Enabling a Reliable STT-MRAM Main Memory Simulation

Kazi Asifuzzaman
Barcelona Supercomputing Center
Universitat Politècnica de Catalunya
Barcelona, Spain

Rommel Sánchez Verdejo
Barcelona Supercomputing Center
Universitat Politècnica de Catalunya
Barcelona, Spain

Petar Radojković
Barcelona Supercomputing Center
Barcelona, Spain

ABSTRACT

STT-MRAM is a promising new memory technology with very desirable set of properties such as non-volatility, byte-addressability and high endurance. It has the potential to become the *universal memory* that could be incorporated to all levels of memory hierarchy. Although STT-MRAM technology got significant attention of various major memory manufacturers, to this day, academic research of STT-MRAM main memory remains marginal. This is mainly due to the unavailability of publicly available detailed timing parameters which are required to perform a cycle accurate main memory simulation. Our study presents a detailed analysis of STT-MRAM main memory timing and propose an approach to perform a reliable system level simulation of the memory technology. We seamlessly incorporate STT-MRAM timing parameters into DRAMSim2 memory simulator and use it as a part of the simulation infrastructure of the high-performance computing (HPC) systems. Our results suggests that, STT-MRAM main memory would provide performance comparable to DRAM, while opening up various opportunities for HPC system improvements. Most importantly, our study enables researchers to conduct reliable system level research on STT-MRAM main memory, and to explore the opportunities that this technology has to offer.

CCS Concepts

•**Computer systems organization** → *Processors and memory architectures;* •**Hardware** → Non-volatile memory; •**Computing methodologies** → Massively parallel and high-performance simulations;

Keywords

STT-MRAM, Main memory, High-performance computing.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© {Owner/Author | ACM} {2017}. This is the author's version of the work. It is posted here for your personal use. Not for redistribution. The definitive Version of Record was published in {https://dl.acm.org/citation.cfm?id=3132416},
http://dx.doi.org/10.1145/3132402.3132416

1. INTRODUCTION

Memory systems are major contributors to the deployment and operational costs of large-scale high-performance computing (HPC) clusters [1][2][3], as well as one of the most important design parameters that significantly affect system performance [4][5]. For decades, DRAM devices have been the dominant building blocks for main memory systems in server and high-performance computing market. However, it is questionable whether this technology will continue to scale to meet the needs of next-generation systems.

Therefore, significant effort is invested in research and development of novel memory technologies. One of the candidates for next-generation memory is Spin-Transfer Torque Magnetic Random Access Memory (STT-MRAM). STT-MRAM is a novel, byte-addressable, non-volatile memory technology with high endurance. Although STT-MRAM technology was introduced only around ten years ago, STT-MRAM devices are already approaching DRAM in terms of capacity, frequency and device size. Actually, various STT-MRAM commercial products already found their way to some segments of the memory market.

STT-MRAM technology got significant attention of various major memory manufacturers. However, academic research on this technology is still marginal, and academia is struggling to conduct a reliable STT-MRAM main memory simulation. Although frequently used, simplistic memory models can introduce significant errors in the analysis of the overall system performance [6][7]. Therefore, detailed timing parameters are a *must-have* for any evaluation or architecture exploration study of STT-MRAM main memory . However, these detailed parameters are not publicly available because STT-MRAM manufacturers are reluctant to release any delicate information on the technology. Also, being a rapidly evolving technology, it is difficult even for the manufacturers to predict the exact timing for an upcoming STT-MRAM main memory device.

The main objective of our work is to understand and publish detailed STT-MRAM main memory timing parameters enabling a reliable system level simulation of the novel memory technology. The approach that we present converged through research cooperation with Everspin technologies Inc., one of the leading MRAM manufacturers, and it provides reliable STT-MRAM timing parameters while releasing no confidential information about any commercial products.

All information from the STT-MRAM main memory manufacturers clearly indicate that the STT-MRAM technology is (and will be) incorporated into the DDRx interface

and protocol, facilitating a seamless integration into the rest of the system. This actually provides a lot of information about the STT-MRAM timing parameters — it indicates that most of the timings will not change from DRAM to STT-MRAM main memory, as detailed in Section 2.3). For the parameters that will change due to differences in DRAM and STT-MRAM storage cell, we have to accept that there is no reliable information on how these timing parameters will change for the upcoming STT-MRAM devices. Therefore, we strongly argue that the best thing that we can do is a sensitivity analysis on these parameters.

Finally, we seamlessly incorporate our STT-MRAM timing analysis into the DRAMSim2 [6] memory simulator and use it as a part of the simulation infrastructure of the high-performance computing systems running SPEC 2006 benchmark suite. Our results show a fairly narrow overall performance deviation in response to significant variations in key timing parameters.

Overall, this study demonstrates an approach to perform a reliable cycle accurate simulation of STT-MRAM main memory, effortlessly incorporated into a widely accepted main memory simulator. The presented approach enables researchers to conduct system level research on the STT-MRAM main memory, and to explore the opportunities that this technology has to offer.

The rest of the article is organized as follows. Section 2 introduces STT-MRAM technology, its development trend in recent years and its timing parameters. Section 3 describes the experimental environment used in our study, while Section 4 presents and analyzes the results. Section 5 discusses opportunities and challenges of STT-MRAM memory systems. Finally, Section 6 discusses the related work, and Section 7 presents the conclusions of the study.

2. STT-MRAM MAIN MEMORY

In this section, we introduce STT-MRAM technology and its development trend in recent years. We also discuss why it is important for STT-MRAM main memory to be compatible with the existing systems, mainly CPUs and memory controllers, and how this compatibility impacts STT-MRAM organization and timing parameters. Finally, we present our reasoning and proposal for STT-MRAM main memory timings.

2.1 Technology overview

Research exploring the magneto-resistance caused by the spin polarized current can be tracked back in the '90s [8][9][10]. Although, significant scientific efforts of optimizing and applying this phenomenon to create a novel non-volatile memory is a relatively new approach. Only around ten years ago, in 2005, Hosomi *et al.* [11] presented a non-volatile memory utilizing spin transfer torque magnetization switching for the first time. In the following years, there has been a notable dedication of memory manufacturers researching this novel non-volatile memory technology.

The storage and programmability of STT-MRAM revolve around a Magnetic Tunneling Junction (MTJ). An MTJ is constituted by a thin tunneling dielectric being sandwiched between two ferro-magnetic layers. One of the layers has a fixed magnetization while the other layer's magnetization can be flipped. As Figure 1 depict, if both of the magnetic layers have the same polarity, the MTJ exerts low resistance therefore representing a logical "0"; in case of opposite polarity of the magnetic layers, the MTJ has a high resistance and represents a logical "1". In order to read a value stored in an MTJ, a low current is applied to it. The current senses the MTJ's resistance state in order to determine the data stored in it. Likewise, a new value can be written to the MTJ through flipping the polarity of its free magnetic layer by passing a large amount of current through it [12].

