Rewriting Flash Memories and Dirty-Paper Coding

Brian M. Kurkoski
Japan Advanced Institute of Science and Technology
Nomi, Japan
kurkoski@jaist.ac.jp

***Abstract*—This paper considers that write-once memory (WOM) codes can be seen as a type of dirty-paper code. The current state of the memory, which is known to the encoder, plays the role of the known interference of dirty-paper coding. Erez, Shamai and Zamir showed that lattice strategies can achieve the capacity of the known-interference channel. In this paper, lattices are used to design a WOM code. Encoding is performed modulo a shaping lattice with respect to a lattice fundamental region to obtain a codeword, to be added to the current state of the memory. The fundamental region is designed to accommodate the limitations of the flash memory system, particularly, that values can only increase. The criterion for evaluation is average number of writes. In order to improve the average number of writes, "coset select" bits are introduced, to maximize the average number of writes. For an eight-dimensional lattice, numerical results for practical parameter choices show a promising trend.**

I. Introduction

Multi-level flash memories allow storing one of q multiple values in each of n cells. These values can easily be increased, but must be decreased using an expensive erase operation. Furthermore, erase operations cause the flash memory to physically deteriorate and eventually fail. Write-once memory codes, or WOM codes [1], are now being investigated by the coding theory community in order to mitigate this problem.

Flash memories use error-correcting codes. When a binary error-correcting code is applied to multi-level flash memory, typically using Gray coding, the codewords form a sphere packing [2]. Such sphere packings are not linear. On the other hand, lattices are linear sphere packings. Clearly, linear structures have a certain advantages over non-linear ones, and the error-correcting properties of lattices have been extensively studied [3, Ch. 3]. Recently, studies of the WOM properties of lattice-based codes have begun [4], [5], [6]. A lattice-based WOM code inherits the error-correcting properties of a lattice, and thus is an appealing structure that both corrects errors and allows rewriting.

A seemingly different problem, communication in the presence of known interference, has also received attention because of the result that interference known to the encoder does not reduce capacity [7]. Dirty-paper coding provides a method to reduce the effects of interference, for example, in wireless communications settings. Erez, Shamai and Zamir showed that lattice strategies can achieve this result [8] [9], by computing signals modulo a shaping lattice.

This paper takes the perspective that lattice WOM coding is a type of dirty-paper coding problem. The current state of the memory $\mathbf{s}$, which is known to the encoder, plays the role of the known interference. Encoding is performed modulo a shaping lattice with respect to a fundamental region to obtain a codeword $\mathbf{x}$. In wireless systems, the interference is implicitly added by the channel, but in flash memory systems, the state of the memory is explicitly changed to $\mathbf{x}+\mathbf{s}$.

However, WOM codes and wireless communication systems have different power constraints. Wireless systems have an average power constraint on $\mathbf{x}$, but $\mathbf{x}$ may take on positive and negative values and there is no constraint on $\mathbf{x}+\mathbf{s}$. In lattice WOM codes, there is a peak power constraint on $\mathbf{x}+\mathbf{s}$, and $\mathbf{x}$ must have positive values. The peak power constraint corresponds to the maximum value V that the flash cell can store. The positive-value constraint corresponds to only allowing increases in the cell values.

Accordingly, in this paper, a lattice WOM code is designed using dirty-paper coding lattice strategies, but the fundamental region is tuned for the restrictions of the problem. The proposed lattice WOM code is shaped using a cubic (or rectangular) fundamental region with only positive values. The criterion for evaluation is average number of writes. In order to improve the average number of writes, "coset select" bits are introduced. This results in a one-to-many mapping from information bits to codewords. Then, codeword which maximizes the average number of writes is selected.

In Sec. II, background on lattice codes and fundamental regions is given. In Sec. III, a simplified description of dirty paper coding, lattice WOM codes, and the comparisons, are given. In Sec. IV, the proposed lattice WOM code construction is given. In Sec. V, numerical results are given. The paper concludes with discussion in Sec. VI.

II. Lattices and Lattice Codes

A. Lattices and Fundamental Regions

An n-dimensional lattice Λ is an infinite additive group, and it is a subgroup of $\mathbb{R}^n$. A lattice is defined by a generator matrix G, and is the set of all points:

$$\mathbf{x} = G \cdot \mathbf{b}, \tag{1}$$

where $\mathbf{b} \in Z^n$, the set of integer vectors.

