Writing More Than Once on a Write-Once Memory

Paul H. Siegel

Electrical and Computer Engineering
Jacobs School of Engineering
and
Center for Magnetic Recording Research
University of California, San Diego

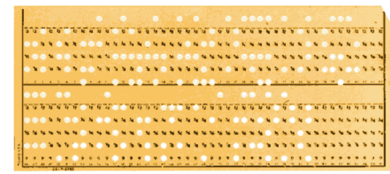
June 11, 2012

Outline

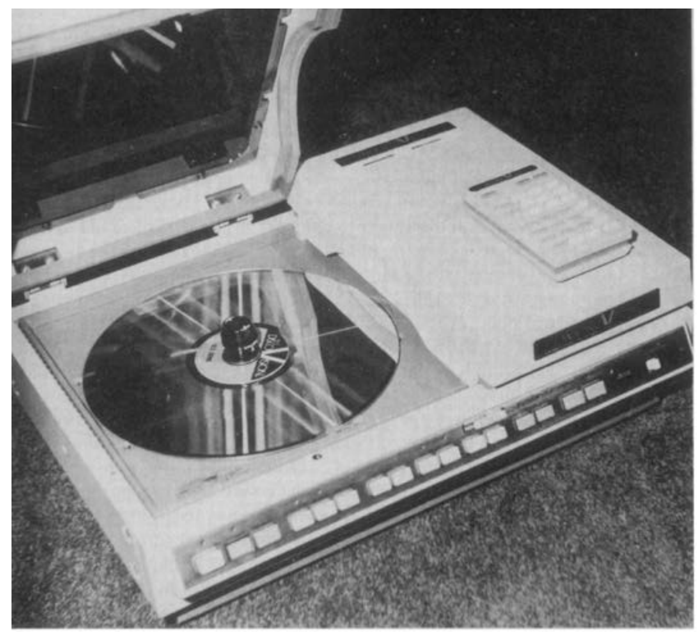
- Write-Once-Memory (WOM) model
- WOM codes: How to re-use a WOM
- Binary WOM codes
  - Constructions and bounds
- Non-binary WOM codes
  - Constructions and bounds
- Concluding remarks

Motivation - 1982

- Punch cards
- Optical disks



LaserDisc™ DVD™



Optical Disk Recording

SCIENCE, VOL. 215, 12 FEBRUARY 1982

Optical Disk Technology and Information

Charles M. Goldstein

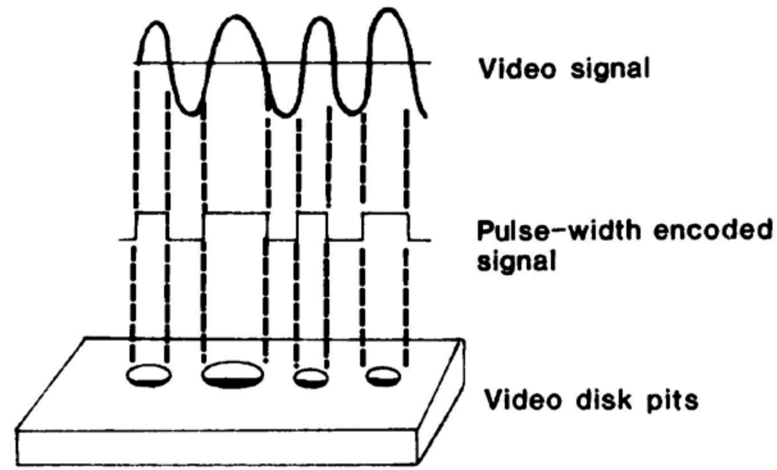


Fig. 1. Pulse-width encoding.

Can a previously recorded optical disk be re-used?

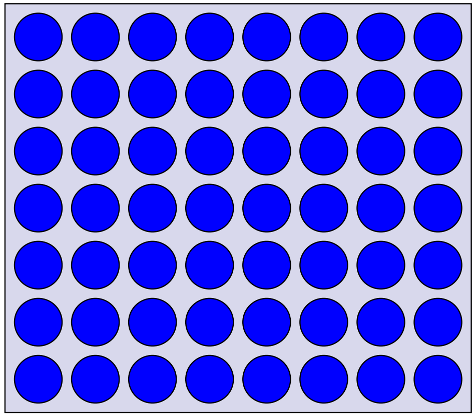
Motivation - 2012

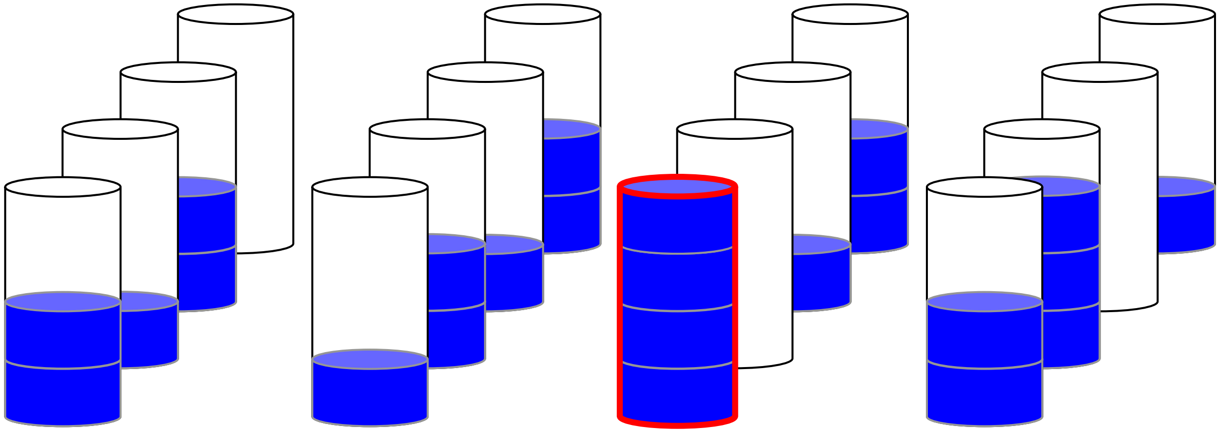
- Flash memory

- A flash memory "block" is an array of $\sim 2^{20}$ "cells".
- A cell is a floating-gate transistor with q "levels". corresponding to the voltage induced by the number of electrons stored on the gate.
- Terminology:
  - Single-level cell (SLC) stores 1 bit per cell ($q=2$)
  - Multi-level cell (MLC) stores 2 (or more) bits per cell ($q=4$ or more).

Flash Programming

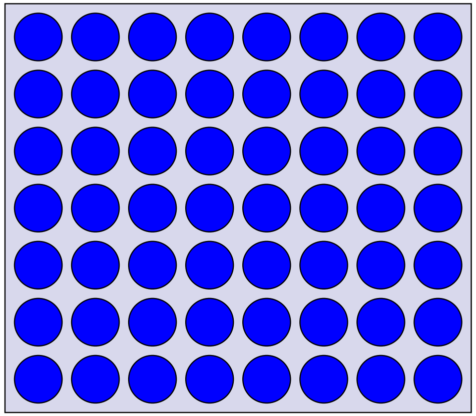
- To increase a cell level, you just add more electrons.
- To reduce the cell level, you must first erase the entire block of cells and then reprogram the block to reflect the updated data.

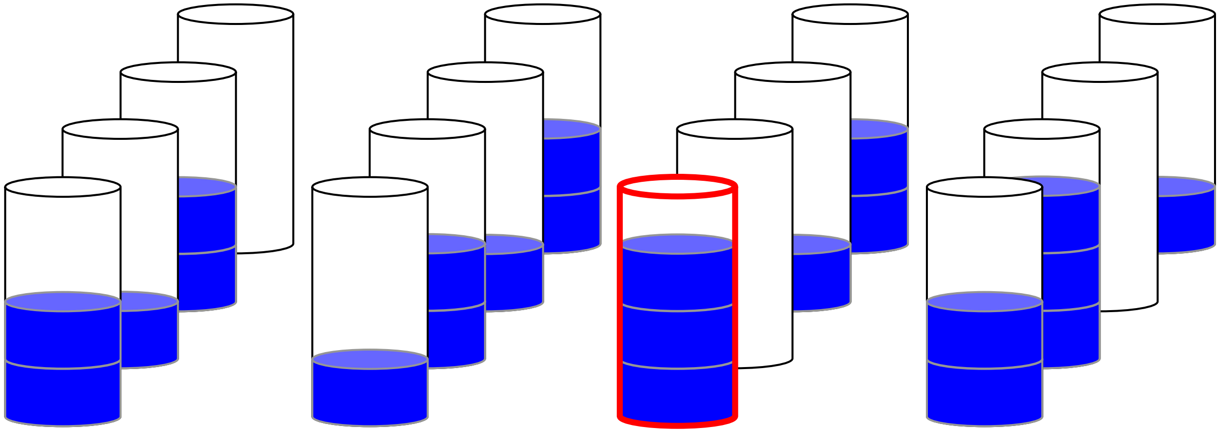




Flash Programming

- To increase a cell level, you just add more electrons.
- To reduce the cell level, you must first erase the entire block of cells and then reprogram the block to reflect the updated data.





Flash Memory Endurance

- Block erasure degrades the flash memory cells.
- Flash memory endurance (also called lifetime) is measured in terms of the number of program and erase (P/E) cycles it tolerates before failure.
- SLC flash memory lifetime is $\sim 10^5$ P/E cycles.
- MLC flash memory lifetime is $\sim 10^4$ P/E cycles.

Can new data be written to a flash memory cell without first erasing the entire block?

Write-Once Memory (WOM) Model

- Introduced in 1982 by Rivest and Shamir
- An array of “write-once bits” (or wits) with 2 possible values: 0 and 1.
- Initial state of every wit is 0.
- Each wit can be **irreversibly** programmed to 1.

Can a WOM be rewritten?

How to Reuse a "Write-Once" Memory*

RONALD L. RIVEST

MIT Laboratory for Computer Science, Cambridge, Massachusetts

AND

ADI SHAMIR

Weizmann Institute of Science, Rehovot, Israel

"TROUBLE HIM NOT; HIS WITS ARE GONE." KING LEAR, III.vi.89

***Information and Control* , vol. 55 nos. 1-3, December 1982**

The Mother of all WOM codes

"Only 3 wits are needed to write 2 bits twice"

Data	1st Write	2nd Write
00	000	111
01	100	011
10	010	101
11	001	110

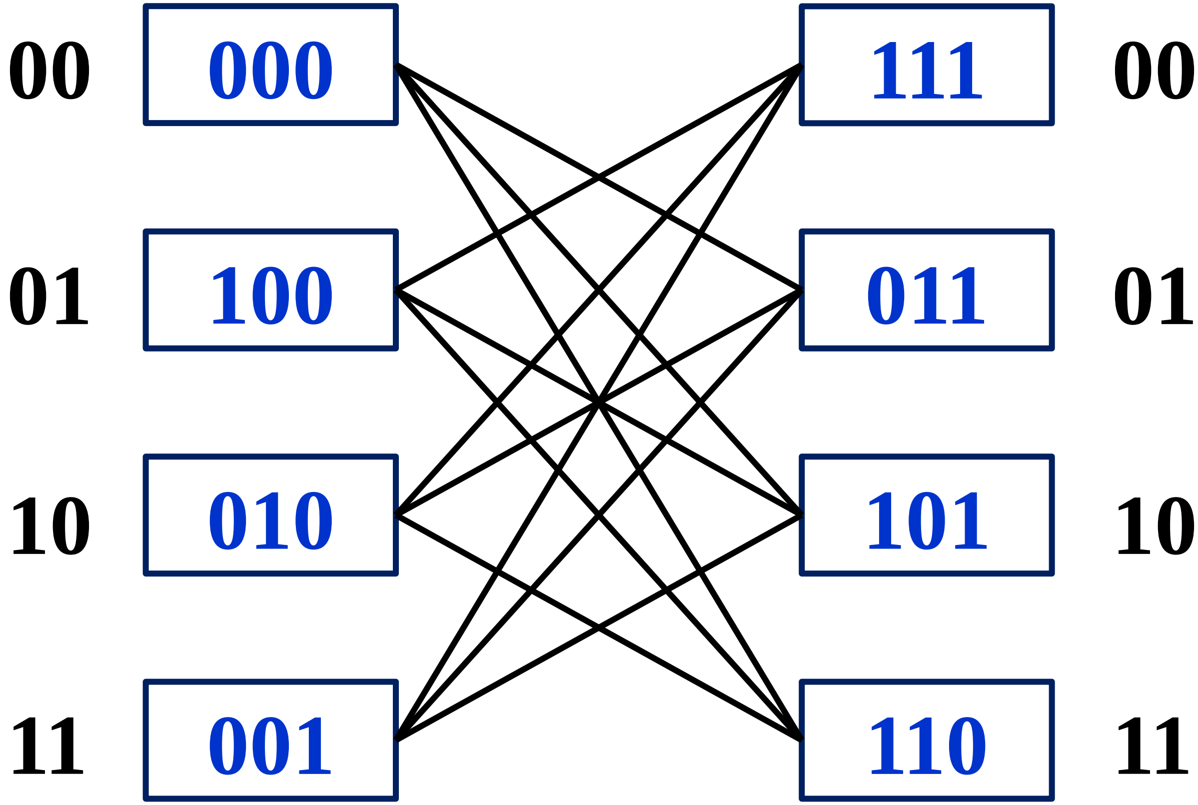
Encoding and Decoding

Data	1st Write	2nd Write
00	000	111
01	100	011
10	010	101
11	001	110

- 1st write:
  - Encode 2-bit word using 1st-write codebook.
- 2nd write:
  - Decode first 2-bit woes from the written codeword.
  - If new 2-bit word is the same as the first, there is no change to the written codeword.
  - If new 2-bit word is different, encode using 2nd-write codebook. This never changes a written 1 to a 0.
- Decoding: Each codeword is associated with a unique 2-bit data pattern.

