Codes for Memory Cells with Unreachable Levels

Haider Al Kim[1,2], Vladimir Sidorenko[1]
[1]Institute for Communications Engineering, Technical University of Munich (TUM), Germany
[2]Electronic and Communications Engineering, University of Kufa (UoK), Iraq
Email: haider.alkim@tum.de, vladimir.sidorenko@tum.de

***Abstract*—This paper focuses on the problem of unreachable memory cells (UMCs). A cell is considered *s-unreachable* if it cannot store values for higher levels $> s$. We first provide codes for UMCs and prove that the same parameters as in the coding schemes in recent work on partially stuck memory cells (PSMCs) can be used for UMCs. In the second part, we present a various code construction built upon poly-alphabetic codes, allowing different alphabets to be used in each coordinate. We improve upon previous works for some code parameters while considering certain patterns of unreachable cell distributions. The latter construction gives UMC codes that reach the Singleton-like bound derived for polyalphabetic codes, and our Gilbert–Varshamov-like-bound built on polyalphabetic codes outperforms GV-like bound on partially stuck-at code constructions.**

***Index Terms*—flash memories, phase change memories, non-volatile memories, defective memory, unreliable cells, error-correcting codes, Singleton bound, Gilbert-Varshamov bound**

I. Introduction

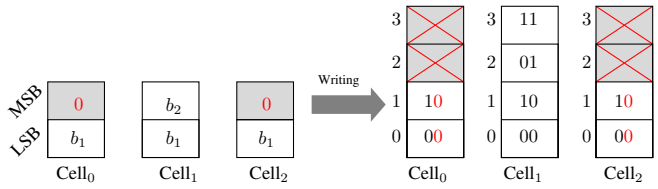
The demand for dependable storage solutions, particularly for *non-volatile memories* (NVMs) like *flash memories* and *phase change memories* (PCMs), is continuously growing for various applications. NVM is a type of memory that retains information even when not powered, and these multi-level devices offer permanent storage, fast data access, lower power consumption, enhanced physical resilience, and scalable capacity [2].

However, a major challenge with NVM technology is that its read channel degrades over time, leading to a *decline in reliability*. The repeated programming and erasing of flash memory leads to damage such as charge trapping in oxide and interface states [3], [4]. The trapped charges (the *stuck-at* levels; see [5], [6]) prevent a cell from switching its level, even when new charges are added or removed.

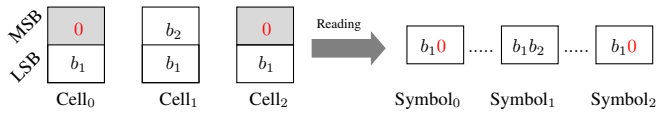
PCM cells, on the other hand, switch between an *amorphous* state and a *crystalline* state. They may become *unreliable* (also called *unreachable* [7]–[9]) when they fail to switch states during cooling and heating processes, resulting in cells only holding a single phase. In multi-level PCM cells, defects may occur in the extreme states or in the *sub-states of the crystalline*. Each multi-level cell holds one of the q levels and can be considered as a symbol over a discrete alphabet of size q.

The decline in the performance of NVM devices can be addressed by employing system-level channel codes, as these memories function as point-to-point communication systems. The use of error correction schemes for unreliable memories dates back to the 1970s [11].

Unreachable memory cell restriction has been clearly defined in [9]. It is seen as a *dual* problem for *partially stuck memory cells* (PSMC) (defined in [1], [5]). A cell is called *partially-stuck-at level s* if it can only store values which are at least s. Contrary to partially stuck memory cells in which the lower levels cannot be used, a cell is said to be *unreachable* at a level s wherever it is allowed to store *at most* that level. For example, a cell is $q-2$-unreachable (assuming q possible levels), thereby permissibly programming it until reaching its $q-2$ value (cf. Definition 1). Figure 1 depicts the writing and reading processes in an MLC memory with two unreachable cells.



(a) Writing process by the encoder for n cells whose levels are $\{0,1,2,3\}$.



(b) Reading process by the decoder for n cells whose levels are $\{0,1,2,3\}$.

Figure 1: A multi-level cell (MLC) device (cf. Section I) with two unreachable cells where only their *lower* levels are writable. Cell_0 and Cell_2 are only programmable at the first and second levels, e.g, their most significant bits (MSBs) are "0" and cannot be switched to "1" by the writing process by the encoder. In contrary, Cell_1 is a normal cell and reachable at all its four possible values, i.e., both the MSB and the LSB (least significant bit) are programmable. The decoder reads each level index, e.g., an integer from $\{0,1,2,3\}$ and interprets it as two bits (b_1 and b_2) to construct a symbol from the alphabet $\{0,1\}^2$. Here, Symbol_0 represents the readout of the two bits ($b_1 0$) from Cell_0, Symbol_1 represents the readout from Cell_1, and so forth. The concatenation of Symbol_0, Symbol_1, Symbol_2, up to Symbol_{n-1} forms the assembled codeword of length n.

II. Related Work

In [5, Section VIII], the authors provide code constructions (without correcting substitution errors) for unreachable memories at levels $\tilde{\boldsymbol{s}} \in \tilde{\Psi} \subset \mathbb{F}_q^n$ (cf. Section IV-A and Definition 2). The storing process may fail due to substitution errors caused by inter-cell interference noise or other disturbances [2]. Additionally, the reading process may also be unsuccessful [12]. The recent work in [1] presents various code constructions that capable of correcting substitution errors besides *masking* partially stuck memory cells (PSMC). The process of "masking" determines codewords whose entries coincide with the writable levels [1], [5]. Al Kim et al. [1][1] propose improvements upon the required redundancy from [5, Construction 5] and from [6]. Gabrys et al. in [9] suggest

This work has received funding from the German Research Foundation (Deutsche Forschungsgemeinschaft, DFG) under Grant No. WA3907/1-1, the European Union's Horizon 2020 research and innovation program through the Marie Sklodowska-Curie under Grant No. 713683, and the EU Horizon 2020/European Research Council (ERC) Starting Grant under Grant No. 801434. V. Sidorenko is on leave from the Institute for Information Transmission Problems, Russian Academy of Sciences.

[1]A refined and extremely extended version of the paper: H. Al Kim, S. Puchinger and A. Wachter-Zeh, "Error Correction for Partially Stuck Memory Cells," 2019 XVI International Symposium "Problems of Redundancy in Information and Control Systems" (REDUNDANCY), Moscow, Russia, 2019, pp. 87-92, doi: 10.1109/REDUNDANCY48165.2019.9003352.