STT-MRAM for Real-Time Embedded Systems: Performance and WCET Implications

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

Mikel Fernandez
Barcelona Supercomputing Center
Barcelona, Spain

Petar Radojković
Barcelona Supercomputing Center
Barcelona, Spain

Jaume Abella
Barcelona Supercomputing Center
Barcelona, Spain

Francisco J. Cazorla
Barcelona Supercomputing Center
Barcelona, Spain

ABSTRACT

STT-MRAM is an emerging non-volatile memory quickly approaching DRAM in terms of capacity, frequency and device size. Intensified efforts in STT-MRAM research by the memory manufacturers may indicate a revolution with STT-MRAM memory technology is imminent, and therefore it is essential to perform system level research to explore use-cases and identify computing domains that could benefit from this technology. Special STT-MRAM features such as intrinsic radiation hardness, non-volatility, zero stand-by power and capability to function in extreme temperatures makes it particularly suitable for aerospace, avionics and automotive applications. Such applications often have real-time requirements — that is, certain tasks must complete within a strict deadline. Analyzing whether this *deadline* is met requires Worst Case Execution Time (WCET) Analysis, which is a fundamental part of evaluating any real-time system. In this study, we investigate the feasibility of using STT-MRAM in real-time embedded systems by analyzing average system performance impact and WCET implications.

CCS CONCEPTS

• **Computer systems organization** → *Processors and memory architectures*; *Embedded systems*; • **Hardware** → Non-volatile memory; • **Computing methodologies** → **Real-time simulation**;

1 INTRODUCTION

For decades, DRAM devices have been the dominant building blocks for main memory systems in most computing systems. However, it is unclear whether this technology will continue to meet the needs of next-generation memory systems due to its reliability issues induced from extreme scaling [1]. Therefore, significant efforts have been 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 fourteen years ago [2], 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 [3]. Therefore, now it is the time to perform system level research to explore use-cases and identify computing domains that could benefit from this technology.

Special STT-MRAM features such as intrinsic radiation hardness, non-volatility, zero stand-by power and capability to function in extreme temperatures makes it particularly suitable for aerospace, avionics and automotive domains, in which applications often have real-time requirements – that is, certain tasks must complete within a strict deadline. Analyzing whether this *deadline* is met requires Worst Case Execution Time (WCET) Analysis, which is a fundamental part of evaluating any real-time system. Therefore, it is crucial to ensure STT-MRAM's correct functionality and capability of complying with the timing requirements of real time applications through system level analysis.

We quantify the performance impact of replacing conventional DRAM with the STT-MRAM main memory. In this study, we investigate the feasibility of using STT-MRAM in real-time embedded systems by analyzing STT-MRAM main memory impact on average system performance and WCET. To the best of our knowledge, this is the first study which provides a comprehensive insight on STT-MRAM's performance and WCET implications as the *main memory*, specifically on real-time embedded systems.

We focus on a Cobham Gaisler's NGMP like architecture [4] as a representative multicore processor. In a validated simulator, we model STT-MRAM main memory with recently-published detailed timing parameters that are supported by a leading STT-MRAM manufacturer. STT-MRAM's suitability for the real-time embedded systems is validated on benchmarks provided by the European Space Agency (ESA) and EEMBC Autobench suite [5] by analyzing performance and WCET impact. To portray a broader spectrum of scenarios, we further extend the scope of our experiments executing additional benchmarks with high memory utilization from Mediabench [6]. For all applications under study, we also investigate performance with *deterministic* and *probabilistic* platforms.

Our results show that systems comprising STT-MRAM main memory can be analyzed with the same WCET approaches used in the systems with conventional DRAM. This is a compelling finding, as it is fundamental to reduce STT-MRAM adoption costs, without requiring new tools that must undergo costly qualification process [7]. In quantitative terms, our results show that STT-MRAM main memory in real-time embedded systems provides performance and WCET comparable to conventional DRAM, while opening up opportunities to exploit various advantages.

The rest of the paper is organized as follows. Section 2 introduces STT-MRAM technology, its development trend in recent years, its organization and CPU interface and special advantages. Section 3

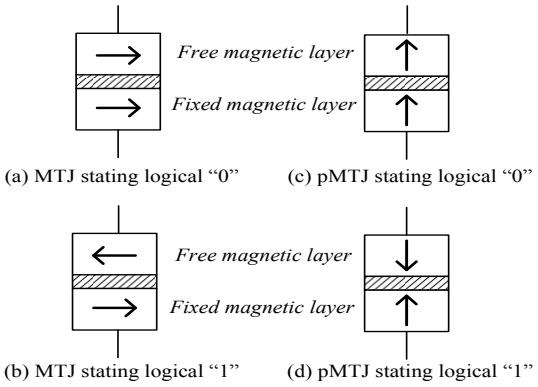

Figure 1: STT-MRAM cell

describes the experimental environment used in the study. Section 4 shows results of performance analysis and Section 5 presents WCET implications of STT-MRAM and DRAM. Section 6 discusses the related works and Section 7 presents the conclusions of the study.

2 STT-MRAM

2.1 Technology overview

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 depicts, 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 [8].

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 [9][10]; see Figure 1(c) and (d).

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 [11], reducing the capacity gap between these two technologies from 2000× in 2003 to only 4× in 2016.

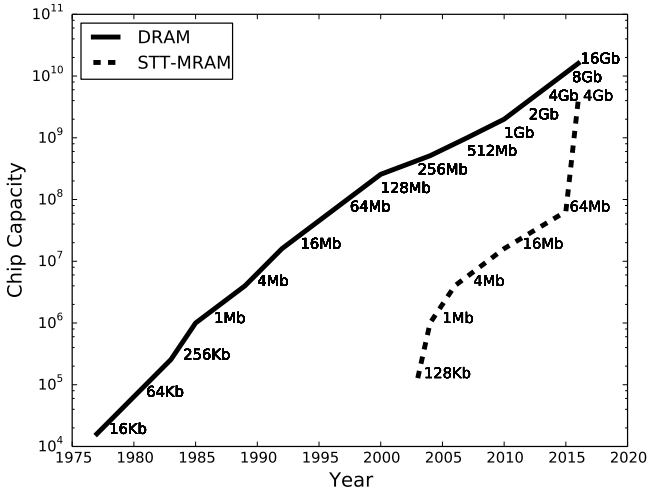

Figure 2: DRAM and STT-MRAM capacity growth in years

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 [12].

2.3 Organization and CPU interface

Although STT-MRAM is catching-up rapidly in terms of cell size, capacity and frequency, DRAM still has one great advantage — it is a standardized plug-and-play device. Today, we have various DRAM and CPU manufacturers and OEMs with full compatibility between devices — we can connect any CPU (Intel, AMD, ARM-based) to any DRAM (Samsung, Micron, Hynix) as long as they follow the same DDR*x* 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.

