Towards a Flexible and Comprehensive Evaluation Approach for Adressing NVM Integration in Cache Hierarchy

Pierre-Yves Péneau, Florent Bruguier, David Novo, Gilles Sassatelli, Abdoulaye Gamatié

Towards a Flexible and Comprehensive Evaluation Approach for Adressing NVM Integration in Cache Hierarchy

Pierre-Yves Péneau, Florent Bruguier,
David Novo, Gilles Sassatelli and Abdoulaye Gamatié

Abstract—Emerging Non Volatile Memories (NVMs) are considered as potential candidates for replacing SRAM in future processor architectures. They offer higher density and near-zero leakage power, which is particularly interesting to reduce the overall system energy consumption. Nevertheless, NVMs can suffer from higher access costs in latency and dynamic energy consumption. Existing literature covers a large panel of techniques to mitigate these issues. However, design space explorations on NVMs are rarely discussed. In this paper, we present a comprehensive design exploration phase based on the NVSim tool to identify relevant NVM technology configuration. Selected designs are evaluated at Last-Level Cache (LLC) of multicore architectures by using a fast trace-based simulation. Energy-Delay-Product (EDP) is often used as a prime metric to evaluate NVMs integration. This papers discusses the usage of the EDP by adopting two perspectives of analysis : i) LLC only and ii) entire memory hierarchy. We show that the former, widespread in the literature, could lead to biased conclusions regarding of the impact of the NVM. It is therefore advisable to use a global perspective to accurately assess changes on the memory hierarchy.

Index Terms—Architecture, Design Space Exploration, Low-Power Design, Memory Hierarchy, STT-MRAM

1 Introduction

Multicore system design has become the *de-facto* paradigm to meet the ever-increasing performance requirements in modern systems. This however comes at the expense of higher memory requirements, notably for on-chip caches. The emergence of new memory technologies [1] opens opportunities for tackling the crucial question of leakage power. In particular, emerging Non Volatile Memories (NVMs) are considered as promising candidates for the replacement of SRAM. NVMs feature a higher density and a near-zero leakage power compared to SRAM. They enable larger cache designs and favor the decrease of energy. This is particularly beneficial since an increasing fraction of the power dissipated in modern on-chip systems is due to the static power as the design technology scales [2]. Nevertheless, compared to SRAM, NVMs suffer from higher access latency and dynamic energy, especially for writes in memory. The integration of such technologies in the memory hierarchy must be therefore carefully considered in order to avoid a detrimental impact on performance.

The literature devoted to NVMs [1] shows a wide spectrum of techniques for mitigating their drawbacks. It explores typical design integration opportunities through SRAM/NVM hybrid cache designs [3], [4], [5], [6], [7], architecture-level mechanisms such as cache replacement policies [8], [9] or cache organization [10], circuit/technology level optimization and designs [11], [12], [13] and software approaches involving OS/compiler [14], [15], [16], [17], [18].

These studies confirmed the potential benefit of NVM integration in cache memory hierarchy. Nevertheless, as shown in Table 1 on part of the aforementioned studies, the underlying assumptions they made about the considered NVM design criteria are rarely discussed or justified. In addition, the energy gains are often evaluated from a partial perspective, by focusing only on a given cache level. In this paper, we propose to discuss how NVM exploration should be exposed according to the design objectives. The energy-efficiency is assessed by considering both cache level only and entire memory system perspectives. Results show that according to the perspective, the conclusions resulting from our analysis vary. These considerations, which seem to be often neglected in literature, are of prime importance for a consistent evaluation of NVM integration impact in memory systems.

This paper addresses the above concerns by promoting a flexible and comprehensive evaluation approach. For illustration, we consider the Spin-Transfer Torque Magnetic RAM (STT-MRAM) technology, which is particularly interesting for its advanced maturity compared to other NVMs.

To reach our goal, we combine two fast simulators enabling to easily evaluate different memory designs within multicore architectures: NVSim [20], a performance, energy, and area model for NVMs, and ChampSim [21], a fast and flexible trace-based simulator. We promote a systematic parameter exploration as the preliminary step for selecting the most suitable NVM candidate designs, to be integrated within Last-Level Caches (LLCs), while improving energy-

• *Authors are with the Montpellier Laboratory of Informatics, Robotics and Microelectronics (CNRS, University of Montpellier), France.*
E-mail: firstname.lastname@lirmm.fr

TABLE 1: NVM-based design focuses in selected studies

References	Memory design optimization criteria	Energy gain evaluation perspective
[4]	N/A	L1 cache only
[10], [12], [19]	N/A	L2 cache only
[3]	Write EDP	L2 + Main memory
[8], [11]	N/A	L3 cache only

efficiency. We show how different designs (or configurations) can lead to variable outcomes. In order to limit the scope of the explorations, we consider the following integration constraints: *a)* replacing SRAM by STT-MRAM in the LLC cannot slowdown execution time by more than 5%; and *b)* the integrated STT-MRAM must fit into the SRAM silicon footprint of the LLC in the reference system. Two STT-MRAM design optimizations are considered to meet these constraints: latency-optimized and area-optimized. As a figure of merit for measuring the energy-efficiency of the entire memory system, we use Energy-Delay-Product (EDP). Our contributions can be summarized as follows:

- a comprehensive NVM memory design exploration and identification of optimized STT-MRAM designs that favor energy-efficiency in memory hierarchy.
- the assessment of selected memory designs within a multicore system model by using the NVSim and ChampSim simulators. Both memory energy and performance are evaluated with the SPEC CPU2006 benchmark suite [22]. EDP improvements of up to 27% are shown compared to the reference SRAM-based design.
- we show that an analysis based on the LLC only is too restrictive to assess the real changes in the whole memory hierarchy.

The rest of this paper is organized as follows: Section 2 presents our design space exploration approach; Section 3 introduces the experimental setup, Section 4 shows the results and explores the evaluation methodology, limitations of our work are discussed in Section 5 and Section 6 concludes.

2 Memory design characterization

This section presents our approach for the design exploration phase of NVM at LLC. We first discuss the choice of the Last-Level Cache and then present our method for design space exploration. We advocate that such method should be clearly discussed in papers to help readers to understand design choices and enhance reproducibility.

