ECC-Map: A Resilient Wear-Leveled Memory-Device Architecture with Low Mapping Overhead

Natan Peled
Technion – Israel Institute of Technology

Yuval Cassuto
Technion – Israel Institute of Technology

Abstract

New non-volatile memory technologies show great promise for extending the memory hierarchy, but have limited endurance that needs to be mitigated toward their reliable use closer to the processor. Wear leveling is a common technique for prolonging the life of endurance-limited memory, where existing wear-leveling approaches either employ costly full-indirection mapping between logical and physical addresses, or choose simple mappings that cannot cope with extremely unbalanced write workloads. In this work, we propose ECC-Map, a new wear-leveling device architecture that can level even the most unbalanced and adversarial workloads, while enjoying low mapping complexity compared to full indirection. ECC-Map is evaluated on common synthetic workloads, and is shown to significantly outperform existing wear-leveling architectures.

1 Introduction

The new device architecture we present in this paper aims to solve the wear-leveling problem by enhancing the flexibility of the mapping between logical line addresses (LLAs) and physical line addresses (PLAs), while keeping the costs associated with this new mapping small and controlled. The fundamental problem of wear-leveling mapping architectures is that they must be able to map frequently-written LLAs flexibly across the PLA space, and tracking a flexible workload-dependent mapping costs memory and processing resources. The main idea of ECC-Map is to use *a family of mapping functions* to maintain the LLA-PLA mapping. A large family of functions gives more flexibility than the simple functions used in prior work, and at the same time much lower cost than demanded by a mapping table in memory. Around these mapping functions we design the entire mapping architecture and its algorithms, which we summarize in the following by stating ECC-Map's main ingredients, which, to the best of our knowledge, have not been used in prior wear-leveling architectures. 1) **A family of** efficiently computable **mapping functions** with properties allowing effective reclaiming of unused wear. Each member of the family is defined by an integer *mapping index*. 2) **A sliding window** bounding the range of mapping indices used throughout the device at a given time. The window size controls the mapping complexity. 3) **Selective remapping** of specific logical addresses from their current physical locations to a new location determined by a subsequent mapping index. 4) **A remapping trigger** invoked when a physical location reaches a designated wear threshold based on either a write-count estimate or reliability estimate.

2 Summary of Results

2.1 The mapping functions

To implement the family of functions, we use (see Figure 1) encoding functions of cyclic error-correcting codes (ECC), used elsewhere for error correction and detection, including as cyclic redundancy check (CRC) codes. We choose these functions for the several advantages they offer: 1) efficient hardware

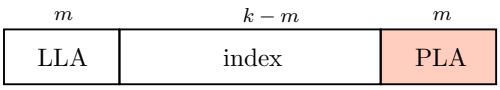

Figure 1: LLA-to-PLA mapping. The input LLA and the function index (in white) comprise the input to the ECC encoder. The encoder's output (shaded orange) gives the PLA resulting from the mapping. The inverse mapping (PLA-to-LLA) is implemented by the exact same encoder function (with shifted inputs), exploiting the cyclic property of the code.

implementation, 2) simple reverse mapping, and 3) spreading an LLA mapping across the entire PLA space. The third feature is in general not satisfied by alternative options such as cryptographic pseudo-random permutations. Common cryptographic functions also require significantly higher computation load relative to cyclic ECC encoding.

2.2 Remapping operations

With the use of multiple mapping functions, different LLAs can be mapped by different mapping indices, which allows *selective remapping* of heavily written LLAs by incrementing their mapping index, while keeping other LLAs at their current mapping indices and physical locations. Thanks to the device over-provisioning, it is possible to remap an LLA with minimal change to the mapping of other LLAs. Such a mapping is triggered by specifying a wear threshold $\phi < w_{max}$ (where w_{max} is the endurance of the memory lines), with the following policy: a *host write* to an LLA mapped to a PLA that had *exceeded* ϕ *writes* will be written after remapping the LLA to a different PLA. The value of ϕ is an optimization variable, set to vacate a worn PLA "just in time" to keep it usable for all future remappings. Although ϕ is given as a count of physical writes, implementing the remapping trigger does not require maintaining PLA write counters (which would be expensive). Instead, reaching a threshold of ϕ can be detected by a reliability measurement of the PLA, for example by counting the number of bit errors corrected by the decoder of the data error-correcting codes.

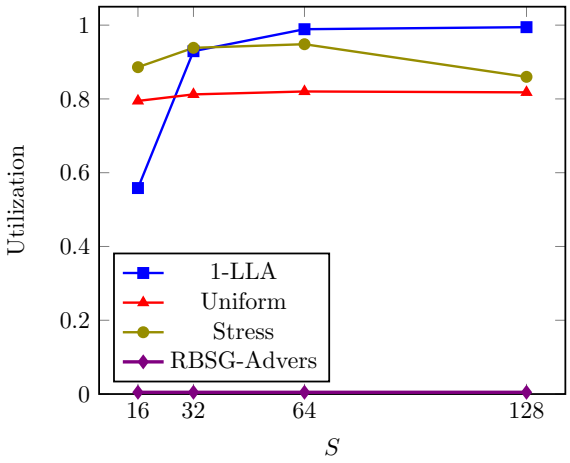

Figure 2: Utilization as a function of the mapping window size S.

2.3 Evaluation

To bound the mapping cost of ECC-Map, we introduce a cost parameter S, and restrict all LLAs to have mapping indices in a subset of S consecutive indices. This subset changes as a sliding window throughout the device lifetime. That is, the mapping-index set is $\{\text{base}, \text{base}+1, \ldots, \text{base}+S-1\}$, for some integer state variable base. As our performance measure we use the $Utilization = \frac{\#Host\ writes}{w_{max} \cdot N}$, where N is the number of PLAs in the device (the denominator is a fundamental upper bound on the number of host writes, hence the utilization is a number between 0 and 1). A sample result showing the advantage of ECC-Map on three common synthetic workloads is given in Figure 2, as a function of S. The 1-LLA workload is the most challenging of the three: writing constantly to a single LLA until device end of life. For comparison, the plot shows as a horizontal line the utilization of the main prior architecture (RBSG) [1] on the 1-LLA workload.

References

[1] M. Qureshi, J. Karidis, M. Franceschini, V. Srinivasan, L. Lastras, and B. Abali. Enhancing lifetime and security of PCM-based Main Memory with Start-Gap Wear Leveling. In *2009 42nd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*.