Exploration of Magnetic RAM Based Memory Hierarchy for Multicore Architecture

Sophiane Senni, Lionel Torres, Gilles Sassatelli, Anastasiia Butko, Bruno Mussard

Exploration of Magnetic RAM based memory hierarchy for multicore architecture

Sophiane Senni[1,2], Lionel Torres[2], Gilles Sassatelli[2]
and Anastasiia Bukto[2]
LIRMM – UMR CNRS 5506 – University of Montpellier 2
Montpellier, France
{name[2]}@lirmm.fr

Sophiane Senni[1,2] and Bruno Mussard
Crocus technology
Rousset, France
{ssenni[1], bmussard}@crocus-technology.com

Abstract— **Today's memory systems mainly integrate SRAM, DRAM and FLASH technologies. SRAM and DRAM are generally used for cache and working memory, while FLASH memory is used for non-volatile storage at low speed. But all are facing to manufacturing constraints in the most advanced node, which compromises further evolution. Besides, with the increasing size of the memory system, a significant portion of the total system power is spent into memories. Magnetic RAM (MRAM) technology is a very attractive alternative offering simultaneously reasonable performance and power consumption efficiency, high density and non-volatility. While MRAM is always under severe investigation to improve manufacturing process, the state of the art shows that this memory technology can be accessed in less than 5ns with a read/write dynamic energy not so far to SRAM dynamic energy. Besides, non-volatility of MRAM can be used for optimizing leakage current thanks to instant on/off policies. This paper demonstrates how current characteristics of MRAM can be used into memory hierarchy of multiprocessor chips (CMPs). The goal is to highlight the interest to use MRAM for cache memory in order to keep overall application performance saving static power.**

Keywords—MRAM, NVM, Memory hierarhy, VLSI, SoC, Embedded Systems

I. Introduction

Because it is the fastest memory technology, SRAM is currently chosen to design the upper level of cache memories in order to reach the best performance, particularly for multiprocessor architecture. Today's SRAM issue decreasing the technology node is the high leakage current. DRAM occupies a lower level of the memory hierarchy as it is slower, but has higher density than SRAM. This technology is also power consuming due to its refresh policy to not lose data stored. Finally, we may find FLASH as the last level of the memory hierarchy, used for its high storage and non-volatility capabilities. To overcome performance and power issues of this multi-core era, some non-volatile memory technologies (NVMs) emerged in the past years. ITRS considered Spin-Transfer Torque MRAM (STT-MRAM), Resistive RAM (RRAM) and Phase-Change RAM (PCRAM) as the most promising candidates to be used in future embedded systems. Table I compares these new memory technologies with current memories. While being non-volatile, MRAM combines good scalability, low leakage and radiation hardness. For a same die footprint, MRAM can be used instead of SRAM to get four to seven times larger memory, which can lead to significant improvement of overall system performance and power consumption. However, as other memory technologies, MRAM has also its drawbacks. The main issues of this technology are latency and dynamic energy, especially for a write operation. Compared to SRAM, MRAM write latency and write energy are around three to ten times higher. But last results at device level from Toshiba [1] on MRAM is very encouraging as it show, for perpendicular STT, an access time of around 4ns with read/write energy per bit comparable to SRAM.

MRAM bit is a Magnetic Tunnel Junction (MTJ) which consists of two ferromagnetic layers separated by a thin insulating barrier. The information is stored as the magnetic orientation of one of the two layers, called the Free Layer (FL). The other layer, called the Reference Layer or Fixed Layer (RF), provides a fixed reference magnetic orientation required for reading and writing. Fig. 1 illustrates a typical STT-MRAM cell consisting of one CMOS access transistor and one MTJ (1T-1MTJ).

In this paper, we explore integration of STT-MRAM into the memory hierarchy of multiprocessor architecture. Both performance and energy are evaluated using a processor architecture simulator and a circuit-level model simulator for NVMs. We will demonstrate that use of STT-MRAM is an attractive alternative to optimize overall system power consumption without lost in performance.

