Abstract—Spin Transfer Torque Magnetoresistive RAM (STT-MRAM) has been recently deemed as one promising main memory alternative for high-end mobile processors. With process technology scaling, the amplitude of write current approaches that of read current in deep sub-micrometer STT-MRAM arrays. As a result, read disturbance errors (RDEs) emerge. Both high current restore required (HCRR) reads and low current long latency (LCLL) reads can guarantee read reliability and utterly remove RDEs. However, both of them degrade system performance, because of extra restores or a longer read latency. And neither of them always achieves the better performance when running a wide variety of applications. In this paper, we present two architectural techniques to boost read performance for STT-MRAM based main memories in the presence of RDEs. We first propose Smash Read (S-RD) to shorten the latency of HCRR reads by injecting a larger read current. We further introduce Flexible Read (F-RD) to dynamically adopt different types of read schemes, S-RD and LCLL, to maximize main memory system performance. On average, our techniques improve system performance by 9~13% and reduces total energy by 4~8% over all existing read schemes including HCRR and LCLL.

I. INTRODUCTION

Power hungry DRAM based main memory significantly hurts the battery lifetime of mobile and handheld computing platforms including tablets and smartphones. For instance, DRAM main memory with periodic refreshes consumes 38.5% of total energy in a smartphone [7]. In near future, the increasingly frequent DRAM refreshes will further limit the whole system performance [19]. To improve energy efficiency, recent studies [23, 16, 23, 11] architect Spin Transfer Torque Magnetoresistive RAM (STT-MRAM) as an alternative to traditional DRAM in main memory systems.

STT-MRAM has become an off-the-shelf technology to implement energy efficient off-chip main memories. Unlike DRAM, the non-volatility of STT-MRAM completely removes the necessity to do refreshes in main memories [11, 23]. Compared to DRAM, the STT-MRAM based main memory saves 20% energy, but still maintains a similar performance to DRAM [32]. Moreover, several memory manufacturing companies, like Everspin [23, 11] and Hitachi [31, 21], have published their STT-MRAM chips with DDRx interface.

With process technology scaling, the amplitude of STT-MRAM write current decreases significantly due to the reduced cell area and the improvement of magnetic material [10]. On the contrary, it is difficult for the STT-MRAM read current to scale with process technology, since it depends on the sensitivity and the resolution of sense amplifiers, which cannot guarantee a reliable sensing with a small current [2]. Therefore, the STT-MRAM read current does not obviously diminish and stays almost the same since 180nm process technology [21]. At 32nm process node, the read and write currents are close enough to trigger a non-trivial number of Read Disturbed Errors (RDEs) [31], i.e., the contents in cells may be accidentally changed during read operations.

To mitigate emerging RDEs, two methods have been proposed for future STT-MRAM memories. The first way is to perform a restore writing the sensed out data back into cells after each read operation to ensure the data integrity [31, 30]. This method is called destructive high current restore required (HCRR) read. Although the critical read latency is not prolonged, one extra restore after each sensing operation increases memory bank busy time and may block the following reads. The other approach is to redesign the sense amplifier, so that the cell content can be sensed out with a smaller amplitude yet longer duration current [20, 14]. The redesigned sense amplifier [20, 14] has small enough read current that does not introduce any RDE, but it does increase the critical read latency. This method is referred to as non-destructive low current long latency (LCLL) read. Neither method can always achieve the maximum performance when running different applications or different execution phases in one benchmark. In this paper, we present Flexible Read to adaptively adopt destructive read and non-destructive read to maximize system performance for a large spectrum of applications. In general, our contributions can be summarized as follows:

- We explore the trade-off between read current and read latency to exhibit the possibility to accelerate the destructive HCRR reads. We propose Smash Read to shorten read latency, if the destructive HCRR read is adopted to sense data in deep sub-micrometer STT-MRAM arrays.
- We present Flexible Read to dynamically switch between Smash Reads and LCLL reads to improve main memory performance while executing a wide range of benchmarks. By monitoring the free read / write queue entry number in the on-chip memory controller, Flexible Read adaptively issue destructive Smash Reads or non-destructive LCLL...
reads to minimize memory bank occupation time.

- We evaluated our proposed designs. On average, our results show that our techniques improve performance by 9~13% and reduce energy consumption by 4~8% over the state-of-the-art read techniques in STT-MRAM based main memories.

The rest of this paper is organized as follows. Section II introduces the STT-MRAM background and our design motivation. Section III elaborates the design details of our techniques. Section IV describes the experimental methodology. We report and analyze simulation results in Section V. Section VI describes more related works. We conclude the paper in Section VII.

