SMART: STT-MRAM Architecture for Smart Activation and Sensing

Byoungchan Oh
University of Michigan
Ann Arbor, Michigan
bcoh@umich.edu

Nilmini Abeyratne
University of Michigan
Ann Arbor, Michigan
sabeyrat@umich.edu

Nam Sung Kim*
University of Illinois at Urbana-Champaign
Champaign, Illinois
nskim@illinois.edu

Ronald G. Dreslinski
University of Michigan
Ann Arbor, Michigan
rdreslin@umich.edu

Trevor Mudge
University of Michigan
Ann Arbor, Michigan
tnm@umich.edu

ABSTRACT

STT-MRAM is a promising drop-in replacement for DRAM for main memory, because it can offer higher energy efficiency than DRAM with comparable latency. Implementing STT-MRAM similar to traditional DRAM memory, that is keeping the same policies while only replacing the technology, however, severely limits the goodness that this new technology can offer. STT-MRAM needs to employ current sense amps which require an order of magnitude more space and power than typical DRAM voltage sense amps. To manage the high cost of sense amps, STT-MRAM decouples bit-lines from sense amps and shares one sense amp across 16 to 128 bit-lines, exploiting the non-destructive nature of its read operation. This sense amp sharing reduces the size of row buffers and, as a result, incur more row-buffer misses (*i.e.*, higher activation energy and lower performance). Other issues arise if STT-MRAM is required to be compatible with current DRAM interfaces and policies.

To cost-effectively address these issues, we propose SMART, which, unlike DRAM and conventional STT-MRAM, proposes sensing bit-lines after receiving a column access command instead of a row activation command. This results in several benefits: larger pages, fewer sense amps, lower activation power, higher bank-level parallelism, shorter latency, fewer address pins, and more efficient repairing of defective columns than conventional STT-MRAM. Our evaluation shows that SMART consumes 11% (39%) lower energy while providing 9% (5%) higher performance than conventional STT-MRAM (DRAM) on average. In addition to these benefits, SMART is 6% smaller than conventional STT-MRAM.

*This work was done while the author was in University of Illinois at Urbana-Champaign. His current affiliation is Samsung Electronics.

MEMSYS '19, September 30-October 3, 2019, Washington, DC, USA

ACM ISBN 978-1-4503-7206-0/19/09…$15.00
https://doi.org/10.1145/3357526.3357529

CCS CONCEPTS

• **Hardware → Memory and dense storage**.

KEYWORDS

STT-MRAM, DRAM, DDR, Non-Volatile Memory

ACM Reference Format:
Byoungchan Oh, Nilmini Abeyratne, Nam Sung Kim, Ronald G. Dreslinski, and Trevor Mudge. 2019. SMART: STT-MRAM Architecture for Smart Activation and Sensing. In *Proceedings of the International Symposium on Memory Systems (MEMSYS '19), September 30-October 3, 2019, Washington, DC, USA.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3357526.3357529

1 INTRODUCTION

STT-MRAM is emerging as a promising drop-in replacement for DRAM because of its faster speed and higher endurance than other non-volatile memory (NVM) technologies [2, 8, 28, 35, 48, 50, 56]. However, STT-MRAM has some disadvantages compared to DRAM. One such disadvantage is the need to use current-sensing sense amps that takes up an order of magnitude more space and consumes higher power than voltage-sensing sense amps that DRAM uses. To offset the cost of implementing current sense amps, STT-MRAM leverages the non-destructive nature of its read operation to place fewer sense amps than the number of bit-lines: sharing one sense amp with every 16 to 128 bit-lines in each bank [5, 6, 24, 29, 59, 61]. This trick suffers from two limitations.

First, fewer sense amps leads to smaller row buffers (i.e. small page size) since the number of sense amps determines the size of row buffers [17]. Large pages provide higher performance with more row-buffer hits when data locality is high, but consume more energy when data locality is poor (known as the overfetching problem). Small pages, in contrast, consume less energy when locality is poor, but give lower performance with more row-buffer misses when locality is good. STT-MRAM row buffers are a lot smaller than DRAM row buffers, but still larger than a column access. Therefore, STT-MRAM suffers more row-buffer misses than DRAM without completely eliminating the overfetching problem.

Second, when such STT-MRAM attempts to adhere to policies designed for DRAM, it suffers from a column address fragmentation problem [48, 61]. Specifically, DRAM requires 16,384 sense amps to sense all the bit-lines of a 2KB page and latch the data until the next row is opened. If not, the data will be lost due to the destructive nature of DRAM's read operation. Therefore, DRAM performs both activation and sensing at the same time with a single row activation (`ACT`) command. Applying the same activation/sensing method to STT-MRAM would require sending both a row address and a partial column address at the same time in order to select a subset of bit-lines to connect to the handful of available sense amps through multiplexers. That is, STT-MRAM with the same capacity as DRAM requires more address pins to send not only a row address but also some part of a column address together with the row command[1]. Besides, such a fragmentation of a column address between row and column commands considerably worsens efficiency and flexibility of column repair mechanisms as a row or column command has only partial column address information [25, 63].

To address these shortcomings, we propose SMART, a new cost-effective STT-MRAM, that implements an activation and sensing policy that builds on the strengths of STT-MRAM . Specifically, exploiting non-destructive reads in STT-MRAM, we propose to provide only two sets of 64 sense amps[2] per bank and make a read command instead of a row activation command sense bit-lines. This deceptively simple change, which is not possible for DRAM because of the destructive nature of its read operation, offers the following advantages over conventional designs.

