STT-RAM Aware Last-Level-Cache Policies for Simultaneous Energy and Performance Improvement

Fazal Hameed* and Jeronimo Castrillon†
Center for Advancing Electronics Dresden (cfaed), Technische Universität Dresden, Germany
Email: *fazal.hameed@tu-dresden.de, †jeronimo.castrillon@tu-dresden.de

***Abstract*—High capacity Last Level Cache (LLC) architectures have been proposed to mitigate the widening processor-memory speed gap. These LLC architectures have been realized using DRAM or Spin-Transfer-Torque Random Access Memory (STT-RAM) memory technologies. It has been shown that STT-RAM LLC provides improved energy efficiency compared to DRAM LLC. However, existing STT-RAM LLC suffers from increased energy consumption by fetching unncesseary cache lines into the row buffer. To address this problem, we propose a *Selective Read Policy* for STT-RAM LLC. This policy is based on the idea of fetching those cache lines into the row buffer that are likely to be reused. This policy restricts the number of unneeded cache line reads and thereby reduces the energy consumption. While the *Selective Read Policy* provides energy reduction, it introduces a latency penalty. To address this problem, we propose three performance optimizations namely *Row Buffer Tags Bypass Policy*, *LLC Data Cache*, and a *Tag Organization*. These optimizations reduce the impact of the latency penalty of *Selective Read Policy* while considering STT-RAM and application characteristics. For an 8-core system, we show that our synergetic policies reduce the average LLC dynamic energy consumption by 72.6% and improve the system performance by 1.3% compared to the recently proposed STT-RAM LLC. Compared to the state-of-the-art DRAM LLC, our architecture reduces the LLC dynamic energy consumption by 90.6% and improves the system performance by 1.4%.**

I. Introduction and Background

The widening processor-memory speed severely restraints the performance of multi-core system. The problem becomes worse when these systems run complex applications with large capacity and bandwidth requirements. A potential solution to mitigate this problem is to employ die-stacked memory technologies as a large capacity Last-Level-Cache (LLC) [1]–[7]. A larger LLC improves the system performance via reducing the number of high latency accesses to bandwidth-limited off-chip memory.

In state-of-the-art LLC [3]–[5], each LLC row (2048 bytes with 32 64-byte columns) consists of four LLC sets. Each LLC set is divided into seven 64-byte cache lines (i.e. each column store a single cache line) and one tag column. The 64-byte tag column stores the tags of seven cache lines within LLC set. The tag lookup latency (i.e. 20 cycles) in existing LLC is a severe bottleneck, because it requires to read the tag column before accessing the cache line for every LLC access. To reduce tag lookup latency, the proposal in [3], [5], [6] provides a small low latency Tag-Cache which holds the tags of the recently accessed LLC sets. After a Tag-Cache miss, the tags of adjacent cache sets are loaded into the Tag-Cache exploiting the fact that adjacent cache sets will likely be accessed in the near future. A Tag-Cache hit provides significantly fast tag lookup (i.e. 2 cycles) compared to the scenario when the tags are read from the LLC.

Recent research has shown that Spin-Transfer-Torque Random Access Memory (STT-RAM) LLC [4] provides significant energy saving compared to DRAM LLC [1]–[3], [7]. However, existing STT-RAM LLC architecture [4] still faces the following key challenges and drawbacks that are addressed in this work.

1) Reduced Row Buffer Utilization: Contemporary STT-RAM LLC holds multiple banks where each bank is equipped with a Row Buffer (RB) [4]. Data in an STT-RAM LLC is typically accessed after it is fetched to the RB. Any subsequent to the same row (so-called RB hit) will bypass the STT-RAM bank access and the data is directly read from the RB. Such RB locality reduces the access latency compared to when accessing the data from the STT-RAM bank. The total energy consumption in existing STT-RAM LLC is dominated by reading large amount of data from an STT-RAM bank into the RB [4] while majority of them are not reused in the future.
2) Redundant storage of tags: Exisiting LLC [3]–[6] architecture stores the tags both in the RB and the Tag-Cache. This duplicate storage of tags exacerbates both LLC access latency and energy consumption.
3) High number of tag column read/write: Previously proposed LLC [3]–[5] architectures require to read four tag columns after a Tag-Cache miss before loading them in the Tag-Cache. This increases both the latency and energy consumption.

