Performance and Power Estimation of STT-MRAM Main Memory with Reliable System-level Simulation*

KAZI ASIFUZZAMAN, Barcelona Supercomputing Center & Universitat Politècnica de Catalunya, Spain
ROMMEL SÁNCHEZ VERDEJO, Barcelona Supercomputing Center & Universitat Politècnica de Catalunya, Spain
PETAR RADOJKOVIĆ, Barcelona Supercomputing Center, Spain

It is questionable whether DRAM will continue to scale and will 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 an emerging non-volatile memory with a lot of potential that could be exploited for various requirements of different computing systems. Being a novel technology, STT-MRAM devices are already approaching DRAM in terms of capacity, frequency and device size. Although STT-MRAM technology got significant attention of various major memory manufacturers, academic research of STT-MRAM main memory remains marginal. This is mainly due to the unavailability of publicly available detailed timing and current parameters of this novel technology, which are required to perform a reliable main memory simulation on performance and power estimation. This study demonstrates an approach to perform a cycle accurate simulation of STT-MRAM main memory, being the first to release detailed timing and current parameters of this technology from academia — essentially enabling researchers to conduct reliable system level simulation of STT-MRAM using widely accepted existing simulation infrastructure. The results show a fairly narrow overall performance deviation in response to significant variations in key timing parameters, and the power consumption experiments identify the key power component that is mostly affected with STT-MRAM.

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

Additional Key Words and Phrases: STT-MRAM, Main memory, High-performance computing.

ACM Reference Format:
Kazi Asifuzzaman, Rommel Sánchez Verdejo, and Petar Radojković. 2021. Performance and Power Estimation of STT-MRAM Main Memory with Reliable System-level Simulation. *ACM Trans. Embedd. Comput. Syst.* 1, 1, Article 1 (January 2021), 25 pages. https://doi.org/10.1145/3476838

*Extension of the conference paper: Kazi Asifuzzaman, Rommel Sánchez Verdejo, and Petar Radojković. *Enabling a reliable STT-MRAM main memory simulation.* In Proceedings of the International Symposium on Memory Systems (MEMSYS), 2017.

Authors' addresses: Kazi Asifuzzaman, Barcelona Supercomputing Center & Universitat Politècnica de Catalunya, Spain, kazi.asifuzzaman@bsc.es; Rommel Sánchez Verdejo, Barcelona Supercomputing Center & Universitat Politècnica de Catalunya, Spain, rommel.sanchez@bsc.es; Petar Radojković, Barcelona Supercomputing Center, Spain, petar.radojkovic@bsc.es.

1539-9087/2021/1-ART1 $15.00
https://doi.org/10.1145/3476838

1 INTRODUCTION

Today, we live our life depending on a range of computing systems from smartphones, desktop computers, laptops, embedded systems to high performance computing (HPC) systems. Most people of the society are direct or indirect users of these systems, and such systems provide the crucial support to uphold the modern and efficient lifestyle. All of these computing systems innately incorporate some form of memory systems which are indispensable for their operation and functionality. In modern computing systems, *Memory* usually refers to the *Main Memory* of the system, which holds and provides data to perform computation. Since processor design has undergone an unprecedented development in recent years and subsequently expanded the gap between processor and memory speed, it made the memory system a vital design aspect to improve performance.

Memory systems are major contributors to the deployment and operational costs of large-scale high-performance computing (HPC) clusters [35][60][64], as well as one of the most important design parameters that significantly affect system performance [26][72]. For many years, DRAM devices have been the dominant building blocks for main memory in most computing systems including desktop, laptop, smartphones and HPC systems. DRAM technology became matured over decades of persistent research and development. For production in mass volume, DRAM has also become a very affordable technology for deployment across all computing domains. However, for a long time researchers have so heavily concentrated on improving processor and cache design that, improving the main memory itself has been far-overlooked. As a result, we are *hitting the memory wall*, which would not be solved unless main memory technology is significantly improved. Perhaps the greatest challenge that DRAM faces is technology scaling. Extreme scaling is forcing to shrink the size of the DRAM capacitor in an unprecedented ratio making it increasingly vulnerable to errors. Due to several design challenges, this approach of DRAM scaling is not expected to sustain forever. We need innovations, new methods and perhaps new technologies to move forward to meet next-generation computing needs.

Therefore, significant effort is invested in research and development of novel memory technologies. Among the emerging memory technologies notable candidates are Phase Change Memory (PCM), Resistive RAM (RRAM) and Spin-Transfer Torque Magnetic Random Access Memory (STT-MRAM), along with the 3D-stacked variants of DRAM namely Hybrid Memory Cube (HMC) and High Bandwidth Memory (HBM). Although all of these technologies have their own advantages, two of the major contenders, PCM and RRAM have significantly slower access time in comparison to DRAM, which makes them unsuitable to be considered as a main memory alternative. Since HMC and HBM are practically based on DRAM technology, STT-MRAM stands out to be most promising novel memory technology which has a true potential to be a main memory alternative. STT-MRAM is a novel, byte-addressable, non-volatile memory technology with high endurance. Although STT-MRAM technology was introduced only around fourteen 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 due to the lack of publicly available detailed timing parameters of STT-MRAM main memory. This is because, STT-MRAM manufacturers are reluctant to release these detailed parameters or any delicate information on the technology. Also, being a rapidly evolving technology, it is difficult even for the manufacturers to predict the exact timing and current parameters for an upcoming STT-MRAM main memory device.

Although frequently used, simplistic memory models (not using detailed parameters) can introduce significant errors in the analysis of the overall system performance [54][70]. Therefore, detailed timing parameters are a *must-have* for any evaluation or architecture exploration study of STT-MRAM main memory.

The main objective of our work is to understand and develop a methodology to reliably simulate this novel technology and publish corresponding detailed STT-MRAM main memory parameters to enable a reliable system level simulation. 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 parameters while releasing no confidential information about any commercial products. In this study, we evaluate STT-MRAM only as the main memory of a HPC system, and we do not investigate STT-MRAM as a cache memory alternative, neither we suggest any changes in cache memory architecture.

We seamlessly incorporate our STT-MRAM timing and current parameter analysis into the DRAMSim2 [54] 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. The results also identify the key power component that is mostly affected with STT-MRAM. 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 analysis of detailed parameters. Section 3 describes the experimental environment used in our study, while Section 4 presents and analyzes the results. Section 5 discusses opportunities 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 simulation challenges with this technology and our approach of estimating its timing and current parameters.

2.1 Technology overview

Research exploring the magneto-resistance caused by the spin polarized current can be tracked back in the '90s [10][61][31]. Although, significant scientific efforts of optimizing and applying this phenomenon to create a novel non-volatile memory is a relatively new approach. Only around fourteen years ago, in 2005, Hosomi *et al.* [19] 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 [73].

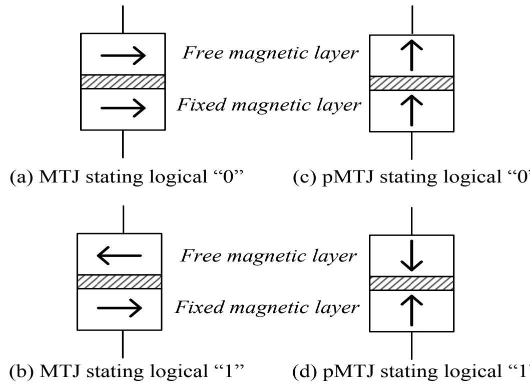

Fig. 1. STT-MRAM cell

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 [22]. Janusz et al. has reported to achieve good write performance with pMTJ down to 11 nm device size [48].

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 [1][47][46].

2.2 Development trend

Around fourteen-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 [53], 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 [14].

The STT-MRAM device improvements come mainly from the enhancements in the MTJ design. With the recent variation of pMTJ, different memory manufacturers have demonstrated 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, therefore, now it is the time to perform system level research with this technology to explore and identify the computing domains that could be benefited from the advantages that this technology has to offer.

