

# $d$ -imbalance WOM Codes for Reduced Inter-Cell Interference in Multi-Level NVMs

Evyatar Hemo and Yuval Cassuto

Department of Electrical Engineering, Technion – Israel Institute of Technology  
evyatar.hemo@gmail.com, ycassuto@ee.technion.ac.il

**Abstract**—In recent years, due to the spread of multi-level non-volatile memories (NVM),  $q$ -ary write-once memories (WOM) codes have been extensively studied. By using WOM codes, it is possible to rewrite NVMs  $t$  times before erasing the cells. The use of WOM codes enables to improve the performance of the storage device, however, it may also increase errors caused by inter-cell interference (ICI). This work presents WOM codes that restrict the imbalance between code symbols throughout the write sequence, hence decreasing ICI. We first specify the imbalance model as a bound  $d$  on the difference between codeword levels. Then a 2-cell code construction for general  $q$  and input size is proposed. An upper bound on the write count is also derived, showing the optimality of the proposed construction. The new codes are also shown to be competitive with known codes not adhering to the bounded imbalance constraint.

## I. INTRODUCTION

In many multi-level non-volatile memory (NVM) technologies there is an inherent asymmetry between increasing and decreasing the level to which a cell is programmed. In particular, in flash memories cell levels are represented by quantities of electrical charge, and removing charge is known to be much more difficult than adding charge. This asymmetry implies significant access limitations, whereby erasing cells must be done simultaneously in large groups of order  $10^6$  cells (called blocks). From this limitation stem many of the serious performance issues of flash, most prominently low write rates and accelerated cell wear.

A possible solution for reducing erasure operations and increasing the lifetime of flash memories is using write-once memory (WOM) codes. The use of WOM codes in flash memories enables multiple writes before executing the costly erasure operation. Indeed, it was shown [10] that by using WOM codes in multi-level NVMs, it is possible to reduce write amplification, and thus increase the lifetime of the device. This justifies the recent extensive study of  $q$ -ary WOM codes [1], [3], [4], [5], [7] that generalize the original binary WOM model [12] to multi-level flash.

In light of this promise, the main issue holding back WOM codes from deployment seems to be concerns related to inter-cell interference (ICI) [2]. Since WOM codes allow updating pages *in place* and non-sequentially, there is a potential risk that these updates will disturb adjacent pages. The risk of ICI disturbance becomes more significant as cell levels are updated to much higher levels than their neighbors.

Hence in this paper we propose ICI mitigating WOM codes that maintain a degree of balance between the physical levels of the cells throughout the write sequence. As a consequence, the level difference between adjacent memory cells is constrained to be up to an imbalance parameter  $d$  chosen for the code. Our main contribution is a  $d$ -imbalance 2-cell code construction that yields codes for general values of  $q$  and

input sizes. We also derive an upper bound on the number of guaranteed writes given the imbalance parameter, and show that our construction is optimal. We conclude with a comparison table showing the numbers of writes our codes offer in favorable comparison even to unconstrained existing codes. The uniqueness of this work over prior ICI codes is that it mitigates ICI within the WOM framework. Whereas known ICI-WOM codes [9] only constrain the transition of individual cells at an individual write, our codes jointly maintain balance between the symbols of the WOM codeword throughout the write sequence. It is possible to extend the balance property beyond 2 cells by concatenating multiple copies of the 2-cell code and using an appropriate inter-codeword update function. The information loss from coding in short 2-cell blocks is minor considering the advantages. For example, it was shown [3] that when the input size  $M$  of the code is order  $\sqrt{q}$ , the ratio between the 2-cell code's sum-rate to the (variable-rate) WOM capacity tends to 1 as  $q$  grows.

## II. BACKGROUND AND DEFINITIONS

### A. Inter-cell interference (ICI) due to lateral charge spreading

In flash memories, the electrical charge of one memory cell can change the charge of its neighboring cells through parasitic capacitance-coupling effect [8]. This effect is referred to as inter-cell interference (ICI), and it is one of the most dominant sources for errors in flash memories [2]. Recently, a new 3D vertical *charge-trap* flash memory was commercially introduced [11]. This flash device was reported to have low ICI due to capacitance-coupling, however, it suffers from ICI due to charge migration between adjacent cells, termed as *lateral charge spreading* effect. It was shown in [6] that if the level difference between adjacent cells is low, the lateral charge spreading effect is significantly reduced. Therefore, a coding scheme that balances the charge levels of adjacent cells is likely to reduce this form of ICI.

