NV-SP: A New High Performance and Low Energy NVM-Based Scratch Pad

Ameer Shalabi[1], Kolin Paul[2], Tara Ghasempouri[1], Jaan Raik[1]
[1] Department of Computer Systems, Tallinn University of Technology, Estonia
[2] Department of Computer Science & Engg. Indian Institute of Technology Delhi, India
email:{Ameer.Shalabi, Tara.Ghasempouri, Jaan.Raik}@taltech.ee, kolin@cse.iitd.ac.in

***Abstract*—Non-Volatile Memory technologies are rising as a candidate for a universal memory. NVMs offer solutions for the high power consumption that contemporary memory suffers from. Hence, we propose augmenting the traditional SRAM cache with an additional NVM device instead of entirely replacing SRAM with NVM. The L1 instruction-cache is augmented with a Non-Volatile Scratch-Pad, coined NV-SP, that stores instructions causing the highest number of misses. Experiments were evaluated for performance and energy of the SRAM I-cache and the NV-SP when implemented using Magnetic RAM and Phase-Changing RAM technologies. Results have shown that MRAM NV-SP had effectively improved the performance of the I-cache.**

I. Introduction

Current computer systems employ different types of semiconductor-based memory to store information. From the volatile, low capacity, high speed registers and caches on-chip to the large capacity, slow, peripheral memory devices. However, there are two key issues in the current semiconductor-based memory: Firstly, the rapid increase in density causes larger amounts of power consumption. Secondly, the speed gap between different levels of the memory hierarchy causes bottlenecks at each level.

To combat the first issue, designers were forced to implement circuit design techniques, such as power and clock gating [1], to reduce standby mode power consumption, which can cause performance degradation. As for the second issue, on-chip, specialized caches, known as the First Level (L1) Caches, were introduced to the system. Such caches stored data closer to the CPU, thus circumventing the speed gap between the fast CPU and slow peripheral memory. Furthermore, for embedded System-on-Chip (SoC), the issue of speed gap was further combated with introducing a Scratch Pad memory to store smaller amounts of information that are constantly needed [2]. Scratch Pad memories, unlike caches, hold information permanently and are manually configured. Scratch Pads do not have eviction policies, thus the information stored in them is never changed automatically and can be configured depending on the execution nature of the workload.

In response to the persistence of these issues, the research community has been paying more attention to Non-Volatile Memory (NVM) technologies and their potential to offer a universal memory device. NVMs offer low-power and high-density storage. Although NVMs are high cost and are slower than SRAM, they were seen as the potential solution for many issues arising in nanoscale electronics.

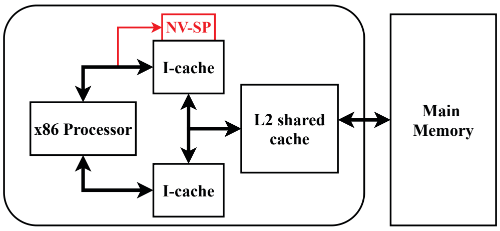

Fig. 1: Architecture under evaluation with the NV-SP

Several new technologies have been introduced based on NVM. For example, Spin Transfer Torque MRAM (STT-MRAM) [3], an implementation of Magnetic RAM (MRAM) used in this work, has been shown to realize a fully working memory cell that is used to store information in MRAM arrays at low energy cost [3] [4]. Furthermore, [5] and [6] showed that phase-changing material can be used to create a low-energy Phase-Changing RAM (PCRAM) cell with high performance and low energy cost.

While PCRAM is more directly towards neuromorphic systems [6], Senni et al. [7] tested potential applications for the different MRAM implementations in a traditional von-Neumann memory system. Namely, STT-MRAM and TAS-MRAM [8] were evaluated for performance and energy consumption for L2 cache as well as evaluating the STT-RAM for the L1 caches in different scenarios, but this evaluation did not include comparing STT-MRAM with other NVM technologies. Furthermore, the evaluation in [7] was an implementation of a traditional von-Neumann processor system that did not fully take advantage of the potential low energy cost that STT-MRAM offers.

