Flash Memory SUMMIT

UCSD

An Information Theory Approach for Flash Memory

Eitan Yaakobi, Paul H. Siegel, Jack K. Wolf
University of California, San Diego
Center for Magnetic Recording Research

Santa Clara, CA USA
August 2008

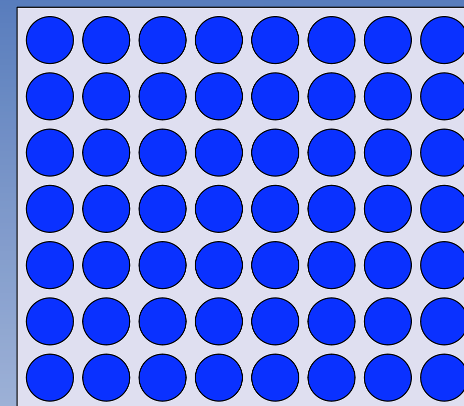
CMRR
Center for Magnetic Recording Research

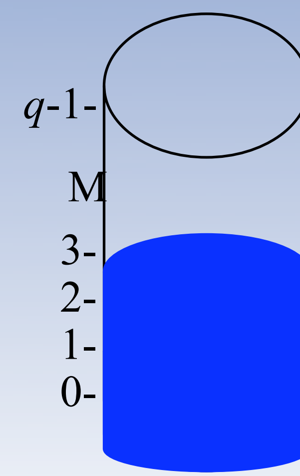
Outline

- Multi-level flash memory model
- Coding for flash memory – recent work
- Single-bit storage
  - Unbiased and biased inputs
- Extension to non-binary symbols
- Results on multiple-bit storage
- Summary

Multi-Level Flash Memory Model

- The memory consists of a block of cells
  - Each cell has q different levels.
  - The different levels are represented by different amounts of electrons in the cell.
  - The cell's level is increased by pulsing electrons.
  - In order to reduce level, the cell and its containing block ($\sim 10^5$ cells) must be reset to level 0 before rewriting – a very EXPENSIVE operation.
- This generalizes the Write Once Memory (WOM) model.





Coding for Multi-Level Flash Memory

- The general problem: How to represent the data efficiently such that resetting operations are postponed as much as possible?

Coding for Flash Memory: Recent Work

- Floating codes (Jiang, Bohossian and Bruck, 2007, and Jiang and Bruck, 2008)
  - k l-ary symbols are stored using n memory cells.
  - Individual symbols changed in separate writes.
  - Goal is to maximize the number of writes before resetting is required.
- Buffer coding for flash memory (Bohossian, Jiang and Bruck, 2007)
  - A buffer of size r is stored using n memory cells

Single Bit Storage

- Storing a single bit of information
  - One cell of q-levels.
  - The encoder: If the bit differs from the previously stored bit, then the cell level is increased by one.
  - The decoder: The value of the stored bit is equal to the cell level modulo 2.

Single Bit Storage

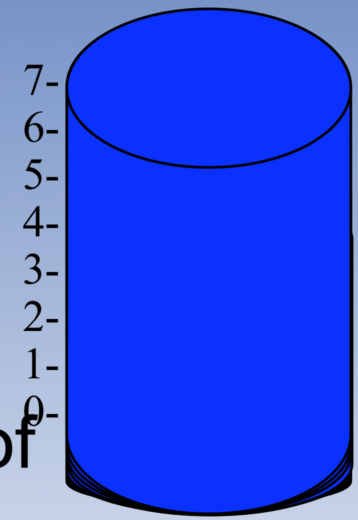
- Example:

  Input Sequence Cell Level

  1, 0, 1, 0, 0, 0, 1, 1, 0, 1

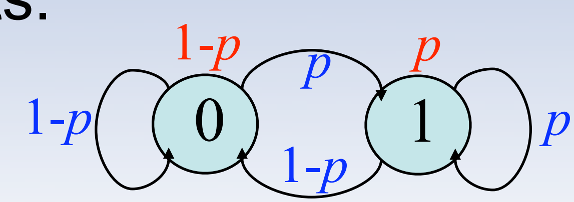
  1, 2, 3, 4, 4, 4, 5, 5, 6, 7

- The same approach extends to a memory with multiple cells. The sum of all cell levels modulo 2 indicates the stored bit value.