Another Representation

- No 2nd write codeword changes a 1 to a 0.



Binary WOM Codes

- An $< M_1,\ldots, M_t >/n$ binary WOM code is a coding scheme that guarantees any sequence of t writes using alphabet sizes $M_1,\ldots, M_t$ on n cells.
- We consider two cases
  – unrestricted rate: $M_1,\ldots, M_t$ may differ.
  – fixed rate: $M = M_1 = \ldots = M_t$.
- Rivest-Shamir code is a $< 4,4 >/3$ WOM code.

<26,26> / 7 Binary WOM Code [RS82]

- Stores 26 messages twice in 7 binary cells

	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	*A*	*H*	*G*	*G*	*F*	*Y*	*L*	*w*	*E*	*Z*	*Y*	*r*	*X*	*f*	*p*	*n*	*D*	*W*	*V*	*z*	*U*	*d*	*j*	*o*	*T*	*w*	*k*	*e*	*l*	*t*	*d*	*u*
1	*C*	*S*	*R*	*c*	*Q*	*i*	*o*	*z*	*P*	*p*	*i*	*h*	*u*	*e*	*x*	*y*	*O*	*z*	*s*	*j*	*s*	*n*	*i*	*w*	*v*	*c*	*q*	*g*	*f*	*k*	*b*	*m*
2	*B*	*N*	*M*	*z*	*L*	*b*	*g*	*m*	*K*	*u*	*t*	*b*	*n*	*g*	*f*	*w*	*J*	*w*	*r*	*h*	*k*	*v*	*x*	*y*	*m*	*j*	*p*	*s*	*o*	*q*	*c*	*i*
3	*I*	*k*	*m*	*q*	*l*	*c*	*k*	*u*	*w*	*t*	*e*	*o*	*s*	*d*	*j*	*v*	*u*	*b*	*d*	*f*	*g*	*e*	*t*	*p*	*y*	*x*	*n*	*l*	*h*	*r*	*z*	*a*

- Letter in row I, column J stored as the 7-bit binary representation of I*32+J
- 1st write: Upper case letters
- 2nd write: Lower case letters

WOM Code Sum-Rate

- For an $< M_1,\ldots, M_t > / n$ binary WOM code the *sum-rate* R is the total number of bits stored per cell in all t writes
- Thus,

$$R = \sum_{i=1}^{t} R_i$$

where

$$R_i = \frac{\log_2 M_i}{n}$$

Examples

- $<4,4>/3$ WOM code

$$R = R_1 + R_2 = 2\frac{\log_2 2^2}{3} = \frac{4}{3} \approx 1.3333$$

- $<26,26>/7$ WOM code

$$R = R_1 + R_2 = 2\frac{\log_2 26}{7} \approx 1.3429$$

What is the largest achievable sum-rate for a ***t***-write WOM code on ***n*** cells?

Achievable rate region

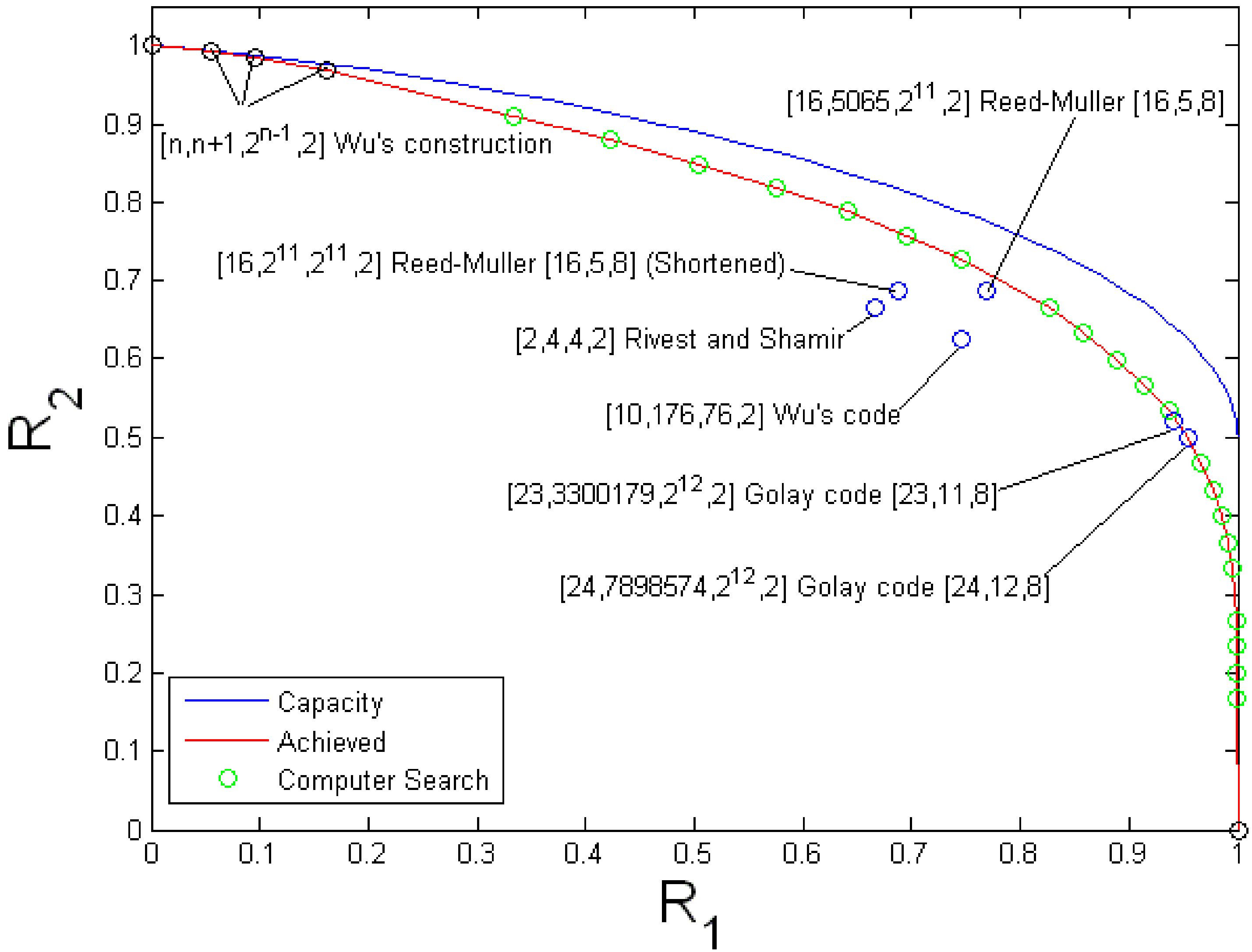
[Heegard 1986, Fu and Han Vinck 1999]

- For a binary WOM the ***t***-write achievable rate region is given by:

$$
\begin{aligned}
R^{(t)} = \Big\{ (R_1, \ldots, R_t) \,\Big|\, & R_1 \le h(p_1), \\
& R_2 \le (1-p_1)h(p_2), \ldots, \\
& R_{t-1} \le \left(\prod_{i=1}^{t-2} (1-p_i) \right) h(p_{t-1}), \\
& R_t \le \prod_{i=1}^{t-1} (1-p_i) \text{ where } 0 \le p_1, \ldots, p_{t-1} \le 1/2 \Big\}.
\end{aligned}
$$

$$
\left[h(p) = -p \log_2 p - (1-p) \log_2 (1-p) \right]
$$

Achievable region: 2-write WOM codes



WOM Capacity

- The *unrestricted-rate capacity* $C^{(t)}$ of a t-write binary WOM is the maximum of the achievable sum-rates.
  - It has been shown that $\mathbf{C^{(t)} = \log_2(t+1)}$.
- The *fixed-rate capacity* $C_0^{(t)}$ of a t-write binary WOM does not have a simple expression, but can be computed recursively.

Capacity: 2-write WOM

- The unrestricted-rate capacity of 2-write binary WOM is:

$$C^{(2)} = \max_{(R_1,R_2)\in R^{(2)}} \left(R_1 + R_2\right)$$
$$= \max_{p\in[0,\frac{1}{2}]} \left(h(p) + (1-p)\right)$$

- This sum is maximized when p=1/3, implying

$$R_1 \approx 0.918296,\ R_2 = 2/3$$

$$C^{(2)} = \log_2 3 \approx 1.5849$$

Fixed-rate Capacity: 2-write WOM

- The fixed-rate capacity of a 2-write binary WOM is:

$$C_0^{(2)} = h(p^*) + (1 - p^*)$$

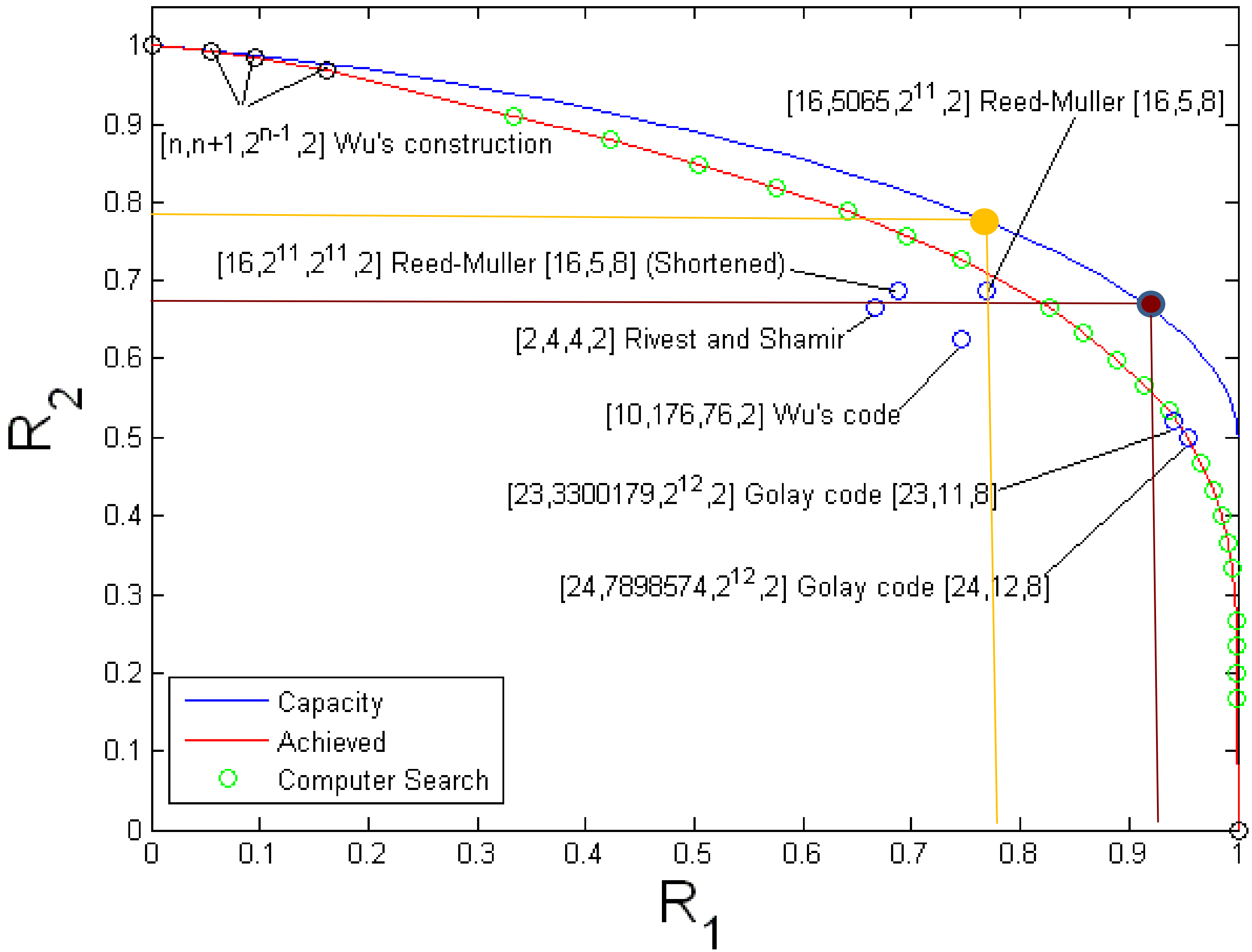
where $p^* \approx 0.227$ satisfies

$$h(p^*) = (1 - p^*)$$

- This implies $R_1 = R_2 \approx 0.773$

$$C_0^{(2)} \approx 1.5458.$$

Achievable region: 2-write WOM codes



Coset Coding Construction
[Cohen, Godlewski, and Merxx 1986]

