A Constraint Scheme for Correcting Massive Asymmetric Magnitude-1 Errors in Multi-level NVMs

Evyatar Hemo
Yuval Cassuto

Technion – Israel Institute of Technology
Electrical Engineering

Coding Seminar 28/06/2015
The Blessing/Curse of Multi-Level

2-Level Cell

q-Level Cell

Blessing: More bits/cell
Curse: Smaller margins
Asymmetric Magnitude-1 Retention Errors

Y. Cai et al. "Error Patterns in MLC NAND Flash Memory: Measurement, Characterization, and Analysis" DATE 2012
Flexible decoder/encoder which can change its error correcting capability can extend the life of the media.
Correcting Asymmetric Magnitude-1 Errors

1. Give up LSB (all even levels)  
   - Correct all errors in block  
   [Ahlswe, Aydinian, Khachatrian, Tolhuizen, 2004]

2. Use either all-even or all-odd levels (even/odd)  
   - Correct half errors in block

3. Encode LSB with a binary symmetric-error code (e.g. BCH)  
   - Correct $t$ errors in block ($t$ small compared to block size $n$)  
   [YC, Schwartz, Bohossian, Bruck, 2010]

4. Use L1-metric code  
   - A stronger error model  
   [Bose, Tallini, 201X]
1. Give up LSB (all even levels)
   • Correct all errors in block

2. Use either all-even or all-odd levels (even/odd)
   • Correct half errors in block

3. Encode LSB with a binary code (e.g. BCH)
   • Correct t errors in block (t small compared to block size n)
The Non-Consecutive Constraint (NCC)

- NCC(n,q): a q-level, n-cell block does not contain cells with consecutive levels
- For example, for q=8, n=6, a valid NCC codeword is

  1 6 6 1 1 3

- While the following word is not

  1 6 6 4 1 3

- For (n=6, q=8) the information rate
  \[ R = 0.7934 > 0.722 > 0.667 \]

  NCC even/odd All even
Information Rate - Comparison

Information rates for $q=8$

\[
\mathcal{R}_{NCC}(n, q) = \log_q \left[ \sum_{k=1}^{\frac{q}{2}} k! \cdot S(n, k) \cdot \binom{q-k+1}{k} \right]/n
\]
**NCC’s Correction Capability**

- **Good example (for NCC(10,8))**: 
  
  | 6 | 3 | 6 | 6 | 3 | 3 | 6 | 3 | 3 | 6 |
  
  **Histogram:**
  
  | 0 | 0 | 0 | 5 | 0 | 0 | 5 | 0 |
  
  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |

- **This codeword can correct any 2 errors or less**
  
  - For example, received histogram:

  | 0 | 0 | 2 | 3 | 0 | 0 | 5 | 0 |
  
  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
**NCC’s Correction Capability**

- **Bad example (for NCC(4,8)):**

  ![Histogram](image)

  **Histogram:**

<table>
<thead>
<tr>
<th>0</th>
<th>0</th>
<th>0</th>
<th>2</th>
<th>0</th>
<th>0</th>
<th>2</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
</tr>
</tbody>
</table>

- **This codeword cannot correct even a single error!**
  - For example, received histogram:
Objective: decode a received codeword to the nearest NCC codeword

Assuming i.i.d errors, our decoding algorithm is ML optimal

The algorithm uses Dynamic Programming

1. Converting to histogram representation
2. Dividing the histogram to sections
3. Applying decoding algorithm on each section
4. Mapping back the histogram to a codeword
NCC Decoding – Example (1)

- **Codeword:**
  
  \[
  \begin{array}{cccccccccccc}
  6 & 1 & 1 & 6 & 6 & 1 & 1 & 3 & 6 & 1 \\
  \end{array}
  \]

- **The codeword suffers 3 errors:**
  
  \[
  \begin{array}{cccccccccccc}
  6 & 1 & 1 & 5 & 6 & 0 & 1 & 2 & 6 & 1 \\
  \end{array}
  \]

- **The histogram of the corrupted codeword is:**
  
  \[
  \begin{array}{cccccccc}
  1 & 4 & 1 & 0 & 0 & 1 & 3 & 0 \\
  0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\
  \end{array}
  \]

  NCC violation!
  Section 1

  NCC violation!
  Section 2
We start with Section 2:

There are two ways to settle the constraint-violation:

3 moves
1 move
(more likely)
Now Section 1:

Level 0 moves up
1 move

Level 1 moves up
4 moves

Level 2 moves up
1 move

The violation is settled in 2 moves

The violation is settled in 4 moves
Overall we get:

We return to the codeword:

Corrected 3 errors!
**Proposition:** Let $h$ be a histogram vector of some memory word. $h$ is a histogram of a legal NCC codeword iff

\[
diag(h) M h = 0
\]

where

\[
M = \begin{pmatrix}
0 & 1 & 0 & \cdots & 0 \\
0 & 0 & 1 & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & 0 \\
0 & 0 & 0 & \cdots & 1 \\
0 & 0 & 0 & 0 & 0
\end{pmatrix}
\]

**Proof:**

\[
diag(h) M h = \begin{bmatrix}
h_0 \cdot h_1, h_1 \cdot h_2, \ldots, h_{q-2} \cdot h_{q-1}, 0
\end{bmatrix}
\]
Algebraic formalism – cont.

\[ h = h_1 + T \cdot d \]

Some other (erroneous?) histogram

Original NCC histogram

Vector of errors in the histogram

Operator of channel errors

<table>
<thead>
<tr>
<th>( h_1 )</th>
<th>0</th>
<th>5</th>
<th>0</th>
<th>1</th>
<th>0</th>
<th>0</th>
<th>4</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>( d )</th>
<th>0</th>
<th>1</th>
<th>0</th>
<th>1</th>
<th>0</th>
<th>0</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>( h )</th>
<th>1</th>
<th>4</th>
<th>1</th>
<th>0</th>
<th>0</th>
<th>1</th>
<th>3</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td></td>
</tr>
</tbody>
</table>

\[ T = \begin{pmatrix}
0 & 1 & 0 & 0 & \cdots & 0 \\
0 & -1 & 1 & 0 & \cdots & 0 \\
0 & 0 & -1 & 1 & \cdots & 0 \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & \cdots & -1 & 1 \\
0 & 0 & 0 & 0 & \cdots & -1 \\
\end{pmatrix}_{q \times q} \]
**Definition 1:** The histogram $h_1$ is **t-confusable** as $h_2$ if after suffering $t$ errors (or less) a maximum-likelihood decoder returns $h_2$. 

$$D^\downarrow(h_1, h_2)$$

$D^\uparrow(h_1, h_2)$

Erroneous histogram

$h_1 + T \cdot d$
Definition 2: The downward and upward distances are given by:

\[
D^\downarrow (h_1, h_2) = \|d(h_1, h_2)\|, \\
D^\uparrow (h_1, h_2) = \|\hat{d}(h_1, h_2)\|.
\]

where \(d\) and \(\hat{d}\) are the downward and upward difference vectors given by:

\[
d(h_1, h_2) = \frac{T^\dagger (h_2 - h_1) + |T^\dagger (h_2 - h_1)|}{2},
\]

\[
\hat{d}(h_1, h_2) = -\frac{T^\dagger (h_2 - h_1) - |T^\dagger (h_2 - h_1)|}{2}.
\]

Inverse operator of channel errors:

\[
T^\dagger = \begin{pmatrix}
0 & 0 & 0 & 0 & \cdots & 0 \\
0 & -1 & -1 & -1 & \cdots & -1 \\
0 & 0 & -1 & -1 & \cdots & -1 \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & \cdots & -1 & -1 \\
0 & 0 & 0 & \cdots & 0 & -1
\end{pmatrix}_{q\times q}
\]

\[
T^\dagger \cdot T = I
\]
Theorem 1: $h_1$ is $t$-confusable with some other NCC histogram $h_2$ iff the following conditions hold:

A. $d(h_1, h_2) \leq h_1, \hat{d}(h_1, h_2) \leq h_2$

B. $D^\uparrow(h_1, h_2) \leq D^\downarrow(h_1, h_2) \leq t$.

\[ D^\downarrow(h_1, h_2) = \|d(h_1, h_2)\| + \|c\| \]

\[ D^\uparrow(h_1, h_2) = \|\hat{d}(h_1, h_2)\| + \|c\| \]
From Proposition 1 and Theorem 1 we get the element-wise solution for $t$-confusable histograms:

$$h_i \cdot [(d_i - d_{i-1}) - (\hat{d}_i - \hat{d}_{i-1})] + h_{i-1} \cdot [(d_{i+1} - d_i) - (\hat{d}_{i+1} - \hat{d}_i)] + [(d_i - d_{i-1}) - (\hat{d}_i - \hat{d}_{i-1})] \cdot [(d_{i+1} - d_i) - (\hat{d}_{i+1} - \hat{d}_i)] = 0$$

Corollary 1: if $h_i > 2t, \forall i$ in which $h_i \neq 0$ then $h$ has no $t$-confusable histograms.

**Proof:** can be deduced explicitly from the expression above but it is also intuitive…

Histogram pattern:

- $h_{i-1}$
- $h_i$
- $>t$
- $>2t$
Theorem 1 enables us finding all the histograms that cannot be decoded successfully.