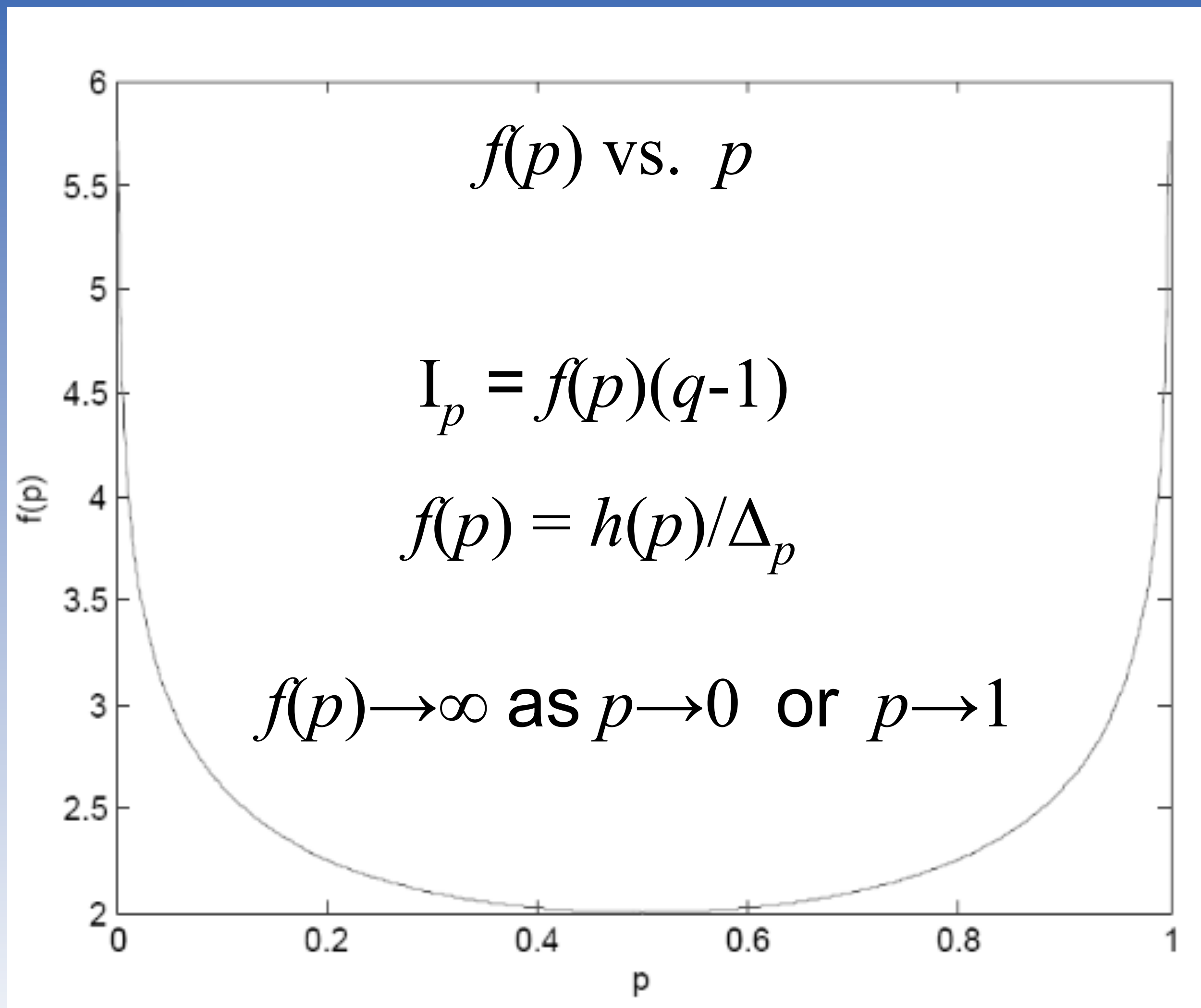


Single Bit Storage

- The maximum number of writes (worst case) is q-1.
- For equiprobable binary inputs, writing a bit increases the cell level with probability ½.
- The expected number of writes before a reset is $2(q-1)$.
- For a biased sequence, $Pr[1] = p$
  - The average increase of the cell level is $\Delta_p = 2p(1-p)$.
  - The expected number of writes is $W_p = (1/\Delta_p)(q-1)$.
  - The average amount of information stored before a reset is $I_p = h(p)W_p = (h(p)/\Delta_p)(q-1)$ bits.