- Let $C[n,k]$ be a binary linear block code with parity-check matrix H.
- 1st write: Write a "syndrome" s_1 of $r=n-k$ bits by means of a low-weight "error vector" y_1 such that $H\cdot y_1 = s_1$.
- 2nd write: Write another "syndrome" s_2 of r bits by finding (if possible) a vector y'_2 not "overlapping" y_1 such that $H\cdot y'_2 = s_1+s_2$.
- Write $y_2= y_1+ y'_2$ and decode using

$$H\cdot y_2=H\cdot (y_1+ y'_2) = s_1 + (s_1+s_2) = s_2$$

<4,4>/3 as a "Coset" WOM Code

- Let $C[n,k]$ be the binary 3-repetition code with parity-check matrix

$$H = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}$$

- 2-bit "syndromes" $s_1 = 00, 01, 10, 11$ correspond to vectors $y_1 = 000, 100, 010, 001.$
- All 2 x 2 submatrices of H are invertible, so, given y_1 we can find non-overlapping vector y'_2 such that $H \cdot y'_2 = s_1 + s_2$, and then write $y_2 = y_1 + y'_2$.

<4,4>/3 Coset Coding Example

$$H = \begin{bmatrix} 0 & \boxed{\begin{matrix} 1 & 1 \\ 0 & 1 \end{matrix}} \\ 1 & \end{bmatrix}$$

- 1^{st} write: Encode $s_1 = 01$ into $y_1 = 100$
  satisfying $H \cdot y_1 = s_1$
- 2^{nd} write: Decode $y_1 = 100$ to $s_1 = 01.$
  Encode $s_2 = 10$ by finding non-overlapping y'_2 such that
  $H \cdot y'_2 = s_1 + s_2 = 11,$ namely $y'_2 = 001$
  and writing $y_2 = y_1 + y'_2 = 100+001=101.$

[Note: 101 correctly decodes to $s_2 = 10$]

Generalized Coset Coding

[Wu 2010, Yaakobi et al. 2010]

- Let $\mathbf{C[n,n\text{-}r]}$ be a code with $r \times n$ parity-check matrix H.
- For a vector $v \in \{0,1\}^n$, let H_v be the matrix H with $\mathbf{0}$'s in the columns that correspond to the positions of the $\mathbf{1}$'s in v.
- **1st Write**: write a vector $v_1 \in V_C = \{ v \in \{0,1\}^n \mid \text{rank}(H_v) = r\}$.
- **2nd Write**: Write an r-bit vector s_2 as follows:
  - Decode $s_1 = H \cdot v_1$
  - Find non-overlapping v'_2 with $H \cdot v'_2 = s_1 + s_2$

    (possible because $\text{rank}(H_{v_1}) = r$).
  - Write $v_2 = v_1 + v'_2$ to memory.
- **Decoding**: Compute $H \cdot v_2 = H \cdot (v_1 + v'_2) = s_1 + (s_1 + s_2) = s_2$.
- [Note: The set V_C is independent of the choice of H.]

Sum-Rate Results

- The construction works for **any** code $C[n,k]$.
- The rate of the first write is:

$$R_1(\mathrm{C}) = (\ \log_2|V_C|\ )\ /n$$

- The rate of the second write is:

$$R_2(\mathrm{C}) = r/n$$

- Thus, the sum-rate is:

$$R(C) = (\log_2|V_C| + r)/n$$

- **Goal**: Choose a code ***C*** to maximize $R(C)$.

Specific Constructions

- [n,k,d]=[16,5,8] first-order Reed-Muller code:

  $|\boldsymbol{V}_C| = 5065$, $(R_1, R_1)=(0.7691, 0.6875)$, so $\boxed{R \approx 1.4566.}$

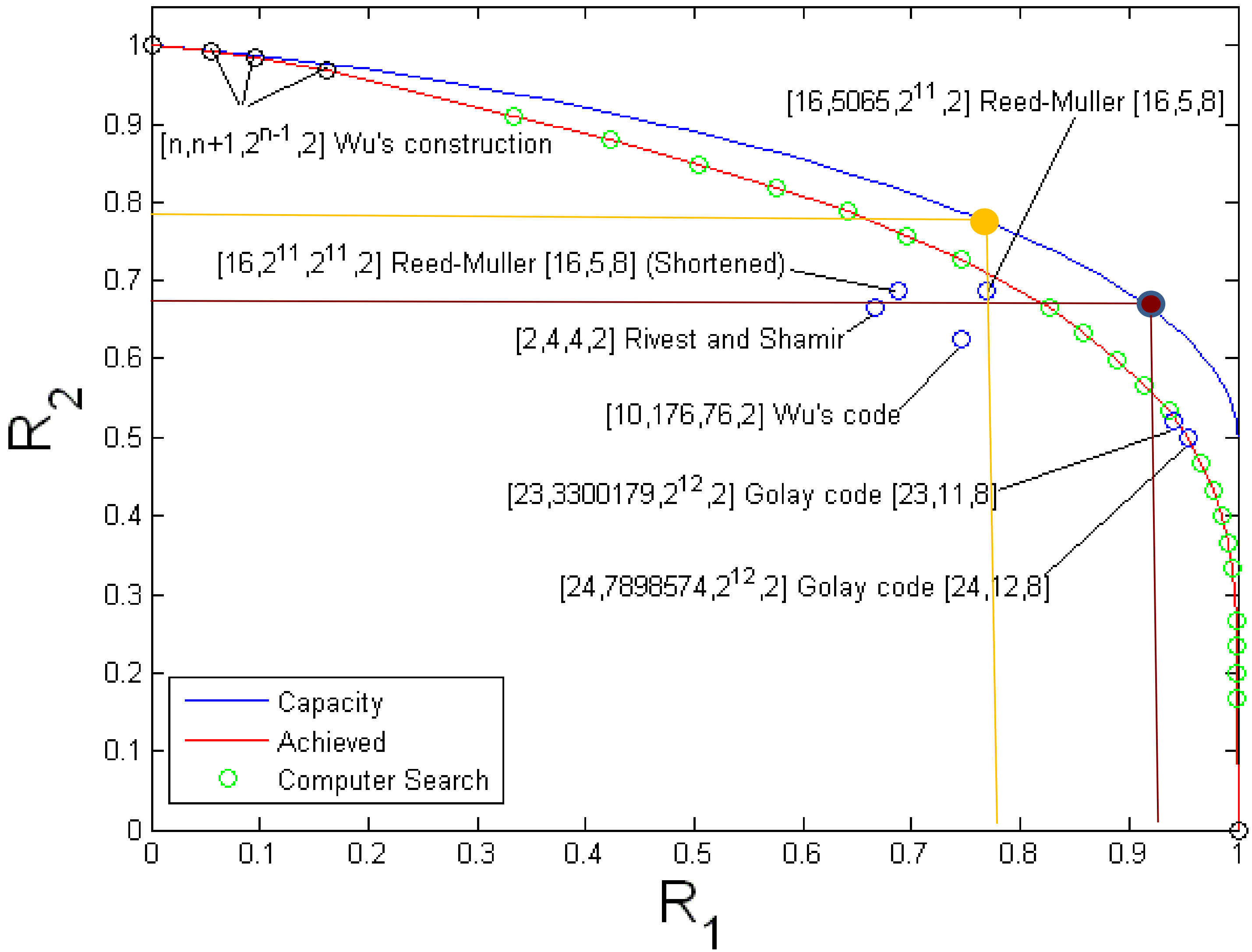
  Restricting first write to $2^{11} < 5065$ messages yields a fixed-rate code with $R_1 = R_2 = 11/16$, so $\boxed{R \approx 1.375.}$

- [n,k,d]=[23,12,8] Golay code:

  $|\boldsymbol{V}_C| = 3300179$, $(R_1, R_1)=(0.9415, 0.5217)$, so $\boxed{R \approx 1.4632.}$

- Previous best constructions:
  - Fixed-rate: R-S <26, 26>/7 with $R \approx 1.34$
  - Unrestricted rate: Wu <176,76>/10 with $R \approx 1.371$

Achievable region: 2-write WOM codes



Random codes and WOM Capacity
[Yaakobi et al. 2010 and Wu 2010]

- Recall: The 2-write achievable rate region is

$$R^{(2)} = \{(R_1, R_2) | \, R_1 \le h(p),\, R_2 \le 1-p,$$
$$\text{for } 0 \le p \le 1/2\}.$$

- **Theorem**: For any $(R_1, R_2) \in R^{(2)}$ and $\varepsilon > 0$, there exists a linear code C satisfying

$$R_1(C) \ge R_1 - \varepsilon \quad \text{and} \quad R_2(C) \ge R_2 - \varepsilon$$

Proof: Use the "coset coding" construction with a randomly chosen $(n-k)$ x k parity-check matrix with $k = \lceil np \rceil$ where $R_1 \le h(p),\, R_2 \le 1-p$.

Computer search results

- Computer search using "randomly" chosen H.
  - Best unrestricted-rate WOM code (22x33):

