Exploiting Reuse Information to Reduce Refresh Energy in On-Chip eDRAM Caches

Alejandro Valero, Julio Sahuquillo, Salvador Petit, and José Duato
Department of Computer Engineering
Universitat Politècnica de València
Valencia, Spain
alvabre@gap.upv.es, {jsahuqui, spetit, jduato}@disca.upv.es

ABSTRACT

This work introduces a novel refresh mechanism that leverages reuse information to decide which blocks should be refreshed in an energy-aware eDRAM last-level cache. Experimental results show that, compared to a conventional eDRAM cache, the energy-aware approach achieves refresh energy savings up to 71%, while the reduction on the overall dynamic energy is by 65% with negligible performance losses.

Categories and Subject Descriptors

B.3.2 [**Design Styles**]: Cache memories

Keywords

MRU-Tour; on-chip caches; selective refresh

1. PROPOSED APPROACH

Refresh operations in on-chip eDRAM caches incur in a significant fraction of the total dynamic energy consumed by these memories. Prior works have addressed this energy overhead by reducing the impact of inter-cell variability on refresh energy [3].

Unlike these proposals, this work pursues to minimize the number of refresh operations by applying selective refresh in an *energy-aware* eDRAM last-level (L2) cache. The proposal aims to avoid refreshing *useless* lines in order to save energy and prevent performance losses. The devised refresh policy exploits reuse information to decide whether a cache block should be refreshed. To this end, the proposal works on the MRU-Tour (MRUT) concept [2], which is referred to as the number of times that a block enters in the MRU position of the LRU stack. Based on the observation that most blocks in L2 caches exhibit a single MRUT at the time they are evicted, the refresh mechanism does not periodically refresh the target block if it has only one MRUT.

To save energy, the proposed energy-aware cache only accesses in a first stage the tag array and a predicted cache bank in the data array. If the requested block is not stored in the predicted bank, then the target bank is accessed in a second stage. This mechanism always predicts the same physical bank, and MRU blocks are stored in that bank by performing data movements between ways similar to as done in [1]. Each bank implements two cache ways.

2. EXPERIMENTAL RESULTS

The proposed cache has been modeled with SimpleScalar and CACTI to obtain performance and energy consumption, respectively, for SPEC benchmarks.

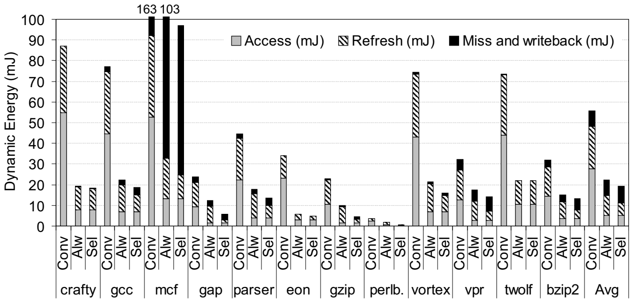

Figure 1: Dynamic energy consumption (in mJ).

Figure 1 plots the dynamic energy of a 2MB-16way L2 cache organization classified into *Access*, *Refresh*, and *Miss and writeback* energy. The latter covers the expenses of accessing to a 2GB DRAM main memory. The energy consumption of the data movements between ways has been taken into account in the *Access* category. Label *Conv* refers to a conventional eDRAM cache that accesses in parallel all the tags and all the banks and uses a conventional refresh policy. *Alw* and *Sel* refer to the conventional and selective policies, respectively, both applied in the energy-aware scheme.

Compared to *Conv*, both *Access* and *Refresh* energy are largely reduced by the energy-aware scheme mainly because it accesses first just the MRU bank. As observed, *Sel* reduces the refresh consumption with respect to *Alw*, and it compensates the increase in the *Miss and writeback* energy caused by requests to non-refreshed blocks. Overall, the refresh savings of *Sel* are on average by 71% with respect to *Conv*. This percentage is by 65% when the whole energy is considered. These benefits come with minimal performance degradation (by 1.3% on average) with respect to *Conv*.

3. ACKNOWLEDGMENTS

This work was supported by the Spanish Ministerio de Economía y Competitividad (MINECO) and Plan E funds, under Grants TIN-2009-14475-C04-01 and TIN2012-38341-C04-01.

4. REFERENCES

[1] J. Lira et al. Implementing a hybrid SRAM/eDRAM NUCA architecture. In *Proc. 18th Int'l Conf. High Perform. Comput.*, pages 1–10, 2011.

[2] A. Valero et al. Combining Recency of Information with Selective Random and a Victim Cache in Last-Level Caches. *ACM Trans. Arch. Code Opt.*, 9(3):16:1–16:20, 2012.

[3] C. Wilkerson et al. Reducing Cache Power with Low-Cost, Multi-bit Error-Correcting Codes. In *Proc. 37th Int'l Symp. Comput. Arch.*, pages 83–93, 2010.