Industrial patents [13][14][15] suggest STT-MRAM manufacturers are adopting STT-MRAM technology into DDR*x* interface and protocols in order to enable a seamless integration into rest of the system. STT-MRAM data array structure is very similar to that of DRAM. 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 DRAM, in terms of number and size of the structures such as ranks, banks, sub-arrays, rows, columns, and row buffers. Finally, STT-MRAM CPU interface is DRAM compatible.

2.4 STT-MRAM special advantages

In this section we make a qualitative analysis of STT-MRAM specific features. In particular, intrinsic radiation hardness, non-volatility, zero stand-by power and capability to function in extreme temperatures offer a great opportunity to explore its usability in real-time embedded systems in aerospace and automotive domains, where computer systems must operate with guaranteed behavior on harsh

Table 1: DRAM vs STT-MRAM for embedded real-time systems

Feature	DRAM	STT-MRAM
Radiation-hard	-	+++
Standby power	-	+++
Temperature tolerance	+	+++
Storage capacity	++	++
Access speed	+++	++
Endurance	++	+++

environments under stringent constraints. A summary of the main differences between DRAM and STT-MRAM is provided in Table 1.

2.4.1 Radiation hardness. A *Single Event Upset (SEU)* occurs when a state of a memory cell or transistor is erroneously changed by the striking of a charged particle such as ions, photons or alpha particles [16]. Continuous scaling of CMOS devices has further amplified the chances of being affected, which is a serious concern for DRAM and SRAM main memories and caches, which account for a large fraction of the silicon in computing systems.

Microelectronic devices deployed in space are particularly vulnerable to such events. For instance, it has been reported that soft error rates due to radiation grow by a factor of 650× when moving from sea level to 12,000m of altitude [17], a usual altitude for commercial planes. Radiation in the space further exacerbates the issue due to the lack of the Earth atmosphere to mitigate radiation. These phenomena lead to increased bit upset rates that require expensive coding and scrubbing techniques to guarantee error correction.

STT-MRAM offers a promising solution to this problem as it replaces charge-based storage with Magnetic Tunnelling Junction (MTJ), which stores data in the form of magnetic resistance that is intrinsically tolerant to radiation. STT-MRAM memory chips have reportedly been deployed on space bound satellites [18].

2.4.2 Zero standby power. Electronic devices in aerospace and automotive domains are usually deployed once to be operated for a long period of time without regular maintenance. Due to its non-volatility, STT-MRAM also ensures that no data is lost if an unexpected power down or voltage drop takes place. Implementing appropriate measures, the operations can resume from the same point as it was interrupted. Also, STT-MRAM having long term data retention with zero standby power is set to offer great advantage from the power consumption perspective. For instance, many instruments in space missions are operated at a given (low) frequency, taking pictures or measurements every second or minute. STT-MRAM allows activating and deactivating systems with negligible power cost, without requiring any form of backup space.

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.

2.4.3 Operational Temperature. STT-MRAM's another crucial feature is being operational under an extended range of temperatures. One of the STT-MRAM manufacturers states that a Grade 1 qualified MRAM will contain data for 20 years being operational

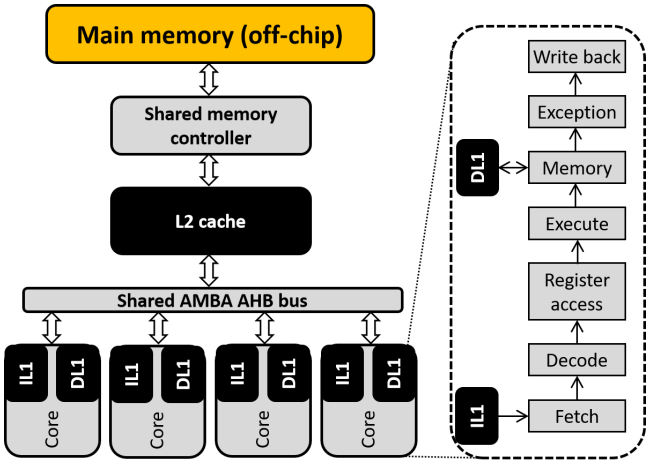

Figure 3: Schematic view of the Next Generation Microprocessor (NGMP)

under extreme temperatures ranging from -40°C to 125°C. [19]. This makes STT-MRAM suitable to be used both in aerospace (extreme cold) and automotive (occasionally extreme hot) parts.

2.4.4 Integrated memory. Having comparable speed and density to DRAM, with unlimited endurance and long retention time, STT-MRAM really opens up the opportunity to use it as single memory replacing DRAM and long term storage from the conventional real-time embedded systems.

3 EXPERIMENTAL ENVIRONMENT

In this section we describe the target hardware platform, simulator infrastructure, and benchmarks used to carry out the experiments.

3.1 Processor platform

For our experiments, we model a Cobham Gaisler LEON4 Next Generation Microprocessor (NGMP) [4]. The NGMP is targeted to be used for future space missions by the European Space Agency (ESA). It is a good representative of advanced real-time embedded processors, which start introducing multiple cores per processor. The most important features of NGMP CPU are summarized in Figure 3 and Table 3. Each core has a private 16 KiB L1 instruction and data cache, while a 256KiB L2 cache is shared among all four cores. The cores are connected through a 128-bit AHB AMBA bus arbitrated by a round-robin policy.

3.2 Main Memory platform

We model the STT-MRAM timings based on the parameters that are recently published in collaboration with Everspin Technologies Inc., one of the leading STT-MRAM manufacturers [20]. The published timing parameters enable a reliable methodology to simulate STT-MRAM without releasing confidential information about any product. We briefly summarize the procedure and reasoning of estimating the timing parameters.

Table 2: DRAM and STT-MRAM parameters associated with row operation (DDR2-667 cycles)

Timing Parameters	Description	DRAM	ST-1.2	ST-1.5	ST-2.0
tRCD	Row to column command delay	5	6	8	10
tRP	Row precharge	5	6	8	10
tFAW	Four row activation window	13	16	20	26
tRRD	Row to Row activation delay	3	4	5	6
tRFC	Refresh cycle time	43	0	0	0

STT-MRAM memory devices are DDR*x* compatible, with the same or very similar organization and CPU interface, as the conventional DRAM. Also, both, DRAM and STT-MRAM main memory devices use a row buffer as the interface between the cell-arrays and the memory bus. Since the circuitry beyond the row buffer for DRAM and STT-MRAM is essentially the same — once the data is in the row buffer, STT-MRAM timing parameters for the consequent operations are the same as DRAM. Therefore, the values of all the timing parameters that are not associated with row operations do not change from DRAM to STT-MRAM (e.g. tBURST, tCAS, tRTP, tWTR etc.).

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 STT-MRAM row operations would deviate from DRAM[1]. Since, there is no reliable information on how these timing parameters will change for the upcoming STT-MRAM devices. A sensitivity analysis is performed on timing parameters that would deviate from DRAM.

In this study, we select three sets of STT-MRAM timings, with 1.2×, 1.5× and 2× slower row-related operations w.r.t. DRAM, as summarized in Table 2. *ST-1.2* timing parameters are *optimistic* and would correspond to major enhancements of the STT-MRAM technology, while *ST-2.0* parameters are *pessimistic* estimations. Simulations performed with these timing parameters give us a reliable range of possible system performance impact for upcoming STT-MRAM main memory devices [20].