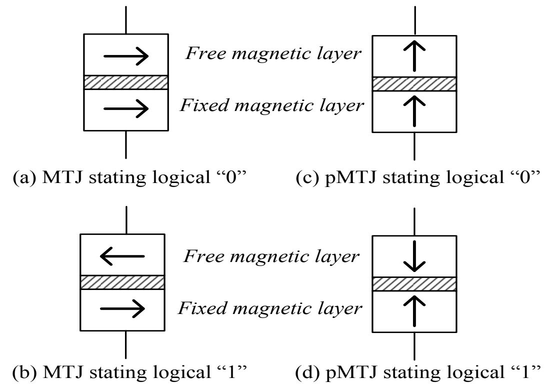

Figure 1: STT-MRAM cell

A more recent variation of MTJ is perpendicular MTJ (pMTJ). In contrast with the conventional MTJ, the poles of pMTJ magnetic layers are perpendicularly aligned with the plane of the wafer; see Figure 1(c) and (d). In 2010, Ikeda et al. presented pMTJ for the first time and demonstrated that it requires much lower write current than the conventional MTJ [13]. Recently, Janusz et al. has reported to achieve good write performance with pMTJ down to 11 nm device size [14].

Other variants of STT-MRAM cell design incorporated advanced 2T-2MTJ, 3T-2MTJ and 4T-2MTJ cells in a pursuit to improve performance and energy efficiency [15][16][17].

2.2 Development trend

Around ten-years-old, STT-MRAM is rapidly catching-up the mature DRAM technology. Figure 2, shows an approximate timeline of DRAM and STT-MRAM chip capacity development, and clearly illustrates the diminishing gap between these two technologies.

Development of DRAM devices started back in the '70s, and by the year 2003, DRAM chip capacity could reach upto 256Mb. Around at the same time, first reported STT-MRAM chip appeared with the capacity of 128Kb, which is a 2000× smaller capacity than the DRAM (note the logarithmic scale of the vertical axes). DRAM chip capacity gradually increased and reached 16Gb by the year 2016. Following a sharp incline, STT-MRAM chip capacity increased to 4Gb by the same year [18], reducing the capacity gap between these two technologies from 2000× in 2003 to only 4× in 2016.

Promising development has also been made improving STT-MRAM's bus frequency. While the first generation of DDR SDRAM had 133Mhz bus frequency, present day DDR3 and DDR4 compatible STT-MRAM are catching-up with the frequencies of the high-end DRAM devices [19].

The STT-MRAM device improvements come mainly from the enhancements in the MTJ design. With the recent variation of pMTJ, different memory manufacturers have demon-

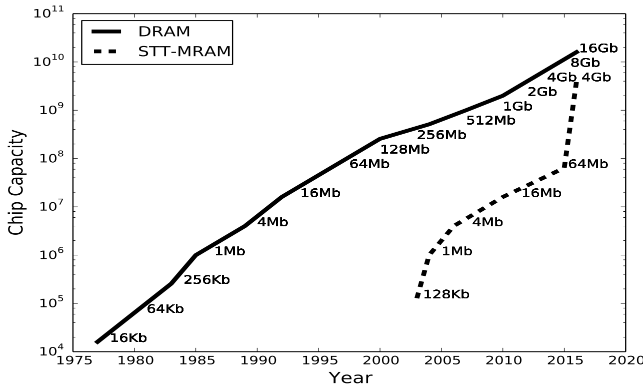

Figure 2: DRAM and STT-MRAM capacity growth in years

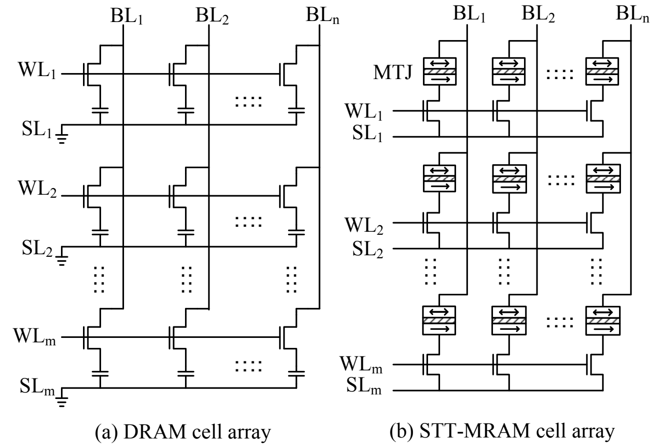

Figure 3: STT-MRAM and DRAM cell-array

strated a fierce competition to achieve the smallest device size for MTJs. In 2011, Samsung developed pMTJ at 17nm. In 2016, IBM demonstrated 11nm STT-MRAM junction. By the end of the year 2016, IMEC researchers reported to develop world's smallest pMTJ at 8nm.

An intensified effort in STT-MRAM research by the memory manufacturers may indicate a revolution with STT-MRAM memory technology is imminent, and we can expect to see a lot of exciting developments with this memory technology in near future.

2.3 Organization and CPU interface

Although the STT-MRAM is catching-up rapidly in terms of cell size, capacity and frequency, DRAM still have one great advantage — it is a standardized plug-and-play device. Today, we have various DRAM and CPU manufacturers and OEMs, and we have a full compatibility — we can connect any CPU (Intel, AMD, ARM-based) to any DRAM (Samsung, Micron, Hynix) as long as they follow the same DDRx standard. Although we probably take this for granted, it is very important to understand that this standardization requires tremendous effort and it is done only for main-stream products (technologies) with volumes that justify the investment.

Since STT-MRAM is a new technology with no specific standard, the manufacturers have two options. One would be to make a STT-MRAM main memory system from scratch, leading to a microarchitecture, interface and protocols fine-tuned for STT-MRAM technology. However, this would also require CPU manufacturers and OEMs to adopt their products to the STT-MRAM memory, by deploying, e.g., STT-MRAM specific memory controllers. Another option would be to bring it back on the DDRx, to adjust STT-MRAM microarchitecture and interface to this standard. This approach may not lead to an optimal STT-MRAM main memory device, but it probably is the only practical way to make STT-MRAM easily integrated into the existing systems. Publicly available product information and patents [20][21][22] from STT-MRAM manufacturers clearly indicate that they selected the second approach — incorporation of the STT-MRAM technology into DDRx interface and protocols enabling a seamless integration into the rest of the system. Therefore, STT-MRAM data array structure is very similar to that of DRAM (see Figure 3). In both designs, DRAM and STT-MRAM, transistors are used to access a selected set of cells, and the only fundamental difference is in the cell type, capacitor in the case of DRAM and MTJ in the case of STT-MRAM. Also, overall STT-MRAM device organization is essentially the same as the DRAM, in terms of number and size of the structures such as ranks, banks, sub-arrays, row, columns, and row buffers. Finally, STT-MRAM CPU interface is DRAM compatible.

