Efficient Codes for Writing Equal-bit Information in a WOM Twice

Zhongtao Li, Shuai Zhao, and A.J. HAN VINCK
University of Duisburg-Essen, Duisburg, Germany
Email: li.zhongtao@hotmail.com

Ru Jia
Beijing University of Posts and Telecommunications, Beijing, China

Abstract—In this paper, we concentrate on WOM coding for writing equal-bit information twice (WOM2E). We studied the definition and coding solution of WOM2E. We encoded information via a codeword table and found that WOM2E achieved a better WOM-rate. However, the codeword table for large WOM2E cells is difficult to determine. We also introduced the coding sum rate to define the WOM2E efficiency. The sum rate of WOM2E is the sum of the ratios of information bits to the number of cells each time data is written to memory. In order to obtain the maximum coding sum rate, we can utilize more cells. Furthermore, we discover that, as the number of cells in memory increases, the sum rate is perfectly asymptotic to the logarithm. We use Matlab to calculate the upper bound of the sum rate of WOM2E, and find that it has a value of 1.546.

Index Terms—WOM; WOM2E; Memory Coding

I. INTRODUCTION

In the paper “How to reuse a ‘Write-Once’ Memory” [1], Rivest and Shamir proposed the concept of Write-Once Memory (WOM). WOM consists of a number of binary “write-once” memory elements (or positions), which are termed “cells.” The initial state of each cell is 0. During the process of writing information, the content of each cell can be changed only once from state 0 to state 1, i.e., this process is irrevocable. State 0 denotes that the bit value in this cell is 0 and state 1 denotes that the bit value in this cell is 1. The bit value can only increase in WOM.

In practice, because WOM is irrevocable, it is a perfect tamper-proof storage media for preventing original information from being edited [2]. Furthermore, the cost per bit of WOM is very cheap. Thus, it is suitable for an abundance of industrial applications.

However, the “write-once” property might cause unnecessary costs in terms of revising a fraction of cells of WOM. Advances in coding techniques enable us to rewrite WOM several times [3]. Nevertheless, if there are cells with the value 0 in a used WOM, the user may write new information by changing those cells to 1. Consequently, its bit-capacity is much greater than the number of cells in memory.

If a WOM can be written twice, it is termed WOM2. This is the basis of multi-write WOM. Let $R_i$ denote the coding rate of the i-th writing operation, so that the WOM2 sum coding rate $R_{\text{sum}} = R_1 + R_2$. If the coding rates $R_1$ and $R_2$ are equal, this is known as writing WOM equal-bit information twice (WOM2E). In practice, this is the most important utilization. Hence, we focus on the WOM2E problem.

II. REUSING WOM

A. Multi-time and Two-time WOM

Once a cell’s value is 1, it cannot be changed back to 0. Thus, the cell is non-reusable. On the other hand, cells storing a value of 0 after the first writing can be rewritten with new information. Using this method, cells can be reused multiple times (see Figure 1). The WOM might be rewritten with new information until all the cells are in State 1.

If WOM is only reused once, it is known as WOM2. This is the basis of multi-write WOM. Let us denote the probability of a cell being in state 0 as $p$. Then, the probability of the cell changing from 0 to state 1, i.e., this process is irrevocable. State 0 denotes that the bit value in this cell is 0 and state 1 denotes that the bit value in this cell is 1. The bit value can only increase in WOM.

In practice, because WOM is irrevocable, it is a perfect tamper-proof storage media for preventing original information from being edited [2]. Furthermore, the cost per bit of WOM is very cheap. Thus, it is suitable for an abundance of industrial applications.

However, the “write-once” property might cause unnecessary costs in terms of revising a fraction of cells of WOM. Advances in coding techniques enable us to rewrite WOM several times [3]. Nevertheless, if there are cells with the value 0 in a used WOM, the user may write new information by changing those cells to 1. Consequently, its bit-capacity is much greater than the number of cells in memory.