### B. WOM codes

Our focus in this paper is on limited-imbalance codes in the WOM model, because the in-place re-writing of WOM codes makes them especially prone to ICI. We first review some necessary background on  $q$ -ary WOM codes.

**Definition 1.** A fixed rate **WOM code**  $C(n, q, t, M)$  is a code applied to a size  $n$  block of  $q$ -ary cells, and guaranteeing  $t$  writes of input size  $M$  each.

A WOM code is specified through a pair of functions: the *decoding* and *update* functions.

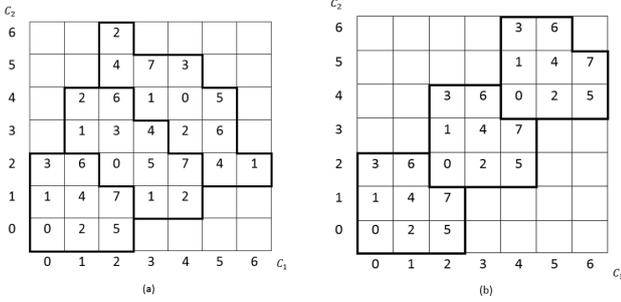
**Definition 2.** The *decoding* function defined as  $\psi : \{0, \dots, q-1\}^n \rightarrow \{0, \dots, M-1\}$ , which maps the current levels

of the  $n$  cells to the data input in the most recent write. The update function is defined as  $\mu : \{0, \dots, q-1\}^n \times \{0, \dots, M-1\} \rightarrow \{0, \dots, q-1\}^n$ , which specifies the new cell levels as a function of the current cell levels and the new data value at the input. By the WOM requirement, the  $i$ -th cell level output by  $\mu$  cannot be lower than the  $i$ -th cell level in the input.

**Definition 3.** The code's **physical state** is defined as the  $n$   $q$ -ary levels to which the cells are currently programmed. The code's **logical state** is the data element from  $\{0, \dots, M-1\}$  returned by  $\psi$  on the current physical state.

A write region spanned from a physical state is a set of physical states accessible from it under the WOM requirement. The size of this set we call the *area* of the write region. If at a given physical state the code admits more write(s), then this physical state must span a write region with area at least  $M$ .

**Example 1.** Let us consider two sample WOM codes. In Fig. 1 (a) we have the decoding function of  $C(n=2, q=7, t=3, M=8)$  constructed by Construction 3 in [3]. This code is applied on a pair of  $q=7$ -level memory cells, enabling  $t=3$  guaranteed writes of size  $M=8$  each. In Fig. 1 (b) we have a code  $C(n=2, q=7, t=3, M=8)$ , offering the same number of writes. Considering Fig. 1 (a),



**Figure 1.** Sample  $n=2$  WOM constructions (from [3]). (a) - Decoding function  $\psi$  for a code  $C(2,7,3,8)$ . (b) - Decoding function  $\psi$  for another code  $C(2,7,3,8)$ . Physical states are represented by  $(c_1, c_2)$  and logical states are labeled inside each square.

let us assume we want to perform three writes of the logical states 7, 6 and 2 using this WOM code. For the first write the logical state is 7 and the physical state is (2, 1). When updating the logical state to 6, the physical state becomes (2, 4). For the third write of 2, the physical state becomes (2, 6). After the third write, we reach a physical state with level difference of 4 between the cells. As a consequence, given that the pair of cells are adjacent, cell 1 is likely to suffer from ICI. The code in Fig. 1 (b) maintains a better balance between the two cell levels, but will offer fewer writes if extended beyond  $q=7$ .

In order to reduce ICI, we now define the  $d$ -imbalance model for WOM codes.

**Definition 4.** A  $d$ -imbalance WOM code  $C_{d-imb}(n, q, t, M)$  is a WOM code that guarantees that after **each write** the physical states of the cells  $c_i$ ,  $1 \leq i \leq n$ , must satisfy

$$\max_{i,j:i \neq j} |c_i - c_j| \leq d, \quad (1)$$

for any write sequence.