For example, all 1-confusible histograms must contain the following patterns:
Theorem 3: Given that $t \leq 3$ uniformly selected errors occurred in a uniformly selected $NCC(n \geq 7, q)$ codeword, the probability $P_t$ to successfully correct all $t$ errors is bounded by below by

$$1 - \frac{\sum_{k=1}^{\frac{q}{2}} k! F_{2t+1} \cdot \left[ \frac{(q-k+1)}{k} - 1 \right] \cdot \frac{2^k t}{n}}{M} \leq P_t$$

where $M$ is the size of the NCC code and

$$F_{2t+1}(n, k) = S(n, k) - S_{2t+1}(n, k)$$

where $S_r(n, k)$ are the $r$-associated Stirling numbers of the second kind.
Recall Corollary 1: if $h_i > 2t$, $\forall i$ in which $h_i \neq 0$ then $h$ has no $t$-confusable histograms.

\[ F_{2t+1}(n, k) = S(n, k) - S_{2t+1}(n, k). \]
This is an upper bound on the number of codewords that cannot always be successfully decoded after enduring $t$ errors.

\[ \sum_{k=1}^{q/2} k!F_{2r+1} \cdot \left[ \binom{q-k+1}{k} - 1 \right] \]

The number of combinations of $k$ non-consecutive values out of $q$.

Represents the level set: $0, 2, 4, \ldots, 2k-2$ which can always correct $t$ errors.

We now need to multiply it with the probability for uncorrectable error to occur.
When $k$ levels are occupied there are at most $k$ levels that can endure uncorrectable errors.

Each such level can contain at most $h_i = 2t$ cells and can cause unsuccessful decoding when enduring $h_i/2$ errors or more.

We now count all the error patterns which can cause uncorrectable errors:

For $t \leq 3$ and $n \geq 7$ it can be verified that

$$P(n, t) \leq \frac{2kt}{n}$$

Overall we get

$$1 - \sum_{k=1}^{q/2} k! F_{2t+1} \cdot \left[ \frac{(q-k+1)}{k} - 1 \right] \cdot \frac{2k-1}{n} \leq P_t$$
Theorem 2: for n-cell, q-level NCC, when q is fixed and $n \to \infty$ the probability to correct a single error is bounded by:

$$P_1 \geq 1 - n \left(1 - \frac{2}{q}\right)^n (C - o(1))$$

Full-correction capabilities of NCC:

| $q = 8$ | $R_{NCC}$ | $||e||$ | Correction Probability |
|---------|-----------|--------|------------------------|
| 5       | 0.816     | 1      | 0.801                  |
| 9       | 0.752     | 2      | 0.967                  |
| 13      | 0.726     | 3      | 0.993                  |
| 17      | 0.712     | 4      | 0.998                  |
| 5       | 0.816     | 5      | 0.998                  |
| 9       | 0.752     | 6      | 0.998                  |

Prob. to correct 5 errors in length 13 codewords
Equal-Rate Performance Comparison

 Codes comparison $R \approx 0.726$

- no ECC
- NCC
- BCH(31,6,15)
- even/odd

![Graph showing codes comparison](image)
SSD Operation: with LDPC Outer Code

- For raw BER of ~0.08:
  - The even/odd scheme gives 0.0203
  - BCH even worse: 0.035
  - The NCC gives 0.0065

Success.

NCC – Advantages

**Performance:**
Best correction for moderate to high error rates

**Complexity:**
O(q) maximum likelihood decoder

**Flexibility:**
Code length sets rate/correction tradeoff
Work in progress...

- Deriving bounds on the NCC performance
- Comparing NCC performance with other ECC and fundamental bounds
- Enhancing the flexibility of the NCC
Thank You!
Algorithm 2: DecodeSection

```
input : S
output: actionVec

//initialization
W_1(σ) = C(σ(b_1)), W_1(σ̅) = C(σ̅(b_1))
Π_1(σ) = Π_1(σ̅) = ∅

//tables fill
for j = 2 to N_b(S)
    α = argmin_{[σ, σ̅]} (W_{j-1}(σ), W_{j-1}(σ̅))

//for odd-length bursts:
    W_j(σ) = C(σ(b_j)) + W_{j-1}(σ)
    Π_j(σ) = [Π_{j-1}(σ), σ]
    W_j(σ̅) = C(σ̅(b_j)) + min(W_{j-1}(σ), W_{j-1}(σ̅))
    Π_j(σ̅) = [Π_{j-1}(α), α]

//for even-length bursts:
    W_j(σ) = C(σ(b_j)) + min(W_{j-1}(σ), W_{j-1}(σ̅))
    Π_j(σ) = [Π_{j-1}(α), α]
    W_j(σ̅) = C(σ̅(b_j)) + W_{j-1}(σ)
    Π_j(σ̅) = [Π_{j-1}(σ), σ]

end

//making final decision
α = argmin_{[σ, σ̅]} (W_{N_b(S)}(σ), W_{N_b(S)}(σ̅))
actionVec = [Π_{N_b}(α), α]
```