3.3 Simulation infrastructure: SoCLib & DRAMSim2

The NGMP processor is simulated with a SoCLib-based simulator [21]. The simulator is designed to conceptually separate the functional emulation from the timing behavior. Functional emulation executes the instructions according to a particular Instruction Set Architecture (ISA) and provides all the relevant information about the instruction execution, such as the instruction address, registers use, instruction type, result, etc. The timing simulator analyzes the timing behavior of instructions for a given hardware implementation, e.g., it determines the latency of load instructions. It is built in a modular way so each hardware component maps to a component of the timing simulator. This allows for extensions such as the addition of more accurate memory models.

[1]Rows of the DRAM or STT-MRAM cells constitute the cell arrays. Row operations access directly to the memory cells. Asifuzzaman et. al. [20] also detail the differences between the DRAM and STT-MRAM row operations.

Table 3: Next Generation Microprocessor (NGMP): Main features

Feature	Description
Cores	4
ISA	SPARC v8
Pipeline stages	Fetch, decode, register, execute, memory, exceptions, commit
Core Frequency	150 MHz
Superscalar	No
Out-of-Order	No
L1 D-cache	Private (per-core) 16KiB, 32 byte/line, 4-way Write-through, Write no-allocate
L1 I-cache	Private (per-core) 16KiB, 32 byte/line, 4-way
L2 cache	Shared: 4 cores Unified: Data and Instructions 256KiB, 32 byte/line, 4-way Copy-back, Write-allocate
FPU	Double precission IEEE-754

Simulator parameters used in the study have been previously verified [22] to accurately model the behavior of the GR-CPCI-LEON4-N2X [23], a board implementing the NGMP processor.

In this study we consider a system in which the NGMP CPU is connected to a DDR2-667 memory device. Both DRAM and STT-MRAM main memories are simulated with DRAMSim2 simulator [24], that is integrated with the SoCLib simulator through a fairly simple interface. DRAMSim2 is a cycle accurate model of a DRAM main memory validated against manufacturer Verilog models. For the simulation of the DRAM, we use the timing parameters from the automotive DDR2 SDRAM data-sheet provided by Micron Technology, Inc. [25] Estimation of the timing parameters for the STT-MRAM main memory is described in Section 3.2.

3.4 Timing analysis of real-time systems

Real-time embedded systems are subject to strict timing constraints as defined by applicable safety standards. Failing to meet specific deadlines for those systems may lead to fatal consequences, specially for the most (safety or mission) critical applications.

Since timing is a critical concern in real-time embedded systems, validation and certification of these systems requires sufficient evidence that tasks will complete within assigned time budgets, i.e. before specific deadlines. This evidence is typically provided using timing analysis techniques that estimate the Worst-Case Execution Time (WCET) of tasks running in the target system.

There are several approaches to perform the timing analysis from static timing analysis (STA) to measurement-based timing analysis (MBTA), each one with its own pros and cons [26]. STA is performed statically without executing the code [27]. It has been proven the most convenient solution for very simple microcontrollers, where accurate and reliable timing models of the processor can be built.

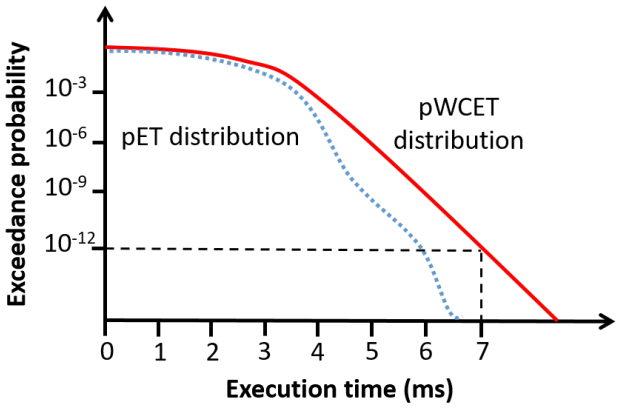

Figure 4: pWCET distribution: Probability (Y-axis) that the application execution time (in any given run) exceeds the corresponding time on the X-axis. In this example, the pWCET estimate (7ms) is exceeded with a probability of 10^{-12}.

However, STA faces severe limitations when considering complex hardware and software, which challenge the reliability of timing models and the tightness of timing bounds.

In this paper we focus on measurement-based timing analysis (MBTA) [28][29]. MBTA builds on the collection and operation of execution time measurements of the application running on the target platform. It is the most widely adopted solution by industry due to its relatively low cost of applicability. MBTA has been shown to provide trustworthy estimates for highest-criticality software in Avionics [30] when it runs on simple processors.

Increased hardware and software complexity, however, reduces the confidence that can be placed on WCET estimates derived with MBTA [27]. Statistical techniques have been studied for MBTA to derive bounds to execution time distributions. In particular, measurement-based probabilistic timing analysis (MBPTA) [31] has matured in recent years. MBPTA delivers a probabilistic WCET (pWCET) function that upper-bounds the (probabilistic) execution time distribution of the program (pET) at any exceedance probability, see Figure 4. MBPTA has been successfully applied to industrial case studies [32][33] and its impact on certification has been addressed [34].

3.5 Measurement-based probabilistic timing analysis (MBPTA)

MBPTA has been complemented with solutions that inject randomization in program's timing behavior to relieve the user from controlling those *jittery resources* affecting the execution time variability of a program. Randomization makes that the potential behavior that a given jittery resource (e.g. caches) can exhibit, is naturally (and randomly) explored in every new test, enabling the derivation of probabilistic guarantees. This is in contrast to *deterministic* platforms where randomization is not enabled and execution time is not expected to deviate. For the probabilistic platform, randomization has been implemented at hardware level (e.g. random arbitration

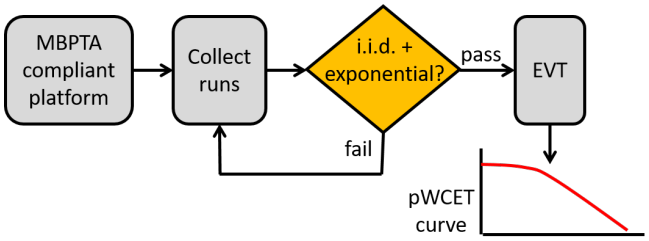

Figure 5: Schematic of the MBPTA application process

policies and random placement/re-placement techniques) that are now part of a commercial product for the space domain [35]; and with software techniques that work at the compiler/linker level [36] or source code level [37]. In order to enforce probabilistic guarantees to hold during operation, randomizations must be kept enabled, so that the execution time distribution analyzed matches (or upper-bounds) that during operation. Then, by using MBPTA on the collected execution measurements, reliable pWCET estimates are obtained. Figure 5 illustrates the process of obtaining pWCET estimates. In this work we build upon MBPTA-CV [31], a MBPTA technique whose implementation has been recently made publicly available [38]. In the following section we summarize the steps of the MBPTA process used in this study.