$$< M_1 M_2 > / n = < 2^{(33\times 0.8261)}, 2^{22} > / 33$$

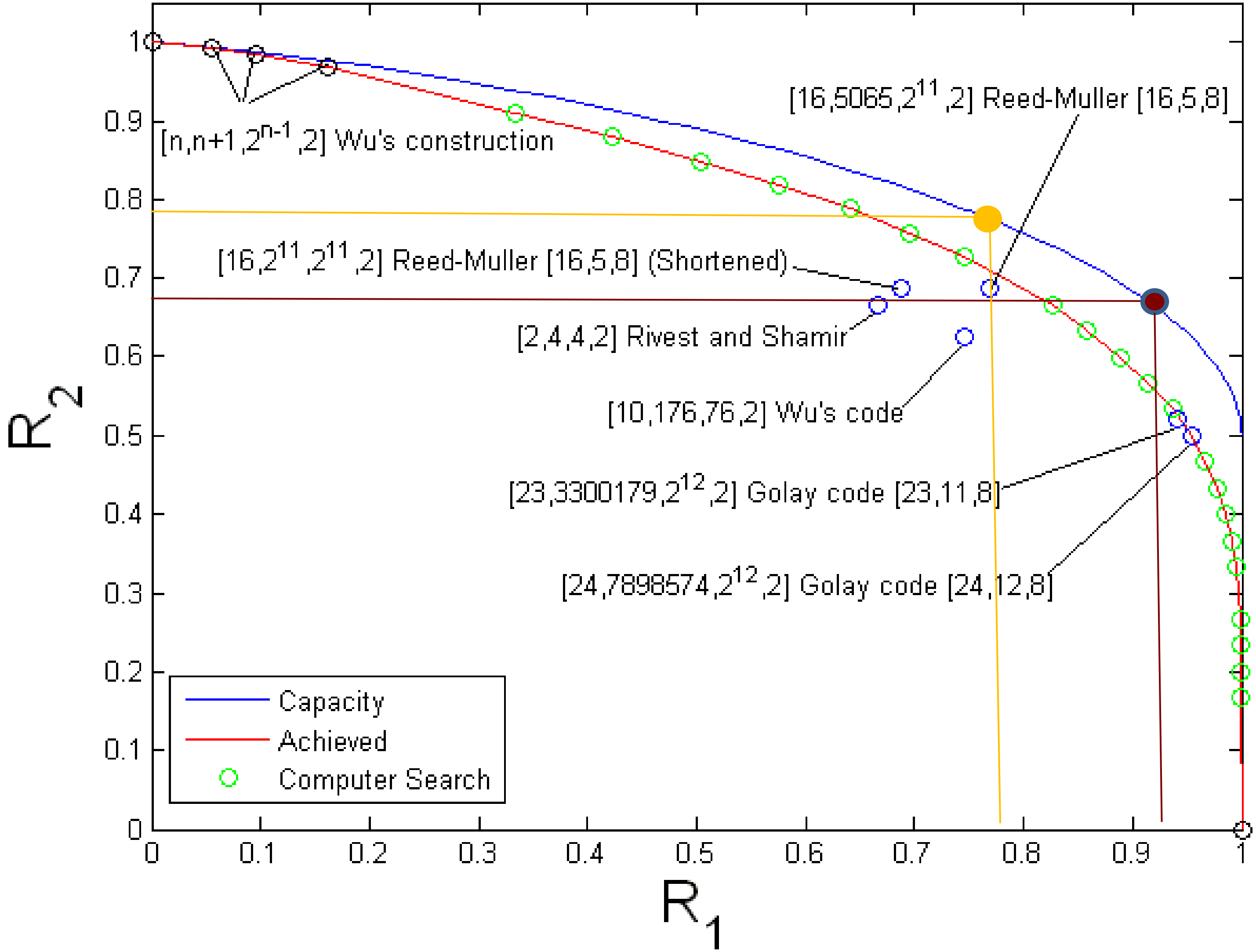
$$\boxed{R \approx 1.4928}$$

  - Best fixed-rate WOM code (24x33):

$$< M_1 M_2 > / n = < 2^{24}, 2^{24} > / 33$$

$$\boxed{R \approx 1.4546}$$

Rate Region and Code Constructions



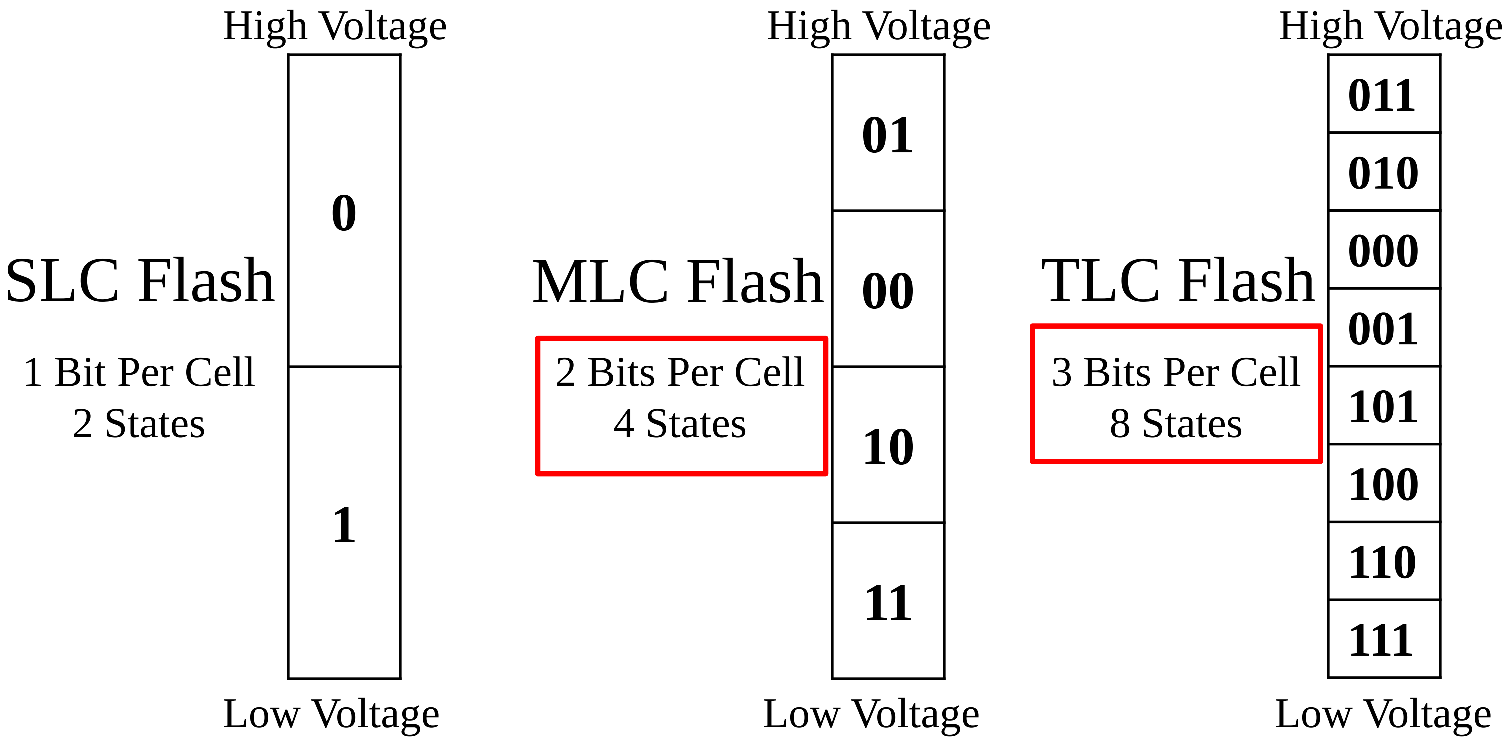
Capacity-achieving 2-write codes
[Shpilka 2012]

- Efficient capacity-achieving construction based upon modified "coset coding".
  - 1^{st} write: Program any binary vector of weight at most m (fixed).
  - 2^{nd} write: Use a set of matrices (derived using the Wozencraft code ensemble) such that at least one of them succeeds on the second write.

3-write Binary WOM Codes

- Recall that $\boldsymbol{C^{(3)} = \log_2(3+1)=2}$.
- [Kayser et al. 2010]: General construction based upon 2-write ternary WOM code, yielding sum–rate $\boxed{R \approx 1.61}$ (with $R \approx 1.66$ the best it can achieve).
- [Shpilka 2011]: Construction based upon efficient 2-write WOM codes, yielding sum-rate $\boxed{R \approx 1.8.}$
- [Yaakobi & Shplika 2012]: Further refinements leading to sum-rate $\boxed{R \approx 1.88.}$

WOM codes for Non-Binary Flash



Non-Binary WOM-Codes

- Each cell has q levels $\{0,1,\ldots,q\text{-}1\}$.
- The achievable rate region of non-binary WOM-codes was given by Fu and Han Vinck, 1999.
- The maximal sum-rate of a ***t*-**writes, ***q***-ary WOM code is

$$C = \log\binom{t+q-1}{q-1}$$

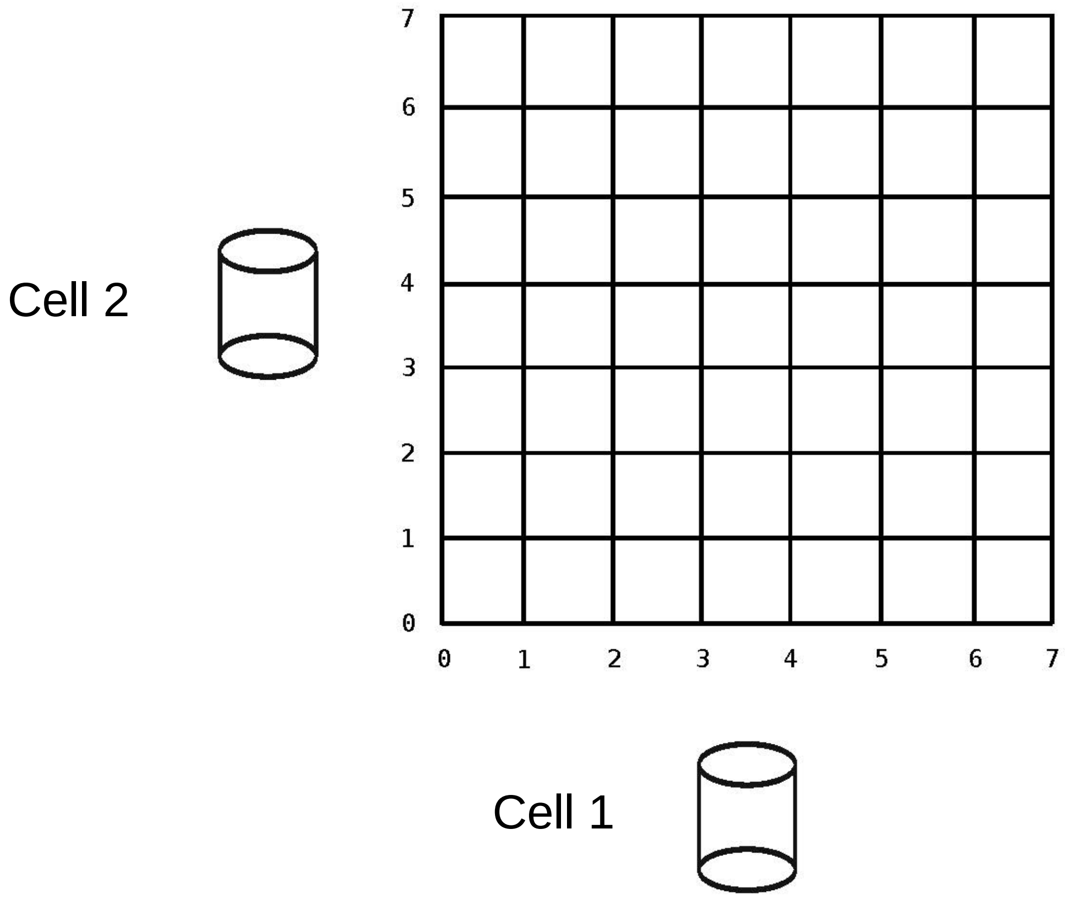
- Random "partition" coding achieves capacity.
- Recent works give specific code designs.

Lattice-based q-ary WOM Codes

[Kurkoski 2012, Bhatia et al. 2012]

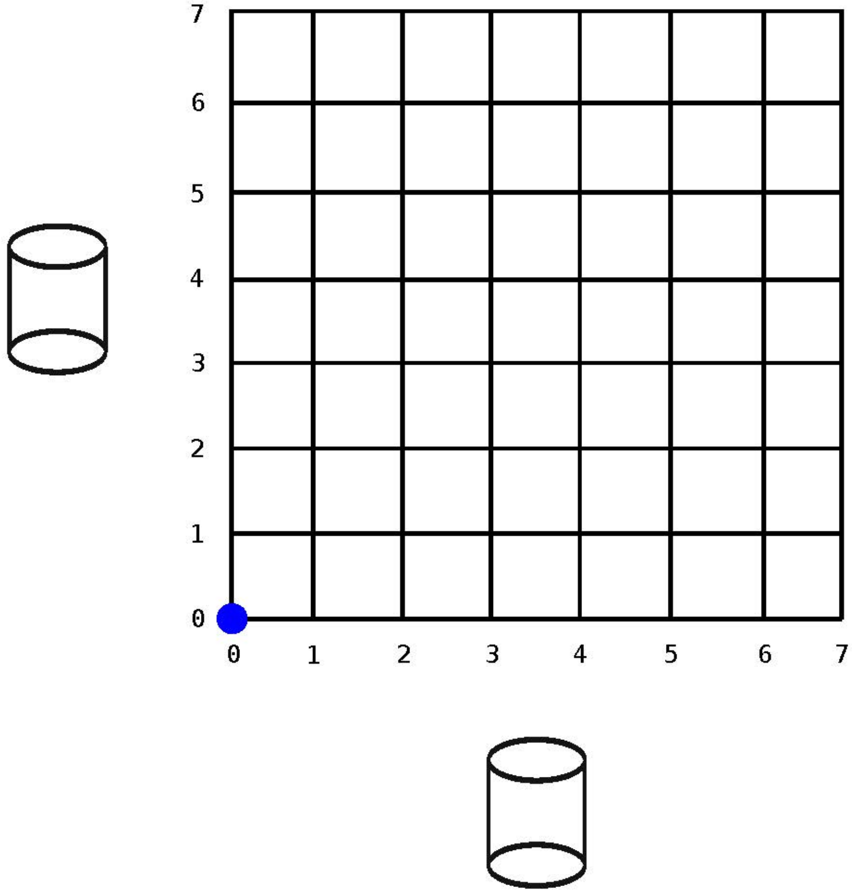
- Lattice-based WOM codes for multi-level flash provides a possible way to combine increased endurance with error resilience.
- Techniques developed for lattice-based data modems can be applied in the design of WOM codes with worst-case optimal sum-rates.
- The key tool is "continuous approximation".