A  $d$ -imbalance code guarantees that the level imbalance between cells cannot exceed  $d$ . Therefore, all cells sustain similar

(same up to  $d$ ) levels of charge injection, thus imposing control on the ICI disturbance. When  $d = q-1$ , we get a standard unconstrained WOM code without balancing properties. As  $d$  decreases, the balancing improves, but the added constraints may lead to lower re-write capabilities.

### III. OPTIMAL $d$ -IMBALANCE CONSTRUCTION

Before showing our main construction, we prelude this section with a discussion on which  $d$  parameters would be interesting to consider. Given  $n$  and  $M$ , the  $d$  imbalance parameter of a code  $C_{d-imb}(n, q, t, M)$  cannot be less than  $\lceil \sqrt[n]{M} \rceil - 1$ . That is because in any physical state, at most  $(d+1)^n$  states are accessible for the next write while keeping the  $d$  imbalance constraint. So to be able to write any of the  $M$  values in the next write we must have  $M \leq (d+1)^n$ .

**Example 2.** For  $n=2$  and  $M=8$ , the lowest possible imbalance parameter is  $d=2$ . It turns out that for this extreme case the simple “diagonal stacking” construction of Fig. 1(b) is an optimal  $C_{2-imb}(2, q, t, 8)$  code with  $t = \lfloor (q-1)/2 \rfloor$  writes. It is straightforward to generalize this construction to produce  $d = (a-1)$ -imbalance codes  $C_{a-1-imb}(2, q, t, a^2-1)$ , for any  $2 < a \in \mathbb{Z}$ , and providing  $t = \lfloor (q-1)/(a-1) \rfloor$  writes.

Requiring maximal balance (minimal  $d$ ) results in weak codes with small numbers of writes. A better tradeoff between balancing and re-write efficiency is obtained when  $d$  is relaxed from the extreme value, in which case good balancing (low ICI errors) is achieved while getting more writes. This will be the case we handle in our following construction.

#### A. Construction

We now turn to present a construction where  $d$  is 1 larger than in the extreme case of Example 2.

**Construction 1.** For any  $q$ , we define an  $n=2$  WOM code with  $M = a^2 - 1$ ,  $2 < a \in \mathbb{Z}$ , and  $d = a$ -imbalance parameter as follows.

##### 1) Decoding function:

The decoding function is specified in Fig. 2. The number shown at position  $(c_1, c_2)$  represents the logical state as returned by the decoding function  $\psi(c_1, c_2)$ .

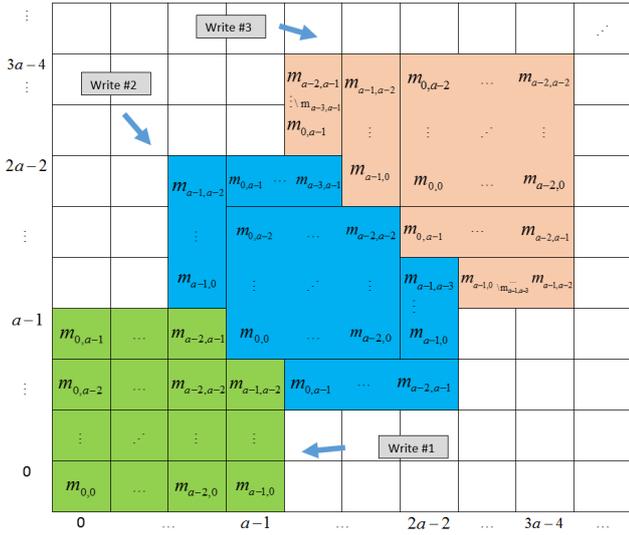
##### 2) Update function:

The update function is specified with the aid of 3 distinctly colored regions in Fig. 2 which represent the worst case regions of the 3 writes. The update function determines the new physical state  $(c'_1, c'_2)$  given the current state  $(c_1, c_2)$  and the new value to be written  $m$  as follows:

- locate all physical states with  $(c''_1, c''_2) \geq (c_1, c_2)$  element-wise, for which  $\psi(c''_1, c''_2) = m$ .
- $(c'_1, c'_2)$  is chosen as the pair  $(c''_1, c''_2)$  that minimizes the sum of coordinates  $|c''_1, c''_2| - (c_1, c_2)|$ .