(1) SMART offers the illusion of providing 16× larger row buffers with 8× fewer sense amps than conventional STT-MRAM. That is, SMART gives you access to a 2KB page with only 128 sense amps per bank, whereas conventional STT-MRAM gives you access to only 128B pages with 1,024 sense amps per bank. Furthermore, SMART saves latency on subsequent column accesses to an open row. Conventional STT-MRAM repeatedly results in a longer delay (`tRC` = $27.5ns$) and higher power to access different columns in the same row which were not selected and sensed by the previous row command. This is because conventional STT-MRAM sees such memory accesses as row-buffer misses and needs a new row command to select and sense different bit-lines. In contrast, SMART recognizes such memory accesses as row-buffer hits and only needs another column access command to select and sense different bit-lines which were already connected to the necessary cells by the previous row activation command.

(2) SMART consumes ∼88% less activation power than conventional STT-MRAM. Specifically, conventional STT-MRAM senses 1,024 bit-lines as part of a row activation, and sensing power dominates the activation power. In contrast, SMART consumes 16× less sensing power than conventional design because it senses only 64 bit-lines (*i.e.*, granularity of a column access) as part of a column access. Hence, SMART practically eliminates the overfetching problem. Furthermore, whenever a memory write access demands an activation of another row, conventional STT-MRAM unnecessarily consumes sensing power because sensing is part of the row activation. However, SMART does not consume any sensing power for such a memory write access, because sensing is not part of a row activation but part of a column read access.

(3) SMART offers shorter latency for memory accesses. Specifically, high sensing power imposes `tRRD` and `tFAW` constraints in DRAM and STT-MRAM and limits the number of row activation commands in a certain time period. As SMART significantly reduces sensing power, it can eliminate these two constraints and handle more row activation commands in a shorter time period than conventional designs. As previously mentioned, conventional STT-MRAM performs sensing as part of row activation while sensing is unnecessary for memory write accesses. Moreover, sensing constitutes a notable fraction of activation time. Therefore, SMART can also reduce latency of memory write accesses especially to different rows compared with conventional STT-MRAM.

(4) SMART needs fewer pins and offer 10.7× more efficient repairs of defective columns than conventional STT-MRAM. In particular, SMART does not select, connect, or sense specific bit-lines as part of row activation. That is, it needs to receive only a row address for activation like DRAM. Additionally, because SMART receives the full column address with a column command, it can use the same efficient mechanism as DRAM for repairing defective bit-lines.

(5) SMART eliminates the need for sending separate precharge commands because SMART can overlap closing an already open row and opening a new row during the row activation. As a result, SMART not only offers 11% shorter latency for memory accesses which are directed to the same bank but incur row-buffer misses, but also consumes less command bus bandwidth than conventional STT-MRAM.

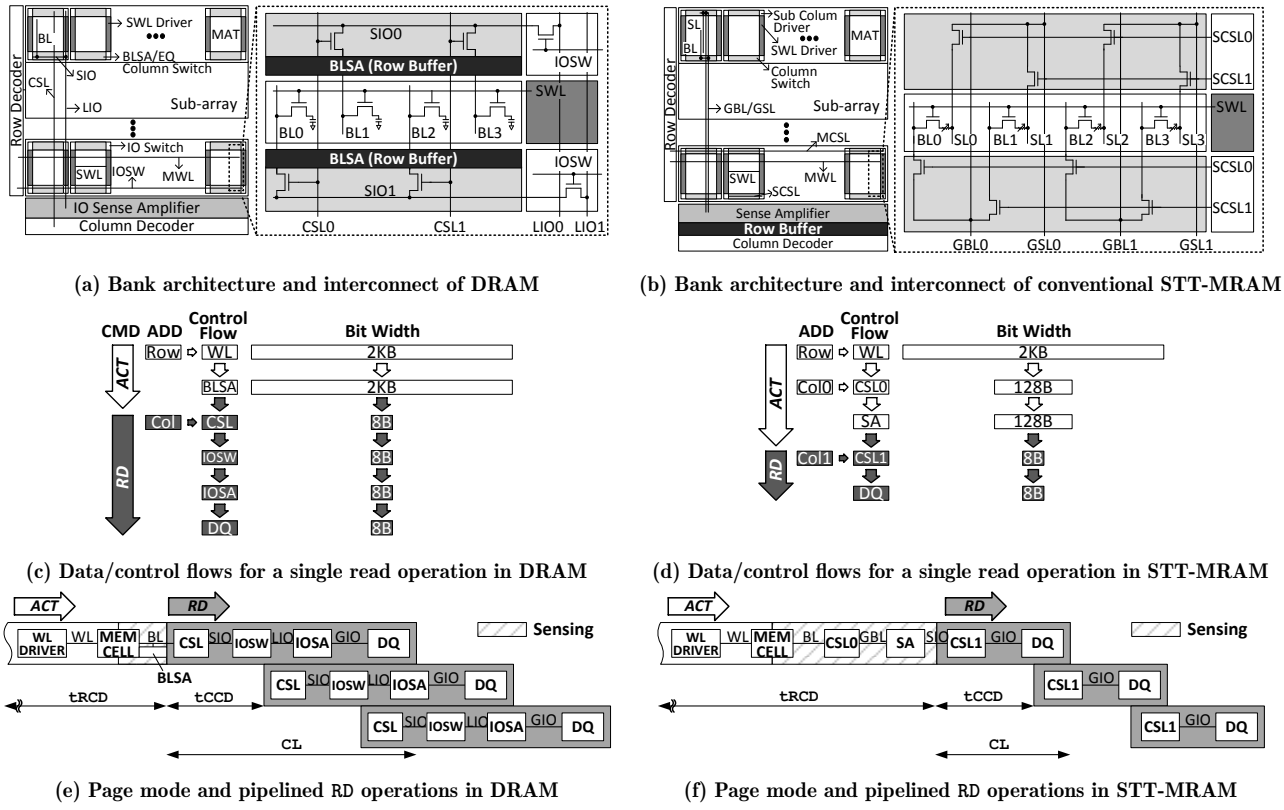
In summary, (1)–(5) reduce energy and improve performance. Our evaluation shows that SMART consumes 11% and 39% lower energy while providing 9% and 5% higher performance than conventional STT-MRAM and DRAM on average, respectively. In addition to these benefits, SMART is 6% smaller than conventional STT-MRAM.