2-cell 8-level WOM



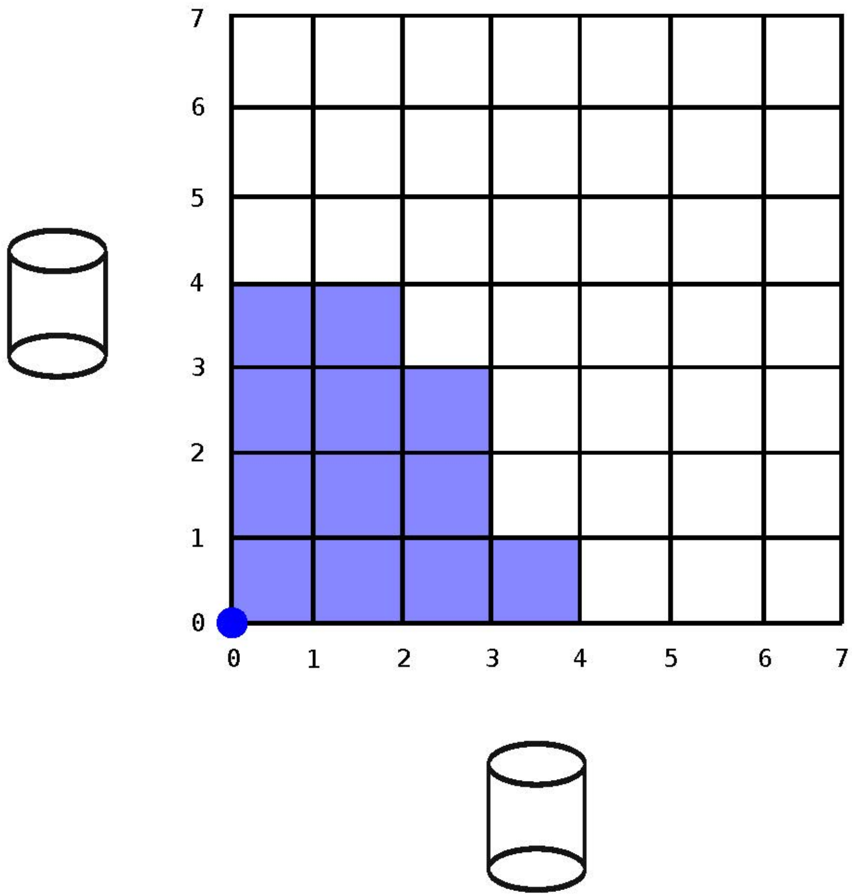
- The x and y axis denote the cell levels in [0,7].

2-write *q*-level WOM Code



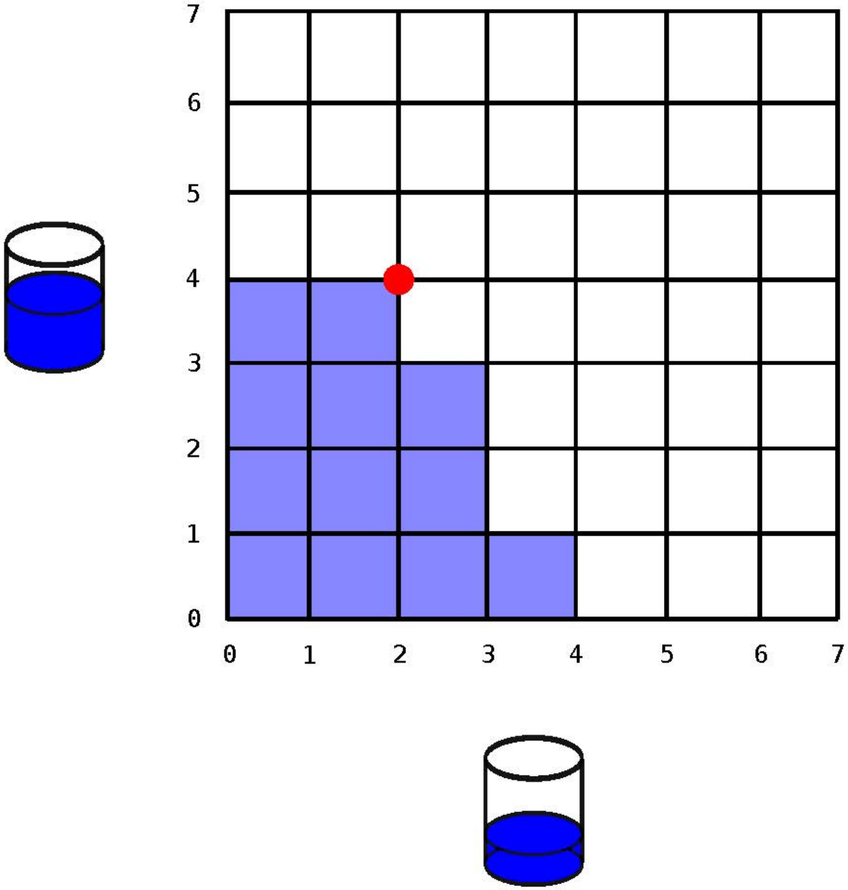
- The initial level on each cell is 0.

2-write q-level WOM Code



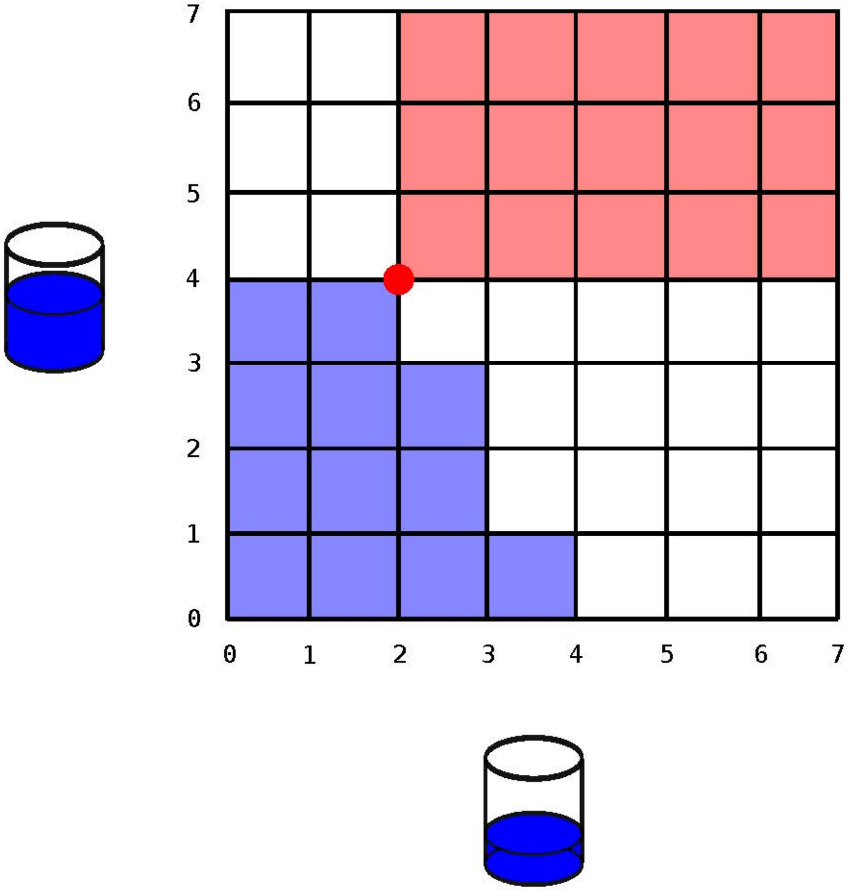
- Messages on the first write are encoded to points in the first write region, shown in blue.

2-write *q*-level WOM Code



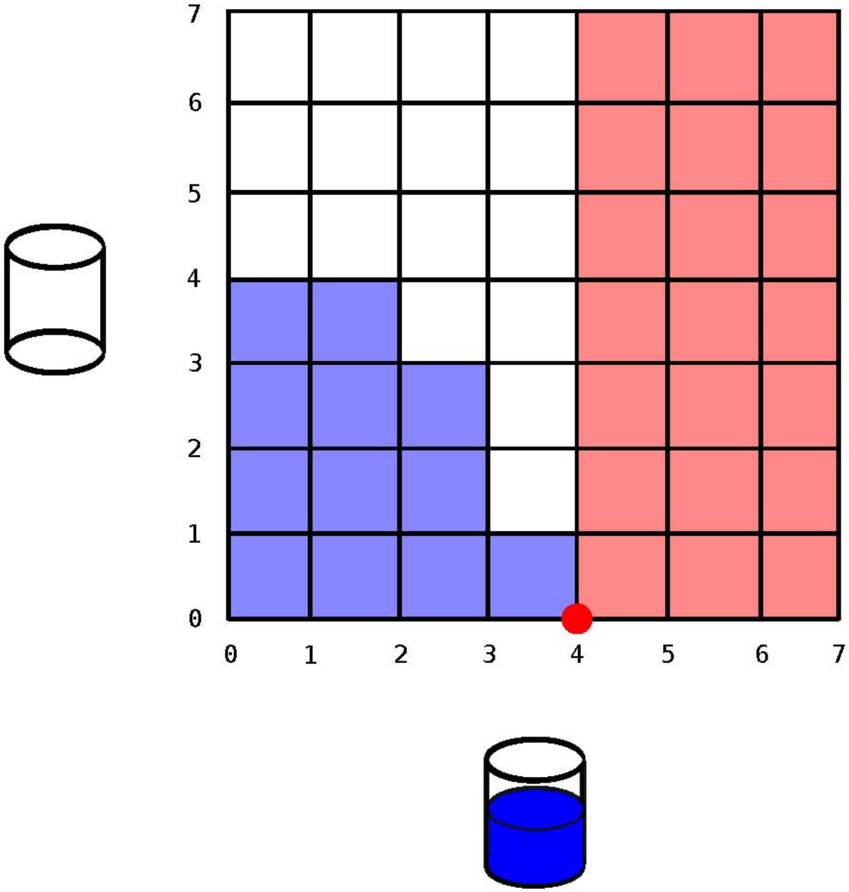
- The written cell levels (2,4) are indicated by the red dot.

2-write q-level WOM Code



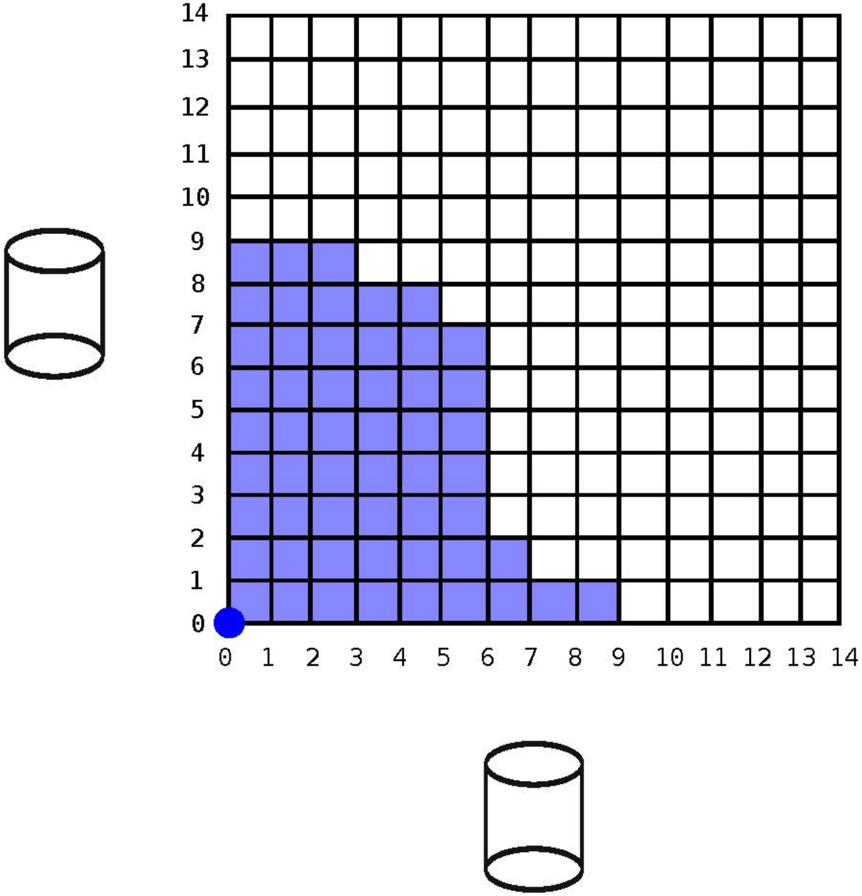
- Messages on 2nd write must encode to the red region.