The evaluation framework used in this work was inspired by the one used in [7]. However, the framework developed for this work further incorporates a stand-alone NVM-based device alongside the traditional SRAM caches instead of replacing SRAM with NVM. This can shed light on the effectiveness of using NVMs to create memory systems that fall outside the traditional von-Neumann architecture. Fig.1 shows the conceptualization of such NVM-based device.

The focus of this work will be directed towards the L1 caches of the cache system: more specifically, the on-chip *Instruction cache*. SRAM I-caches suffer from high miss latency and require to be constantly powered-on to retain information.

2159-3477/20/$31.00 ©2020 IEEE
DOI 10.1109/ISVLSI49217.2020.00020

I-caches have a higher frequency of Read accesses compared to their Write accesses frequency [7]. Thus, misses caused by Read accesses have higher effect on the overall performance of the I-cache. To reduce this effect, this work sets forth the following contributions:

- A new NVM-based Scratch Pad, coined Non-Volatile Scratch Pad (NV-SP) is introduced to the memory system.
- A framework is developed for evaluating the NV-SP's potential in improving the performance and reducing the access energy and leakage energy of the I-cache.
- New formulas are created to evaluate performance, access energy and leakage energy of the SRAM I-cache when augmented with a configurable NV-SP implemented using PCRAM or MRAM.
- Specialized simulators are used to obtain performance measures and energy estimations of the I-cache before and after the introduction of the the NV-SP.
- Applications are profiled where the top *k* instructions which suffer the largest number of cache misses are moved into the NV-SP.

This paper is organized as follows: Section II presents the methodology used in this work. Section III discusses the experimental results obtained during the experiment. Finally, in Section IV, conclusions from this work are presented.

II. Methodology

The main contribution of this paper is developing a framework to evaluate the potential use of NVM technologies for improving the performance and reducing the access energy and leakage energy of low-level memory hierarchy. This is done by extracting results from the simulators and evaluating the performance of the I-cache before and after incorporating the NV-SP to it. This is also done for I-cache access energy and leakage energy. This section explains the proposed framework and discusses the decisions made during its development.

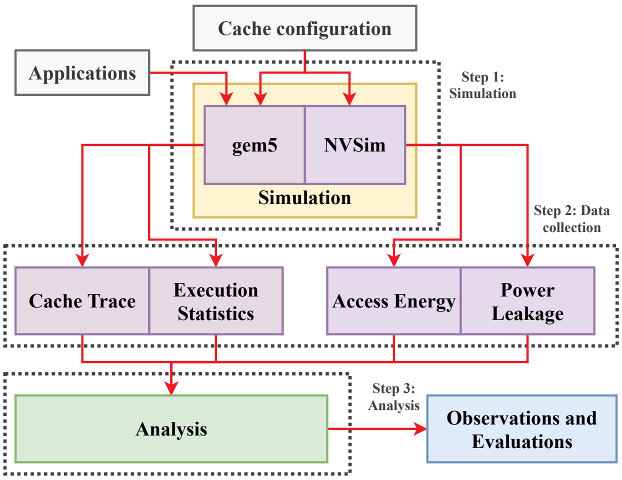

Fig. 2: Evaluation framework steps

As shown in Fig.2, the evaluation framework is comprised of three steps. During step 1, two cache configurations are simulated. The first when the I-cache is not augmented with the NV-SP. The second configuration is when the NV-SP is introduced to the I-cache. Once both the cache configurations are simulated, data is collected from the simulators in step 2. Analysis is preformed on the collected data in step 3. Finally, the observations and evaluation of the simulators output in step 2 are reported.

A. Step 1 : Simulation

In this step, two simulators are used to produce performance measures and energy estimations. The first simulator is gem5 [9]. It is widely used by the research community to simulate processor architectures. It offers a highly configurable simulation framework with diverse CPU models and multiple Instruction Set Architectures (ISAs). It is used in this experiment because it offers flexibility when defining the specifications of the different levels of the cache system. The second simulator is NVSim. NVSim is a circuit-level model for NVM performance, energy, and area estimations, which supports various NVM technologies, including STT-RAM and PCRAM. NVSim also supports volatile memory technologies such as SRAM and DRAM [10]. Depending on a given configuration, NVSim estimates the access latency, access energy and leakage energy of the NVM technology chip. This helps in finding the optimal NVM chip design space for achieving the best performance, area, or energy. NVSim is used in this work to find the optimal performance and energy estimations of STT-MRAM and PCRAM memory cells as well as the performance and energy estimations of a 4 kB SRAM cache. Using those simulators, the configuration under evaluation is created as follows:

Component	Configuration
Processor	Single-Core, 1 GHz, 32-bit, x86
L1 I/D-cache	Private, 4kB, 2-way associative
L2 cache	Shared, 64kB, 4-way associative
Main memory	DRAM, DDR3, 100-cycle latency

TABLE I: System components' configuration

1) The Base Architecture: In this paper, gem5 is used to simulate an x86 processor with an I-cache, a D-cache, an L2 cache, and a Main memory. Table I shows the system components' configuration used in this work. The objective of selecting this architecture is to focus on the direct interaction between the processor and the L1 I-cache before and after the incorporation of the NV-SP. Using a Single-Core will give insight into the direct impact of adding a scratch pad to the L1 memory hierarchy.

2) The NV-SP: Three sizes of 1, 2, and 4-kiloBytes are used to evaluate the impact of the NV-SP on the performance and energy of the I-cache. Since this architecture executes 32-bit instructions, each instruction will be at 4 Bytes of size. Each of the NV-SP sizes of 1 kB, 2 kB, and 4 kB will store 256, 512, and 1024 instructions respectively. These sizes were used to accommodate the small size of the applications

	Performance	Access Energy	leakage energy
Before NV-SP	**(1)** $(Ic_{ML} * \#m) + (Ic_{HL} * \#h)$	**(2)** $(SRAM_{RE} * \#h) + (SRAM_{WE} * \#m)$	**(3)** $SRAM_{PL} * T$
After NV-SP	**(4)** $(Ic_{ML} * (\#m - \#m_{NV})) + (Ic_{HL} * \#h)$	**(5)** $(SRAM_{RE} * \#h) + (SRAM_{WE} * (\#m - \#m_{NV})$	**(6)** $SRAM_{PL} * T_n$
	(7) $NV_{RL} * \#m_{NV} + NV_{WL} * \#NV_i$	**(8)** $NV_{RE} * \#m_{NV} + NV_{WE} * \#NV_i$	**(9)** $NV_{PL} * T_{NVa}$

TABLE II: Formulas used in Step 3

simulated as well as to give insight on the impact of different sizes of the NV-SP on the final results. The NV-SP requires write and read circuits adding additional latency and energy leakage to the system. Compared to SRAM, read and write circuits of the NV-SP are significantly smaller. In addition, the operation conditions of the NV-SP technologies are modelled on configurations and operation presented in [10]. The NV-SP is incorporated into the system as illustrated in Fig.1.

3) Applications: Table III shows the applications chosen for this experiment. These applications are chosen based on their simple implementation and small size. This allowed any modifications needed to accommodate the constraints set forth by the architecture's 32-bit compatibility. The first type of applications is a collection of cryptographic algorithms based on an open source implementation of cryptography algorithms [11]. The second type of applications used in this work is a collection of non-cryptographic algorithms. These algorithms were chosen based on their computational nature i.e., JPEG encode/decode [12] for stream-like behavior. The bzip2 [13] for compression and decompression algorithms. specrand [14] for random number generation.

Once the simulators are configured and ran, preliminary data is collected in order to perform the analysis and produce final results.

B. Step 2 : Data Collection

For this work, data is collected from both the simulators mentioned previously as follows:

1) gem5 Data: gem5 produces statistics and cache traces that show information regarding the propagation of data within the cache system providing measurements of the memory's accesses, misses, hits, and report their respective rates. gem5 preliminary results are obtained from both statistical and cache trace files produced by the simulator. For the purposes of this work, information regarding the number of accesses, number of hits, and number of misses of the I-cache and information regarding the duration of simulation are extracted.

2) NVSim Data: NVSim estimations are used to evaluate the performance and energy of a Scratch Pad memory when designed with each of the NVM technologies. The preliminary results from NVSim are obtained from the direct output of the simulator. The output of the simulator contains estimations regarding the performance, access energy, and leakage energy of the NVM technologies under evaluation.