2.1 Leveraging the characteristics of STT-MRAM

Due to its higher access latency compared to SRAM, the STT-MRAM technology is *a priori* less suited to L1 caches, which require short response times. This constraint is relaxed in the memory hierarchy, notably at the LLC, thanks to the following observations. First, writes in LLC are generally not on the critical path of program execution. A CPU does not usually wait for a response on a write and can continue its execution. In addition, the cache access latency with STT-MRAM is dominated by the cell latency over the access logic for a wider range of cache sizes thanks to the high density of this technology. Hence, a larger STT-MRAM LLC would not have a significant latency gap compared to a SRAM LLC with similar size.

Large LLC memory capacity allows to store more data and generally avoids costly accesses to the main memory. For the same cache capacity, STT-MRAM requires a smaller silicon footprint than SRAM thanks to its higher density. In other words, STT-MRAM provides larger cache memory capacity for the same silicon area. In this work, we exploit this feature, to enlarge the LLC capacity up to the silicon area of the reference SRAM LLC. Hence, the following constraint must be satisfied:

$$A_{sram} \geq A_{stt}\,, \tag{1}$$

where A_{sram} is the silicon area of the reference LLC in SRAM and A_{stt} is the silicon area of the LLC in STT-MRAM. The reference LLC selected in our study is a SRAM cache with a storage capacity of 4MB. We use NVSim to determine that the STT-MRAM LLC size can be increased up to 16MB within the reference cache area constraint (see more details in Section 3.2).

In the rest of the paper, we exploit the intrinsic density of the STT-MRAM to increase the LLC size, combined with two optimized memory designs that target STT-MRAM latency and area. The former memory design mitigates the execution time overhead induced by the higher access latencies, while the latter reduces both dynamic and static energy consumption (at the cost of slightly higher access latency). Since EDP is the figure of merit in this work, these designs improve energy E and/or execution time D.

2.2 Metric-optimized STT-MRAM memory designs

There are several optimization objectives for the target STT-MRAM configurations: access latency, energy, area, etc. These criteria are usually considered according to cache levels. For instance, a first-level cache should treat CPU requests in a fast manner. Hence, the latency criterion is pre-dominant over the energy. On the other hand, a LLC should be optimized with regard to its corresponding static energy, which is also in relation with the LLC area. Since STT-MRAM has high delay and energy consumption, our motivation is to select configurations that reduce as much as possible these metrics. Moreover, the LLC area constraint specified in Equation (1) must be taken into account. Therefore, to simultaneously address all these criteria, we consider the Area-Delay-Energy Product (ADEP) as the global quality metric to optimize for configuration selection. Note that the final evaluation is based on EDP. The area criterion is only used to refine the entire design space to a smaller one where configuration respects Equation 1.

Given an optimization target (e.g., latency, energy, etc), NVSim can natively tune different design knobs such as memory array structure, sub-array size, sense amplifier and buffer design in order to explore the configuration design space [20]. We explore the design space for three different LLC sizes, 4MB, 8MB and 16MB, as shown in

Figure 1. The considered STT-MRAM model features a $22nm$ technology in NVSim. The corresponding read and write currents are compliant with state-of-the-art projection for this technology scaling [23].

To select latency optimized designs, we extract the best ADEP configuration in the design space. Let us refer to these specific STT-MRAM designs as C_{lat}. For configurations that favors area, we first do an exploration on the power consumption. In fact, it is proportional to the area and should be reduced as much as possible. Considering all designs that have a lower area than C_{lat} configurations, we explore their power consumption and found that when area is at least 10% lower than C_{lat} configuration, the leakage power could be reduced by 40% and up to 62% thanks to circuits optimizations. Let us refer to them as C_{area}.

In Figure 1, the design spaces are represented by grey zones composed of dots. Each dot denotes a particular STT-MRAM design or configuration. Each grey dot is characterized by an energy consumption (not shown[1] explicitly in the plots for the sake of simplicity), an area value and the access latency or delay. The total numbers of generated designs by NVSim for 4MB, 8MB and 16MB caches are respectively 72192, 81232 and 91288. For the sake of visibility, we perform a zoom on the most interesting region of the plots. Nevertheless, these numbers illustrate the fact that choosing a configuration is quite important and should be more discussed.

The dots in black color correspond to Pareto choices generated by NVSim according to the latency/area visualization. Configurations surrounded with a blue square have the best ADEP value among all explored possibilities. They represent the C_{lat} configurations. The designs surrounded with a red circle have the best ADEP value among the configurations with an area at least 10% lower than the C_{lat} designs. They represent the C_{area} configurations.

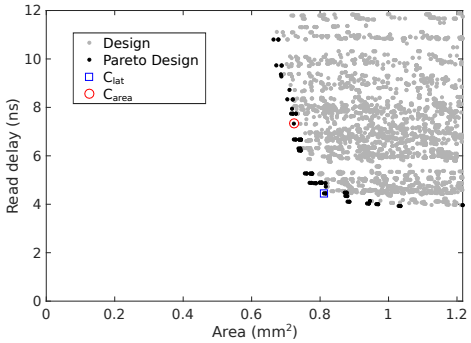
Table 4 summarizes the STT-MRAM configurations obtained with NVSim, which will be considered in the rest of this paper. It also describes the parameters of our reference SRAM design. Compared to the SRAM design, the latency for all C_{lat} designs is on average $2.4\times$ slower for read and $3\times$ slower for write. The second configuration, C_{area}, reduces the silicon footprint compared to C_{lat}, and therefore the leakage power. Nevertheless, it suffers from higher memory access latency. Compared to the reference SRAM design, latencies for read and write are respectively higher by $4.7\times$ and $5.2\times$ on average. This represents an increase of $2\times$ and $1.7\times$ compared to C_{lat}.

On average, the leakage power of SRAM is $8.4\times$ and $16.8\times$ greater compared to C_{lat} and C_{area} designs respectively (i.e., the leakage power of C_{lat} is $2\times$ greater than for C_{area}).

