RC25679 (AUS1312-003) September 11, 2015
Computer Science

IBM Research Report

Design of High-performance, Resilient, STT-MRAM-based Main Memory

Janani Mukundan, Karthick Rajamani, Alexandre Ferreira, Jente Kuang

IBM Research Division
Austin Research Laboratory
11501 Burnet Road
Austin, TX 78758
USA

Kyu-Hyoun Kim, Hillery Hunter, Luis Lastras

IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598
USA

Xiaochen Guo

University of Rochester
500 Computer Studies Building
P.O. Box 270231
Rochester, NY 14627
USA

Research Division
Almaden – Austin – Beijing – Cambridge – Dublin - Haifa – India – Melbourne - T.J. Watson – Tokyo - Zurich

Design of High-performance, Resilient, STT-MRAM-based Main Memory

Janani Mukundan Karthick Rajamani Alexandre Ferreira Jente Kuang
Kyu-Hyoun Kim Hillery Hunter Luis Lastras Xiaochen Guo

ABSTRACT

STT-MRAM is an exciting new memory technology that is being considered as a replacement candidate for SRAM in caches and DRAM in main memory. While the technology has adequate endurance and read/write latencies, challenges to its adoption include relatively high write and read currents per cell and high error rates. We evaluate the feasibility of STT-MRAM as a main memory replacement technology for DRAM, analyze the challenges to its adoption and provide solutions.

We reduce power at a device-chip level by restricting the number of cells concurrently accessed within the chip for each request. We present a device architecture that enables this reduction without affecting its compatibility with existing DRAM DIMM designs. For managing error rates we look in detail at the different error phenomena for STT - MRAM and examine the effectiveness of ECC technology and scrubbing to mitigate them. Using both analytical methods (for error modeling) and simulation methods (for power and performance quantification) we identify the STT-MRAM technology (characteristics) for which feasible ECC and scrubbing solutions can provide adequate error correction. Limiting the number of cells activated in a device for each access (for power feasibility) and adoption of error correction solutions have potential to reduce performance. We propose two optimizations to improve performance – *command compaction* and *banklet architecture* – and quantify their benefits with a suite of memory-intensive workloads.

1. INTRODUCTION

DRAM has been the primary main memory technology since its invention by Dennard [8], and subsequent wide adoption in the 1970s. JEDEC standards for DRAM have provided well-established interfaces across different computing domains allowing for its widespread adoption. However, recent trends have called to question the continued viability of DRAM as the main memory technology. Scaling the 1T-1C cell for DRAM requires increasing the aspect ratio for the storage capacitor so as to store the required amount of charge – this has significant manufacturing challenges [13]. A potential drop in cell capacitance due to manufacturing challenges (as well as increase in chip capacity for meeting growing demands) can worsen refresh overheads [19]. Additionally, increases in cell resistance, increasing leakage in access transistor and lithographic challenges [33] have heightened the concerns with DRAM scaling. Further, multi-core architectures and data-driven applications have driven up main memory capacity and bandwidth needs, driving up power-performance concerns [28].

STT-MRAM is a resistive memory technology in which the change in magnetic spin of electrons in the material produces a measurable change in resistivity. The spin-controlled resistive nature of STT-MRAM avoids capacitance-related scaling challenges and its purported non-volatile nature avoids leakage/refresh power concerns associated with scaling of DRAM. Besides relative immaturity of the technology, STT-MRAM is faced with many challenges not seen with DRAM – (a) high error rates, (b) high write currents, (c) high read currents and (d) complex sensing requirements. Models exist to characterize the error behavior for the technology but we still need to identify solutions to bring down the system error rates to levels that can allow STT-MRAM to be used in place of DRAM. High write currents and sensing requirements have required current test-chip demonstrations of STT-MRAM to opt for significantly smaller row width devices (than DRAM) potentially impacting its density and throughput. For STT-MRAM to be a replacement for DRAM when the latter's scaling proves uneconomical, it needs to have at least competitive power and reliability characteristics. In this paper, we investigate the challenges to STT-MRAM on both these fronts and identify possible solutions to make main memory based on STT-MRAM power- and resilience-competitive with respect to DRAM.

Drawing from various STT-MRAM device demonstrations, we identify an architecture that adopts a mux/demux topology for selecting only a subset of bitlines to be sensed for an activated word-line. This reduces the device-level read/write current loads, providing a power-feasible STT-MRAM main memory device. Among the material properties of STT-MRAM, we found that the *thermal stability factor*, variation in critical current of cells, and ratio of resistivity between the different spin-polarized states (Tunneling Magneto Resistance or TMR) have significant effect on the resilience of the technology to errors. The dominant errors – read sense, write and retention – are shown to vary with the material parameters. At the system level, we leverage suitable mitigation approaches for these errors, namely multi-bit error correction codes (ECC) and scrubbing. We identify the requirements on the material properties for STT-MRAM based

on the cost of the mitigation solutions. We then propose two optimizations to STT-MRAM device access protocol and bank architecture – *command compaction* and *banklet architecture* – which can improve the performance of a feasible STT-MRAM memory design. Our proposals are similar in spirit to others' that have suggested sub-banking for DRAM [26, 12] and using STT-MRAM's unique characteristics for speeding up command sequences [27]. However, we leverage the device architecture we identified for power feasibility of STT-MRAM and do not affect the compatibility of the device interface with today's DRAM (unlike previous proposals) eliminating potential hurdles to the proposals' adoption.

From our error analysis, we conclude that at least 3-bit error correction codes (ECC) will be required for mitigating the dominant errors in STT-MRAM. If the thermal stability factor were to get lower to 50, then 5-bit ECC along with scrubbing at 1 hour intervals is required. For main memory where multiple devices service a request, ECC inside the device is less efficient than the off-chip ECC solution we employ/advocate. With our optimizations, *command compaction* and *banklet architecture*, STT-MRAM obtains 12.5% (and 9% for material that needs more frequent scrubbing) better performance than DRAM on average and 21% (16% with more frequent scrubbing) savings in energy.

Another important takeaway from our error analysis work is that STT-MRAM is not viable as a persistent memory technology at low thermal stability factors - scrubbing would be needed to maintain data retention and avoid multi-bit errors. Therefore, while the technology may be scalable from a manufacturing perspective, its touted non-volatility may not hold without careful attention to materials to bring up the thermal stability factor (which goes down as cell size is reduced).

2. BACKGROUND

2.1 DRAM Organization and Access Methods

A typical DRAM main memory system has one or more memory controllers managing the memory channels connected to DRAM memory. Buffer chips are employed in high performance server memory systems where they facilitate signal integrity for high capacities/bandwidth needs. In such systems, the buffers or the memory controller (or both) also support error correction capabilities. A set of DRAM devices that can operate in lockstep to service a memory request is called a DRAM *rank*. Multiple such ranks can be attached to a DRAM channel. Each DRAM device in a rank is made up of a set of memory arrays (1T-1C storage cells) called a *bank*. Banks operate independently and share only the I/O peripheral circuits. Each bank has a row-buffer which holds the contents of the row that is currently being acted on.

A bank has to be *activated* before a *read* or *write* command can be issued. This involves sensing the contents of a row by charge-transfer process and storing it in the row buffer (and subsequent restoration of charge to the cells). In order to access another row, the current row must be closed or *precharged* which includes overheads for stabilizing the word lines. Due to leakage of the charge in the capacitors, DRAM requires periodic refresh operations to guarantee data retention. DRAM refresh is realized by activating and precharging rows, during which the bank cannot service other access requests. Predictably, as DRAM density increases, refresh has a sizable impact both in performance and energy efficiency.

2.2 STT-MRAM

A STT-MRAM cell consists of a magnetic tunnel junction (MTJ), and an access transistor in series.The MTJ has a pinned magnetic reference layer and a free magnetic layer separated by a dielectric layer. The pinned layer has a fixed magnetic orientation while the free layer can be programmed by changing its magnetic orientation. The MTJ resistance is determined by the relative magnetization between the pinned and free layer. Parallel and anti-parallel magnetization render lower and higher resistivity, respectively, which, in turn, defines the binary states of the memory cell. The write operation is done by switching the free layer magnetization using a current induced spin transfer torque. Similar to DRAM, a pair of bit lines are associated with each column of cells. For a write, a cell's access transistors are connected to the bit lines and the direction of current between the pair of bit lines decides the relative magnetization of the free layer with respect to the reference layer. The read operation is done by sensing the resistance by driving a specific read current driven through the cell. With no charge sharing involved in the sensing process, the strict timing and restoration of cell values after sensing (as is for DRAM) is not required for STT-MRAM.

3. POWER CONSTRAINTS AND SOLUTIONS

Efforts to design solutions for replacing the 1T-1C DRAM cell with a 1T-1R STT-MRAM cell in a memory array are already underway [18, 33, 31] and a number of small density (relative to DRAM) test chip demonstrations have been seen [6, 15, 25, 20]. However, to replace DRAM devices in the commodity market, large volume production of high density chips is necessary.