2.4 Timing parameters: Our proposal

The fact that STT-MRAM memory is DDRx compatible, with the same or very similar organization and CPU interface, provides a lot of information about STT-MRAM timing parameters. Both, DRAM and STT-MRAM main memory devices use row buffer as an interface between the cell-arrays and the memory bus. Since the circuitry beyond the row buffer for DRAM and STT-MRAM would essentially be the same, once the data is in the row buffer, STT-MRAM timing parameters for the consequent operations would be the same as DRAM. For example, tCWD (Column write delay) corresponds to the delay between issuance of the column write command and placement of the data on the bus. Therefore, the value of this timing parameter does not change for STT-MRAM and DRAM. This applies to all the timing parameters that are not associated with row operations such as tBURST, tCAS, tWTR, etc., as summarized in Table 1. The timings are represented in DDR3-1600 cycles, but applies to other DDRx standards as well.

The only fundamental difference in STT-MRAM and DRAM main memory is their storage cell technology, MTJ and capacitor, respectively. Due to the difference in the cell access mechanism of these two memory technologies, the timing parameters associated with the STT-MRAM row operations would deviate from DRAM.[1] DRAM access is a voltage mode operation. To access the cell array, bitlines are precharged to a reference voltage (see Figure 3). The timing parameters associated with this operation is tRP (Row Precharge). Then a voltage is applied on the wordline to activate the access transistors allowing the sensing circuit to sense and move the data to the row buffer. The time it takes from a row access to get the data ready at the row buffer is denoted by tRCD (Row to column command delay). In the

[1]Rows of the DRAM or STT-MRAM cells constitute the cell arrays, see Figure 3. Row operations access directly to the memory cells.

Table 1: **Memory parameters not associated with row operation (DDR3-1600 cycles)**

Timing Parameters	Description	DRAM	ST*x*
tBURST	Burst length	4	4
tAL	Added latency to column access	0	0
tCAS/tCL	Column access strobe latency	11	11
tRTP	Read to precharge delay	6	6
tCCD	Column to column delay	4	4
tWTR	Write to read delay time	6	6
tRTRS	Rank to rank switching time	1	1
tCWD	Column write delay	10	10
tWR	Write recovery time	12	12
tCKE	Next power up for an idle device	4	4
tCMD	Command transport duration	1	1
tXP	Exit power down with DLL on to any valid command	5	5

contrary, STT-MRAM cell array access is a current mode operation and is completely different from the DRAM access mechanism. To read a data stored in an MTJ, a wordline is activated and a small amount of current is applied through corresponding bitline to sense the data (in terms of resistance) in a particular MTJ and eventually transferring it to the row buffer.

STT-MRAM specific timing parameters has neither been standardized nor been released by any industry. This is perhaps due to the perpetual evaluation of the STT-MRAM technology that is constantly changing over a short duration of time. Memory manufacturers, who are developing STT-MRAM are judiciously not revealing these parameters ahead of time; so, at this point, we have to accept that there is no reliable information on how these timing parameters will change for the upcoming STT-MRAM devices. Therefore, we strongly argue that the best we can do is a sensitivity analysis on the parameters that will change from DRAM to STT-MRAM. And we would strongly encourage any STT-MRAM related research to validate its analysis and proposals for various potential STT-MRAM parameters — i.e. to consider that uncertainly of the evolution of this technology.

In this study, we selected three set of timings naming ST-1.2, ST-1.5 and ST-2.0 with deviations of 1.2x, 1.5x and 2x from respective DRAM timing parameters as summarized in Table 2. The presented methodology converged through our research cooperation with Everspin Technologies Inc., However, the timing parameters used in this study does not specifically correspond to any of their commercial products. We believe, simulation performed with these timing parameters gives us an reliable range of possible system performance impact for upcoming STT-MRAM main memory devices. Although, some earlier studies has reported asymmetrical read-write latency for STT-MRAM, we used symmetrical read-write latency in compliance with the latest development and studies of the technology [16][17][23].

In addition to the parameters listed in Table 2, there is a change in STT-MRAM main memory operation sequence as well. In DRAM, when a row is accessed, the storage capacitors are discharged losing the data that it held. This is known as *destructive read*. After the *read* is performed, the data from the row buffer needs to be restored to the data array through a *write-back* before it can issue the next *precharge* command. Whereas, being a non-volatile memory, STT-MRAM read is *non-destructive*; i.e., it does not need to restore the data back to the array. Because of this, STT-MRAM can issue the consequent prechanrge command sooner [24]. Therefore, in specific cases, STT-MRAM tRC (Row cycle) can be shorter than DRAM even with a longer tRCD and and tRP.

Table 2: DRAM and STT-MRAM parameters associated with row operation (DDR3-1600 cycles)

Timing Parameters	Description	DRAM	ST-1.2	ST-1.5	ST-2.0
tRCD	Row to column command delay	11	14	17	22
tRP	Row precharge	11	14	17	22
tFAW	Four row activation window	24	29	36	48
tRRD	Row activation to Row activation delay	5	6	8	10
tRFC[2]	Refresh cycle time	208	1	1	1

We understand that the ranges of the STT-MRAM timing parameters presented in Table 2 may change in future, with new information publicly released and along with the evolution of the technology, but the overall approach that we propose should persist.

2.5 Timing parameters: Dead ends

Our search for reliable STT-MRAM timing parameters was not straight-forward; it lasted three years, involved collaboration with two STT-MRAM memory manufacturers, and we attempted several approaches before reaching to the final methodology. We hope that our experience will motivate future studies targeting STT-MRAM main memory to approach the publicly available timing parameters with a dose of a healthy skepticism, and help them to avoid the unreliable ones.

Initially, we planned to simulate STT-MRAM main memory by using NVMain simulator [25]. After analyzing NVMain STT-MRAM timings, we noticed that several key parameters had values that differ significantly from our understanding of the STT-MRAM main memory, and the timings provided by the manufacturers. For example, NVMain configuration file for a 4GB MRAM[3] listed tRAS (row access strobe) to be 0. Whereas, in DDR*x* standard, tRAS is constituted by tRCD, tCAS, tBURST and the delay for the data restoration which corresponds to 28 cycles in DDR3-1600. In addition, tRCD was set to 14 cycles with an explanation to have it derived from Everspin MR2A16A product datasheet. We could not verify this derivation to be correct. To clear up the confusion, we contacted the NVMain developers asking for a clarification and source of their STT-MRAM parameters, but got no reply. Since we were unable to verify how these timing parameter values were formulated and we had some serious doubts about their values, we had to classify the NVMain STT-MRAM main memory parameters as un-

[2]Being a non-volatile memory STT-MRAM does not need refresh. This parameter is set to 1 to avoid incompatibility with the simulator.