3.5.1 MBPTA compliant platform. MBPTA relies on the use of platforms with specific timing properties [39] – either provided by hardware or software means – that allow obtaining measurements at analysis that *represent* the behavior during operation. In particular, those platforms build upon time upperbounding and time randomization so that measurements at analysis correspond to a distribution (random variable) that upperbounds probabilistically the behavior during operation. By introducing those properties on the hardware/software platform, which has been proved to cause marginal performance degradation [39], collecting representative measurements has been shown to be independent of the use of complex hardware features such as cache hierarchies with intricate behavior (unified data/instruction caches, inclusive caches, etc) and multicores among others. In fact, MBPTA does not pose any explicit constraint on the use of *any* hardware feature as long as specific properties are met in the hardware/software platform and measurement collection process [40].

3.5.2 Collecting Runs. Once measurements are guaranteed to match or upperbound operation time behavior, MBPTA requires a sufficiently large execution time sample. In this study, each benchmark is executed for thousand times with each memory configuration (*DRAM*, *ST-1.2*, *ST-1.5* and *ST-2.0*). It takes few minutes to several hours (depending on the benchmark) to perform one execution.

3.5.3 Independent and identically distributed – i.i.d. test. After we accumulate sufficient execution time measurements from a probabilistic platform for a benchmark with a specific memory configuration, MBPTA-CV assesses whether the execution time measurement are statistically independent and identically distributed (i.i.d.) with appropriate tests. While the process measured is probabilistically

i.i.d. by construction (each benchmark is executed independently with an identical configuration), random samples might sporadically fail to achieve those properties statistically. For instance, we may roll a dice 6 times and obtain statistically non-i.i.d. samples (e.g. six times the same value). However, since the variable observed (execution time) is probabilistically i.i.d., whenever tests are failed, a larger sample needs to be collected since the sample will converge to the variable studied eventually.

3.5.4 Extreme value theory. Once the sample is accepted as i.i.d., MBPTA-CV builds upon Extreme Value Theory (EVT) [41] to deliver a probabilistic WCET (pWCET) estimate of the program. A pWCET estimate (an example is shown in Figure 4 for illustrative purposes) is a continuous function upperbounding the exceedance probability for any high execution time. pWCET functions are typically plotted in the form of a Complementary Cumulative Distribution Function (CCDF), also known as tail distributions, with logarithmic y-axis scale. Their intrepretation is such that the particular probability in the y-axis is an upperbound to the true exceedance probability of the execution time in the x-axis. The exceedance probability can be set arbitrarily low so that it can be deemed irrelevant w.r.t. the requirements of the function (e.g. below 10^{-8} failures per hour). Note that, contrarily to some people believings, WCET estimates can potentially be exceeded since even the most stringent timing analysis processes have some form of residual risk associated to the modelling of the hardware (for STA) or the measurement collection process (MBTA/MBPTA). Hence, upperbounds to the exceedance rates are compatible with vefirication and validation processes even for the most critical functions. In general, appropriate safety measures are designed along those critical functions to either set the system to a safe state on a failure (e.g. stopping the car) or to keep it operational by means of diverse redundancy so that a single fault cannot lead to a full-system failure. We refer the interested reader to the corresponding functional safety standards in each domain, such as ISO26262 in automotive [7] and DO178B in avionics [42]. In order to set an appropriate EVT distribution to the sample, MBPTA-CV builds upon the fact that execution times of real-time programs are finite. This guarantees that execution times can be upperbounded with exponential tails [31]. Hence, MBPTA-CV selects automatically those measurements that belong to the tail of the distribution from the sample, and tests their exponentiality. If the best fit can be rejected to be exponential or a light tail[2], then the sample does not have enough tail measurements and MBPTA-CV instructs the user to collect further measurements. Eventually, since the random variable (execution time) observed has a maximum value, this process converges and sufficient values of the tail will be collected, so they will be properly upperbounded with an exponential tail. Note that MBPTA-CV imposes, by construction of the method, that no less than 50 tail measurements can be accepted to fit the appropriate exponential distribution to the tail. Hence, not only a reliable tail model is obtained, but the confidence interval is necessarily narrow, thus preserving the tightness of the pWCET estimate.

[2]Light tails fall at a higher rate than exponential tails and approach a maximum value asymptotically, so they are naturally upperbounded by exponential tails.

Table 4: Benchmarks used in the study

Suite	Benchmarks	Domain
ESA Applications	obdp, debie	Space
EEMBC Autobench	a2time01, aifftr01, aifirf01, aiifft01, basefp01, bitmnp01, cacheb01, canrdr01, idctrn01, iirflt01, matrix01, pntrch01, puwmod01, rspeed01, tblook01, ttsprk01	Automotive
Mediabench	mesa.texgen, mesa.mipmap, epic.decode, mesa.osdemo	Media

3.5.5 pWCET estimate with confidence interval. The confidence interval, set to usual values in statistics (e.g. 90%, 95% or 99%) illustrates that tail fitting introduces low variability. In general, despite confidence intervals could be used to select the pWCET estimate, we stick to point estimation since the process already inherits some sources of pessimism that guarantee the need for upperbounding, such as the pessimism incurred by enforcing worst-case operation conditions for pWCET estimation, and using exponential tails instead of light tails[3].

3.6 Benchmarks

STT-MRAM's suitability for the real-time embedded systems is validated on benchmarks provided by the European Space Agency (ESA), EEMBC Autobench suite [5] and Mediabench [6]. Table 4 lists the benchmarks used in the study.

The European Space Agency provided two applications, Onboard Data Processing (`obdp`) and `debie`. `obdp` contains the algorithms used to process raw frames coming from the state-of-the-art near infrared (NIR) HAWAII-2RG detector, already used in production systems, such as the Hubble Space Telescope. `debie` is the software that controls an instrument that observes micro-meteoroids and small space debris. It has been already used in the PROBA-1 satellite.

EEMBC Autobench includes 16 benchmark kernels that mimic functionalities of production automotive, industrial, and general-purpose applications. General purpose kernels include bit-manipulation, multiplication, floating-point, matrix, cache and pointer chasing benchmarks, as well as the pulse-width modulation and shift operations typical of encryption algorithms.

In order to further investigate STT-MRAM's performance and WCET traits, we execute four more applications with high memory utilization [43] from Mediabench benchmark suite.

4 PERFORMANCE

In this section we analyze and discuss system performance traits with STT-MRAM in comparison to DRAM. For this experiment, we use a *deterministic* multi-core platform of the NGMP architecture.