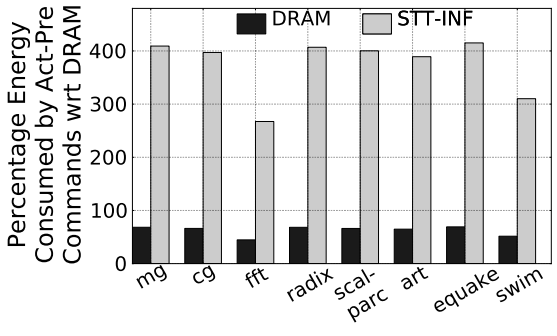

Figure 1: Percentage of energy consumed by the *activate* and *precharge* commands (lower is better), with DRAM and STT-Inf configuration with 1T-1R STT-MRAM cells, relative to total DRAM energy. Both sense a 1KB row during activation. Cell write and sensing currents (section 5.3 used are in table 6).

Naively replacing DRAM cells with STT-MRAM cells in current designs, while keeping the same external periphery and interface circuits would be infeasible due to the high cell

write and sensing currents [27] (see section 5 for values). Assuming a row buffer size of 1KB – a design point that is typical in current DRAM chips – every *activate* command would require sensing 8192 cells. Figure 1 shows the energy consumed in activating a 1 KB row, when running a set of parallel applications on a DRAM configuration and a naive STT-MRAM configuration – STT-Inf, that replaces DRAM cells with STT-MRAM cells. The figure illustrates that writing or sensing from an entire row of cells, like in the case for DRAM, would be impractical from a peripheral circuit and chip power perspective.

Therefore, for a given STT-MRAM chip capacity, this leaves us with two options:

- Increasing the number of banks within a device, each with smaller number of columns, compared to an equivalent DRAM design point.
- Reducing the number of columns, and increasing number of rows within each bank of the device, as suggested in [27]

In the first case, this would increase the number of row decoders (one per bank) and the second would increase the size of the row decoder – both of which would lower the overall array-to-periphery efficiency.

Current (experimental) STT-MRAM designs [17, 25] use a sensing architecture similar to SRAM designs, which has fewer sense-amps per row - just the number needed for their I/O design point rather then one per bitline (pair) in the STT-MRAM array. We recognize that this reduces total sensing current as well as the periphery overhead for a given array size. The incorporation of both the external DRAM interface and SRAM style sense-amp/latch circuitry results in the combined benefit of DRAM-like long rows with low decoder overhead and SRAM-like compact columns with multiplexers and shared sensing circuits for lower array access power.

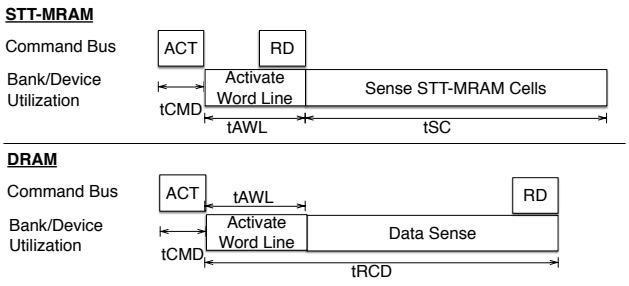

Figure 2: Timing diagrams for the *activate* and *read* commands using our proposed STT-MRAM (top) and current DRAM (bottom) memory access protocols. tAWL – time to raise the word line, tSC – time sensing STT-MRAM cells, tRCD – time to raise the word line and sense the DRAM cells after activate cmd.

Furthermore, for a feasible DRAM replacement, STT - MRAM system designs need to sense only the minimum amount required for a single data burst (64 bytes) reducing the number of sense amplifiers required. To achieve a reasonable row size, we multiplex the sense amplifiers to multiple columns by placing a column multiplexer between the bit lines and sense amplifiers. Operationally, this implies that the sensing can only start after the appropriate column bits have been transmitted by the memory controller (through the *read* command). Therefore, we propose to move the sensing operation from the activate command to the read command. Figure 2 shows the timing diagram for the *activate* and *read* command when using the DRAM access protocol (bottom), and when applying our proposed change to the access protocol – moving the sensing from the *activate* to the *read* command (top).

Table 1: STT-MRAM device parameters for three different cell configurations [30, 7]

Parameter	Description	Value					
Δ	Thermal Stability Factor	100 - 30					
τ_0	Intrinsic Attempt Time	0.024 ns					
τ_D	Cell Character. Time	0.26 ns					
I_r / I_{c0}	Ratio of read current to critical current	$\mu = 0.34$ ($\sigma = 11\%$)					
Parameter	**Description**	**Config-A**		**Config-B**		**Config-C**	
		μ	σ (%)	μ	σ (%)	μ	σ (%)
I_w / I_{c0}	Ratio of write current to critical current	1.75	11	1.75	**7**	1.75	**7**
R_{low}	Low Resistance (parallel state)	1.63	5	1.63	5	**1.33**	5
R_{high}	High Resistance (anti-parallel)	3.233	5	3.233	5	**3.53**	5
TMR (derived)	Tunneling Magneto Resistance	0.97		0.97		**1.65**	

4. RELIABILITY OF STT-MRAM

STT-MRAM suffers from important reliability issues distinct from DRAM and other non-volatile storage technologies [9]. STT-MRAM reads and writes are stochastic in nature. Write errors increase when the write pulse width or the write current is small, leading to difficulty in flipping the magnetic moments of the cell. Read errors occur when the data in the cell is incorrectly sensed, and increase when the read current is high causing change in cell state when reading. STT-MRAM also suffers from retention errors which are caused because of state changes (1 –> 0 or 0 –> 1) due to thermal fluctuations. In the following subsections, we present device and system level error models for the error characteristics of STT-MRAM and analysis of different solutions.

4.1 Device Level Analytical Modeling

The two states of an STT-MRAM cell (parallel and anti-parallel) are separated by an energy barrier E. When no spin current is passed through the cell, the moments of the free layer may get slightly perturbed due to random thermal fluctuations of magnitude k_BT (k_B is the Boltzmann constant, and T is the temperature in kelvin) [5]. If the energy from these random fluctuations exceeds E, it can overcome the energy barrier and change the state of the cell. The height of the energy barrier relative to k_BT is called the thermal stability factor–Δ, i.e., $\Delta = E/k_BT$. Technology scaling is expected to reduce Δ as E is proportional to the volume of

the magnetic tunnel junction. In general, higher Δ is preferred for retaining written values but lower Δ requires less write energy. The different error rates for STT-MRAM, can be derived from the switching probability as given for the two key switching regimes for STT-MRAM.

- Precessional: When more than the critical switching current flows through the STT-MRAM magnetic tunnel junction (MTJ), $I > I_{c0}$. Switching Probability:

$$P_{sw} = \exp\{-4\Delta[-\frac{2\tau(I/I_{c0}-1)}{\tau_D}]\}$$

- Thermal Activation: When the switching current is low ($I < I_{c0}$) and switching of the free layer state occurs because of thermal noise. Switching Probability:

$$P_{sw} = 1 - \exp(\frac{-\tau}{\tau_0}\exp[-\Delta(1-\frac{I}{I_{c0}})])$$

Using the switching probability equations we can calculate the following bit error rates (BER) as a function of the thermal stability factor (Δ) : (a) Write BER, (b) Read Disturb BER, (c) Read Sense BER, and (d) Retention BER. Write BER is affected by the ratio of write current to the critical current of the cell, read sense BER is affected by the TMR of the device and the effectiveness of the sensing circuitry, and retention BER is affected by thermal stability and the desired retention period (which is how long the stored value has to be retained in the cell). We examine how these error rates change with improvements to key physical characteristics for STT-MRAM. Table 1 provides details on the key device and circuit level parameters for modeling reliability in STT-MRAM cells. In particular, we study three configurations: (a) Config–A is based on data for a perpendicular spin torque junction STT-MRAM from Worledge et al. [30], and Bedau et al. [7], (b) Config–B has better write current stability (σ of 7% for I/I_{c0}) and so lower write BER, (c) Config–C has same write current stability as B (better than A) but better TMR (1.65), and so has lower read sense BER. In Table 1, the parameter values for Config–B and Config–C that differ from those for Config–A are in **bold**.

Figure 3 illustrates the effect on the various error rates as Δ reduces (due to technology scaling) for the three STT-MRAM cell configurations:
(a) For all three configurations, write BER and read sense BER tend to dominate the total BER at higher values of Δ. But as Δ reduces, retention errors and read disturb errors start dominating the total BER, with retention errors being much more dominant amongst the two.
(b) Reducing the variation (σ) on the write current to critical current ratio from 11% (Config–A) to 7% (Config–B, Config–C) improves the write error rate by four orders of magnitude (10e-5 to 10e-9). Similarly, improving the TMR of the cell from 0.97 (Config–A, Config–B) to 1.65 (Config–C) improves the read sensing error by three orders of magnitude (10e-5 to 10e-8).
(c) Retention errors become dominant at lower values of Δ. This suggests that as STT-MRAM scales, it might require periodic *scrubbing* operations (background read operations with error detection and correction to avoid uncorrectable multi-bit errors).