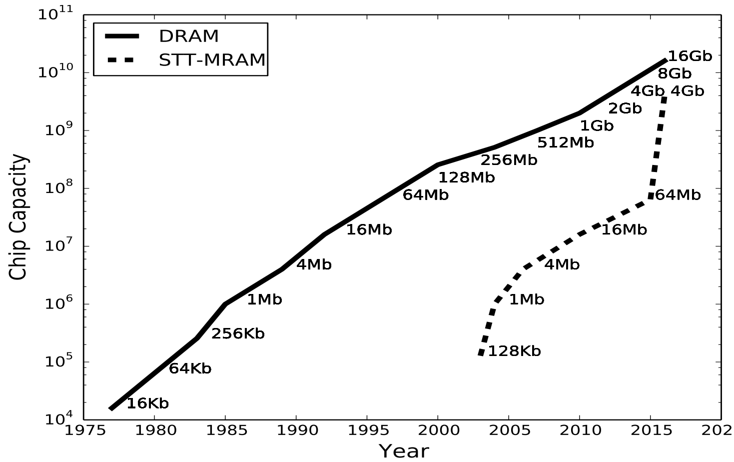

Fig. 2. DRAM and STT-MRAM capacity growth in years

2.3 STT-MRAM Simulation Challenges

To find suitable use cases for STT-MRAM main memory, it is essential to conduct reliable simulation of STT-MRAM. However, simulation of STT-MRAM main memory with detailed timing parameters has been a challenging task due to the unavailability of reliable estimations of timing and power consumption parameters. Few previous studies attempted to evaluate STT-MRAM as the main memory of a system.

Meza *et al.* [30] use a cycle-accurate DDR3-DRAM memory simulator and estimate STT-MRAM parameters based on Fujitsu's 16kb test-chip built in 2010 with 0.13μm technology [16]. The authors assume, that t_{WR} and t_{RCD} parameters for STT-MRAM main memory would change on a range of twice as slow to twice as fast with respect to DRAM. In our opinion, it is difficult to estimate a reasonable assessment of STT-MRAM main memory using such a wide range of values for key latency parameters. The study also does not provide any information about latency components before main memory device, making it it infeasible to repeat the study or to quantify the impact of the STT-MRAM t_{WR} and t_{RCD} parameters to the *overall* main memory access latency.

Kultursay *et al.* [37] compare DRAM and STT-MRAM performance by simulating fixed latencies for row buffer hit (30 ns) and conflict (50 ns) without specifying the breakdown or inclusion of the latency components for this delay. The source of these estimations are not revealed in the paper. The authors also state that they modified CACTI to model STT-MRAM, however there is no information how this modification was formulated, taking into account the fact that CACTI is widely used as a cache memory simulator, but least likely to be used to simulate main memories. In fact, main memories have a very different organization (ranks, banks, columns, rows etc.), interface (row-buffer) and capacity (in GBs), in comparison to cache memories. These differences lead to a very different timing implementations of main memories than cache memories. Therefore, findings of any cache memory study of a particular memory technology, can not be conceivably adopted to simulate the main memory, even of the same technology.

Suresh *et al.* [67] simulate STT-MRAM read and write operations with a fixed latency of 35 ns, obtaining these estimations from ITRS report, 2013 [25]. The study provides no information about the latency components before main memory device, or DRAM device latencies.

In general, previous studies use obsolete STT-MRAM timing parameters or parameters with no reliable source. STT-MRAM main memory evaluation is incomplete without cycle-accurate

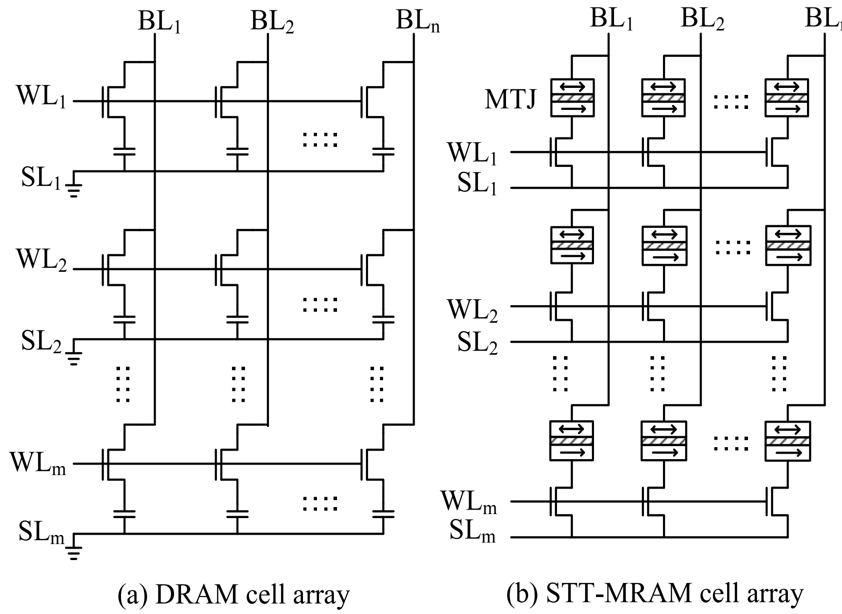

Fig. 3. STT-MRAM and DRAM cell-array

simulation with reliable timing parameters. The lack of detailed timing parameters is also the main problem for any STT-MRAM microarchitectural exploration, improvement and evaluation. In addition to this, the *before main memory device latency* is not validated versus real systems, or it is directly omitted from the simulation infrastructure analysis (addressed in Section 3.3).

Our search for reliable STT-MRAM timing parameters was not straightforward; it lasted for four years and involved collaboration with two STT-MRAM memory manufacturers.

Initially, we planned to simulate STT-MRAM main memory by using the NVMain simulator [52]. After analyzing NVMain STT-MRAM timings, we noticed that several key parameters had values that differ significantly from our understanding of STT-MRAM main memory, as well as the timings provided by manufacturers. For example, the NVMain configuration file for a 4GB MRAM[1] 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 the Everspin MR2A16A product datasheet. We could not verify this derivation to be correct.

A couple of studies [71][37] simulate STT-MRAM main memory by integrating publicly available STT-MRAM cell parameters into the CACTI [45] cache simulator. Using a *cache simulator* to estimate timing and energy parameters of a *main memory* is not a straightforward approach. Main memory 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 completely different parameter values. We failed to find any information on how CACTI could be adopted for main memory simulation, and the studies that use this approach provide no information on how they bridged the gap between cache and main memory simulation.

2.4 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 [33][49][32] 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.

2.5 Timing parameters: Our approach

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.[2] 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

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

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

Timing Parameters	Description	DRAM	STx
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

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

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 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 publicly 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 consistent 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.

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[3]	Refresh cycle time	208	1	1	1

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

In this study, we selected three set of timings naming ST-1.2, ST-1.5 and ST-2.0 with deviations of 1.2×, 1.5× and 2× 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. Another important aspect that needs to be addressed while modeling STT-MRAM memories is the asymmetrical Read/Write of STT-MRAM which is caused by the MTJs inherent torque asymmetry [6]. Several works propose that STT-MRAM can also be configured to adopt symmetrical Read/Write [4][13][47][46]. Since, in this study we are comparing STT-MRAM against DRAM, we used symmetrical Read/Write scheme, which is also compliant with the information we received on the STT-MRAM commercial product from the manufacturer.

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 [71]. Therefore, in specific cases, STT-MRAM tRC (Row cycle) can be shorter than DRAM even with a longer tRCD and and tRP.

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.