For a lattice Λ, a fundamental region Ω has the property that any $\mathbf{x} \in \mathbb{R}^n$ can be uniquely written as $\mathbf{x} = \omega + \lambda$, with

This research was supported in part by the Ministry of Education, Science, Sports and Culture; Grant-in-Aid for Scientific Research (C) number 23560439.

An earlier version of this work was presented at the Information Theory Workshop 2011, University of California San Diego.

$\omega \in \Omega$ and $\lambda \in \Lambda$. The Voronoi fundamental region $\mathcal{V}$, the set of points in $\mathbb{R}^n$ closer to $\mathbf{0}$ than any other member of Λ, is an important fundamental region.

Consider the operation of modulo Λ with respect to a fundamental region Ω [10]. A point $\mathbf{x}$ is quantized to $\lambda \in \Lambda$ with respect to Ω as:

$$Q_\Omega(\mathbf{x}) \quad = \quad \lambda, \tag{2}$$

if $\mathbf{x} \in \lambda + \Omega$. Then, the operation modulo Λ with respect to Ω is given by:

$$\mathbf{x} \bmod_\Omega \Lambda = \mathbf{x} - Q_\Omega(\mathbf{x}) \tag{3}$$

Two fundamental regions of interest are the Voronoi region $\mathcal{V}$, and the n-cube $\mathbb{B}_V$:

$$\mathbb{B}_V \quad = \quad \{\mathbf{x} \in \mathbb{R}^n | 0 \leq x_i < V\}, \tag{4}$$

where the side length is V. For a thorough treatment of lattices, refer to [3].

B. Lattice codes and Their Labeling

To construct a lattice code $\mathcal{C}$, one approach is to use a fundamental region Ω as a shaping region, so that:

$$\mathcal{C} \quad = \quad \Lambda \cap \Omega. \tag{5}$$

Infinite lattices Λ are distinguished from finite lattice codes $\mathcal{C}$.

Labeling is assigning information labels, integers or bits, to the members of $\mathcal{C}$. When Ω is the Voronoi fundamental region $\mathcal{V}$ a labeling scheme exists [11]. However, it requires computationally demanding quantization.

When the fundamental region is a cube $\Omega = \mathbb{B}_M$, efficient labeling is possible if the following conditions are satisfied. G is a lower-triangular matrix:

$$G \quad = \quad \begin{bmatrix} g_{11} & 0 & 0 & \cdots & 0 \\ g_{21} & g_{22} & 0 & \cdots & 0 \\ g_{31} & g_{32} & g_{33} & \cdots & 0 \\ \vdots & & & & \vdots \\ g_{n1} & g_{n2} & g_{n3} & \cdots & g_{nn} \end{bmatrix} \tag{6}$$

and normalized such that $|\det(G)| = 1$. The codebook $\mathcal{C} = \Lambda \cap \mathbb{B}_M$ has M^n points. It is required that M/g_{ii} is an integer for all $i = 1, \ldots, n$. $\mathbf{b} \in \mathcal{B}$ is a vector of integers with $b_i \in \{0, 1, \ldots, \frac{M}{g_{ii}} - 1\}$. Generally, $G \cdot \mathbf{b}$ is not in $\mathcal{C}$, but:

$$\mathbf{x} \quad = \quad G \cdot \mathbf{b} \bmod_{\mathbb{B}} M\Lambda \tag{7}$$

is in $\mathcal{C}$. Integers representing

$$\sum_{i=1}^{n} \log_2 \frac{M}{g_{ii}} \tag{8}$$

bits of information can be encoded to the M^n points of $\mathcal{C}$, expressed as a mapping g:

$$g \;:\; \mathcal{B} \rightarrow \mathcal{C}. \tag{9}$$

Details of encoding are straightforward, but are omitted, refer to [12], [4].

Decoding, that is, the mapping $\mathcal{C} \rightarrow \mathcal{B}$ is likewise straightforward. If $\mathbf{x} \in \mathcal{C}$, then let $\mathbf{b}' = G^{-1}\mathbf{x}$. The encoded integers then are:

$$b_i \quad = \quad b'_i \mod \frac{M}{g_{ii}} \tag{10}$$

for $i = 1, 2, \ldots, n$.

III. Dirty Paper Coding and WOM Codes

A. Background on Dirty Paper Coding

Dirty paper coding is a technique to mitigate the effect of interference at the receiver [7] [9]. Lattice coding methods can be used to achieve the well-known result "known interference does not reduce capacity" on the AWGN channel [8], and a simplified summary is given here.