4.2 System Level Error Correction Modeling

Different error correction schemes have been proposed for and are in use in memory systems today. Simple, lower complexity schemes like SECDED are employed for single-burst data transfer protection on memory buses; efficient, multi-bit protection codes like Bose-Chaudhuri-Hocquenghem (BCH) have been proposed for use in soft-error prone environments [2] or for memories with high error rates like STT-MRAM [14, 5], and symbol-based protection schemes like Reed-Solomon (RS) have been in use on high-performance memory system to support functions like chip-kill in combination with ECC for transfers on the bus. We use a system level reliability model for STT-MRAM-based main memory, which allows us to determine the total uncorrected error rate for multi-tiered error detection and correction schemes employed at different levels of the memory hierarchy. The analytical model takes as input the following configuration parameters:

- Memory system parameters like average read and write latencies, bus and pipeline flush latencies, and the granularity of ECC (bit/symbol level).
- ECC parameters detailing the different error protection schemes that need to be evaluated at different levels of the memory hierarchy.

For our analysis, we vary the capability of our ECC schemes from one bit correction (ECC1) to eight bit correction (ECC8) for each cache line of size 64B. In addition we examine *scrubbing* to mitigate retention errors and reduce multi-bit error occurrences. In scrubbing, the system firmware accesses each memory location, reads the value, performance the ecc check, and writes back the correction, if needed. The scrubbing operations are scheduled such that within a specified interval (= desired retention period) the entire memory is scrubbed once. Higher the frequency of scrubbing, lower the errors for a demand access but higher the performance impact (from interference for normal accesses) and energy cost associated with scrubbing.

4.3 Analysis of Error Correction Options

We use cache-line-level (64B) uncorrected error rate as the main measure of acceptability for error correction solutions. We fix 1e-17 as the threshold for acceptable error rates for this analysis (this corresponds to an average of 1 uncorrected error in 29.5 years for a steady access rate of 12.8GB/s). We examine a range of thermal stability factor (Δ) from 70, representative of many current demonstrations, to smaller values down to 40, potentially realizable with technology scaling. We examine ECC1 through ECC8 (1-bit to 8-bit correction) and scrubbing intervals starting at 24 hours down to 1 second – during which every (cache-line address) location in memory is scrubbed.

The results are captured in Figures 4- 7, each figure corresponding to one particular scrub interval across all the ECC options (x-axis) and every Δ option for STT-MRAM models Config–A, Config–B and Config–C (separate lines on the plot). On the y-axis are the uncorrected error rates on a log-scale. The horizontal line (at 1e-17) marks the threshold we set for acceptable error rate, all points below the line are material-ECC options that meet our error-threshold criteria. The vertical line between ECC5 and ECC6 marks the

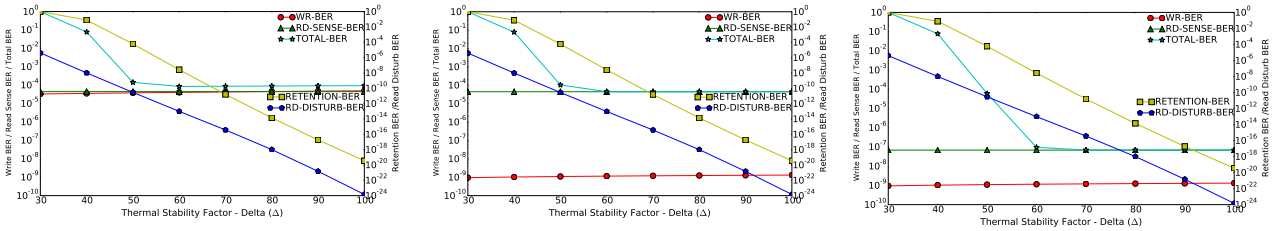

Figure 3: Thermal Stability Factor vs Bit Error Rates for STT-MRAM devices using Config–A, Config–B and Config–C [1]

boundary for 12.5% storage overhead similar to what is allowed in existing DRAM-based main memory frame-works employing ECC (1 additional chip for 8 chips, 2 for 16 etc.). With storage overhead as 11 ECC bits per bit to be corrected (BCH), a payload of 64 bytes with ECC6 has the total number of bits ($ECC + data = 11 * 6 + 512$) as 578, just a shade over $64 * 8 * 12.5\%8 = 576$ bits. In the figures, the points below the horizontal line meet the error threshold (1e-17) and of those the ones to the left of the vertical line can be accommodated with storage overheads equivalent to that allowed in current DRAM-DDRx ECC frameworks. ECC6, ECC7 and ECC8 may become feasible with bigger payload assumptions (say cacheline of 128 bytes used in some high-performance server systems), coding schemes that require less redundant bits per bit to be corrected, or a higher tolerance to storage overheads associated with error correction.

Qualifying material and error mitigation options: It is apparent that stronger the error correction solution greater the material choices that qualify. For Config–A, write and read-sense errors dominate at $\Delta = 60, 70$ – ECC8 is required to meet our error threshold. Reducing the Scrub interval to 1 hour or smaller allows lower stability factors (50) to become acceptable when using ECC8. For Config–B, read-sense is dominant at $\Delta = 60, 70$ and meets our error threshold only with the highest correction option, ECC8. With lower Δ, retention errors also add up – a scrub interval of 1 hour for $\Delta = 50$, and 1 second for $\Delta = 40$ along with ECC7 and ECC8 are required to qualify Config–B[1] [2].

For Config-C, ECC3 is adequate to qualify for $\Delta = 60, 70$ where the write and read-sense errors dominate. $\Delta = 50$ requires a scrub interval of 1 hour before it allows Config–C to qualify with ECC5. $\Delta = 40$ requires a 1 second scrub interval and ECC5 to qualify. Config–C alone qualifies at ECC5 or lower, i.e., within the storage allowed for ECC in DDRx-DRAM ECC frame-works. As Δ is lowered STT-MRAM can no longer claim to be a persistent memory technology as it requires fairly frequent scrubbing to combat worsening retention errors.

To summarize, improvements in manufacturing and technology (Config–C) as well as advanced system-level error correction solutions are needed to qualify STT-MRAM from an acceptable error rate consideration (for 12.5% ECC storage overhead as in server memory systems today).

[1] there is a small drop in read-sense BER with decreasing delta that enables ECC7 with smaller scrub intervals to enable qualification at lower Δ

[2] Config–A, B and C with $\Delta = 40$ for 24 hours and STT-MRAM A and B with $\Delta = 40$ for 60 minutes have error rates higher than the plotted range and so are not shown.

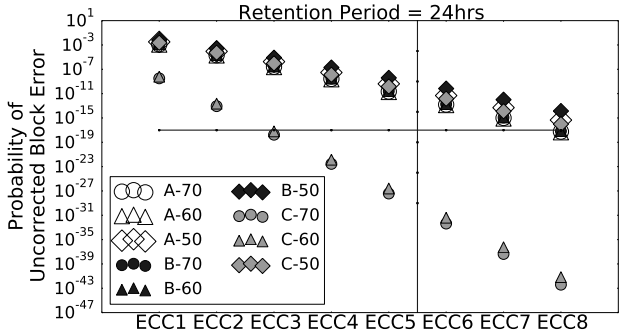

Figure 4: System level error modeling for Configs A, B and C at a retention period of 24 hours.

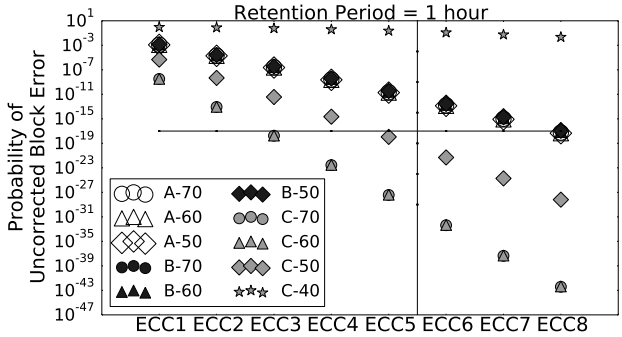

Figure 5: System level error modeling for Configs A, B and C at a retention period of one hour.