The bottom-left region in Fig. 2 has all  $M = a^2 - 1$  logical states  $m_{i,j}$ ,  $0 \leq i, j \leq a-1$  excluding  $i = j = a-1$ , accessible for the first write. Each of the other two regions has all the  $M = a^2 - 1$  logical states accessible from every physical state in the region to the left and down. Hence the update function supports any sequence of 3 written values without exceeding the top-right region.

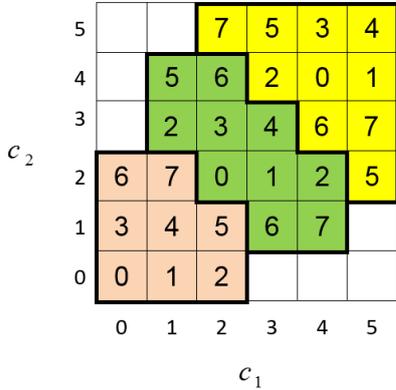
We next examine the imbalance parameter of Construction 1. It is straightforward to see from Fig. 2 that all physical states  $(x, y)$  used by the update function satisfy  $|y-x| \leq a$ , as required.



**Figure 2.** Decoding function of the  $d = a$ -imbalance WOM code  $C_{a-imb}(2, q, t, a^2 - 1)$ . The notation  $m_{a-1,0} \dots m_{a-1,a-2}$  represents all the logical states  $m_{a-1,0}$  to  $m_{a-1,a-2}$  excluding  $m_{a-1,a-3}$ . The three colored regions represent the first worst case regions of the three writes of this code.

Before discussing the extension of Construction 1 beyond 3 writes, we give an example for the special case  $a = 3$ .

**Example 3.** In this example, we demonstrate Construction 1 for  $M = 8$  (corresponding to  $a = 3$ ), and  $q = 6$ . The 3 writes of the resulting code  $C_{3-imb}(2, 6, 3, 8)$  are presented in Fig. 3, where the labels  $m_{i,j}$  are given by  $m_{i,j} = i + ja$ . It can be checked that



**Figure 3.** Decoding function and update regions for the  $d = 3$ -imbalance WOM code  $C_{3-imb}(2, 6, 3, 8)$ . The three colored regions represent the worst case regions of the three writes of the code.

for any sequence of 3 write logical states the update function of Construction 1 can succeed without exceeding level 5 at any of the cells.

**Extending Construction 1 to general  $q$ .** To extend the decoding and update functions of Construction 1 to general  $q$ , we copy the three regions of Fig. 2 and lay out the copies such that the origin of a new copy is placed on the top-right corner of the previous copy. Note that such extension requires us to relabel the logical states along the main diagonal because the origin logical state and the top-right logical state are different ( $m_{0,0}$  vs.  $m_{a-2,a-2}$ ). Observe in Fig. 2 that the logical-state values on the main diagonal are of the form  $m_{i,i}$ , and that these

values do not appear elsewhere in the two-dimensional array. This means that we can lay out these values in a cyclic fashion across copies. That is, at physical state  $(j, j)$  in the extended array we place logical value  $m_{j \bmod a-1, j \bmod a-1}$ . Apart from the main diagonal, the extended copies of the three regions have the same assignment of logical values as the base copy of Fig. 2. In Example 3 we place the origin of a second copy at physical state  $(5, 5)$ , and get 3 more writes while relabeling the main diagonal from  $0, 4, 0, 4, 0, 4$  in the first copy, to  $4, 0, 4, 0, 4, 0$  in the second copy.

We now derive the number of guaranteed writes of a  $d$ -imbalance code produced by Construction 1.

**Theorem 1.** For any  $q$  and  $2 < a \in \mathbb{Z}$ , a  $d = a$ -imbalance WOM code  $C_{a-imb}(2, q, t, a^2 - 1)$  constructed by Construction 1 guarantees

$$t = \left\lfloor \frac{3(q-1)}{3a-4} \right\rfloor \quad (2)$$

writes.