II. BACKGROUND AND MOTIVATION

A. STT-MRAM Basics

STT-MRAM is one type of emerging non-volatile memory technologies. Unlike DRAM, STT-MRAM takes advantage of the resistance of MTJs (Magnetic Tunnel Junctions), rather than the electrical charges in capacitors, to store data. Figure 1(a) shows one STT-MRAM cell example with one transistor and one MTJ (1T-1MTJ) [9]. The MTJ contains two ferromagnetic layers partitioned by an oxide barrier layer (MgO). The magnetization direction of one ferromagnetic layer is fixed (reference layer), while the other (free layer) can be switched by an electrical pulse. When the magnetic fields of two layers are parallel, the MTJ resistance is low and the cell indicates a ‘0’. If the magnetic fields are anti-parallel, the MTJ resistance is high and the cell represents a ‘1’. The STT-MRAM cell size below 30nm is smaller than 8F² [4], and is comparable to DRAM cell size 6F².

![STT-MRAM Cell](image)

(b) Write Current Scaling [31].

Fig. 1. 1T-1MTJ cell and its scaling trend.

To write ‘1’ (‘0’) a STT-MRAM cell, a positive (negative) voltage applied between source line (SL) and bit line (BL) produces a current flow from SL to BL. The write current is large enough to change the magnetization direction of MTJ free layer. The read operation is similar to the write operation [8], but only creates a smaller amplitude and a shorter duration current to sense the data.

A flurry of previous studies [29, 26] adopt STT-MRAM to replace at least a portion of SRAM to build a large capacity and low power on-chip caches. Several more recent works [32, 16] architect STT-MRAM with (LP-)DDRx interface as main memories. With little modification, the array structure and the micro-command scheduling of STT-MRAM can be fully compatible with off-the-shelf (LP-)DDRx interfaces [32, 16, 23, 11]. Therefore, it is easy for various existing computing platforms, e.g., mobile devices or servers, to integrate STT-MRAM into their main memory architectures.

B. Inevitable Read Disturbance

The write current amplitude of STT-MRAM scales with its MTJ area. The relation between write current and MTJ area is shown in Equation 1 [9]:

\[ I_w = A \cdot \left( J_{c0} + \frac{C}{T_w} \right) \]  \hspace{1cm} (1)

where \( I_w \) denotes the STT-MRAM write current; \( A \) indicates the MTJ area; \( J_{c0} \) represents the critical current density at zero temperature; \( T_w \) is the write current duration; \( C \) and \( \alpha \) are fitting parameters. A recent work [3] has proved that STT-MRAM is able to scale and stay in a small cell size beyond 22nm. The area of MTJ diminishes exponentially with the shrunk feature size. As Figure 1(b) shows, the reduction on write current resulting from a smaller MTJ area is significant over different process technology nodes. The trend of write current reduction continues to even 15nm.

On the contrary, the STT-MRAM read current does not scale with feature size. Figure 1(b) compares the read and write currents with different feature sizes. For large process technology nodes, e.g., 180nm, the read current is much smaller than the write current. And thus, the difference between read and write currents is significant. However, for small feature sizes, e.g., 32nm, shrinking read current is challenging, as it is very difficult for conventional STT-MRAM sense amplifiers to sense data correctly using below 20μA current [34]. So, the read current stays about the same and the read margin is fast decreasing in deep sub-micrometer STT-MRAMs. At 32nm node, the read and write currents are so close such that some reads may disturb (i.e., write) their being-read cells [31]. This is referred to as read disturbance in STT-MRAM. We adopt Equation 2 from [9] to describe read disturbance rate:

\[ P = 1 - \exp\left( -\frac{t}{\tau} \exp\left( -\Delta_0 \left( 1 - \frac{I}{I_{c0}} \right) \right) \right) \]  \hspace{1cm} (2)

where, \( P \) is the read disturbance rate; \( I \) denotes the read current; and \( t \) indicates the read pulse width. \( \tau \) is inverse of the attempt frequency; \( \Delta_0 \) indicates magnetic memorizing energy without any current and magnetic field; and \( I_{c0} \) represents critical switching current at 0K. Both \( \Delta_0 \) and \( I_{c0} \) are proportional to MTJ area, so they tend to become smaller with technology scaling. We selected the parameters, like \( t, \Delta_0 \), and \( I_{c0} \), from a previous well-known STT-MRAM model [27, 33], which has been validated again existing STT-MRAM industrial products.