We execute each application under study with DRAM and three sets of STT-MRAM timing configurations, namely *ST-1.2*, *ST-1.5* and *ST-2.0*. From their individual *execution time* and *number of executed*

[3]Note that the true bound must be a light tail, but due to the difficulties of selecting the right one reliably, we stick to exponential tails, which upperbound all light tails by construction [31]

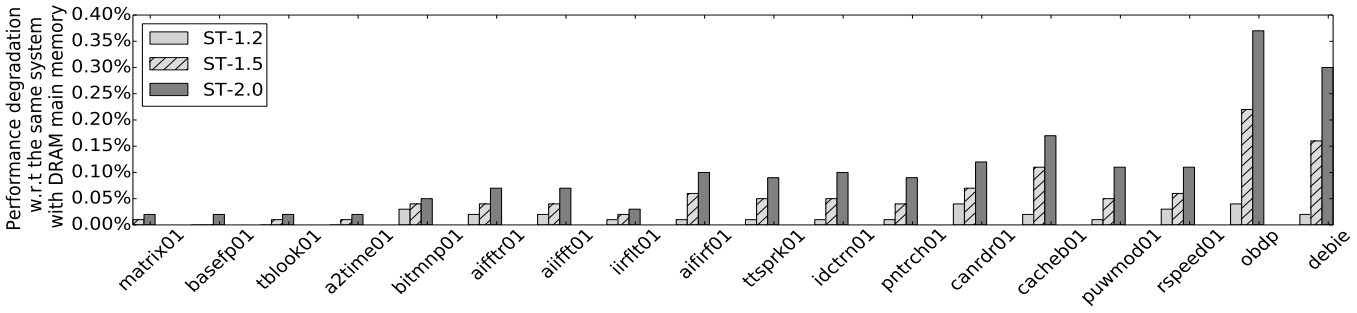

Figure 6: Average performance degradation w.r.t to DRAM. All the benchmarks under study show negligible performance slowdown for respective configurations of STT-MRAM.

instructions we calculate Cycles Per Instruction (CPI) values for each application with DRAM and each STT-MRAM configurations. We measure overall performance slowdown by the change of CPI values between systems with DRAM and STT-MRAM. Figure 6 shows overall system performance impact of different STT-MRAM configurations for the benchmarks under study. The bars represent the system performance degradation (from DRAM) for the corresponding benchmark listed at the X-axis. The different bars represent different STT-MRAM configurations. The results show STT-MRAM produces a negligible performance impact for all the benchmarks. For *ST-1.2* configuration, slowdown ranges from 0% (`matrix01`) to 0.04% (`obdp`). *ST-1.5* introduces slowdown ranging from 0% (`basefp01`) to 0.22% (`obdp`). *ST-2.0* shows a similar trend of low impact to the overall system performance. In the worst case, the system performance degradation is 0.37% (`obdp`).

To investigate more in this regard, we execute four applications with high memory utilization [43] from Mediabench benchmark suite. Figure 7 shows benchmarks with high memory utilization pay a higher performance penalty, although not very significant. For *ST-1.2* configuration, slowdown ranges from 0.44% (`mesa.texgen`) to 1.03% (`mesa.osdemo`). *ST-1.5* introduces slowdown ranging from 1.57% (`mesa.mipmap`) to 3.39% (`mesa.osdemo`). In the worst case, the performance degradation is 5.62% (`mesa.osdemo`) for *ST-2.0* configuration.

Within the scope of the performance analysis, we go a step further and analyze the performance degradation due to the modifications required to introduce *Randomization* to the target platform (See Section 3.4). The results show *Randomization* does not introduce any significant performance overhead to the system (0.061% deviation is the worst case — `mesa.osdemo` with *ST-2.0* configuration).

The results suggest, in the aspect of performance, STT-MRAM can be a good contender for aerospace and automotive applications.

5 EVALUATION: WCET

In this section we present and compare WCET estimates between the conventional DRAM and STT-MRAM memory systems. In particular, we use the measurement-based probabilistic timing analysis (MBPTA) described in Section 3.4 to compute probabilistic

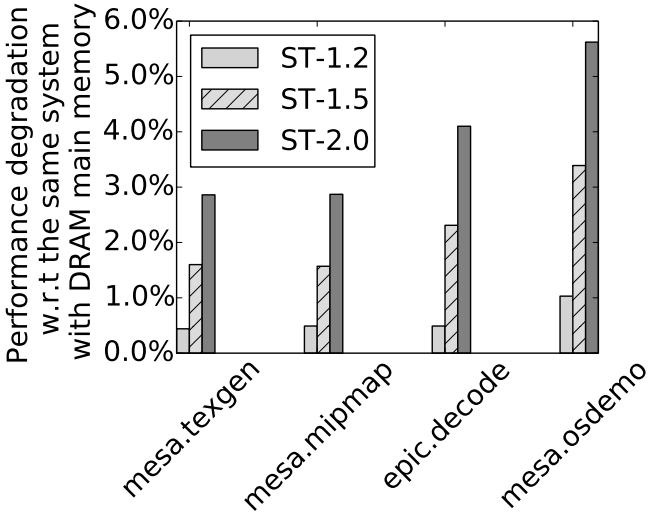

Figure 7: Applications with high memory utilization suffer from significant performance degradation with STT-MRAM

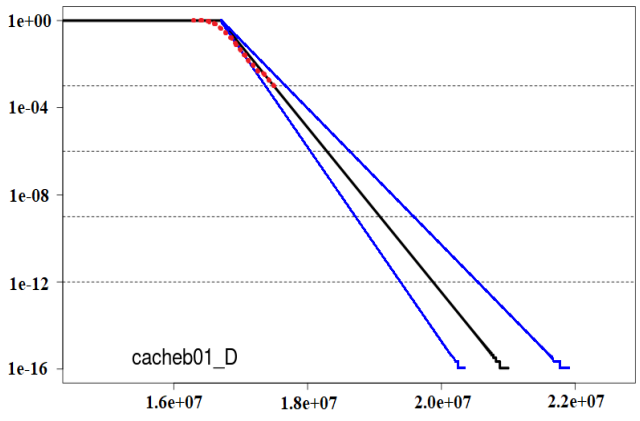

Figure 8: pWCET distribution for one of the scenarios under experiment: EEMBC benchmark `cacheb01` executed with DRAM main memory. Exceedance probability (Y-axis) of the corresponding execution time (X-axis), at any given run.