2.6 STT-MRAM power estimations

Power estimations are inseparable parts of evaluating any novel technology. Performance estimation accompanied with power consumption help to decide where a particular technology may serve best. A technology that is very good from a performance perspective but has a high power consumption may not be suitable for specific use cases, (i.e. battery supported hand-held devices). Similarly, a technology which is power efficient but severely lacks in performance may also not be an ideal choice for some computing domains. To perform an estimation on STT-MRAM main memory power

Current Parameters	Description	DRAM	ST-1.2	ST-1.5	ST-2.0
IDD0	**Active–Precharge Current**	**1305**	**1566**	**1957**	**2610**
IDD1	**Active–Read–Precharge Current**	**1395**	**1674**	**2092**	**2790**
IDD2P	Precharge Power–Down Exit Current	846	846	846	846
IDD2Q	Precharge Quiet Standby Current	1030	1030	1030	1030
IDD2N	Precharge Standby Current	1050	1050	1050	1050
IDD3P	Active Power–Down Current	990	990	990	990
IDD3N	Active Standby Current	1310	1310	1310	1310
IDD4	**Operating Burst Current**	**1765**	**2118**	**2647**	**3530**
IDD5	Refresh Current	1940	0	0	0
IDD6	Self Refresh Current	246	0	0	0
IDD7	**Bank Interleave Read Current**	**2160**	**2592**	**3240**	**4320**

Table 3. DRAM and STT-MRAM current parameters used in the study (in mA.)

consumption, it is essential to have detailed configuration parameters for power estimation. Since STT-MRAM is a novel technology which is constantly evolving, it is challenging to have a stable set of power configuration parameters. In addition, since STT-MRAM is in a competitive development stage, the manufacturers are being very careful not to release any key information regarding the power models of their devices. At this point, there are no definitive information released publicly (i.e. detailed data-sheet) from any manufacturer on STT-MRAM current parameters. However, through our research cooperation, we manage to acquire data-sheet with detailed parameters of an actual STT-MRAM main memory product from Everspin, and we carefully build our work around it, without explicitly releasing the exact numbers as they are protected for commercial reasons.

As explained in Section 2.5, the only fundamental difference between DRAM and STT-MRAM main memory is their storage cell technology, therefore the only current parameters that would change for STT-MRAM are the ones associated with accessing these cells. We carefully analyze the current parameters that model power consumption of a DRAM device, and based on our understanding of STT-MRAM operation, we assume that among DRAM current parameters, only four will change for STT-MRAM. These parameters correspond to *Active–Precharge Current* (`IDD0`), *Active–Read–Precharge Current* (`IDD1`), *Operating Burst Current* (`IDD4`) and *Bank Interleave Read Current* (`IDD7`). Since, STT-MRAM employs current-mode operation to access its cells, it requires comparatively more current than DRAM's voltage-mode cell operations. Therefore, we perform a sensitivity analysis on the higher side of these parameters using the same methodology that we used to estimate the timing parameters, (see Section 2.5). Considering STT-MRAM does not require refresh, *Refresh Current* (`IDD5`) and *Self Refresh Current* (`IDD6`) are set to `0`. The remaining current parameters which generally fall into *Power-Down* and *Standby* category do not change from DRAM to STT-MRAM since they are not associated with any operation accessing the cells. A recent research report from IBM also affirms that these current parameters do not deviate from DRAM to STT-MRAM [27].

All current parameters used in this study to estimate STT-MRAM power consumption is listed in Table 3. In the table, the first column represents the generic representation of the current parameters, the second column lists the description of current parameters, the third column lists DRAM current parameters for a DDR3-1600 device [56]; fourth, fifth and sixth column list the current parameters for ST-1.2, ST-1.5 and ST-2.0, with deviations of 1.2×, 1.5× and 2× from respective DRAM current parameters.

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 eleven integer and twelve floating point benchmarks from the SPEC CPU 2006 suite [18]. Table 4 lists the benchmarks with their application areas used for the study.

3.2 CPU Simulation

In order to evaluate STT-MRAM main memory system, we simulate an Intel Sandy Bridge-EP E5-2670 processor, which is a dominant architecture in HPC systems [68]. 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.

Benchmark	Application Area	Language
h264ref	Video Compression	C
libquantum	Quantum Computing	C
perlbench	Programming Language	C
gcc	C Compiler	C
mcf	Combinatorial Optimization	C
gobmk	Artificial Intelligence	C
hmmer	Gene Sequence Analysis	C
sjeng	Artificial Intelligence	C
xalancbmk	XML Processing	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

Table 4. SPEC CPU 2006 benchmarks used in the study.

We use ZSim [57] 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 perform is to upgrade and validate ZSim for Intel Sandy Bridge processor. The ZSim upgrade is done by following the Intel documentation [24], and it comprises several steps. First, we adjust the latency of numerous instructions, and add support the for the new x86 vector instruction extensions i.e. AVX, SSE3, that are supported by Sandy Bridge and are not supported by Westmere. We also improved the fusion of the instructions into a single micro-op, and we increase 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 [23]. 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

	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

Table 5. Cache parameters of Sandy Bridge E class processor used in the study.

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

Table 6. Main memory simulator settings.

in Table 5. In all three levels of cache memory, we used the Least Recently Used (LRU) cache replacement policy and for the L3 cache level we implement the slice allocation hash function as explained by Maurice *et al.* [43].

3.3 Main memory simulation

Both DRAM and STT-MRAM main memory are simulated with DRAMSim2 [54]. 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 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 6 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 referred to as *before main memory device latency* (discussed in Section 2.3). 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 [2].

3.4 Validation

We validate 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* [44]. The *lmbench* benchmark essentially measures the access time of random accesses to an array of a given size. In our experiments, we cover the array sizes from 4KB (fitting into the L1 cache), to 4GB (main memory access). Finally, we validate 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 with DRAM and three STT-MRAM main memory configurations. In this paper, we report performance and power deviation with DRAM (baseline) and STT-MRAM memory configurations.

4 RESULTS

4.1 Performance analysis

In this section, we present the results of STT-MRAM main memory performance impact in comparison to DRAM. We extend our results for both *open page* and *close page* policies. In *open page* policy,

Fig. 4. ST-1.2 Configuration (integer benchmarks)

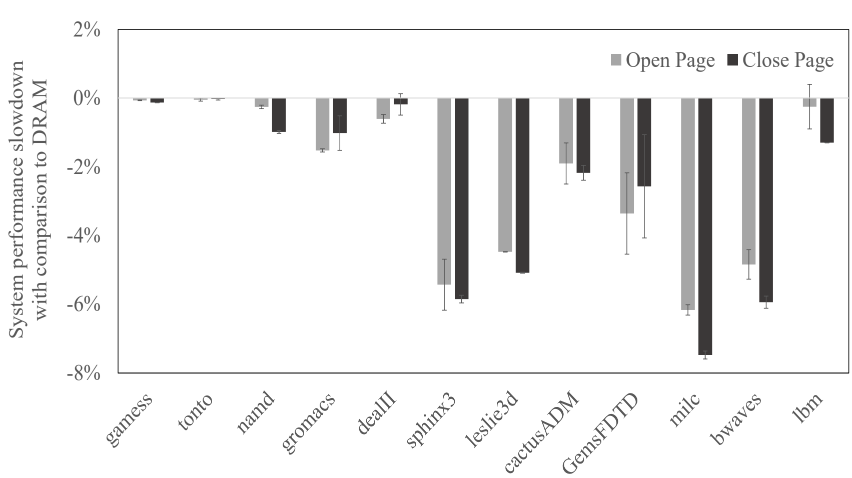

Fig. 5. ST-1.2 Configuration (floating point benchmarks)

a page is kept open to take advantage of subsequent "page hits" until a "page conflict" appears. In contrary, in *close page* policy, a page is immediately closed after a read or write operation is performed.

For STT-MRAM main memory, we test three sets of timings, namely ST-1.2, ST-1.5, ST-2.0. In the 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 the overall system performance impact of the ST-1.2 configuration with SPEC integer benchmarks. The horizontal bars represent system performance deviation for the corresponding benchmarks listed on the 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.

