# Flip-N-Write: A Simple Deterministic Technique to Improve PRAM Write Performance, Energy and Endurance

Sangyeun Cho Hyunjin Lee

Presenter: Brian Wongchaowart
March 17, 2010

# Motivation

Suppose that you had a rewritable storage medium with the following characteristics:

- The values of individual bits can be changed independently.
- Updating a bit from 0 to 1 or from 1 to 0 is a relatively expensive operation (in time, energy, or both), compared to the cost of leaving a bit unchanged.

How can you minimize the cost of updating the information stored in this medium?

# Practical Justification: PRAM

- Phase-change random access memory (PRAM) may soon replace flash memory and DRAM in many applications.
- Each memory cell contains a material that has two phases with very different electrical properties.
- An "amorphous phase" exhibits high resistivity, while a "crystalline phase" has much lower resistivity.
- Reading the bit value stored in a cell consists of sensing its resistivity (a fast, low-power operation).

- In order to change the bit value stored in a PRAM cell, the phase-change material must be brought into a different phase by heating.
- Heating the phase-change material to its crystallization temperature for a sufficiently long period of time causes it to assume its crystalline state.
- Heating it to a yet higher temperature for a short period of time makes the material amorphous.
- Both of these operations require high-power current pulses.

# Worst-Case Number of Bit Updates

- Suppose that the storage medium is accessed as an array of $n$-bit words, where $n$ is even.
- Each array element must be able to store one of $2^n$ different logical word values, but the number of bits used to physically represent each logical word and the mapping between the logical word values and their physical representations is unspecified.
- We now consider the problem of limiting the worst-case number of physical bit update operations required to store an arbitrary logical word value to an array element (making the simplifying assumption that updating a bit from 0 to 1 and from 1 to 0 have the same cost).

- If each array element is physically stored as a bit string of length $n$, then there must be a one-to-one mapping between the $2^n$ logical word values that can be stored in the array element and the $2^n$ possible bit strings of length $n$ that can reside on the storage medium.
- In the worst-case, the unique physical representation of a new logical word value to be stored will be the bitwise complement of the bit string currently stored in the medium, meaning that all $n$ bits must be updated.
- Thus limiting the worst-case number of bit update operations requires using at least $n + 1$ bits to store one of $2^n$ logical word values.

# Hamming WEM Codes

- The problem of limiting the worst-case number of bit update operations in this model was formalized in 1989 by Ahlswede and Zhang as a problem in coding theory.[1]
- We would like to store $M$ different messages (logical word values) in a storage medium called a WEM (write-efficient memory).
- Each message $m_i$, for $1 \leq i \leq M$, is associated with a subset $C_i$ of $\{0, 1\}^n$ (the bit strings of length $n$), such that $C_i$ and $C_j$ are disjoint for $i \neq j$.
- Any member of $C_i$ is a valid physical representation of message $m_i$ when stored on the medium.

[1] R. Ahlswede and Z. Zhang, "Coding for Write-Efficient Memory," *Information and Computation* 83, no. 1 (1989): 80–97.

# Hamming WEM Codes

- Suppose that the medium currently holds the bit string $a \in C_i$.
- In order to update the message stored on the medium from $m_i$ to $m_j$, some bit string $b \in C_j$ must be written to the medium.
- Because we want to minimize the number of bit update operations required, we always choose the bit string $b \in C_j$ that minimizes the Hamming distance between $a$ and $b$.
- Our objective is to design a collection $\{C_1, C_2, \ldots, C_M\}$ of pairwise-disjoint subsets of $\{0, 1\}^n$ such that given a bit string $a \in C_i$ for arbitrary $i$, it is possible to transform $a$ into some bit string $b \in C_j$ using no more than $D$ bit update operations for arbitrary $j$. This is called an $(n, M, D)$ Hamming WEM code.

# Flip-N-Write

- We will restrict our attention to the case where $M = 2^n$ for a positive even integer $n$ and each message is stored on the medium as a bit string of length $n + 1$.
- It will be seen that Flip-N-Write is the natural $(n + 1, 2^n, n/2)$ Hamming WEM code for this setting.
- Flip-N-Write was indirectly described by Ahlswede and Zhang in 1989 ("the collection of cosets of a perfect *linear* channel code is a perfect WEM code") and was later independently rediscovered by Sangyeun Cho as a practical technique for PRAM.

# Derivation of Flip-N-Write

- We first show that the best achievable upper bound on the worst-case number of bit update operations is $n/2$, given the assumption that we want to be able to store $2^n$ different messages, where $n$ is even, using $n+1$ bits.
- We then show that the collection of cosets of a binary repetition code of length $n+1$—that is, the perfect binary linear channel code consisting of just the two codewords $0^{n+1}$ and $1^{n+1}$—is a $(n+1, 2^n, n/2)$ Hamming WEM code.

## Lemma

$$\sum_{k=0}^{n/2} \binom{n+1}{k} = 2^n.$$

Proof.