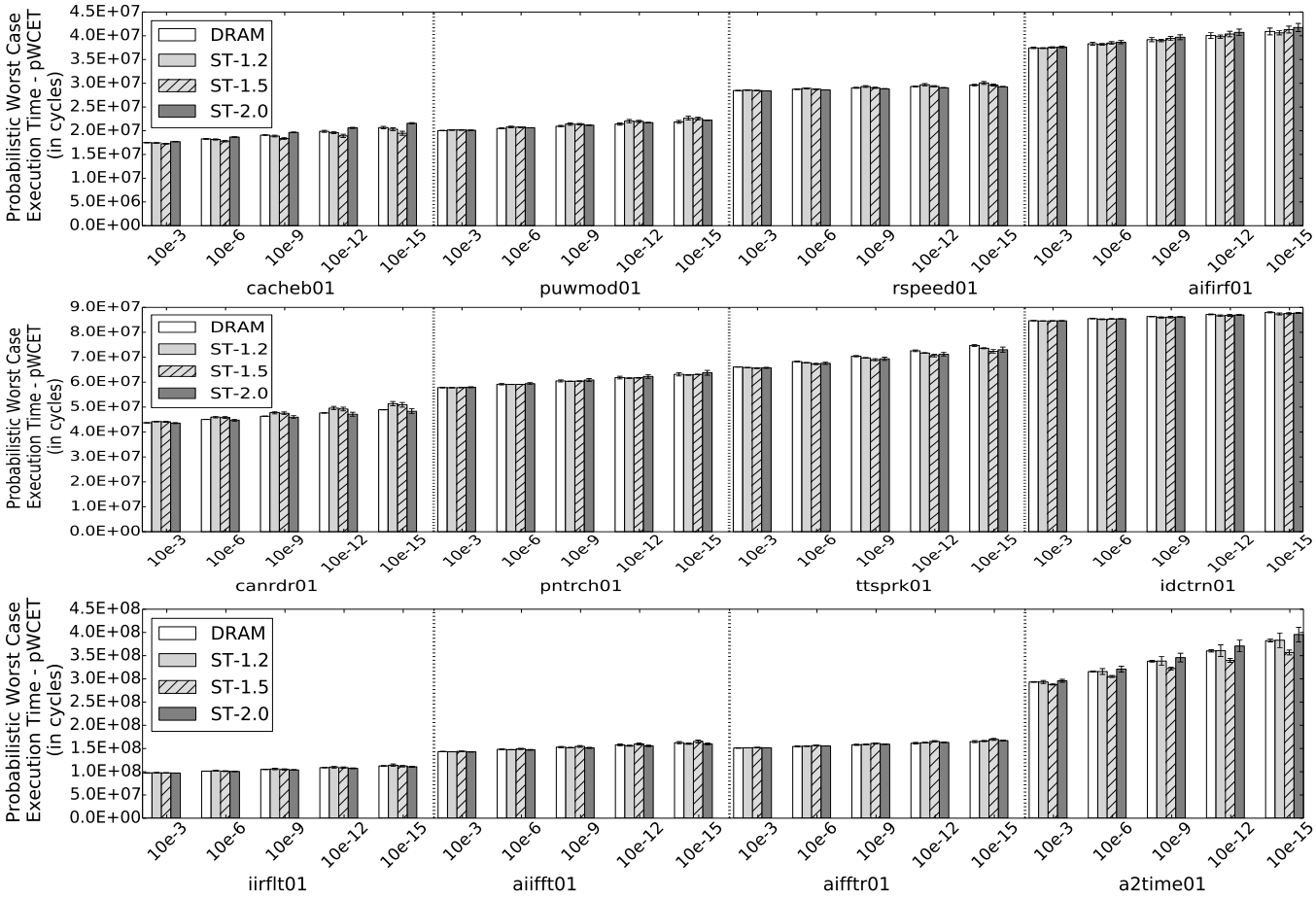

Figure 9: Probabilistic Worst Case Execution Time (pWCET) for benchmarks `cacheb01`, `puwmod01`, `rspeed01`, `aifirf01` (Top); `canrdr01`, `pntrch01`, `ttsprk01`, `idctrn01` (Middle); and `iirflt01`, `aiifft01`, `aifftr01`, `a2time01` (Bottom). X-axis lists benchmarks and exceedance probability.

WCET (pWCET) estimates for each benchmark and system configuration.

Figure 8 illustrates the pWCET distribution for the `cacheb01` benchmark executed in the system with the DRAM memory. The figure shows the probability (Y-axis) that the execution time in any `cacheb01` run exceeds the corresponding pWCET on the X-axis. The solid line corresponds to the pWCET point estimate, while the thin (blue) lines correspond to the 95% confidence interval. For example, the chart shows that, for the exceedance probability of 10^{-12}, the pWCET ranges between $19.3e^6$ and $20.5e^6$ cycles (95% confidence interval) while the point estimate is $19.8e^6$ cycles. The dotted (red) line represents actually measured execution times.

We estimate the pWCET distribution for 80 scenarios: two ESA, fourteen EEMBC [4], and four Mediabench benchmarks with four memory timing parameters: *DRAM*, *ST-1.2*, *ST-1.5*, and *ST-2.0*. Since, it would be a tedious task to plot and compare individual pWCET distribution charts for the 80 scenarios, we represent the results with a summarized appearance. In Figures 9 and 10 we plot the pWCET (Y-axis) for five different exceedance probabilities, 10^{-3}, 10^{-6}, 10^{-9}, 10^{-12}, 10^{-15} (Y-axis). The solid bars show the pWCET point estimate, while the error bars denote 95% confidence interval, as explained in Figure 8 . In order to increase the visibility of the results, the benchmarks are distributed among the figures based on their pWCET.

From the charts in Figures 9 and 10, we can see that for all benchmarks under study, pWCET slightly increases for decreasing exceedance probability, as expected. As shown in Figure 10, `mesa.texgen` has the widest confidence interval in relative terms across benchmarks. In this case, the confidence interval for DRAM at an exceedance probability of 10^{-15} is 98.6%-101.7%, normalized w.r.t. the point estimation for DRAM. Using the same reference, ST-2.0, the most pessimistic scenario has a confidence interval of 94.9%-101.1%, thus overlapping with the confidence interval for DRAM, which allows claiming that no configuration can be proven superior to the other. In general, we observe the very same trend

[4]We exclude `bitmnp01` and `matrix01` EEMBC benchmarks for WCET estimation because they did not fulfil the requirements of MBPTA statistical analysis (See Section 3.5).