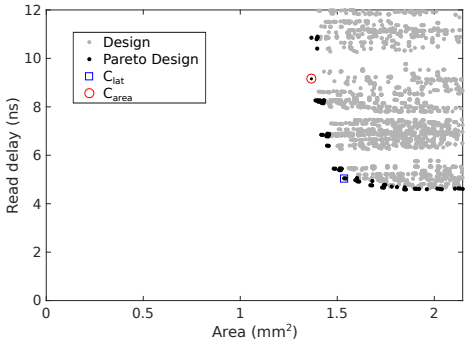
3 Experimental Setup

In this section, we detail our experimental setup based on NVSim and ChampSim. Energy models for cache hierarchy and main memory are also presented. Finally, we explain the nomenclature for the naming of LLC configurations.

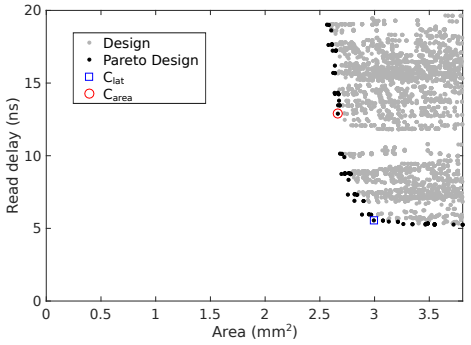
1. The energy values for selected configurations are however reported in Table 4.



(a) Design space exploration for a 4MB cache



(b) Design space exploration for a 8MB cache



(c) Design space exploration for a 16MB cache

Fig. 1: Explored configuration spaces: blue-square denotes the configuration with the best Area-Delay-Energy Product (ADEP) and red-circle configuration has the best ADEP among the configurations with at least 10% lower area than the blue-square one.

3.1 Simulation framework

Simulations are conducted with the ChampSim framework, a simulator [21] used for the Cache Replacement Championship [24] at ISCA conference. Some attractive features of ChampSim are the availability of various cache replacement techniques and a faster trace-based simulation compared to alternative tools such as SimpleScalar [25], [26] or gem5 [27], [28], [29], [30]. Note that gem5 also supports trace-driven simulation [31], [32] for accelerating design evaluation. We define a platform composed of 4 out-of-order cores with a private L1-I/D of 32KB and a unified L2 of 256KB. All cores share the LLC that uses the Least-Recently Used (LRU) replacement policy. All these configuration parameters are summarized in Table 2.

TABLE 2: Experimental setup information

L1 (I/D)	32KB, 8-way, LRU, Private, 1ns		
L2	256KB, 8-way, LRU, Unified, 2ns		
L3	Varying size/policy, 16-way, Shared		
L3 size	4MB	8MB	16MB
L3 latency	See Table 4 and Equation (2)		
Hawkeye budget	58.7KB	114.6KB	266.4KB
CPU	4 core, out-of-order, 4GHz		
Main mem. size/latency	8GB, hit: 55 cycles, miss: 165 cycles		

We use a set of traces of the SPEC CPU2006 benchmark suite provided with ChampSim. Each trace represents an isolated region of interest of 1 billion instructions. Each core executes a single-threaded application during 1 billion instructions. The cache warm-up takes 200 millions instructions while the remaining 800 millions instructions are used to report execution statistics. When a core finishes its 1 billion instructions, it continues to read the trace to simulate an activity on the memory hierarchy until all cores reach 1 billion of executed instructions. The extra activity related to this mechanism is not reported in the final results. We consider workloads composed each by four SPEC CPU2006 applications (see Table 3). We execute 20 workloads and report the geometric mean for execution time and energy values for each cache configuration.

	Core 0	Core 1	Core 2	Core 3
mix1	gobmk	libquantum	perlbench	xalancbmk
mix2	astar	bwaves	lbm	zeusmp
mix3	cactusADM	lbm	milc	perlbench
mix4	bwaves	lbm	sphinx3	wrf
mix5	astar	cactusADM	GemsFDTD	perlbench
mix6	cactusADM	GemsFDTD	gobmk	soplex
mix7	astar	cactusADM	leslie3d	sphinx3
mix8	bwaves	libquantum	perlbench	sphinx3
mix9	cactusADM	gobmk	milc	soplex
mix10	bzip2	gobmk	lbm	perlbench
mix11	astar	gobmk	milc	soplex
mix12	gobmk	leslie3d	libquantum	perlbench
mix13	bwaves	bzip2	gobmk	wrf
mix14	gobmk	lbm	leslie3d	milc
mix15	cactusADM	gobmk	milc	perlbench
mix16	bwaves	bzip2	gobmk	leslie3d
mix17	astar	bzip2	leslie3d	xalancbmk
mix18	gobmk	libquantum	wrf	xalancbmk
mix19	gobmk	lbm	milc	zeusmp
mix20	milc	perlbench	wrf	zeusmp

TABLE 3: Workloads details

3.2 Calibration of interconnect transfer latency

The total access time of LLC in SRAM is usually dominated by the transfer delay of the interconnect, and not by the cache access itself [33]. This mitigates the impact of the potential performance penalty resulting from the integration of STT-MRAM in LLC. Therefore, we define the total access time L_T for the LLC as follows:

$$L_T = L_I + L_C\,, \tag{2}$$

where L_I is the interconnect latency and L_C the cache latency.

Our evaluation framework is calibrated based on an Intel i7 processor where the LLC latency for a 2MB SRAM cache is $5ns$, i.e., $L_T = 5ns$. We extract the cache access latency of a 2MB SRAM cache and we obtain $L_C = 1.42ns$. We calculate the interconnect latency $L_I = L_T - L_C = 3.58ns$ that is used as an offset added to each cache latency L_C mentioned in Table 4. Then, we convert this latency in cycles w.r.t. the CPU frequency.

TABLE 4: Cache configuration parameters for SRAM and STT-MRAM

	SRAM	STT-MRAM C_{lat}			STT-MRAM C_{area}		
Size (MB)	4	4	8	16	4	8	16
Read lat. (ns)	2.10	4.45	5.05	5.55	7.33	9.16	12.89
Write lat. (ns)		6.05	6.31	6.52	8.05	10.85	13.63
Read en. (nJ)	0.10	0.26	0.29	0.31	0.27	0.27	0.28
Write en. (nJ)	0.09	0.30	0.33	0.35	0.31	0.31	0.32
Leakage (mW)	124.25	8.98	16.72	32.90	5.41	7.10	12.41
Area (mm^2)	3.02	0.81	1.54	2.99	0.72	1.37	2.66

3.3 Energy models

For each cache level, we extract the energy cost of each memory operation, i.e., read and write, and multiply it by the number of reads and writes observed on this cache level, as follows:

$$E^i = R^i \times E^i_R + W^i \times E^i_W + T \times P^i_{leak}\,, \tag{3}$$

where i is the i^{th} cache level; R^i and W^i are respectively the numbers of reads and writes; E^i_R and E^i_W are respectively the costs of a read and write operation; T is the execution time and P^i_{leak} is the leakage power of the i^{th} cache level.