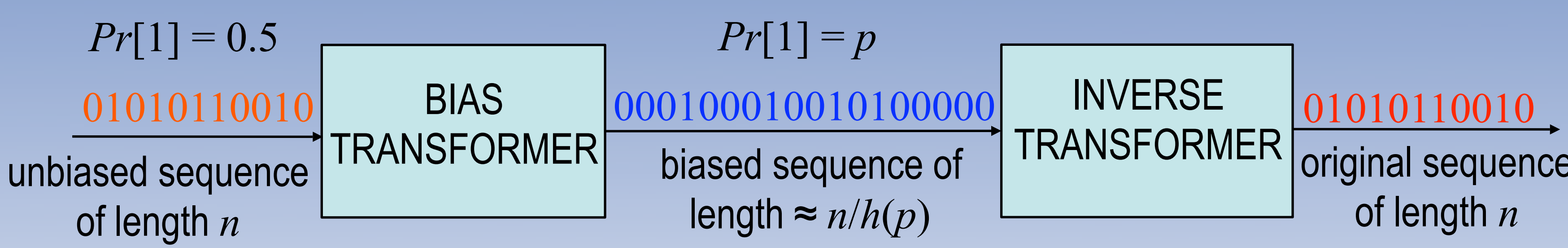


Single Bit Storage



Single Bit Storage

- Converting an unbiased sequence to a biased one:



Extension to Non-binary Inputs

- The previous approach extends to non-binary inputs over an alphabet of size l.
- One input symbol is stored each time.
- The memory may contain multiple q-level cells.
- The sum of all cell levels modulo l indicates the last stored input value.

Single Symbol Storage

- Assume the symbol probabilities are $p_0, p_1, \ldots, p_{l\text{-}1}$
- The average increase of the cell level is

$$\Delta \equiv \Delta(p_0, \ldots, p_{l\text{-}1}) = S_{0 \le i, j < l} \, 2 p_i p_j (i\text{-}j)(\text{mod } l)$$

$$= (l/2)(1\text{-}\ S_{0 \le i < l} \, p_i^{\,2})$$

- The average number of writes

$$W(p_0, \ldots, p_{l\text{-}1}) = (1/\Delta)\ (q\text{-}1)$$

- The average amount of l-ary information stored is

$$I(p_0, \ldots, p_{l\text{-}1}) = h(p_0, \ldots, p_{l\text{-}1}) \times W(p_0, \ldots, p_{l\text{-}1})$$

- Goal: maximize $I(p_0, \ldots, p_{l\text{-}1})$ over all $p_0, \ldots, p_{l\text{-}1}$
- $h(p_0, \ldots, p_{l\text{-}1}) /\ L$ diverges if $p_i \to 1$, for any i.

Single Symbol Storage

- Assuming the symbol sequence is unbiased, the average increase in the cell levels per write is

$$\Delta(1/l,\ldots,1/l) = (l/2)(1 - S_{0\le i<l}(1/l)^2) = (l-1)/2$$

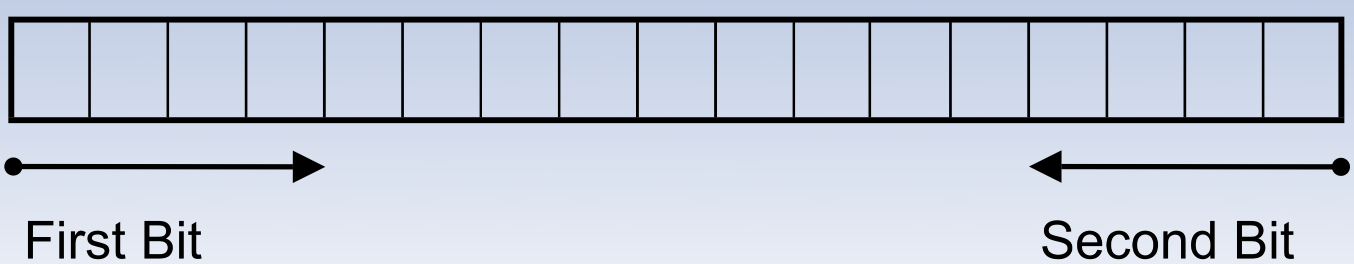
  - For example, for $l=4$, the average increase is $(4-1)/2=1.5$.
  - If however the quaternary symbol is considered as two bits, then
    - Each bit has an average increase of 0.5,
    - In total the average cell level increase is 1.
- It is better to represent the symbols in binary form.
- How to represent more than one bit?