2-write q-level WOM Code



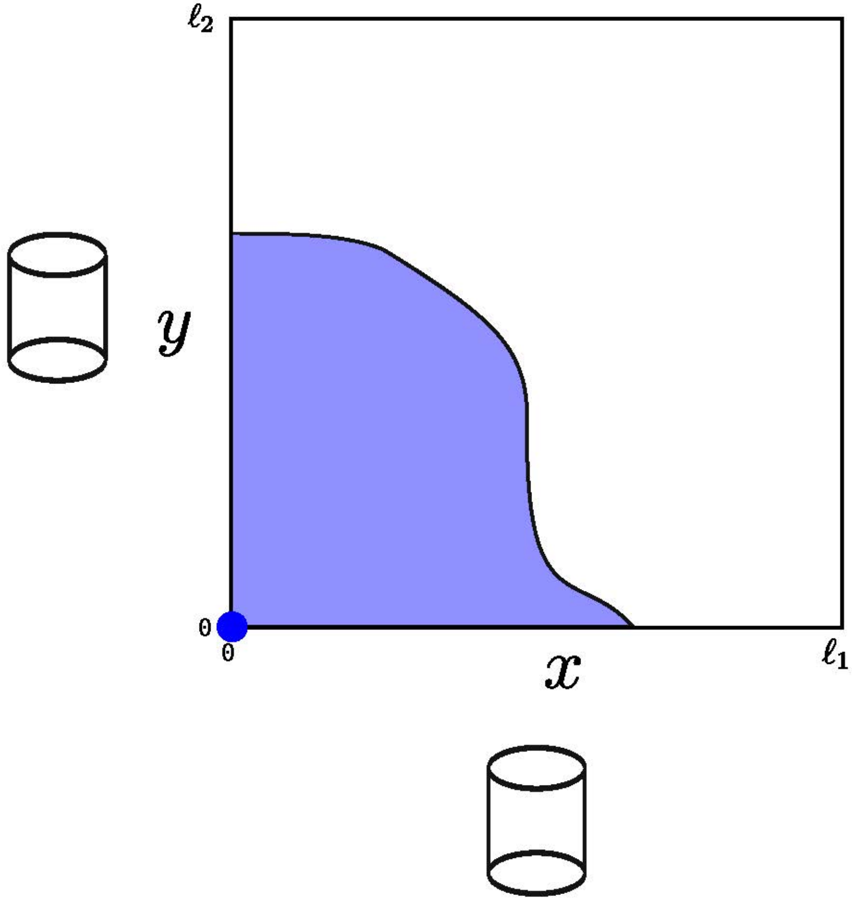
- The 2nd second write region depends on 1st write
- We want to optimize the worst-case sum-rate.

2-write q-level WOM Code



- When the number of levels q is large, the lattice becomes denser.

Continuous Approximation



Region
$[0,\ell_1]\text{x}[0,\ell_2]$

- For large q, we approximate the discrete levels by a continuous region whose area reflects the number of messages.

Continuous Approximation: 2-writes

- First write:
  - Number of messages

$$V_1 = |\mathbb{L}_1|$$

  - Message encoded to

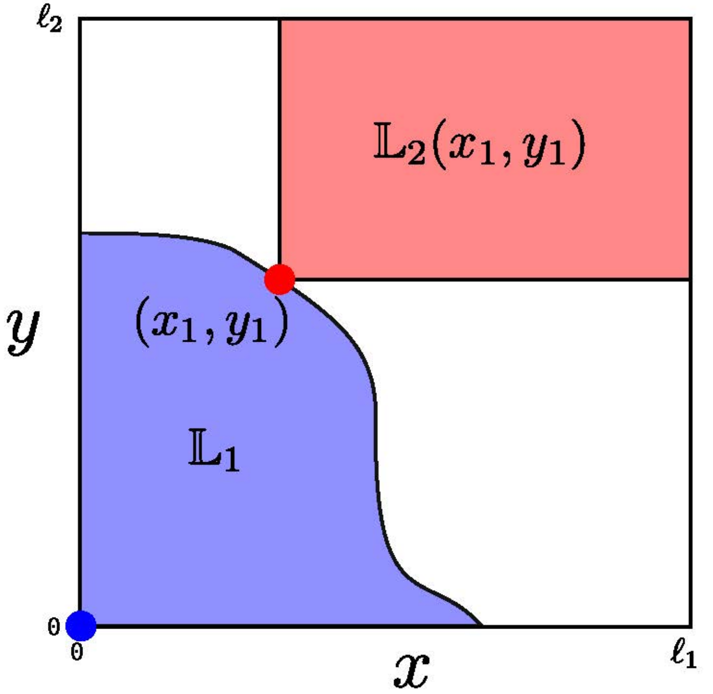
$$(x_1, y_1) \in \mathbb{L}_1$$

- Second write:
  - Message encoded to

$$(x_2, y_2) \in \mathbb{L}_2(x_1, y_1)$$

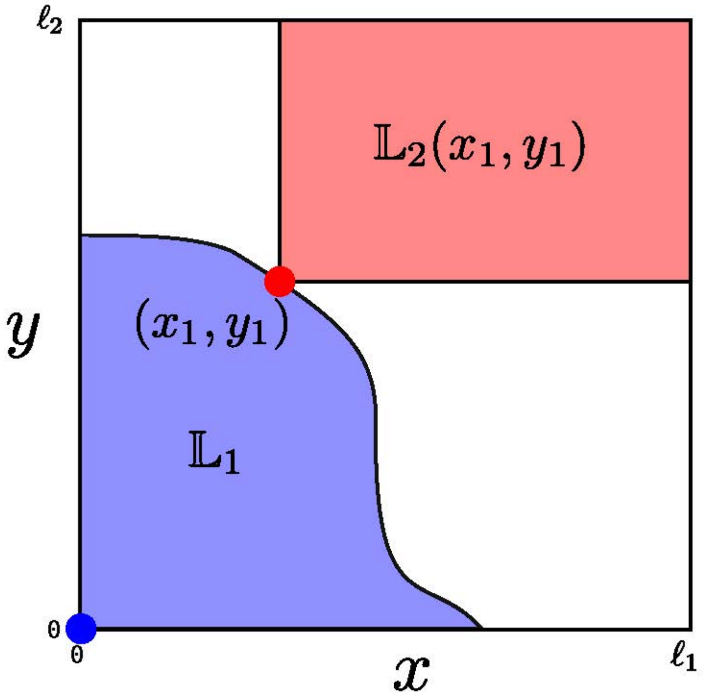
  - Number of 2nd-write messages that can be stored in worst case:

$$V_2 = \min_{(x_1, y_1) \in \mathbb{L}_1} |\mathbb{L}_2(x_1, y_1)|$$



Optimal Worst-case Sum-rate: 2-writes

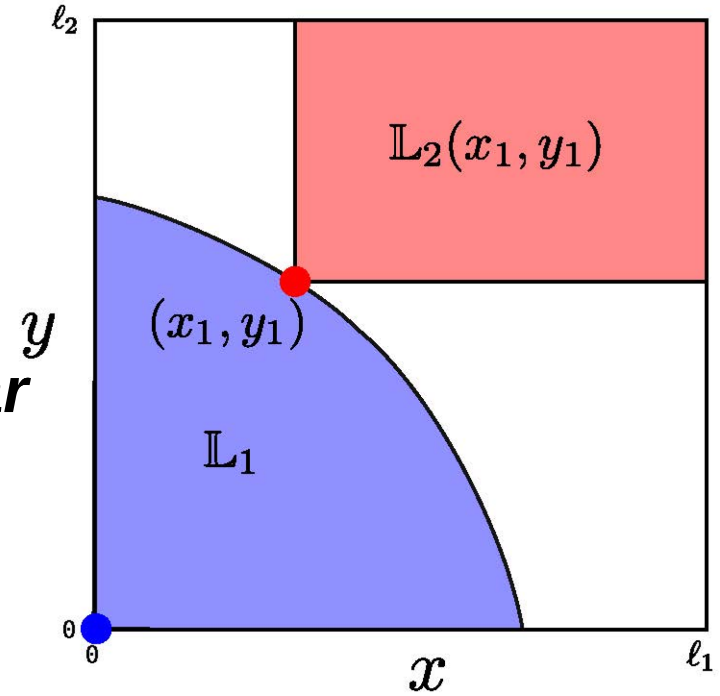
- We want to find the 1st -write region Λ_1 that maximizes the total number of messages on both writes when the first encoding is the point $(x_1, y_1) \in \mathbb{L}_1$ with the fewest choices in its 2nd-write region.
- So, we find Λ_1 that maximizes:



$$V_1 \cdot V_2 = \left|\mathbb{L}_1\right| \times \min_{(x_1, y_1) \in \mathbb{L}_1} \left|\mathbb{L}_2\left(x_1, y_1\right)\right|$$

2-write Worst-case Sum-rate Region

- The region Λ_1 that maximizes the worst-case total number of messages on both writes is a ***rectangular hyperbola*** defined by



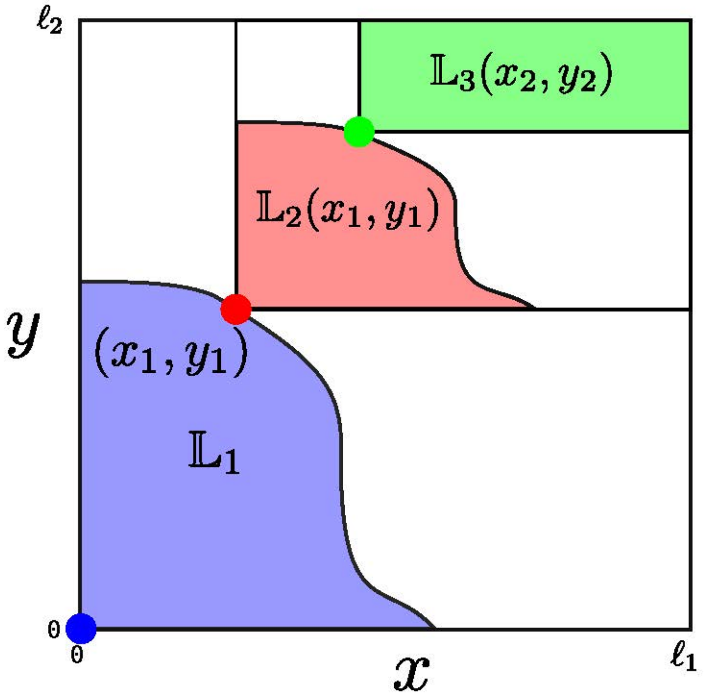
$$\mathrm{L}_1 = \left\{ (x, y) \;\middle|\; \left(1 - \frac{x}{\ell_1}\right)\left(1 - \frac{y}{\ell_2}\right) \geq \omega_2 \right\}$$

where $\omega_2 \approx 0.2847$. The resulting sum-rate is given by:

$$V_1 \cdot V_2 = \tfrac{1}{2}\omega_2 \left(1 - \omega_2\right)\left(\ell_1 \ell_2\right)^2$$

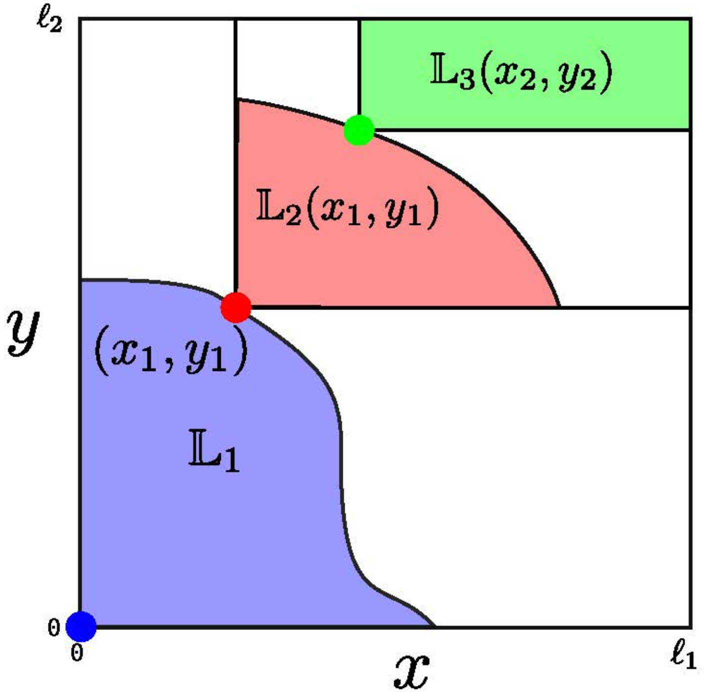
Continuous approximation: 3 writes

- For 3 writes on 2 cells, similar reasoning shows that the optimal boundary of the second-write region $\mathbb{L}_2(x_1, y_1)$ is a rectangular hyperbola that maximizes the number of messages for the 2nd and 3rd writes.