[3]NVMain configuration file describes the parameters of the 4GB MRAM device. This configuration file was released in 2015, even before 64MB devices were manufactured.

reliable and discard them for using in our experiments.

A couple of studies [24][26] simulate STT-MRAM main memory by integrating publicly available STT-MRAM cell parameters into the CACTI [27] cache simulator. Using *cache simulator* to estimate timing and energy parameters of a *main memory* is not a straight-forward approach. Main memory devices devices have higher capacity by several orders of magnitude, different organization (DIMMs, ranks, banks, chips, rows, columns) and interface (e.g. row buffer), which would yield a completely different parameter values. We failed to find any information on how CACTI could be adopted for a main memory simulation, and the studies who use this approach provide no information on how they bridged the gap between cache and main memory simulation.

2.6 Energy parameters

Although we understand the importance of evaluating energy consumption, at this point, such evaluation on energy components of high-density STT-MRAM main memory is infeasible due to the lack of publicly available up-to-date resources. Estimation of STT-MRAM energy components are a part of our ongoing work.

3. SIMULATION INFRASTRUCTURE

We analyze system performance impact with STT-MRAM main memory in comparison to DRAM main memory. In this section we present the application benchmark suite, CPU and main memory simulator used for this study.

3.1 Benchmark suite

STT-MRAM main memory was evaluated on a set of eight integer and twelve floating point benchmarks from the SPEC CPU 2006 suite [28]. Table 3 lists the benchmarks with their application areas used for the study.

Table 3: SPEC CPU 2006 benchmarks used in the study

Benchmark	Application Area	Language
h264ref	Video Compression	C
libquantum	Quantum Computing	C
perlbench	Programming Language	C
gobmk	Artificial Intelligence	C
hmmer	Gene Sequence Analysis	C
sjeng	Artificial Intelligence	C
aster	Path-finding Algorithm	C++
bzip2	Compression	C
gamess	Quantum Chemistry	Fortran
tonto	Quantum Chemistry	Fortran
namd	Molecular Dynamics	C++
gromacs	Molecular Dynamics	C,Fortran
dealII	Finite Element Analysis	C++
sphinx3	Speech Recognition	C,Fortran
leslie3d	Fluid Dynamics	Fortran
cactusADM	General Relativity	C,Fortran
GemsFDTD	Computational Electromagnetics	Fortran
milc	Quantum Chromodynamics	C
bwaves	Fluid Dynamics	Fortran
lbm	Fluid Dynamics	C

3.2 CPU Simulation

In order to evaluate STT-MRAM main memory system, we simulated an Intel Sandy Bridge-EP E5-2670 processor, which is a dominant architecture in HPC systems [29]. Intel Sandy Bridge-EP E5-2670 comprises eight cores operating at 3.0 GHz. Although the processors support hyper-threading at core level, this feature is disabled, as in most of the HPC systems. Sandy Bridge processors are connected to main memory through four DDR3-1600 channels.

Table 4: Cache parameters of Sandy Bridge E class processor used in the study

	L1-Data	L2	L3
Size	32 KB	256 KB	20 MB
Latency (in CPU cycles)	4	8	28
Cache line size	64 Byte	64 Byte	64 Byte
Set associativity	8 way	8 way	20 way

We used ZSim [30] system simulator for the experiments. Developed by researchers from MIT and Stanford University, ZSim is designed for simulation of large-scale systems. However, ZSim was originally developed to simulate Intel Westmere architecture which is obsolete at this point. One of the tasks that we had to perform was to upgrade and validate ZSim for Intel Sandy Bridge processor. The ZSim upgrade was done by following the Intel documentation [31], and it comprised several steps. First, we adjusted the latency of numerous instructions, and added support the for the new x86 vector instruction extensions i.e. AVX, SSE3, that are supported by Sandy Bridge and were not supported by Westmere. We also improved the fusion of the instructions into a single micro-op, and we increased the number of entries in the reorder buffer from 128 (Westmere) to 168 (Sandy Bridge). Finally, the simulated hardware platform comprises a detailed model of Sandy Bridge-EP E5-2670 cache hierarchy [32]. This Sandy Bridge E class processor has eight cores, dedicated L1 instruction and data cache of 32 KB each, dedicated L2 cache of 256 KB and a shared L3 cache of 20 MB, summarized in Table 4. In all three levels of cache memory, we used the Least Recently Used (LRU) cache replacement policy and for the L3 cache level we implemented the slice allocation hash function explained Maurice *et al.* [33].

3.3 Main memory simulation

Both DRAM and STT-MRAM main memory is simulated with DRAMSim2 [6]. DRAMSim2 is a cycle accurate model of a DRAM main memory. All major components in a modern memory system are modeled as their own respective objects within the source code, including: ranks, banks, command queue, the memory controller, etc. DRAMSim2 is developed by University of Maryland and it is validated

Table 5: Main memory simulator settings

Parameters	Values
NUM_CHANS	4
JEDEC_DATA_BUS_BITS	64
TRANS_QUEUE_DEPTH	32
CMD_QUEUE_DEPTH	32
EPOCH_LENGTH	100000
ROW_BUFFER_POLICY	close_page
ADDRESS_MAPPING_SCHEME	scheme2
SCHEDULING_POLICY	rank_then_bank_round_robin
QUEUING_STRUCTURE	per_rank

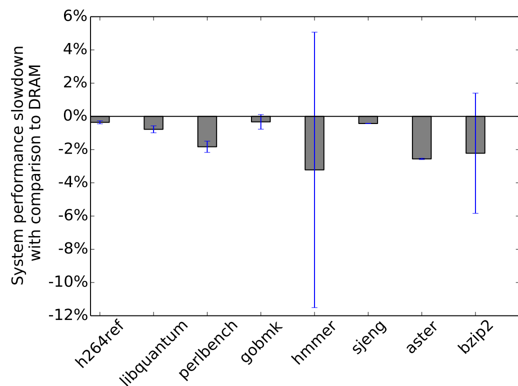

Figure 4: ST-1.2 Configuration (integer benchmarks): Speedup ranges from 0.3% (gobmk) to 3.2% (hmmer), and it is 1.4% on average.