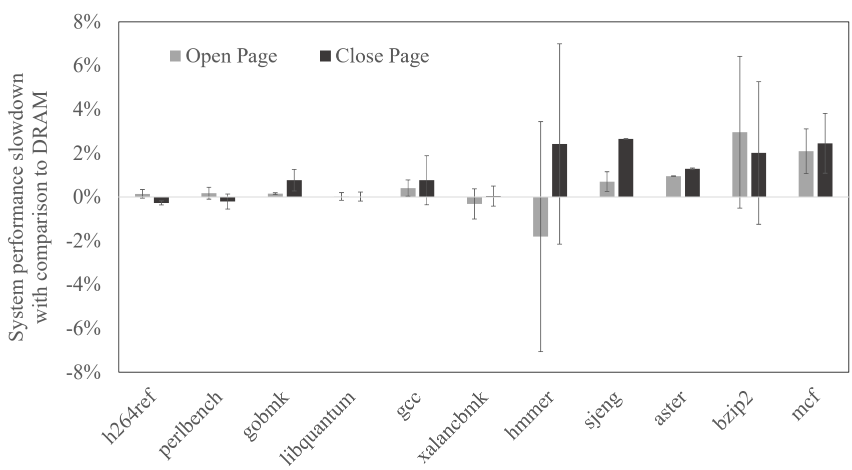

Fig. 6. ST-1.5 Configuration (integer benchmarks)

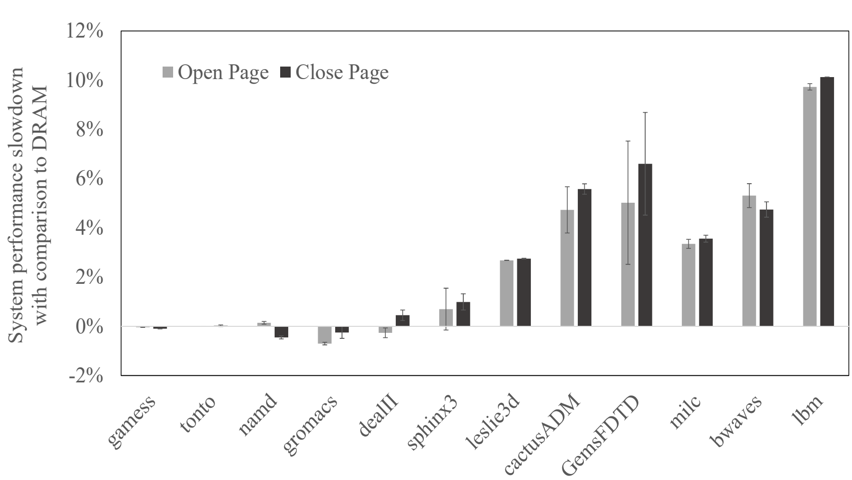

Fig. 7. ST-1.5 Configuration (floating point benchmarks)

With open page policy, the speedup ranges from 0.0% (`h264ref`) to 4.8% (`mcf`); and 2.2% in average. And with close page policy, the results does not change much: speedup ranging from 0.3% (`gobmk`) to 4.8% (`mcf`); and 2.2% in average. Floating point benchmarks with the ST-1.2 configuration follow a similar trend but with a higher amplitude, see Figure 5. Four out of twelve benchmarks achieved around 5% system performance improvement in comparison to DRAM for both open and close page policies. Average speedups for all the benchmarks are 2.4% (open page) and 2.7% (close page).

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, as detailed in Section 2.5. Unlike DRAM, STT-MRAM has a non-destructive read which does not have to write-back; meaning it can issue precharge commands sooner [71]. Hence, STT-MRAM tRC (Row cycle) for this configuration can be shorter than DRAM even with a longer tRCD and and tRP.

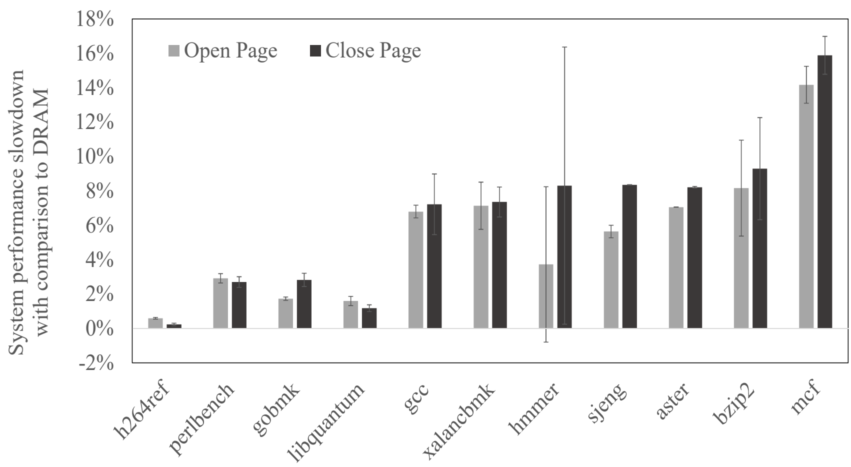

Fig. 8. ST-2.0 Configuration (integer benchmarks)

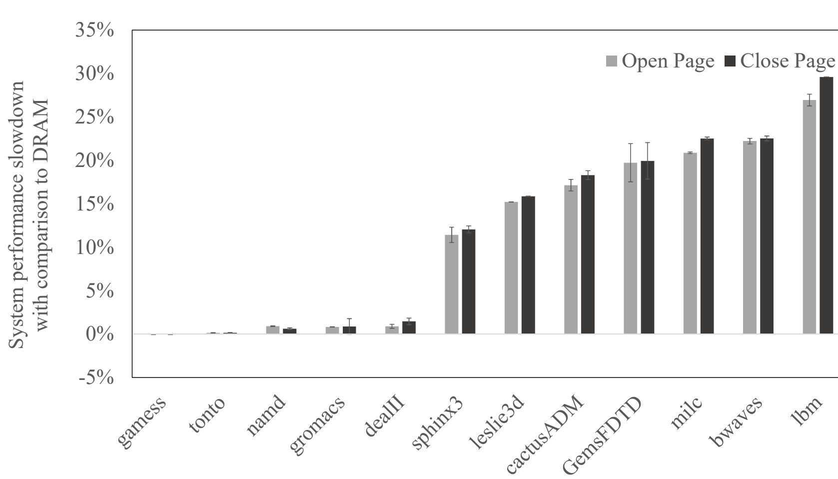

Fig. 9. ST-2.0 Configuration (floating point benchmarks)

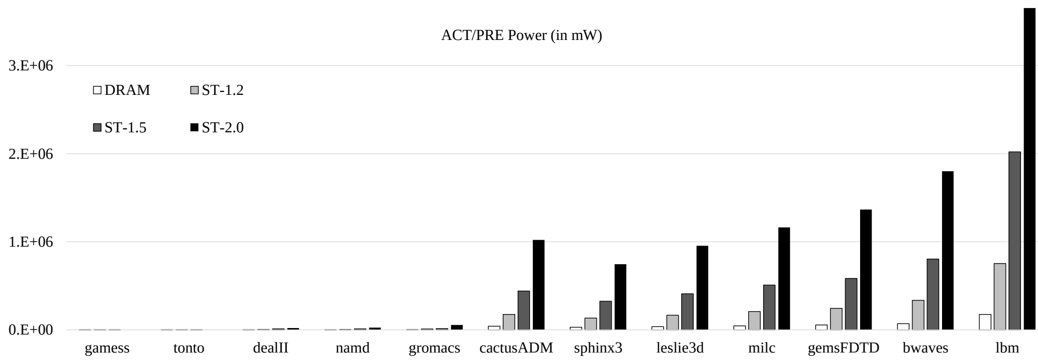

Fig. 10. Activate/Pre-charge power consumption of floating point benchmarks

ST-1.5 results, see Figure 6, show performance degradation for most integer benchmarks, 0.5% with open page policy and 1.1% with close page policy on average. However, the benchmarks `h264ref` and `perlbench` with close page policy, as well as `xalancbmk` and `hmmer` with open page policy, still experience a speedup on the STT-MRAM memory systems. Floating point benchmarks, experience higher slowdowns, 2.5% with open page policy and 2.8% with close page policy on average and around 10% in the worst case (`lbm`) for both policies, see Figure 7.

For configuration ST-2.0, all benchmarks experience slowdown w.r.t. to the DRAM. With open page policy slowdown of the integer benchmarks ranges between 0.6% (`h264ref`) and 14.1% (`mcf`), and it is 5.4% on average. With close page policy the slowdown of the integer benchmarks ranges between 0.2% (`h264ref`) and 15.8% (`mcf`), and it is 6.5% on average , see Figure 8. Floating point benchmarks are even more sensitive to the delays realized in the ST-2.0 configuration, see Figure 9. Five of the benchmarks experience a slowdown of less than 2% with both policies, but for the remaining ones the slowdown ranges between 11.4% (`sphinx3`) and 26.9% (`lbm`) with open page policy; and 12% (`sphinx3`) and 29.6% (`lbm`) with close page policy, leading to the average slowdown of 11.3% and 11.9% for open and close page policy, respectively.