Chip-level ECC: We also examined single-bit and double-bit corrections for every 64-bits from each STT-MRAM chip servicing part of the data for a cache line request. The implementation feasibility of chip-level ECC has been shown in [14]. This contrasts to the system level solutions we discussed earlier where error detection and correction is done at 64-byte granularity. Table 2 summarizes the resulting error rates at the same 64-Byte granularity of system-level memory accesses (for which we set 1e-17 as acceptable error rate). Even double-bit correction at the chip-level is unable to qualify even for the higher Δ levels and most optimistic material configuration– Config–C. Note that the within-chip storage overheads for the correction schemes considered here

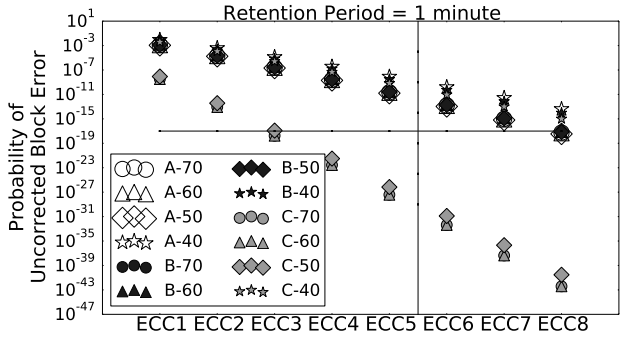

Figure 6: **System level error modeling for Configs A, B and C at a retention period of one minute.**

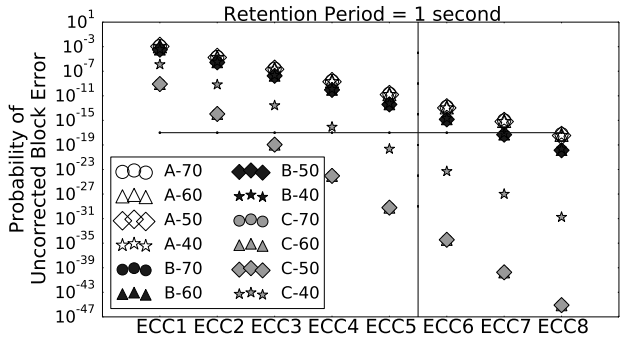

Figure 7: **System level error modeling for Configs A, B and C at a retention period of one second.**

would be 12.5% for single-bit correction and 21.9% for double-bit correction.

Table 2: **STT-MRAM probability of an access with an uncorrected error at Δ = 70,60 and 50 for different single-bit and double-bit chip-level ECC – no configuration shows acceptable error rate**

	A,24h	B,24h	C,24h	C,1h
Total BER	8.77e-5	4.47e-05	6.98e-08	6.97e-08
Block (64B) Uncorrected Error Rates				
MChip-1-70	1.33e-04	3.37e-05	8.25e-11	8.25e-07
MChip-2-70	2.46e-07	3.28e-08	1.24e-16	1.24e-16
MChip-1-60	1.23e-04	3.37e-05	1.55e-10	8.44e-11
MChip-2-60	2.27e-07	3.28e-08	3.23e-16	1.29e-16
MChip-1-50	3.45e-04	1.85e-04	6.13e-05	1.19e-07
MChip-2-50	1.07e-06	4.21e-07	8.00e-08	6.81e-12

Overheads of error correction: For the BCH codes, correction latency at current buffer chip frequencies (GHz) is linear in the number of bits that need to be corrected. Investing in stronger error correction capabilities at the buffer level has a minimal effect on performance –(a) because the implementation of codes have become fast, and (b) because the detection overhead added to all read accesses is very small (1 memory cycle) and the correction overhead can be paid quite infrequently by adopting variable latency decoding techniques, e.g., 2 memory cycle overhead for 1-2 bit error correction and 16 memory cycle overhead for up to 8-bit error correction (like ECC-8). Memory controllers and buffer chip already have area devoted for ECC for other purposes (bus transfer protection, chip-kill) and codes like BCH can be realized with relatively small area demands.

The storage overhead for external-to-chip implementation for BCH code is also modest, about 11 bits per bit to be corrected. For the ECC5 code (the highest possible within the 12.5% storage overheads for DDRx-DRAM ECC frameworks) this would amount to 55 bits (10.7%) for a payload of 64B=512 bits and 88 bits for ECC8 (17.2%). With a payload of 128 bytes the overhead percentages would drop in half allowing even ECC10 to be accommodated.

Chip-level error correction solutions will be critical for environments without off-chip correction capabilities (embedded systems), but they would need more improvements than we have projected in technology and/or lower acceptability criteria to be feasible, as seen earlier. The bit efficiency of the codes also worsens when applied for smaller payloads leading to higher storage overheads for similar correction capability. Further the correction latency overhead (higher at typically lower clock speeds within the memory device), even if not tremendous, would need to be accounted for in each access under the current fixed-latency access memory device access protocols. For these reasons, we believe ECC to address the STT-MRAM read sense and write errors should be done at the MC/buffer chip level in server memory systems. To examine the performance impact of frequent scrubbing and ECC we undertake simulation studies, discussed in subsequent sections.

5. METHODOLOGY FOR SIMULATIONS

5.1 Architectural Model

The baseline processor model integrates eight cores (4 issue, out-of-order, 32KB L1 and 4MB shared L2), and supports a DDR4-1600 memory subsystem with one independent, address-interleaved memory channel. Our memory subsystem model (DIMM structure, timing, and power) follows the Micron DDR4 DRAM specification [16]. The parameters of the memory system, and the DDR4 DRAM power model are shown in Tables 5 and 6. We implement our models by extending the SESC simulation environment [22] with DRAMSim2 [23], which has been modified to model JEDEC's DDR4 environment. DRAMSim2 is suitably modified with additional timing parameters for modeling STT-MRAM. While we model a single memory channel for simplicity, our results are equally valid for multi-channel memory systems with workloads scaled suitably for the increased levels of memory capacity and bandwidth.

5.2 Configurations and Applications

Table 3 shows the configurations discussed in the simulation studies. This selected set of configurations allows us to examine the full range of overheads associated with the feasible STT-MRAM option - Config C - with least error mitigation to most and compare with DRAM. DRAM-NRF denotes DRAM with no penalty for refresh to call out the impact of refresh.

We evaluate nine memory-intensive parallel applications, running eight threads each, to completion. Our parallel workloads constitute a good mix of scalable scientific programs from different benchmark suites, as shown in Table 4.

Table 3: Configurations Evaluated

Name	Type	Access Size	Reliabiliy
DRAM	1T-1C	1 KB	
DRAM-NRF	1T-1C	1 KB	No Refresh Penalty
C-24hΔ70	1T-1R	8 B	ECC 3b/64B, $\Delta = 70$, Scrub 24h
C-1mΔ50	1T-1R	8 B	ECC 3b/64B, $\Delta = 50$, Scrub 1min
STT-NoScrub-NoEcc	1T-1R	8 B	No ECC, No Scrubbing
C-24h-NoEcc	1T-1R	8 B	No ECC, Scrub 24h
C-1m-NoEcc	1T-1R	8 B	No ECC, Scrub 1min

Table 4: Applications Studied

Simulated parallel applications and their input sets		
Data Mining		[21]
scalparc	Decision Tree 125k pts., 32 attributes	
NAS OpenMP		[4]
mg	Multigrid Solver, Class A	
cg	Conjugate Gradient, Class A	
SPEC OpenMP		MinneSpec-Large [3]
swim-omp	Shallow water model	
equake-omp	Earthquake model	
art-omp	Self-organizing Map	
Splash-2		[29]
ocean	Ocean movements, 514 x 514	
fft	Fast Fourier transform, 1M points	
radix	Integer radix sort, 2M integers	

Table 5: Memory Timing and Configuration

DDR4-1600 DRAM Timing			
tRCD	11	tRRD	4
tCL	11	tRTRS	1
tWL	10	tRAS	20
tCCD	4	tRC	39
tWTR	11	BL/2	4
tWR	11	tCKE	4
tRTP	6	tRFC	280
tRP	11	tREFI	7.8μs
STT-MRAM Timing			
tSC	19		
tWC	19		
tAWL	1		
tPRE_STT	2		
tPRE_SA	4		

Main Memory configuration	
Transaction Queue	32 entries
Peak Data Rate	12.8 GB/s
DRAM bus frequency	1600 MHz
Number of Channels	1
DIMM Configuration	dual rank
Number of Banks	16/rank
Number of Columns	8192/bank
Device size	8Gb
Row Buffer Size	1 KB
Address Mapping	Page Interleave
Row Policy	Open Page
Burst Length	8

5.3 Power Model

Power supply current (IDD) consumption numbers for different memory types have been estimated using a memory power model developed for this study, and are shown in Table 6. The power model estimates power consumption based on commodity (DDR4) DRAM array structure and chip architecture with necessary changes to cell and sensing schemes for STT-MRAM operations and banklet architecture. To accurately estimate power consumption numbers at different operational modes, the total chip power or energy consumption has been broken down into several sub modules and their unit components as follows:

- Cell and S/A: B/L sensing, B/L precharging, S/A read current, cell write current
- RAS chain: W/L activation, W/L precharging, row decoder
- Data path: Local I/O, Global I/O, 2nd S/A, Full-swing I/O (DIO), Data pipeline
- CLK and control: Command and address CLK tree, command decoder and logic, address pipeline

Table 6: IDD values for the configurations considered in this study running a DDR4-1600 framework.