For the main memory, we consider a DRAM model built from Micron Technology [34] (see Table 5). We model a 8GB DDR3 with 2 DIMM, 8 ranks per DIMM, 8 banks per ranks, organized with 16×65536 columns with 64B on each row. Thus, each bank contains 64MB of data, each rank 512MB, and each DIMM 4GB. We use the following equation to compute the energy consumption [35]:

$$E_m = E_a + E_b + E_c\,, \tag{4}$$

$$E_a = R \times RD + W \times WR\,, \tag{5}$$

$$E_b = (R_M + W_M) \times (PRE + ACT)\,, \tag{6}$$

$$E_c = (T/T_{REF}) \times REF + T \times ACT_{BG}\,, \tag{7}$$

where E_m is the total main memory energy consumption; E_a is the consumption due to read and write, E_b is the consumption due to row buffer misses that leads to pages precharge and activation, and E_c is the static energy due to refresh and active background. Details are the following : R and W are respectively the total numbers of reads and writes (i.e., hits and misses); RD and WR are respectively the unit cost per read and write; R_M and W_M are respectively the numbers of read and write misses; PRE and ACT are respectively the cost of page precharge and activation; T is the execution time, T_{REF} is the self-refresh frequency; REF is the cost of a refresh and ACT_{BG} is the active background energy consumption. We do not consider low-power mode, which reduces ACT_{BG}. The full configuration is summarized in Table 5.

TABLE 5: Main memory configuration [34]

RD	WR	REF	PRE	ACT	ACT_{BG}	T_{REF}
0.47 nJ	0.47 nJ	46.33 nJ	0.22 nJ	0.38 nJ	0.027 W	64 ms

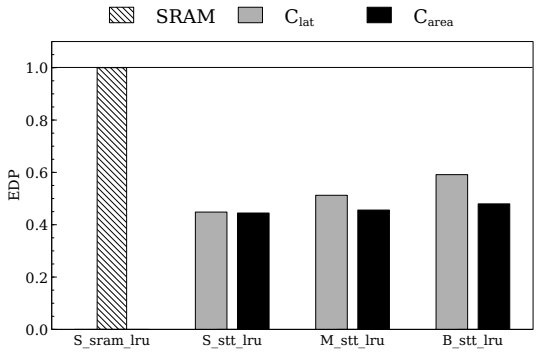


Fig. 2: EDP results with the LLC only perspective

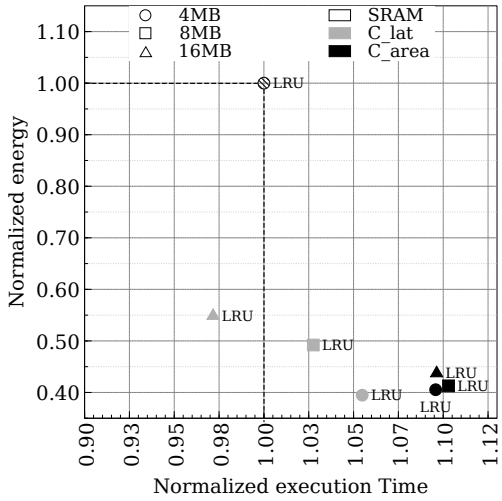


Fig. 3: Trade-off between normalized execution time and LLC energy consumption

3.4 Nomenclature of LLC configurations

In the sequel, we use the following notation convention: *X_Y_Z*, where *X*, *Y* and *Z* respectively denote cache size, technology and applied replacement policy. For the cache size, we distinguish 3 cases: *S* (small) is 4MB, *M* (medium) is 8MB and *B* (big) is 16MB. For instance, the reference LLC setup, i.e., 4MB SRAM with LRU policy, is noted *S_sram_lru*. For STT-MRAM configurations, we add the *_*C_{lat} and *_*C_{area} suffixes to refer to their corresponding setups, as described in Table 4.

4 Evaluation and Analysis of Selected Designs

In this section, we first conduct an analysis of the experimental results by adopting two perspectives. The first one, called *LLC only perspective*, is widely used in the literature. It consists in analyzing the impact of a memory technology change at the level where this memory is introduced (e.g., L1, main memory etc). The second one, called *entire memory perspective*, is less presented although it provides more informations. We show that the first option could lead to wrong design decision, and advocate for the use of a larger perspective. In both case, the LRU replacement policy is considered.

In a second step, we propose to improve our experimental results by adopting a new cache replacement strategy to mitigate the numbers of write on the LLC.

Unless specified otherwise, all results in this section are normalized w.r.t. the reference setup, i.e., *S_sram_lru*.

4.1 Results analysis with different perspective

4.1.1 LLC only perspective

Figure 2 shows the EDP for the LLC only. We observe that both C_{lat} and C_{area} outperform the SRAM reference. The best EDP with LRU is achieved by *S_stt_lru_*C_{area} with an improvement of 56%. Regardless of the cache size, each C_{area} configuration always outperforms its C_{lat} equivalent. Figure 2 also show minimal variation between all C_{area} configurations (3.5% at maximum), while C_{lat} configurations exhibits a more perceptible variation of 14.3%. This means that increasing the cache size has almost no impact on EDP with C_{area} designs, but affects the EDP with C_{lat} configurations. This could be explain by the distribution of the energy and the execution time depicted on Figure 3. Results for all C_{lat} and C_{area} configurations are normalized to the SRAM reference. One can observe that with C_{area} designs, distribution of E and D stays stable with minor variations. Conversely, C_{lat} designs have an impact on both energy and delay. As the cache size increases, execution time is reduced while energy consumption of the LLC is increased. From *S_stt_lru_*C_{lat} to *B_stt_lru_*C_{lat}, time is reduced by 8% and energy consumption is higher by 15%. Note that *B_stt_lru_*C_{lat} is the only configuration which provides lower results than the SRAM reference in terms of energy consumption and execution time. However, this design represents the worst configuration in terms of EDP when considering the LLC only perspective.

