Constructing generalized position modulation codes for increased rewrites

September 7, 2011

Algebraic Aspects of Coding Theory Workshop, EPFL

Katie Haymaker

Department of Mathematics
University of Nebraska-Lincoln
(Joint work with Christine Kelley)
Outline

- Background
- Position modulation codes
- New construction using component WOMs
- Comparison with original position modulation code
- Conclusions
Flash memory structure

The memory is organized into blocks of cells in which each cell can be charged up to one of $q$ levels.

- A block has $\sim 10^5$ cells and can be typically erased $\sim 100,000$ times before significant deterioration.

- While increasing the charge of a cell is easy, decreasing the charge is costly. When a cell has to be erased, the whole block containing that cell is erased.

- Information is encoded and stored in a cell state vector, of length up to $2^{17}$ cells.
Coding for flash memories

Some goals in code design are to

- maximize the number of rewrites
- distribute erasing evenly to increase lifetime
- incorporate error correcting schemes for charge leakage and overshooting, as well as read and write errors
General approaches for codes

- WOM and generalized WOM
  (Kuznetsov, Tsybacov, 1974), (Rivest & Shamir, 1982), (Fiat, Shamir, 1984), (Merkx, 1984),
  (Alshwede, Zhang, 1994)

- Floating codes
  (Jiang, Bohossian, Bruck, 2007), (Chierichetti, Finucane, Liu, Mitzenmacher, 2010)

- Rank modulation codes
  (Jiang, Schwartz, Bruck, 2008)

- Position modulation codes
  (Wu, Jiang, 2009)

- Rewriting codes with error correction
  (Zémor, Cohen, 1991), (Jiang, 2010)

And lots more...
Example: Rivest-Shamir WOM code

A WOM code that maps 2 information bits to 3 coded bits and tolerates two writes.

<table>
<thead>
<tr>
<th>Information</th>
<th>1\textsuperscript{st} write</th>
<th>2\textsuperscript{nd} write</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>000</td>
<td>111</td>
</tr>
<tr>
<td>01</td>
<td>100</td>
<td>011</td>
</tr>
<tr>
<td>10</td>
<td>010</td>
<td>101</td>
</tr>
<tr>
<td>11</td>
<td>001</td>
<td>110</td>
</tr>
</tbody>
</table>
Example: Rivest-Shamir WOM code

A WOM code that maps 2 information bits to 3 coded bits and tolerates two writes.

<table>
<thead>
<tr>
<th>Information</th>
<th>1\textsuperscript{st} write</th>
<th>2\textsuperscript{nd} write</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>000</td>
<td>111</td>
</tr>
<tr>
<td>01</td>
<td>100</td>
<td>011</td>
</tr>
<tr>
<td>10</td>
<td>010</td>
<td>101</td>
</tr>
<tr>
<td>11</td>
<td>001</td>
<td>110</td>
</tr>
</tbody>
</table>

For example, the sequence $\begin{bmatrix} 11 \end{bmatrix} \rightarrow \begin{bmatrix} 10 \end{bmatrix}$ would be written as
Example: Rivest-Shamir WOM code

A WOM code that maps 2 information bits to 3 coded bits and tolerates two writes.

<table>
<thead>
<tr>
<th>Information</th>
<th>1\textsuperscript{st} write</th>
<th>2\textsuperscript{nd} write</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>000</td>
<td>111</td>
</tr>
<tr>
<td>01</td>
<td>100</td>
<td>011</td>
</tr>
<tr>
<td>10</td>
<td>010</td>
<td>101</td>
</tr>
<tr>
<td>11</td>
<td>001</td>
<td>110</td>
</tr>
</tbody>
</table>

For example, the sequence $\boxed{11} \rightarrow \boxed{10}$ would be written as

$\boxed{001} \rightarrow \boxed{101}$
Parameters of rewriting codes

Let $\langle v_1, \ldots, v_t \rangle/n$ denote a wom code $C$ on $n$ cells, representing $v_i$ messages on the $i^{th}$ write.