[1]DRAM uses the same set of address pins to receive both row and columns addresses in a time multiplexed manner. Since a row address typically needs more bits than a column address, the number of row address bits determines the number of address pins in DRAM.

[2]In this paper, we assume DRAM and STT-MRAM devices, each consisting of eight ×8 devices for 64-bit I/O. Therefore, each device must sense 64 bits to support the burst length of 8 for a single column access command. In addition, the page size in this paper means a page size per chip (*i.e.*,2KB), but not per rank (*i.e.*,16KB).

2 CHALLENGES IN ARCHITECTING STT-MRAM

In this section, we compare STT-MRAM with DRAM and identify challenges and limitations in conventional STT-MRAM.



(a) Bank architecture and interconnect of DRAM

(b) Bank architecture and interconnect of conventional STT-MRAM

(c) Data/control flows for a single read operation in DRAM

(d) Data/control flows for a single read operation in STT-MRAM

(e) Page mode and pipelined RD operations in DRAM

(f) Page mode and pipelined RD operations in STT-MRAM

Figure 1: Bank architecture and operation of DRAM (left) and conventional STT-MRAM (right).

2.1 Large Sense Amps with High Power Consumption

STT-MRAM is expected to offer cell read/write speed comparable to DRAM and good endurance ($> 10^{15}$) [23, 27, 44, 58, 62]. A recent study demonstrated 4Gb LPDDR2-compatible STT-MRAM with $9F^2$ cell size and sub-50ns read/write speed [48]. Such characteristics make STT-MRAM a promising alternative to $\sim 7F^2$ DRAM. However, STT-MRAM poses unique challenges especially in implementing sense amps (SA) [3, 4, 22, 24, 39, 54].

First, STT-MRAM need a current SA with a reference current generator because sensing small difference in on/off resistance is challenging, which is further worsened by process variations. This requires STT-MRAM to adopt a very complex current SA that is an order of magnitude larger than a voltage SA employed by DRAM.

Second, sensing in STT-MRAM consumes high power. DRAM SAs simply charge/discharge bit-lines (BLs) once per sensing, whereas STT-MRAM SAs need to continuously flow current to BLs until they reach a level sufficient for sensing. To eliminate the power consumed by the continuous current flow, separate buffers are implemented and data in SAs are copied to the buffers so that the SAs can be turned off immediately after sensing bit-lines [29, 54].

2.2 Limitations with Shared Sense Amps

DRAM couples every BL with a dedicated SA in each sub-array (Fig. 1a) due to the destructive nature of its read operation. On the other hand, STT-MRAM decouples SAs from BLs with multiplexers and shares one SA with 16 to 128 BLs [5, 24, 48, 61], as depicted in Fig. 1b. This exploits the non-destructive nature of its read operation to manage the cost of SAs.

DRAM in page mode senses cell states of a row and stores them into a row buffer[3] when `ACT` is received. This allows DRAM to precharge/activate a row once for multiple column accesses, as depicted in Fig. 1c. Such an operating mode has not changed in modern DRAM [10, 18, 20, 21]. Fig. 1d describes serving a read request in STT-MRAM following the same page mode as DRAM but sharing one SA with 16 BLs to reduce the cost. An `ACT` command first asserts a word-line (WL) connected to 16,384 cells (2KB). Since there are only 1,024 SAs (*i.e.*, 128B row buffer), a column selection signal (CSL0) uses part of the column address (Col0) to select one BL out of 16 BLs. The column selection signal (MCSL-SCSL) in an STT-MRAM bank shown in Fig. 1b corresponds to the WL selection signal (MWL-SWL). The remaining part of the column address (Col1) provided in the column access