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 set to be twice as slow as DRAM, the system performance degrades only by an average of 5.4% and 11.3% for integer and floating point benchmarks (open page policy), respectively. For the ST-1.5 configuration, the STT-MRAM main memory based system experiences an average slowdown of only 0.5% for the integer and 2.5% for the floating point benchmarks (open page policy). For STT-MRAM main memory with the ST-1.2 configuration, we actually measure a speedup w.r.t. to DRAM.

4.2 Power consumption

In this section, we present comparative power measurements of DRAM and STT-MRAM configurations of the main memory. We model DRAM and STT-MRAM power with the current parameters listed in Table 3. We incorporate these parameters into DRAMSim2 simulation infrastructure to obtain power measurements in three groups: *Activation and Pre-charge power*, *Burst Power* and *Background Power*. *Activation and Pre-charge power* corresponds to the power consumption to

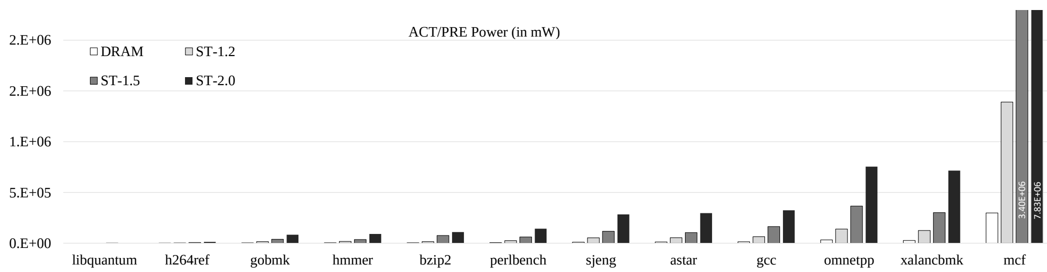

Fig. 11. Activate/Pre-charge power consumption of integer benchmarks

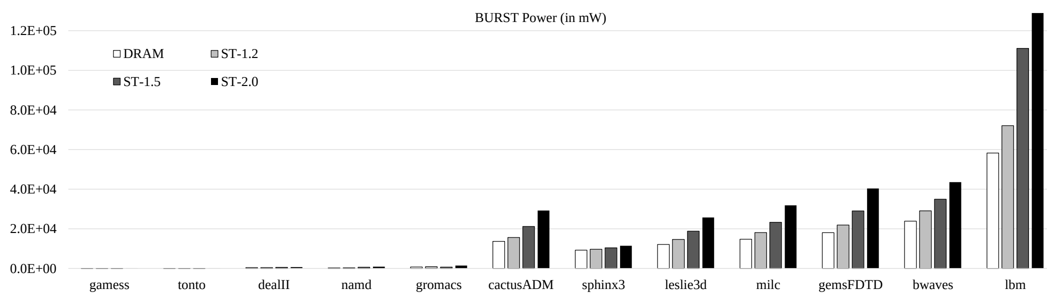

Fig. 12. Burst (Read/Write) power consumption of floating point benchmarks

activate/pre-charge rows, *Burst Power* refers to the power consumption for read/write activities, and *Background Power* relates to all other power consumption (excluding *Refresh Power* [4]).

Figure 10 to 15 present overall power consumption for floating point and integer benchmarks under study. In these figures, X-axis lists the power consumption in mili-Watts (mW) for DRAM and three configurations of STT-MRAM while Y-axis lists the benchmarks.

Figure 10 and 11 shows power consumption due to activation/pre-charge activity for floating point and integer benchmarks, respectively. The results suggest that, *activation and pre-charge power* consumption significantly increases for STT-MRAM configurations w.r.t DRAM. For floating point benchmarks, the average ACT/PRE power consumption increases 335.2% (4.3×) for ST-1.2 configuration, 983.5% (10.7×) for ST-1.5 configuration and 2445.2% (25.4×) for ST-2.0 configuration. For integer benchmarks, the average ACT/PRE power consumption increases 361.7% (4.6×) for ST-1.2 configuration, 1094.5% (12×) for ST-1.5 configuration and 2502.56% (25.8×) for ST-2.0 configuration. This is expected since, this operation activates rows, and the access mechanism of STT-MRAM (current-mode operation) cells are significantly different in comparison to DRAM (voltage-mode operation).

Figure 12 and 13 report power consumption originating from read/write activities for floating point and integer benchmarks, respectively. *Burst power* is moderately increased for STT-MRAM configurations in comparison to DRAM. For floating point benchmarks, burst power consumption increases on an average of 17.1% for ST-1.2 configuration, 56.2% for ST-1.5 configuration and 117.6% for ST-2.0 configuration. For integer benchmarks, burst power consumption increases on an average of 25.1% for ST-1.2 configuration, 72.9% for ST-1.5 configuration and 171.8% for ST-2.0 configuration.

[4]We do not compare *Refresh Power* since STT-MRAM does not require refresh and its corresponding current parameters are set to 0.

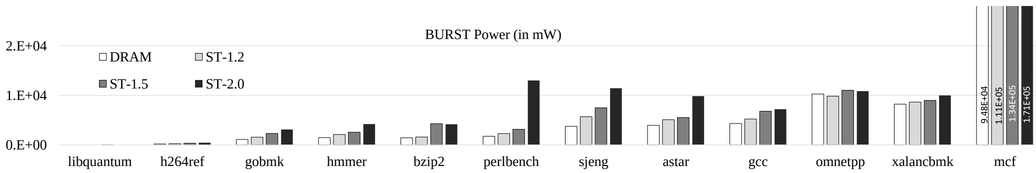

Fig. 13. Burst (Read/Write) power consumption of integer benchmarks

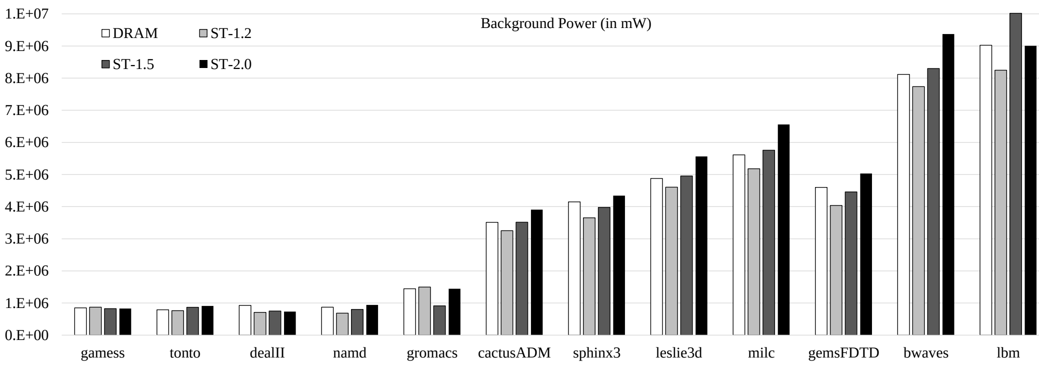

Fig. 14. Background power consumption of floating point benchmarks

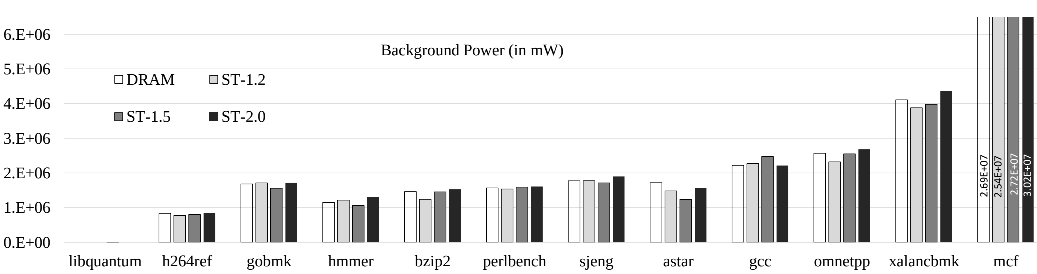

Fig. 15. Background power consumption of integer benchmarks

Figure 14 and 15 represents *background power* for all benchmarks under study. The results suggest that background power consumption does not deviate much from DRAM to STT-MRAM systems. For floating point benchmarks, background power consumption decreases on an average of 8.4% for ST-1.2 configuration, 3.8% for ST-1.5 configuration and increases 5.9% for ST-2.0 configuration. For integer benchmarks, background power consumption decreases on an average of 4.44% for ST-1.2 configuration, 3.6% for ST-1.5 configuration and increases 4% for ST-2.0 configuration.