A dirty paper codebook $\mathcal{C}_{\text{dpc}}$ is constructed using a "shaping lattice" Λ_{s} and a fundamental region Ω for this lattice, that is, $\mathcal{C}_{\text{dpc}} = \Lambda \cap \Omega$. Λ_{s} may be a scaled version of Λ. Suppose that the transmitter intends that the receiver should decode $\mathbf{x} \in \mathcal{C}_{\text{dpc}}$, but in the presence of known additive interference $\mathbf{s}$. That is, if $\mathbf{x}$ is transmitted, then $\mathbf{x} + \mathbf{s}$ is received (recall channel noise is being ignored). The transmit power constraint is $\mathbf{x} \in \mathcal{C}_{\text{dpc}}$. The transmitter cannot pre-subtract $\mathbf{s}$ because this may violate the transmit power constraint. Instead, the transmitter sends:

$$(\mathbf{x} - \mathbf{s}) \bmod_\Omega \Lambda_{\text{s}}, \tag{11}$$

which does satisfy the power constraint. The received signal is the above expression plus $\mathbf{s}$. The receiver correctly finds the intended codeword $\mathbf{x}$ by computing the received signal $\bmod_{\mathcal{V}} \Lambda_{\text{s}}$:

$$\big((\mathbf{x} - \mathbf{s}) \bmod_{\mathcal{V}} \Lambda_{\text{s}}) + \mathbf{s} \bmod_{\mathcal{V}} \Lambda_{\text{s}} = \mathbf{x}, \tag{12}$$

where $\mathbf{x}$ is the intended codeword.

Because noise has been ignored, this explanation also ignores lattice inflation and random dither needed to achieve capacity on AWGN channels. However, it captures the essential idea of dirty paper coding using lattices.

The Voronoi region is often used as the fundamental region, that is, $\Omega = \mathcal{V}$. This choice tends to satisfy the average power constraint P with $\sum x_i^2 \leq P$, which is common in communications settings. As the dimension grows large, the Voronoi region of good lattices tends to resemble a sphere. In summary, the shaping lattice Λ_{s} and the corresponding fundamental region Ω should be choose to satisfy the power constraints of the problem.

B. WOM Encoding and Decoding

In conventional WOM codes, n cells can store one of q integer values each, that is each cell contains one value of $\{0, 1, \ldots, q-1\}$. The lattice-WOM approach generalizes this, allowing a memory cell to store an arbitrary value between 0 and V, that is $V = q - 1$. Then, n cells can store one point of an n-dimensional lattice.

To construct the WOM codebook, use an n-cube with side length V, $\mathbb{B}_V = [0, V)^n$. Then, the entire set of lattice points in this space is called the WOM codebook $\mathcal{C}_{\text{wom}}$:

$$\mathcal{C}_{\text{wom}} = \Lambda \cap \mathbb{B}_V. \tag{13}$$

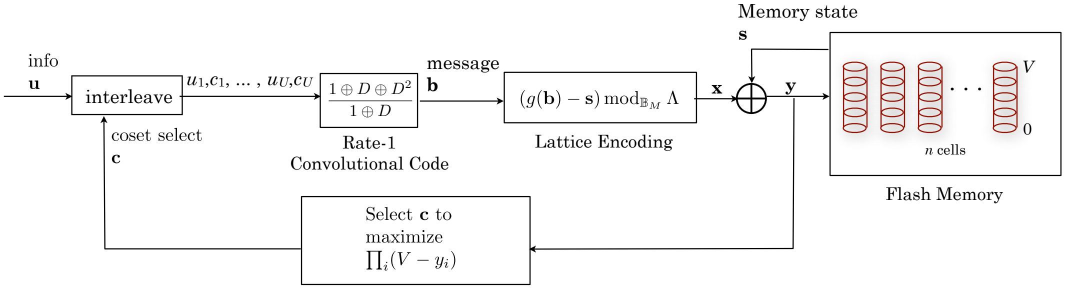
Memory state
s
info
u
interleave
$u_1, c_1, \ldots, u_U, c_U$
$\frac{1 \oplus D \oplus D^2}{1 \oplus D}$
message
b
$(g(\mathbf{b}) - \mathbf{s}) \bmod_{\mathbb{B}_M} \Lambda$
x
y
V
0
n cells
coset select
c
Rate-1
Convolutional Code
Lattice Encoding
Flash Memory
Select c to maximize $\prod_i (V - y_i)$

Fig. 1. Block diagram of lattice WOM encoding.