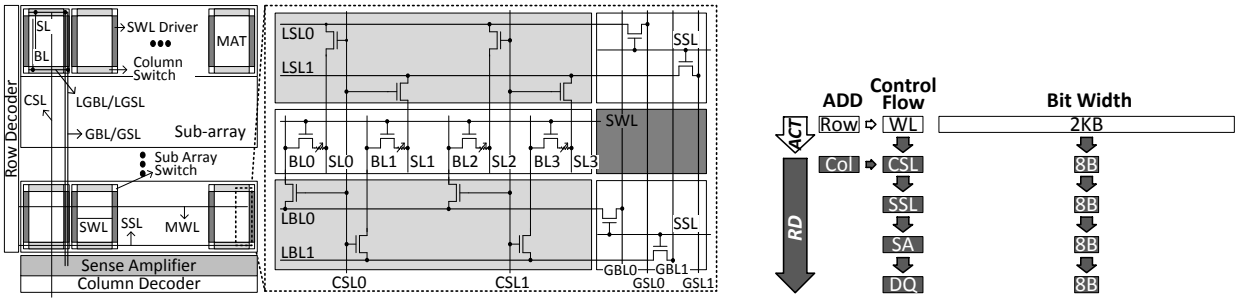
[3]The SAs shared by two adjacent sub-arrays serve as a row buffer.

command (RD or WR) selects global bit-lines (GBLs). Such STT-MRAM, however, suffers from the following limitations.
Limitation 1: Longer latency. DRAM places bit-line sense amps (BLSAs) [28, 48] above and below a sub-array because the charge sharing limits the BL length in DRAM [26, 38]. Because the local I/O (LIO) lines are too long to be driven by small BLSAs, larger I/O sense amps (IOSAs) are placed near the column decoder of each bank to assist the transfer of data through the LIO lines. In contrast, STT-MRAM does not require BLSAs [28, 48], placing only column multiplexers above and below a sub-array with one set of SAs at the bottom of a bank consisting of 128 sub-arrays. The column multiplexers connect one BL in a group of BLs to a SA through a GBL, which makes the sensing path of STT-MRAM longer that that of DRAM. That is, the sensing path of STT-MRAM is the height of a bank while that of DRAM is the height of a sub-array. This in turn increases tRCD, minimum time from ACT to RD or WR while decreasing CL as shown in Fig. 1f [61].
Limitation 2: Smaller pages. Consider STT-MRAM sharing one SA with N BLs where N is 16∼128 in prior work [1, 24, 48]. When STT-MRAM has 16,384 cells connected to a WL like DRAM, the page size of such STT-MRAM becomes $1/N$ of that of DRAM. Page-size-sensitivity analyses in our background research indicated that streaming benchmarks with sequential accesses can see IPC degradation as high as 35% for a page size of 1/16 of 2KB. Although small pages reduce power, larger pages are better in most cases when considering performance [9, 55, 60, 65]. With fewer SAs than BLs (*i.e.*, cells in a row), STT-MRAM encounters the following three cases: (1) an access to another row (*i.e.*, row miss); (2) an access to the same row with BLs selected and sensed by previous ACT (*i.e.*, row hit); or (3) an access to the same row but BLs which are not selected and sensed by previous ACT yet (*i.e.*, row hit but row buffer miss). The third case needs to be handled like a row miss because connecting appropriate BLs to SAs and sensing them are coupled with ACT in STT-MRAM sharing one SA with many BLs. That is, some of a column address becomes part of a row address since they are needed before sensing appropriate BLs.

Consequently, selecting different BLs which were not selected by the previous ACT always demands another ACT. Because new activation can be performed only after deselecting the old BLs, the third case (row hit but row buffer miss), also generates a new precharge (PRE) prior to the ACT [19, 29, 48, 61].
Limitation 3: Lower repair efficiency. Splitting a column address between row activation and column access commands, *referred to as column address fragmentation in this paper*, creates two important challenges in managing chip yield and compatibility with existing DDR interfaces. Typically, the minimum repair granularity is a row or a column. Any redundant WL can replace any defective WL in the same bank and any redundant BL can replace any faulty BL in the same mat. This technique is known as any-to-any replacement [64]. In an STT-MRAM architecture that shares one SA with multiple BLs, however, neither ACT nor RD/WR have the full column address information. STT-MRAM uses the partial column address from ACT to select one BL in each of the 64 BL groups in each mat. Therefore, if the partial column address were to replace a defective BL in a BL group, it needs to replace every BL selected by the same partial column address. Consequently, STT-MRAM needs at least one redundant BL for every BL group (*i.e.*, 64 redundant BLs per mat). On the other hand, STT-MRAM uses the partial column address from RD or WR to select 64 BL groups from 1,024 BL groups (*i.e.*, the number of SAs). Hence, if it were to use the partial column address to replace a defective BL in a BL group, it needs to replace every BL in the BL group. That is, STT-MRAM needs at least one redundant BL group in each mat, (*i.e.*, 16 BLs per group). Both cases negatively affect chip yield because they limit the flexibility of replacement and increase the minimum repair granularity.
Limitation 4: Higher pin cost. In DRAM, row and column addresses share the same address pins because they are delivered at different times [18, 21]. The number of address pins are equal to the number of row address bits because there are typically more rows than columns (*e.g.*, 64K rows and 2K columns in ×8-8Gb DRAM). In the shared SA structure, however, more address pins are needed to send both the row address and a partial column address (to assist BL selection) at once. For example, we need 4 more pins for $N = 16$. Therefore, the shared SA architecture is not compatible with conventional DDR interfaces. To solve this problem, comboAS [48, 61] proposes waiting to start row activation until after RD or WR sends a column address; but this always increases CL by the row activation time. LPDDR2-NVM [19, 31] proposes another command, PREACT to deliver a part of the row address before sending ACT so that STT-MRAM can compose a complete row address after receiving the ACT command. This also increases row activation time while consuming more command bus bandwidth.