4.1.2 Entire memory hierarchy perspective

Figure 4 shows the results in terms of EDP for all designs w.r.t. the SRAM reference. We observe that STT-MRAM C_{lat} always outperforms *S_sram_lru* in terms of EDP. This first observation is the opposite of previous results when considering the LLC only. This situation illustrates the importance of the perspective for the interpretation of the results. When considering the entire memory hierarchy, C_{lat} designs worth to be investigated. The best EDP for C_{lat} with LRU is achieved by *B_stt_lru_*C_{lat} with an average improvement of 16%.

Regarding C_{area} designs, *B_stt_lru_*C_{area} is the only configuration that preserves EDP w.r.t. *S_sram_lru*. Others configurations degrade EDP up to 3.5%. These results are in conflict with previous observations when we consider the LLC only perspective. Now, C_{area} designs have a negative impact or no impact on the EDP, while an improvement of up to 56% has been previously observed.

Figure 5 depicts the Pareto distribution of execution time and energy consumption when considering the entire

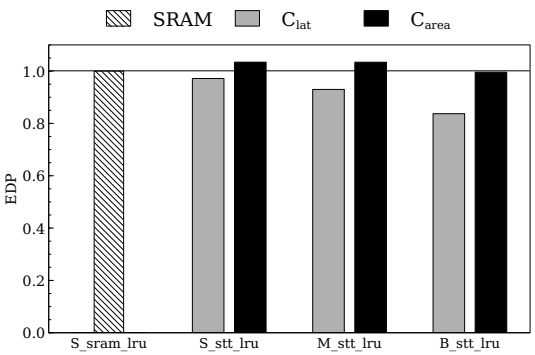


Fig. 4: EDP results with the entire memory hierarchy perspective

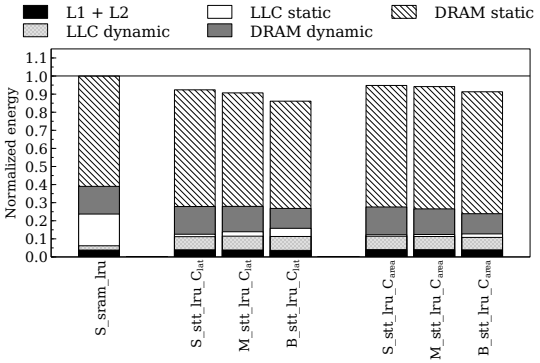


Fig. 6: Energy breakdown for all architectures with LRU

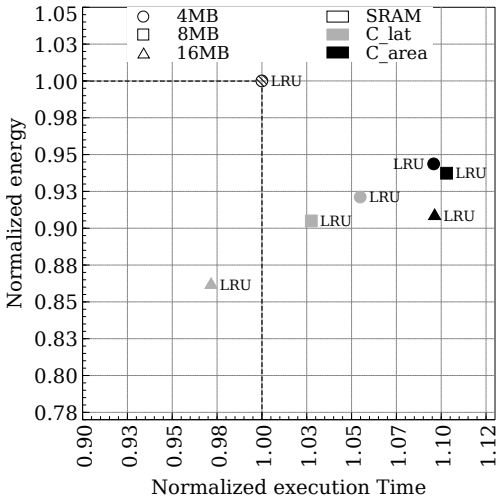


Fig. 5: Trade-off between normalized execution time and memory system energy consumption

memory hierarchy. As in Figure 3, results are normalized to the SRAM reference. Execution time remains identical, only the energy consumption varies. For C_{area} designs, trends remains identical and we observe a small impact on the energy consumption. Compared to LLC only perspective, *B_stt_lru_*C_{lat} now reduces the energy consumption, which was not the case before. However, this decrease is around 3% and remains marginal. The interesting part concerns C_{lat} designs, where trends are now reversed. As the cache size is increased, it reduces both the execution time and the energy consumption. Moreover, *B_stt_lru_*C_{lat} is the only configuration of STT-MRAM C_{lat} with LRU that outperforms *S_sram_lru* for both energy and execution time, leading to a best EDP result. This observation is in contrast with Figure 3, where *B_stt_lru_*C_{lat} is also the only configuration providing improvements w.r.t. the reference, while being the worst in terms of EDP. To explain this result, Figure 6 presents the breakdown of the energy consumption between the different parts of the architecture.

Results show that the main memory is responsible of a large part of the memory hierarchy energy consumption (between 81% and 87%). We note that this disproportion may come from the lack of low-power mode in our model. The dynamic energy consumption of the main memory is identical for both C_{lat} and C_{area} designs. It varies according to the LLC size. On the contrary, the static energy consumption of the main memory differs for both designs: it is around 70% for C_{lat} and between 71% and 75% for C_{area}. This gap comes from the execution time overhead induced by C_{area} designs. The higher the execution time, the higher the static energy consumption of the main memory. This overhead is depicted on Figure 5 and is around 10% for C_{area} configurations.

Hence, the low overhead in terms of execution time for C_{lat} designs, especially *B_stt_lru_*C_{lat} which exhibits a speedup, decreases the energy consumption of the main memory and explains the EDP results and the Pareto plot of Figures 4 and 5 respectively.

The energy breakdown also explains EDP results when considering the LLC only perspective. Regarding the LLC energy consumption between C_{lat} and C_{area}, the latter always provides a lower energy consumption than the former. It is particularly true as the cache size increases. This comes from the lower leakage power of C_{area} and explains the EDP results observed in Figure 2.

In Section 1, we mentioned that execution time should not be increased by more than 5%. Figures 3 and 5 show that none of the C_{area} designs and one C_{lat} design do not respect this constraint. Although this slowdown remains small considering the latency overheads discussed in Section 2.2, we propose to investigate a new cache replacement policy for the LLC in order to reduce the execution time overhead.