*Proof:* With the periodic extension of Fig. 2, we know that for  $t = 3\ell$  writes,  $\ell$  integer,  $q - 1 = \ell(3a - 4)$  is sufficient. Substituting  $\ell = t/3$  we get  $t = 3(q - 1)/(3a - 4)$  as required. To complete the proof, we need to show (2) for  $t = 3\ell + r$  writes, also for the cases  $r = 1, 2$ . In these cases, the last  $r$  writes each increases  $q$  by  $a - 1$ . Therefore, we have

$$q - 1 = \ell(3a - 4) + r(a - 1). \quad (3)$$

Substituting  $\ell = (t - r)/3$  and rearranging, we get

$$t = \frac{3(q-1) - r}{3a-4}. \quad (4)$$

It appears that the expression in the right-hand side of (4) may be smaller than the right-hand side of (2). We show that this cannot happen. From (3) we know that  $q - 1 \equiv r(a - 1) \pmod{3a - 4}$ . Therefore,  $3(q - 1) \equiv 3r(a - 1) \equiv r(3a - 3) \equiv r \pmod{3a - 4}$ . Now expanding (2), we get

$$\left\lfloor \frac{3(q-1)}{3a-4} \right\rfloor = \frac{3(q-1)}{3a-4} - \frac{3(q-1) \bmod (3a-4)}{3a-4} = \frac{3(q-1) - r}{3a-4}, \quad (5)$$

which proves (2) for all  $t$ . The fact that the periodic extension of Construction 1 has  $d = a$  is immediate from Fig. 2. ■

Substituting into (2) the special case  $a = 3$ ,  $q = 6$ , given in Example 3, we indeed get  $t = 3$ .

### B. Upper bound on the guaranteed number of writes

We now derive an upper bound on the number of guaranteed writes of a  $d$ -imbalance code that shows that Construction 1 gives optimal codes. Optimality will be proved for the special case  $a = 3$ , that is, for codes with  $M = 8$  and  $d = 3$  imbalance. A similar technique can extend the upper bound to more general  $a$  values. We start with the following definitions and lemmas.

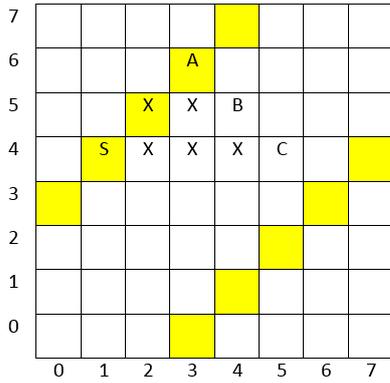
**Definition 5.** If a WOM code  $i$ -th write starts at state  $(x_i, y_i)$  and ends at state  $(x_{i+1}, y_{i+1})$ , then the (non-negative) **write sum** of the  $i$ -th write is defined as  $x_{i+1} - x_i + y_{i+1} - y_i$ .

The write sum is a powerful technique to obtain upper bounds. For  $M = 8$  it has been shown [3] that without balancing constraints, write sum of 3 is both sufficient and necessary for

every write. The following lemma is key to our upper bound, because it shows cases where a write sum of 4 is necessary.

**Lemma 2.** For any code  $C_{3\text{-imb}}(2, q, t, 8)$ , if a write starts from state  $(x, y)$  satisfying  $|y - x| = d = 3$ , then the write region of  $(x, y)$  must contain at least two states of write sum 4 (or higher).

*Proof:* Let us assume w.l.o.g that the start state is state S showing on Fig. 4. All 5 states marked as X have write sum of 3 or less. The write sum of states A, B, and C is 4. Therefore, in order for the write region to have area at least  $M = 8$ , it must include at least two additional states out of A, B, C, or some other state with higher write sum. ■

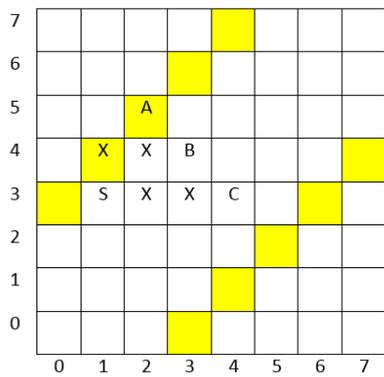


**Figure 4.** Proof of Lemma 2 – only 5 states (marked by X) have write sum of 3 or less. States not on or between the shaded diagonals are forbidden due to imbalance greater than  $d = 3$ .