3 SMART ARCHITECTURE

Due to destructive reads in DRAM, the SAs have to sense every cell in the row after ACT asserts the WL. This requires the number of SAs to be equal to the number of cells in a row. Therefore, the number of cells in a row determines both the page size and the activation power in DRAM. This makes it impossible for DRAM to offer both the high performance of large pages and the low power of small pages at the same time. In contrast, the non-destructive reads in STT-MRAM mean that SAs do not need to sense cell states after ACT asserts the WL. Exploiting such a property, we re-define the page mode operation to sense BLs following the RD command instead of the ACT command. We preserve the original intent of the page mode concept in the sense that whole pages are accessed without repeated activation. In the remainder of this section, we first present the details of the SMART architecture and then discuss the five key benefits of SMART over conventional STT-MRAM and/or DRAM.



(a) Bank architecture (b) Data and control flow

Figure 2: SMART bank architecture and data/control flow.

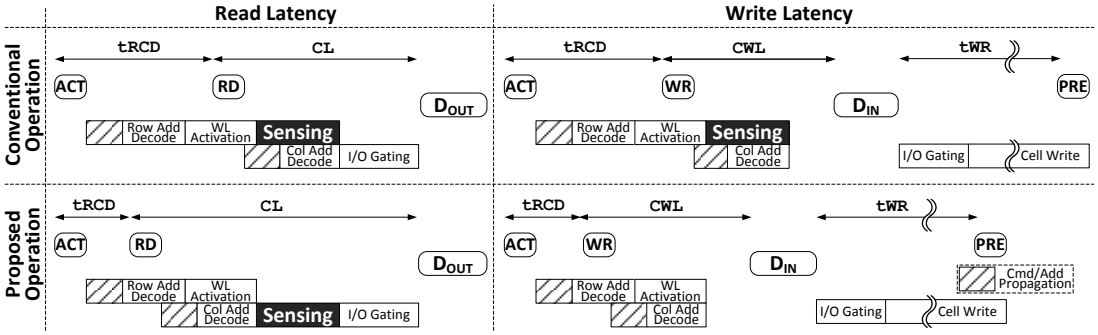


Figure 3: Change of read and write accesses after making sensing operation triggered by a RD command.

3.1 Re-architecting STT-MRAM

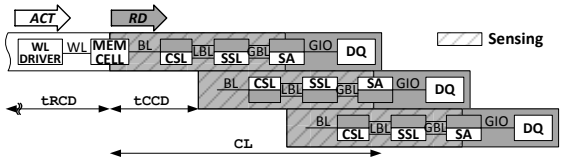
In SMART a given ACT command only asserts the WL. That is, ACT does not sense any BL. The subsequent RD command will sense only the 64 BLs specified by the column address. This order of operations allows SMART to provide any page size by sharing 64 SAs per bank along with other significant benefits which will be discussed in Sec. 3.2.

To efficiently support such ACT and RD commands for SMART, we propose a bank architecture depicted in Fig. 2a.

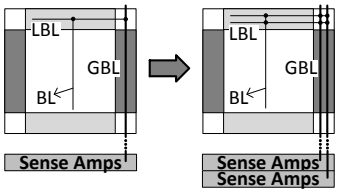
Traditional I/O interconnect of DRAM has a hierarchy as depicted in Fig 1a. The data wiring in SMART is similar to DRAM, but different from the conventional STT-MRAM. In DRAM, the interconnect from a memory cell to an SA is called the BL and the interconnect from a SA (row buffer) to DQ is called IO. For a given column access, 64 BLs/BLSAs in a sub-array are selected by column selection lines (CSLs) and connected to segmented I/O (SIO) lines in a sub-array through column switches (*i.e.*, multiplexers). Then, the SIO lines are connected to Local I/O (LIO) lines running vertically across a bank through I/O switches (IOSWs) [32]. The SIO, LIO and IOSW in DRAM correspond to Local BL (LBL), GBL, and sub-array selection line (SSL), respectively, in SMART. Lastly, 64 SAs are placed at the bottom of each bank where IOSAs (*cf*. Sec. 2.2) are located in DRAM. In summary, SMART has a bank architecture similar to DRAM, but it does not need 16,384 SAs to implement a 2KB page like DRAM and it does not limit itself to implementing only $1/N$ of a page like the conventional shared SA structure.

This SMART bank architecture does not increase latency of memory read accesses, because the total amount of time for performing ACT and RD remains the same as conventional STT-MRAM. As shown in Fig. 3(left), compared with conventional STT-MRAM, SMART simply reduces the amount of time for ACT (tRCD) while increasing the amount of time for RD (CL). On the other hand, as shown in Fig. 3(right), SMART can decrease latency of memory write accesses. Specifically, compared with STT-MRAM, ACT-WR does not consume any time for sensing BLs without affecting CWL, time between the moment at which a WR command is sent and the moment at which its first data (D_{IN}) is placed.