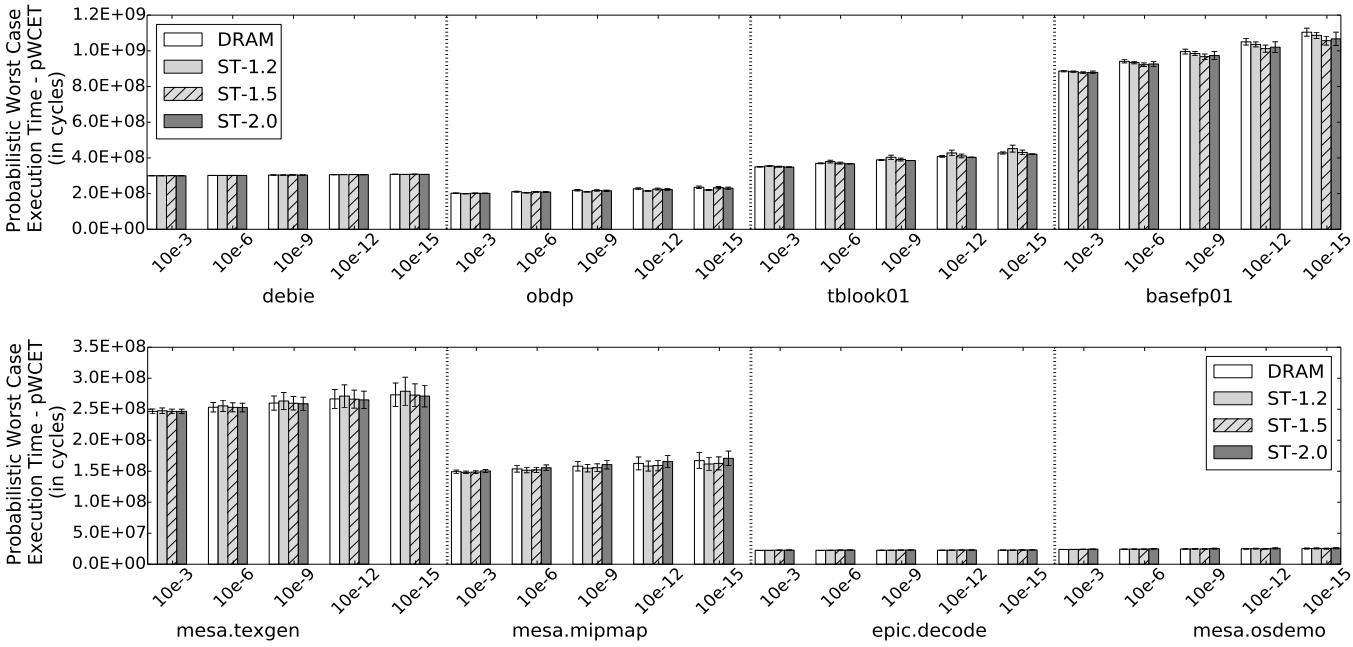

Figure 10: Probabilistic Worst Case Execution Time (pWCET) for benchmarks `debie`, `obdp`, `tblook01`, `basefp01`. (Top); and `mesa.texgen`, `mesa.mipmap`, `epic.decode`, `mesa.osdemo`. (Bottom). X-axis lists Benchmarks & Exceedance probability.

for all benchmarks, with overlapping confidence intervals between DRAM and all STT-MRAM configurations.

Most of the benchmarks show insignificant deviation in pWCET estimates for DRAM and STT-MRAM configurations. Some benchmarks (e.g. `cacheb01`, `canrdr01`, `a2time01`, `basefp01` etc.) show minor but visible fluctuations in pWCET estimations for different memory configurations. For example, in few cases STT-MRAM configurations offer better results than expected (i.e. faster than DRAM and/or faster STT-MRAM configurations). This relates to the intrinsic pessimism of EVT to fit pWCET curves. Eventually, high values in the random samples may fit in a slightly narrower value range due to pure random reasons, which allows EVT to find slightly tighter pWCET estimates. As we decrease the exceedance probability (e.g. down to 10^{-15}), discrepancies naturally amplify, but they are still within few percent points w.r.t. the reference DRAM setup.

There are two main observations from the WCET estimation results. First, our results confirm that the measurement-based probabilistic timing analysis (MBPTA) described in Section 3.4 can be applied for the system comprising STT-MRAM main memory. The effort for the pWCET analysis, including the benchmark runs and the statistical analysis, does not change from the DRAM to the STT-MRAM main memory. Each benchmark converged to produce a valid WCET estimate approximately with a thousand runs. This is fundamental to reduce STT-MRAM adoption costs, without requiring new tools that must undergo a costly qualification process [7]. Second, the results show that the pWCET estimates have very narrow confidence interval, and that there is negligible difference between WCET estimations with DRAM and STT-MRAM systems.

6 RELATED WORK

Most of the STT-MRAM system-level research so far, focused on the suitability of this technology for on-chip cache memories. In addition to this, few studies analyze the possibility to replace DRAM main memory with STT-MRAM modules. No study, to our knowledge, has yet evaluated STT-MRAM particularly for real-time embedded systems with performance and WCET analysis.

Meza *et al.* [44] 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.* [45] evaluate STT-MRAM as a main memory for SPEC CPU2006 workloads and show that, without any optimizations, early-design STT-MRAM [46] 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.* [47] analyze the 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.* [48] evaluate STT-MRAM main memory for high-performance computing and analyze the performance impact when DRAM is simply replaced with STT-MRAM. The presented results suggest 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 their effect on the overall system performance [24][49].

Asifuzzaman *et al.* [20] also present a detailed analysis of STT-MRAM main memory timing and propose an approach to perform a reliable system level simulation of the memory technology. These parameters are accepted by the community and included into the main DRAMSim2 distribution.

Jiang *et al.* [50] 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.* [51] using CACTI [52] cache simulator and NVSim [53].

Simone *et al.* [54] advocate to exploit unlimited endurance of STT-MRAM as small-capacity rad-hard memories due to their inherent resistance to radiation.

7 CONCLUSIONS

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. Intensified efforts in STT-MRAM research by the memory manufacturers may indicate a revolution with STT-MRAM memory technology is imminent, and therefore, it is now the time to explore computing domains that can benefit from this technology. Special STT-MRAM features such as intrinsic radiation hardness, non-volatility, zero stand-by power and capability to function in extreme temperatures offer a great opportunity to explore its usability in real-time embedded systems, particularly in space, avionics and automotive domains.

In this study, we investigate the feasibility of using STT-MRAM in these domains by analyzing system performance impact and worst case execution time (WCET) implications. In our opinion, this is of vital importance to perform a head-to-head comparison of a new technology to the conventional one on the same platform without proposing ambitious optimizations. Because, such proposals may actually obscure the critical information where the technology stands as-is, or how far it is from being used as a standard replacement of the conventional one.

The results suggest, in the aspect of performance, STT-MRAM can be a good contender for aerospace and automotive applications. For WCET analysis, the results confirm MBPTA can be applied for the system comprising STT-MRAM main memory. The effort for the WCET analysis, including the benchmark runs and the statistical analysis, does not change from the DRAM to the STT-MRAM main memory. This is fundamental to reduce STT-MRAM adoption costs, without requiring new tools that must undergo a costly qualification process [7]. The results also show that the WCET estimates have very narrow confidence interval, and that there is negligible difference between WCET estimates with DRAM and STT-MRAM systems.

Overall, this study presents the first comprehensive exploration of possibilities to use STT-MRAM in the real-time embedded domain and reveals that STT-MRAM would provide performance and WCET estimates comparable to DRAM while opening up several key advantages that the domain could benefit from.

8 ACKNOWLEDGMENT

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). Jaume Abella was partially supported by the Ministry of Economy and Competitiveness under Ramon y Cajal postdoctoral fellowship RYC-2013-14717.

REFERENCES

[1] Y. Kim, R. Daly, J. Kim, C. Fallin, J. H. Lee, D. Lee, C. Wilkerson, K. Lai, and O. Mutlu. Flipping bits in memory without accessing them: An experimental study of dram disturbance errors. In *41st ACM/IEEE International Symposium on Computer Architecture (ISCA)*, 2014.

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

[3] Everspin Technologies, Inc. STT-MRAM Products. https://www.everspin.com/stt-mram-products, 2018.

