Lattices for Error Correction and Rewriting in Flash Memories

University of Electro-Communications
Tokyo, Japan

Brian M. Kurkoski
kurkoski@ice.uec.ac.jp

Presented at:
Non-Volatile Memories Workshop
March 6-8, 2011
University of California San Diego

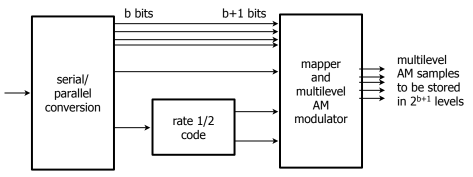
Summary This poster proposes using lattices to encode data in flash memories:

• For error correction, lattices combined with Reed-Solomon codes form a coded-modulation system that have about 1.7 to 1.9 dB lower SNR than existing BCH code systems.

• For rewriting flash memories, rewriting codes can be constructed from lattices at high rates.

Coded Modulation for Memories

In 2000, Lou and Sundberg suggested using trellis-coded modulation for memories. But for flash memories, convolutional codes do not outperform BCH codes [Sun et al., 2007].



H.-L. Lou and C.-E. Sundberg, "Increasing storage capacity in multilevel memory cells by means of communications and signal processing techniques," IEE Proceedings Circuits, Devices and Systems, vol. 147, pp. 229–236, August 2000.

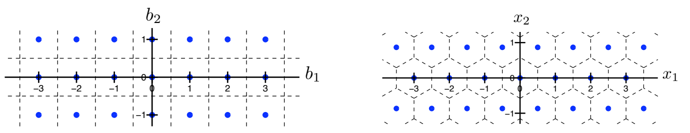
Lattices

- In n-dimensions, a lattice with generator G is subgroup of $\mathbb{R}^n$:

$$\mathbf{x} = G \cdot \mathbf{b},$$

where $\mathbf{b}$ is a vector of integers

- Lattices are codes over real numbers
- Codebook $\mathcal{C}$ is the lattice points inside side length-M cube



As dimension n increases, packing density, coding gain, etc. improves

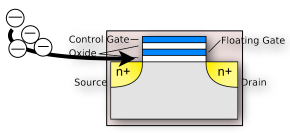
n	Lattice	Gain (dB)
2	A_2	0.84
4	D_4	1.9
8	E_8	3.7
12	K_{12}	4.5
16	Λ_{16}	5.5
24	Λ_{24}	7.1
	$\vdots$	

H. Conway and N. Sloane, *Sphere packings, lattices and groups.* Springer-Verlag, 3rd ed., 1999. p. 74.

Merits of Lattices for Flash Memories

Flash cells store charge, a continuous quantity.

- Assume signal between 0 and V.
- (other systems quantize to q levels)



flash cell

Rewriting codes using lattices

- Code over real numbers has a natural ordering, important for rewriting codes
- Lattices can correct errors (many existing rewriting constructions do not correct errors)

No synchronization problems

- Carrier-based systems use QAM, QPSK constellations for synchronization
- Memories are always synchronized

Multilevel

- Magnetic recording systems are binary, cannot use lattices
- Flash memories are multi-level

Demerits

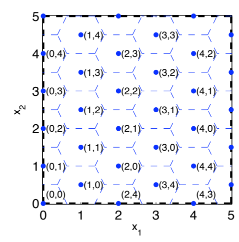
- Soft-input lattice decoding is not easy with current flash architectures (but see "Soft-Input Architecture" on this poster).
- Existing LDPC-coded modulation has excellent coding gains

Storing Lattice Values in Flash

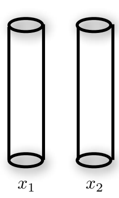
The values of an n-dimensional lattice are stored in n flash cells:

- Assume signal between 0 and V (i.e. do not quantize to q levels)
- Codebook $\mathcal{C}$ is the lattice points inside side length-M cube
- If G is lower triangular, then mapping $\mathcal{B} = \{0, 1, \ldots, M-1\}^n \longrightarrow \mathcal{C}$ is efficient

$\{0, 1, 2, 3, 4\}^2 \longrightarrow$



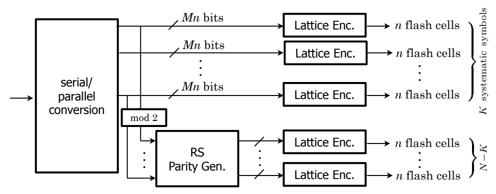
2-dimensional lattice



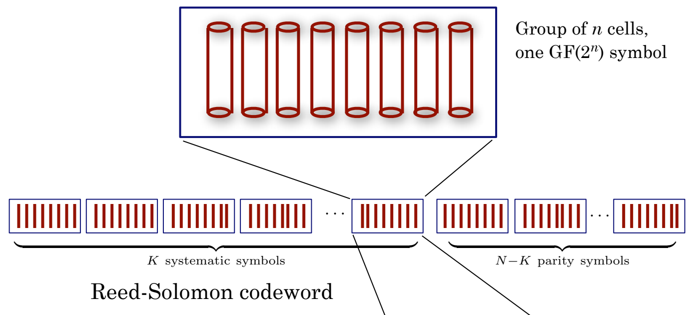
2 flash cells

Coded Modulation with Lattices and Reed-Solomon Codes

Propose coded-modulation system using lattices and a systematic (N,K) Reed-Solomon code over GF(2^n)



- Each GF(2^n) symbol corresponds to one group of flash cells.
- Only encode mod 2 data values (increases the rate) — lattice Euclidean distance is important.
- Lattice decoding errors are bursty, so Reed-Solomon codes are well suited.
- For flash memories, Reed-Solomon codes have lower decoding complexity than BCH codes [Chen et al., 2008].



