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

Alejandro Valero
Julio Sahuquillo
Salvador Petit
José Duato

1. INTRODUCTION

Refresh operations in on-chip eDRAM caches incur in a significant fraction of the total dynamic energy consumed by these memories as shown in Figure 1. Refresh energy increases with the cache capacity and grows up to 62% for a 16MB cache.

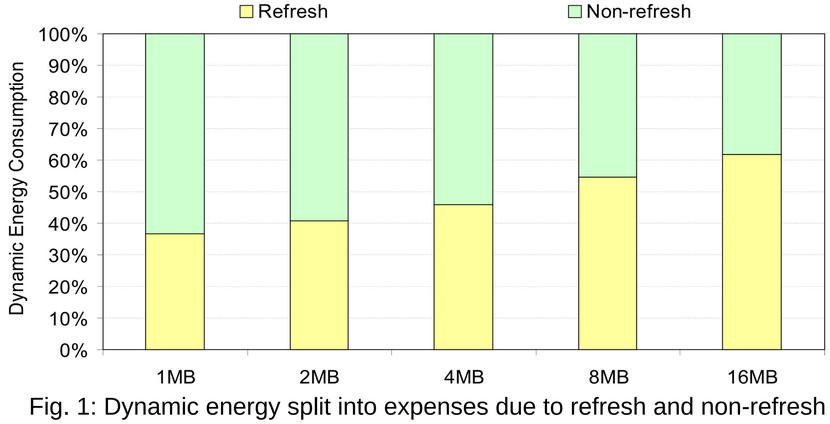

Fig. 1: Dynamic energy split into expenses due to refresh and non-refresh operations in conventional eDRAM last-level caches

This work pursues to minimize the number of refresh operations by applying selective refresh in an energy-aware eDRAM last-level (L2) cache. The proposed refresh policy aims to avoid refreshing useless lines in order to save energy and prevent performance losses, The devised mechanism 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. REUSE INFORMATION: MRUT CONCEPT

The number of MRUTs of a block is defined as the number of times that the block becomes the MRU while it resides in the cache. As shown in Figure 2, the first MRUT of a block starts when the block is brought into the cache, continues while the block is being accessed in the MRU position, and finishes when another block is referenced. The second MRUT of the block starts when it is accessed again in a non-MRU location and so on.

Based on the observation that most blocks in L2 caches exhibit a single MRUT at the time they are evicted under the LRU replacement algorithm, the proposed selective refresh policy skips the refresh operations of blocks with just one MRUT.

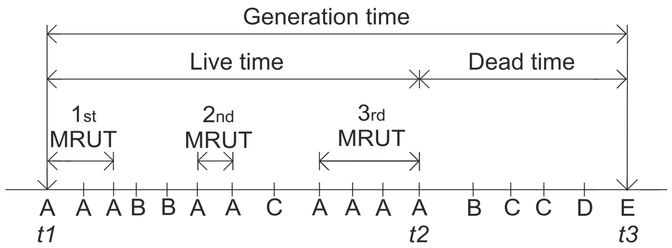

Fig. 2: Generation time of block A and its MRUTs

3. ENERGY-AWARE eDRAM CACHE ARCHITECTURE

To save energy, the proposed energy-aware cache only accesses in a first stage the tag array and a predicted bank in the data array as can be seen in Figure 3. If the requested block is not stored in the predicted bank, then only 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. Each bank has two cache ways.

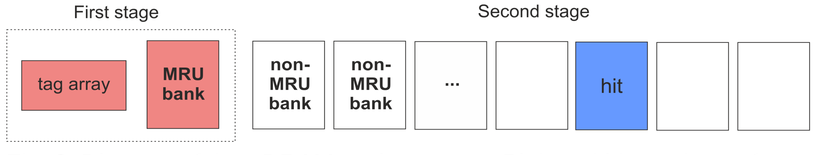

Fig. 3: Diagram of the eDRAM cache access. Pink and blue colors refer to the accessed parts in the first and the second stages (if any), respectively

4. EXPERIMENTAL RESULTS

The proposed cache has been modeled with CACTI and SimpleScalar to obtain energy consumption and performance, respectively, for SPEC benchmarks. It has been evaluated a 2MB-16way L2 cache with an access time to the tag and data array of 2 and 8 cycles, respectively. 500M instructions were run after skipping the initial 1B instructions.