Conventional WOM codes are defined on $\mathcal{C}_{\text{wom}} = \{0, 1, \ldots, q-1\}^n$, and this is regarded as a special case of lattice WOM codes.

The second codebook $\mathcal{C}_{\text{local}}$ is analogous to the dirty paper codebook, using the cube $\mathbb{B}_M$, with $M \leq V$:

$$\mathcal{C}_{\text{local}} = \Lambda \cap \mathbb{B}_M. \tag{14}$$

A generalized WOM encoding function follows, which is slightly different from the standard definition. On write i, a message from $\mathcal{B} = \{0, 1, \ldots, 2^B - 1\}$ (a B-bit message) is written to the memory if the current state of the memory is $\mathbf{s} \in \mathcal{C}_{\text{wom}}$ by an encoding function E_i:

$$E_i : \mathcal{B}_i \times \mathcal{C}_{\text{wom}} \quad \rightarrow \quad \mathcal{C}_{\text{local}} \tag{15}$$

such that

$$E_i(\mathbf{b}, \mathbf{s}) \quad \succeq \quad \mathbf{0} \tag{16}$$

where $\succeq$ indicates $\geq$ in all positions. The value written to memory is $\mathbf{s} + E_i(\mathbf{b}, \mathbf{s})$. (With conventional definitions, for example [13], the encoder provides the value written to memory exactly. The current definition is clearly suited to the "known interference" channel model. The conventional decoding function will be identical.)

Now the encoder function E is cast as dirty-paper coding; the subscript i is dropped in the sequel. Use a shaping lattice $\Lambda_{\text{s}} = M\Lambda$, and a fundamental region $\mathbb{B}_M$. Then the encoder function is:

$$E(\mathbf{b}, \mathbf{s}) = \left(g(\mathbf{b}) - \mathbf{s}\right) \bmod_{\mathbb{B}_M} \Lambda_{\text{s}} \tag{17}$$

and then $\mathbf{y} = E(\mathbf{b}, \mathbf{s}) + \mathbf{s}$ is stored in memory; $g(\cdot)$ maps to a codeword using (9). Note that this point $\mathbf{y}$ should be an element of $\mathcal{C}_{\text{wom}}$, or if any coordinate exceeds V then a failure should be declared.

The decoder function is:

$$D(\mathbf{y}) \quad = \quad \mathbf{y} \bmod_{\mathbb{B}_M} \Lambda_{\text{s}} \tag{18}$$

so that:

$$\begin{aligned} D(\mathbf{y}) &= D\Big(\left(g(\mathbf{b}) - \mathbf{s}\right) \bmod_{\mathbb{B}_M} \Lambda_{\text{s}} + \mathbf{s}\Big) && (19) \\ &= \left(\left(g(\mathbf{b}) - \mathbf{s}\right) \bmod_{\mathbb{B}_M} \Lambda_{\text{s}} + \mathbf{s}\right) \bmod_{\mathbb{B}_M} \Lambda_{\text{s}} && (20) \\ &= g(\mathbf{b}) \bmod_{\mathbb{B}_M} \Lambda_{\text{s}} && (21) \\ &= g(\mathbf{b}), && (22) \end{aligned}$$

and by applying g^{-1}, the original message $\mathbf{b}$ may be found.

In this way, this lattice WOM code can be seen as a kind of dirty-paper code. The current state of the memory $\mathbf{s}$ plays the role of the known interference. In dirty-paper coding, the shaping region is a Voronoi region, to satisfy the average power constraint requirement. In the case of the WOM code, the shaping region is cubic with only non-negative values, to satisfy the restriction that cell values can only be increased.

IV. WOM Code Construction

It is possible to construct a WOM code directly using the above method. For example, let $M = \frac{V}{2}$. The number of points in $\mathcal{C}_{\text{wom}}$ that decode to the same information appears in $\mathcal{C}_{\text{wom}}$ 2^n times. There is a linearity relationship between these points. Unfortunately, the linearity leads to a difficulty: if one of these point is bad from the perspective of the average number of writes, then other points in this set may be bad as well. An alternative approach is needed.

