LA project

Shiv Sharma

November 2020

1 Construction of PRIO codes using coset coding with hamming code

1.1 Introduction

Flash memory is a form of non volatile memory that is prevalent in today's day and age for storing information. The flash devices ranges from Solid state drive to memory cards.

A flash memory consists of an array of cells, in which information bits are stored in the form of the amount of charge. In multilevel flash memory, each cell can represent one of more than two levels, and these levels are distinguished by multiple read thresholds. The triple-level cell (TLC) can represent one of eight levels, that is, it stores three bits. Each level corresponds to a three-bit sequence as shown in Table, and each bit represents one logical page. Then, a group of such cells stores three logical pages, referred to as pages 1–3.

Level	Page 1	Page 2	Page 3
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

To read pages 1–3, the required numbers of read thresholds are 1, 2, and 4, respectively. Therefore, the average number of read thresholds is 2.33. Multiple read thresholds are required in order to read a single logical page in multilevel flash memory. However, use of a large number of read thresholds degrades the read performance of the flash memory device. In order to solve this problem, the random input/output (RIO) code is used in which one logical page can be read using a single read threshold.

In the RIO code, the data of all logical pages are stored simultaneously; thus, all the data are known in advance. Parallel RIO (P-RIO) code are used to increase the efficiency by performing the encoding of each page in parallel.

P-RIO codes are constructed using coset coding. Coset coding is a technique that constructs RIO codes using linear binary codes. Hamming codes are used as the linear codes, to leverage the information on the data of all logical pages to construct the P-RIO codes. The constructed codes store more pages than RIO codes constructed via coset coding with Hamming codes.

1.2 Coset coding

In this paper, a linear binary code of length n and dimension k is referred to as an (n, k) code. coset coding is used to construct an $[n, n-k, t]$ WOM code using (n, k) code C, where t depends on C[4]. Let H be the parity check matrix of C. For all $(d, c) \in \{0,1\}^{n-k} \times Im\left(\mathcal{E}^{i-1}\right)$ with $i \in \mathcal{I}_t$, the encoding map $\mathcal{E}$ is as follows.

$$\mathcal{E}(d, c) = c + x$$

where

$$x \in \left\{v \mid v \in \{0,1\}^n, vH^T = d - cH^T, I(c) \cap I(v) = \emptyset\right\}$$

Further, for all $c \in Im\left(\mathcal{E}^i\right)$ with $i \in \mathcal{I}_t$, the decoding map $\mathcal{D}$ is

$$\mathcal{D}(c) = cH^T$$

The following theorem has been proven in [4]. Theorem 1: When the linear code is the (7,4) Hamming code, coset coding can be used to construct the [7,3,3] WOM code. Let r be an integer greater than 3. Then a $\left[2^r - 1, r, 2^{r-2} + 2\right]$ WOM code can be constructed via coset coding with a $(2^r - 1, 2^r - r - 1)$ Hamming code.

1.3 RIO Code

In this paper, it is assumed that each cell of a flash memory represents one of q levels $(0, 1, \ldots, q-1)$. These levels are distinguished by $(q-1)$ different read thresholds. For each $i \in \mathcal{I}_{q-1}$, we denote the threshold between levels $(i-1)$ and i by i. For the state of n cells $c = (c_1, \ldots, c_n) \in \{0, \ldots, q-1\}^n$, the operation for read threshold i is denoted by $RT_i(c)$. We define $RT_i(c) = (r_1, \ldots, r_n) \in \{0,1\}^n$ where for each $j \in \mathcal{I}_n, r_j = 1$ if $c_j \geq i$; otherwise , $r_j = 0$. For any $c_1, \ldots, c_{q-1} \in \{0,1\}^n$ such that $c_1 \leq c_2 \leq \ldots \leq c_{q-1}$, the following property holds: For each $i \in \mathcal{I}_{q-1}$,

$$RT_{q-i}\left(\sum_{j=1}^{q-1} c_j\right) = c_i$$

RIO code is a coding scheme that stores t pages in a $(t+1)-$ level flash memory, such that page i can be read using read threshold $(t+1-i)$ for each $i \in \mathcal{I}_t$[1]. From (6), pages $1, \ldots, t$ are encoded into $c_1, \ldots, c_t$, respectively, such that $c_1 \leq c_2 \leq \cdots \leq c_t$, and the cell state is set to $\sum_{i=1}^{t} c_i$ where c_i is a binary vector for each $i \in \mathcal{I}_t$.