4.2 Evaluation with advanced cache replacement policy

Performance of C_{area} can be enhanced by considering advanced cache replacement techniques that reduce both execution time [36] and energy consumption [9], [37]. Here, we illustrate the latter case by replacing LRU with the Hawkeye replacement policy [38]. This policy bases its eviction decisions by reconstructing the MIN algorithm [39], a theoretical and optimal [40] algorithm in terms of miss avoidance. By avoiding miss events, the replacement policy prevents expensive requests to the main memory. This has a positive impact on both energy and execution time. Main memory will be less solicited, which saves dynamic energy. The extra time induced by external requests (see Table 2) is saved, leading to a faster execution and a reduction of the static energy consumption for the entire architecture. Note

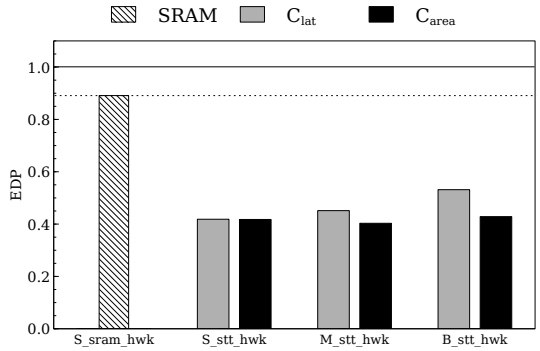


Fig. 7: Energy-Delay-Product considering the Hawkeye replacement policy from the LLC only perspective

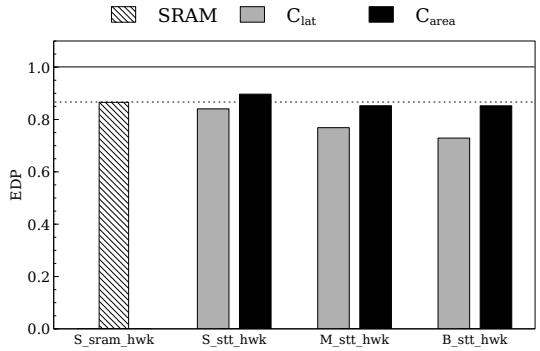


Fig. 8: Energy-Delay-Product considering the Hawkeye replacement policy from the entire memory hierarchy perspective

that Hawkeye hardware budget is negligible. On average, it represents 1.5% of the total cache capacity in our setup. Thus, it fits into the initial area budget A_{sram} since all STT-MRAM configurations preserve the constraints defined in Section 2.1.

Regardless of the perspective, the Hawkeye replacement policy has no negative impact in terms of EDP. For the LLC only view (Figure 7), all C_{lat} and C_{area} designs still outperform the SRAM reference and *S_sram_hwk*, a SRAM LLC with the Hawkeye replacement policy. However, the impact of Hawkeye remains minimal : up to 6% and 5% for C_{lat} and C_{area} respectively.

Considering the entire memory depicted on Figure 8, one can observe that both *M_stt_hwk_*C_{area} and *B_stt_hwk_*-C_{area} configurations ,which previously exhibits a higher EDP than the reference, now outperform both *S_sram_lru* and *S_sram_hwk*. EDP improvement w.r.t. *S_sram_lru* is 14.7% for *M_stt_hwk_*C_{area} and *B_stt_hwk_*C_{area}. The best EDP improvement is achieved by *B_stt_hwk_*C_{lat} by 27% w.r.t. *S_sram_lru*.

Let us compare the two perspective. On Figure 7, Hawkeye improves EDP results. However, it is used on an already-positive use case since EDP is improved by at least 40% without Hawkeye. Results also show that the impact of this modification in terms of execution time is low, up to 6%. This puts into perspective the usage of an architectural improvement. Even considering the small hardware budget and without time constraint as discussed in Section 1, the gain appears to be not sufficient to justify the integration of a complex replacement strategy to the LLC. Considering the entire memory perspective, Figure 8 shows more significant results. In such a case, the addition of an enhanced replacement strategy make two STT-MRAM designs fit into our initial constraints defined in Section 1. Moreover, Hawkeye is now responsible of a gain from 10.3% to 27%.

5 Limitation and Discussion

This section proposes a discussion about the limitation of our study in terms of accuracy, and advocates for the usage of a global memory perspective when changing the memory technology. We also discuss our approach for presenting results.

In this paper, experiments have been conducted with NVSim and ChampSim. Both tools, especially NVSim, are known for their lack of accuracy. Moreover, we do not model a low-power mode for the Micron DRAM. Therefore we overestimates its total energy consumption. While we agree with this statement, we believe that our approach remains valid regardless of the simulation tools and their accuracy. Numerous energy consumption models [41] agree on the following (simplified) relation :

$$E_{total} = \alpha Core + \beta Cache + \gamma Dram \quad \text{with } \alpha \gg \beta \text{ and } \beta \ll \gamma \,, \tag{8}$$

where E_{total} is the total energy consumption of a system and α, β and γ the factors for the energy consumption of cores, caches and main memory respectively. A large part of the literature devoted to NVMs tries to reduce β without considering γ. Since $\gamma \gg \beta$, this can lead to under/overestimate the impact of a new microelectronic or architectural design for NVMs. We illustrate this situation in Section 4, where different perspectives of analysis lead to different observations, with and without an advanced cache replacement policy. We also show that with a large perspective which consider all memory level, results in terms of EDP are easier to understand.

Regarding the literature, characteristics of NVMs are subject to debate. Numbers can vary from a factor of two and even more, while the ratio between read and write in terms of energy and/or latency is subject to a large gap [14]. This situation makes difficult the comparison with other papers. To obtains these characteristics, authors have different choices like using a simulator (NVSim, CACTI), an in-house model (sometimes derivated from an industrial partner) or re-use already published numbers from papers. These different sources increase the difficulty for authors to assess their proposal against the state of the art. In an effort to mitigate this situation, we advocate for the usage of relative numbers instead of absolute values for energy, performance, EDP, or any metric that is commonly used on the literature.

6 Conclusion

In this paper, we first present a design space exploration for NVMs based on NVSim. NVSim is often used as a black box where authors simply say that they use it and obtain such configuration. However, NVSim is a complex simulator that generates a large amount of memory designs. A wrong configuration of the tool could lead to a detrimental choice w.r.t. the final metric of evaluation. Hence, we rather perform an design exploration based on three carefully choose criteria : area, delay and energy. We identified and evaluated the impact of two STT-MRAM memory designs on energy-efficiency in terms of EDP: memory access latency-optimized versus area-optimized. The former is twice faster while the latter consumes twice less leakage power, influencing either D or E regarding of the EDP metric. We believe that such exploration should be more discussed in the literature to explain authors choices based on NVSim results.