C. Step 3 : Analysis

In this section, nine formulas are introduced to evaluate the effectiveness of the proposed NV-SP device. Three formulas, explained in *C.1*, are used to evaluate the I-cache before incorporating the NV-SP. Formulas in *C.2* are used to evaluate both the I-cache and NV-SP after incorporating the NV-SP.

C.1) Performance and Energy Evaluation before NV-SP: At this step, to evaluate the performance, access energy, and leakage energy of the I-cache, the following equations from Table II were used:

- Formula **(1)**: Using gem5 statistics of the number misses $(\#m)$ and hits $(\#h)$ and using NVSim miss latency $(SRAM_{ML})$ and hit latency $(SRAM_{HL})$ estimations, the total latency (performance) caused by all the misses is calculated.
- Formula **(2)**: Using NVSim SRAM Read $(SRAM_{RE})$ and Write $(SRAM_{WE})$ access energy estimations, the total access energy of the I-cache is calculated.
- Formula **(3)**: Using NVSim leakage energy $(SRAM_{PL})$ estimations, the total leakage energy of the I-cache during the execution time (T) can be calculated.

Measurements from **(1)**, **(2)**, and **(3)** are used as the SRAM baseline for comparing the performance, access energy, and leakage energy of I-cache before and after the incorporation of the NV-SP.

C.2) Performance and Energy Evaluation after NV-SP: At this step, to evaluate the performance, access energy, and leakage energy of the I-cache, additional parameters must be accounted for since the latency of the Read and Write operations differ depending on the technology used in the NV-SP and the size of the NV-SP under evaluation. The following equations from Table II were used:

- Formula **(4)**: The number of instructions causing the highest number of misses $(\#NV_i)$ are moved to the NV-SP. By removing the misses caused by the instructions stored in the NV-SP $(\#m_{NV})$, a new total miss latency (performance) can be produced for the I-cache.
- Formula **(5)**: Similarly, since $(\#NV_i)$ are moved to the NV-SP, access energy needed to mitigate $(\#m_{NV})$ is no longer happening in the I-cache. By removing $(\#m_{NV})$ from $(\#m)$, a new total access energy is calculated for the I-cache.
- Formula **(6)**: The new execution time (T_n) is used to calculate the total leakage energy of the I-cache.

When $\#NV_i$ are moved to the NV-SP, $\#m_{NV}$ becomes the number of Read accesses to the NV-SP, since the NV-SP is only accessed when these misses occur. The number of NV-SP Write accesses is equal to $\#NV_i$, since these instructions are only written once and never evicted. The NV-SP is evaluated using the following formulas from Table II:

- Formula **(7)**: Using NVSim Read latency (NV_{RL}) and Write latency (NV_{WL}) estimations for the NVM used in

	Cryptographic applications						Non-cryptographic Applications				
	AES	DES	ARCFOUR	Twofish	Blowfish	ROT13	Djpeg	Cjpeg	bzip2-d	bzip2-c	specrand
Misses	18258	2764	1852	5594	1953	1676	6460	6934	257412	637481	1489776
Miss Rate	1.60%	0.22%	0.77%	1.33%	0.11%	0.86%	1.54%	0.40%	0.26%	0.65%	3.34%

TABLE III: Baseline I-cache simulation results for experiment applications before incorporating the NV-SP. LRU replacement policy is used.

the NV-SP, the total latency (performance) of the NV-SP is calculated.

- Formula **(8)**: Using NVSim Read Energy (NV_{RE}) and Write Energy (NV_{WE}) estimations for the NVM used in the NV-SP, the total access Energy of the NV-SP is calculated.
- Formula **(9)**: Using NVSim leakage energy (NV_{PL}) estimations for the NVM used in the NV-SP, and since the NV-SP is only leaking power when it is accessed (T_{NVa}), the leakage energy of the NV-SP is calculated.

By summing the results from **(4)** with **(7)**, and results from **(5)** with **(8)**, and results from **(6)** with **(9)**, new evaluations of performance, access energy, and leakage energy are produced respectively. These measurements are compared to the measurements produced in *C.1*.