Definition 2: An $[n, l, t]$ RIO code is a coding scheme that enables storage of t pages of l data bits in $n(t+1)$ -level cells, such that each page can be read using a single read threshold. For each $i \in \mathcal{I}_t$, the encoding map of page $i\mathcal{E}_i$ is defined by

$$\mathcal{E}_i : \{0,1\}^l \times Im\left(\mathcal{E}_{i-1}\right) \to \{0,1\}^n$$

where $Im\left(\mathcal{E}_0\right) = \{(0, \ldots, 0)\}$. For all $(d, c) \in \{0,1\}^l \times\ Im\left(\mathcal{E}_{i-1}\right), c \leq \mathcal{E}_i(d, c)$ is satisfied. The cell state is the sum of the $\mathcal{E}_i$ results for all $i \in \mathcal{I}_t$. Further, for each $i \in \mathcal{I}_t$, the decoding map of page $i\mathcal{D}_i$ is defined by

$$\mathcal{D}_i : Im\left(\mathcal{E}_i\right) \to \{0,1\}^l$$

such that $\mathcal{D}_i\left(\mathcal{E}_i(d, c)\right) = d$ for all $(d, c) \in \{0,1\}^l \times Im\left(\mathcal{E}_{i-1}\right)$ If the cell state is $c \in \{0, \ldots, t\}^n$, the argument of $\mathcal{D}_i$ is $RT_{t+1-i}(c)$ for each $i \in \mathcal{I}_t$ From Definitions 1 and 2, it is apparent that the construction of an $[n, l, t]$ RIO code is equivalent to that of an $[n, l, t]$ WOM *code*[1] Example 2: The [3,2,2] RIO code based on the [3,2,2] WOM code is shown in Table III. As an example, let the data of pages 1 and 2 be 10 and 01, respectively. Then, from Table II, $\mathcal{E}_1(10, 000) = 010$ and $\mathcal{E}_2(01, 010) = 011$. Therefore, the cell state is 021 .

1.4 PRIO code

In the RIO code, each page encoding depends on the page data only. In contrast, for P-RIO code, information on the data of all pages is leveraged during encoding of each page *Definition*: An $[n, l, t]$ P-RIO code is an $[n, l, t]$ RIO code for which all pages are encoded in parallel. The encoding map $\mathcal{E}$ is defined by

$$\mathcal{E} : \prod_{i=1}^{t}\{0,1\}^{l} \rightarrow \prod_{i=1}^{t}\{0,1\}^{n}$$

For all $(d_1, \ldots, d_1) \in \Pi_{i-1}^{t}\{0,1\}', c_1 \leq c_2 \leq \cdots \leq c_t$ is satisfied, where $(c_1, \ldots, c_t) = \varepsilon\left(d_1, \ldots, d_t\right)$. The cell state is the sum of the components of the $\mathcal{E}$ result. For each $i \in I_t$ the decoding map D_i of page i is defined by

$$\mathcal{D}_i : \{0,1\}^{n} \rightarrow \{0,1\}^{l}$$

such that $\mathcal{D}_i\left(c_i\right) = d_i$ for all $(d_1, \ldots, d_c) \in \prod_{i-1}^{t}\{0,1\}^{t}$ where $(c_1, \ldots, c_t) = \varepsilon\left(d_1, \ldots, d_t\right)$. If the cell state is $c \in \{0, \ldots, t\}^{n}$, the argument of $\mathcal{D}_i$ is $RT_{i+1-i}(c)$ for each $i \in \mathcal{I}_{1-}$ An algorithm to construct P-RIO code has been proposed in a previous study [3] , and run to yicld P-RIO codes that store two pages for moderate code lengths. These codes have parameters for which RIO codes do not exist [3].

Coset coding is used to construct P-RIO codes. When Hamming codes are used as the linear codes $[7, 3, 3]$ RIO code and [15,4,6] RIO code can be constructed. Then, we leverage the information on the data of all pages to construct P-RIO codes that store more pages than these RIO codes.