If a WOM can be written twice, it is termed WOM2. This is the basis of multi-write WOM. Let $R_i$ denote the coding rate of the i-th writing operation, so that the WOM2 sum coding rate $R_{\text{sum}} = R_1 + R_2$.

In the best scenario of WOM2, every cell has changed from 0 to 1 after the two writes. Figure 3 illustrates this scenario, in which all cells in state 1 after the second writing.

B. Two-time Equal-bit Information WOM

In practice, WOM equal-bit information twice (WOM2E) has more advantageous. For example, if the
WOM2E is used to store information (including price, date produced, and so on) about one product, the length of this information is fixed for one specific type of product [6]. Thus, it needs to write information with the same number of bits.

![Figure 2. One binary source X writes two times](image)

Figure 2. One binary source X writes two times

![Figure 3. One binary source X writes two times and ensures full memory use](image)

Figure 3. One binary source X writes two times and ensures full memory use

By its information theory definition, entropy is a measure of the expected value of a random variable. It represents the amount of information before being received. This is also known as information entropy. If we encode the information to a binary source \( x \in \{0,1\} \), the binary entropy is shown in (1) [7]:

\[
    h(x) = -x \cdot \log_2^x - (1-x) \cdot \log_2^x
\]

In first writing, WOM stores \( h(p) \) bits. If every cell in memory changes from 0 to 1 after writing to memory twice, i.e., \( p' = 0 \) and \( 1 - p' = 1 \) (see Figure 3), WOM stores the maximum \( (1-p) \) bits in second writing. For WOM2E, \( h(p) = 1-p \) (see Figure 4).

The common point in Figure 4 is the solution of the equation \( h(p) = 1-p \). Using Matlab, we obtain a value of \( p = 0.227 \).

![Figure 4. The relationship between h(p) and (1-p)](image)

Figure 4. The relationship between \( h(p) \) and \( (1-p) \)

### III. EFFICIENT WOM2E CODE

#### A. A <4, 4>/3 WOM2E Code

The code table is very important to show how to record binary information \( t \) times in WOM. Let us take the first WOM-code construction as an example. This code [1] was designed for the storage of two bits twice using three cells (see Figure 5).

![Figure 5. The <4, 4>/3 WOM-code on the Boolean 3-cube](image)

Figure 5. The <4, 4>/3 WOM-code on the Boolean 3-cube

Let \( n \) denote the number of cells of WOM. Arrays \( V \) show the number of codewords. The \( i \)-th writing is also called the \( i \)-th generation. \( t \) denotes the total writing number. Hence, \( 1 \leq i \leq t \). Usually, the WOM2E is described by \( <V_1, V_2> / n \) (\( V_1 = V_2 \)), and it can also be described by \( <V >^2 / n \) [8].

For example, a \(<4, 4>/3 \) WOM-code represents that we can write \( 2^2 \) different kinds of binary code, which implies two bits of information, two times using three cells via the WOM coding. Table I lists the \(<4, 4>/3 \) WOM codewords.

<table>
<thead>
<tr>
<th>Data bit X</th>
<th>First write X'</th>
<th>Second write X''</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>000</td>
<td>111</td>
</tr>
<tr>
<td>01</td>
<td>001</td>
<td>110</td>
</tr>
<tr>
<td>10</td>
<td>010</td>
<td>101</td>
</tr>
<tr>
<td>11</td>
<td>100</td>
<td>011</td>
</tr>
</tbody>
</table>

Table I. A <4, 4>/3 WOM CODEWORD TABLE

Take the third column as an example. After the first writing, this WOM contains the information 001. The first blank presents 000, which is not equal to the information already saved, so this blank is filled by the X'' codeword of 000, which is 111. The second blank present, 001, is equal to the information already saved, so this blank is filled by 001 itself, and so on.

#### B. A <5, 5>/4 WOM2E Code

Table IV shows a <5, 5>/4 WOM binary information codeword table. Because each possibility has exactly one X' on the first write and one X'' the second time, this kind of encoding is also called “one-to-one” coding.