5 STT-MRAM OPPORTUNITIES

Some of the STT-MRAM main memory advantages were already analyzed in the context of other non-volatile memory technologies and other application domains [38] [7] [69]. Here, we briefly summarize the ones that are of main interest in the HPC domain.

5.1 DRAM refresh

In a DRAM cell, the information is stored as a charge in small capacitors that have to be refreshed periodically in order to preserve the content. DRAM refresh degrades system performance because it interferes with application memory accesses. Also, refresh increases energy consumption, directly, because refresh operations consume energy, and indirectly, because degradation of system performance increases execution time, and therefore overall energy consumption. Performance and energy overheads of DRAM refresh have been deemed insignificant for long, but with the increasing frequency and size of DRAM devices, refresh becomes an important factor of memory performance and energy consumption. In state-of-the-art 32 Gb DRAM devices (specified in the DDR4 standard), refresh contributes to more than 20% of the DRAM energy consumption and degrades the memory bandwidth by more than 30%. In the upcoming 64 Gb devices, it is estimated that refresh will degrade memory throughput and increase the energy consumption by more than 50% [5][41]. STT-MRAM is a non-volatile technology and, therefore it requires no refresh. Thus, a great performance and energy advantage over the DRAM technology can come from resolving the memory refresh problem.

5.2 Memory errors

One of the leading causes of hardware failures in modern HPC clusters are main memory DRAM errors [58][21][62][63]. In the future, DRAM errors will pose an even larger threat to the reliability of HPC systems. First, the number of memory errors will increase because the amount of DRAM in HPC systems keeps growing at a consistent rate [21]. Another source of increasing the memory error rate is the scaling of the DRAM technology [34]. DRAM cells are getting smaller and they hold a decreasing amount of charge, which makes them more vulnerable to any disturbance and data corruption. Also, the distance between DRAM elements is already so small that electromagnetic coupling causes undesired interactions between the adjacent cells. STT-MRAM is a non-volatile 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.

Clearly, STT-MRAM is not a failure-free technology. However, its retention failures (stochastic in nature), can be efficiently controlled with proper ECC mechanisms [3]. A Recent study of Pajouhi *et al.* [51] analyzes the interactions between device parameters, bit-cell level parameters and different ECCs to optimize the robustness and energy-efficiency of a STT-MRAM cache. A compelling follow up would be to extend this work on STT-MRAM main memory.

5.3 Check-pointing

HPC applications consist of thousands of tightly-coupled processes, so failure of a single process results in a global failure that terminates the whole application. Therefore, system reliability is a first-class requirement, mainly because of a costly re-execution of long-running HPC jobs. In large HPC systems, fault tolerance is usually provided by the *checkpoint-restart* approach. Checkpointing periodically saves the system recovery information into a persistent storage. In case that a failure occurs, instead of repeating the whole experiment, the recovery information is used to *restart* the application from the last checkpoint [12]. Checkpoint-restart comes at an additional cost because it interrupts the normal execution of the applications, and puts significant stress to the interconnect network and the IO storage. In current HPC systems, between 15% and 45% of the time is spent on checkpointing, restarting and partial re-computation of the work lost since the last checkpoint [8][9]. This overhead will increase with the size of HPC systems, and it is estimated that on a 100,000-node cluster, checkpoint-restart activities will require 65% of the overall machine time [15]. One of the main sources of checkpoint-restart overheads is moving large amounts of data through the network to the remote storage [50]. In HPC systems with the STT-MRAM main memory, the memory system itself can be used to permanently store system recovery information. In order to prevent the system crashes in case that the whole server fails (e.g. due to power supply failure), system recovery information could be stored in main memory of the near-by servers. This would remove the pressure from the interconnect network and storage system and therefore would significantly reduce the checkpointing overhead. Low-overhead checkpoint-restart would also enable advanced techniques in batch scheduling, because it would relax the requirement that the jobs are executed continuously and without interruption, from the beginning until the end [17]. In order to improve the system utilization, the scheduler could perform a checkpoint of a given job, put it on hold and then restart it at an appropriate moment. This could, for example, allow system shutdown without queue draining period, or running large jobs in off-peak hours without the need to limit their execution time.

6 RELATED WORK

6.1 STT-MRAM main memory

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

Meza *et al.* [30] 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.* [37] evaluate STT-MRAM as a main memory for SPEC CPU2006 workloads and show that, without any optimizations, early-design STT-MRAM [65] 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.* [67] 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.* [2] 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 [54][70].

Jiang *et al.* [28] 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.* [71] using CACTI [45] and NVSim [11].

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.* [40] 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.* [74] 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.* [42] 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 [59][39][29][66]. Li *et al.* [39] 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.* [29] 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.* [59] propose a ultra-low retention time STT-MRAM caches supported by a DRAM-like refresh policy. Sun *et al.* [66] further exploit the scenario by deploying STT-MRAM with multiple retention levels. Smullen *et al.* [59] and Sun *et al.* [66] 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 [45]. 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 2.3. Recent studies have continued to investigate STT-MRAM's suitability as the last level cache (LLC) for different domains (e.g. from embedded systems to high performance computing) [36][20][55].

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 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 DDR*x* 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 consistent 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 apply similar methodology to estimate current parameters of STT-MRAM. We identify four current parameters that would change for STT-MRAM w.r.t DRAM and we perform a sensitivity analysis on these parameters to achieve an estimation of STT-MRAM power consumption.

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 systems. The results of our simulations show that, for the most realistic (considering the ongoing development) configuration *ST-1.2*, the SPEC 2006 benchmarks suffers an average system slowdown of less than 3%. Results from the power estimation indicates that STT-MRAM power consumption increases significantly for *Activation and Pre-charge power* while *Burst Power* increases moderately and *Background Power* does not deviate much from DRAM.

An intensified effort of memory manufacturers in STT-MRAM research promises exciting developments on this technology in near future. Now, with the reliable detailed 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 the by the Spanish Government (contract PID2019-107255GB) and Generalitat de Catalunya (contracts 2017-SGR-1328 and 2017-SGR-1414). The authors wish to thank Terry Hulett, Duncan Bennett and Ben Cooke from Everspin Technologies Inc., for their technical support.

REFERENCES

[1] K. Abe, H. Noguchi, E. Kitagawa, N. Shimomura, J. Ito, and S. Fujita. 2012. Novel Hybrid DRAM/MRAM Design for Reducing Power of High Performance Mobile CPU. In *IEEE International Electron Devices Meeting (IEDM).*

[2] Kazi Asifuzzaman, Milan Pavlovic, Milan Radulovic, David Zaragoza, Ohseong Kwon, Kyung-Chang Ryoo, and Petar Radojković. 2016. 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)).*

[3] C. Augustine, A. Raychowdhury, D. Somasekhar, J. Tschanz, K. Roy, and V.K. De. 2010. Numerical Analysis of Typical STT-MTJ Stacks for 1T-1R Memory Arrays. In *IEEE International Electron Devices Meeting (IEDM).*

[4] S. Ben Dodo, R. Bishnoi, and M. B. Tahoori. 2020. Secure STT-MRAM Bit-Cell Design Resilient to Differential Power Analysis Attacks. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 28, 1 (2020), 263–272.

[5] Ishwar Bhati, Mu-Tien Chang, Zeshan Chishti, Shih-Lien Lu, and Bruce Jacob. 2015. DRAM Refresh Mechanisms, Penalties, and Trade-Offs. In *IEEE Transactions on Computers.*

[6] R. Bishnoi, M. Ebrahimi, F. Oboril, and M. B. Tahoori. 2014. Asynchronous Asymmetrical Write Termination (AAWT)

for a low power STT-MRAM. In *2014 Design, Automation Test in Europe Conference Exhibition (DATE)*. 1–6.