- $\langle v \rangle^t/n$ denotes a code where $v_1 = v_2 = \cdots = v_t$.

- The rate of $C$ is
  \[
  \frac{\log_2 (v_1 \cdots v_t)}{n}.
  \]

- A wom is shown to have "capacity" up to $n \log(n)$ bits (Rivest and Shamir, 1982).

- On $q$-levels, the deficiency of a flash code is $\delta(C) = n(q - 1) - t$. 
Using position modulation with a component WOM code

The position modulation code (Wu, Jiang 2011) uses the number and positions of non-zero groups of cells and the contents of these groups to store information.

**Main result:** Given $v_1, \ldots, v_t$, with $v_i \geq 2$, there exists a position modulation code that achieves a rate that is at least half the optimal rate.
Using position modulation with a component WOM code

- The position modulation code (Wu, Jiang 2011) uses the number and positions of non-zero groups of cells and the contents of these groups to store information.

  Main result: Given $v_1, \ldots, v_t$, with $v_i \geq 2$, there exists a position modulation code that achieves a rate that is at least half the optimal rate.

- Variation on position modulation: Delay the soft erase operation employed by position modulation by using a wom code on groups of cells.
Example

- $h$ groups, each with $m$ cells
- use a $\langle v \rangle^t / m$ wom code for each group
- construct a wom that allows $T$ writes
Example

Start by choosing $k_1$ groups.
Write a first generation word from $C$ on each of the $k_1$ groups.
Example

Now choose $k_2$ groups from the remaining zero groups, and write a first generation word on them; write a second generation word on the initial $k_1$ groups.
Example

Continue in a similar way with the third generation write.
Write generation: 4
Suppose $C$ allows $t = 3$ writes.
Before the fourth write, soft-erase the first $k_1$ groups of cells by setting them all to one.
Example

Write generation: 5
Example

Write generation: 6
Example

Write generation: 7
Example

All groups have been written on as much as $C$ allows; soft-erase by setting all cells to one.
Details of pm + wom codes

Starting with a $\langle v \rangle^T / m$ wom code, we construct a length $hm$ wom code, where $T$ is the proposed number of writes for the construction. Set $s = T - t$. 
Details of pm + wom codes

Starting with a $\langle v \rangle^t / m$ wom code, we construct a length $hm$ wom code, where $T$ is the proposed number of writes for the construction. Set $s = T - t$.

- Let $k_1, \ldots, k_s$ denote the number of groups chosen on successive writes.
- In order to guarantee $T$ writes, choose $k_i \geq 1$ for each $i$.
- Set $N_i = k_1 + \cdots + k_i$. 
Details of pm + wom codes

Starting with a $\langle v \rangle^t/m$ wom code, we construct a length $hm$ wom code, where $T$ is the proposed number of writes for the construction. Set $s = T - t$.

- Let $k_1, \ldots, k_s$ denote the number of groups chosen on successive writes.
- In order to guarantee $T$ writes, choose $k_i \geq 1$ for each $i$.
- Set $N_i = k_1 + \cdots + k_i$.

On the first write, we can represent a variable of cardinality up to

$$u_1 = \sum_{k_1=1}^{h-s+1} \binom{h}{k_1} (v - 1)^{k_1}.$$
Details of pm + wom codes

Starting with a $\langle v \rangle^f/m$ wom code, we construct a length $hm$ wom code, where $T$ is the proposed number of writes for the construction. Set $s = T - t$.

- Let $k_1, \ldots, k_s$ denote the number of groups chosen on successive writes.
- In order to guarantee $T$ writes, choose $k_i \geq 1$ for each $i$.
- Set $N_i = k_1 + \cdots + k_i$.

On the first write, we can represent a variable of cardinality up to

$$u_1 = \sum_{k_1=1}^{h-s+1} \binom{h}{k_1} (v - 1)^{k_1}.$$

On the $i^{th}$ write:

$$u_i = (v - 1)^{N_{i-1} - N_{i-t}} \sum_{k_i=1}^{h-s+1-N_{i-1}} \binom{h - N_{i-1}}{k_i} (v - 1)^{k_i}.$$
Conditions on $C$

- The write-generation should be identifiable from the code word.
- Adjustments are necessary if the all-ones word is a code word.
- To create a code of large block size, we can use nearly optimal component codes of short block length.
Let $C$ be the $\langle v^t \rangle / m$ wom code we are using on the groups of cells.

