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:

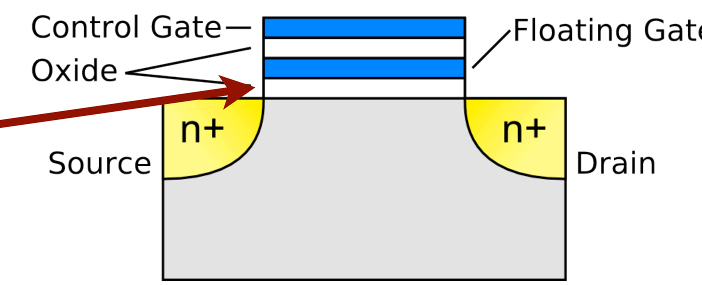
- Gain of 1.8 dB
- Reed-Solomon decoder has lower complexity than BCH decoder for fixed block length
- A low-complexity soft-input decoding scheme

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”



http://elec424.rice.edu

Single-level flash use 1 bits/cell

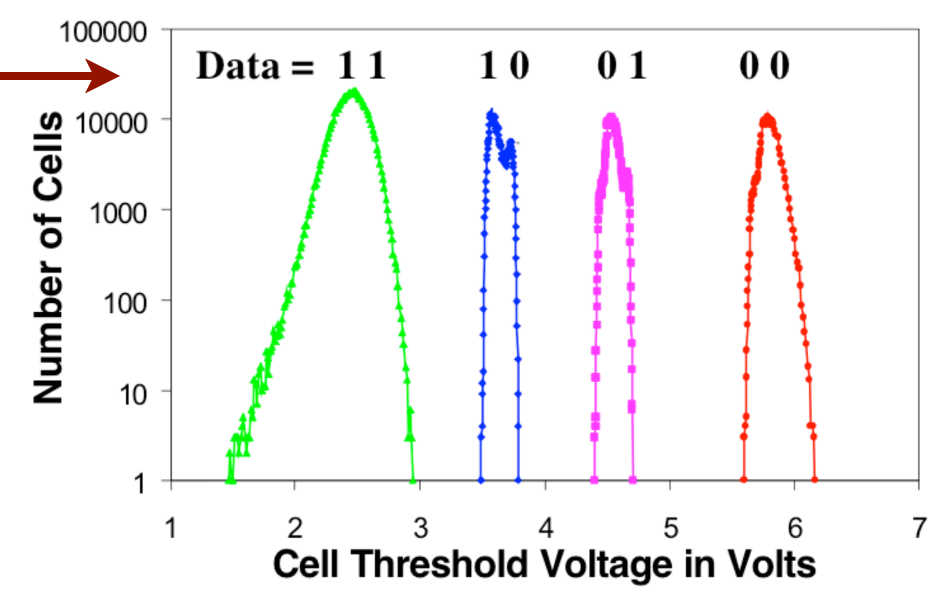
Multi-level flash uses 2+ bits/cell