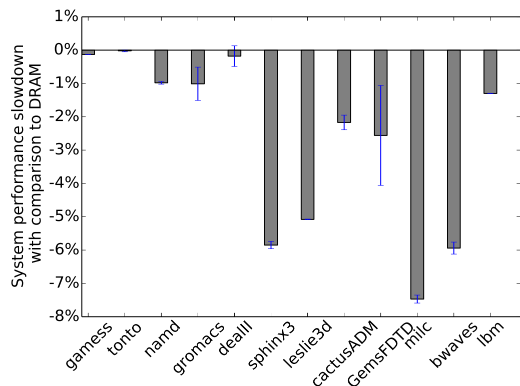

Figure 5: ST-1.2 Configuration (floating point benchmarks): Speedup ranges from 0% (tonto) to 7.4% (milc), and it is 2.7% on average.

against manufacturer Verilog models. DRAMSim2 can be integrated with various CPU simulators through fairly simple interface. Table 1 and 2 summarizes DRAM and STT-MRAM main memory parameters used in this study, while Table 5 lists the simulator settings for main memory.

Simple integration of ZSim and DRAMSim2 may lead to an underestimation of the main memory access latency. ZSim simulates memory access up to the last level of cache, while DRAMSim2 is focused on the detailed timing simulation of the memory device. This means that a direct merge of ZSim and DRAMSim2 would not consider to the delay contributed by all the circuitry between the last level cache and main memory device, including the memory controller and the memory channel. In order to account for this delay, we introduce an *extra latency* of 70ns between ZSim and DRAMSim2. Estimation of this *extra latency* has been validated in real machine [34].

3.4 Validation

We have validated the simulation infrastructure against the actual hardware comprising Sandy Bridge EP E5-2670 processor connected to four DDR3-1600 channels. CPU pipeline is validated by using a set of synthetic benchmarks with a main loop comprised of a single instruction type. Different version of the synthetic benchmarks test in-order and out-of-order execution. Our test suite is comprised of 519 synthetic benchmarks, covering almost all instructions included in the instruction set architecture (ISA) of the Sandy Bridge EP E5-2670 processor. Cache hierarchy and main memory latency is validated with *lmbench* [35]. The *lmbench* benchmark essentially measures the access time of random accesses to an array of a given size. In our experiments, we covered the array sizes from 4KB (fitting into the L1 cache), to 4GB (main memory access). Finally, we validated the overall simulation infrastructure using SPEC CPU 2006 benchmarks, by comparing its execution on the actual hardware with the simulated one.

3.5 Methodology

In the experiments summarized in this paper, we simulated eight instances of SPEC CPU 2006 benchmarks running on a single Sandy Bridge socket, i.e. one benchmark instance per core. Each benchmark instance was executed for 50 billion instructions. To compare DRAM and STT-MRAM memory systems, we measured the performance for each process under study in two main memory configurations. In this paper, we report average performance difference between the DRAM (baseline) and the STT-MRAM memory system, and standard deviation of all the measurements.

4. RESULTS

In this section we present the results from our simulations experimenting STT-MRAM main memory performance impact in comparison to DRAM. For STT-MRAM main memory, we test three sets of timings namely ST-1.2, ST-1.5, ST-2.0. In ST-1.2 configuration, the specific STT-MRAM timing parameters: tRCD, tRP, tFAW and tRRD are 1.2× slower w.r.t. the corresponding DRAM timings. Similar applies to ST-1.5 and ST-2.0 configuration, as summarized in Table 2.

Figure 4 shows overall system performance impact of ST-1.2 configuration on SPEC integer benchmark. The horizontal bars represent system performance deviation for the corresponding benchmarks listed at X axis. This deviation has been measured by the change of Cycles per Instruction (CPI) values between systems with DRAM and STT-MRAM main memory. Actually, for all the integer benchmarks we detect a negative CPI change meaning that the benchmarks experience a speedup with the STT-MRAM main memory. The speedup ranges from 0.3% (gobmk) to 3.2% (hmmer). Floating point benchmarks with ST-1.2 configuration follow a similar trend but with a higher amplitude, see Figure 5. Four out of twelve benchmarks achieved more than 5% system performance improvement in comparison to DRAM. Average speedup for all the benchmarks is 2.7%.

Although, ST-1.2 is apparently configured to be comparatively slower w.r.t DRAM, the results with this configuration report performance improvement (speedup) for all benchmarks over DRAM. This is due to the operation sequence of STT-MRAM, which is different from DRAM, detailed in Section 2.4. Unlike DRAM, STT-MRAM has a non-destructive read which does not have to write-back; meaning it can issue precharge command sooner [24]. Hence, STT-MRAM tRC (Row cycle) for this configuration can be shorter than DRAM even with a longer tRCD and and tRP.

ST-1.5 results, see Figure 6, show performance degrada-

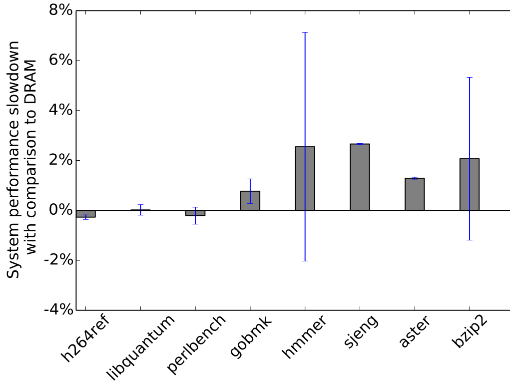

Figure 6: ST-1.5 Configuration (integer benchmarks): Slowdown ranges from -0.2% (h264ref) to 2.6% (sjeng), and it is 1.1% on average.

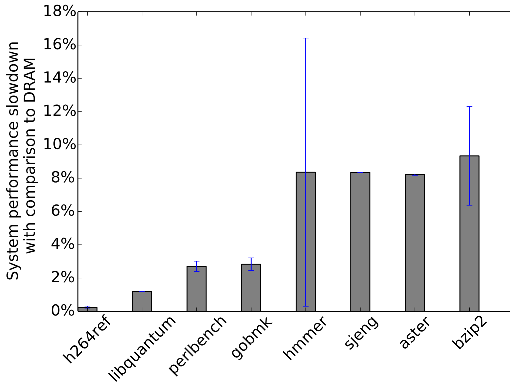

Figure 7: ST-2.0 Configuration (integer benchmarks): Slowdown ranges from 0.2% (h264ref) to 9.3% (bzip2), and it is 5.1% on average.