TABLE I. Memory technologies comparison

	Current memory technologies			**Emerging NVM technologies**		
	SRAM	***DRAM***	***Flash***	***STT***	***RRAM***	***PCRAM***
Cell size (F^2)	>100	6-8	4-5	8-20	6-10	6-10
Speed	<10 ns	10-60 ns	1 µs-1 ms	1-10 ns	~10ns	~50 ns
Static Power	Yes	Yes	No	No	No	No
Endurance	-	-	10^4	10^{15}	10^5	10^8
Non-volatility	No	No	Yes	Yes	Yes	Yes

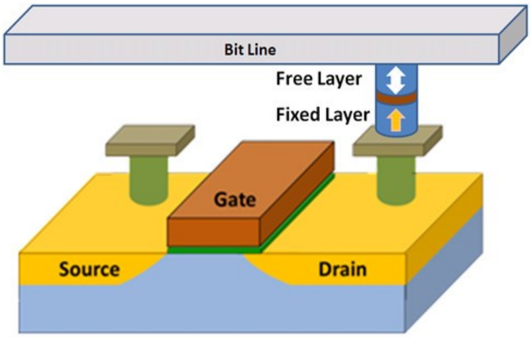

Fig. 1. Typical 1T-1MTJ perpendicular STT-MRAM bit cell

II. Exploration Flow

A. NVSIM Simulator

NVSIM [2], a modified environment of CACTI [3], is a circuit-level model for NVM performance, energy, and area estimation, which supports various NVM technologies, including STT-MRAM, PCRAM, RRAM, and legacy NAND Flash. It also includes the volatile SRAM memory. NVSIM is successfully validated against industrial NVM prototypes in [2], and it is expected to help boost architecture-level NVM-related studies. With NVSIM, we can estimate electrical features of a complete memory chip such as read/write access time, power consumption and so on, which can be used to calibrate a memory hierarchy of, for instance, a processor architecture simulator.

B. GEM5 Simulator

GEM5 [4] is a cycle accurate processor architecture simulator whose accuracy was validated against real hardware platform in [5]. It currently supports most commercial ISAs like ARM, ALPHA, MIPS, Power, SPARC and x86. The simulator's modularity allows these different ISAs to plug into the generic CPU models and the memory system without having to specialize one for the other. GEM5 can simulate a complete processor-based system with devices and operating system in full system mode and it supports also simulation of multi-core systems. The use of GEM5 allows us to define the overall processor system architecture, including the memory hierarchy specifications: cache size, L1/L2 cache and main memory latencies. Hence, we are able to extract execution time and all the memory transactions for a given application: number of L1/L2 read/write accesses, cache hits and misses, among other parameters.

C. Evaluation flow

Combining NVSIM with GEM5 allows us to evaluate different memory hierarchy strategies using SRAM and STT-MRAM in order to find the best trade-off in terms of performance and power consumption. Memory hierarchy defined in GEM5 can be calibrated in access latency using simulation results of NVSIM.

III. Experimental Setup

For our study, we propose to use some applications of SPLASH-2 benchmark suite [6], which are mostly in the area of High Performance Computing (HPC), to evaluate the impact of STT-MRAM for shared L2 cache on four-core processor architecture and its impact for L1 cache on two-core architecture. Table II gives details on input sets used for the benchmarks. We considered a 1GHz 32-bit RISC ARMv7 processor, with a complete Linux operating system running on top of it. We assume a two-level cache configuration: private 32kB L1 Instruction-cache (I-cache) 4-way associative, private 32kB L1 Data-cache (D-cache) 4-way associative, shared 512kB L2 cache 8-way associative. The main memory is a DDR3 type whose latency is fixed to 100 cycles.

IV. Performance Evaluation

Performance comparison between SRAM and STT-MRAM is made at node 45nm. First of all, we characterize each level of the memory hierarchy by simulation using NVSIM in order to calibrate latency parameters in GEM5. Table III describes performances of SRAM and STT-MRAM L1/L2 cache. As expected, STT-MRAM write latency is higher than SRAM write latency. Concerning hit latency, STT-MRAM is faster than SRAM for L2 cache. It is not surprising since STT-MRAM is denser than SRAM. As a result, for the same capacity, the L2 cache total area for STT-MRAM is smaller than the SRAM one, which results in smaller bitline delay. This difference on hit latency in favor of STT-MRAM is noticeable only for large cache capacity.

Fig. 2 shows the total execution time of several benchmarks of SPLASH-2 for the four-core architecture and for two scenarios: a baseline scenario using a SRAM-based L2 cache (SRAM) and a second scenario with a STT-MRAM based L2 cache (SRAM_STT). Results are normalized to the execution time spent for the SRAM-based L2 cache scenario. Observing Fig. 2, we can notice performances of the two scenarios are quite similar for the benchmarks simulated, and sometimes execution time is lower using STT-MRAM-based L2 cache. It could be explain by a smaller hit latency for STT-MRAM comparing to SRAM. Also, analyzing amount of read/write accesses in L2, we approximately have a ratio of 2,5:1 in average, in favor of read operations.