The initial value of a cell is 0000 for a four-cell WOM. On the first writing, the cell can become one of the five
different codewords shown in Table V. The first writing coding rate \( R_1 = \frac{\log_2^5}{4} = 0.58 \) bit/cell.

The second writing is shown in Table VI. The second writing coding rate \( R_2 = \frac{\log_2^5}{4} = 0.58 \) bit/cell. The sum efficiency is \( R_1 + R_2 = 1.16 \) bits/cell. This is greater than the efficiency of WOM if memory is only used once: coding rate \( R = 1 \) bit/cell.

However, the \( <5, 5>/4 \) coding rate is lower than \( <4,4>/3 \) solution (in section III). A \( R = \frac{2 \cdot \log_2^{5}}{3} = 1.333 \) bits/cell. The second writing coding rate \( R_2 = \frac{\log_2^5}{4} = 0.6 \) bits/cell. The sum coding rate is \( R_1 + R_2 = 1.2 \) bits/cell.

### TABLE III. SECOND WRITE OF \( <4, 4>/3 \) WOM

<table>
<thead>
<tr>
<th>First writing codeword</th>
<th>000</th>
<th>001</th>
<th>010</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>000</td>
<td>111</td>
<td>111</td>
<td>111</td>
</tr>
<tr>
<td>1</td>
<td>110</td>
<td>001</td>
<td>110</td>
<td>110</td>
</tr>
<tr>
<td>2</td>
<td>101</td>
<td>010</td>
<td>101</td>
<td>101</td>
</tr>
<tr>
<td>3</td>
<td>011</td>
<td>011</td>
<td>011</td>
<td>100</td>
</tr>
</tbody>
</table>

### TABLE IV. A ONE TO ONE \( <5, 5>/4 \) WOM CODER TABLE

<table>
<thead>
<tr>
<th>Data bit X</th>
<th>First write X'</th>
<th>Second write X''</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>1111</td>
<td></td>
</tr>
<tr>
<td>0001</td>
<td>1110</td>
<td></td>
</tr>
<tr>
<td>0010</td>
<td>1101</td>
<td></td>
</tr>
<tr>
<td>0100</td>
<td>1011</td>
<td></td>
</tr>
<tr>
<td>1000</td>
<td>0111</td>
<td></td>
</tr>
</tbody>
</table>

### TABLE V. FIRST WRITE OF \( <5, 5>/4 \) WOM

<table>
<thead>
<tr>
<th>Data bit X1</th>
<th>Data bit X2</th>
<th>Data bit X3</th>
<th>Data bit X4</th>
<th>Data bit X5</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>0001</td>
<td>0100</td>
<td>1000</td>
<td></td>
</tr>
</tbody>
</table>

### TABLE VI. SECOND WRITE OF \( <5, 5>/4 \) WOM

<table>
<thead>
<tr>
<th>First writing codeword</th>
<th>0000</th>
<th>0001</th>
<th>0100</th>
<th>1000</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1111</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1110</td>
<td>111</td>
<td>111</td>
<td>111</td>
</tr>
<tr>
<td>2</td>
<td>1110</td>
<td>010</td>
<td>110</td>
<td>110</td>
</tr>
<tr>
<td>3</td>
<td>1110</td>
<td>011</td>
<td>011</td>
<td>100</td>
</tr>
</tbody>
</table>

### TABLE VII. A ONE TO ONE \( <8, 8>/5 \) WOM CODER TABLE

<table>
<thead>
<tr>
<th>Data bit X</th>
<th>First write X'</th>
<th>Second write X''</th>
</tr>
</thead>
<tbody>
<tr>
<td>00000</td>
<td>11111111</td>
<td></td>
</tr>
<tr>
<td>00001</td>
<td>11111011</td>
<td></td>
</tr>
<tr>
<td>00100</td>
<td>11111111</td>
<td></td>
</tr>
<tr>
<td>01000</td>
<td>11111011</td>
<td></td>
</tr>
<tr>
<td>10000</td>
<td>11111111</td>
<td></td>
</tr>
</tbody>
</table>

With the new codeword table, we can find a suitable way to write a five-cell WOM twice with 3-bit information each time (see Table XI). The second coding rate \( R_2 = \frac{\log_2^{5}}{3} = 0.6 \) bits/cell. The sum coding rate is \( R_1 + R_2 = 1.2 \) bits/cell.