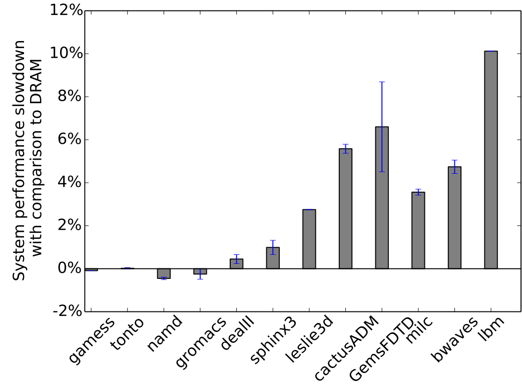

Figure 8: ST-1.5 Configuration (floating point benchmarks): Slowdown ranges from 0% (gamess) to 10.1% (lbm), and it is 2.8% on average.

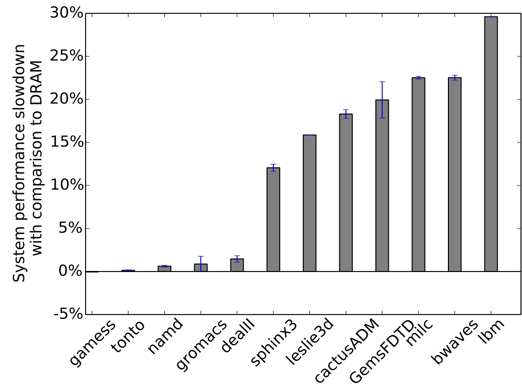

Figure 9: ST-2.0 Configuration (floating point benchmarks): Slowdown ranges from 0% (gamess) to 29.6% (lbm), and it is 11.9% on average.

tion for most integer benchmarks, 1.1% on average. Benchmarks h264ref and perlbench, however, still experience a speedup on the STT-MRAM memory systems. Floating point benchmarks, experience higher slowdowns, 2.8% on average and 10% in the worst case (lbm), see Figure 8.

For the most pessimistic configuration ST-2.0, all benchmarks experience slowdown w.r.t. to the DRAM. Slowdown of the integer benchmarks ranges between 0.2% (h264ref) and 9.3% (bzip2), and it is 5.1% on average, see Figure 7. Floating point benchmarks are even more sensitive to the delays realized in ST-2.0 configuration, see Figure 9. Five of the benchmarks experience slowdown less than 2%, but for the remaining ones the slowdown ranges between 12% and 29.6%, leading to the average slowdown of 11.9%

Overall, the results indicate that the system performance experience a minor impact for variation of STT-MRAM timing parameters associated with row operations: tRCD, tRP, tFAW and tRRD. Even when these timing parameters are pessimistically set to be twice as slow as DRAM, the system performance degrades only by average of 5.1% and 11.9% for integer and floating point benchmarks, respectively. For ST-1.5 configuration, STT-MRAM main memory based system experiences an average slowdown of only 1.1% for integer and 2.8% for the floating point benchmarks. For STT-MRAM main memory with ST-1.2 configuration, we actually measure a speedup w.r.t. to DRAM.

5. STT-MRAM OPPORTUNITIES

STT-MRAM main memory would provide performance and capacity comparable to DRAM systems, while opening up various opportunities for HPC system improvements. STT-MRAM is a non-volatile technology and therefore, it requires no refresh. Thus, a performance and energy advantage over DRAM technology can come from resolving the memory refresh problem. STT-MRAM is also a technology that mitigates the transient faults caused by magnetic or electrical interference, that account for a significant portion of the overall memory faults. Since STT-MRAM technology would improve the reliability of the memory systems, the complexity and overheads of the contemporary error correction approaches can be reduced [34].

However, its adoption as alternative main memory technology is limited due its high production cost as compared to DRAM, a mature technology with huge production volumes. Therefore, if we really want to make STT-MRAM an alternative to DRAM in main memory systems, we have to find domains and use cases so that STT-MRAM primary development cost can be justified with *significant* improvements in features of interest.

6. RELATED WORK

6.1 STT-MRAM main memory

To the best of our knowledge, only four studies analyze suitability of STT-MRAM for main memory in high-performance computing (HPC) and server domain, and one targeting mobile devices.

Meza *et al.* [36] analyze architectural changes to enable small row buffers in non-volatile memories, PCM, STT-MRAM, and RRAM. The study concludes that NVM main memories with reduced row buffer size can achieve up to 67% energy gain over DRAM at a cost of some performance degradation. Kultursay *et al.* [26] evaluate STT-MRAM as a main memory for SPEC CPU2006 workloads and show that, without any optimizations, early-design STT-MRAM [37] is not competitive with DRAM. The authors also propose *partial write* and *write bypass* optimizations that address time and energy-consuming STT-MRAM write operation. Optimized STT-MRAM main memory achieves performance comparable to DRAM while reducing memory energy consumption by 60%.

Suresh *et al.* [38] analyze design of memory systems that match the requirements of data intensive HPC applications with large memory footprints. The authors propose a complex 5-level memory hierarchy with SRAM caches, EDRAM or HMC last level cache, and non-volatile PCM, STT-MRAM, or FeRAM main memory. The study also analyzes using a small DRAM off-chip cache that filters most of the accesses to the non-volatile main memory and therefore reduces a negative impact on performance and dynamic energy consumption of NVM technologies.

Asifuzzaman *et al.* [34] evaluate STT-MRAM main memory for high-performance computing and analyzes the performance impact when DRAM is simply replaced with STT-MRAM. The presented results suggests that 20% slower STT-MRAM main memory induces negligible system performance impact, while opening up opportunities to provide some highly desired properties such as non-volatility, zero stand-by power and high endurance.

In all studies that target HPC and server domain, DRAM and various STT-MRAM main memory designs are evaluated by using *average* read and write latencies. This approach fails to account for the highly complex behavior of modern memory systems and may under-report the their affect on the overall system performance [6][7].

Jiang *et al.* [39] propose using STT-MRAM main memory in mobile devices. The main objective of their study is to save the energy of the DRAM refresh, by using the non-volatile memory technology. The authors also propose two STT-MRAM microarchitectual enhancements that would improve the STT-MRAM performance in the presence of the read disturbance errors. The proposal is evaluated based on the STT-MRAM parameters targeting LPDDR devices estimated by Wang *et al.* [24] using CACTI [27] and NVSim [40].

All of these studies are very important because they explore potential target markets for STT-MRAM technology, which leads to a better understanding of its market value. However, these studies share a general weakness — questionable estimation of STT-MRAM timing parameters. The studies either use average latency or use timing information with no available source which could be validated. We acknowledge this difficulty and that is why, in our study, we focus on understanding the timing parameters. We believe, our study will improve future STT-MRAM main memory research, in both exploratory and microarchitectural evaluations.

6.2 STT-MRAM on-chip caches

Most of the system-level research so far, focused on suitability of STT-MRAM for on-chip cache memories. In general, these studies propose to exploit STT-MRAM's non-volatility, zero stand-by power, and higher density with respect to SRAM to design next-generation caches.