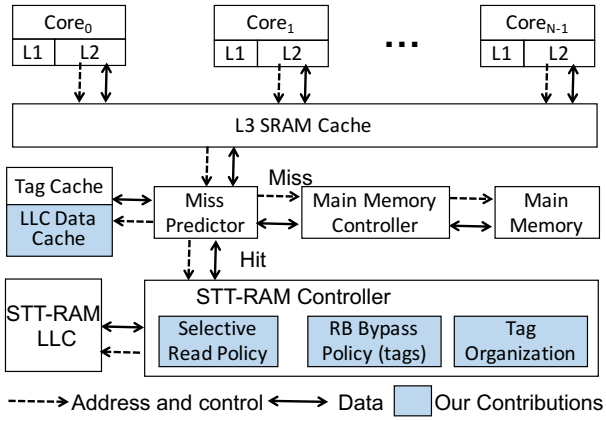

Fig. 1. An STT-RAM based LLC architecture for an N-core system highlighting our contributions.

II. Proposed STT-RAM LLC Architecture

Figure 1 depicts the STT-RAM based LLC architecture enumerating the novel contributions proposed in this work. This work presents novel STT-RAM aware and application-aware policies, while addressing the above-mentioned challenges. The details of each policy can be found in [8] (https://dl.acm.org/citation.cfm?id=3132414). In particular, this work makes the following major contributions:

A. Selective Read Policy

This work proposes a *Selective Read Policy* that exploits the fact that most of the cache lines are unnecessarily fetched into the RB as they are not likely to be reused in the near future. Figure 2 illustrates the low RB utilization showing that the probability for the RB content

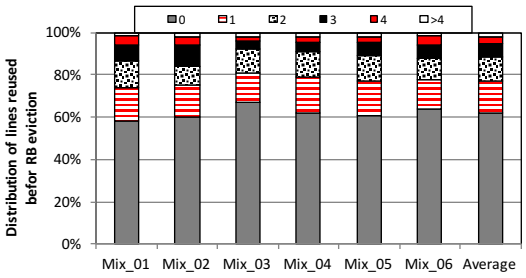

Fig. 2. Distribution of number of unique cache lines reused before RB eviction using 2KB RB size for SPEC2006 application mixes (see [8] for details

to be accessed before RB eviction is less than 40%. The *Selective Read Policy* classifies the cache lines as *highly-reused* and *lowly-reused* cache lines and it fetches only *highly-reused* into the RB. This reduces the number of RB fetches significantly and, therefore, also reduces the energy consumption. However, identifying the location of *highly-reused* cache lines requires an extra 24 cycles latency when using existing tag read policy [4]. The *Selective Read Policy* reduces the energy consumption by 72.2% compared to recently proposed STT-RAM LLC [4]. However, on the downside, it degrades the system performance by an average of 3.6% due to extra latency penalty to locate *highly-reused* cache lines.

B. RB Tags Bypass Policy

Existing LLC tag read policy [3]–[6] always fetches the tags (and the lines) in the RB before inserting them in the Tag-Cache. However, this policy induces an extraneous 24 cycles latency penalty for the *Selective Read Policy*. To reduce this latency overhead, this work proposes a *RB Tags Bypass Policy*. The proposed policy bypasses the RB for accessing the tags. Instead of fetching the tags into the RB, the proposed policy stores the tags directly in the Tag-Cache. This optimization reduces the latency penalty of *Selective Read Policy* from 24 to 6 cycles. In addition, it also reduces the energy consumption, as it avoids duplicate storage of tags. As a result, the *RB Tags Bypass Policy* reduces the performance degradation of the *Selective Read Policy* from 3.6% to 0.8%. In addition, it provides additional LLC dynamic energy saving of 14.7%.

C. LLC Data-Cache Organization

To further mitigate the latency penalty of the *Selective Read Policy*, this work proposes a small *LLC Data Cache* (LDC) organization that stores the cache lines which are likely to be accessed in near future. The *LDC* provides fast LLC access latency because for an *LDC* hit, it access the cache line from the low-latency *LDC* in two cycles due to its smaller structure. In contrast, access latency of reading a cache line from the STT-RAM LLC varies from 22 to 66 cycles depending on various scenarios (i.e. it is a RB hit/miss or a Tag-Cache hit/miss). The *LDC* alone improves the performance by 2% compared to an STT-RAM LLC without *LDC*. However, this performance improvment comes at the cost of a slight energy increase of 4.3%.