Multiple Bit Storage

- We are interested in efficient storage of more than one bit.
- Floating codes
  - k l-ary symbols are stored using n memory cells.
  - Efficient algorithms:
    – Jiang, Bohossian and Bruck, 2007:
      – $k=l=2$ (two bits) and arbitrary n,q
      – $k=3$, $l=2$ (three bits) and arbitrary n,q
      – The construction for $k=l=2$ is proved to be optimal

Multiple Bit Storage

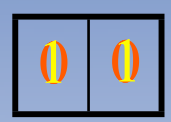
- Example: Another construction for storing two bits in a row of n cells, q-levels each
  - The first bit uses cells from left-to-right.
  - The second bit uses cells from right-to-left.
  - When the written cells intersect, the last cell represents two bits.
  - The maximum number of writes (worst case) is $(n-1)(q-1) + \lfloor (q-1)/2 \rfloor$ before resetting is required.



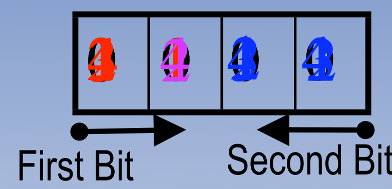
Multiple Bit Storage

- Example: Storing two bits using 4 cells of 5 levels each:

Bits State



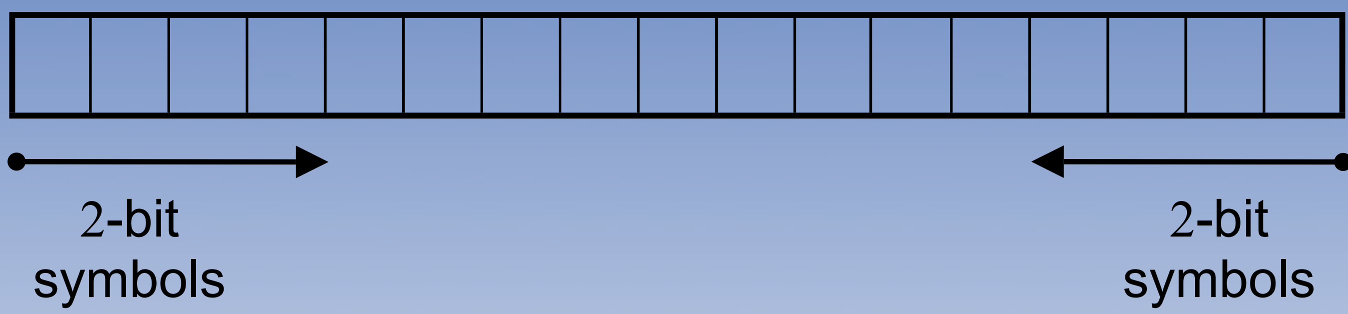
Cells State



- When the last cell represents two bits, its residue modulo 4 sets the bits value:
  - 0 – (0,0)   1 – (0,1)   2 – (1,0)   3 – (1,1)

Multiple Symbol Storage

- Quaternary (2-bit) symbols can also be stored efficiently.
- Using the preceding approach, up to 4 bits can be stored.



- Jiang and Bruck, 2008:
  - Efficient constructions for $3 \leq k \leq 6$ bits.
  - Construction for arbitrary number of bits.
- The problem of optimal storage of an arbitrary number of bits remains open.

Summary

- We presented a model for multi-level flash memory.
- We reviewed recent results pertaining to efficient coding algorithms that aim to minimize the need for memory resets.
- We proposed new algorithms for single-symbol (binary or non-binary) storage.
- We presented some extensions to the case of multiple-bit and multiple-symbol storage.

Acknowledgements

- The authors thank
  - Prof. Alex Vardy (UCSD) for useful discussions and for his joint work on multiple-symbol storage strategies.
  - Amy Chen (UCSD) for a joint project in implementing the results for single-bit storage.