As discussed in Section 1, existing studies often focus on one cache level only. In this paper, we compare the LLC only and the entire memory system perspectives. Results with the former are the opposite of the latter, even with the addition of an architectural improvement. This indicates that a LLC only perspective is too restrictive and can lead to inadequate design decision. Architects should have generally this in mind, by leveraging suitable frameworks such as the one shown here, in order to fully assess the impact of technology changes.

Acknowledgements

This work has been funded by the French ANR agency under the grant ANR-15-CE25-0007-01, within the framework of the CONTINUUM project.

References

[1] S. Mittal, J. S. Vetter, and D. Li, "A Survey of Architectural Approaches for Managing Embedded DRAM and Non-Volatile On-Chip Caches," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 6, pp. 1524–1537, 2014.

[2] "International Technology Roadmap for Semiconductors," 2015.

[3] S. Mittal and J. S. Vetter, "AYUSH: A Technique for Extending Lifetime of SRAM-NVM Hybrid Caches," *IEEE Computer Architecture Letters*, vol. 14, no. 2, pp. 115–118, 2014.

[4] J. Wang, Y. Tim, W.-F. Wong, Z.-L. Ong, Z. Sun, and H. H. Li, "A Coherent Hybrid SRAM and STT-RAM L1 Cache Architecture for Shared Memory Multicores," in *19th Asia and South Pacific Design Automation Conference (ASP-DAC)*. IEEE, 2014, pp. 610–615.

[5] S. Senni, L. Torres, G. Sassatelli, A. Gamatie, and B. Mussard, "Exploring mram technologies for energy efficient systems-on-chip," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 6, no. 3, pp. 279–292, 2016.

[6] S. Senni, R. M. Brum, L. Torres, G. Sassatelli, A. Gamatié, and B. Mussard, "Potential applications based on NVM emerging technologies," in *Proceedings of the 2015 Design, Automation & Test in Europe Conference & Exhibition, DATE 2015, Grenoble, France, March 9-13, 2015*, W. Nebel and D. Atienza, Eds. ACM, 2015, pp. 1012–1017. [Online]. Available: http://dl.acm.org/citation.cfm?id=2757049

[7] S. Senni, T. Delobelle, O. Coi, P. Peneau, L. Torres, A. Gamatié, P. Benoit, and G. Sassatelli, "Embedded systems to high performance computing using STT-MRAM," in *Design, Automation & Test in Europe Conference & Exhibition, DATE 2017, Lausanne, Switzerland, March 27-31, 2017*, D. Atienza and G. D. Natale, Eds. IEEE, 2017, pp. 536–541. [Online]. Available: https://doi.org/10.23919/DATE.2017.7927046

[8] C. Liu, Y. Cheng, Y. Wang, Y. Zhang, and W. Zhao, "NEAR: A Novel Energy Aware Replacement Policy for STT-MRAM LLCs," in *IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2018, pp. 1–5.

[9] P.-Y. Péneau, D. Novo, F. Bruguier, L. Torres, G. Sassatelli, and A. Gamatié, "Improving the Performance of STT-MRAM LLC Through Enhanced Cache Replacement Policy," in *International Conference on Architecture of Computing Systems*. Springer, 2018, pp. 168–180.

[10] Z. Liu, M. Mao, T. Liu, X. Wang, W. Wen, Y. Chen, H. Li, D. Wang, Y. Pei, and N. Ge, "TriZone: A Design of MLC STT-RAM Cache for Combined Performance, Energy, and Reliability Optimizations," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 37, no. 10, pp. 1985–1998, October, 2018.

[11] L. Liu, P. Chi, S. Li, Y. Cheng, and Y. Xie, "Building Energy-Efficient Multi-Level Cell STT-RAM Caches With Data Compression," in *22nd Asia and South Pacific Design Automation Conference (ASP-DAC)*. IEEE, 2017, pp. 751–756.

[12] P. Zhou, B. Zhao, J. Yang, and Y. Zhang, "Energy Reduction for STT-RAM Using Early Write Termination," in *Proceedings of the 2009 International Conference on Computer-Aided Design*. ACM, 2009, pp. 264–268.

[13] K. Kuan and T. Adegbija, "Energy-Efficient Runtime Adaptable L1 STT-RAM Cache Design," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2019.

[14] R. Bouziane, E. Rohou, and A. Gamatié, "Compile-Time Silent-Store Elimination for Energy Efficiency: An Analytic Evaluation for Non-Volatile Cache Memory," in *Proceedings of the Rapido'18 Workshop on Rapid Simulation and Performance Evaluation: Methods and Tools*. ACM, 2018, p. 5.

[15] ——, "How Could Compile-Time Program Analysis help Leveraging Emerging NVM Features?" in *EDiS: Embedded and Distributed Systems*, Oran, Algeria, Dec. 2017, pp. 1–6. [Online]. Available: https://hal.inria.fr/hal-01655195

[16] P.-Y. Péneau, R. Bouziane, A. Gamatié, E. Rohou, F. Bruguier, G. Sassatelli, L. Torres, and S. Senni, "Loop Optimization in Presence of STT-MRAM Caches: a Study of Performance-Energy Tradeoffs," in *2016 26th International Workshop on Power and Timing Modeling, Optimization and Simulation (PATMOS)*. IEEE, 2016, pp. 162–169.

[17] R. Bouziane, E. Rohou, and A. Gamatié, "Energy-efficient memory mappings based on partial wcet analysis and multi-retention time stt-ram," in *Proceedings of the 26th International Conference on Real-Time Networks and Systems*, ser. RTNS '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 148–158. [Online]. Available: https://doi.org/10.1145/3273905.3273908

[18] F. M. Q. a. Pereira, G. V. Leobas, and A. Gamatié, "Static prediction of silent stores," *ACM Trans. Archit. Code Optim.*, vol. 15, no. 4, Nov. 2018. [Online]. Available: https://doi.org/10.1145/3280848

[19] S. Yazdanshenas, M. R. Pirbasti, M. Fazeli, and A. Patooghy, "Coding Last Level STT-RAM Cache for High Endurance and Low Power," *IEEE computer architecture letters*, vol. 13, no. 2, pp. 73–76, 2013.