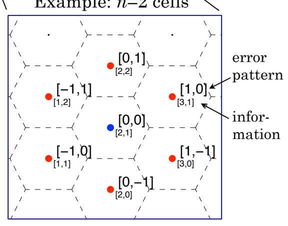
Decoding

1. Perform lattice-by-lattice decoding.
2. Perform Reed-Solomon decoding.
3. Using correct RS symbol, correct lattice decoding errors.

Correction of lattice decoding errors

- Assume that the Reed-Solomon decoder provides the correct symbol.
- Therefore, correct value mod 2 known.
- When a lattice error occurs, with high probability, a transmitted point (blue) will be decoded as a neighboring point (red).

Example: n=2 cells



A hex lattice point has 6 neighbors.

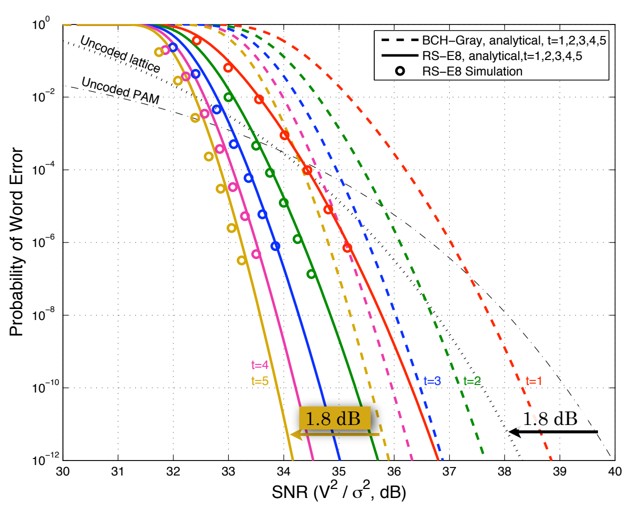
To distinguish the two true error patterns, compute the Euclidean distance between the received signal and each candidate. Shortest distance wins.

For the E8 lattice, a GF(2^8) symbol is sufficient to distinguish the 240 neighbors, except for a sign change.

true error pattern	mod 2 error pattern
$[0,1]$, $[0,-1]$	$[0,1]$
$[1,0]$, $[-1,0]$	$[1,0]$
$[-1,1]$, $[1,-1]$	$[1,1]$

Numerical Results

- Evaluation using an AWGN system, compared with a Gray-coded pulse-amplitude (PAM) system using BCH codes.
- The E8 lattice has about 1.8 dB gain over PAM lattice. Comparing Reed-Solomon and BCH codes of the same rate, this gain is preserved.



Complexity of E8 Lattice Decoding

Two algorithms exist to find the E8 lattice point closest to $\mathbf{x} \in \mathbb{R}^8$.

Coset Decoding (about 104 steps) $f(\mathbf{x})$ is $\mathbf{x}$ rounded to nearest integer. $g(\mathbf{x})$ has least reliable position rounded "wrong way."

$$\mathbf{y}_1 = \begin{cases} f(\mathbf{x}) & \text{if } \sum f(\mathbf{x}) \text{ is even} \\ g(\mathbf{x}) & \text{otherwise} \end{cases} \qquad \mathbf{y}_2 = \begin{cases} f(\mathbf{x}+\frac{1}{2}) & \text{if } \sum f(\mathbf{x}+\frac{1}{2}) \text{ is even} \\ g(\mathbf{x}+\frac{1}{2}) & \text{otherwise} \end{cases}$$

If $||\mathbf{x}-\mathbf{y}_1||_2 < ||\mathbf{x}-\mathbf{y}_2||_2$ then output $\mathbf{y}_1$. Otherwise, output $\mathbf{y}_2$.

"Construction A" Decoding (about 72 steps)

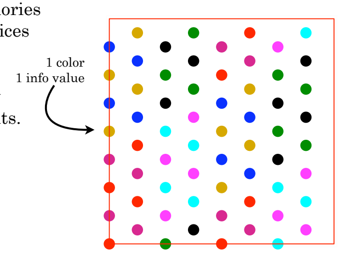
1. Find $\mathbf{y}$ and $\mathbf{z} \in \mathbb{Z}^8$ such that $\mathbf{x} = \mathbf{y} - 4\mathbf{z}$ and $-1 \le y_i < 3$.
2. S denotes the set of i for which $1 < y_i < 3$. For $i \in S$, replace y_i by $2 - y_i$.
3. Decode $\mathbf{y}$ as a first-order Reed-Muller code of length 8. Output $\mathbf{c}$.
4. For $i \in S$, change c_i to $2 - c_i$. Output $\mathbf{c} + 4\mathbf{z}$.