Current	DRAM (mA)	STT-INF (mA)	STT-Scrub-xx (mA)
IDD0	22	106	18
IDD2N	3	3	3
IDD2P	13	13	13
IDD3N	6	6	6
IDD3P	17	17	17
IDD4R	55	55	62
IDD4W	60	72	72
IDD5	88	0	0

- I/O: Data, command and Address input receiver, DLL, I/O CLK tree, data output buffer and DC Generators

Components in each unit have been modeled using two categories - capacitive (e.g. I/O lines and CLK signals) and constant current (e.g. cell write current, input receiver, etc.). Cell access transistor and B/L sense amplifiers were modeled at transistor level using 22nm PTM (Predictive Technology Model) for low-power application with high-K metal gate [1]. B/L sense amplifier circuit was based on [24]. Modeling for unit capacitance for different signals used micro stripline model with different line space and inter-metal layer thickness according to conventional DRAM design. Cell read and write current are critical to STT-MRAM power numbers. In this paper, write current of 40uA was used based on [11], and 10uA was assumed as the cell read current based on relative read to write current/energy ratios found in other work.

Table 6 lists the IDD parameters for DRAM and STT-MRAM as a result of the above power modeling methodology. The differences among the configurations are in IDD0 (activate-precharge), IDD4R (Read), IDD4W (Write) and IDD5 (Refresh) currents. Current for sensing is part of IDD0 for DRAM and STT-INF and so is higher for them. STT-INF with current-mode sensing has a much higher sensing current than for DRAM (more than an order of magnitude higher) and so its IDD0 is higher. IDD4R for the STT-MRAM configurations (with various scrubbing intervals) are higher as the sensing is part of the read command driven activity for them. But because sensing is limited to 64 bits for these configurations they show a much smaller increase in IDD4R with respect to DRAM/STT-INF than the increase in IDD0 for STT-INF (for which sensing is at full row=8192 bits granularity). IDD4W for all the STT-MRAM variants is same as they all write just the data (corresponding cells) for a particular write. Versus DRAM, the IDD4W values are still higher as writing to the STT-MRAM cells requires higher current (changing spin polarization versus charge sharing).

5.4 Error Model

The uncorrected block error rate for the various STT-MRAM device configurations that was calculated using our analytical model are provided as inputs to the simulator (DRAMSim2). At the memory controller, when a read completes, we check the probability that the block has an error. If an error is seen, there is a latency introduced to model the ECC engine (16 memory cycles for ECC8, 4 memory cycles for ECC3) for the correction. The read data is sent back to the upper level caches after this penalty has been observed. To model

the scrubbing operation on STT-MRAM devices in DRAMSim2, the configuration file is first changed to include the appropriate scrubbing frequency for the devices under consideration. The row address for the scrubbing operation is calculated in a similar way to DRAM refresh. Next, at the memory controller level, when the scrubbing frequency is met (tSCRBI), we issue a read operation to the corresponding row. When the read is completed, we check for any retention errors (using retention error probability), and if found, we model the error correction with a delay and then issue a write operation to the same row address. The scrub-introduced read and write operations show up in the memory controller queue like regular memory access requests.

6. OVERHEAD FOR ERROR CORRECTION

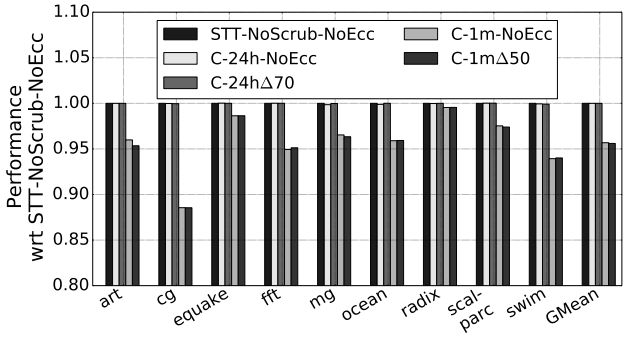

Figure 8: **Performance (higher is better) for the configurations studied**

We examine the performance impact of the error correction solutions using the simulation frame-work and workloads discussed earlier. The results are captured in Figure 8. As baseline we use the STT-MRAM system with no scrubbing and no error injection (`STT-NoScrub-NoECC`). To separate scrubbing overhead from error correction (ECC) overhead we also run the workloads with scrubbing but no error injections – this would give us the upper-bound on the scrubbing overhead. We evaluate two STT-MRAM options at the opposite ends of the spectrum when considering ECC requirements - Config–C with $\Delta = 70$ that would require the least system-level error correction overhead, and Config–C with $\Delta = 50$ which requires the most. For both configurations, we choose a correction capability of three bits per cache line. Recall from section 4.3, these configurations fall under feasible solutions in our analytical framework.

Because of the low level of errors on a demand access the overhead of error correction is very low - this is the difference in overhead between the `C-24h-NoECC, C-1m-NoEcc` options and `C-24hΔ70, C-1mΔ50` options. While we have shown Config–C configurations that pass with ECC3, the same is true for those that require higher levels of ECC because of the low incidence of on-demand ECC correction. The correction mechanisms are required to deliver reliable memory but do not come in the way of performance.

This is not true with scrubbing – for lower Δ, the impact seems to range from 1.4% to 11.5% for the 1m interval. Note that while we examined scrubbing at 1s interval as a solution for low Δ, its infeasibility becomes apparent when we examine its performance impact - for the 2-rank/channel configuration, a 1-sec scrub interval would require between 16GBs-32GB/s bandwidth (read to read+error-correct+write) just for scrubbing on a channel whose peak bandwidth is only 12.8GB/s!

7. PERFORMANCE OPTIMIZATIONS

Limiting sensing to the access size for STT-MRAM instead of row size prevents leveraging row-buffer locality in contrast to DRAM. Error mitigation can also impact performance as seen earlier. However, STT-MRAM does not have the same limitations as DRAM because of the capacitive nature of the latter. We propose (a) a protocol optimization – *Command Compaction*, and (b) an architecture optimization -*Banklets* – exploiting STT-MRAM's inherent differences with DRAM while maintaining compatibility with DRAM access commands and interface.

7.1 Command Compaction

Due to its capacitive nature, DRAM suffers from two important issues: (a) destructive reads, and (b) refresh. STT-MRAM, unlike DRAM, does not present such limitations, allowing for a number of effective timing optimizations in the memory access protocol. We exploit the non-capacitive nature of STT-MRAM cells by removing the *restore* phase after data sensing and the *recovery* phase after data writes.

Tighter Sequencing of WAR and RAW Operations:

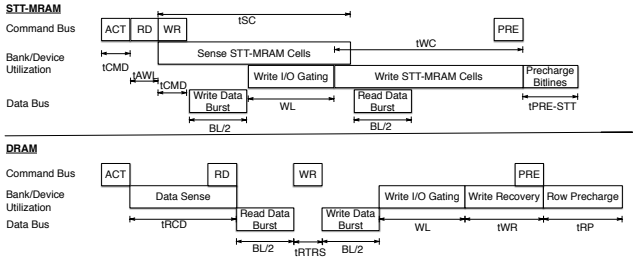

Figure 9: **Timing diagrams for a *write* followed by a *read* command sequence for STT-MRAM and DRAM. Timing parameters are described in table 5.**

The optimization to the STT-MRAM memory access protocol comes from being able to tightly sequence back-to-back read and write operations. Consider the following command sequence: *activate bank0*, *read bank0*, *write bank1*, *precharge bank1*. In DRAM, the burst for the *write* command follows after the burst for the *read* command has been completed. However, in STT-MRAM, since we have moved the sensing to the *read* command, this additional increase in read latency(tSC) can be compensated by overlapping the burst of the *write* command with the sensing operation of the *read* command. Figure 9 plots the timing diagram for the above tight sequencing of write-after-read operations in our proposed STT-MRAM memory access protocol. During the sensing of the STT-MRAM cells (tSC window), the data for the write operation is transmitted to the write drivers for the appropriate bank. Once the *read* command has been issued, there is a small window of cycles during which the *write* command can be issued to guarantee that the read and write

bursts do not collide, either on the data bus or in the external I/O gating circuitry[3]. The rationale behind the following optimization is that tRTRS + WL + BL/2 $\leq$ tSC. Therefore, for the above described command sequence, the total delay for STT-MRAM would be: tAWL + tCMD + BL/2 + WL + tWC + tPRE-STT (37 cycles). For DRAM, the total delay amounts to tRCD + BL/2 + tRTRS + BL/2 + WL + tWR + tRP (52 cycles). This compact interleaving of *read* and *write* commands can occur between different ranks, different banks and between different banklets (Section 7.2) in the same bank. The same principles hold for compacting write-after-read command sequences as well.