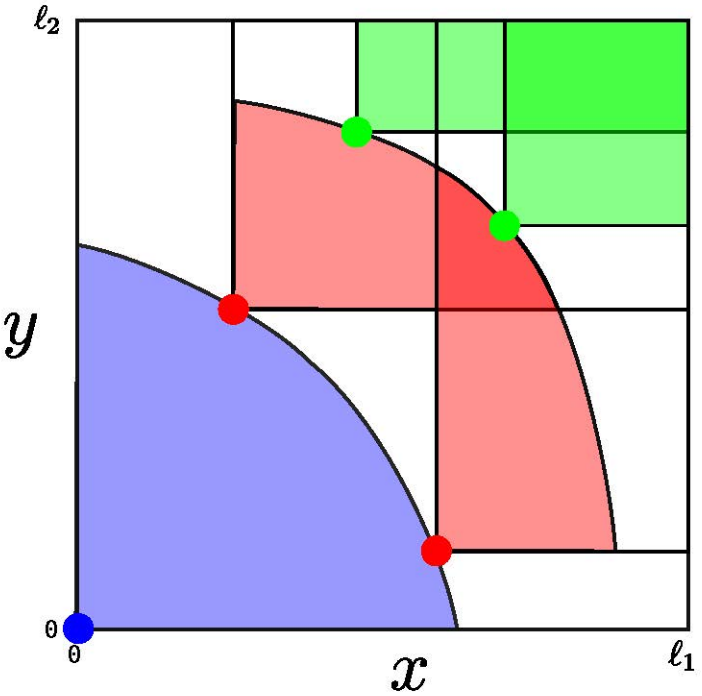
Continuous approximation: 3 writes

- For 3 writes on 2 cells, similar reasoning shows that the optimal boundary of the second-write region $\mathbb{L}_2(x_1, y_1)$ is a rectangular hyperbola that maximizes the number of messages for the 2nd and 3rd writes.



Continuous approximation: 3 writes

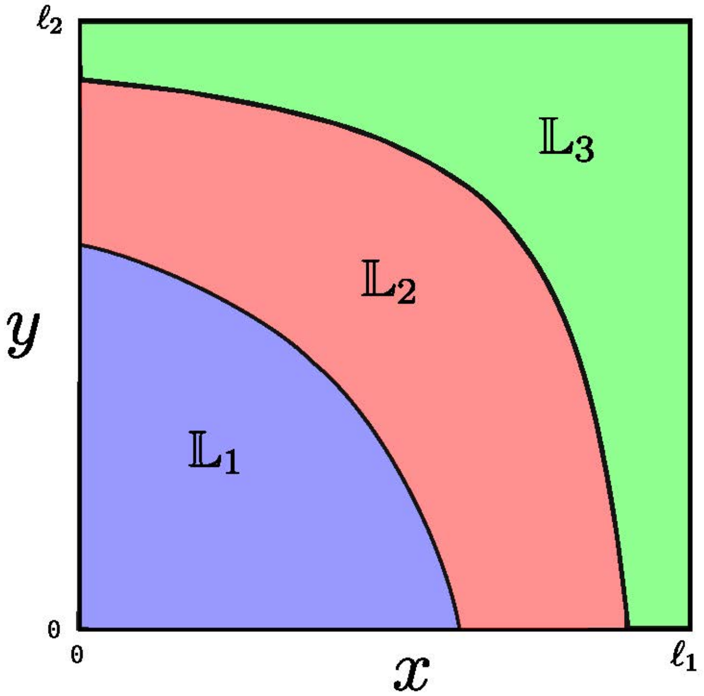
- For 3 writes on 2 cells, similar reasoning shows that the optimal boundary of the second-write region $\Lambda_2(x_1, y_1)$ is a rectangular hyperbola that maximizes the number of messages for the 2nd and 3rd writes.



- If Λ_1 is a rectangular hyperbola, the corresponding boundaries line up perfectly.

Continuous approximation: 3 writes

- For 3 writes on 2 cells, similar reasoning shows that the optimal boundary of the second-write region $\mathbb{L}_2(x_1, y_1)$ is a rectangular hyperbola that maximizes the number of messages for the 2nd and 3rd writes.



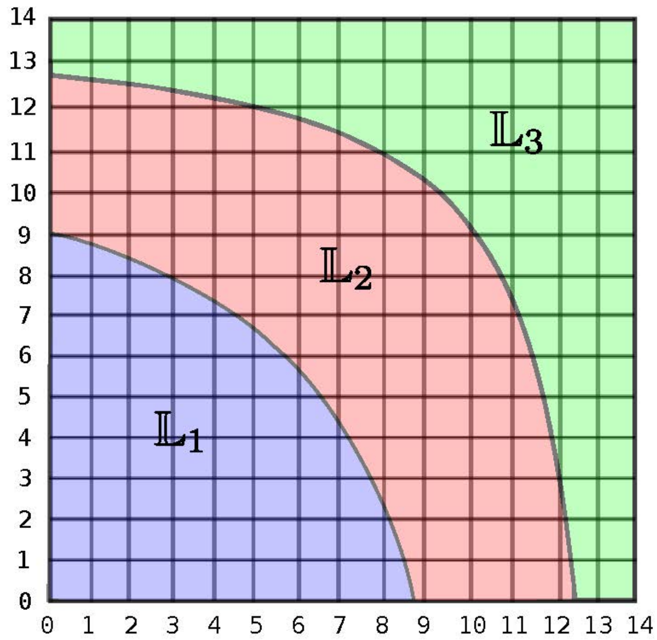
- If Λ_1 is a rectangular hyperbola, the corresponding boundaries line up perfectly.
- The optimal write-region boundaries are all hyperbolas.

Generalizations

- For 2 cells, $t > 3$ writes, unrestricted rates, the optimal worst-case sum-rate is achieved when the boundaries of the write regions are all rectangular hyperbolas.

- Further generalizations characterize the optimal write regions for n cells, t writes, for both fixed-rate and unrestricted-rate WOM codes.