D. Observations and Evaluations

The results from gem5 and NVSim are collected in Step 2 and are used to produce the final results.

1) gem5 Simulation Results: Table III shows the number misses and miss rate of the I-cache for the applications executed during simulation. It shows that the miss rates of the applications are fairly high. This is due to the small size of the I-cache. It is notable that *specrand* had the highest miss count and miss rate compared to any other application. This is largely due to the nature of the application as it executes a large variety of instructions at different times, causing a large number of misses to occur.

Size	**Estimation**	**SRAM**	**PCRAM**	**MRAM**
1 kB	Read Latency	—	0.153 ns	1.582 ns
	Write Latency	—	150.091 ns	10.124 ns
	Read Enegy	—	0.001 nJ	0.008 nJ
	Write Energy	—	3.241 nJ	0.032 nJ
	leakage energy /s	—	8.303 mW	0.762 mW
2 kB	Read Latency	—	0.159 ns	1.584 ns
	Write Latency	—	150.096 ns	10.125 ns
	Read Enegy	—	0.001 nJ	0.008 nJ
	Write Energy	—	3.241 nJ	0.032 nJ
	leakage energy /s	—	16.56 mW	2.892 mW
4 kB	Read Latency	0.324 ns	0.267 ns	1.766 ns
	Write Latency	0.227 ns	150.122 ns	10.206 ns
	Read Enegy	0.004 nJ	0.004 nJ	0.027 nJ
	Write Energy	0.003 nJ	10.533 nJ	0.106 nJ
	leakage energy /s	20.189 mW	31.771 mW	5.377 mW

TABLE IV: NVSim estimations for SRAM, PCRAM, and MRAM designs in [10]. 1kB and 2kB SRAM were not used in this work.

2) NVSim Estimation Results: Estimations for access latency, access energy, and leakage energy per second in Table IV were obtained for 1kB, 2kB, and 4kB PCRAM and MRAM to be used in measuring the access latency, access energy, and leakage energy of the NV-SP. However, since SRAM is only used to estimate the performance and energy of the I-cache, estimations for SRAM were only obtained at 4kB.

NVSim Access Latency Results: From Table IV, the following observations can be made regarding the access latency of the NVM technologies:

- PCRAM clearly has the advantage of Read latency over MRAM. Since PCRAM Read operation is similar to electrical discharge, the information stored in PCRAM requires little additional circuitry to be released from it.
- PCRAM is overwhelmingly outperformed when it comes to Write latency. Write operation requires less time for MRAM memory compared to PCRAM due to the nature of the Write operations for each of the technologies.

NVSim Access Energy Results: From Table IV, the following observations can be made regarding the access energy of the NVM technologies:

- PCRAM requires much less Read energy compared to MRAM. This is largely due to the complexity of the circuitry used for the Read operation in both cells.
- PCRAM requires higher access energy compared to MRAM. This is becasue PCRAM requires high voltage pulses to be applied to change the material state in the cell [6].
- MRAM maintains lower Write energy that is 100 times lower of PCRAM Write energy at all sizes, yet MRAM's low Write access energy is still high compared to SRAM Write access energy at 4kB.

NVSim leakage energy Results: As for leakage energy, Table IV shows a significant increase of leakage energy in PCRAM as the size increases. This is attributed to the long duration and higher voltage pulses needed for the PCRAM to change the state of its material (read and write operations). However, SRAM needs to be constantly operating. This means that the duration of time in which leakage energy happens in the SRAM is the total execution time. Due to their non-volatile nature, MRAM and PCRAM only need to operate when they are being accessed. This means that the leakage energy of PCRAM may be higher than that of SRAM, but since PCRAM is non-volatile, it does need to be enabled for the entire duration of execution.

III. Experimental Results

1) Performance: Fig.3 shows the execution time of all the applications when using MRAM technology in the NV-SP implementation. Fig.3, along with Table III, draw a correlation between improvement of execution time and the miss rate of an application before incorporating the NV-SP. Applications with highest miss rates, *specrand* and *AES* with 3.34% and 1.60% miss rates respectively, experienced execution time improvement up to 22.35% and 16.49% respectively. Nonetheless, after introducing the MRAM NV-SP to the system, all the applications showed performance improvement for all the MRAM NV-SP sizes compared to the SRAM baseline.

