Coding and Bounds for Partially Defective Memory Cells

Haider Al Kim[1,2], Sven Puchinger[3], Antonia Wachter-Zeh[1]
[1]Institute for Communications Engineering, Technical University of Munich (TUM), Germany
[2]Electronic and Communications Engineering, University of Kufa (UoK), Iraq
[3]Department of Applied Mathematics and Computer Science, Technical University of Denmark (DTU), Denmark
Email: haider.alkim@tum.de, svepu@dtu.dk, antonia.wachter-zeh@tum.de

***Abstract*—This paper considers coding for *partially stuck* memory cells. Such memory cells can only store partial information as some of their levels cannot be used due to, e.g., wearout. First, we present a code construction for masking such partially stuck cells while additionally correcting errors. Second, we derive a sphere-packing and a Gilbert-Varshamov bound for codes that can mask a certain number of partially stuck cells and correct errors additionally. A numerical comparison between the new bounds and our constructions for any u partially stuck cells $\leq n$ memory cells shows that our construction matches the Gilbert–Varshamov-like bound for several code parameters.**

***Index Terms*—flash memories, phase change memories, non-volatile memories, defective memory, (partially) stuck cells, sphere packing bound, Gilbert-Varshamov bound**

I. Introduction

The demand for reliable memory solutions and in particular for non-volatile memories such as *phase-change memories* (PCMs) for different applications is steadily increasing. These multi-level devices provide permanent storage and a rapidly extendable capacity. The key characteristic of PCM cells is that they can switch between two main states: an amorphous state and a crystalline state. PCM cells may become *defect* (also called *stuck*) if they fail in switching their states. Therefore, cells can only hold a single phase [3], [4]. In multi-level PCM cells, failure may occur at a position in either of extreme states or in the partially programmable states of crystalline. In [1], a cell that can only hold levels at least a certain reference level $s > 0$ is called *partially stuck*. For multi-level PCMs, the case $s = 1$ is particularly important since this means that a cell can reach all crystalline sub-states, but cannot reach the amorphous state. Similar to PCMs, (partial) defects can occur in flash memory cells. In order to write information in a new write, either all current levels are only increased or a whole block has to be erased. Erasing the whole block reduces the lifespan of the flash memory device. The suggested mechanism to deal with these defective memory cells is called *masking*. Masking determines a codeword that matches the stuck level of the stuck memory. Therefore, it can be stored properly in the defective memory.

II. Related Work

In [2], code constructions for masking stuck memory cells were proposed. In addition to masking the stuck cells, it is possible to correct errors that occur during the storing and reading processes. A generator matrix of a specific form was constructed for this purpose. In [1], improvements on the redundancy necessary for masking *partially* stuck cells are achieved, and lower and upper bounds are derived. However, the paper does not consider error correction in addition to masking. In this paper, we combine the method of the reduced redundancy necessary for masking from [1] with the capability to correct additional errors from [2].

III. Preliminaries

A. Notations

For a prime q, let $\mathbb{F}_q$ denote the finite field of order q. Denote $[f] = \{0, 1, \dots, f-1\}$ for $f \in \mathbb{Z}_{>0}$. Let n be the total number of cells, u be the number of partially stuck cells, and t be the number of random errors. Let s_{ϕ_i} denote the partially stuck level at position i, where $i \in [u]$, and $\boldsymbol{\phi} = \{\phi_0, \phi_1, \cdots, \phi_{u-1}\} \subseteq [n]$ denotes the positions of the the partially stuck cells. For our construction, let k_1 be the number of information symbols, l be the number of symbols required for masking, and r be the required redundancy for error correction. Let $\zeta(n, d_0^\perp, q) := n - \lfloor \log_q(\sum_{j=0}^{d_0^\perp-2} \binom{n-1}{j}(q-1)^j) \rfloor - 1$ and $\xi(n, h_q(\delta_0^\perp), \varepsilon) := (1 - h_q(\delta_0^\perp) - \varepsilon)n$, where $\delta_0^\perp = \frac{d_0^\perp}{n}$ is the relative distance and its value $0 \leq \delta_0^\perp < 1 - \frac{1}{q}$ while $d_0^\perp$ is the dual minimum distance of the code $\mathcal{C}_0^\perp$, $0 < \varepsilon < 1 - h_q(\delta_0^\perp)$, and $h_q(\delta_0^\perp)$ is the q-ary entropy function. We use $\zeta(n, d_0^\perp, q)$ and $\xi(n, h_q(\delta_0^\perp), \varepsilon)$ in Theorem 3 to prove the existence of an error correcting and masking code.