In DRAM, the SIO-LIO lines are electrically isolated from the GIO lines, as shown in Fig. 1a. This allows DRAM to internally pipeline more than two consecutive RD commands which can be issued at every tCCD interval (typically ∼5ns) without waiting for long CL (typically ∼13.75ns), as shown in Fig. 1e. SMART, however, has longer read time than tCCD because RD also senses BLs, as shown in Fig. 4a. To hide such long sensing time for consecutive RD commands, we double the number of LBL and GBL lines, column multiplexers, and SAs, as shown in Fig. 4b. The two sensing paths take turns to serve consecutive RD commands so that SMART can handle one RD command at every tCCD interval, as depicted in Fig. 4a. Sec. 4.1 shows that in spite of doubling the number of sensing



(a) Page mode operation and pipelined RD commands



(b) Enhanced bank architecture

Figure 4: SMART page mode and bank architecture.

paths, SMART decreases the cost of STT-MRAM by ∼6% because the SMART uses 8× fewer SAs than conventional STT-MRAM.

3.2 Benefits

With the bank architecture presented in Sec. 3.1, SMART provides the following notable benefits over conventional STT-MRAM without hurting read performance while improving write performance.

Benefit 1: Larger pages and fewer SAs. SMART's re-defined page mode can give the illusion of larger pages with fewer SAs than the conventional STT-MRAM. Specifically, `ACT` only asserts the WL to connect 16,384 cells to 16,384 BLs. It is a subsequent `RD` command that selects appropriate 64 BLs and sense them. Then, SMART needs only two sets of 64 SAs per bank for 2KB pages whereas conventional STT-MRAM implements 1,024 SAs per bank for 128B pages. Because conventional STT-MRAM implements 16× fewer SAs than BLs, it repeatedly consumes a long `tRC` time for accessing different columns in the same row that were not selected and sensed by the previous `ACT` command. SMART, on the other hand, does not consume `tRC` for such column accesses because it only needs another `RD` command to connect and sense these columns. This significantly reduces the latency of sequential memory accesses. Note that the SAs in conventional STT-MRAM consume ∼9% of STT-MRAM space based on an adapted version of DRAMSpec [40]. That is, SMART can reduce the space and activation power consumed by SAs to 12.5% of conventional STT-MRAM. This is sufficient not only to negate the cost increased by another read path but also to significantly reduce sensing power and memory access latency, as elaborated below.

Benefit 2: Lower activation power with fewer SAs. `ACT` energy is a large portion of total energy in DRAM. Fig. 5 illustrates the relationship between page size and `ACT` energy in DRAM, STT-MRAM and SMART. In a DRAM, WL activation for a 2KB page connects 16,384 cells to 16,384 BLs. Although we consider higher WL voltage (V_{PP}) than the BL voltage (V_{DD}) and low efficiency of the charge pump to generate

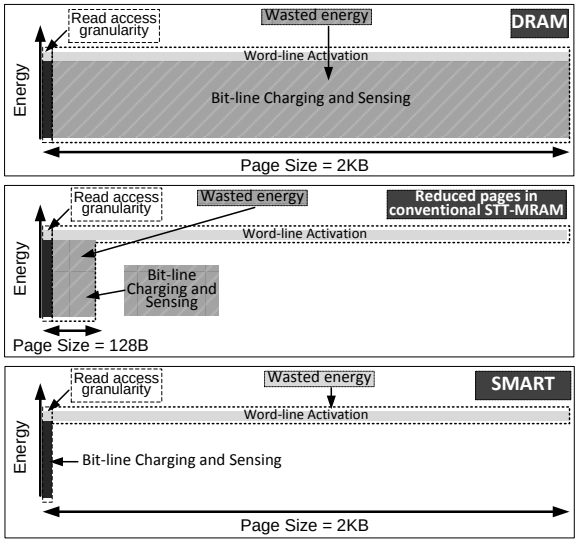


Figure 5: Relation between page size and `ACT` energy.

V_{PP} (∼30% [37]), charging/discharging the BLs dominates the `ACT` energy because of the large number of BLs. Since a `RD` command accesses only ∼0.4% (= 64/16,384) of cells in a row, sensing all 16,384 BLs for only a few `RD` accesses, referred to as the overfetching problem, wastes a significant amount of energy as shown in Fig. 5(top). To reduce `ACT` energy, DRAM architectures with smaller pages and fine-grained activation have been proposed [9, 55, 60, 65]. In conventional STT-MRAM, `ACT` senses only $1/N$ BLs and consumes $1/N$ of activation energy, as depicted in Fig. 5(middle). Nonetheless, it still wastes a considerable amount of energy since an STT-MRAM SA consumes higher power than a DRAM SA. In addition, high `ACT` power limits bank-level parallelism (will be discussed in Benefit 3). In contrast, SMART reduces `ACT` energy while providing larger pages with fine-grained activation at a smaller cost than STT-MRAM.

Benefit 3: Shorter latency with lower activation power. High `ACT` power affects not only total memory energy but also overall memory performance. Specifically, simultaneous activation of multiple cells connected to one row, each charging/discharging 16,384 BLs, draws a large amount of current. This requires some time to recover from the voltage drop of the power delivery network, which is enforced by `tRRD` (RAS to RAS delay) and `tFAW` (four activation window). If there are two read accesses to different banks, the second `ACT` command can be scheduled between the first `ACT` and `RD` commands, but not until `tRRD` has elapsed from the first `ACT` command, as depicted in Fig. 6(top). `tFAW` limits the number of `ACT` commands to four within a `tFAW` time window. Therefore, these constraints limit bank-level parallelism and they are imposed on not only DRAM but also other memory technologies. For example, LPDDR2-NVM also defines `tRRD` and `tFAW` [19].