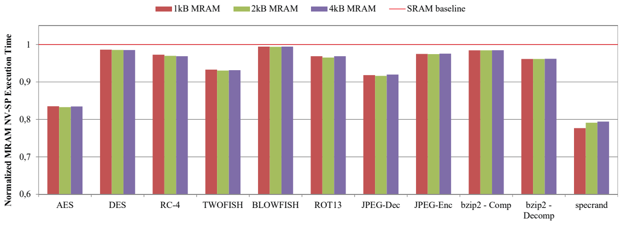

Fig. 3: MRAM performance

Higher miss rate meant that more time is being spent on miss mitigation. When using MRAM, moving instructions with the highest number of misses into the NV-SP reduced the total number of miss mitigation cycles, causing performance improvement.

Fig.4 shows the execution time of all the applications when using PCRAM technology in the NV-SP implementation. Unlike Fig.3, Fig.4 draws a correlation between the performance improvement/degradation and the size of the NV-SP as well as a correlation between performance degradation and lower miss rate. Similar to MRAM NV-SP, applications with higher miss rates generally show higher performance improvement after incorporating the PCRAM NV-SP. However, applications with lower miss rate show performance degradation after incorporating 2kB and 4kB PCRAM NV-SP.

Applications with high miss rates such as *specrand* show performance improvement of 23% for all PCRAM NV-SP sizes. Similarly, all the non-cryptographic applications show performance improvement. Furthermore, applications with low number of misses and moderate miss rate such as *RC-4* experienced performance degradation of up to 10.19% for 4 kB. These results show that the size of the PCRAM NV-SP has an effect on the performance. Although PCRAM has a very high Write latency, the majority of accesses to the PCRAM NV-SP are Read accesses which mitigates the high Write latency. For applications with a low miss rate, the impact of the Write access latency increases leading to performance degradation.

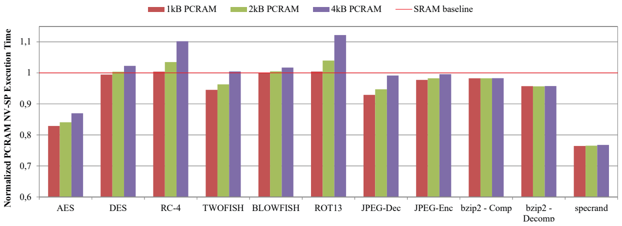

Fig. 4: PCRAM performance

	MRAM			PCRAM		
	1kB	2kB	4kB	1kB	2kB	4kB
AES	97.28%	97.25%	97.23%	97.28%	97.26%	97.33%
DES	99.66%	99.64%	99.62%	99.67%	99.65%	99.68%
ARC4	98.95%	98.85%	98.74%	99.03%	99.00%	99.71%
TWOFISH	97.80%	97.73%	97.68%	97.82%	97.78%	98.03%
BLOWFISH	99.82%	99.81%	99.80%	99.83%	99.81%	99.83%
ROT13	98.68%	98.56%	98.52%	98.80%	98.78%	99.86%
JPEG-Dec	97.53%	97.54%	97.59%	97.89%	97.59%	97.62%
JPEG-Enc	99.59%	99.59%	99.60%	99.61%	99.59%	99.60%
bzip2-c	99.54%	99.54%	99.54%	99.54%	99.54%	99.54%
bzip2-d	98.86%	99.25%	99.25%	98.86%	99.25%	99.25%
specrand	94.27%	94.27%	94.27%	94.27%	94.27%	94.27%

TABLE V: MRAM and PCRAM access energy. SRAM baseline is 100%

2) Access Energy: Table V shows the total access energy of all the applications when using MRAM and PCRAM technology in the NV-SP implementation. While Table V shows that using the MRAM NV-SP implementation did not have a huge impact on the total access energy after incorporating the NV-SP, a slight improvement can be noted for *specrand* and *AES*. This shows a correlation between the high miss rates of *specrand* and *AES* and slightly lower access energy. In this case, MRAM NV-SP has a much lower number of accesses compared to the I-cache. A lower number of accesses is needed for retrieving an instruction when it is stored in the MRAM NV-SP compered to the number of accesses needed to mitigate an I-cache miss. Nonetheless, this improvement on the number of accesses is overshadowed by the relatively high access energy of the MRAM compared to SRAM as shown in Table IV.