- 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



Atwood, et al. 1997

Conventional Flash ECC:
BCH codes on Gray-coded PAM

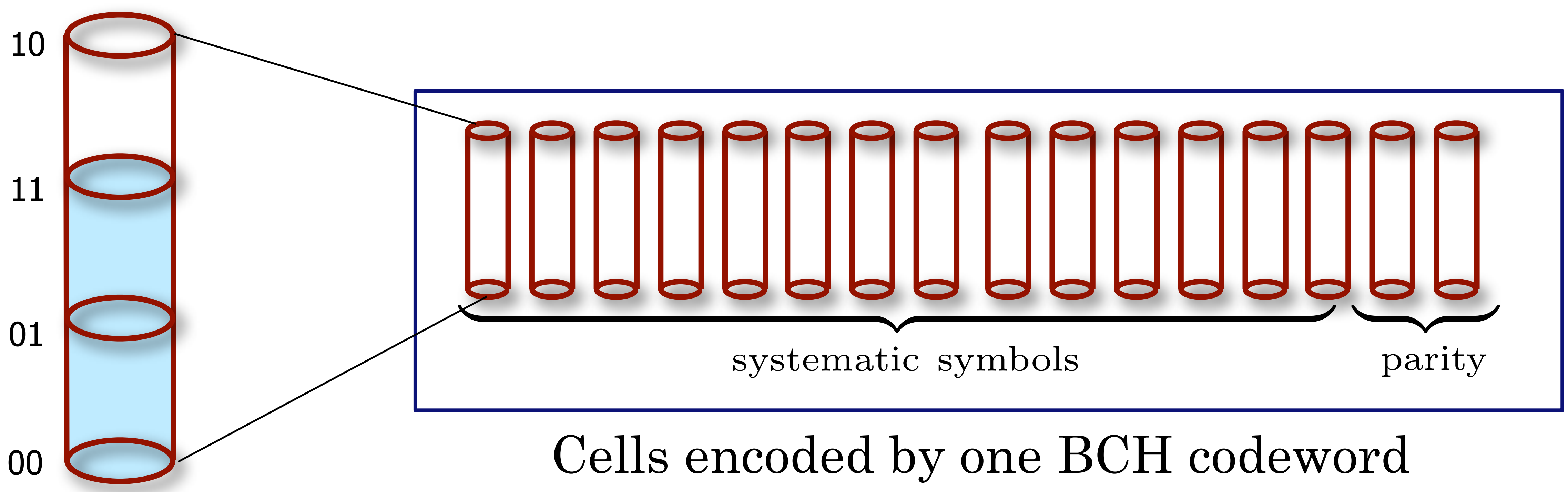
For PAM with AWGN model likely error is a shift to adjacent symbol

- Gray coding is used

Errors are random and uncorrelated

- BCH codes are used
- Error-correction better than Reed-Solomon

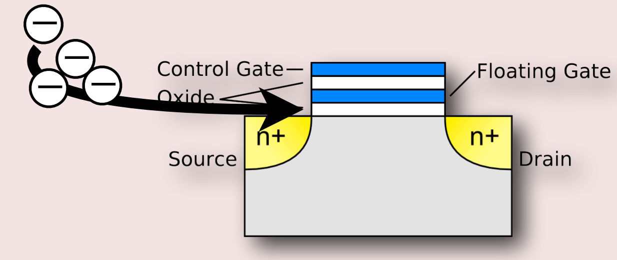
Codes rates tend to be very high, 0.95 to 0.99



