Amit Mitigating Inter-Cell Coupling Effects in MLC NAND Flash via Constrained Coding

2 authors, including:

Yitzhak Birk
Technion - Israel Institute of Technology
94 PUBLICATIONS 2,049 CITATIONS

All content following this page was uploaded by Yitzhak Birk on 29 December 2015.

The user has requested enhancement of the downloaded file.
Mitigating Inter-Cell Coupling Effects in MLC NAND Flash via Constrained Coding

Amit Berman and Yitzhak Birk
{bermanam@tx, birk@ee}.technion.ac.il
Technion – Israel Institute of Technology
August, 2010
Agenda

- Problem Definition: Inter-Cell Coupling
- Related Work
- Novel Solution: Constrained Coding System
- An Example
- Conclusions
FG-FG inter-cell coupling causes the charge in one cell to affect a neighboring cell’s threshold voltage.
When considering each cell in isolation, the observed phenomenon is a “widening” of the threshold voltage distributions.
Neglecting $C_{FGXY}$, and assuming $Q_{FG}=0$ the floating gate voltage due to ICC is:

$$V_{FG} = \frac{C_{ONO}V_{CG} + C_{FGX}(V_1 + V_2) + C_{FGY}(V_3 + V_4) + V_{FGCG}(V_5 + V_6)}{C_{TUN} + C_{ONO} + 2C_{FGX} + 2C_{FGY} + 2C_{FGCG}}$$
Coupling with Program & Verify

- **Program & Verify:**
  - Charge is added to a cell in small increments
  - $V_t$ is checked after each addition
  - Programming ceases upon reaching the desired $V_t$

- Therefore, $V_t$ of any given cell is affected only charge changes made to its neighbors *after* its own charging has been completed.

The effect of inter-cell coupling depends on the programming scheme.
Existing Coupling-Mitigation Schemes

- Proportional programming
  [Fastow et al, USP 6,996,004]

- Intelligent read decoding
  [Li et al, USP 7,301,839]
Proportional Programming [Fastow et al]

- Concurrent, incremental programming of all cells, tailored for near-simultaneous completion.

- Pros:
  - Desired $V_t$ for all cells (altered only by the last pulse of each neighbor);
  - Narrow distributions.
  - Insensitive to coupling parameters.
  - Simple read

- Shortcomings:
  - Complicated, possibly slow programming
  - Can’t account for next line if programmed later
  - Can’t fully compensate when “pull” is greater than desired level (would require negative “bias”)

A. Berman and Y. Birk
Intelligent Read Decoding [Li et al]

- Simple, conventional programming
- Based on coupling equations, parameters and on programming scheme, decode smartly to offset coupling effects.

Pros:
- Simple programming
- Overlapping distributions are separated by decoding

Cons:
- Must know coupling parameters; no variation allowed.
- Requires accurate reading of $V_t$
- Complex, slow read
Our Approach: Constrained Coding

- Forbid certain adjacent-cell level combinations:
  - Criterion depends on programming order
  - Threshold is a design trade-off
- Programming: use only permissible combinations (legal code words)
- Decoding: use inverse mapping
Constrained Coding – Main Features

- **Pros:**
  - Limits the effect of inter-cell coupling → narrow distributions → many levels
  - Fairly simply encoding and decoding
  - Only need to know an upper bound on coupling coefficients

- **Cons:**
  - Code rate <1 → some loss of capacity relative to ideal with narrow distributions.
Constrained Coding - Remarks

- Can easily be combined with ECC

- Complementary to the previous schemes and can be combined with them:
  - Semi-accurate programming + minimal restrictions
  - Some restrictions with simpler intelligent read decoding
Constrained Coding System

Source 00101… → Constrained Encoder → Flash Memory → Constrained Decoder → Destination 00101…

All combinations available

Decoder recovers original data

Santa Clara, CA
August 2010

A. Berman and Y. Birk
Example: 1-D, “Breadth 1st” Coding

- 1-D: a single row of cells is considered
- Programming (charge & verify)
  - All >0 cells programmed to level 1
  - All >1 cells programmed to level 2
  - ...
- Sequence eligibility criterion:

\[ D(C) = \max \{ N_L - C, 0 \} + \max \{ N_R - C, 0 \} < T \]

- \( T \) represents a trade-off:
  - Large \( T \): efficient coding, but wider distributions and fewer levels
  - Small \( T \): opposite pros and cons

\( N_L, C, N_R \): respective target levels

\[ BL_{i-1} \quad BL_i \quad BL_{i+1} \]

\[ WL_j \quad N_L \quad C \quad N_R \]
Required Redundancy (T=5,2 bpc)

\[
Redu(S) = 1 - \lim_{l \to \infty} \frac{\log_2 N(l; S)}{l} = \frac{\log_2 n}{\log_2 n} = 0.0483
\]

• Notation:
  • N(l;S) - number of legal (permissible) l-symbol code words
  • n - number of program levels in a cell
  • S - language of all legal code words
• The required redundancy is (at least) 4.83%
Capacity Implication (T=5)

• Assumption: constrained coding permitted an increase in the number of levels from 4 to 5.

• Baseline: 
  \[ 1.0 \cdot \log_2 (4) = 2 \]

• Constrained coding: 
  \[ 0.95 \cdot \log_2 (5) = 2.2 > 2 \]

• A 10% increase in capacity
We build graph of the constraint language

- With 4 levels per cell, this example excludes the combinations (sequences) 3-0-3, 3-0-2 and 2-0-3.
Design of encoder/decoder block (cont.)

- For demonstration, consider code rate = 2/3
- For this, we can build a lookup table and use it.
The design can also be implemented with state machine. E.g., to exclude 3-0-3:

**Design of encoder-decoder block (cont.)**

- Encoding Input
  - 00/331
  - 01/322
  - 02/332
  - 03/000
  - 10/111
  - 11/222
  - 12/231
  - 13/232
  - 20/233
  - 21/131
  - 22/132
  - 23/133

- Encoding Output
  - 30/323
  - 31/333
  - 32/223
  - 33/113

- Transition States
  - 00/000
  - 01/011
  - 02/012
  - 03/022
  - 10/010
  - 11/001
  - 12/021
  - 13/002
  - 20/111
  - 22/100
  - 23/333
  - 30/223
  - 31/323
  - 32/113
  - 33/123
Conclusions

• Constrained coding can be used to chop off the tail of $V_t$ distributions with only a minor reduction in coding rate

• Can be used beneficially to increase capacity or to increase reliability

• Can replace proportional programming and intelligent decoding or complement them

• Detailed papers in preparation
• A patent application has been filed by Technion
End

Questions?

Amit Berman and Yitzhak Birk
{bermanam@tx, birk@ee}.technion.ac.il
Technion – Israel Institute of Technology
August, 2010