The next two lemmas show how any 3-imbalance code must get to the “problematic” state S of Fig. 4.

**Lemma 3.** For any code  $C_{3\text{-imb}}(2, q, t, 8)$ , if a write starts from state  $(x, y)$  satisfying  $|y - x| = d - 1 = 2$ , then the write region of  $(x, y)$  must contain at least three states of write sum 3 (or higher), at least one of which has  $|y' - x'| = d = 3$  or write sum at least 4.

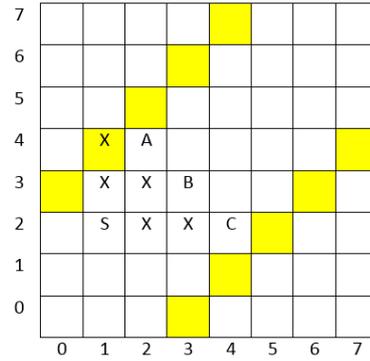
*Proof:* Let us assume w.l.o.g that the start state is state S showing on Fig. 5. All 5 states marked as X have write sum of 2 or less. The write sum of states A, B, and C is 3. Therefore, in order for the write region to have area at least  $M = 8$  with write sum 3, it must include the states A, B, C, and state A has  $|y' - x'| = d = 3$ . If A is not included, then a state with write sum 4 is required. ■



**Figure 5.** Proof of Lemma 3 – state A is required for write sum of 3 or less.

**Lemma 4.** For any code  $C_{3\text{-imb}}(2, q, t, 8)$ , if a write starts from state  $(x, y)$  satisfying  $|y - x| = d - 2 = 1$ , then the write region of  $(x, y)$  must contain at least two states of write sum 3 (or higher), at least one of which has  $|y' - x'| = d - 1 = 2$  or write sum at least 4.

*Proof:* Let us assume w.l.o.g that the start state is state S showing on Fig. 6. All 5 states marked as X have write sum of 2 or less. The write sum of states A, B, and C is 3. Therefore, in order for the write region to have area at least  $M = 8$  with write sum 3, it must include two states out of A, B, C, and both A,C have  $|y' - x'| = d - 1 = 2$ . If neither of A,C is included, then a state with write sum 4 is required. ■



**Figure 6.** Proof of Lemma 4 – A or C are required for write sum of 3 or less.

We are now ready to state the upper bound.

**Theorem 5.** Given a  $d = 3$ -imbalance WOM code  $C_{3\text{-imb}}(2, q, t, 8)$ , the number of guaranteed writes is upper bounded by

$$t \leq \left\lfloor \frac{3(q-1)}{5} \right\rfloor. \quad (6)$$

*Proof:* Throughout the proof we use lower bounds on write sums, but for convenience we omit the term “at least” when we state the value of the write sums. If write sums are strictly larger than the values quoted below, the proof is still correct, and we may even reach the desired lower bound on total write sum earlier by skipping lemmas that assume the lower quoted value. By a simple area argument, after the first write any  $M = 8$  WOM code needs to use a state with write sum 3. Next we show that we can invoke Lemmas 4, 3, 2 in sequence to get lower bounds on the write sums of the second, third, and fourth write, respectively. Because all states with sum 3 have  $|y - x|$  at least 1, the conditions of Lemma 4 are satisfied after the first write (see state S in Fig. 6). By Lemma 4, after the second write any code needs a state that satisfies the conditions of Lemma 3 (see state A in Fig. 6 and state S in Fig. 5). By Lemma 3, after the third write any code needs a state that satisfies the conditions of Lemma 2 (see state A in Fig. 5 and state S in Fig. 4). Altogether we conclude that for the second, third, and fourth writes we need write sums of  $3 + 3 + 4 = 10$ . From Lemma 2, after the fourth write there are *two* states of sum 3, which implies that one of them has  $|y - x|$  at least 1, and we can again invoke Lemmas 4, 3, 2 in sequence for the subsequent three writes requiring additional  $3 + 3 + 4 = 10$  write sums. Continuing this argument periodically, for  $t = 1 + 3\ell$  writes,  $\ell$  integer, we need a total write sum  $s \geq 3 + 10\ell$ , and