7.2 Banklet Architecture

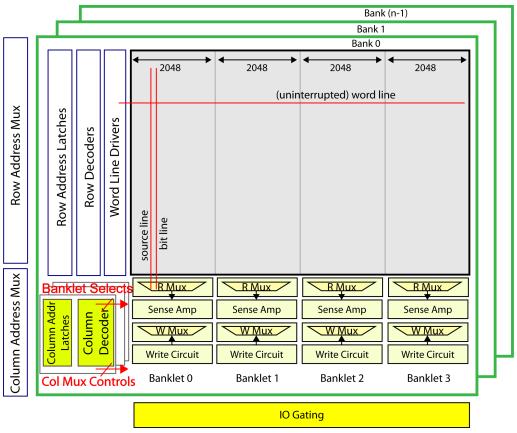

Figure 10: STT-MRAM bank internal structure showing four banklets. Each banklet is 2048 columns wide, and uses a 32:1 multiplexing topology, reducing 2048 bit lines to 64 bit lines, which are then fed to the sense-amps/write drivers.

To overcome the loss in row buffer locality, we propose a new architecture which logically segments the memory array into smaller units called *banklets*. Figure 10 presents the internal structure of a bank with banklets. A bank building block features (i) one uninterrupted 2Kx8K tiled cell array with continuous word lines, bit lines and source lines for optimal density and (ii) peripheral circuits in four 2K-wide adjoined instances, i.e., *banklets*. Each of them contains independently controlled read and write column switches, sense amplifiers and write driver circuits. The 2 MSB column address bits decode banklets and 5 LSB column address bits decode the read and write column switches and multiplexers. All banklets belonging to the same bank share the same active row. Each banklet serves a single data burst and is connected to a fraction of the bit lines. Our proposed architecture preserves the memory array (bank) and row decoder as a single unit, but divides the external circuitry (sense-amplifier, write drivers, column multiplexers etc.) into per-banklet segments. When a given word line within a bank is selected, the four banklets in it can perform concurrent and independent read/write operations under one shared row address, re-couping some of the row-buffer locality lost in going to 8B accesses. Trade-off between peak power and locality exploitation can be achieved by supporting an appropriate number of banklets to a bank.

The banklet functionality is realized, by decoding the necessary most significant column address bits, without physically partitioning the cell array or instituting electrically isolated hierarchical word line architecture. Alternative design choices for smaller cell-array component, like a hierarchical word line design, carry the overhead of decreased density, higher wiring complexity, and increased manufacturing cost. This is because shape interactions between the many at-design-rule (word line drivers and decoders) and sub-design-rule (cell arrays) regions impose area efficiency penalties and additional lithographic steps in mask preparation, double/triple patterning, and fabrication process. In contrast, our approach of simultaneously preserving cell array integration efficiency and peripheral circuit compactness contributes to the desirable attributes of low latency, low power, intra-bank access concurrency and interface simplicity at the sub-array level making it more viable for DRAM-like (high density) main memory applications.

7.3 Performance Evaluation

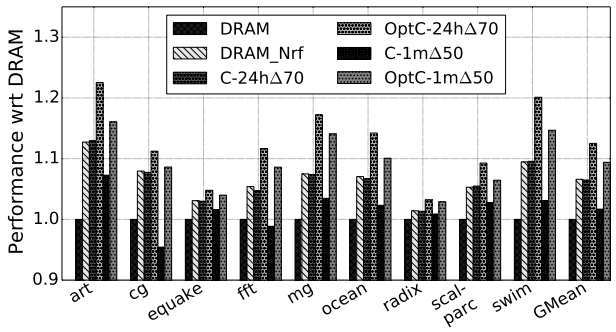

Figure 11: Performance (higher is better) for the configurations studied, normalized to the DRAM configuration.

Figure 11 shows the performance obtained by the configurations considered in this study normalized to the performance of the DRAM configuration. A critical observation can be made from this plot: Performance loss due to DRAM refresh is significant in high density memory. The DRAM-Nrf configuration, modeling zero refresh penalty, is 7% better on average, and 9% better on the applications that are highly memory bound.

The STT-MRAM configuration with a 24 hour scrubbing period, and an error correction capability of three bits per cache line – C-24hΔ70 – has an improvement in performance of 6.5%. Recall that STT-MRAM designs do not suffer from precharge and write recovery issues. Also, a scrubbing period of 24 hours has no impact on performance as seen from figure 8. Hence, the improvement in performance when compared to DRAM. As we increase the frequency of scrubbing to one minute – C-1mΔ50 – scrub requests compete more with normal requests and the speedup wrt DRAM goes down to 1.7%. Applying our performance optimizations to both STT-MRAM configurations – OptC-24hΔ70 and OptC-1mΔ50 – helps improve performance by 12.5% and 9% respectively. Next we analyze these gains further.

[3]This window include the tRTRS delay required to change the direction of the bus master.

Command Compaction:

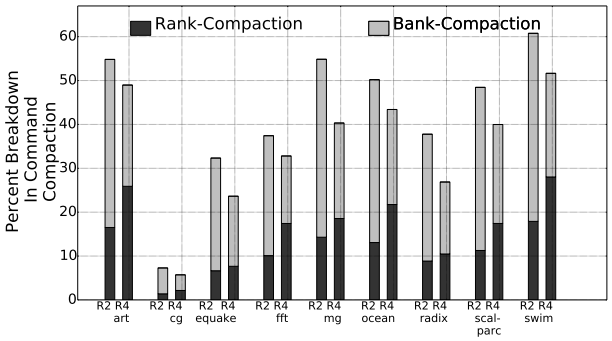

Figure 12: Percentage breakdown in Command Compaction for a two rank (R2) and four rank (R4) system, when running the OptC-24hΔ70 configuration on the parallel workloads.

OptC-24hΔ70 and OptC-1mΔ50 designs utilize our command compacting protocol optimization, enabling tight interleaving of read and write operations. We plot the percentage of operations issued by the memory controller that benefit from the compaction of commands due to the new access protocol, for a two rank (*R2*) and a four rank (*R4*) system running the OptC-24hΔ70 configuration, in figure 12. *Rank-Compaction* refers to the percentage of read and write operations issued to different ranks that benefit from the tighter interleaving allowed by our proposed protocol. *Bank-Compaction* refers to percentage of read and write operations issued to different banks within a rank by making use of the tighter interleaving,

A higher compaction percentage indicates improved performance because read and write operations can operate concurrently in different banks or ranks in the system, leading to a better utilization of the memory bandwidth. From the plot, we can observe that for the two rank system (*R2*), the applications *art, fft, mg, ocean, scalparc* and *swim*, have a higher percentage ($\sim$ 50%) of reads and writes that make use of the improved command compaction. Consequently, they also show improved performance over the other applications (figure 11). The applications *radix* and *equake*, are not as memory intensive and do not benefit from command compaction despite having a reasonable compaction percentage ($\sim$ 30%). Finally, the application *cg*, shows improved performance, but very low compaction percentage. However, from figure 11 we can infer that *cg* benefits mostly from the elimination of refresh overheads, and the reduction in precharge and write recovery latency – the configurations DRAM-Nrf and C-24hΔ70 have high improvements in performance when compared to DRAM.

We also observe that in a two rank system, most of the read-write compaction is due to banks. This is due to the address interleaving which places sequential addresses in consecutive banks in the same rank, before mapping them to different ranks (favoring bank-level parallelism over rank-level parallelism). Increasing the number of ranks in the system allows for higher *Rank-Compaction*, but reduces the overall compaction percentage, due to the increase in intrinsic parallelism of the system due to more ranks.

Banklet Architecture:

In our experiments, we found that the applications were unable to completely stress the memory system. However, the banklet architecture was designed to provide maximum benefit for workloads that have the capability of stressing the memory system with maximum load (due to the additional banklet level concurrency that can be obtained). To put maximal load on the memory system and assess the potential for banklet optimization, we run the memory simulator against address traces from the application suite providing requests as fast as the memory system can service them. With this address-trace-driven simulation, realized bandwidth is equivalent to the performance improvement potential and consequently the measure we examine.

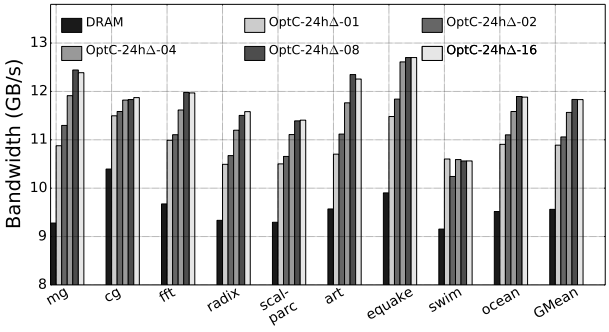

Figure 13: Performance (higher is better) for the DRAM, and OptC-24hΔ70 configurations with varying number of banklets (1-16), normalized to the DRAM configuration when running the memory traces from the parallel workloads.