B. Definitions

1) Partially Defective Cells: A cell is called partially defect (*partially stuck at level* s), where $s \in [q]$, if it can only store values which are at least s. If a cell is partially stuck at 0, it is a non-defect cell which can store any of the q levels.

2) (u, t)-PSMC: An $(n, \mathcal{M}_{u,t})_q$ *(u, t)-partially-stuck-at-masking code* $\mathcal{C}$ is a coding scheme consisting of an encoder $\mathcal{E}$ and a decoder $\mathcal{D}$. The inputs of the encoder $\mathcal{E}$ are: the set of locations of u partially stuck cells $\boldsymbol{\phi}$, the partially stuck levels $s_{\phi_0}, s_{\phi_2}, \dots, s_{\phi_{u-1}} \in [q]$, and a message $\boldsymbol{m} \in \mathcal{M}_{u,t}$, where $\mathcal{M}_{u,t}$ is a message space of cardinality $|\mathcal{M}_{u,t}|$. It outputs a vector $\boldsymbol{c} \in \mathbb{F}_q^n$ which fulfills $c_{\phi_i} \geq s_{\phi_i}$ for all $i \in [u]$. The decoder is a mapping that takes $\boldsymbol{c}+\boldsymbol{e} \in \mathbb{F}_q^n$ as input and returns the correct message $\boldsymbol{m}$ for all error vectors $\boldsymbol{e}$ of Hamming weight at most t.

IV. Codes for (Partially) Defective Memories

Construction 1. *Let n, u, t, k, k_1, r, l be positive integers with $u, t \leq n$, $k_1 = n - l - r$ and $k = k_1 + l$. Suppose that there are matrices $\boldsymbol{P} \in \mathbb{F}_q^{k_1 \times r}$ and $\boldsymbol{H}_0 \in \mathbb{F}_q^{(n-k_1-r)\times n}$ is a systematic parity-check matrix of a code $\mathcal{C}_0$ with parameters $[n, k_1 + r, d_0 \geq u - q + 3]_q$, such that*

$$\boldsymbol{G} = \begin{bmatrix} \boldsymbol{G}_1 \\ \boldsymbol{H}_0 \end{bmatrix} = \begin{bmatrix} \boldsymbol{0}_{k_1 \times l} & \boldsymbol{I}_{k_1} & \boldsymbol{P}_{k_1 \times r} \\ & \boldsymbol{H}_0 & \end{bmatrix}$$

is a generator matrix of a $[n, k = l + k_1, d \geq 2t + 1]_q$ code $\mathcal{C}$.

Based on these definitions, we define a coding scheme in Algorithm 1 and Algorithm 2.

Theorem 1. *The coding scheme in Construction 1 is a (u, t)-PSMC with $s_{\phi_i} = 1$ of length n and cardinality $\mathcal{M}_{u,t} = q^{k_1}$.*

The proof is similar to the proof of [1, Theorem 8] for the masking only that uses $\boldsymbol{H}_0$ whose $r = 0$. In contrast, since

Algorithm 1: Encoding

Input: Message: $m = (m_0, m_1, \ldots, m_{k-1}) \in \mathbb{F}_q^{k_1}$ and $\boldsymbol{\phi}$

Find $\boldsymbol{z} = (z_0, z_1, \cdots z_{l-1}) \in \mathbb{F}_q$ as explained in the proof of [1, Theorem 7]

Compute $\boldsymbol{c} = ((\boldsymbol{m} \cdot \boldsymbol{G}_1) + (\boldsymbol{z} \cdot \boldsymbol{H}_0)) \mod q$

Output: Codeword $\boldsymbol{c} \in \mathbb{F}_q^n$

Algorithm 2: Decoding

Input: $\boldsymbol{y} = \boldsymbol{c} + \boldsymbol{e} \in \mathbb{F}_q^n$

$\hat{\boldsymbol{c}} \leftarrow$ decode $\boldsymbol{y}$ in the code $\mathcal{C}$

Unmasking Process:

- $\hat{\boldsymbol{z}} \leftarrow (\hat{c}_0, \hat{c}_1, \cdots \hat{c}_{l-1})$
- $\hat{\boldsymbol{w}} = (\hat{w}_0, \hat{w}_1, \cdots, \hat{w}_{n-1}) \leftarrow (\hat{\boldsymbol{c}} - \hat{\boldsymbol{z}} \cdot \boldsymbol{H}_0) \mod q$
- $\hat{\boldsymbol{m}} \leftarrow (\hat{w}_l, \hat{w}_{l+1} \ldots, \hat{w}_{n-r-1})$

Output: Message vector $\hat{\boldsymbol{m}} \in \mathbb{F}_q^{k_1}$

Construction 1 is capable to correct errors, $r \neq 0$.

In summary, Algorithm 1 encodes the message $\boldsymbol{m}$ into a codeword of the code $\mathcal{C}$ such that all u partially-stuck positions are ≥ 1 (i.e., masked). Algorithm 2 then corrects up to t errors using the code $\mathcal{C}$ and recovers the message vectors. This means that the coding scheme is a (u, t)-PSMC.

V. Bounds on the Cardinality and Minimum Distance

A. Sphere-Packing Bound on PSMCs

Theorem 2. *Any (u,t)-PSMC over $\mathbb{F}_q$ of cardinality $\mathcal{M}_{u,t}$ satisfies:*

- *for non-overlapping, i.e., the errors are only allowed to happen at positions $\boldsymbol{\Psi} = \{\Psi_0, \Psi_1, \ldots, \Psi_{n-u-1}\} = [n] \setminus \boldsymbol{\phi}$, then we have:*

$$\mathcal{M}_{u,t} \cdot \sum_{j=0}^{t} \binom{n-u}{j} (q-1)^j \leq q^{n-u} \prod_{i=0}^{u-1} (q - s_{\phi_i}),$$

- *for overlapping, i.e., the errors are allowed to happen in any cell, i.e., $\boldsymbol{\Psi} \subseteq [n]$ under the assumption that $s_{\phi_i} < q-1$, for $i = 0, \ldots, u-1$, then we have:*

$$\mathcal{M}_{u,t} \cdot \sum_{j=0}^{t} \sum_{j_1=0}^{j} \tbinom{n-u}{j_1} (q-1)^{j_1} \tbinom{u}{j-j_1} \prod_{i \in \mathcal{J}} (q-1-s_{\phi_i})$$
$$\leq q^{n-u} \prod_{i=0}^{u-1} (q - s_{\phi_i}),$$

where $\mathcal{J}$ denotes the set of cardinality $j - j_1$ of stuck cells that is affected by errors.

B. Gilbert–Varshamov Bound on PSMCs

Theorem 3 (Gilbert-Varshamov-like bound)**.** *Suppose there are positive integers u, t, n, $d_0^{\perp} \geq d \geq 2t+1$, $d_0 \geq u-q+3$, $l \leq k-1$, and $q \geq 2$ that satisfy:*

$$q^{l+1} \cdot \sum_{i=0}^{2t} \binom{n}{i} (q-1)^i \leq q^n,$$

$$q^{n-l+1} \cdot \sum_{j=0}^{u-q+2} \binom{n}{j} (q-1)^j \leq q^n,$$

$$q^k \cdot \sum_{i=0}^{2t} \binom{n}{i} (q-1)^i \leq q^n.$$

Then, there is an $[n, k, \geq 2t+1]_q$ code $\mathcal{C}$ that has a subcode $\mathcal{C}_0^{\perp}$ $[n, l]_q$ whose dual distance is $d_0 \geq u-q+3$.

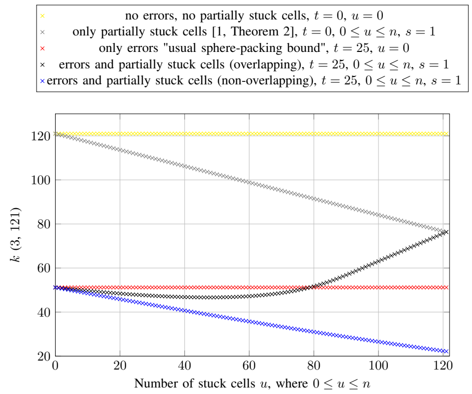

Figure 1. **Sphere-packing bounds**: Comparison for $k(q,n)$ information symbols for the classical sphere-packing bound ("only errors") and our sphere-packing-like bounds ("errors and stuck cells") for non-overlapping and overlapping errors. The chosen parameters are $q = 3$, and $n = (q^5 - 1)/(q-1)$.