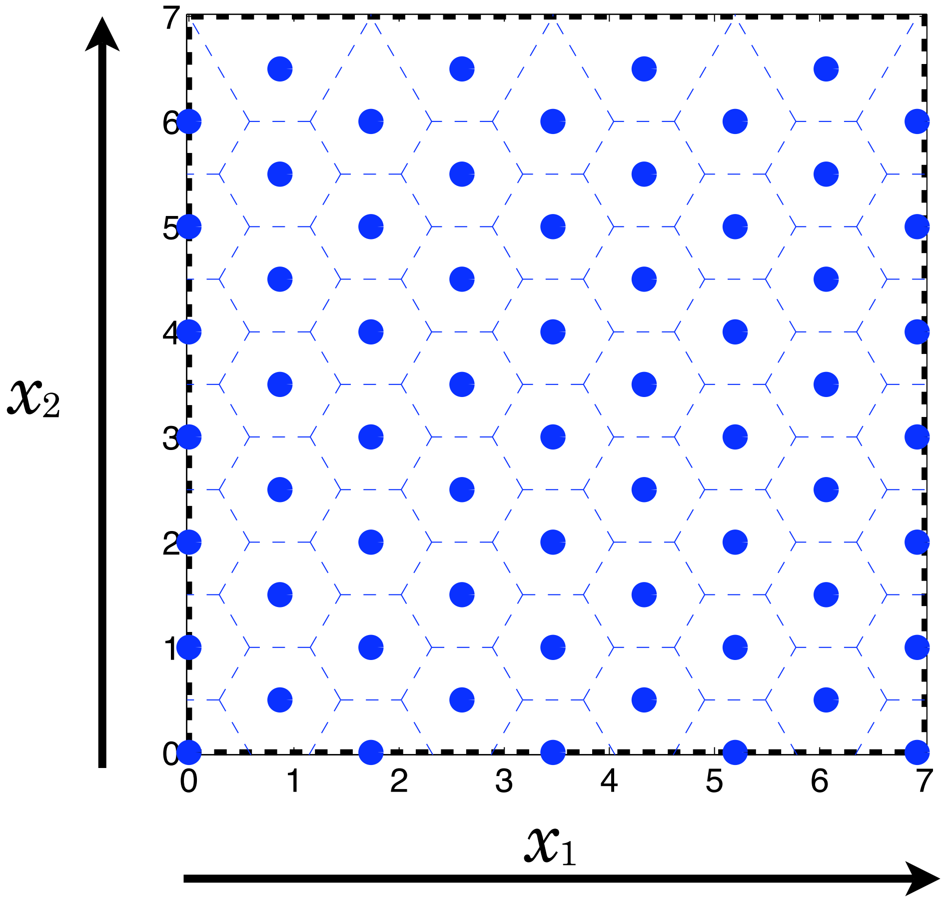
Cells encoded by one BCH codeword

One 4-level cell

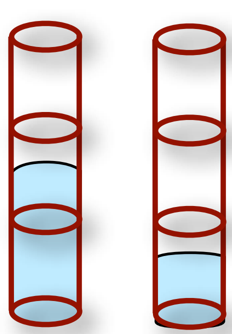
Lattices for Flash



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



continuous values 0 to V



x_1 x_2

$$G = \begin{bmatrix} 1 & 0 \\ \frac{\sqrt{3}}{2} & 1 \end{bmatrix}$$

Lattices are "codes" over real numbers

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

$\mathbf{x}$: lattice point; G: matrix; $\mathbf{b}$: integer vector

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.

E8 Lattice (Gosset Lattice)

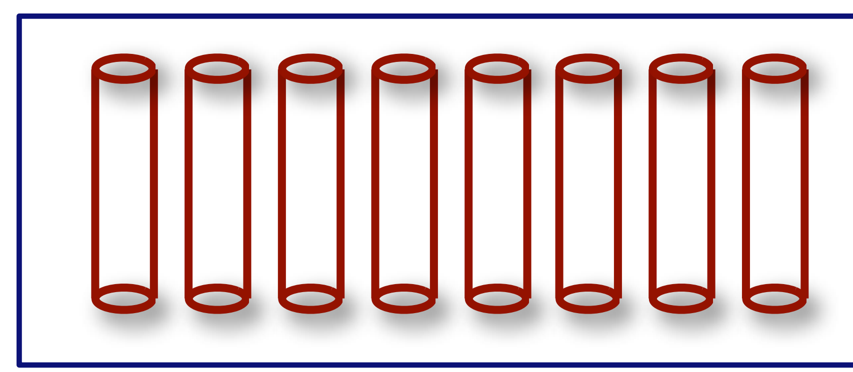
- Best-known lattice in eight dimensions
- lattice $d_{\min}$ is $\sqrt{2}$ =1.414
  - versus $d_{\min} = 1$ for PAM
- Efficient soft-input decoding algorithm
- Coordinates are integers or half integers
- kissing number: 240 nearest neighbors at $d_{\min}$
  - versus 6 for this hex lattice

$$\begin{bmatrix} \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2} & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2} & -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2} & 0 & -1 & 1 & 0 & 0 & 0 & 0 \\ \frac{1}{2} & 0 & 0 & -1 & 1 & 0 & 0 & 0 \\ \frac{1}{2} & 0 & 0 & 0 & -1 & 1 & 0 & 0 \\ \frac{1}{2} & 0 & 0 & 0 & 0 & -1 & 1 & 0 \\ \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & -1 & 2 \end{bmatrix}$$