since  $s \leq 2(q-1)$ , we get  $2(q-1) \geq 3 + 10\ell = 3 + 10(t-1)/3$ . Rearranging, we get  $t \leq \lceil 3(q-1)/5 + 1/10 \rceil = \lceil 3(q-1)/5 \rceil$  as needed. To complete the proof, we need to show (6) for  $t = 1 + 3\ell + r$  writes, for the cases  $r \in \{1, 2\}$ . In this case, the last  $r$  writes each requires write sum of 3, and we get  $s \geq 3 + 10\ell + 3r$ . Because both Lemmas 4, 3 show existence of two states with write sum 3, one of these final states has  $|y-x| \geq 1$ . Thus we can tighten the relation between  $s$  and  $q$  to  $s \leq x+y \leq q-1+q-2 = 2q-3$ . Now we write  $2q-3 \geq 3 + 10\ell + 3r = 3 + 10(t-1)/3 - r/3$ . Rearranging, we get  $t \leq \lceil 3(q-1)/5 - 1/5 + r/10 \rceil \leq \lceil 3(q-1)/5 \rceil$ , and for the last inequality we used the fact that  $r \leq 2$ . ■

Note that for  $a = 3$  the number of writes guaranteed by Construction 1 is  $\lceil 3(q-1)/5 \rceil$ , which is equal to the upper bound (6), hence Construction 1 is optimal.

### C. Performance comparison

Table I presents a summary of known results and bounds for 2-cell,  $M = 8$  WOM codes [3]. By examining the table, we can notice that for  $q = 8$ , and  $q = 16$  (which are currently the practical values of  $q$  for NVMs), using Construction 1 does not compromise the number of writes compared to unconstrained WOM, while it provides a better imbalance  $d = 3$ . Using  $d = 2$  constructions does compromise the number of writes for all  $q$  values, including  $q = 8$ . It was also verified numerically that for  $M = 8$ ,  $q \leq 16$  and  $d = 3$ -imbalance, codes constructed by Construction 1 reach the write-count upper bound of unconstrained codes for every value of  $q$  in this range.

Actually, as we can see in the next corollary, WOM codes

TABLE I  
NUMBER OF WRITES FOR 2-CELL,  $M = 8$  WOM CODES

q	t - Upper bound d-unconstrained	t - Construction 1 d = 3	t - Construction in [3] d = 2
8	4	4	3
16	9	9	7
20	12	11	9
32	20	18	15

constructed by Construction 1 are good WOM codes even if ignoring the  $d$ -imbalance property.

**Corollary 6.** When  $q \geq 1 + \left\lceil \frac{(a^2-2)(3a-4)}{a-2} \right\rceil$ , a WOM code  $C_{a-imb}(2, q, t, a^2 - 1)$ ,  $2 < a \in \mathbb{Z}$  from Construction 1 guarantees higher number of writes than a WOM code  $C(2, q, \tilde{t}, a^2 - 1)$  constructed by Construction 2 in [3].

*Proof:* The number of writes guaranteed by Construction 1 is given by Theorem 1, while the number writes guaranteed by Construction 2 in [3] is given by

$$t = \left\lceil \frac{(q-1)(a+1)}{a^2-2} \right\rceil. \quad (7)$$

So, we look for the lowest value of  $q$  which guarantees strictly higher number of writes for the  $a$ -imbalance code. Due to the floor function applied on the number of writes, we demand that

$$\frac{3q-3}{3a-4} \geq \frac{(q-1)(a+1)}{a^2-2} + 1. \quad (8)$$

This inequality holds for  $q \geq 1 + (a^2 - 2)(3a - 4) / (a - 2)$ . Ceiling this expression ends the proof. ■

In addition to offering strictly more writes for these  $q$  values, the codes  $C_{a-imb}(2, q, t, a^2 - 1)$  have at least as many writes as  $C(2, q, \tilde{t}, a^2 - 1)$  for all values of  $q$ . This makes them the best known (unconstrained) WOM codes for these parameters and general  $a$  (for  $a = 3$  [3] has better codes than  $C(2, q, \tilde{t}, a^2 - 1)$ .)