Li *et al.* [41] propose to integrate STT-MRAM with SRAM to construct a hybrid adaptive on-chip cache architecture that offers low power consumption, low access latency and high capacity. The authors evaluate hybrid SRAM / STT-MRAM cache on a set of PARSEC and SPLASH-2 workloads, and report a 37% reduction of power consumption along with 23% performance improvement compared to SRAM based design. Zhou *et al.* [42] observe that many bits in the STT-MRAM cache are re-written with the same value. As, early STT-MRAM cell design write operation requires significant energy, such unnecessary writes can be avoided to reduce power consumption. They introduce *early write termination*, a scheme which terminates redundant bit writes for STT-MRAM caches and achieves upto 80% of write energy reduction for SPEC 2000, SPEC 2006 and SPLASH-2 benchmarks.

Chang *et al.* [43] compares STT-MRAM and eDRAM as a replacement of SRAM for last level caches. The study identifies specific weaknesses of each technology and analyzes the trade-offs associated with each of these technologies for implementing last level caches. The study concludes, if refresh is effectively controlled, eDRAM based last level cache becomes a viable, energy-efficient alternative for multi-core processors.

Various studies propose to trade-off STT-MRAM's non-volatility to improve write latency and energy consumption [44][45][46][47]. Li *et al.* [45] indicate that majority of cache data stay active for much shorter time duration than the data retention time assumed in the STT-MRAM designs. The authors suggest that, the retention time can be aggressively reduced to achieve significant switching performance and power improvements. Jog *et al.* [46] formulate the relation between retention time and write latency in order to find optimal retention time for an efficient STT-MRAM cache hierarchy. Smullen *et al.* [44] propose a ultra-low retention time STT-MRAM caches supported by a DRAM-like refresh policy. Sun *et al.* [47] further exploit the scenario by deploying STT-MRAM with multiple retention levels. Smullen *et al.* [44] and Sun *et al.* [47] propose architectures with SRAM L1 cache along with relaxed-retention STT-MRAM L2 and L3 cache. The hybrid cache architectures are evaluated on SPEC 2006 and PARSEC benchmarks and they show significant performance improvement over conventional SRAM-based designs while reducing energy consumption.

The studies perform analysis of STT-MRAM cache latencies, area, leakage and dynamic power based on publicly available STT-MRAM cell parameters and CACTI [27]. Unfortunately, these STT-MRAM timing and energy parameters could not be used to simulate main memory because such devices have higher capacity (by several orders of magnitude), and different organization and interface, as detailed in Section 2.5.

7. CONCLUSIONS

STT-MRAM main memory got significant attention of various major memory manufacturers, and expecting to bring a revolution in the memory market. However, academic research on this technology is still marginal, and academia is struggling to conduct a reliable STT-MRAM main memory simulation. In order to overcome this problem, this study thoroughly analyzes and publishes detailed STT-MRAM main memory timing parameters enabling a reliable system level simulation of this technology. The study is based on the fact that STT-MRAM main memory devices is and will be incorporated into the DDRx interface and protocol, indicating that most of the timings will not change from DRAM to STT-MRAM main memory. For the parameters that will change due to differences in DRAM and STT-MRAM storage cell, we have to accept that there is no reliable information on how these timing parameters will change for the upcoming STT-MRAM devices. Therefore, we strongly argue that the best we can do at this point is a sensitivity analysis on these parameters. The approach that we present converged through research cooperation with Everspin technologies Inc., and it provides reliable STT-MRAM timing parameters while releasing no confidential information about any commercial products.

We also seamlessly incorporate STT-MRAM timing parameters into DRAMSim2 memory simulator and use it as a part of the simulation infrastructure of the high-performance computing systems. The results of our simulations show that STT-MRAM main memory would provide performance comparable to DRAM systems, while opening up various opportunities for HPC system improvements. An intensified effort of memory manufacturers in STT-MRAM research promises exciting developments on this technology in near future. Now, with the reliable detailed timing parameters that we publish, we would strongly encourage academia to also explore the opportunities that this technology has to offer.

8. ACKNOWLEDGMENTS

This work was supported by BSC, Spanish Government through Programa Severo Ochoa (SEV-2015-0493), by the Spanish Ministry of Science and Technology through TIN2015-65316-P project and by the Generalitat de Catalunya (contracts 2014-SGR-1051 and 2014-SGR-1272). This work has also received funding from the European Union's Horizon 2020 research and innovation programme under ExaNoDe project (grant agreement No 671578). The authors wish to thank Terry Hulett, Duncan Bennett and Ben Cooke from Everspin Technologies Inc., for their technical support.

9. REFERENCES

[1] Peter Kogge, et al. ExaScale Computing Study: Technology Challenges in Achieving Exascale Systems, September 2008.

[2] Avinash Sodani. Race to Exascale: Opportunities and Challenges. Keynote Presentation at the 44th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO), 2011.

[3] Rick Stevens, et al. A Decadal DOE Plan for Providing Exascale Applications and Technologies for DOE Mission Needs. Presentation at Advanced Simulation and Computing Principal Investigators Meeting, March 2010.

[4] B. Jacob. *The Memory System: You Can't Avoid It; You Can't Ignore It; You Can't Fake It. M.* Morgan & Claypool Publishers, Reading, Massachusetts, 2009.

[5] Wm. A. Wulf and Sally A. McKee. Hitting the memory wall: Implications of the obvious. *SIGARCH Comput. Archit. News*, 1995.

[6] P. Rosenfeld, E. Cooper-Balis, and B. Jacob. DRAMSim2: A Cycle Accurate Memory System Simulator. *IEEE Computer Architecture Letters*, 2011.

[7] David Wang, et al. DRAMsim: A Memory System Simulator. *SIGARCH Comput. Archit. News*, 33(4), 2005.

[8] B. Dieny, et al. Giant magnetoresistive in soft ferromagnetic multilayers. *Phys. Rev. B*, 1991.

[9] J.K. Spong, et al. Giant Magnetoresistive Spin Valve Bridge Sensor. *IEEE Transactions on Magnetics*, 32(2):366–371, 1996.

[10] J. A. Katine, et al. Current-Driven Magnetization Reversal and Spin-Wave Excitations in Co /Cu /Co Pillars. *Phys. Rev. Lett.*, 84:3149–3152, 2000.

[11] M. Hosomi, et al. A Novel Nonvolatile Memory with Spin Torque Transfer Magnetization Switching: Spin-RAM. In *IEEE International Electron Devices Meeting*, 2005.

[12] Yuan Xie. Modeling, Architecture, and Applications for Emerging Memory Technologies. *IEEE Design Test of Computers*, 2011.