- Messages are represented by the number and placement of nonzero groups, as well as by the contents of those groups.

- The all-zeros word is read as 0, and the all-ones word is read as an ‘erase’ symbol.
Decoding

Decoding has two cases:

- If there are zero groups left, then the most recently written groups will have first-generation words from $C$. The information is decoded from the number and positions of the most recently written groups, as well as the words written in all the non-zero groups.

- If there are no zero groups left, then the information is decoded from the words contained in the active groups.
Original position modulation code

There is no component wom code, so each group of \( m \) cells can store one of \( 2^m - 1 \) symbols.

The soft-erase operation is performed after a group of cells has been written on and before the next group of cells is chosen to store information.
Original position modulation code

- Encoding and decoding can be achieved with polynomial complexity.
- Wu and Jiang observe that $m = 2$ or $m = 3$ generally give better performance ($m = 2$ is used to obtain the result ensuring half the optimal rate).
- Position modulation codes out-perform wom codes of comparable complexity for $t = 5, 6, 8, 9, 10$. 
Comparison to pm + wom code

**Complexity:** encoding and decoding for position modulation can be done with polynomial complexity, so the complexity of pm + wom depends on the wom code $C$.

**Performance comparison:**

- By delaying the soft-erase operation, fewer cells get arbitrarily increased to one. This is at the cost of the amount of information that each group of cells can represent.
- Original position modulation code starts with a given amount of information and determines the block length that yields a good rate.
- The pm+wom scheme allows for flexibility of block length and group size, without significant loss of rate.
## Comparison of rates

<table>
<thead>
<tr>
<th></th>
<th>Known wom code</th>
<th>PM code</th>
<th>pm+wom code</th>
</tr>
</thead>
<tbody>
<tr>
<td>t=2</td>
<td>1.34</td>
<td>1.14</td>
<td>–</td>
</tr>
<tr>
<td>t=3</td>
<td>1.49</td>
<td>1.35</td>
<td>1.46</td>
</tr>
<tr>
<td>t=4</td>
<td>1.60</td>
<td>1.49</td>
<td>1.54</td>
</tr>
<tr>
<td>t=5</td>
<td>1.57</td>
<td>1.63</td>
<td>1.66</td>
</tr>
<tr>
<td>t=6</td>
<td>1.60</td>
<td>1.71</td>
<td>1.73</td>
</tr>
<tr>
<td>t=7</td>
<td>1.82</td>
<td>1.81</td>
<td>1.79</td>
</tr>
<tr>
<td>t=8</td>
<td>1.65</td>
<td>1.88</td>
<td>1.83</td>
</tr>
<tr>
<td>t=9</td>
<td>1.67</td>
<td>1.95</td>
<td>1.87</td>
</tr>
<tr>
<td>t=10</td>
<td>1.63</td>
<td>2.01</td>
<td>1.90</td>
</tr>
</tbody>
</table>

- The first two columns use low-complexity, fixed-information wom codes from Wu and Jiang (2011).
- The pm+wom rates were calculated assuming the maximum value of $v_i$ is used for all $i$. 
Conclusions

We showed

- a generalization of position modulation codes that uses a component wom code;

- pm + wom codes that achieve rates that are competitive with standard position modulation codes for small values of $t$, while allowing for a wider range of group sizes and block lengths.
Conclusions

We showed

- a generalization of position modulation codes that uses a component wom code;
- pm + wom codes that achieve rates that are competitive with standard position modulation codes for small values of $t$, while allowing for a wider range of group sizes and block lengths.

Future work:

- show that for particular choices of $C$, the pm+wom code can achieve rates at least $\tilde{\sim}$ the optimal rate;
- generalization for multilevel;
- analyze performance when using an error-correcting component wom code.
References