We can also notice that the number of guaranteed writes offered by Construction 1 can reach the unconstrained upper bound for some other values of  $M$  and  $q$ . Table II presents such pairs of  $M$ ,  $q$  values (the  $q$  values are taken in the practical range  $8 \leq q \leq 16$ ).

TABLE II  
 $M$ ,  $q$  VALUES FOR WHICH CONSTRUCTION 1 ATTAINS THE (UNCONSTRAINED) UPPER BOUND.

M	values of q
15	9
24	9,10,12,13,16
35	8,11,15
48	8,9,10,13,14

### IV. CONCLUSION AND FUTURE WORK

In this work we presented  $d$ -imbalance WOM codes designed to reduce inter-cell interference in multi-level NVMs. General simple to implement construction was given and analyzed. We also derived an upper bound on the number of guaranteed writes of a  $d$ -imbalance WOM code and showed that our proposed construction is optimal for some parameters of the code. Future work can include extending the presented two-cell WOM codes to WOM codes for  $n \geq 3$ .

### V. ACKNOWLEDGMENT

This work was supported by the Israel Science Foundation, by the Israel Ministry of Science and Technology, and by a GIF Young Investigator grant.

### REFERENCES

- [1] A. Bhatia, M. Qin, A.R. Iyengar, B.M. Kurkoski, P.H. Siegel, "Lattice-based WOM codes for multilevel flash memories," IEEE J. Sel. Areas Commun., vol.32, no.5, pp.933-945, May 2014.
- [2] Y. Cai, O. Mutlu, E.F. Haratsch, and K. Mai, "Program interference in MLC NAND flash memory: Characterization, modeling, and mitigation," in IEEE 31st Inter. Conf. on Computer Design, pp. 123-130, Oct. 2013.
- [3] Y. Cassuto, E. Yaakobi, "Short q-ary fixed-rate WOM codes for guaranteed rewrites and with hot/cold write differentiation," IEEE Trans. Inform. Theory, vol.60, no.7, pp.3942-3958, July 2014.
- [4] R. Gabrys, E. Yaakobi, L. Dolecek, S. Kayser, P.H. Siegel, A. Vardy, and J.K. Wolf, "Non-binary WOM-codes for multilevel flash memories," Proc. IEEE Inform. Theory Workshop, pp. 40-44, Brazil, Oct. 2011.
- [5] K. Haymaker and C.A. Kelley, "Geometric WOM codes and coding strategies for multilevel flash memories," arXiv:1206.5762v1, May 2012.
- [6] Y. Kim, et al. "Modulation coding for flash memories," in Computing, Networking and Communications (ICNC), 2013 International Conference on, pp.961-967, 28-31 Jan. 2013.
- [7] B. Kurkoski, "Lattice-based WOM codebooks that allow two writes," in International Symp. on Inform. Theory and its Applications, Honolulu, Hawaii, pp. 101-105, Oct. 2012.
- [8] J.D. Lee, S.H. Hur, and J.D. Choi, "Effects of floating-gate interference on NAND flash memory cell operation," IEEE Electron. Device Lett., vol. 23, no. 5, pp.264266, May 2002.
- [9] Q. Li, "WOM codes against inter-cell interference in NAND memories," in Communication, Control, and Computing (Allerton), 49th Annual Allerton Conference on, pp.1416-1423, 28-30 Sept. 2011.
- [10] S. Odeh and Y. Cassuto, "NAND flash architectures reducing write amplification through multi-write codes," IEEE 30th Symposium on Mass Storage Systems and Technologies (MSST), pp.1-10, June 2014.
- [11] K.T. Park, et al. "A world's first product of three-dimensional vertical NAND Flash memory and beyond," in 14th Annual Non-Volatile Memory Technology Symposium (NVM-TS), pp.1-5, 27-29 Oct. 2014.
- [12] R. L. Rivest and A. Shamir, "How to reuse a write-once memory," Information and Control, vol. 55, no. 1, pp. 1-19, Dec. 1982.