[13] S. Ikeda, et al. A perpendicular-anisotropy CoFeB–MgO magnetic tunnel junction. In *Nature Materials*, volume 9, pages 721–724, 2010.

[14] J. J. Nowak, et al. Dependence of voltage and size on write error rates in spin-transfer torque magnetic random-access memory. *IEEE Magnetics Letters*, 7:1–4, 2016.

[15] K. Abe, et al. Novel Hybrid DRAM/MRAM Design for Reducing Power of High Performance Mobile CPU. In *IEEE International Electron Devices Meeting (IEDM)*, 2012.

[16] H. Noguchi, et al. A 250-MHz 256b-I/O 1-Mb STT-MRAM with Advanced Perpendicular MTJ Based Dual cell for Nonvolatile Magnetic Caches to Reduce Active Power of Processors. In *Symposium on VLSI Technology (VLSIT)*, 2013.

[17] R. Nebashi, et al. A 90nm 12ns 32Mb 2T1MTJ MRAM. In *IEEE International Solid-State Circuits Conference*, 2009.

[18] K. Rho, et al. 23.5 A 4Gb LPDDR2 STT-MRAM with compact 9F2 1T1MTJ cell and hierarchical bitline architecture. In *2017 IEEE International Solid-State Circuits Conference (ISSCC)*, 2017.

[19] Everspin Technologies, Inc. Everspin displays both the 1Gb DDR4 Perpendicular ST-MRAM device and a 1GByte DDR3 Memory Module (DIMM) at Stand A3-545. https://www.everspin.com/news/everspin-previews-upcoming-products-electronica, 2016.

[20] H. Kim, et al. Magneto-resistive memory device including source line voltage generator, 2013. US Patent App. 13/832,101.

[21] H.R. Oh. Resistive Memory Device, System Including the Same and Method of Reading Data in the Same, 2014. US Patent App. 14/094,021.

[22] C. Kim, et al. Magnetic random access memory, 2013. US Patent App. 13/768,858.

[23] Everspin Technologies, Inc. Everspin Enhances RIM Smart Meters with Instantly Non-Volatile, Low-Energy MRAM Memory. http://www.everspin.com/everspin-embedded-mram, 2015.

[24] Jue Wang, Xiangyu Dong, and Yuan Xie. Enabling High-performance LPDDRx-compatible MRAM. ISLPED, 2014.

[25] M. Poremba and Y. Xie. Nvmain: An architectural-level main memory simulator for emerging non-volatile memories. In *2012 IEEE Computer Society Annual Symposium on VLSI*, 2012.

[26] E. Kultursay, et al. Evaluating STT-RAM as an Energy-Efficient Main Memory Alternative. In *IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*, 2013.

[27] Naveen Muralimanohar, Rajeev Balasubramonian, and Norman P. Jouppi. CACTI 6.0: A Tool to Understand Large Caches. *HP Technical Report HPL-2009-85*, 2009.

[28] John L. Henning. SPEC CPU2006 Benchmark Descriptions. *SIGARCH Comput. Archit. News*, 2006.

[29] Top500. Top500 Supercomuter Sites. http://www.top500.org/, 2017.

[30] Daniel Sanchez and Christos Kozyrakis. Zsim: Fast and accurate microarchitectural simulation of thousand-core systems. In *Proceedings of the 40th Annual International Symposium on Computer Architecture*, ISCA, 2013.

[31] Intel Corporation. Intel 64 and IA-32 Architectures Software Developer Manuals, 2017.

[32] Intel. Intel® 64 and IA-32 Architectures Optimization Reference Manual, 2015.

[33] Clémentine Maurice, et al. *Reverse Engineering Intel Last-Level Cache Complex Addressing Using Performance Counters*. 2015.

[34] Kazi Asifuzzaman, et al. Performance Impact of a Slower Main Memory: A Case Study of STT-MRAM in HPC. In *Proceedings of the Second International Symposium on Memory Systems*, (MEMSYS), 2016.

[35] Larry McVoy and Carl Staelin. Lmbench: Portable Tools for Performance Analysis. In *Proceedings of the Annual Conference on USENIX Annual Technical Conference*, (ATEC), 1996.

[36] Jing Li Justin Meza and Onur Mutlu. Evaluating Row Buffer Locality in Future Non-Volatile Main Memories. *Safari Technical Report No. 2012-002*, 2012.

[37] Guangyu Sun, et al. A Novel Architecture of the 3D Stacked MRAM L2 Cache for CMPs. In *IEEE 15th International Symposium on High Performance Computer Architecture (HPCA)*, 2009.

[38] A. Suresh, P. Cicotti, and L. Carrington. Evaluation of Emerging Memory Technologies for HPC, Data Intensive Applications. In *IEEE International Conference on Cluster Computing (CLUSTER)*, 2014.

[39] Lei Jiang, et al. Improving read performance of STT-MRAM based main memories through Smash Read and Flexible Read. In *21st Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2016.

[40] X. Dong, et al. NVSim: A Circuit-Level Performance, Energy, and Area Model for Emerging Nonvolatile Memory. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 31(7):994–1007, 2012.

[41] Jianhua Li, C.J. Xue, and Yinlong Xu. STT-RAM Based Energy-Efficiency Hybrid Cache for CMPs. In *IEEE/IFIP 19th International Conference on VLSI and System-on-Chip (VLSI-SoC)*, 2011.

[42] Ping Zhou, et al. Energy Reduction for STT-RAM Using Early Write Termination. In *IEEE/ACM International Conference on Computer-Aided Design - Digest of Technical Papers*, 2009.

[43] M. T. Chang and P. Rosenfeld and S. L. Lu and B. Jacob. Technology comparison for large last-level caches (L3Cs): Low-leakage SRAM, low write-energy STT-RAM, and refresh-optimized eDRAM. In *IEEE 19th International Symposium on High Performance Computer Architecture*, 2013.

[44] C.W. Smullen, et al. Relaxing non-volatility for fast and energy-efficient STT-RAM caches. In *IEEE 17th International Symposium on High Performance Computer Architecture (HPCA)*, 2011.

[45] Hai Li, et al. Performance, Power, and Reliability Tradeoffs of STT-RAM Cell Subject to Architecture-Level Requirement. *IEEE Transactions on Magnetics*, 47(10):2356–2359.

[46] A. Jog, et al. Cache Revive: Architecting Volatile STT-RAM Caches for Enhanced Performance in CMPs. In *49th ACM/EDAC/IEEE Design Automation Conference (DAC)*, 2012.

[47] Zhenyu Sun, et al. Multi Retention Level STT-RAM Cache Designs with a Dynamic Refresh Scheme. In *44th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, 2011.