At 32nm, the single bit read disturbance error (RDE) rate for a conventional sense amplifier with 20μA read current is 3.38E−7 [33]. Since the LPDDR3 STT-MRAM has only 512 bits page size [32], the page error rate is 1.73E−4. Even with a Four Errors Correctable Five Errors Detectable (4E5D) BCH ECC, the page error rate is still larger than 1.71E−4, which is much higher than an acceptable page level main memory error rate 1E−6 [25].

C. Destructive Read

To overcome RDEs, some commercialized STT-MRAM chips [31] adopted destructive sensing (read) scheme with restore, i.e., performing a restore operation after each conventional read. Since the data sensed by and locked in the conventional sense amplifiers remains correct and is not influenced by RDE, so writing the data back recovers the disturbed content in cells. This sensing scheme is referred to as High Current
**Restore Required** (HCRR) read. The same method has been widely adopted in traditional DRAMs. The read in DRAM reduces the charge in the cells belonging to a row. During a read, after reading and latching the data, the sense amplifiers rewrite the data in the accessed row before sending data to output [16].

### D. Non-destructive Read

To avoid trigger RDEs, the read current should be kept low enough. However, it is difficult for a conventional sense amplifier [20, 14] to sense correct data with a low amplitude current ($\leq 20 \mu A$). Instead of applying a large current during reads and performing restores, low current sense amplifiers enlarge the sensing margin and reduce RDEs by prolonging the sensing time. Besides data sensing, low current sense amplifiers require extra sensing stages to compare bit line voltages to multiple reference voltages, so that the sensing margin degradation caused by the variations in STT-MRAM cells can be minimized. Without destroying the data inside each cell, the low current sense amplifier [14] reads the data with $\sim 10 \mu A$ current and $3 \times$ the sensing latency of the conventional sense amplifier [33]. We call this method **Low Current Long Latency** (LCLL) read.

### E. Motivation

To read data out of deep sub-micrometer STT-MRAM arrays correctly, the main memory controller can adopt both High Current Restore Required (HCRR) reads and Low Current Long Latency (LCLL) reads. Although HCRR reads in deep sub-micrometer STT-MRAM arrays maintain the same read latency as that of STT-MRAMs fabricated by old process technologies, the restore after each sensing operation still increases the bank busy time. And thus, the following reads may be blocked by these extra restores. On the contrary, LCLL reads simply prolong the latency for each sensing operation, which is always on the critical path and has direct impact on the entire computing system performance. Both read schemes for deep sub-micrometer STT-MRAMs degrade the main memory performance. And neither of them can always achieve the best performance for different types of applications.

#### Fig. 2. Motivation Example (RT: restore; H-R: high current read; L-R: low current read; HCRR: H-R + RT; LCLL: L-R: The latency of each type of operation can be viewed in Table I).

Figure 2 shows one motivation example to describe why one single read scheme cannot always have the best performance for all applications, and an adaptive read method switching between two read schemes is a must. In this example, one bank needs to process three reads $R_0$, $R_1$, and $R_2$. The data can be returned to CPU after each high current (H-R) or low current (H-R) read. One restore (RT) has to be performed after each H-R to correct the disturbed data. If the arrival interval between reads is long enough indicating the bank is relatively idle. As Figure 2(a) shows, the HCRR scheme finishes three reads much faster. However, in Figure 2(b), if the arrival interval between reads is short in a busy bank, the LCLL technique costs less time to complete three reads. Although the LCLL read spends more time in serving $R_0$ than the HCRR does, it finishes $R_1$ and $R_2$ much earlier. So, overall, the LCLL scheme still has better main memory system performance.

### III. PROPOSED TECHNIQUES

In this section, we propose our techniques to mitigate the performance degradation caused by the state-of-the-art read schemes, LCLL and HCRR reads, in deep sub-micrometer STT-MRAM arrays. We first propose our Smash Read to accelerate the existing HCRR read. And then, we present Flexible Read to dynamically switch between Smash Reads and LCLL reads with the guidance of the bank busy time information.

#### A. Smash Read

The HCRR reads degrade main memory bandwidth, because extra restore operations increase the bank busy time and may block the following reads. To reduce the negative impact of the restore operation, we propose Smash Read (S-RD) to directly shorten the latency of read (sensing) operation.