First, we use a *probabilistic argument* for the GV bound as in [5] to prove that the code $\mathcal{C}_0^{\perp}$ of parameters $[n, l, d_0^{\perp}]_q$ generated by a matrix $\boldsymbol{H}_0 \in \mathbb{F}_q^{l \times n}$ whose dual code $\mathcal{C}_0$ of minimum distance $d_0 \geq u - q + 3$ simultaneously exists. Second, we need to show that we can extend $\mathcal{C}_0^{\perp}$ to obtain a code $\mathcal{C}$ of parameters $[n, k, d]_q$ generated by a full rank matrix of larger dimension $l+1 \leq k$ so that we apply the method given in [5] with a suitable adaptation which is a restricted version for linear codes from the suggested method in [6] to extend $\mathcal{C}_0^{\perp}$. If $\xi(n, h_q(\delta_0^{\perp}), \varepsilon) < \zeta(n, d_0^{\perp}, q)$, then $\exists$ (u,t)-PSMC with dimension $k - l = \lceil \zeta(n, d_0^{\perp}, q) \rceil + 1 - \lfloor \xi(n, h_q(\delta_0^{\perp}), \varepsilon) \rfloor$ as illustrated in Figure 2.

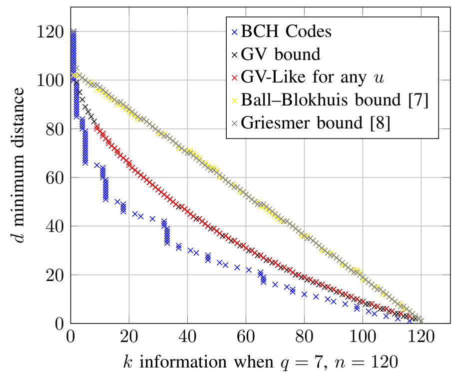

Figure 2. Comparisons of bounds of our new GV-Like-Bound for $n = 120$ and $q = 7$. It shows there exist error correcting and masking codes that satisfy GV bound starting after a certain dimension ($k = 9$) and particular minimum distance ($d = 81$). For this plot, $9 \leq \lceil \zeta(n, d_0^{\perp}, q) \rceil + 1 \leq 116$ while $0 \leq \lfloor \xi(n, h_q(\delta_0^{\perp}), \varepsilon) \rfloor \leq 110$.

References

[1] A. Wachter-Zeh and E. Yaakobi, "Codes for Partially Stuck-at Memory Cells," *IEEE Trans. Inf. Theory*, vol 62, no. 2, pp.639-654, 2016.

[2] C. Heegard, “Partitioned Linear Block Codes for Computer Memory with 'Stuck-at' Defects,” *IEEE Trans. Inf. Theory*, vol. 29, no. 6, pp. 831–842, 1983.

[3] B. Gleixner, F. Pellizzer, and R. Bez, “Reliability Characterization of Phase Change Memory,” in *2009 10th Annual NVMTS*. IEEE, 2009, pp. 7–11.

[4] A. Pirovano, A. Redaelli, F. Pellizzer, F. Ottogalli, M. Tosi, D. Ielmini, A. L. Lacaita, and R. Bez, “Reliability Study of Phase-Change Nonvolatile Memories,” *IEEE Trans Device Mater Reliab*, vol. 4, no. 3, pp. 422–427, 2004.

[5] R. R. Varshamov,"Estimate of the number of signals in error correcting codes," *Dokl. Akad. Nauk SSSR*, 117: 739–741, 1957

[6] E. N. Gilbert, "A comparison of signalling alphabets", *Bell Syst. tech. j.*, 31: 504–522, 1952 doi:10.1002/j.1538-7305.1952.tb01393.x.

[7] S. Ball and A. Blokhuis, “A Bound for the Maximum Weight of a Linear Code,” *SIAM J. Discrete Math.*, vol. 27, no. 1, pp. 575–583, 2013.

[8] J. H. Griesmer, “A Bound for Error-Correcting Codes,” *IBM J Res Dev*, vol. 4, no. 5, pp. 532–542, 1960.