### IV. CODING EFFICIENCY

#### A. WOM2E Rate

The efficiency of WOM coding can be represented by the code rate, which is also an i-th generation array \( R_i \). It can be calculated as the number of information bits, which can also be calculated as \( \log_2^{v_i} \), divided by the number of cells (see (2)) [9]:

\[
R_i = \frac{\log_2^5}{n}
\]  

After the i-th writing, the efficiency of WOM is equal to the sum of the individual rates for each writing (see (3)):

\[
R_{\text{sum}} = \sum_{i=0}^{n} R_i = \sum_{i=0}^{n} \frac{\log_2^5}{n}
\]

In “On the capacity of generalized write-once memory with state transitions described by an arbitrary directed cyclic graph” [10], Fu and Vinck proved that the capacity region of a binary t-write WOM is:

\[
R_r, R_{r+1}, \ldots, R_{t-1} \leq h(p_1), R_r \leq (1 - p_1)h(p_2), \ldots, \frac{1}{2} - \frac{1}{2} \leq 0 \leq p_1, \ldots, p_{t-1} \leq \frac{1}{2}\]

C. A \( <8, 8>/5 \) WOM2E Code

The five-cell WOM2E is based on the \( <2, 2>/2 \) and \( <4, 4>/3 \) WOM codes already discussed. WOM2E can be made up of two parts: one part containing two cells and the other containing three. The two cells have two possibilities, and the three cells have four possibilities on each write. Thus, all five cells can have \( 2 \cdot 4 = 8 \) possibilities on each write.

In Table VII, the bold numbers indicate how the two-cell WOM2E encodes, and the other numbers show how the three-cell WOM2E encodes.

The first writing is shown in Table VIII, and the first coding rate \( R_1 = \frac{\log_2^5}{5} = 0.6 \) bits/cell. The second writing in Table IX, the italic codewords denote those where WOM2E in state 1 is changed to 0 again. Obviously, these codewords are impossible.

However, with five-cell WOM2E, there are actually \( 2^2 = 32 \) different possibilities. The previous codeword table only uses 16 codewords. Thus, the codeword table can be changed to that shown in Table X.

For any binary information \( X, X' \) is the codeword used in the first writing and \( X', X'', X''' \) and \( X'''' \) are the codewords used in the second writing. Because there is only one codeword \( X' \) in the first writing for each \( X \), but more than one codeword in the second writing, this kind of encoding is called “one-to-many.”

\[
R_i \leq h(p_1), R_r \leq (1 - p_1)h(p_2), \ldots, R_{t-1} \leq (1 - p_{t-1})\frac{1}{2}
\]
If one cell changes from 0 to 1, the number of possibilities will be \( 2^n - b \). If we use combinatory to define this possibility, we can get
\[
\binom{n}{b} = \frac{n^b}{b!}.
\]

In n-cell WOM2E, once one cell is changed from 0 to 1, it cannot be used to write new information. It can be seen as a blocked cell. Let \( b \) denote the number of blocked cells after the first write, then the number of cells available for use in the second write is \( n-b \). [13]

They also proved that the maximum achievable rate for a binary t-write WOM is (see (4)) [11]:
\[
R_{\text{max}} = \log_2(1+t)
\tag{4}
\]

The main goals of this paper are to develop codes to write information to WOM twice, and determine the overall efficiency. Thus, \( t = 2 \) in this case.

Using (4), the maximum rate of WOM2 is:
\[
R_{\text{max}} = \log_2(1+t) = \log_2(1+2) = 1.585.
\]

If the WOM writes information of the same number of bits, the rate of each writing is also the same, i.e., \( R_1 = R_2 \). Hence, only two situations remain [12]:
\[
\begin{align*}
\{(R_1, R_2) | R_1 = 1-p = R_2, h(p), \frac{1}{2} \leq p \leq 1 \} \\
\{(R_1, R_2) | R_1 + R_2 = \log_2 \frac{2}{3} = \log_2 \frac{1}{1.5} \leq R_1 \leq \log_2 \frac{2}{3}
\end{align*}
\]

Using Matlab, we obtain \( p \approx 0.227 \) and \( R_1 = R_2 \approx 0.773 \) (see Figure 6).