[7] A.M. Caulfield, J. Coburn, T. Mollov, A. De, A. Akel, Jiahua He, A. Jagatheesan, R.K. Gupta, A. Snavely, and S. Swanson. 2010. Understanding the Impact of Emerging Non-Volatile Memories on High-Performance, IO-Intensive Computing. In *International Conference for High Performance Computing, Networking, Storage and Analysis (SC)*.

[8] John T. Daly. 2007. ADTSC Nuclear Weapons Highlights: Facilitating High Throughput ASC Calculations. *Technical Report LALP-07-041, Los Alamos National Laboratory, Los Alamos, NM, USA* (2007).

[9] John T. Daly, Lori A. Pritchett-Sheats, and Sarah E. Michalak. 2008. Application MTTFE vs. Platform MTTF: A Fresh Perspective on System Reliability and Application Throughput for Computations at Scale. In *Workshop on Resiliency in High Performance Computing*.

[10] B. Dieny, V. S. Speriosu, S. S. P. Parkin, B. A. Gurney, D. R. Wilhoit, and D. Mauri. 1991. Giant magnetoresistive in soft ferromagnetic multilayers. *Phys. Rev. B* (1991).

[11] X. Dong, C. Xu, Y. Xie, and N. P. Jouppi. 2012. 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 (2012), 994–1007.

[12] E. N. (Mootaz) Elnozahy, Lorenzo Alvisi, Yi-Min Wang, and David B. Johnson. 2002. A Survey of Rollback-recovery Protocols in Message-passing Systems. *ACM Comput. Surv.* (2002).

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

[14] Everspin Technologies, Inc. 2016. 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).

[15] Kurt Ferreira, Rolf Riesen, Ron Oldfield, Jon Stearley, James Laros, Kevin Pedretti, Todd Kordenbrock, and Ron Brightwell. 2009. Increasing Fault Resiliency in a Message-Passing Environment. *Sandia National Laboratories, Tech. Rep.* (2009).

[16] D. Halupka, S. Huda, W. Song, A. Sheikholeslami, K. Tsunoda, C. Yoshida, and M. Aoki. 2010. Negative-Resistance Read and Write Schemes for STT-MRAM in 0.13um CMOS. In *IEEE International Solid State Circuits Conference*.

[17] Paul H Hargrove and Jason C Duell. 2006. Berkeley Lab Checkpoint/Restart (BLCR) for Linux Clusters. *Journal of Physics* (2006).

[18] John L. Henning. 2006. SPEC CPU2006 Benchmark Descriptions. *SIGARCH Comput. Archit. News* (2006).

[19] M. Hosomi, H. Yamagishi, T. Yamamoto, K. Bessho, Y. Higo, K. Yamane, H. Yamada, M. Shoji, H. Hachino, C. Fukumoto, H. Nagao, and H. Kano. 2005. A Novel Nonvolatile Memory with Spin Torque Transfer Magnetization Switching: Spin-RAM. In *IEEE International Electron Devices Meeting*.

[20] G. Hu, J. J. Nowak, M. G. Gottwald, S. L. Brown, B. Doris, C. P. D'Emic, P. Hashemi, D. Houssameddine, Q. He, D. Kim, J. Kim, C. Kothandaraman, G. Lauer, H. K. Lee, N. Marchack, M. Reuter, R. P. Robertazzi, J. Z. Sun, T. Suwannasiri, P. L. Trouilloud, S. Woo, and D. C. Worledge. 2019. Spin-transfer torque MRAM with reliable 2 ns writing for last level cache applications. In *2019 IEEE International Electron Devices Meeting (IEDM)*. 2.6.1–2.6.4.

[21] Andy A. Hwang, Ioan Stefanovici, and Bianca Schroeder. 2012. Cosmic Rays Don't Strike Twice: Understanding the Nature of DRAM Errors and the Implications for System Design. In *17th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*.

[22] S. Ikeda, K. Miura, H. Yamamoto, K. Mizunuma, H. D. Gan, M. Endo, S. Kanai, J. Hayakawa, F. Matsukura, and H. Ohno. 2010. A perpendicular-anisotropy CoFeB–MgO magnetic tunnel junction. In *Nature Materials*, Vol. 9. 721–724. Issue 9.

[23] Intel. 2015. Intel® 64 and IA-32 Architectures Optimization Reference Manual. (2015).

[24] Intel Corporation. 2017. Intel 64 and IA-32 Architectures Software Developer Manuals. (2017).

[25] ITRS. 2015. 2013 International Technology Roadmap for Semiconductors. http://www.itrs.net/Links/2013ITRS/Home2013.htm. (2015).

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

[27] Karthick Rajamani Jente Kuang Kyu-Hyoun Kim Hillery Hunter Luis Lastras Xiaochen Guo Janani Mukundan, Alexandre Ferreira. 2018. Design of High-performance, Resilient, STT-MRAM-based Main Memory. https://domino.research.ibm.com/library/cyberdig.nsf/papers/83C96B8F98D488A9852582B20043C181. (2018).

[28] Lei Jiang, Wujie Wen, D. Wang, and L. Duan. 2016. 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)*.

[29] A. Jog, A.K. Mishra, Cong Xu, Yuan Xie, V. Narayanan, R. Iyer, and C.R. Das. 2012. Cache Revive: Architecting Volatile STT-RAM Caches for Enhanced Performance in CMPs. In *49th ACM/EDAC/IEEE Design Automation Conference (DAC)*.

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

[31] J. A. Katine, F. J. Albert, R. A. Buhrman, E. B. Myers, and D. C. Ralph. 2000. Current-Driven Magnetization Reversal and Spin-Wave Excitations in Co /Cu /Co Pillars. *Phys. Rev. Lett.* 84 (2000), 3149–3152. Issue 14.

[32] C. Kim, D. Kang, H. Kim, C.W. Park, D.H. SOHN, Y.S. Lee, S. Kang, H.R. Oh, and S. Cha. 2013a. Magnetic random access memory. (2013). https://www.google.es/patents/US20130311717 US Patent App. 13/768,858.

[33] H. Kim, S.K. Kang, D.H. SOHN, D.M. Kim, and K.C. Lee. 2013b. Magneto-resistive memory device including source line voltage generator. (2013). https://www.google.es/patents/US20130329489 US Patent App. 13/832,101.

[34] Yoongu Kim, R. Daly, J. Kim, C. Fallin, Ji Hye Lee, Donghyuk Lee, C. Wilkerson, K. Lai, and O. Mutlu. 2014. Flipping Bits in Memory without Accessing them: An Experimental Study of DRAM Disturbance Errors. In *ACM/IEEE 41st International Symposium on Computer Architecture (ISCA).*

[35] Peter Kogge, Keren Bergman, Shekhar Borkar, Dan Campbell, William Carlson, William Dally, Monty Denneau, Paul Franzon, William Harrod, Kerry Hill, Jon Hiller, Sherman Karp, Stephen Keckler, Dean Klein, Robert Lucas, Mark Richards, Al Scarpelli, Steven Scott, Allan Snavely, Thomas Sterling, R. Stanley Williams, and Katherine Yelick. 2008. ExaScale Computing Study: Technology Challenges in Achieving Exascale Systems. (Sept. 2008).

[36] M. Komalan, S. Sakhare, T. H. Bao, S. Rao, W. Kim, C. Tenllado, J. I. Gómez, G. S. Kar, A. Furnemont, and F. Catthoor. 2017. Cross-layer design and analysis of a low power, high density STT-MRAM for embedded systems. In *2017 IEEE International Symposium on Circuits and Systems (ISCAS).* 1–4.

[37] E. Kultursay, M. Kandemir, A. Sivasubramaniam, and O. Mutlu. 2013. Evaluating STT-RAM as an Energy-Efficient Main Memory Alternative. In *IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS).*

[38] Dong Li, Jeffrey S. Vetter, Gabriel Marin, Collin McCurdy, Cristian Cira, Zhuo Liu, and Weikuan Yu. 2012. Identifying Opportunities for Byte-Addressable Non-Volatile Memory in Extreme-Scale Scientific Applications. In *Proceedings of the 26th IEEE International Parallel and Distributed Processing Symposium (IPDPS).*

[39] Hai Li, Xiaobin Wang, Zhong-Liang Ong, Weng-Fai Wong, Yaojun Zhang, Peiyuan Wang, and Yiran Chen. 2011a. Performance, Power, and Reliability Tradeoffs of STT-RAM Cell Subject to Architecture-Level Requirement. *IEEE Transactions on Magnetics* (2011).

