The E8 Lattice and Error Correction in Multi-Level Flash Memory

Brian M. Kurkoski
kurkoski@ice.uec.ac.jp
University of Electro-Communications
Tokyo, Japan

ICC 2011
IEEE International Conference on Communications
Kyoto, Japan
Monday, June 6, 2011
Overview

This presentation describes a novel coded-modulation method using
- E8 lattice as a constellation
- Reed-Solomon codes for error correction

Developed for flash memories
- Background on flash memories
- Conventional ECC uses BCH codes with Gray-coded PAM

Outline the proposed method
- Writing lattice points to flash memory
- To increase code rate, Reed-Solomon codes protect only mod-2 sum of data
- Decoder must resolve lattice error patterns

Benefits of proposed scheme:
- Gain of 1.8 dB
- Reed-Solomon decoder has lower complexity than BCH decoder for fixed block length
- A low-complexity soft-input decoding scheme
Flash Memories

- Invented by Fujio Masuoka at Toshiba; also coined the word “flash”
- Mechanically durable, non-volatile memory
- By late 1990s, made digital cameras and MP3 players viable
- Fast random reads, replacing hard disk drives for some server applications
- Now in some laptop computers

Consists of huge array of memory cells
Charge stored on a “floating gate” or “cell”

Single-level flash use 1 bits/cell
Multi-level flash uses 2+ bits/cell
- 3 bits/cell flash is now commercially available
Increasing the data density reduces cost / GB
But, increases the influence of noise
- Multi-level flash requires ECC
Conventional Flash ECC:
BCH codes on Gray-coded PAM

For PAM with AWGN model likely error is a shift to adjacent symbol
  ➢ Gray coding is used
Errors are random and uncorrelated
  ➢ BCH codes are used
  ➢ Error-correction better than Reed-Solomon
Codes rates tend to be very high, 0.95 to 0.99

One 4-level cell

Cells encoded by one BCH codeword
Lattices are “codes” over real numbers
\[ x = G \cdot b \]

Lattices, and related sphere packings
- Higher packing density
- More efficient
- Correct errors
- Achieve channel capacity \( n \to \infty \)

- QAM: 2 dimensional rectangular lattice
- Higher dimension lattices have better distance properties.

- Flash cells store charge, which is continuous
- What if we ignore discrete levels, and allow real values?

Lattices for Flash
- Flash cells store charge, which is continuous
- What if we ignore discrete levels, and allow real values?

\[ G = \begin{bmatrix} 1 & 0 \\ \sqrt{3}/2 & 1 \end{bmatrix} \]
**E8 Lattice (Gosset Lattice)**

- Best-known lattice in eight dimensions
- Lattice $d_{\text{min}}$ is $\sqrt{2} = 1.414$
  - versus $d_{\text{min}} = 1$ for PAM
- Efficient soft-input decoding algorithm
- Coordinates are integers or half integers
- Kissing number: 240 nearest neighbors at $d_{\text{min}}$
  - versus 6 for this hex lattice

![E8 generator matrix]

**Storing Lattice Point in Flash**

- One coordinate stored in one flash cell.
- So, 8 flash cells store 1 lattice point
- 8 cells encode $\log_2(V^8)$ information bits
E8 Lattice (Gosset Lattice)

- Best-known lattice in eight dimensions
- lattice $d_{\text{min}}$ is $\sqrt{2} \approx 1.414$
  - versus $d_{\text{min}} = 1$ for PAM
- Efficient soft-input decoding algorithm
- Coordinates are integers or half integers
- kissing number: 240 nearest neighbors at $d_{\text{min}}$
  - versus 6 for this hex lattice

Storing Lattice Point in Flash

- one coordinate stored in one flash cell.
- So, 8 flash cells store 1 lattice point
- 8 cells encode $\log_2(V^8)$ information bits
Proposed Encoding

- Block of 8 cells corresponds to 1 Reed-Solomon symbol
- Use an \((N,K)\) RS code
- Use a RS code over \(\text{GF}(2^8)\) - 8 bits symbols
  - Each block has \(\log_2(V^8)\) information bits,
  - BUT, is only protected by 8 bits from the Reed-Solomon codeword
  - SO, the modulo-2 information value is the Reed-Solomon symbol
Resolving Errors: 
Decoding With Only Mod-2 Error Pattern

Decoding sequence:
1. Perform lattice-by-lattice decoding.
2. Perform Reed-Solomon decoding on mod 2 value.
3. Some RS symbols will be corrected.
4. Resolve any errors

Resolve errors to recover original data:
- Lattice error causes RS symbol error: transmitted point (blue) will be decoded as a neighboring point (red) [with high probability]
- Correct RS symbol is known
- Therefore, information mod 2 known.
- Find true error pattern in lookup table.

There are two candidate error patterns
- Winner: shortest Euclidean distance to received sequence

For E8, GF(2^8) symbol distinguishes 240 neighbors, (except for sign)
- Lattice decoding errors are “bursty”
- RS with one symbol per lattice point is well suited
- GF(2^8) is well-suited for E8 lattice
Numerical Results

Compared:
- AWGN noise model
- BCH & Gray-coded PAM
- Proposed E8 & RS
- Code rates: 2.95~2.99 bits/cell (for raw 3 bit/cell)

Uncoded E8 lattice has 1.8 dB gain over uncoded PAM
This gain is preserved when the codes are added

Truth in advertising:
- E8 is a soft-input algorithm
- PAM is hard decisions
- Not unfair ....

Brian Kurkoski, University of Electro-Communications
Current architectures separate flash memory chip and ECC
- very difficult to implement soft-input LPDC decoding, etc.
- Hybrid solution: low-complexity soft-input lattice decoding on chip
  - hard decision Reed-Solomon ECC in external controller.
Discussion

Proposed E8 lattices for coded modulation flash memories using lattices
  ➢ QAM/convolutional codes for memories [Lou Sundberg 2000] [Sun et al 2007]
Reed-Solomon codes for error correction
  ➢ QAM/Reed-Solomon codes [Laneman Sundberg 2001]
  ➢ E8 lattice with convolutional codes [Calderbank Sloane 1986]

Benefits:
  ➢ Gained about 1.8 dB
  ➢ RS decoding is lower complexity than flash memory BCH decoding
  ➢ Used soft-input lattice decoding
  ➢ “architecture friendlier” than full soft-input decoding algorithms:
    ■ LDPC codes
    ■ Koetter-Vardy RS decoding
Decoding Complexity for E8 Lattice

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

**Coset Decoding** (about 104 steps) \( f(x) \) is \( x \) rounded to nearest integer. \( g(x) \) has least reliable position rounded “wrong way.”

\[
y_1 = \begin{cases} f(x) & \text{if } \sum f(x) \text{ is even} \\ g(x) & \text{otherwise} \end{cases} \quad y_2 = \begin{cases} f(x + \frac{1}{2}) & \text{if } \sum f(x + \frac{1}{2}) \text{ is even} \\ g(x + \frac{1}{2}) & \text{otherwise} \end{cases}
\]

If \( ||x - y_1||_2 < ||x - y_2||_2 \) then output \( y_1 \). Otherwise, output \( y_2 \).

“Construction A” Decoding (about 72 steps)

1. Find \( y \) and \( z \in \mathbb{Z}^8 \) such that \( x = y - 4z \) and \(-1 \leq y_i < 3\).
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 \( y \) as a first-order Reed-Muller code of length 8. Output \( c \).
4. For \( i \in S \), change \( c_i \) to \( 2 - c_i \). Output \( c + 4z \).