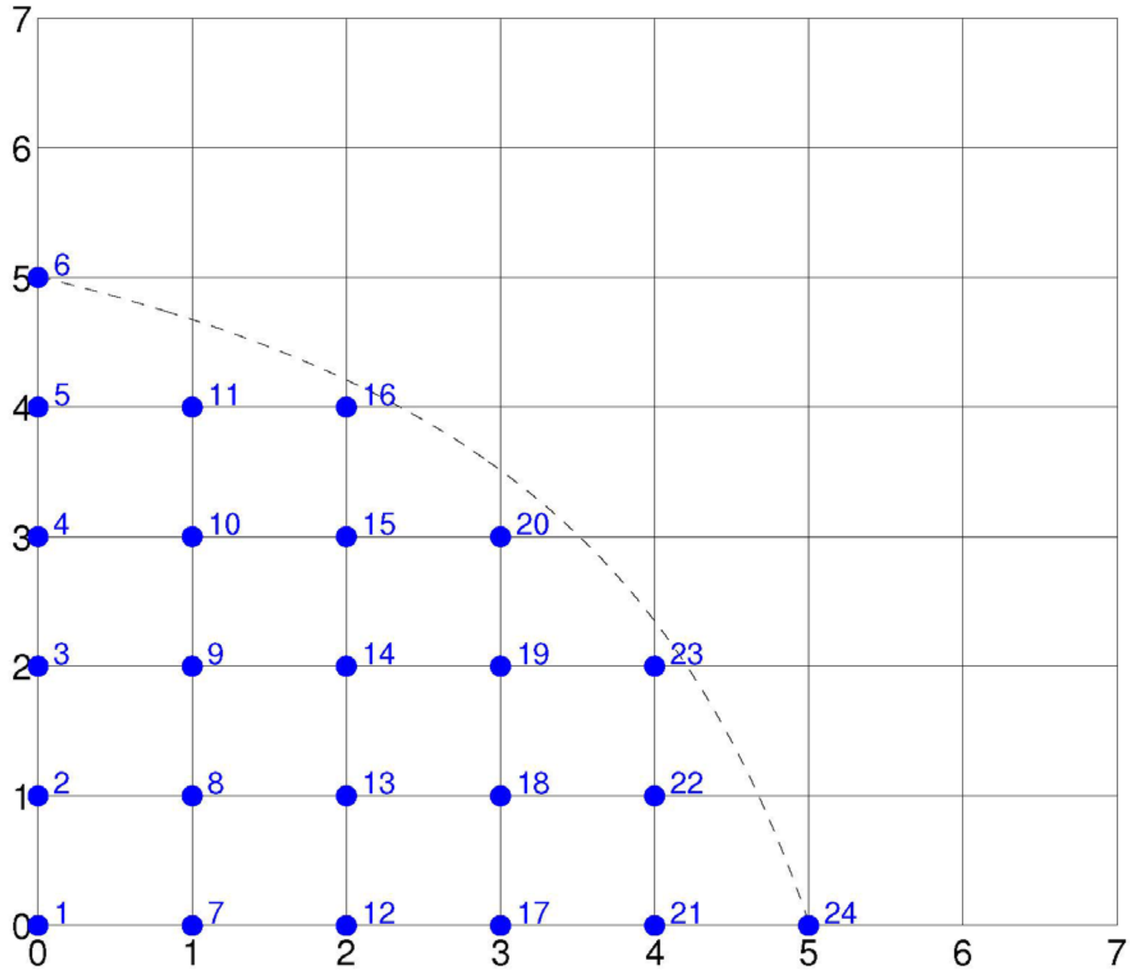
Codes for Discrete-level Cells



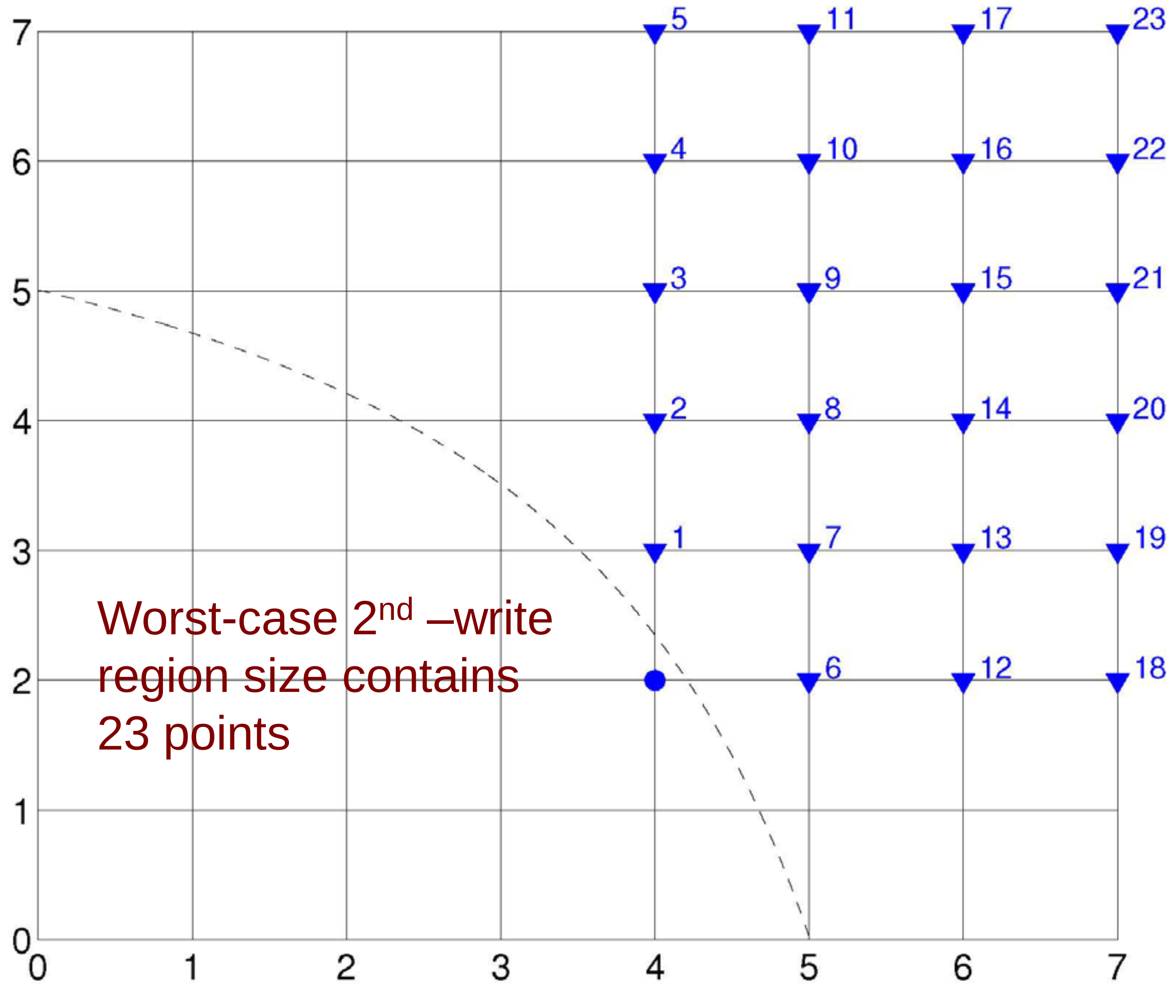
- To design codes for cells with q levels, we quantize the optimal write regions, creating corresponding codeword regions.
- Messages in i-th write are encoded into cell-level pairs in the i-th region.
- Consistent labeling of messages to codewords is needed

Example: 2-write 8-level WOM Code

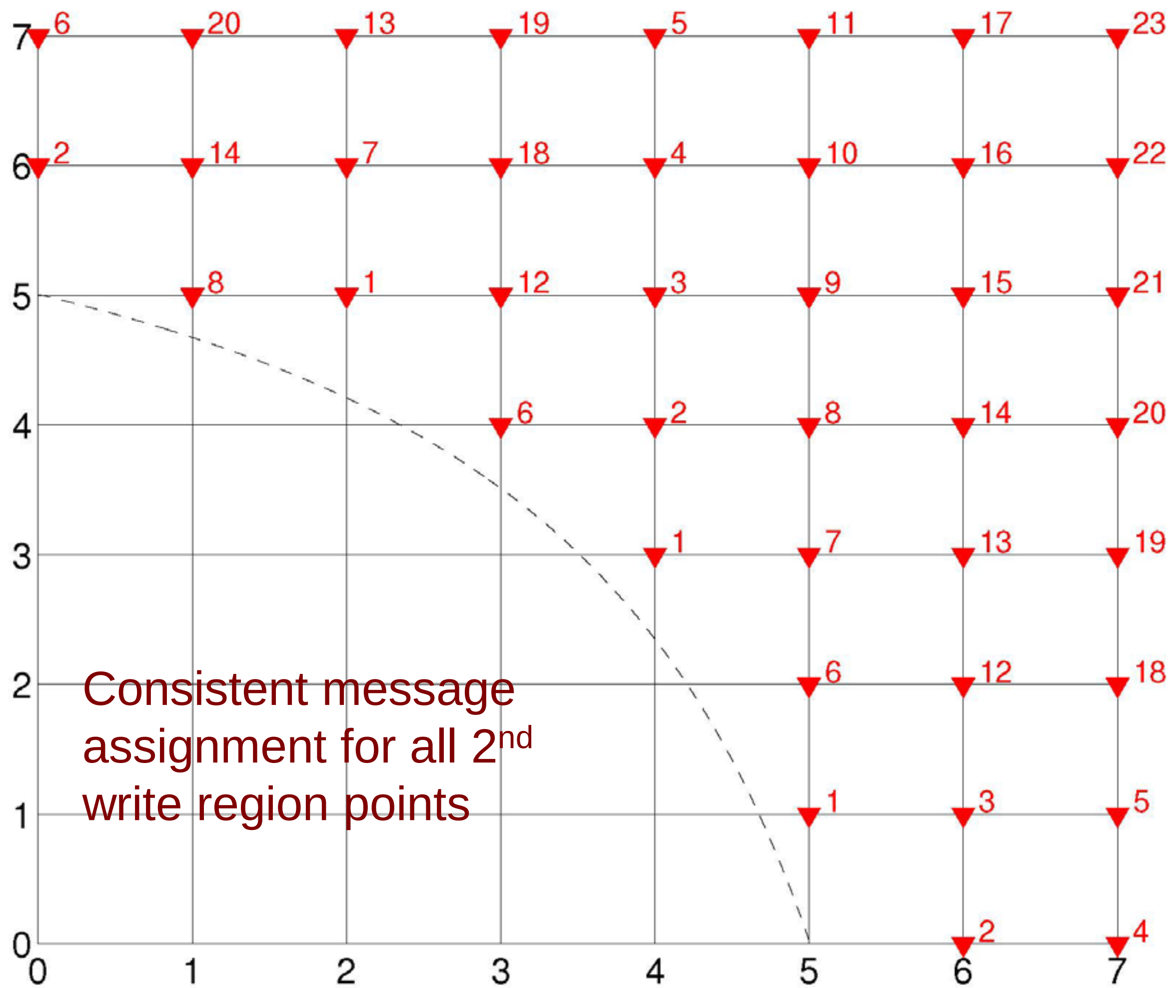
24 points in 1st write region



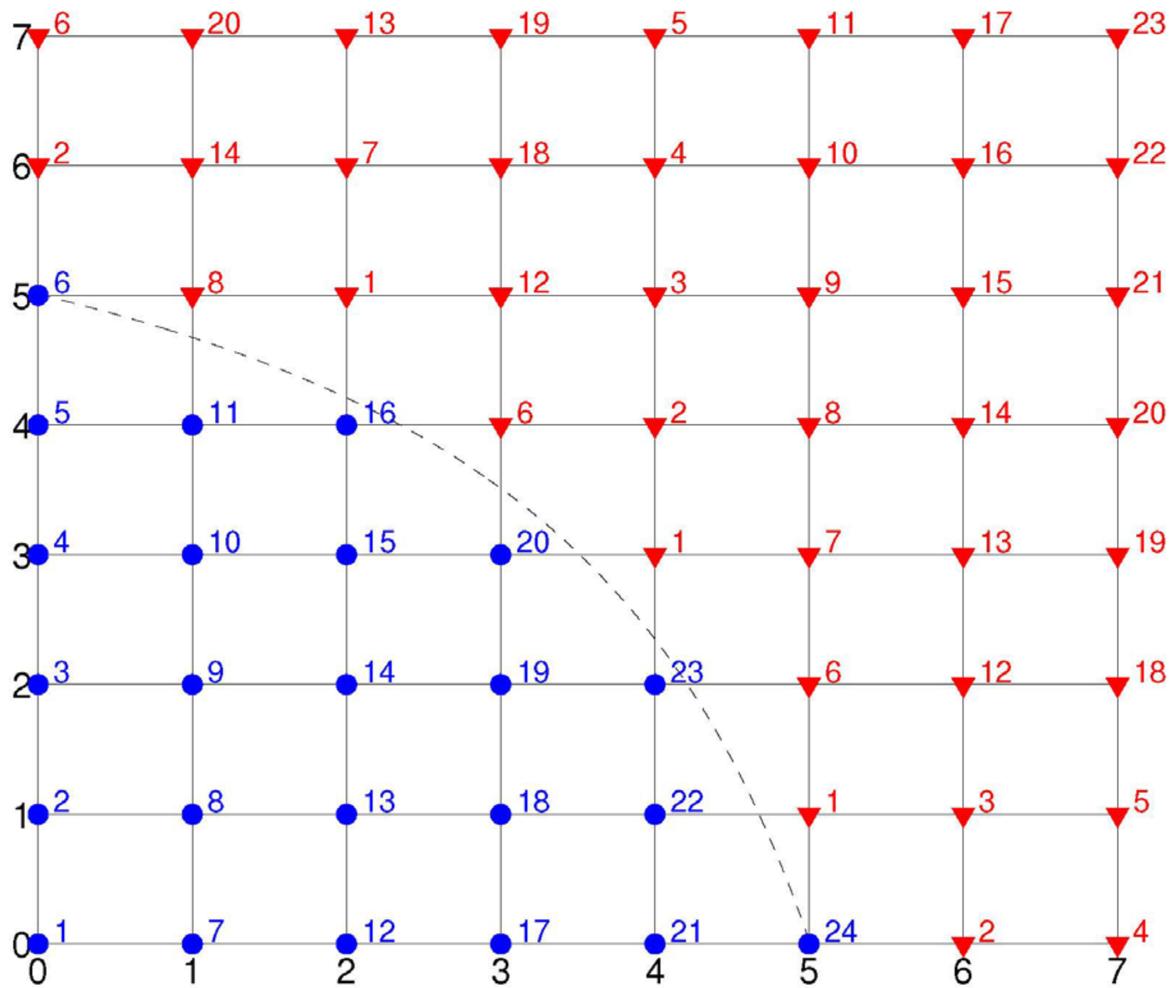
Example: 2-write 8-level WOM Code



Example: 2-write 8-level WOM Code



Example: 2-write 8-level WOM Code



Concluding Remarks

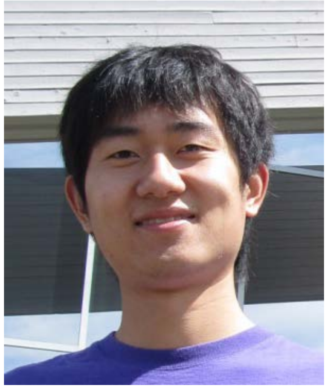
- WOM codes offer the possibility of increasing flash endurance by reducing the number of program-erase cycles.
- Recent studies show they may reduce write amplification [Luojie et al. 2012].
- WOM codes have been proposed as a way to combat inter-cell interference [Li 2011].
- The combination of error-correction and WOM coding is an active area of research.
- Progress has been made in the design and analysis of WOM codes, but **much remains to be done**!

Thanks to My Students

Aman Bhatia

Scott Kayser

Minghai Qin



Eitan Yaakobi

Thanks to My Colleagues

Prof. Jack Wolf | Prof. Brian Kurkoski | Prof. Alex Vardy

Thanks to
Pioneers in Coding for Flash

Prof. Andrew Jiang

Prof. Shuki Bruck

Thanks to My Sponsors

NSF

Western Digital®

TOSHIBA
Leading Innovation

LSI®

MARVELL®

Thank You for Your Attention

- Using the < 26,26 > / 7 Rivest-Shamir code

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

T		0	0	0	**1**	**1**	0	0
H		0	0	0	0	0	0	**1**
A		0	0	0	0	0	0	0
N		**1**	0	0	0	0	0	**1**
K		**1**	0	0	**1**	0	0	0
Y		0	0	0	0	**1**	0	**1**
O		0	**1**	**1**	0	0	0	0
U		0	0	**1**	0	**1**	0	0

v		0	0	**1**	**1**	**1**	0	0
e		0	0	**1**	**1**	0	**1**	**1**
r		0	0	0	**1**	0	**1**	**1**
y		**1**	0	**1**	0	**1**	**1**	**1**
m		**1**	0	**1**	**1**	0	0	0
u		0	0	**1**	**1**	**1**	**1**	**1**
c		0	**1**	**1**	**1**	0	0	**1**
h		**1**	**1**	**1**	**1**	**1**	0	0