The conventional read operation for a target STT-MRAM cell issues a low amplitude current through the MTJ discharging the bit-line by a fixed differential voltage ($\Delta V_{BL}$). By measuring the differential voltage, the sense amplifier distinguishes between ‘0’ and ‘1’ at the end of a read operation. The relation between read current and read latency can be described by Equation 3 [22]:

$$T_{RD} = \frac{C_{BL} \cdot \Delta V_{BL}}{I_{RD}}$$

(3)

where $T_{RD}$ indicates the read latency; $C_{BL}$ denotes the capacitance of bit line; $\Delta V_{BL}$ represents the bit line discharged differential voltage, which is also the resolution of a sense amplifier; and $I_{RD}$ is the read current. From this equation, we can see that, for one sense amplifier, the read latency is reversely proportional to the read current. Therefore, by enlarging read current, the read latency can be reduced correspondingly.

However, one obvious disadvantage of a larger read current is that more read disturbance errors (RDEs) emerge. This is

---

*S3 Our first technique. We elaborate this technique in Section III-A.
because, as Equation 2 shows, a larger amplitude current incurs a larger read disturbance probability. Fortunately, the restore operation, which has been already adopted by a HCRR read, can cover and correct these emerging RDEs. Therefore, in our S-RD, there is also a restore.

![Graph showing read latency vs. current](image)

**Fig. 4. Read latency vs. current.**

Due to the small difference between two resistance states, the sensing latency dominates the read latency of STT-MRAMs. We explored the relation between read current and read latency in Figure 4 through NVsim [6], which considers not only sensing latency, but also pre-/decoding delay and word/bit-line latencies. We adopted a baseline configuration including a conventional sense amplifier with 20μA read current and a write driver for a 32nm STT-MRAM MTJ with 30μA write current from [33]. The read operation of MTJ cells is performed during the row activation of the LPDDR3-533MHz STT-MRAM interface, whose latency is 13 cycles [32]. We assume the largest current that our baseline STT-MRAM chip can supply for one cell should be not larger than the write current amplitude. By boosting read current (20μA) to the write current amplitude (30μA), the read latency can be shortened to 9 cycles.

**B. Flexible Read**

Although a larger read current accelerates the read operation, S-RD with a shorter read latency is still encumbered by the restore operation. As Figure 3 shows, on average, S-RD degrades the system performance by 13% over Ideal case and cannot outperform LCLL for some benchmarks. Therefore, we propose Flexible Read (F-RD) to dynamically issue S-RD and LCLL reads to maximize system performance for different benchmarks.

**Fig. 5. The Flexible Read Implementation.**

Figure 5 shows the implementation of our F-RD. Our baseline memory controller (MC) [32] buffers read and write requests which are issued into the main memory system as a physical command queue. Since the MC adopts a FR-FCFS scheduling policy, it maximizes main memory bandwidth by prioritizing requests that hit in the row buffers of STT-MRAM banks over other requests, including older ones. If no request hits in row buffers, the FR-FCFS issues older requests first to memory. And thus, the MC classifies requests by banks, packages them into virtual bank queues and reorders requests within one virtual bank queue to maximize row buffer hit rate.

We add a F-RD scheduler to support our scheme. By monitoring the occupied entry number in each virtual bank queue, the F-RD scheduler estimates whether one particular bank is busy or idle and decides what type of read should be issued into the bank. As Figure 6(a) shows, if the occupied entry number in one virtual bank queue is smaller than a preset F-RD threshold (TH), the F-RD scheduler issues Smash Reads. Otherwise, LCLL reads are scheduled. The F-RD scheduler attaches one bit to each entry in virtual bank queues to distinguish one entry storing a F-RD request or a LCLL request, if the entry stores a read request.

![Threshold Control](image)

**Fig. 6. The Flexible Read Policy.**

in one virtual bank queue is smaller than a preset F-RD threshold (TH), the F-RD scheduler issues Smash Reads. Otherwise, LCLL reads are scheduled. The F-RD scheduler attaches one bit to each entry in virtual bank queues to distinguish one entry storing a F-RD request or a LCLL request, if the entry stores a read request.

**Fig. 7. Request Distribution in all banks.**

Figure 7 shows the request distributions of *astar* and *sjeng* in all banks. From this figure, we can see that the requests from CPU cores do not evenly distribute among all banks. Some banks receive more requests than others, because of the unbalanced main memory address mapping. We adopted our baseline main memory address mapping from [12]. The unbalanced request distribution among banks is also observed by [12, 5]. Therefore, THs used in different banks should be different to reflect the relative fullness and idleness of each bank.