E8 generator matrix

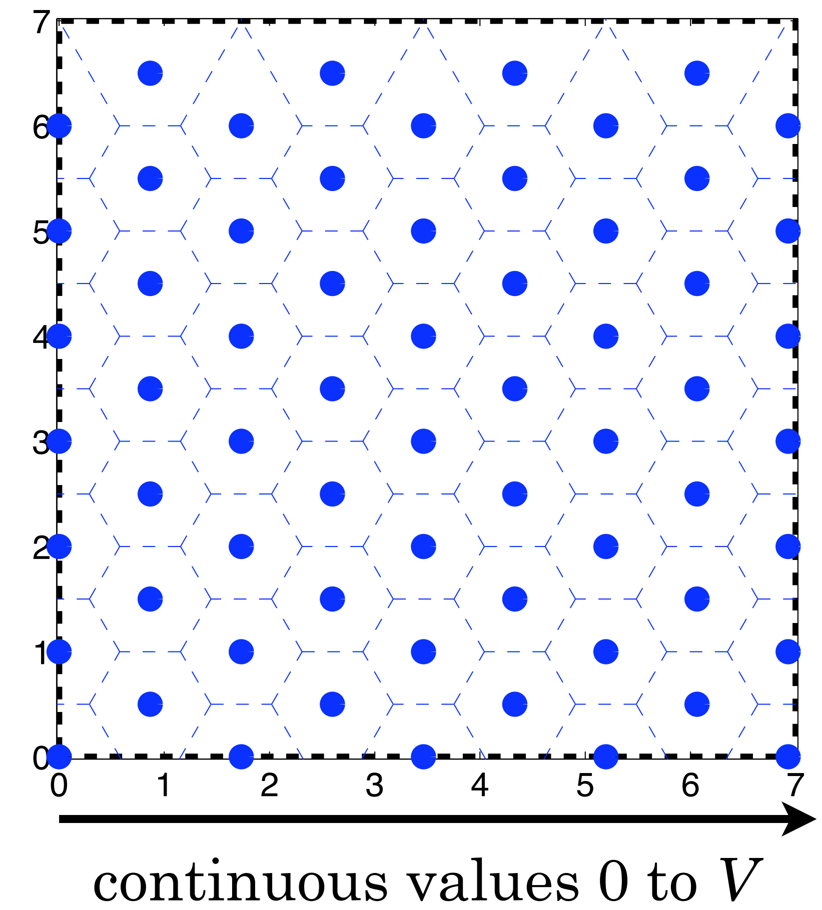
Storing Lattice Point in Flash

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



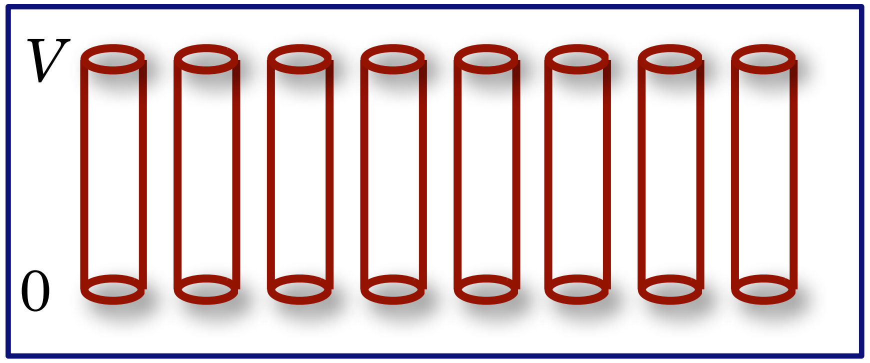
E8 Lattice (Gosset Lattice)