To overcome this problem, the message $\mathbf{b}$ is formed of information bits $\mathbf{u}$, and non-information "coset select" bits $\mathbf{c}$. This reduces the amount of information that can be encoded, while increasing the number of cosets. The encoder will select the coset which has the codeword that maximizes the number of future writes. A mapping function Φ will be introduced that distributes these points in such a way that this problem is reduced, and the average number of writes is increased.

Concretely, U bits of information, $\mathcal{U} = \{0, 1, \ldots, 2^U - 1\}$ and C "coset select" bits $\mathcal{CS} = \{0, 1, \ldots, 2^C - 1\}$ are combined:

$$\Phi : \mathcal{U} \times \mathcal{CS} \rightarrow \mathcal{B}. \tag{23}$$

An information sequence appears 2^C times in the subcode encoding. Since $|\mathcal{C}_{\text{local}}| = M^n$, we have $U + C = n \log_2 M$. Cosets are formed only in the simple sense of a group over the binary vector,

$$u_1, u_2, \ldots, u_U, c_1, c_2, \ldots, c_U, \tag{24}$$

where the set of 2^U binary vectors with $c_i = 0$ forms a subgroup of all 2^{U+C} binary sequences. Cosets of this subgroup are selected with non-zero c_i. In particular, "coset select" does not refer to cosets of the lattice code.

Various mapping functions Φ were investigated, and the following was found to be most effective. Two bit sequences are $u_1, u_2, \ldots, u_U$ and $c_1, c_2, \ldots, c_C$. If $U = C$, these are interleaved as $c_1, u_1, c_2, u_2, \ldots, c_C, u_U$; if $U > C$, then some

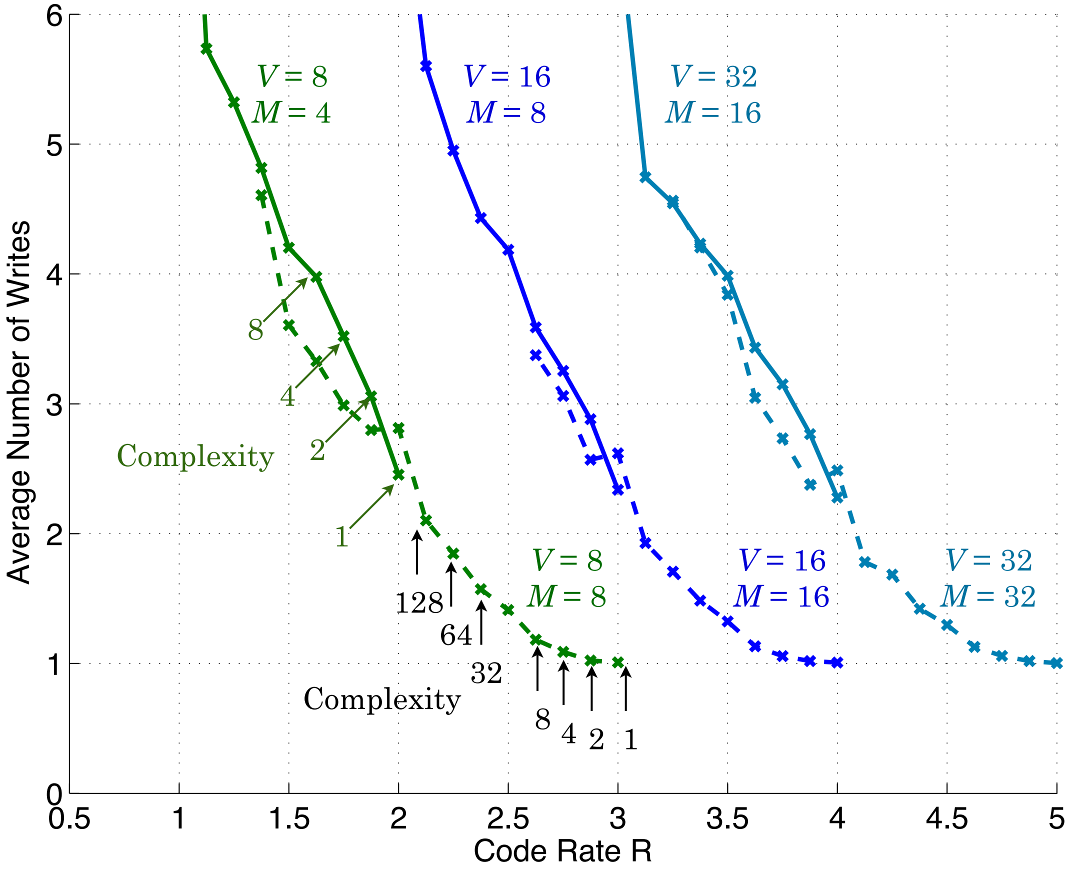

Fig. 2. Average number of writes for various constructions of the proposed lattice WOM code.