4.1. Dynamic Energy

Figure 4 plots the dynamic energy consumption classified into Access, Refresh, and Miss and writeback energy. The latter covers the expenses of accessing to a 2GB DRAM main memory. Label Conv refers to a conventional eDRAM cache that accesses in parallel all the tags and all the banks of the data array 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

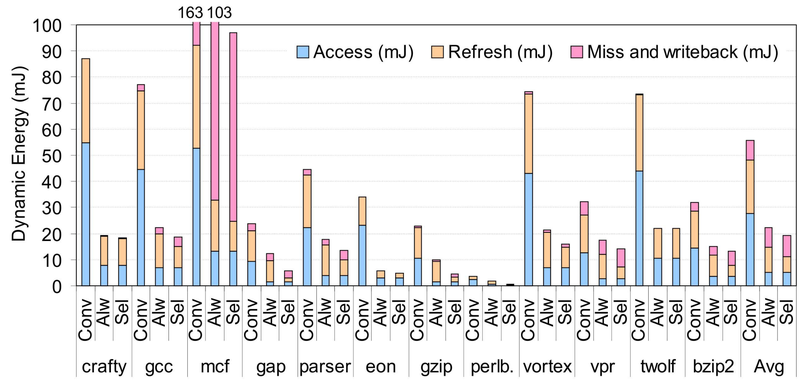

Fig. 4: Dynamic energy (in mJ) of the studied caches and refresh policies

reduces the refresh consumption with respect to Alw, and it compensates the increase in he 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.

4.2. Performance

Figure 5 shows the slowdown of the energy-aware cache with respect to the conventional scheme (Conv). In a few applications like mcf, the energy-aware approach performs better than Conv despite accessing first just the MRU bank and applying the Sel refresh policy. This is due to Conv accesses in parallel all the banks, which can increase bank contention. The differences between Alw and Sel are due to the induced main memory accesses of non-refreshed blocks. Nevertheless, the slowdown of Sel is only on average by 1.3% with respect to Conv.

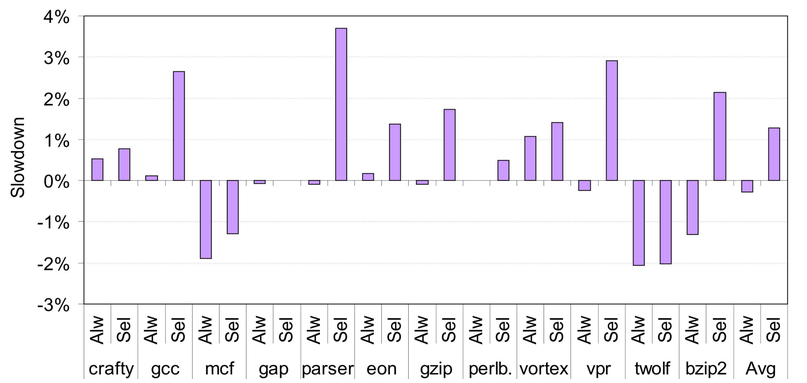

Fig. 5: Slowdown (%) with respect to the conventional eDRAM cache

5. CONCLUSIONS

This work introduced a selective refresh mechanim for on-chip last-level eDRAM caches that leverages reuse information. The proposal woks on the number of MRUTs of a block, which refers to the number of times that a block occupies the MRU position of the LRU stack. To reduce refresh energy, the mechanism skips the refresh of blocks having a single MRUT since most blocks exhibit this number of MRUTs when they are evicted. In addition, the proposal has been applied in a cache architecture that accesses first a predicted bank storing the MRU data to obtain further energy savings. Experimental results have shown that, compared to a conventional eDRAM cache, refresh energy is reduced on average by 71%, while the overall dynamic energy savings are up to 65%. These benefits are obtained with minimal impact on performance (by 1.3% on average).

27th International Conference on Supercomputing (ICS)
Eugene, OR, USA, 2013

GAP Parallel Architectures Group

GOBIERNO DE ESPAÑA MINISTERIO DE ECONOMIA Y COMPETITIVIDAD

Plan E Plan Español para el Estímulo de la Economía y el Empleo