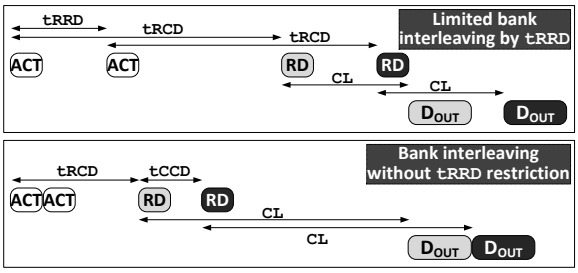
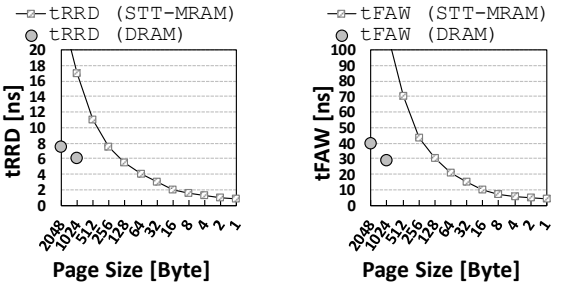
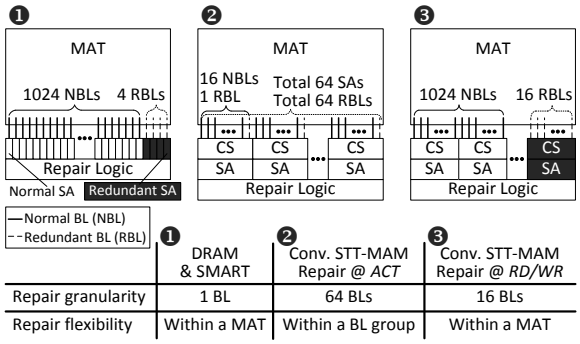


Figure 6: Impact of tRRD on bank-level parallelism.



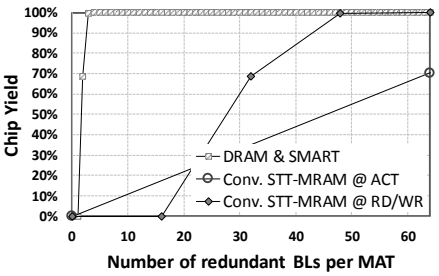
(a) tRRD **(b) tFAW**

Figure 7: tRRD and tFAW for various page sizes. tRRD and tFAW for DRAM are obtained from a DDR3 datasheet [36].



	❶ DRAM & SMART	❷ Conv. STT-MAM Repair @ *ACT*	❸ Conv. STT-MAM Repair @ *RD/WR*
Repair granularity	1 BL	64 BLs	16 BLs
Repair flexibility	Within a MAT	Within a BL group	Within a MAT

(a) Example of various column repair schemes



(b) Chip yield according to repair schemes

Figure 8: Comparison of the three repair schemes.

We plot tRRD and tFAW for various page sizes in Fig. 7, where the maximum activation current values are determined by a method of prior work [30]. tRRD and tFAW of conventional STT-MRAM for 128B pages are $\sim$6.2ns and $\sim$30ns, respectively. Since a current SA consumes far more power than a voltage SA, STT-MRAM needs longer tRRD than DRAM for the same page size. In contrast, SMART activates 16× fewer SAs and consumes less power than conventional STT-MRAM. Note that SMART ACT does not consume any sensing power and the recovery time for activating 64 SAs is short enough compared to tCCD ($\sim$5ns). Hence, SMART can practically eliminate these two constraints and handle multiple ACT–RD commands to different banks back to back. This significantly reduces the latency of memory accesses especially when applications exhibit poor locality where subsequent memory accesses would be mapped to different banks rather than to the same page in the same bank. For example, SMART can serve two ACT–RD commands to two different banks without consuming tRRD, as shown in Fig. 6(bottom).

Lastly, SMART also consumes shorter time and less power than conventional STT-MRAM for memory write accesses. This is because an ACT command of SMART does not consume time and power for sensing which is unnecessary for write accesses, whereas conventional STT-MRAM still does.

Benefit 4: Fewer pins and more efficient repair. Unlike conventional STT-MRAM, SMART does not demand a partial column address with ACT, and uses the same number of address pins as DRAM. That is, SMART does not suffer from column address fragmentation discussed in Sec. 2.2. Column address fragmentation splits a column address into ACT and RD and makes repairing a mat less efficient and flexible for conventional STT-MRAM. This can significantly increase the cost of repairing mats or decrease the yield of STT-MRAM chips. Fig. 8a describes the column repair schemes for DRAM and conventional STT-MRAM. In DRAM (❶), we can replace any 1,024 BLs with any 4 redundant BLs in a mat, and we repair a BL with a RD or WR command comprising a complete column address [15]. As SMART also exposes a complete column address to a RD or WR command, it can adopt the same column repair scheme as DRAM. In conventional STT-MRAM, however, neither a ACT command nor a RD/WR command has a complete column address. This makes repairing BLs far less efficient and flexible than DRAM or SMART.

Consider STT-MRAM with 1,024 BLs per mat and N (= 16) BLs per SA, (*i.e.*, 1,024/N = 64 BL groups). If we are to repair a BL with ACT (❷), we need one redundant BL for every N BLs (= 64 redundant BLs) as ACT can select only one BL in each BL group. On the other hand, if we are to repair a BL with RD or WR (❸), we need to replace the entire BL group including a defective BL with a redundant BL group (= 16 redundant BLs) because RD or WR can select only BL groups.