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

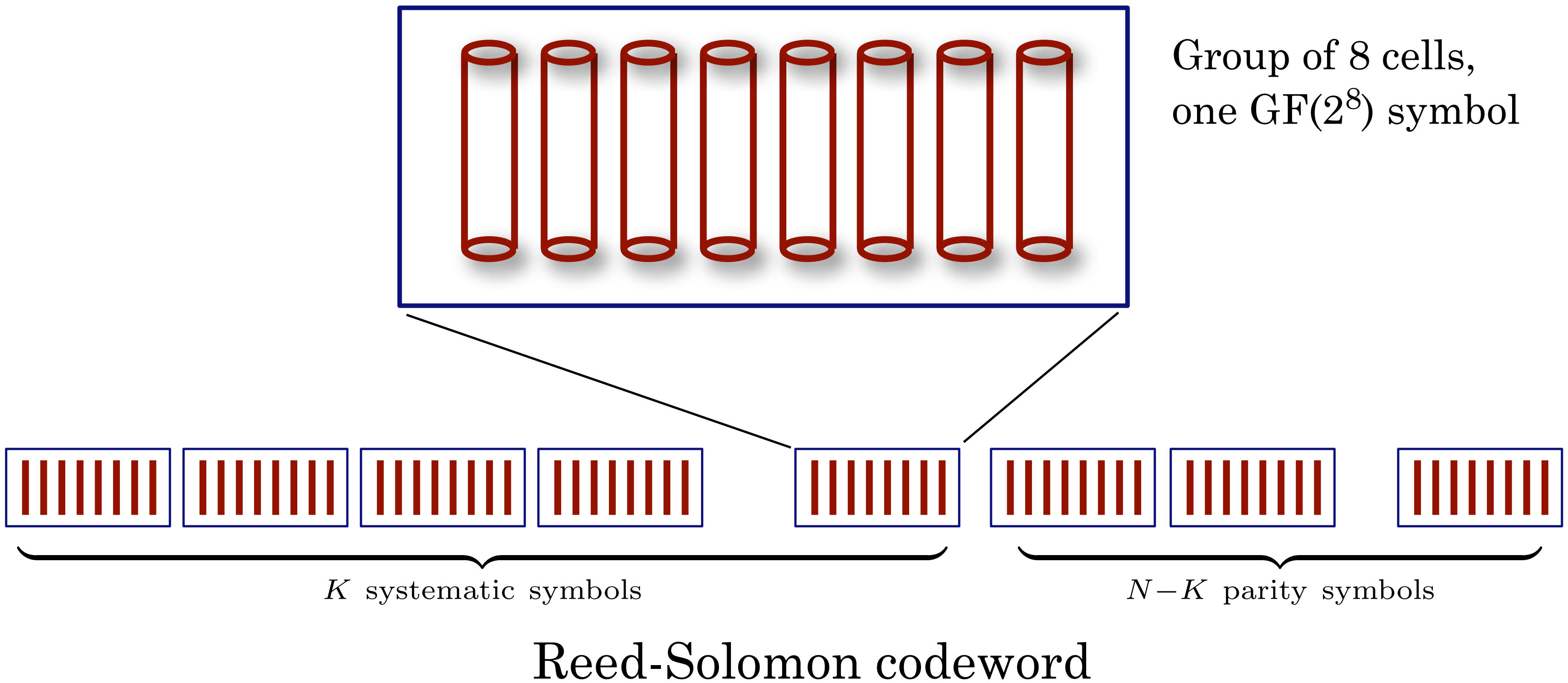


Storing Lattice Point in Flash

- 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

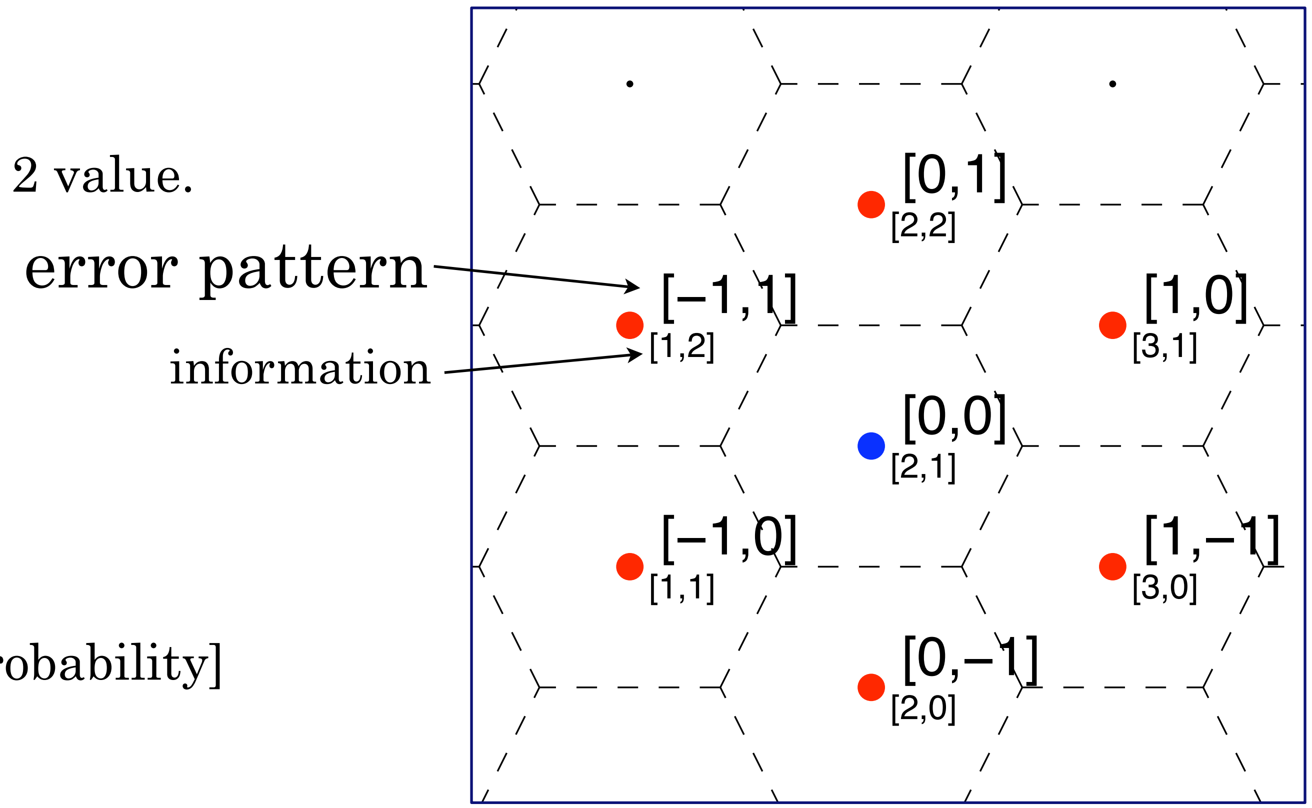


- Block of 8 cells corresponds to 1 Reed-Solomon symbol
- Use an (N,K) RS code
- Use a RS code over GF(2^8) - 8 bits symbols
  - Each block has $\log_2(V^8)$ information bits,
  - BUT, is only protected by 8 bits from the Reed-Solomon codeword
  - SO, the modulo-2 information value is the Reed-Solomon 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



A hex lattice point has 6 neighbors.