TABLE II. SPLASH-2 benchmarks input sets

Benchmark	Input set
fft	2^{20} total complex data points
lu1	Contiguous blocks, 512x512 Matrix, Block = 16
lu2	Non-Contiguous blocks, 512x512 Matrix, Block = 16
ocean1	Contiguous partitions, 514x514 Grid
ocean2	Non-Contiguous partitions, 258x258 Grid
radix	4M Keys, Radix = 4K

TABLE III. CACHE FEATURES

Field	32 kB L1 cache		512 kB L2 cache	
	SRAM	STT-MRAM	SRAM	STT-MRAM
Hit latency	1.25 ns	1.94 ns	4.28 ns	2.61 ns
Hit energy	0.024 nJ	0.095 nJ	0.27 nJ	0.28 nJ
Write latency	1.05 ns	5.94 ns	2.87 ns	6.25 ns
Write energy	0.006 nJ	0.04 nJ	0.02 nJ	0.05 nJ
Static power	22 mW	3.3 mW	320 mW	23 mW

Fig. 3 depicts the total execution time of a two-core architecture for three scenarios : a baseline scenario using a total SRAM-based L1 cache (SRAM), a second scenario with a total STT-MRAM-based L1 cache (STT_SRAM), a third scenario with a STT-MRAM-based L1 I-cache and a SRAM-based L1 D-cache (iSTT/dSRAM_SRAM), and a last scenario using STT-MRAM-based L1 D-cache and a SRAM-based L1 I-cache (dSTT/iSRAM_SRAM). Principally for fft, lu1 and lu2 benchmarks, execution time is bigger for both STT_SRAM and dSTT/iSRAM_SRAM scenarios. Since these benchmarks compute a very large amount of data, the most critical part in the memory hierarchy is the L1 D-cache memory. Using STT-MRAM for L1 D-cache will degrade overall performance because of its high write latency. Reducing the use of this memory technology only on L1 I-cache and keeping SRAM on L1 D-cache improves the overall performance to be almost the same as our baseline scenario. Indeed, in our case, all the benchmarks simulated are entirey cached. Hence, the number of writes in L1 I-cache is limited comparing to the number of writes in L1 D-cache.

V. ENERGY EVALUATION

Table III describes energy consumption of SRAM and STT-MRAM based L1/L2 cache. As expected, write access energy is higher for STT-MRAM whereas hit energy is almost the same in L2 cache for the two memory technologies. But the considerable gain of STT-MRAM over SRAM is on the leakage power: STT-MRAM is more than 10x less power consuming than SRAM. Indeed, most of the static power of memory systems comes from cell arrays. Because intrinsically non-volatile, STT-MRAM cell has zero standby power, and the CMOS access transistor does not need to be power supplied. All static power for STT-MRAM memory is due to peripheral circuitry such as address decoding, drivers and sense amplifiers.

Fig. 4 displays the total L2 dynamic energy. While total L2 read energy is comparable for the two architecture scenarios, total write energy is much higher for STT-MRAM based L2 cache due to its high write energy per bit access comparing to SRAM. However, because L2 cache is much more accessed by read operations, the total L2 dynamic energy is not so high using STT-MRAM instead of SRAM.

Observing Fig. 5 and 6, we note the major benefit for using STT-MRAM technology. Simulation result shows a gain over SRAM of more than 90% in terms of static power consumption for L2 cache. For total L1 cache, i.e. including all the L1 caches of each core, we save more than 80%, 40%, and 25% of static energy for the STT_SRAM, iSTT/dSRAM_SRAM and dSTT/iSRAM_SRAM scenarios respectively. This large gap in leakage power between the two memories makes STT-MRAM-based cache memory a very attractive alternative to save energy keeping overall application performance.