D. Tag Organization

This work proposes a *Tag Organization* that reduces the number of STT-RAM column accesses and LLC miss rate compared to existing proposals [4], [5]. The LLC row in the proposed tag organization contains a single 30-way associative set with 2 tag columns and 30 cache lines. In this organization, a read request needs an access to 2 tag columns (in contrast to 4 tag column accesses in [4]) before loading the tags in the Tag-Cache. Thus, it further reduces the latency penalty of *Selective Read Policy* by 4 cycles. In addition, it reduces LLC miss rate via providing a single 30-way set compared to four 7-way sets in existing proposal. Furthermore, it saves STT-RAM energy consumption via reading and updating less number of tag columns.

E. Putting It All Together

This section provides the performance and energy gains of the synergistic policies relative to state-of-the-art DRAM [3] and STT-RAM LLC [4]. The synergistic policies combines the performance advantages of *RB Tags-Bypass Policy*, *LLC Data Cache*, and *Tag Organization* and the energy gains of the *Selective Read Policy* and *Tag Organization*. The net gain is substantially higher than the gain of a single policy. The evalutions demonstrate that our combined policies improve the average performance (1.3% and 1.4%) and energy consumption (90.6% and 72.6%) compared to recently proposed DRAM [3], [5] and STT-RAM [4] LLC architectures. The *Selective Read Policy* is the leading contributor in this energy reduction.

III. Conclusions

Simultaneously reducing STT-RAM LLC energy and latency necessitates efficient policies in order to satisfy these conflicting requirements. This work addresses how to design STT-RAM LLC to simultaneously reduce energy consumption and latency, while taking into account both application and STT-RAM characteristics. To this end, our *Selective Read Policy* reduces the energy consumption by restricting the number of unneeded STT-RAM column reads. While this introduces a latency penalty, we decrease the access latency by a new *Row Buffer Tags Bypass Policy*, an *LLC Data Cache* (*LDC*) organization, and a *Tag Organization*. Our evalutions demonstrate that our proposal provides simultaneous performance and energy improvements compared to existing works.

Acknoledgments

This work was partly supported by the German Research Foundation (DFG) within the Cluster of Excellence ‘Center for Advancing Electronics Dresden’ (cfaed).

References

[1] G. Loh *et al.*, “Supporting Very Large DRAM Caches with Compound Access Scheduling and MissMaps,” *IEEE Micro Magazine, Special Issue on Top Picks in Computer Architecture Conferences*, pp. 70–78, 2012.

[2] M. Qureshi *et al.*, “Fundamental Latency Trade-offs in Architecting DRAM Caches,” in *Proceedings of the 45th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, 2012, pp. 235–246.

[3] F. Hameed *et al.*, “Architecting On-Chip DRAM Cache for Simultaneous Miss Rate and Latency Reduction,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, pp. 651–664, April 2016.

[4] F. Hameed *et al.*, “Architecting STT Last-Level-Cache for Performance and Energy Improvement,” in *2016 17th International Symposium on Quality Electronic Design (ISQED)*, March 2016, pp. 319–324.

[5] F. Hameed *et al.*, “Reducing Latency in an SRAM/DRAM Cache Hierarchy via a Novel Tag-Cache Architecture,” in *Proceedings of the 51st Design Automation Conference (DAC’14)*, 2014.

[6] C.-C. Huang *et al.*, “ATCache: Reducing DRAM Cache Latency via a Small SRAM Tag Cache,” in *Proceedings of the 23rd International Conference on Parallel Architectures and Compilation Techniques (PACT)*, 2014, pp. 51–60.

[7] F. Hameed *et al.*, “Rethinking On-chip DRAM Cache for Simultaneous Performance and Energy Optimization,” in *Proceedings of the 19th conference on Design, Automation and Test in Europe (DATE)*, March 2017.

[8] F. Hameed *et al.*, “Efficient STT-RAM Last-Level-Cache Architecture to replace DRAM Cache,” in *Proceedings of the International Symposium on Memory Systems (MemSys 17)*, October 2017.