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

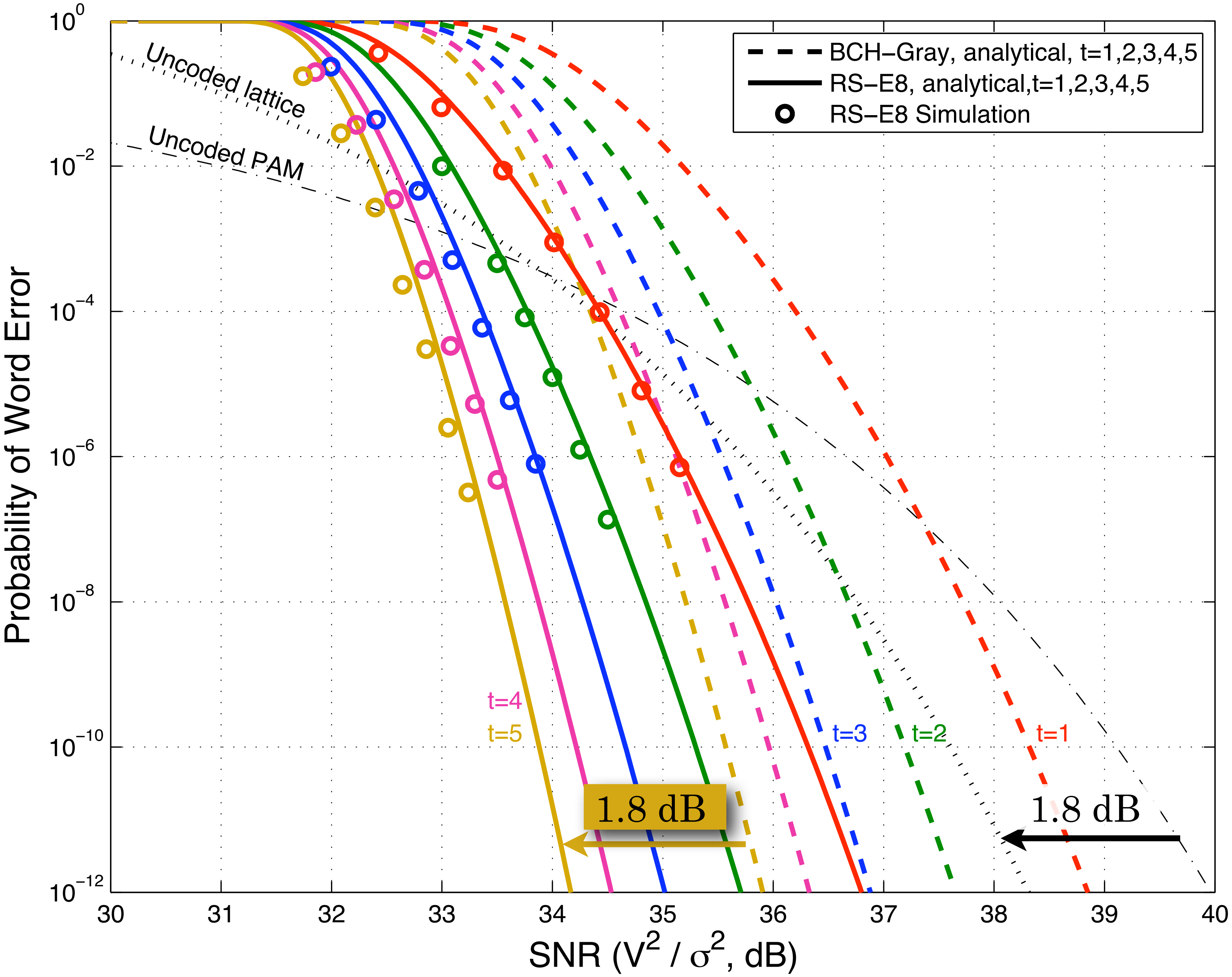
- Winner: shortest Euclidean distance to received sequence

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

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)

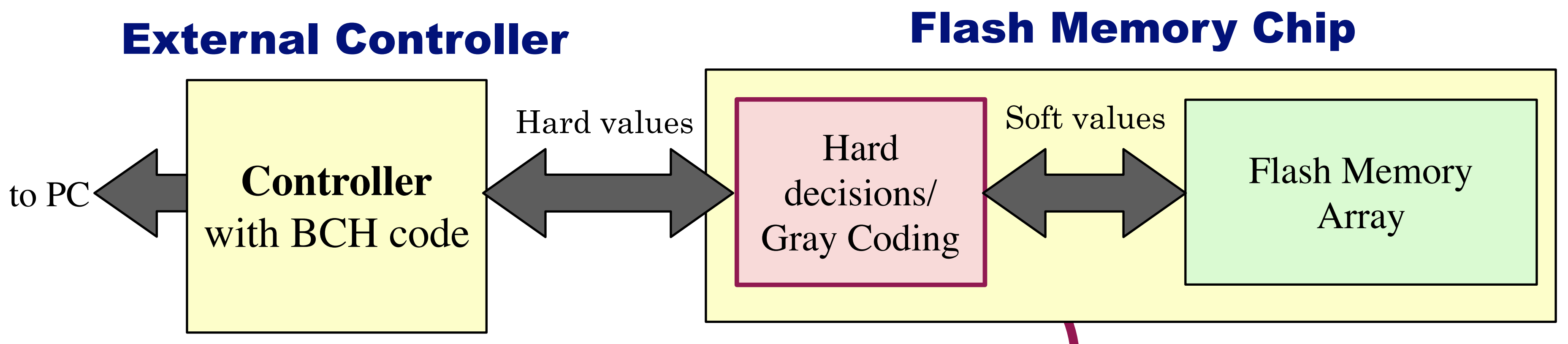
Uncoded E8 lattice has 1.8 dB gain over uncoded PAM

