LECTURE 8: MULTI-LEVEL RE-WRITE CODES
(n,k,t) WOM code:
Write k bits t times, on n cells

HOST → PHY WRITE
k bits

PHY WRITE

0 → 1 yes
1 → 0 no

Flash Media

n cells

WW ... W EW W W ... W E ...
t writes
Multi-Write (=Re-Write) Access

- Example $t = 2$
  - Writes:
    0, 1, 2, 4, 5, 6, 8, 9, 10, 12, 0, 4, 8, 12
1) Decoding Function

\[ \psi \left( c_1, \cdots, c_n \right) = \left( y_1, \cdots, y_k \right) \]
\[ c_i \in \{0,1\} \]

2) Update Function

\[ \mu \left( c_1, \cdots, c_n ; y_1, \cdots, y_k \right) = \left( c'_1, \cdots, c'_n \right) \]

current cell values \quad new data \quad new cell values

1 \rightarrow 0
Binary → q-ary

- **Storage rate:** q-ary WOM codes need lower redundancy for the same number of writes

- **Complexity:**
  - Exponential in \( n \)
  - Polynomial in \( q \)
Multi-Level Re-Write Codes

1) Decoding Function

\[ \psi \left( c_1, \ldots, c_n \right) = \left( y_1, \ldots, y_k \right) \]

\[ c_i \in \{0, \ldots, q - 1\} \]

2) Update Function

\[ \mu \left( c_1, \ldots, c_n ; y_1, \ldots, y_k \right) = \left( c_1', \ldots, c_n' \right) \]

current cell values  new data  new cell values
Single-Cell $q$-ary Code

$n=1$, $k=1$, $t=q-1$
Single-Cell $q$-ary Code

$n=1, \ k=2, \ t=(q-1)/3$

- $y=00$
- $y=01$
- $y=10$
- $y=11$
Single-Cell, General k

$n=1$, $k$, $t=(q-1)/(2^k-1)$

t is inverse exponential in $k$
2 Cells, k=3

Option 1: concatenate

2 x q-ary cells → 1 x 2q-1 ary cell

\[ t = \left\lfloor \frac{2q - 2}{2^3 - 1} \right\rfloor = \left\lfloor \frac{2}{7} (q - 1) \right\rfloor \]
2 Cells, $k=3$

$n=2$, $k=3$

Option 2: 2D code

$\{0, 1, \ldots, 7\} \leftrightarrow (y_1, y_2, y_3)$

Option 1

Option 2

$t = ?$
2 Cells, k=3: 2D Code

\[ t = \left\lfloor \frac{1}{2} (q - 1) \right\rfloor \]
Can We Do Better?

t = \left\lfloor \frac{1}{2} (q - 1) \right\rfloor
Step 1: Tiling

Lattice Tiling:

\[ v_1 = (2, 2) \]
\[ v_2 = (3, -1) \]

Translate to:

\[ a_1 v_1 + a_2 v_2 \]

\[ a_1, a_2 \text{ integers} \]
### t=4 Writes with q=8 Levels

<table>
<thead>
<tr>
<th>8</th>
<th>0 2 5 1 4 7 3 6 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>7 3 6 0 2 5 1 4 7</td>
</tr>
<tr>
<td>6</td>
<td>5 1 4 7 3 6 0 2 5</td>
</tr>
<tr>
<td>5</td>
<td>6 0 2 5 1 4 7 3 6</td>
</tr>
<tr>
<td>4</td>
<td>4 7 3 6 0 2 5 1 4</td>
</tr>
<tr>
<td>3</td>
<td>2 5 1 4 7 3 6 0 2</td>
</tr>
<tr>
<td>2</td>
<td>3 6 0 2 5 1 4 7 3</td>
</tr>
<tr>
<td>1</td>
<td>1 4 7 3 6 0 2 5 1</td>
</tr>
<tr>
<td>0</td>
<td>0 2 5 1 4 7 3 6 0</td>
</tr>
</tbody>
</table>

**C2**

**C1**
t=4 Writes with q=8 Levels

Problem:
2-increment x4

Example, write

C

7 5 4 6 0 2 5 1 4
5 1 4 7 3 6 0 2
6 0 2 5 1 4 7 3
4 7 3 6 0 2 5 1
3 2 5 1 4 7 3 6 0
2 3 6 0 2 5 1 4 7
1 4 7 3 6 0 2 5
0 2 5 1 4 7 3 6

