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

To cite this version:


HAL Id: lirmm-03341602
https://hal-lirmm.ccsd.cnrs.fr/lirmm-03341602

Preprint submitted on 11 Sep 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers. L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
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 paper 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 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-
TABLE 1: NVM-based design focuses in selected studies

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

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:

\[ A_{sram} \geq A_{stt}, \]  

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 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.
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.

<table>
<thead>
<tr>
<th>Core 0</th>
<th>Core 1</th>
<th>Core 2</th>
<th>Core 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>mix1</td>
<td>gobmk</td>
<td>libquantum</td>
<td>perlbench</td>
</tr>
<tr>
<td>mix2</td>
<td>astar</td>
<td>bwaves</td>
<td>lbm</td>
</tr>
<tr>
<td>mix3</td>
<td>cactusADM</td>
<td>lbm</td>
<td>micl</td>
</tr>
<tr>
<td>mix4</td>
<td>bwaves</td>
<td>lbm</td>
<td>sphinx3</td>
</tr>
<tr>
<td>mix5</td>
<td>astar</td>
<td>cactusADM</td>
<td>GensFDTD</td>
</tr>
<tr>
<td>mix6</td>
<td>cactusADM</td>
<td>GensFDTD</td>
<td>gobmk</td>
</tr>
<tr>
<td>mix7</td>
<td>astar</td>
<td>cactusADM</td>
<td>lesie3d</td>
</tr>
<tr>
<td>mix8</td>
<td>bwaves</td>
<td>libquantum</td>
<td>perlbench</td>
</tr>
<tr>
<td>mix9</td>
<td>cactusADM</td>
<td>gobmk</td>
<td>micl</td>
</tr>
<tr>
<td>mix10</td>
<td>bwaves2</td>
<td>gobmk</td>
<td>lbm</td>
</tr>
<tr>
<td>mix11</td>
<td>astar</td>
<td>gobmk</td>
<td>micl</td>
</tr>
<tr>
<td>mix12</td>
<td>gobmk</td>
<td>lesie3d</td>
<td>libquantum</td>
</tr>
<tr>
<td>mix13</td>
<td>bwaves</td>
<td>bzip2</td>
<td>gobmk</td>
</tr>
<tr>
<td>mix14</td>
<td>gobmk</td>
<td>lbm</td>
<td>lesie3d</td>
</tr>
<tr>
<td>mix15</td>
<td>cactusADM</td>
<td>gobmk</td>
<td>micl</td>
</tr>
<tr>
<td>mix16</td>
<td>bwaves</td>
<td>bzip2</td>
<td>gobmk</td>
</tr>
<tr>
<td>mix17</td>
<td>astar</td>
<td>bzip2</td>
<td>lesie3d</td>
</tr>
<tr>
<td>mix18</td>
<td>gobmk</td>
<td>libquantum</td>
<td>wrf</td>
</tr>
<tr>
<td>mix19</td>
<td>gobmk</td>
<td>lbm</td>
<td>micl</td>
</tr>
<tr>
<td>mix20</td>
<td>micl</td>
<td>perlbench</td>
<td>wrf</td>
</tr>
</tbody>
</table>

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:

$$L_T = L_I + L_C,$$

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.

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:

$$E^i = R^i \times R^i + W^i \times W^i + T \times P_{\text{leak}}^i,$$

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 $P_{\text{leak}}^i$ 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 $16 \times 65536$ 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]:

$$E_m = E_a + E_b + E_c,$$

$$E_a = R \times R D + W \times W R,$$

$$E_b = (R_M + W_M) \times (P_{\text{REF}} + A C T),$$

$$E_c = (T/T_{\text{REF}}) \times R E F + T \times A C T_{B G},$$

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; $P_{\text{REF}}$ and $A C T$ are respectively the cost of page precharge and activation; $T$ is the execution time, $T_{\text{REF}}$ is the self-refresh frequency; $REF$ is the cost of a refresh and $ACT_{B G}$ is the active background energy consumption. We do not consider low-power mode, which reduces $ACT_{B G}$. The full configuration is summarized in Table 5.
### 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 36% has been previously observed.

Figure 5 depicts the Pareto distribution of execution time and energy consumption when considering the entire
memory hierarchy. As in Figure 3, results are normalized to the SRAM reference. Execution time remains identical, only the energy consumption varies. For C\textsubscript{area} designs, trends remain identical and we observe a small impact on the energy consumption. Compared to LLC only perspective, B\textsubscript{stat}_lru\_C\textsubscript{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\textsubscript{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\textsubscript{stat}_lru\_C\textsubscript{lat} is the only configuration of STT-MRAM C\textsubscript{lat} with LRU that outperforms S\textsubscript{sram}_lru for both energy and execution time, leading to a best EDP result. This observation is in contrast with Figure 3, where B\textsubscript{stat}_lru\_C\textsubscript{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\textsubscript{lat} and C\textsubscript{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\textsubscript{lat} and between 71% and 75% for C\textsubscript{area}. This gap comes from the execution time overhead induced by C\textsubscript{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\textsubscript{area} configurations.

Hence, the low overhead in terms of execution time for C\textsubscript{lat} designs, especially B\textsubscript{stat}_lru\_C\textsubscript{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\textsubscript{lat} and C\textsubscript{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\textsubscript{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\textsubscript{area} designs and one C\textsubscript{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\textsubscript{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
Thus, it fits into the initial area budget without Hawkeye. Results also show that the impact already-positive use case since EDP is improved by at least 40% 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:

\[ E_{total} = \alpha_{Core} + \beta_{Cache} + \gamma_{Dram} \]

with \( \alpha \gg \beta \) \( (8) \) and \( \beta \ll \gamma \),

where \( E_{total} \) is the total energy consumption of a system and \( \alpha, \beta \) and \( \gamma \) 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 \( \beta \) without considering \( \gamma \). 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 derived 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 a 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


[34] “DDR3-Micron MT41K512M8DA-125 datasheet.”