Furthermore, the access energy of the MRAM NV-SP has a very little effect on the total access energy since the majority of accesses are still happening in the I-cache. This is why the total access energy did not change dramatically. Similarly, PCRAM did not have a huge affect on total access energy after incorporating the PCRAM NV-SP into the system. This can be attributed to the overshadowing of the very high Write access energy of PCRAM. A slight decrease in total access energy can be observed for both *AES* and *specrand* since they require much fewer accesses for the same instructions stored in the PCRAM NV-SP compared to when those instructions are stored in the I-cache.

3) leakage energy: Introducing the MRAM NV-SP to the I-cache has reduced the leakage energy for all the applications. Fig.5 shows the combined leakage energy measurements for I-cache and MRAM NV-SP. It shows that incorporating the MRAM NV-SP has reduced the leakage energy of the I-cache for all applications.

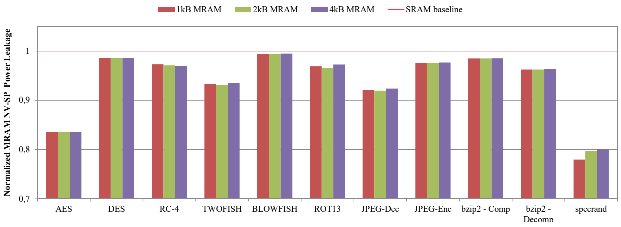

Fig. 5: MRAM leakage energy

Since MRAM has low leakage energy and, due to its non-volatile nature, it is only active when it is accessed, the total leakage energy of the MRAM is very low. Since the I-cache is constantly operating, any reduction in execution time reduces the time in which the I-cache is operating, resulting in lower leakage energy. Since MRAM has improved the performance of all the applications, the I-cache's leakage energy for all the applications is subsequently reduced.

Fig.6 shows the combined leakage energy measurements for I-cache and PCRAM NV-SP. It shows that incorporating the PCRAM NV-SP has reduced leakage energy for applications with high miss rate such as *AES* and *specrand*, while drastically increasing it for applications with low miss rate such as *RC-4* and *ROT13*. This draws a relationship between the leakage energy of the I-cache and performance improvement when introducing the PCRAM NV-SP. This relationship is seen clearly for all the cryptographic applications. *Twofish* cipher had performance improvement at sizes 1 kB and 2 kB causing the leakage energy at those sized to be reduced. However, *Twofish* experienced performance degradation at 4 kB, which in turns increased the leakage energy at that size.

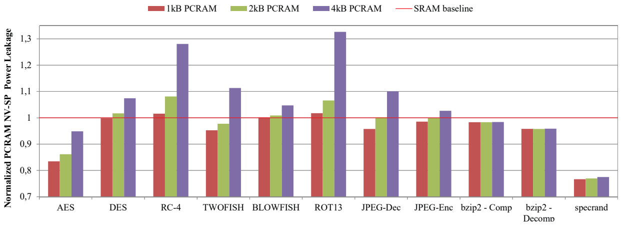

Fig. 6: PCRAM leakage energy

In the case of *JPEG-Decode*, it experiences a relatively high miss rate. When instructions were stored in the NV-SP, accesses to it increased resulting in more enabled time for the NV-SP. In fact, in the case of *JPEG-Decode*, the PCRAM NV-SP contributed 10% of the total combined leakage energy of the I-cache and NV-SP. Furthermore, the major factor in increasing leakage energy for *JPEG-Decode* was the low performance improvement of 0.44% with 4 kB. This when combined with the high leakage energy of the PCRAM at that size, are what caused the increase of the leakage energy to be above the SRAM baseline. Similar effect of performance degradation on the increase of leakage energy can be seen for *JPEG-Encode* as well.