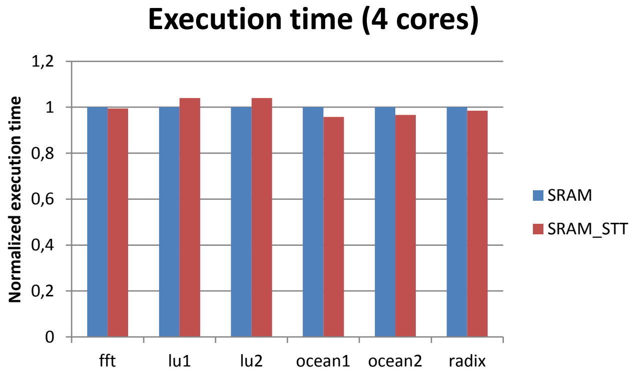

Fig. 2. Execution time on four-core architecture (Normalized to execution time of "SRAM" scenario)

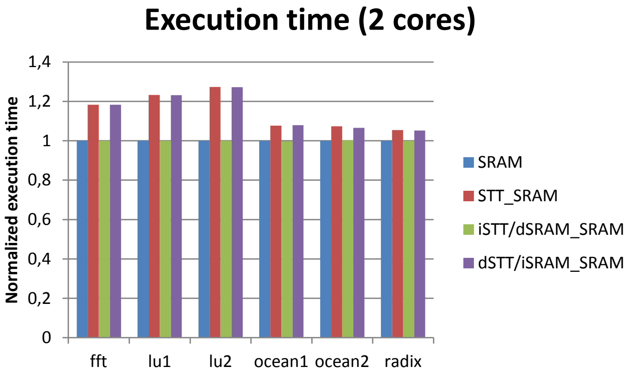

Fig. 3. Execution time for two-core architecture (Normalized to execution time of "SRAM" scenario)

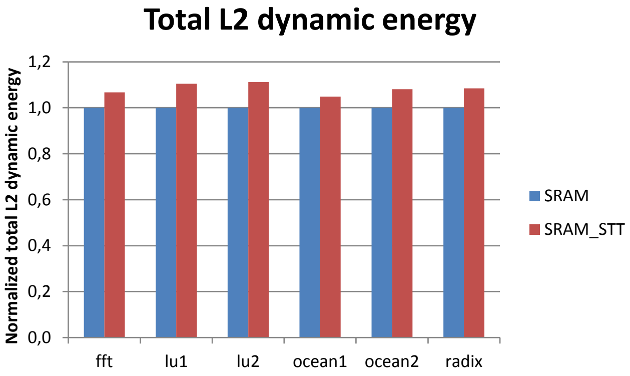

Fig. 4. Total L2 dynamic energy (Normalized to L2 dynamic energy of "SRAM" scenario)

VI. Related Work

Several studies were made upon integration of MRAM into the memory hierarchy of single-core and multi-core architecture. Evaluation of the benefit of 3D stacking ability of MRAM for 3D microprocessor was made in [7]. NUCA study with intra hybrid cache partitioned in regions of different memory technologies including MRAM was explored in [8]. Optimizations techniques such as early write termination which prevent unnecessary writes, or write buffers, to deal with high write latency and high write dynamic energy of MRAM were proposed in [9] and [10]. Trade-off between data retention and write latency of STT-MRAM were analyzed in [11]. All these studies were made on L2 cache or last level cache of the memory hierarchy. In our work, we have been studying impacts of MRAM also on upper level of the memory hierarchy, i.e. L1 cache. Besides, our objective is to explore all cache memory hierarchy strategies directly replacing SRAM with MRAM, taking into account that, for instance, MRAM can be up to seven times larger than SRAM for a same die footprint.

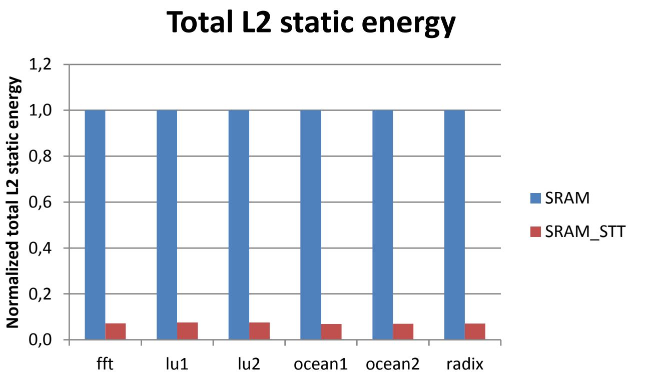

Fig. 5. Total L2 static energy (Normalized to L2 static energy of “SRAM” scenario)