c_i bits are replaced by u_i bits. Then, a rate-1 recursive convolutional code is used. The input is the interleaved sequence of $U+C$ bits, and the output is the same number of bits $U+C$; the code is not terminated. To form the message $\mathbf{b}$, the first $\log_2 \frac{M}{g_{11}}$ bits from the output of the convolutional code are used to form b_1 (refer to Sec. II-B). The following $\log_2 \frac{M}{g_{22}}$ bits are used to form b_2, and so forth. The interleaving and recursive convolutional code non-uniformly distribute the information bits throughout the codebook.

For any $\mathbf{u}$ and $\mathbf{c}$, the message $\mathbf{b}$ can be found as above. From $\mathbf{b}$, the corresponding memory value is found as $\mathbf{y} = E(\mathbf{b}, \mathbf{s}) + \mathbf{s}$ using (17). Any $\mathbf{c} \in \mathcal{C}$ will encode the correct information. It is desired to find corresponding $\mathbf{y}$ that maximizes future writes.

Previously, it has been argued that choosing $\mathbf{y}$ that maximizes the remaining volume:

$$||V - \mathbf{y}|| \quad = \quad \prod_{i=1}^{n} (V - y_i) \qquad (25)$$

should be selected [4], [5], [6]. Specifically for each $\mathbf{c}_i \in \mathcal{CS}$, compute the corresponding $\mathbf{y}_i$ as above. The point which is written to memory is:

$$\mathbf{y} \quad = \quad \arg\max_{\mathbf{y}_i} \big| |V - \mathbf{y}_i| \big|,$$

where arg max is over $i = 1, 2, \ldots, 2^C - 1$. A block diagram of the encoder is given in Fig. 1.

The code rate R is:

$$R \quad = \quad \frac{U}{n} \text{ bits per dimension,} \qquad (26)$$

where $U + C = n \log_2 M$ and $M \leq V$. Choosing V large and C small will increase the rate while reducing the average number of rewrites.

Decoding is the straightforward reverse of the encoder. From $\mathbf{y}$, find the message $\mathbf{b}$ using (18). Because the rate-one convolutional code may be inverted, the function Φ is bijective, and Φ^{-1}:

$$\Phi^{-1} \; : \; \mathcal{B} \rightarrow \mathcal{U} \qquad (27)$$

is easily found.

V. Numerical Results

The E8 lattice is used to evaluate the average number of writes. This dimension $n = 8$ lattice has good packing properties and an efficient decoding algorithm [11]. A generator matrix suitable for encoding is:

$$G \quad = \quad \begin{bmatrix} 1/2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1/2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1/2 & -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1/2 & 0 & -1 & 1 & 0 & 0 & 0 & 0 \\ 1/2 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \\ 1/2 & 0 & 0 & 0 & -1 & 1 & 0 & 0 \\ 1/2 & 0 & 0 & 0 & 0 & -1 & 1 & 0 \\ 1/2 & 0 & 0 & 0 & 0 & 0 & -1 & 2 \end{bmatrix}.$$

The rate-1 convolutional code used for the function Φ is:

$$\frac{1 \oplus D \oplus D^2}{1 \oplus D}. \tag{28}$$

The memory is initially in the state $\mathbf{s} = 0$. In one experiment, a sequence of codewords is repeatedly written to memory, until a failure is detected. The average number of writes is simply the number of codewords written to memory, divided by the number of experiments.

Values of $V = 8, 16, 32$ were considered; these are analogous to $q = 9, 17$ and 33 (The WOM lattice construction does not write $V = 8$, etc. exactly, but $V = 7.5$, etc. The lattice may be scaled to obtain $V = 7$, but this will have a small effect the error performance, see [14].) Cases of $M = V$ and $M = V/2$ were considered, where the later has higher rate. For each one, the rate can be adjusted by adjusting the number of coset select bits C.