Above work is based upon "The E8 Lattice and Error Correction in Multi-Level Flash Memory," to appear in Proceedings of ICC 2011 (Kyoto, Japan), June 2011.

Rewriting Codes Using Lattices

Rewriting codes allow writing flash memories two or more times without erasing. Lattices can be used to construct rewriting codes.

The main idea is to create a one-to-many mapping from information to lattice points.



base codebook

Codebook mapping

- Choose parameter $M \le V$,
- Information is encoded in U bits, $\mathcal{U} = \{0, 1, \ldots, 2^U - 1\}$
- "Coset Select" is encoded in C bits, $\mathcal{CS} = \{0, 1, \ldots, 2^C - 1\}$
  – if $C = 0$, then mapping is one-to-one
- The following encoding mapping is needed:

$$\Phi \; : \; \mathcal{U} \times \mathcal{CS} \longrightarrow \mathcal{B}$$

"Dirty Paper Coding" for Rewriting Flash

Shaping region B is a side length-M cube with corner at 0

- All codewords have positive values.
- Entire space can be covered with translations of B.
- "mod B" is well-defined and easy to compute.
- Codebook is intersection of B and lattice.

Dirty paper (DPC) encoding:

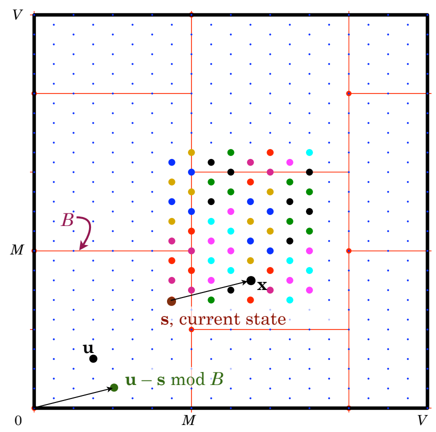
Let $\mathbf{u}$ be a codeword in base codebook.

Known interference $\mathbf{s}$, is current state of memory.

"Transmitted" codeword is $\mathbf{u} - \mathbf{s} \bmod B$ which is positive-valued.

The value in memory is $\mathbf{x} = \mathbf{u} - \mathbf{s} \bmod B + \mathbf{s}$

Decoding in absence of noise: $\mathbf{u} = \mathbf{x} \bmod B$



Numerical Results

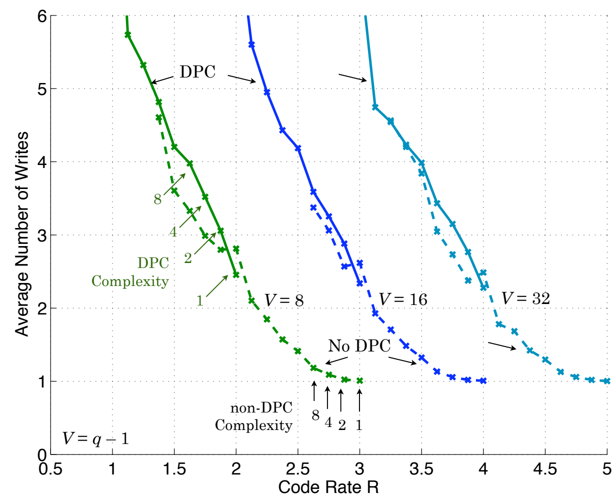
DPC system with E8 lattice:

- Base code only: $V = M$
- DPC: $V = 2M$

Interested in high-rate codes suitable for applications.

Base code ("non-DPC") can achieve highest rates. At slightly lower rate:

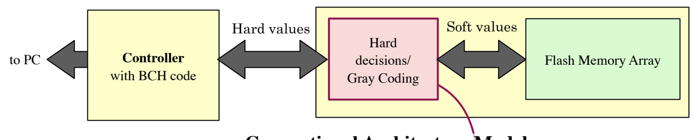
- Has similar average number of writes
- Has much lower complexity



Soft-Input Architecture

Conventional flash memory architecture:

- hard decisions made internally, ECC performed externally

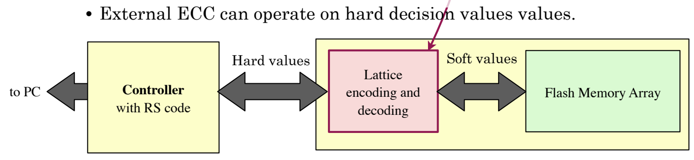


Conventional Architecture Model

On-chip soft-input decoding (e.g. LDPC) is difficult to perform on chip.

Lattice decoding is a "half-way" approach:

- Soft-input lattice decoding is more powerful than simple hard decisions.
- Lattice decoding is less complex than LDPC; can be performed on-chip.
- External ECC can operate on hard decision values values.



Soft-Input Architecture Model

Related work is "Rewriting codes for flash memories based upon lattices, and an example using the E8 lattice," GLOBECOM Workshops (GC Workshops), 2010 IEEE, pp.1861-1865, 6-10 Dec. 2010.