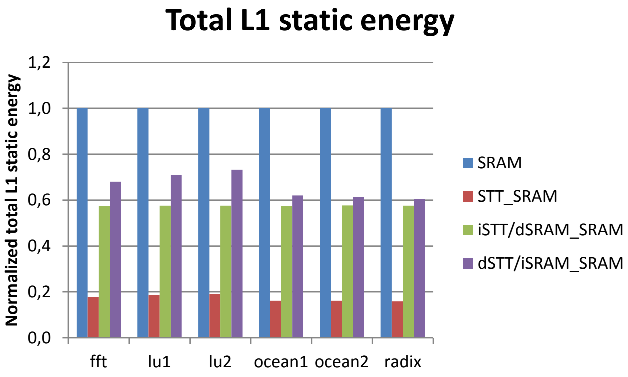

Fig. 6. Total L1 static energy (Normalized to L1 static energy of “SRAM” scenario)

VII. Conclusions

Among the emerging memory technologies, MRAM is a very promising candidate to help resolve one of the major challenges faced in continuing CMOS scaling: power dissipation. For future work, we plan to extend this study with the Thermally Assisted Switching MRAM technology whose implementation can lead to Magnetic Logic Unit (MLU) [12] presenting new logic functionalities compared with a standard MRAM. Fields of use of MLU are quite large including secure microcontroller, SIM/banking cards and magnetic sensors.

Acknowledgment

The Authors wish to acknowledge all people from ADAC team at LIRMM and people from Crocus technology for their support in this work.

References

[1] E. Kitagawa, and S. Fujita, “STT-MRAM cuts power use by 80%,” in eetimes.com. Available online at http://www.eetimes.com/document.asp?doc_id=1280753

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

[3] N. Muralimanohar, R. Balasubramonian, and N. P. Jouppi, “Cacti 6.0: A tool to model large caches,” Published in International Symposium on Microarchitecture, Chicago, Dec 2007, Tech. Rep., Apr. 2009.

[4] N. Binkert, S. Sardashti, R. Sen, K. Sewell, M. Shoaib, N. Vaish,M. D. Hill, D. A. Wood, B. Beckmann, G. Black, S. K. Reinhardt, A. Saidi, A. Basu, J. Hestness, D. R. Hower, and T. Krishna, “The gem5 simulator,” ACM SIGARCH Computer Architecture News, vol. 39, no. 2, pp. 1–7, Aug. 2011.

[5] A. Butko, R. Garibotti, L. Ost, and G. Sassatelli, “Accuracy Evaluation of GEM5 Simulator System,” in the proceedings of the 7th International Workshop on Reconfigurable Communication-Centric Systems-on-Chip (ReCoSoC). 2012, pp. 1–7.

[6] C. Bienia, S. Kumar, and K. Li, “PARSEC vs. SPLASH-2: A Quantitative Comparison of Two Multithreaded BenchmarkSuites on Chip-Multiprocessors,” in Proceedings of IISWC 2008, pp. 47–56, Sept. 2008.

[7] X. Dong, X. Wu, G. Sun, Y. Xie, H. Li and Y. Chen, “Circuit and microarchitecture evaluation of 3D stacking magnetic RAM (MRAM) as a universal memory replacement,” in DAC’08: Proceesings of the 45th annual Design Automation Conference, pp. 554-559, New York, NY, USA, 2008, ACM.

[8] X. Wu, J. Li, L. Zhang, E. Speight, R. Rajamony, and Y. Xie, “Hybrid cache architecture with disparate memory technologies,” in ACM SIGARCH Computer Architecture News, vol. 37, no. 3, 2009, pp. 34–45.

[9] X. Wu, J. Li, L. Zhang, E. Speight, R. Rajamony, and Y. Xie, “Power P. Zhou, B. Zhao, J. Yang, and Y. Zhang, “Energy reduction for stt-ram using early write termination,” in International Conference on Computer-Aided Design, 2009, pp. 264-268

[10] G. Sun, X. Dong, Y. Xie, J. Li, and Y. Chen, “A novel architecture of the 3D stacked MRAM L2 cache for CMPs,” in Proceedings of the International Conference on High-Performance Computer Architecture, 2009, pp. 239–249.

[11] A. Jog el al., “Cache revive: architecting volatile STT-RAM caches for enhanced performance in CMPs,” in 49th Annual Design Automation Conference, 2012, pp. 243–252.

[12] B. Cambou, “Match In Place. A novel way to perform secure and fast user’s authentication,” available online at www.crocus-technology.com