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

[41] Jamie Liu, Ben Jaiyen, Richard Veras, and Onur Mutlu. 2012. RAIDR: Retention-Aware Intelligent DRAM Refresh. In *39th Annual International Symposium on Computer Architecture.*

[42] M. T. Chang and P. Rosenfeld and S. L. Lu and B. Jacob. 2013. 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.*

[43] Clémentine Maurice, Nicolas Le Scouarnec, Christoph Neumann, Olivier Heen, and Aurélien Francillon. 2015. *Reverse Engineering Intel Last-Level Cache Complex Addressing Using Performance Counters.*

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

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

[46] R. Nebashi, N. Sakimura, H. Honjo, S. Saito, Y. Ito, S. Miura, Y. Kato, K. Mori, Y. Ozaki, Y. Kobayashi, N. Ohshima, K. Kinoshita, T. Suzuki, K. Nagahara, N. Ishiwata, K. Suemitsu, S. Fukami, H. Hada, T. Sugibayashi, and N. Kasai. 2009. A 90nm 12ns 32Mb 2T1MTJ MRAM. In *IEEE International Solid-State Circuits Conference.*

[47] H. Noguchi, K. Kushida, K. Ikegami, K. Abe, E. Kitagawa, S. Kashiwada, C. Kamata, A. Kawasumi, H. Hara, and S. Fujita. 2013. 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).*

[48] J. J. Nowak, R. P. Robertazzi, J. Z. Sun, G. Hu, J. H. Park, J. Lee, A. J. Annunziata, G. P. Lauer, R. Kothandaraman, E. J. O'Sullivan, P. L. Trouilloud, Y. Kim, and D. C. Worledge. 2016. Dependence of Voltage and Size on Write Error Rates in Spin-Transfer Torque Magnetic Random-Access Memory. *IEEE Magnetics Letters* 7 (2016), 1–4.

[49] H.R. Oh. 2014. Resistive Memory Device, System Including the Same and Method of Reading Data in the Same. (2014). https://www.google.es/patents/US20140169069 US Patent App. 14/094,021.

[50] R.A. Oldfield, S. Arunagiri, P.J. Teller, S. Seelam, M.R. Varela, R. Riesen, and P.C. Roth. 2007. Modeling the Impact of Checkpoints on Next-Generation Systems. In *24th IEEE Conference on Mass Storage Systems and Technologies.*

[51] Zoha Pajouhi, Xuanyao Fong, and Kaushik Roy. 2015. Device/Circuit/Architecture Co-design of Reliable STT-MRAM. In *Proceedings of the Design, Automation & Test in Europe Conference & Exhibition.*

[52] M. Poremba and Y. Xie. 2012. NVMain: An Architectural-Level Main Memory Simulator for Emerging Non-volatile Memories. In *2012 IEEE Computer Society Annual Symposium on VLSI.*

[53] K. Rho, K. Tsuchida, D. Kim, Y. Shirai, J. Bae, T. Inaba, H. Noro, H. Moon, S. Chung, K. Sunouchi, J. Park, K. Park, A. Yamamoto, S. Chung, H. Kim, H. Oyamatsu, and J. Oh. 2017. 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).*

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

[55] S. Sakhare, M. Perumkunnil, T. H. Bao, S. Rao, W. Kim, D. Crotti, F. Yasin, S. Couet, J. Swerts, S. Kundu, D. Yakimets, R. Baert, H. Oh, A. Spessot, A. Mocuta, G. S. Kar, and A. Furnemont. 2018. Enablement of STT-MRAM as last level cache for the high performance computing domain at the 5nm node. In *2018 IEEE International Electron Devices Meeting (IEDM)*. 18.3.1–18.3.4.

[56] Samsung Electronics Co., Ltd. 2011. 240pin Registered DIMM based on 2Gb D-die. https://www.samsung.com/semiconductor/global.semi/file/ resource/2017/11/ ds_ddr3_2gb_d-die_based_1_35v _rdimm_rev12-2.pdf. (2011).

[57] Daniel Sanchez and Christos Kozyrakis. 2013. ZSim: Fast and Accurate Microarchitectural Simulation of Thousand-core Systems. In *Proceedings of the 40th Annual International Symposium on Computer Architecture (ISCA)*.

[58] Bianca Schroeder, Eduardo Pinheiro, and Wolf-Dietrich Weber. 2009. DRAM Errors in the Wild: A Large-scale Field Study. In *11th International Joint Conference on Measurement and Modeling of Computer Systems*.

[59] C.W. Smullen, V. Mohan, A. Nigam, S. Gurumurthi, and M.R. Stan. 2011. Relaxing non-volatility for fast and energy-efficient STT-RAM caches. In *IEEE 17th International Symposium on High Performance Computer Architecture (HPCA)*.

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

[61] J.K. Spong, Speriosu, Robert E. Fontana, Moris M. Dovek, and T.L. Hylton. 1996. Giant Magnetoresistive Spin Valve Bridge Sensor. *IEEE Transactions on Magnetics* 32, 2 (1996), 366–371.

[62] Vilas Sridharan and Dean Liberty. 2012. A Study of DRAM Failures in the Field. In *International Conference on High Performance Computing, Networking, Storage and Analysis*.

[63] Vilas Sridharan, Jon Stearley, Nathan DeBardeleben, Sean Blanchard, and Sudhanva Gurumurthi. 2013. Feng Shui of Supercomputer Memory: Positional Effects in DRAM and SRAM Faults. In *International Conference on High Performance Computing, Networking, Storage and Analysis*.

[64] Rick Stevens, Andy White, Pete Beckman, Ray Bair-ANL, Jim Hack, Jeff Nichols, Al GeistORNL, Horst Simon, Kathy Yelick, John Shalf-LBNL, Steve Ashby, Moe Khaleel-PNNL, Michel McCoy, Mark Seager, Brent Gorda-LLNL, John Morrison, Cheryl Wampler-LANL, James Peery, Sudip Dosanjh, Jim Ang-SNL, Jim Davenport, Tom Schlagel, BNL, Fred Johnson, and Paul Messina. 2010. 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).

[65] Guangyu Sun, Xiangyu Dong, Yuan Xie, Jian Li, and Yiran Chen. 2009. A Novel Architecture of the 3D Stacked MRAM L2 Cache for CMPs. In *IEEE 15th International Symposium on High Performance Computer Architecture (HPCA)*.

[66] Zhenyu Sun, Xiuyuan Bi, Hai (Helen) Li, Weng-Fai Wong, Zhong-Liang Ong, Xiaochun Zhu, and Wenqing Wu. 2011. Multi Retention Level STT-RAM Cache Designs with a Dynamic Refresh Scheme. In *44th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*.

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

[68] Top500. 2017. Top500 Supercomuter Sites. http://www.top500.org/. (2017).

[69] Jeffrey S. Vetter and Sparsh Mittal. 2015. Opportunities for Nonvolatile Memory Systems in Extreme-Scale High Performance Computing. *Computing in Science and Engineering special issue* (2015).

[70] David Wang, Brinda Ganesh, Nuengwong Tuaycharoen, Kathleen Baynes, Aamer Jaleel, and Bruce Jacob. 2005. DRAMsim: A Memory System Simulator. *SIGARCH Comput. Archit. News* 33, 4 (2005).

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

[72] Wm. A. Wulf and Sally A. McKee. 1995. Hitting the Memory Wall: Implications of the Obvious. *SIGARCH Comput. Archit. News* (1995).

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

[74] Ping Zhou, Bo Zhao, Jun Yang, and Youtao Zhang. 2009. Energy Reduction for STT-RAM Using Early Write Termination. In *IEEE/ACM International Conference on Computer-Aided Design - Digest of Technical Papers*.