The TH management can be viewed in Figure 6(b). First, we set the initial value of TH in each bank as a half of the bank queue entry number. If the bank queue gets full, TH decreases by one. If the bank queue turns to be empty, TH increases by one. In this way, the F-RD scheduler adapts to the bank occupation changes. After each 10 million cycles, TH is reset to a half of the bank queue entry number, so that our F-RD policy does not stay stale across different execution phases in one application.

**C. Design Overhead**

**Smash Read:** We assume our baseline adopts LCLL sense amplifiers [14]. The HCRR sense amplifier is a part of the LCLL sense amplifier [14], so our baseline can also perform HCRR reads. S-RD does not require any additional hardware. Although S-RD increases the read current to the same amplitude of the write current, S-RD also spends a shorter latency in sensing. By NVsim [6], we estimated S-RD increases read energy by ~ 50% over HCRR. However, F-RD tries to issue as less S-RD reads as possible to maximize the main memory throughput. On average, our proposed techniques only increases activation (read) energy by 2%. The detailed result can be viewed in Section V.

**Flexible Read:** F-RD needs three modifications. First, we added one bit in each read/write queue entry to indicate whether this entry stores a S-RD request or a LCLL request. Since the read/write queue includes 64 entries (each of which is 72 bytes including data, address and other auxiliary informa-
ARMv7 cores, 2GHz, out-of-order shared, 8MB, 16-way LRU, 64B line, write back configured as [32].

channel with 8 banks.

controller adopted close page with FR-FCFS scheduling policy [24]. We modeled a main memory system with one 2GB channel with 8 banks.

<table>
<thead>
<tr>
<th>TABLE I</th>
<th>BASELINE CONFIGURATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>4 ARMv7 cores, 2GHz, out-of-order</td>
</tr>
<tr>
<td>SRAM L1</td>
<td>private, I/D separate, 32KB/core, 64B line</td>
</tr>
<tr>
<td>SRAM L2</td>
<td>shared, 8MB, 16-way LRU, 64B line, write back</td>
</tr>
<tr>
<td>Mem. Ctrl</td>
<td>on-chip, 64-entry R/W queues, close-page, FR-FCFS</td>
</tr>
<tr>
<td>LPDDR3</td>
<td>1 channel, 1 rank-per-channel, 8 banks-per-rank.</td>
</tr>
<tr>
<td>STT-MRAM based Main Memory</td>
<td>Ideal: the LPDDR3 timing and current values are configured as [32].</td>
</tr>
<tr>
<td></td>
<td>HCCR: tRCD 13, tRC 34, IDD0(1.2V) 61.44mA</td>
</tr>
<tr>
<td></td>
<td>LCLL: tRCD 16, tRC 21, IDD0(1.2V) 52.13mA</td>
</tr>
<tr>
<td></td>
<td>S-RD: tRCD 9, tRC 30, IDD0(1.2V) 62.78mA</td>
</tr>
</tbody>
</table>

We adopted the LPDDR3 STT-MRAM timing and current configuration from [32] in Ideal scheme, which has HCCR read latency but no restore. Through NVSim [6], we calculated the timing and current values of HCCR, LCLL and S-RD on the basis of Ideal configuration. The significant differences between these scheme configurations are in the activation part including row address to column address delay (tRCD), row cycle time (tRC) and activation current (IDD0(1.2V)). More details of our baseline and other scheme parameters can be found in Table I.

Simulated Workloads: We chose a subset of simulation benchmarks from SPEC CPU2006 [28], Bio-Bench [1] and STREAM [18] to construct our multi-programmed workloads. We adopted performance metric speedup as Speedup = IPCtech/IPCbase from [13], where IPCtech and IPCbase are the IPCs of the setting with scheme tech and the baseline setting, respectively. Each core ran one copy of simulated benchmark. Each simulation warmed memory hierarchy up by 500M instructions and lasted 1 billion instructions.

V. Evaluation

Schemes: we implemented and compared the following schemes in this section:

- HCCR: high current restore required read with short latency and restore recovering correct data.
- LCLL: low current long latency read conducting three stages sensing to eliminate read disturbance errors.
- S-RD: Smash Read issuing a larger current with a shorter read duration. And a restore must be performed after each sensing to guarantee read disturbance error free.
- F-RD: Flexible Read adaptively switching between S-RD and LCLL to maximize main memory bandwidth.
- Ideal: HCRR without restore or read disturbance errors.

This an unrealistic read scheme.