[4] European Space Agency. *GR740: The ESA Next Generation Microprocessor (NGMP).* http://microelectronics.esa.int/gr740/index.html.

[5] J. A. Poovey, T. M. Conte, M. Levy, and S. Gal-On. A Benchmark Characterization of the EEMBC Benchmark Suite. *IEEE Micro*, 2009.

[6] Chunho Lee, Miodrag Potkonjak, and William H. Mangione-Smith. Mediabench: A tool for evaluating and synthesizing multimedia and communicatons systems. In *Proceedings of the 30th Annual ACM/IEEE International Symposium on Microarchitecture*, MICRO 30, 1997.

[7] International Organization for Standardization. *ISO/DIS 26262. Road Vehicles – Functional Safety*, 2009.

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

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

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

[11] 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. 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.

[12] 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.

[13] H. Kim, S.K. Kang, D.H. SOHN, D.M. Kim, and K.C. Lee. Magneto-resistive memory device including source line voltage generator, 2013.

[14] H.R. Oh. Resistive Memory Device, System Including the Same and Method of Reading Data in the Same, 2014.

[15] C. Kim, D. Kang, H. Kim, C.W. Park, D.H. SOHN, Y.S. Lee, S. Kang, H.R. Oh, and S. Cha. Magnetic Random Access Memory, 2013.

[16] D. Chabi, W. Zhao, J. O. Klein, and C. Chappert. Design and analysis of radiation hardened sensing circuits for spin transfer torque magnetic memory and logic. *IEEE Transactions on Nuclear Science*, 2014.

[17] M. Riera, R. Canal, J. Abella, and A. Gonzalez. A detailed methodology to compute soft error rates in advanced technologies. In *2016 Design, Automation Test in Europe Conference Exhibition (DATE)*, pages 217–222, March 2016.

[18] Everspin Technologies, Inc. Case Study: SpriteSat (Rising) Satellite. https://www.everspin.com/aerospace, 2018.

[19] Everspin Technologies, Inc. Automotive.

[20] 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 '17, pages 283–292, 2017.

[21] SoCLib. -, 2003-2012. http://www.soclib.fr/trac/dev.

[22] L. Fossati M. Zulianello F. J. Cazorla J. Jalle, J. Abella. Validating a timing simulator for the ngmp multicore processor. In *2016 DASIA*.

[23] Cobham Gaisler. *GR-CPCI-LEON4-N2X Quad-Core LEON4 Next Generation Microprocessor Evaluation Board*. http://www.gaisler.com/index.php/products/boards/gr-cpci-leon4-n2x.

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

[25] Micron Technology, Inc. *Automotive DDR2 SDRAM*, 2011.

[26] J. Abella, C. Hernandez, E. QuiÃśones, F. J. Cazorla, P. R. Conmy, M. Azkarate-askasua, J. Perez, E. Mezzetti, and T. Vardanega. Wcet analysis methods: Pitfalls and challenges on their trustworthiness. In *10th IEEE International Symposium on Industrial Embedded Systems (SIES)*, pages 1–10, June 2015.

[27] Reinhard Wilhelm, Jakob Engblom, Andreas Ermedahl, Niklas Holsti, Stephan Thesing, David Whalley, Guillem Bernat, Christian Ferdinand, Reinhold Heckmann, Tulika Mitra, Frank Mueller, Isabelle Puaut, Peter Puschner, Jan Staschulat, and Per Stenström. The worst-case execution-time problem—overview of methods and survey of tools. *ACM Trans. Embed. Comput. Syst.*

[28] I. Wenzel, R. Kirner, B. Rieder, and P. Puschner. Measurement-based timing analysis. In *ISOLA*, 2008.

[29] I. Wenzel, R. Kirner, B. Rieder, and P. Puschner. Measurement-based worst-case execution time analysis. In *SEUS Workshop*, 2005.

[30] S. Law and I. Bate. Achieving appropriate test coverage for reliable measurement-based timing analysis. In *ECRTS*, 2016.

[31] Francisco J. Cazorla, Leonidas Kosmidis, Enrico Mezzetti, Carles Hernandez, Jaume Abella, and Tullio Vardanega. Probabilistic worst-case timing analysis: Taxonomy and comprehensive survey. *ACM Comput. Surv.*, 2019.

[32] F. Wartel et al. Timing analysis of an avionics case study on complex hardware/software platforms. In *DATE*, 2015.

[33] M. Fernandez et al. Probabilistic timing analysis on time-randomized platforms for the space domain. In *DATE*, 2017.

[34] Z. Stephenson et al. Supporting industrial use of probabilistic timing analysis with explicit argumentation. In *INDIN*, 2013.

[35] Cobham Gaisler. LEON3 Processor (Probabilistic platform). http://www.gaisler.com/index.php/products/processors/leon3.

[36] L. Kosmidis et al. Probabilistic timing analysis on conventional cache designs. In *DATE*, 2013.

[37] L. Kosmidis et al. TASA: Toolchain-agnostic Static Software Randomisation for Critical Real-time Systems. In *ICCAD*, 2016.

[38] Jaume Abella, Maria Padilla, Joan Del Castillo, and Francisco J. Cazorla. Measurement-based worst-case execution time estimation using the coefficient of variation. *ACM Trans. Des. Autom. Electron. Syst.*, 2017.

[39] Leonidas Kosmidis, Eduardo QuiÃśones, Jaume Abella, Tullio Vardanega, Carles Hernandez, Andrea Gianarro, Ian Broster, and Francisco J. Cazorla. Fitting processor architectures for measurement-based probabilistic timing analysis. *Microprocessors and Microsystems*, 47:287 – 302, 2016.

[40] Francisco J. Cazorla, Tullio Vardanega, Eduardo Quiñones, and Jaume Abella. Upper-bounding Program Execution Time with Extreme Value Theory. In Claire Maiza, editor, *13th International Workshop on Worst-Case Execution Time Analysis*, volume 30 of *OpenAccess Series in Informatics (OASIcs)*, pages 64–76, Dagstuhl, Germany, 2013. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[41] S. Kotz et al. *Extreme value distributions: theory and applications*. World Scientific, 2000.

[42] RTCA and EUROCAE. *DO-178B / ED-12B, Software Considerations in Airborne Systems and Equipment Certification*, 1992.

[43] B. Bishop, T. P. Kelliher, and M. J. Irwin. A detailed analysis of mediabench. In *1999 IEEE Workshop on Signal Processing Systems. SiPS 99. Design and Implementation (Cat. No.99TH8461)*, 1999.

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

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

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

[47] 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.

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

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

[50] Lei Jiang, Wujie Wen, D. Wang, and L. Duan. 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.

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

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

[53] X. Dong, C. Xu, Y. Xie, and N. P. Jouppi. NVSim: A Circuit-Level Performance, Energy, and Area Model for Emerging Nonvolatile Memory. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 31(7):994–1007, 2012.

[54] S. Gerardin and A. Paccagnella. Present and future non-volatile memories for space. *IEEE Transactions on Nuclear Science*, 2010.