Recall that $\binom{n}{k} = \binom{n}{n-k}$. Then

$$\sum_{k=0}^{n/2} \binom{n+1}{k} = \sum_{k=0}^{n/2} \binom{n+1}{n+1-k} = \sum_{k=n/2+1}^{n+1} \binom{n+1}{k}.$$

But since

$$\sum_{k=0}^{n/2} \binom{n+1}{k} + \sum_{k=n/2+1}^{n+1} \binom{n+1}{k} = \sum_{k=0}^{n+1} \binom{n+1}{k} = 2^{n+1},$$

it follows that

$$\sum_{k=0}^{n/2} \binom{n+1}{k} = \frac{2^{n+1}}{2} = 2^n.$$

□

### Theorem

*Any* $(n+1, 2^n, D)$ *Hamming WEM code satisfies* $D \geq n/2$.

### Proof.

- Suppose that the medium currently holds the bit string $a \in C_i$.
- The number of bit strings of length $n+1$ within Hamming distance $D$ of $a$ is exactly

$$\binom{n+1}{0} + \binom{n+1}{1} + \cdots + \binom{n+1}{D} = \sum_{k=0}^{D} \binom{n+1}{k}.$$

- In order for it to be possible to transform $a$ into some bit string $b \in C_j$ using no more than $D$ bit update operations for $1 \leq j \leq 2^n$, we therefore require that $\sum_{k=0}^{D} \binom{n+1}{k} \geq 2^n$.
- But since $\sum_{k=0}^{n/2} \binom{n+1}{k} = 2^n$ by the preceding lemma, $\sum_{k=0}^{D} \binom{n+1}{k} \geq 2^n$ implies $D \geq n/2$.

□

- We now make a critical observation: if every bit string $c \in \{0,1\}^{n+1}$ is within Hamming distance $D$ of some bit string $b \in C_j$, then it immediately follows that some bit string $b \in C_j$ is within Hamming distance $D$ of any bit string $a \in C_i$ currently stored in the medium.
- A perfect binary linear channel code of length $n+1$ with a minimum Hamming distance of $2D+1$ between codewords guarantees that every bit string $c \in \{0,1\}^{n+1}$ is within Hamming distance $D$ of exactly one codeword.
- This means that if we can find a collection of $2^n$ pairwise-disjoint perfect binary linear codes of length $n+1$ with a minimum Hamming distance of $2(n/2)+1 = n+1$ between codewords, then we immediately have a $(n+1, 2^n, n/2)$ Hamming WEM code.

What subsets of $\{0,1\}^{n+1}$ have a minimum Hamming distance of $n+1$ between any pair of members?

- Answer: The set consisting of any bit string of length $n+1$ and its bitwise complement has a minimum Hamming distance of $n+1$ between any pair of its members.
- These $2^n$ subsets of $\{0,1\}^{n+1}$ are called the *cosets* of the binary repetition code $\{0^{n+1}, 1^{n+1}\}$ because they are the sets of the form $\{c \oplus 0^{n+1}, c \oplus 1^{n+1}\}$, for all $c \in \{0,1\}^{n+1}$, where $\oplus$ denotes the bitwise exclusive OR operator.
- For ease of decoding, we associate the coset $\{c, c \oplus 1^{n+1}\}$ with message $c$ $(0 \leq c \leq 2^n - 1)$ when the leading bit of $c$ is 0.

Theorem

*The collection of cosets of a binary repetition code of length $n+1$ is a $(n+1, 2^n, n/2)$ Hamming WEM code.*

Proof.

- There are $2^n$ cosets of $\{0^{n+1}, 1^{n+1}\}$ because every bit string in $\{0,1\}^{n+1}$ belongs to a coset and each coset has exactly two members. The cosets are pairwise disjoint because two cosets that share a member are obviously identical.
- The spheres of radius $n/2$ centered at the members of a coset both contain $\sum_{k=0}^{n/2} \binom{n+1}{k} = 2^n$ bit strings, but these two spheres do not intersect because the distance between their centers is exactly $n+1$. Thus the union of the two spheres contains all $2^{n+1}$ bit strings in $\{0,1\}^{n+1}$, and each bit string is within Hamming distance $n/2$ of the center of one of the spheres (a member of the coset).

□

# Performance of Flip-N-Write

If each of the $2^n$ logical word values is equally likely to be written, then the probability of having to update $k$ positions in the bit string currently stored in the medium is $\binom{n+1}{k}/2^n$ for $0 \leq k \leq n/2$. The expected number of bit updates is thus $\sum_{k=0}^{n/2} k\binom{n+1}{k}/2^n$.

| $n$ | $\sum_{k=0}^{n/2} \frac{k\binom{n+1}{k}}{2^n}$ | $\sum_{k=0}^{n/2} \frac{k\binom{n+1}{k}}{n \cdot 2^n}$ |
|---:|---:|---:|
| 8 | 3.27 | 0.409 |
| 16 | 6.83 | 0.427 |
| 32 | 14.19 | 0.443 |
| 64 | 29.27 | 0.457 |
| 128 | 59.96 | 0.468 |
| 256 | 122.10 | 0.477 |
| 512 | 247.46 | 0.483 |

# Summary of Flip-N-Write

- We need to represent $2^n$ different logical word values, where $n$ is an even integer.
- Each logical word value is given two different physical representations of length $n+1$ bits: a nonflipped representation in which the word is extended by the addition of a 0-bit, and a flipped representation that is the bitwise complement of the nonflipped representation.
- Given any bit string $a$ of length $n+1$ currently stored in the medium and any logical word value $b$ (the new value to be written), then exactly one of the two physical representations of $b$ is within Hamming distance $n/2$ of $a$.

Questions/Comments