Performance: Figure 8 shows the performance improvement of the STT-MRAM main memory with each proposed technique. Each scheme is normalized to the Ideal case, which maintains the same latency as HCRR but does not trigger any read disturbance error or perform a restore. Ideal is an unrealistic scheme, since read disturbance errors are unavoidable in deep sub-micrometer STT-MRAM arrays. In order to guarantee good reliability, either restore or LCLL read must be adopted. Significant performance degradation is caused by these read disturbance error correction methods. On average, compared to Ideal, HCRR has 16.6% IPC degradation because of the extra restore; while LCLL degrades system performance by 13.3% due to the longer read latency.

By issuing larger read current, our S-RD reduces read latency but still pays a restore to re-write sensed data back into STT-MRAM cells to maintain data correctness. On average, S-RD improves system performance by 4.3% and 0.4% over HCRR and LCLL, respectively. S-RD decreases only tRCD, but still has a larger tRC than LCLL. Therefore, S-RD cannot have a better performance than LCLL for all applications. Through dynamically issuing both S-RD reads and LCLL reads, our F-RD boosts system performance by 13.3% and 8.9% over HCRR and LCLL, respectively.

Energy: Figure 9 exhibits the STT-MRAM main memory energy (including peripheral circuit/PHY, activation/preharge, and burst energy) saving of each scheme. Compared to Ideal scheme, HCRR increases activation energy by 24.1% through performing extra restores. Because of its worse performance, HCRR also enlarges peripheral interface energy by 19.8%. Overall, HCRR adds additional 15.6% total main memory energy. LCLL raises activation energy by 12.9%, since it requires two more stage sensing operations. LCLL also prolongs application execution time, so it increases peripheral circuits energy by 15.2%. In total, LCLL boosts main memory energy by 10.9%.
Although, compared to HCRR, our S-RD applies a larger current during a read operation, the read latency is shortened by 31%. And thus, S-RD has a similar activation energy as HCRR (+2%). However, S-RD reduces 5% peripheral interface energy by finishing the application executions faster. On average, S-RD diminishes main memory energy by 3%. By adaptively selecting LLC/LL reads and Smash Reads, F-RD further reduces peripheral interface energy by 11% and 9% over HCRR and LLC. Overall, F-RD reduces main memory energy by 8% and 4% over HCRR and LLC.

F-RD Threshold: We tried different static threshold values for F-RD to decide when to issue S-RD or LLC. So-F-RD represents that the threshold value for F-RD is fixed to n. On average, S5-F-RD has the best main memory performance and improves main memory performance by 2.3% over LLC. Compared to S5-F-RD, both larger and smaller threshold values degrade main memory performance from 2.2% to 4.1%.

![Fig. 10. Static Threshold for F-RD (Normalized to Ideal).](image)

Through adjusting threshold value for each bank at run-time, F-RD dynamically decides when to switch between issuing S-RD and performing LLC/LL, according to the fullness and ideleness of every bank. F-RD improves main memory system performance by 6.1% over the best static scheme, i.e., S5-F-RD. Due to the unbalanced request distribution to different banks with the state of the art address mapping [12], the dynamic calibration on the F-RD threshold in each bank is critical for achieving better performance.

VI. RELATED WORK

STT-MRAM faces various reliability challenges including read and write failures [17], retention failures [26], and read disturbance [30]. Among these reliability issues, read disturbance becomes more significant in deep sub-micrometer STT-MRAM arrays [31]. Either HCRR [31] or LLC [20, 14] attacks read disturbance errors without hurting memory performance or increasing energy. Based on the error tolerance of one application, the dual-mode cache architecture [30] statically decides whether a high accuracy mode (HCRR) read or a low power mode (HCRR without restore) should be performed. Our work adaptively issues Smash Reads and HCRR to maximize the main memory bandwidth. To reduce the restore energy overhead, selective restore [33] does not restore right after a high accuracy mode read in the L2 cache, but restores the line until it is evicted from the L1 cache, if it is clean.

VII. CONCLUSION

With fast write current scaling, the read disturbance has become an inevitable reliability issue for STT-MRAM. Neither HCRR reads or LLC reads can always achieve the best performance for STT-MRAM based main memory. In this paper, we propose Smash Read to accelerate HCRR reads by issuing a larger read current. We further improve the STT-MRAM based main memory performance by Flexible Read, an adaptive read technique to switch between Smash Reads and LLC/LL reads. Our experimental results show that our Flexible Read gains the best performance in a LPDDR3 STT-MRAM based main memory system for a wide variety of applications.

REFERENCES