IV. Conclusion

In this work, a Non-Volatile Scratch Pad (NV-SP) was introduced to the cache system. A framework was developed for evaluating the NV-SP's potential effect on the performance, access energy, and leakage energy of the I-cache. This work showed that using NVM technologies such as PCRAM and MRAM in the form of an NV-SP can greatly impact the performance of the I-cache. On one hand, MRAM improved the performance of all the applications, reduced the access energy of the system, and greatly reduced the leakage energy of the system. PCRAM, on the other hand, had caused performance degradation at larger sizes for applications with low miss rate and low number of misses. Both MRAM and PCRAM helped reduce the overall access energy of the system. While MRAM improved the overall leakage energy of the system, PCRAM showed increase of leakage energy due to the previously mentioned performance degradation. This work showed that an NVM-based Scratch Pad can be used to improve performance and reduce access energy and leakage energy of the cache system. MRAM proved to be a better candidate to provide improvements to the cache system compared to PCRAM.

References

[1] N. Srinivasan, Shalakha D., Sivaranjani D., B. B. T. Sundari, S. Sri Lakshmi G., and N. S. Prakash, "Power Reduction by Clock Gating Technique," *Procedia Technology*, vol. 21, pp. 631–635, 2015.

[2] A. K. I. Mendonca, D. P. Volpato, J. L. Güntzel, and L. C. V. Santos, "Mapping data and code into scratchpads from relocatable binaries," in *2009 IEEE Computer Society Annual Symposium on VLSI*, pp. 157–162, May 2009.

[3] A. V. Khvalkovskiy, D. Apalkov, S. Watts, R. Chepulskii, R. S. Beach, A. Ong, X. Tang, A. Driskill-Smith, W. H. Butler, P. B. Visscher, D. Lottis, E. Chen, V. Nikitin, and M. Krounbi, "Erratum: Basic principles of STT-MRAM cell operation in memory arrays (Journal of Physics D: Applied Physics (2013) 46 (074001))," *Journal of Physics D: Applied Physics*, vol. 46, no. 13, 2013.

[4] J.-G. J. Zhu and C. Park, "Magnetic Tunnel Junctions," vol. 9, no. 11, pp. 36–45, 2006.

[5] H. F. Hamann, M. O'Boyle, Y. C. Martin, M. Rooks, and H. K. Wickramasinghe, "Ultra-high-density phase-change storage and memory," *Nature Materials*, vol. 5, no. 5, pp. 383–387, 2006.

[6] G. W. Burr, M. J. BrightSky, A. Sebastian, H. Y. Cheng, J. Y. Wu, S. Kim, N. E. Sosa, N. Papandreou, H. L. Lung, H. Pozidis, E. Eleftheriou, and C. H. Lam, "Recent Progress in Phase-Change Memory Technology," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 6, no. 2, pp. 146–162, 2016.

[7] S. Senni, A. Gamatie, G. Sassatelli, R. M. Brum, B. Mussard, and L. Torres, "Potential Applications Based on NVM Emerging Technologies," *2015 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 1012–1017, 2015.

[8] S. Bandiera and B. Dieny, "Thermally assisted MRAM," *Handbook of Spintronics*, pp. 1065–1100, 2015.

[9] S. Sardashti, K. Sewell, S. K. Reinhardt, A. Basu, D. A. Wood, T. Krishna, J. Hestness, G. Black, B. Beckmann, N. Vaish, D. R. Hower, M. D. Hill, N. Binkert, M. Shoaib, A. Saidi, and R. Sen, "The gem5 simulator," *ACM SIGARCH Computer Architecture News*, vol. 39, no. 2, p. 1, 2011.

[10] X. Dong, C. Xu, N. Jouppi, and Y. Xie, "NVSim: A circuit-level performance, energy, and area model for emerging non-volatile memory," *Emerging Memory Technologies: Design, Architecture, and Applications*, vol. 9781441995, no. 7, pp. 15–50, 2014.

[11] B. Conte, "crypto-algorithms," 2015.

[12] M. Consortium, "Mediabench," 1997.

[13] I. Enthought, "bzip2-1.0.6," 2013.

[14] S. P. E. Corporation, "999.specrand spec cpu2006 benchmark description," 2006.