[20] X. Dong, C. Xu, Y. Xie, and N. P. Jouppi, "NVSim: A Circuit-Level Performance, Energy, and Area Model for Emerging Non-Volatile Memory," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 31, no. 7, pp. 994–1007, 2012.

[21] "The ChampSim simulator," https://github.com/ChampSim.

[22] J. L. Henning, "SPEC CPU2006 Benchmark Descriptions," *ACM SIGARCH Computer Architecture News*, vol. 34, no. 4, pp. 1–17, 2006.

[23] S. Mittal, "A Survey of Soft-Error Mitigation Techniques for Non-Volatile Memories," *Computers*, vol. 6, no. 1, p. 8, 2017.

[24] "ISCA 2017 Cache Replacement Championship," http://crc2.ece.tamu.edu.

[25] D. Burger and T. M. Austin, "The simplescalar tool set, version 2.0," *SIGARCH Comput. Archit. News*, vol. 25, no. 3, p. 13–25, Jun. 1997. [Online]. Available: https://doi.org/10.1145/268806.268810

[26] M. T. Teimoori, M. A. Hanif, A. Ejlali, and M. Shafique, "Adam: Adaptive approximation management for the non-volatile memory hierarchies," in *2018 Design, Automation Test in Europe Conference Exhibition (DATE)*, 2018, pp. 785–790.

[27] N. Binkert, B. Beckmann, G. Black, S. K. Reinhardt, A. Saidi, A. Basu, J. Hestness, D. R. Hower, T. Krishna, S. Sardashti, R. Sen, K. Sewell, M. Shoaib, N. Vaish, M. D. Hill, and D. A. Wood, "The gem5 simulator," *SIGARCH Comput. Archit. News*, vol. 39, no. 2, p. 1–7, Aug. 2011. [Online]. Available: https://doi.org/10.1145/2024716.2024718

[28] T. Delobelle, P.-Y. Péneau, S. Senni, F. Bruguier, A. Gamatié, G. Sassatelli, and L. Torres, "Flot automatique d'évaluation pour l'exploration d'architectures à base de mémoires non volatiles," in *ComPAS: Conférence en Parallélisme, Architecture et Système*, Lorient, France, Jul. 2016. [Online]. Available: https://hal-lirmm.ccsd.cnrs.fr/lirmm-01345975

[29] T. Delobelle, P.-Y. Péneau, A. Gamatié, F. Bruguier, S. Senni, G. Sassatelli, and L. Torres, "MAGPIE: System-level Evaluation of Manycore Systems with Emerging Memory Technologies," in *EMS: Emerging Memory Solutions*, Lausanne, Switzerland, Mar. 2017. [Online]. Available: https://hal-lirmm.ccsd.cnrs.fr/lirmm-01467328

[30] A. Butko, A. Gamatié, G. Sassatelli, L. Torres, and M. Robert, "Design Exploration for next Generation High-Performance Manycore On-chip Systems: Application to big.LITTLE Architectures," in *ISVLSI: International Symposium on Very Large Scale Integration*. Montpellier, France: IEEE, Jul. 2015, pp. 551–556. [Online]. Available: https://hal-lirmm.ccsd.cnrs.fr/lirmm-01255927

[31] A. Butko, R. Garibotti, L. Ost, V. Lapotre, A. Gamatié, G. Sassatelli, and C. Adeniyi-Jones, "A trace-driven approach for fast and accurate simulation of manycore architectures," in *The 20th Asia and South Pacific Design Automation Conference, ASP-DAC 2015, Chiba, Japan, January 19-22, 2015*. IEEE, 2015, pp. 707–712. [Online]. Available: https://doi.org/10.1109/ASPDAC.2015.7059093

[32] A. Nocua, F. Bruguier, G. Sassatelli, and A. Gamatié, "Elasticsimmate: A fast and accurate gem5 trace-driven simulator for multicore systems," in *12th International Symposium on Reconfigurable Communication-centric Systems-on-Chip, ReCoSoC 2017, Madrid, Spain, July 12-14, 2017*. IEEE, 2017, pp. 1–8. [Online]. Available: https://doi.org/10.1109/ReCoSoC.2017.8016146

[33] J. Li, P. Ndai, A. Goel, S. Salahuddin, and K. Roy, "Design paradigm for robust Spin-Torque Transfer Magnetic RAM (STT MRAM) from circuit/architecture perspective," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 18, no. 12, pp. 1710–1723, 2009.

[34] "DDR3-Micron MT41K512M8DA-125 datasheet."

[35] D. M. Mathew, É. F. Zulian, S. Kannoth, M. Jung, C. Weis, and N. Wehn, "A Bank-Wise DRAM Power Model for System Simulations," in *Proceedings of the 9th Workshop on Rapid Simulation and Performance Evaluation: Methods and Tools*. ACM, 2017, p. 5.

[36] A. Jaleel, K. B. Theobald, S. C. Steely Jr, and J. Emer, "High Performance Cache Replacement Using Re-Reference Interval Prediction (RRIP)," in *ACM SIGARCH Computer Architecture News*, vol. 38, no. 3. ACM, 2010, pp. 60–71.

[37] P.-Y. Péneau, D. Novo, F. Bruguier, G. Sassatelli, and A. Gamatié, "Performance and Energy Assessment of Last-Level Cache Replacement Policies," in *International Conference on Embedded & Distributed Systems (EDiS)*. IEEE, 2017, pp. 1–6.

[38] A. Jain and C. Lin, "Back to the Future: Leveraging Belady's Algorithm for Improved Cache Replacement," in *ACM/IEEE 43rd Annual International Symposium on Computer Architecture (ISCA),*. IEEE, 2016, pp. 78–89.

[39] L. A. Belady, "A Study of Replacement Algorithms for a Virtual-Storage Computer," *IBM Systems journal*, vol. 5, no. 2, pp. 78–101, 1966.

[40] R. L. Mattson, J. Gecsei, D. R. Slutz, and I. L. Traiger, "Evaluation Techniques for Storage Hierarchies," *IBM Systems journal*, vol. 9, no. 2, pp. 78–117, 1970.

[41] M. Dayarathna, Y. Wen, and R. Fan, "Data Center Energy Consumption Modeling: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 732–794, 2015.