The numerical results are shown in Fig. 2. The natural tradeoff between code rate and the average number of writes is clearly seen. For the $V = 8$ case (and similarly for $V = 16, 32$), rates in the range of $R = 1.5$ to 2 can be achieved using either $M = 8$ and a large number of coset select bits, or $M = 4$ and a small number of coset select bits. Roughly speaking, these two coding strategies achieve the same average number of rates. But the complexity increases exponentially in the number of coset select bits.

Encoding complexity requires lattice encoding and well as the coset selection stage. Lattice encoding is generally polynomial complexity, but the coset selection has complexity exponential 2^C in the number of coset select bits C. Efficient encoding requires searching over all cosets. In Fig. 2, the complexity is label for various codes, for $V = 8$. Each time one coset select bit is added, the complexity doubles.

VI. Discussion

WOM coding can be viewed from the perspective of lattice-based communication in the presence of known-interference, also called dirty-paper coding. The similarity is that the state of the memory plays the role of known interference as in dirty-paper coding.

Using the lattice-strategy approach for dirty-paper coding [8], a specific code construction was proposed. But the power constraint particular to each problem imposes restrictions on the design of the code. The code is designed using a lattice fundamental region as a shaping region for the finite code. For dirty-paper codes, the average power constraint is satisfied by the Voronoi fundamental region. But for lattice WOM codes, the positive-only codeword requirement is satisfied by a cubic shaping region. If we take low-rate cases where $M \ll V$, then the result "known interference does not reduce capacity" may be applied to WOM codes, because it allows fairly ignoring the maximum writing level V, but this parameter choice is not practical. A specific lattice WOM code was constructed. Coset select bits were included in order to improve the average number of writes. The numerical results show a promising trend.

References

[1] R. L. Rivest and A. Shamir, "How to reuse a "write-once" memory," *Information and Control*, vol. 55, pp. 1–19, December 1982.

[2] B. M. Kurkoski, "Asymptotic rates for lattice-based WOM codes." Presented at Non-Volatile Memories Workshop 2012, March 2012. San Diego, USA.

[3] J. H. Conway and N. J. A. Sloane, *Sphere Packings, Lattices and Groups*. New York, NY, USA: Springer-Verlag, 3rd ed., 1999. ISBN 0-387-98585-9.

[4] B. M. Kurkoski, "Rewriting codes for flash memories based upon lattices, and an example using the E8 lattice," in *Proceedings IEEE Global Telecommunications Conference*, (Miami, USA), pp. 1923–1927, IEEE, December 2010.

[5] A. Bhatia, A. R. Iyengar, and P. H. Siegel, "Multilevel 2-cell t-write codes," in *Information Theory Workshop 2012*, (Lausanne, Switzerland), 2012.

[6] B. M. Kurkoski, "Lattice-based WOM codebooks that allow two writes," in *International Symposium on Information Theory and its Applications*, (Honolulu, Hawaii, USA), pp. 101–105, 2012.

[7] M. H. M. Costa, "Writing on dirty paper," *IEEE Transactions on Information Theory*, vol. 29, pp. 439–441, May 1983.

[8] U. Erez, S. Shamai, and R. Zamir, "Capacity and lattice strategies for canceling known interference," *IEEE Transactions on Information Theory*, vol. 51, pp. 3820–3833, November 2005.

[9] U. Erez and S. ten Brink, "A close-to-capacity dirty paper coding scheme," *IEEE Transactions on Information Theory*, vol. 51, pp. 3417–3432, October 2005.

[10] U. Erez and R. Zamir, "Achieving $\frac{1}{2}\log(1 + \mathrm{SNR})$ on the AWGN channel with lattice encoding and decoding," *IEEE Transactions on Information Theory*, vol. 50, pp. 2293–2314, October 2004.

[11] J. H. Conway and N. J. A. Sloane, "A fast encoding method for lattice codes and quantizers," *IEEE Transactions on Information Theory*, vol. 29, pp. 820–824, November 1983.

[12] N. Sommer, M. Feder, and O. Shalvi, "Low-density lattice codes," *IEEE Transactions on Information Theory*, vol. 54, pp. 1561–1585, April 2008.

[13] E. Yaakobi, S. Kayser, P. H. Siegel, A. Vardy, and J. K. Wolf, "Efficient two-write WOM-codes," in *Proceedings of the IEEE Information Theory Workshop*, pp. 1–5, September 2010.

[14] B. M. Kurkoski, "The E8 lattice and error correction in multi-level flash memory," in *Proceedings IEEE International Conference on Communications*, (Kyoto, Japan), pp. 1–5, IEEE, June 2011.