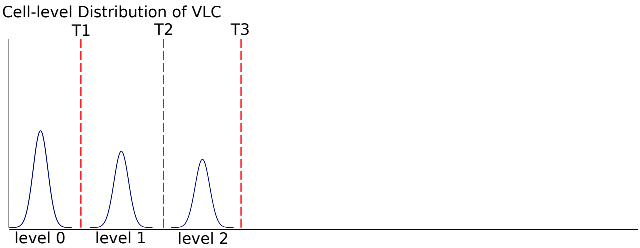
- Set thresholds dynamically.
- Do not fix the number of levels in advance.



Variable Level Cell (VLC)

Main Idea of VLC:

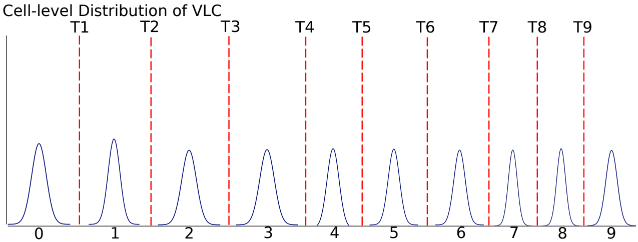
- Set thresholds dynamically.
- Do not fix the number of levels in advance.



Variable Level Cell (VLC)

Main Idea of VLC:

- Set thresholds dynamically.
- Do not fix the number of levels in advance.

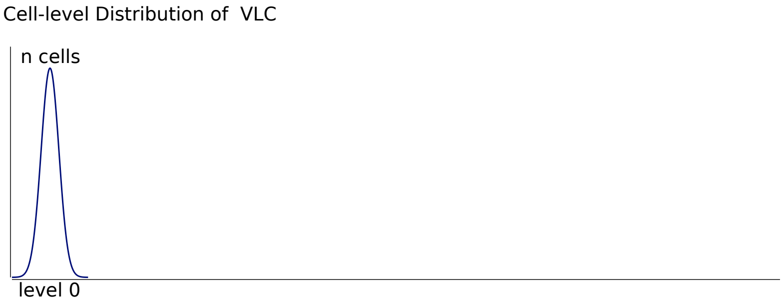


Variable Level Cell (VLC)

- VLC is more adaptive compared to current schemes.
- Programming is more robust to
  - Cell quality degradation/variance;
  - Probabilistic charge injection behavior.
- Multiple levels can be programmed in parallel for higher speed.

Storing Data in VLC

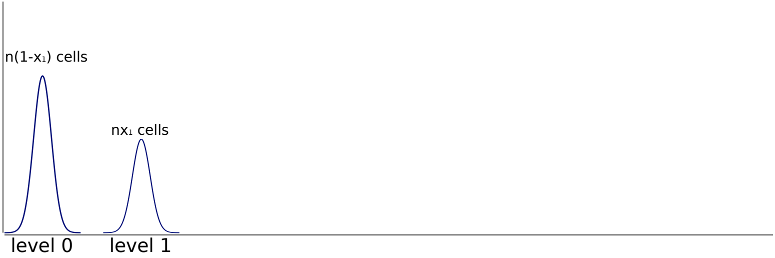
How to store data? One solution for one-write storage:



Storing Data in VLC

- Level 1 can store $nH(x_1)$ bits.
- Reading these $nH(x_1)$ bits will require two threshold comparisons.

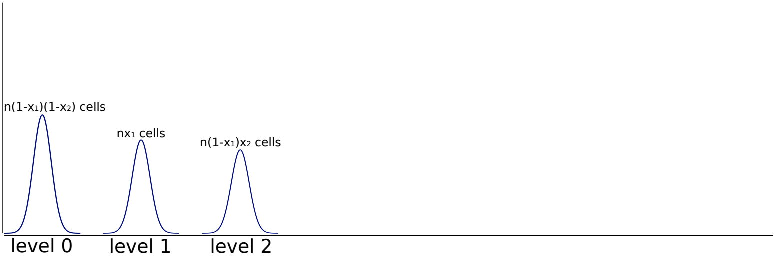
Cell-level Distribution of VLC



Storing Data in VLC

- Level 2 can store $n(1 - x_1)H(x_2)$ bits.
- Reading these $n(1 - x_1)H(x_2)$ bits will require one additional threshold comparison.

Cell-level Distribution of VLC



Capacity of VLC

Assume

- Level 1 can be programmed with probability p_1;
- Level 2 can be programmed with probability $p_1 p_2$;
- Level 3 can be programmed with probability $p_1 p_2 p_3$;
- $\cdots$;
- Level q can be programmed with probability $p_1 p_2 \cdots p_q$, where q is the maximum possible level number.

Capacity of VLC

Define $A_1, A_2, \cdots, A_{q-1}$ recursively:

- Let $A_{q-1} = 2^{p_{q-1}}$;
- For $i = q-2, q-3, \cdots, 1$, let $A_i = (1 + A_{i+1})^{p_i}$.

Theorem

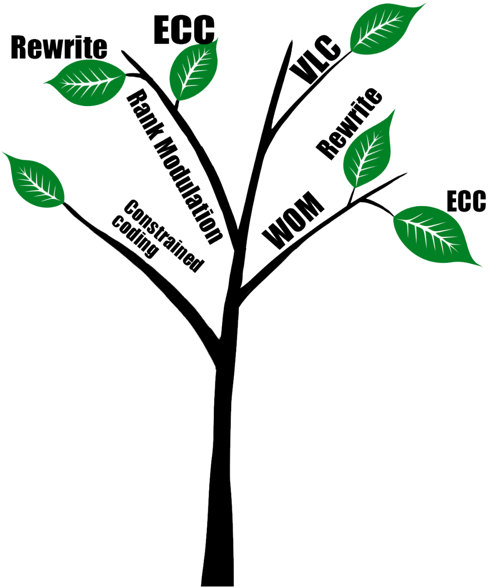
The capacity (expected value) of VLC is

$$C_{VLC} = \log_2 A_1$$

bits per cell.

- For the capacity region of rewriting codes, see:

[1] A. Jiang, H. Zhou and J. Bruck, Variable-level cells for nonvolatile memories, in *Proc. ISIT*, pp. 2489-2493, 2011.


Rewrite
ECC
VLC
Rank Modulation
Rewrite
Constrained coding
WOM
ECC