Figure 13 shows the realized bandwidth when running these memory traces for OptC-24hΔ70 with different number of banklets. The number of banklets varies from one to sixteen. Under higher demands on the memory system, it shows that the use of banklets helps improve the utilization of the channel. The best performance is seen with 8 banklets which obtains a 22% increase in bandwidth on the average compared to DRAM. Note that we have used a 4-banklet configuration for OptC-24hΔ70 and OptC-1mΔ50 in all the results discussed in the rest of the paper.

7.4 Energy Efficiency Results

Figure 14 shows the breakdown in energy consumption for the parallel applications when running the DRAM, DRAM-Nrf,C-24hΔ70, OptC-24hΔ70, C-1mΔ50, and OptC-1mΔ50 configurations, normalized to the DRAM configuration. The total energy consumed by the application is divided into four groups as shown in the plot: (a) *BGND* – Background/idle energy, (b) *ACT-PRE* – for activation and precharge, (c) *BURST* – for reads and writes, and finally (d) *REFRESH* – for refresh (DRAM) or scrub STT-MRAM) commands.

Overall, our proposed OptC-24hΔ70 and OptC-1mΔ50 configurations have a reduction in energy consumption of 21% and 16% when compared to DRAM. The configurations using STT-MRAM devices (C-24hΔ70, C-1mΔ50, OptC-24hΔ70, OptC-1mΔ50), have lower activate-precharge energy consumed. This is because of moving the sensing operation of the cells to the read command from the activate command. Consequently, these configurations see an increase in burst

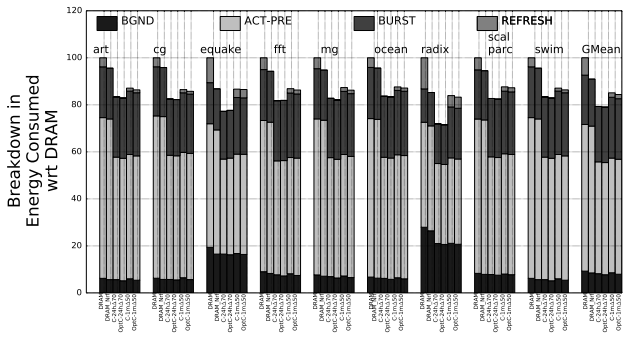

Figure 14: Percentage breakdown of total energy consumption (lower is better) when running the parallel workloads on the DRAM, DRAM-Nrf, C-24hΔ70 , OptC-24hΔ70 , C-1mΔ50 , OptC-1mΔ50 configurations, normalized to the energy consumed by the DRAM configuration.

energy consumed. The configurations C-24hΔ70 and OptC-24hΔ70 (as well as C-1mΔ50 and OptC-1mΔ50) have similar energy consumption, even though OptC-24hΔ70 (and OptC-1mΔ50) has improved performance. This is because the total dynamic activity for both configurations remain the same. The more frequent scrubbing for C-1mΔ50 (and OptC-1mΔ50) has a slightly higher energy cost and so C-24hΔ70 and OptC-24hΔ70 have the lowest energy consumption.

8. RELATED WORK

Zhang et al. [35] analyze the impacts of CMOS and MTJ process variations, switching uncertainties induced by thermal fluctuations and temperature on the performance and reliability of STT-RAM cells. Zhao et al. [36] characterize STT-MRAM failures due to "hard" and "soft" errors, and provide some efficient design solutions like self-enable switching circuitry to tackle these reliability issues. Yang et al. [32] and Emre et al. [10] show how combining circuit level techniques like W/L sizing of transistors, adjusting the pulse width of read and write operations and voltage boosting with strong ECC protection can help reduce the block failure rate. Bel et al. [5] propose the use of multi-bit error correction for STT-MRAM based arrays. They conclude that reducing the non-volatility constraints along with stronger ECC correction helps lower area overhead for the arrays. However, they only consider retention errors in STT-MRAM cells, and do not model a system memory hierarchy. Our work differs from [5] in that we provide a device level analytical model for calculating read, write and retention errors of an STT-MRAM cell, and use those BERs to systematically determine total uncorrected error rate when error detection and correction schemes are employed at different levels of the memory hierarchy.

Udipi et al [26] proposed subbanking in DRAMs. Their proposal split the word line in cache line sizes, making only that subset available when the word line is activated. The large overhead of the additional circuit in the word line limits the minimum size of the subbank. Their proposal moves the sensing to the CAS command, but the high performance is only achieved when a non-JEDEC interface is used. Gulur et al [12] also proposed subbanking in DRAMS. Their proposal differs from Udipi et al [26] by using sub-row buffers so the memory can still serve requests from one sub-bank while sensing a row in another sub-bank. Additional bits necessary to identify the subbank that needs to be activated requires extending the JEDEC DRAM device interface. The performance is achieved from the parallelism of the row buffers, however additional sub-row buffers add area overhead. Zhang et al [34] proposes half-DRAM, where they split the MATs (multiple cell matrices) in two halfs and activate each one separately - maintain the internal architecture and decoding structure unlike [26] and [12]. They also propose timing enhancements based on relaxed constraints if different halfs are used. Our performance optimization proposals differ from the other three by preserving the JEDEC interface, not disrupting array efficiency or adding signficant periphery overheads and using STT-MRAM cell characteristics to aggressively change the timing between commands.

9. CONCLUSIONS

DRAM faces significant challenges scaling to smaller nodes. STT-MRAM offers an alternative with its scaling-friendly, spin-polarization-based resistive storage. But it faces significant reliability challenges and (power related) integration challenges with current memory framework designed for DRAM.

STT-MRAM has power challenges because of much higher writing and sensing currents. We propose adaptations to the writing and sensing infrastructure within a DRAM device to better match it with STT-MRAM array characteristics, reducing the number of concurrent cell accesses and associated power. This allows for an implementable STT-MRAM based high-density main memory device compatible with DDR4 DRAM's logical interface and command structure meeting our STT-MRAM power goals.

We provide an analysis of the different error phenomena for STT-MRAM and identify suitable error correction methods - specifically, off-chip ECC (BCH codes) and scrubbing. The extent of correction required is dependent on improvements in STT-MRAM technology for read-sense and write errors. While retention errors can also be reduced with technology improvements, for direct DRAM replacement its not required. We show that with specific material-manufacturing improvements and adoption of suitable scrubbing and system-level ECC options STT-MRAM-based memory system can attain adequate resilience. We advocate off-chip methods due to minimal power, storage and latency overheads while integrating with current high-performance main memory organization frame-works. Extensive on-chip correction technologies will be needed along with even more material improvements if off-chip correction is not utilized.

We then propose two optimizations for performance – *command compaction* and *banklet architecture* to improve STT-MRAM performance with tighter scheduling and increased concurrency of operations while lowering energy costs. With our optimizations, STT-MRAM shows 12.5% higher performance than DRAM with 21% energy savings on average. If the STT-MRAM material requires higher scrub frequency, then we get 9% higher performance with 16% energy savings.

ACKNOWLEDGEMENTS

This work is sponsored in part by Defense Advanced Research Projects Agency, Microsystems Technology Office (MTO), under contract no. HR0011-13-C-0022. The views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

10. REFERENCES

[1] "Predictive Technology Model," http://ptm.asu.edu.

[2] F. M. Alzahrani and T. Chen, "On-chip TEC-QED ECC for Ultra-Large, Single-Chip Memory Systems," in *IEEE International Conference on Computer Design (ICCD)*, October 1994.

[3] V. Aslot and R. Eigenmann, "Quantitative Performance Analysis of the SPEC OMPM2001 Benchmarks," *sp*, vol. 11, no. 2, pp. 105–124, 2003.

[4] D. H. Bailey *et al.*, "NAS parallel benchmarks," NASA Ames Research Center, Tech. Rep., March 1994, tech. Rep. RNR-94-007.

[5] B. D. Bel, J. Kim, C. H. Kim, and S. S. Sapatnekar, "Improving STT-MRAM density through multibit error correction," in *Design, Automation & Test in Europe Conference & Exhibition, DATE 2014, Dresden, Germany, March 24-28, 2014*, 2014, pp. 1–6.

[6] K. C. Chun, H. Zhao, J. Harms, T.-H. Kim, J.-P. Wang, and C. Kim, "A Scaling Roadmap and Performance Evaluation of In-Plane and Perpendicular MTJ Based STT-MRAMs for High-Density Cache Memory," *Solid-State Circuits, IEEE Journal of*, vol. 48, no. 2, pp. 598–610, 2013.

[7] D.Bedau, H.Liu, J.Z.Sun, J.A.Katine, E.E.Fullerton, S.Mangin, and A. D. Kent, "Spin-transfer Pulse Switching: From the Dynamic to the Thermally Activated Regime," *Applied Physics Letters*, vol. 97(26), 2010.