This gain is preserved when the codes are added

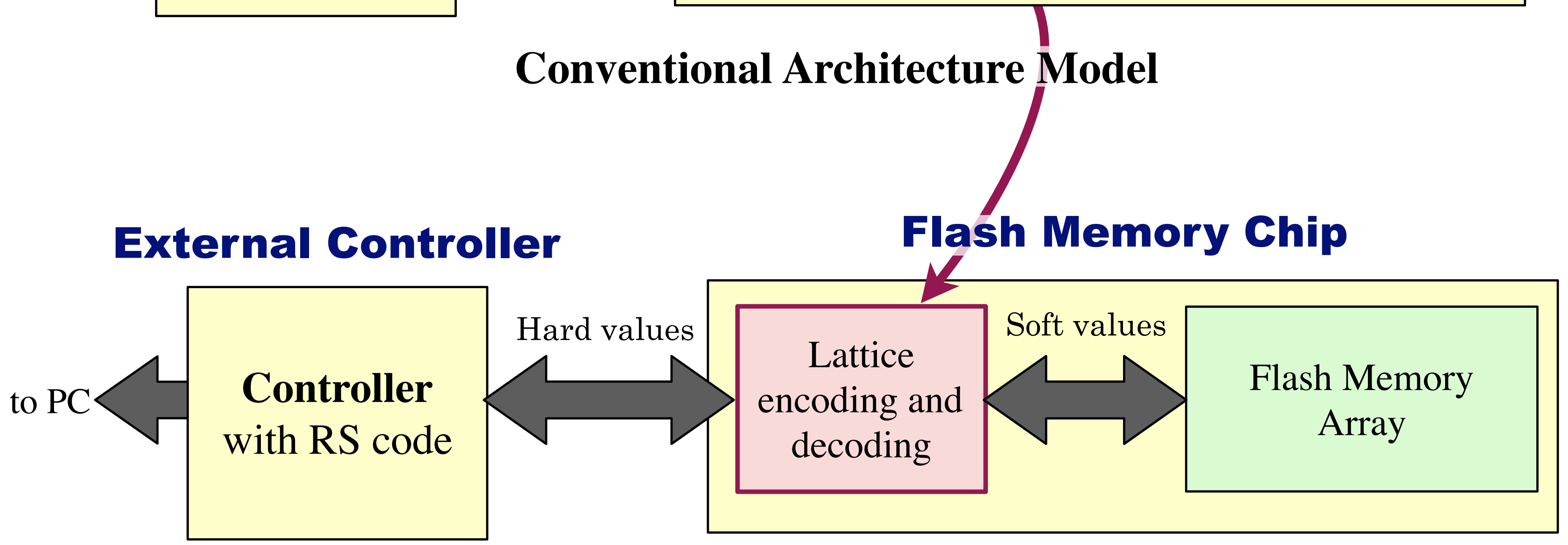
Truth in advertising:

- E8 is a soft-input algorithm
- PAM is hard decisions
- Not unfair ....

Flash Memory: Architectural Considerations



Conventional Architecture Model



Soft-Input Architecture Model

- 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

"Reptiles"

M.C. Escher

Decoding Complexity for E8 Lattice

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 $-1 \leq y_i < 3$.
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 $\mathbf{y}$ 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 $\mathbf{c} + 4\mathbf{z}$.