4 7 5 6
$t=4$ Writes with $q=8$ Levels

Solution:
Nearest $\rightarrow$ Balanced

\[\begin{array}{cccccccc}
7 & 3 & 6 & 0 & 2 & 5 & 1 & 4 \\
5 & 1 & 4 & 7 & 3 & 6 & 0 & 2 \\
6 & 0 & 2 & 5 & 1 & 4 & 7 & 3 \\
4 & 7 & 3 & 6 & 0 & 2 & 5 & 1 \\
2 & 5 & 1 & 4 & 7 & 3 & 6 & 0 \\
3 & 6 & 0 & 2 & 5 & 1 & 4 & 7 \\
1 & 4 & 7 & 3 & 6 & 0 & 2 & 5 \\
0 & 2 & 5 & 1 & 4 & 7 & 3 & 6 \\
\end{array}\]
Lattice Equivalence → Balance

\[ \sum (2,1) (2,1) (2,1) (2,1) = (8,3) \] Nearest

\[ \sum (2,1) (2,1) (2,1) (1,4) = (7,7) \] Balanced

Lattice equivalent \[ (1,4) = (2,1) + (-1,3) = (2,1) + v_1 - v_2 \]
2D with Tiling

- $k=3$: \[ t = \left\lfloor \frac{4}{7} (q - 1) \right\rfloor \]

- General odd $k$, $t=4$:

  \[
  q = 6 \cdot 2^{\frac{k-1}{2}} - 3 \quad \rightarrow \quad q = 5.5 \cdot 2^{\frac{k-1}{2}} - 3
  \]

  No tiling \hspace{1cm} \text{Tiling}
**q-ary WOM Capacity**

**Theorem:** for each q-ary cell, the total information rate in t writes satisfies

\[ R_{sum} \leq C(q, t) = \log_2 \binom{q + t - 1}{t} \] [bits]

**Proof idea:**
How many ways to divide q-1 (increments) among t+1 sets with sizes \( \geq 0 \)?

Upper bound on the number of writes:

\[ tk \leq n \log_2 \binom{q + t - 1}{t} \]
Optimal fixed-rate codes?

• **Problem:** capacity upper bound likely not tight
  - Allows *variable-rate* codes
  - Gap is proven for binary codes
**Fixed-Rate Upper Bound**

**Theorem (n=2):**
Let $s$ be an integer. If $2^k > s(s + 1)/2$, then the number of writes satisfies

$$t \leq \left\lfloor \frac{2(q - 1)}{s} \right\rfloor$$

**Proof idea:**
With $k$ input bits, there exists a worst-case input that increments the sum of levels by at least $s$.

**For k=3:**
Any code satisfies

$$t \leq \left\lfloor \frac{2(q - 1)}{3} \right\rfloor$$
Can We Do $n=2$, $k=3$, $t=3$ with $q=6$?
Can we write 3 times with $q=6$?

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Step 1: easy

Cell 2

Cell 1
Step 2: 2\textsuperscript{nd} write Sudoku

All numbers 0,1,2,3,4,5,7 in 8 positions

Cell 2

Cell 1
Step 2: 2\textsuperscript{nd} write Sudoku

All numbers 0,1,2,3,4,5,6 in 8 positions

Cell 2

Cell 1
2nd write Solution
Step 3: 3rd write Sudoku

Cell 2

Cell 1
3rd write Sudoku

Cell 2

Cell 1
3rd write Sudoku

Cell 2

Cell 1
3rd write Sudoku

Cell 2

Cell 1
3rd write Solution

Got $t=3$!
Optimal code family, k=3

Theorem (k=3): An explicit construction exists with

\[ t = \left\lfloor \frac{2(q-1)}{3} \right\rfloor - 1. \]

Matching upper bound (k=3): Any code satisfies

\[ t \leq \left\lfloor \frac{2(q-1)}{3} \right\rfloor - 1. \]
Practical Re-Write Codes II

All bits equal

Hot/Cold bits

File System log → ReWrite $k=2$ → In-laws’ photos

File System log → ReWrite $1+1$ → hot/cold

In-laws’ photos
Hot+Cold Rewrite

- $k$ bits are written
- $t$ writes for $k$ bits
- 1 write for each cold bit
  (anywhere in the write sequence)