[8] R. Dennard, "Field-effect Transistor Memory," US Patent 3387286, June 1968.

[9] R. Dorrance, "Modeling and Design of STT-MRAMs," Ph.D. dissertation, University of California, Los Angeles, 2011.

[10] Y. Emre, C. Yang, K. Sutaria, Y. Cao, and C. Chakrabarti, "Enhancing the Reliability of STT-RAM through Circuit and System Level Techniques." in *SiPS*. IEEE, 2012, pp. 125–130.

[11] M. Gajek, J. J. Nowak, J. Z. Sun, P. L. Trouilloud, E. J. O'Sullivan, D. W. Abraham, M. C. Gaidis, G. Hu, S. Brown, Y. Zhu, R. P. Robertazzi, W. J. Gallagher, and D. C. Worledge, "Spin Torque Switching of 20nm Magnetic Tunnel Junctions with Perpendicular Anisotropy," *Applied Physics Letters*, vol. 100, no. 13, pp. –, 2012. [Online]. Available: http://scitation.aip.org/content/aip/journal/apl/100/13/10.1063/1.3694270

[12] N. Gulur, R. Manikantan, M. Mehendale, and R. Govindarajan, "Multiple Sub-Row Buffers in DRAM: Unlocking Performance and Energy Improvement Opportunities," in *International Conference on Supercomputing 2012*, June 2012.

[13] S. J. Hong, "Memory Technology Trends and Future Challenges," 2010 IEEE International Electron Devices Meeting (IEDM), 2010, pp. 12.4.1 - 12.4.4.

[14] G. Jan, L. Thomas, S. Le, Y.-J. Lee, H. Liu, J. Zhu, R.-Y. Tong, K. Pi, Y.-J. Wang, D. Shen, R. He, J. Haq, J. Teng, V. Lam, K. Huang, T. Zhong, T. Torng, and P.-K. Wang, "Demonstration of Fully Functional 8Mb Perpendicular STT-MRAM Chips with Sub-5ns Writing for Non-volatile Embedded Memories," in *IEEE Symposium on VLSI Technology*, June 2014.

[15] J. Janesky, N. Rizzo, D. Houssameddine, R. Whig, F. Mancoff, M. DeHerrera, J. Sun, M. Schneider, H. Chia, S. Aggarwal, K. Nagel, S. Deshpande, T. Andre, S. Alam, C. Tan, J. Slaughter, S. Hellmold, and P. LoPresti, "Device Performance in a Fully Functional 800MHz DDR3 Spin Torque Magnetic Random Access Memory," in *Memory Workshop (IMW), 2013 5th IEEE International*, 2013, pp. 17–20.

[16] JEDEC, "JEDEC DDR4 SDRAM Standard," www.jedec.org/standards-documents/docs/jesd79-4.

[17] T. Kawahara, R. Takemura, K. Miura, J. Hayakawa, S. Ikeda, Y. Lee, R. Sasaki, Y. Goto, K. Ito, T. Meguro, F. Matsukura, H. Takahashi, H. Matsuoka, and H. Ohno, "2Mb Spin-Transfer Torque Ram (SPRAM) with Bit-by-Bit Bidirectional Current Write and Parallelizing-Direction Current Read," in *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2007 IEEE International*, 2007.

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

[19] J. Mukundan, H. Hunter, K. hyoun Kim, J. Stuecheli, and J. F. Martinez, "Understanding and Mitigating Refresh Overheads in High-density DDR4 DRAM Systems," in *ISCA '13*. New York, NY, USA: ACM, 2013, pp. 189–200. [Online]. Available: http://doi.acm.org/10.1145/2485922.2485939

[20] H. Noguchi, K. Kushida, K. Ikegami, K. Abe, E. Kitagawa, S. Kashiwada, C. Kamata, A. Kawasumi, H. Hara, and S. Fujita, "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 *2013 Symposium on VLSI Technology (VLSIT)*, June 2013.

[21] J. Pisharath, Y. Liu, W. Liao, A. Choudhary, G. Memik, and J. Parhi, "NU-MineBench 2.0," Northwestern University, Tech. Rep., August 2005, tech. Rep. CUCIS-2005-08-01.

[22] J. Renau, B. Fraguela, J. Tuck, W. Liu, M. Prvulovic, L. Ceze, S. Sarangi, P. Sack, K. Strauss, and P. Montesinos, "SESC Simulator," January 2005, http://sesc.sourceforge.net.

[23] P. Rosenfeld, E. Cooper-Balis, and B. Jacob, "DRAMSim2: A Cycle Accurate Memory System Simulator," 2011.

[24] R. Takemura, T. Kawahara, K. Miura, H. Yamamoto, J. Hayakawa, N. Matsuzaki, K. Ono, M. Yamanouchi, K. Ito, H. Takahashi, S. Ikeda, H. Hasegawa, H. Matsuoka, and H. Ohno, "A 32-Mb SPRAM With 2T1R Memory Cell, Localized Bi-Directional Write Driver and '1'/'0' Dual-Array Equalized Reference Scheme." in *J. Solid-State Circuits*, vol. 45, 2010, pp. 869–879. [Online]. Available: http://dblp.uni-trier.de/db/journals/jssc/jssc45.html

[25] K. Tsuchida, T. Inaba, K. Fujita, Y. Ueda, T. Shimizu, Y. Asao, T. Kajiyama, M. Iwayama, K. Sugiura, S. Ikegawa, T. Kishi, T. Kai, M. Amano, N. Shimomura, H. Yoda, and Y. Watanabe, "A 64Mb MRAM with Clamped-reference and Adequate-reference Schemes," in *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2010 IEEE International*, 2010, pp. 258–259.

[26] A. N. Udipi, R. Balasubramonian, N. Muralimanohar, A. Davis, N. Chatterjee, and N. P. Jouppi, "Rethinking DRAM Design and Organization for Energy-Constrained Multi-Cores," in *The 2010 ACM IEEE International Symposium Computer Architecture (ISCA'10)*, June 2010.

[27] J. Wang, X. Dong, and Y. Xie, "Enabling High-performance LPDDRx-compatible MRAM," in *Proceedings of the 2014 International Symposium on Low Power Electronics and Design*, ser. ISLPED '14. New York, NY, USA: ACM, 2014, pp. 339–344. [Online]. Available: http://doi.acm.org/10.1145/2627369.2627610

[28] M. Ware, K. Rajamani, M. Floyd, B. Brock, J. C. Rubio, F. Rawson, and J. B. Carter, "Architecting for Power Management: The IBM POWER7 Approachi," in *The 16th IEEE International Symposium on High-Performance Computer Architecture (HPCA-16)*, January 2010.

[29] S. C. Woo, M. Ohara, E. Torrie, J. P. Singh, and A. Gupta, "The SPLASH-2 Programs: Characterization and Methodological Considerationsi," in *ISCA-22*, 1995.

[30] D. C. Worledge, G. Hu, P. L. Trouilloud, D. W. Abraham, S. Brown, M. C. Gaidis, J. Nowak, E. J. O'Sullivan, R. P. Robertazzi, J. Z. Sun, and W. J. Gallagher, "Switching Distributions and Write Reliability of Perpendicular Spin Torque MRAM," in *Electron Devices Meeting (IEDM), 2012 IEEE International*, 2010, pp. 12.5.1–12.5.4.

[31] Y. Xie, "Future Memory and Interconnect Technologies," in *Design, Automation Test in Europe Conference Exhibition (DATE), 2013*, 2013, pp. 964–969.

[32] C. Yang, Y. Emre, Y. Cao, and C. Chakrabarti, "Improving Reliability of Non-volatile Memory Technologies Through Circuit Level Techniques and Error Control Coding." *EURASIP J. Adv. Sig. Proc.*, 2012.

[33] J. Yoon and H. Hunter, "Emerging Non-volatile Memory Technology and its Implications to Server Compute Systems," Presented at

Non-volatile Memory Workshop (NVM 2013), VLSI Symposium, 2013.

[34] T. Zhang, K. Chen, C. Xu, G. Sun, T. Wang, and Y. Xie, "Half-DRAM: A High-bandwidth and Low-power DRAM Architecture from the Rethinking of Fine-grained Activation," in *Computer Architecture (ISCA), 2014 ACM/IEEE 41st International Symposium on*, June 2014, pp. 349–360.

[35] Y. Zhang, X. Wang, and Y. Chen, " STT-RAM Cell Design Optimization for Persistent and Non-persistent Error Rate Reduction: A Statistical Design View," in *Proceedings of the International Conference on Computer-Aided Design*, ser. ICCAD '11, 2011, pp. 471–477.

[36] W. Zhao, Y. Zhang, T. Devolder, J.-O. Klein, D. Ravelosona, C. Chappert, and P. Mazoyer, "Failure and Reliability Analysis of STT-MRAM." *Microelectronics Reliability*, vol. 52, no. 9-10, pp. 1848–1852, 2012.