Furthermore, for WOM2E, \( R_{\text{max}} = 2R \). The maximum WOM2E rate \( R_{\text{max}} = 1.546 \) is smaller than \( R_{\text{max}} \approx 1.585 \) (the maximum rate of WOM2).

### B. Rate region

In n-cell WOM2E, once one cell is changed from 0 to 1, it cannot be used to write new information. It can be seen as a blocked cell. Let \( b \) denote the number of blocked cells after the first write, then the number of cells available for use in the second write is \( n-b \). [13]

![Figure 6. The rate region of WOM2E](image)
will be $C = \sum_{i=0}^{b} C^i_n$ [14]. Because there are $n-b$ cells left in the second writing, the maximum number of possible binary codewords is $2^{n-b}$.

If $b$ is equal to $C$, it means the cell is fully used after both writings. As for the $<4, 4>/3$ code, there are a total of $2^3 = 8$ possibilities with three-cell WOM, half of which are used each time. The value of $b$ is calculated as 1, which can be verified in Table I. If $2^{n-b}$ is smaller than $C$, then for the second write, there are fewer possibilities than the first time. Conversely, if $2^{n-b}$ is bigger than $C$, the second write has more possibilities for writing information.

The same possibilities are required in WOM2E, so we choose a small number between $2^{n-b}$ and $C$ as the fixed possibility for both the first and second writing.

In some cases, however, blocking as many cells as possible the first time is not always the best solution. Take four-cell WOM as an example. If the number of blocked cells is two, there are $2^{4-2} = 4$ possibilities the second time. We can get $C = C_0^4 + C_1^4 + C_2^4 = 11$ at the first time. Thus, we can only use the smaller number 4 for each writing.

However, if we change only one cell to 1, there are $2^{4-1} = 8$ possibilities the second time. The number of possibilities used the first time is $C = C_0^4 + C_4^4 = 5$. This is an improvement over the $<5, 5>/4$ WOM code, and has been proved in section III.B.

Using Matlab, we can find the WOM2E rate region for $n \leq 100$ (see Figure 7).

![Figure 7. WOM2E rate for n≤100](image)

V. Conclusion

In this paper, we studied the definition and coding solution of WOM2E. Furthermore, we concentrated on utilizing WOM2E. We encoded information via a codeword table, and found that WOM2E achieved a better WOM-rate. However, the codeword table for large WOM2E cells is difficult to determine.

We also introduced the sum rate to define the WOM2E efficiency. As the number of cells in memory increases, the sum rate is perfectly asymptotic to the logarithm. We showed the upper bound of the sum rate of binary WOM to be 1.585, and proved that the upper bound of the sum rate of binary WOM2E is 1.546. Thus, for each write, the maximum rate is 0.773.

References


Zhongtao Li received his B.E. degree in school of computer science and technology, China University of Mining and Technology, in 2005. He received the M.E. degree in computer science and communications engineering from University of Duisburg-Essen, Duisburg, Germany, in 2009. He is currently a Ph.D. candidate in distributed system department, University of Duisburg-Essen, Duisburg, Germany. His current research interests include channel coding, computer network communication, distributed system and Peer-to-Peer system.

Shuai Zhao received her B.E. degree in school of computer science and technology, Shandong University of Finance, in 2005. She received the M.E. degree in computer science and communications engineering from University of Duisburg-Essen, Duisburg, Germany, in 2013. Her interest is in Information and Communication theory, digital transmission and coding techniques.

Ru Jia is currently a senior student and she will receive her Bachelor degree in Computer Science from Beijing University of Posts and Telecommunications in 2013. Her current research interests include information security, digital watermarking, digital image processing.