We analyze the chip yield for various numbers of redundant BLs in Fig. 8b where we assumed that capacity of a chip is 8Gb, the number of mats is 16,384 in a chip where each mat

Table 1: Comparison with previous studies (Conv-Delay and Conv-Pin are the conventional STT-MRAM designs)

	DRAM	LPDDR2-NVM [5, 19]	Conv-Delay [48, 61]	Conv-Pin [29, 35]	This work (SMART)
Page size	Large (2KB)	Small (32B)	Small (128B)	Small (128B)	Large (2KB)
Sensing operation	During `ACT`	During `ACT`	During `ACT`	During `ACT`	During `RD`
Activation energy	High	Medium	Medium	Medium	Extremely low
Bank-level parallelism	Limited by `tRRD`/`tFAW`	Same as DRAM	Same as DRAM	Same as DRAM	No limitation
Pin-compatible with DDR	Yes	Yes (3-phase addressing)	Yes (delayed `ACT`)	No	Yes
Repair flexibility	High	Low	Low	Low	High
`PRE` technique	SALP [26]: Can violate `tRP` to access different sub-array	No	EarlyPA [61]: Internally perform `PRE` after the sensing and set `tRP`=1	No	No `PRE` command and no `tRP`
Need software/OS support	No	Yes, for write operation	No	No	No

Highlighted green: good features, highlighted yellow: good, but limited

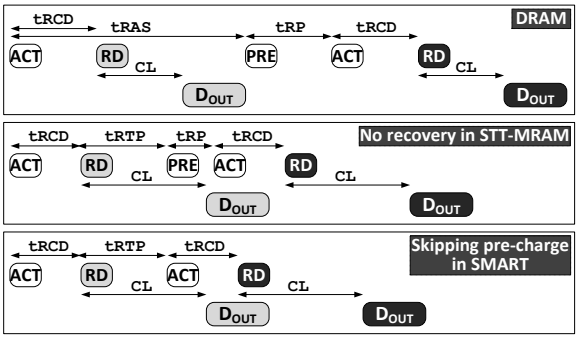


Figure 9: Change of row access cycle on row misses.

has 512 WLs and 1024 BLs, N is 16 for conventional STT-MRAM, memory defects follow a `Poisson` distribution [14, 15], the target bit error rate (BER) after row repair is 10^{-7} and both DRAM/SMART and conventional STT-MRAM use the same row repair scheme. This shows that conventional STT-MRAM has a lower chip yield than DRAM or SMART for the same number of redundant BLs. In other words, to accomplish the same chip yield (99%) under the same BER (10^{-7}), conventional STT-MRAM requires 10.7× more redundant BLs than DRAM and SMART.

Benefit 5: Eliminating precharge commands. To activate another row in the same bank (a row-buffer miss), DRAM needs a `PRE` before an `ACT`. Specifically, `PRE` has two phases: (1) deactivating an asserted WL and (2) initializing BLs before `ACT` senses BLs. (2) also destroys cell states if it is performed before the WL is completely deactivated. Therefore, (1) should complete fully before (2) is started, and the total time for (1) and (2) (`tRP`) increases latency of memory accesses when row-buffer misses occur, as shown in Fig. 9(top). Furthermore, when STT-MRAM follows the same page mode as DRAM, it experiences more row-buffer misses with smaller pages and pays this penalty more frequently than DRAM.

Unlike DRAM, however, STT-MRAM does not need to serialize (1) and (2) because of the non-destructive nature of its cells. Moreover, STT-MRAM immediately transfers the cell states sensed by SAs to registers serving as a row buffer (Sec. 2.1). This allows STT-MRAM to initialize the BLs and SAs immediately after sensing BLs as part of `ACT` and reduce `tRP`. Note that STT-MRAM initializes its BLs by discharging them to V_{SS}. The driving strength of charging BLs is limited to prevent unexpected changes of cell states (read disturbance) [33, 59], but that of discharging BL is not limited. Hence, the amount of time for (2) can be much shorter than DRAM and it is already included in `tRAS` instead of `tRP` [48, 54, 61]. This reduces `tRP` of STT-MRAM by the amount of time for (2). Consequently, as shown in Fig. 9(middle), STT-MRAM can serve the second `RD` command faster than DRAM, but it does not reduce or hide the amount of time for (1), still demanding a separate `PRE` command. In contrast, SMART can overlap the time for (1) (∼3.7ns) with the time to decode a row address and compare the address with addresses in a row repair table [15] during the early phase of `ACT` for the next row (∼4.2ns)[4]. This allows SMART to completely remove `PRE` and not consume any `tRP` before `ACT`, further reducing the latency to handle row-buffer misses. Note that the conventional STT-MRAM could have skipped precharge because immediate BL initialization after sensing and the non-destructive sensing operation are common features. However, prior work [61] only reduced `tRP` without the overlap applied in SMART.

3.3 Discussion

SMART consumes far less space for SAs and redundant BLs than conventional STT-MRAM but more space for another sensing path per bank. Overall, SMART is ∼6% smaller than conventional STT-MRAM (Sec. 4.1). We summarize the key differences among DRAM, LPDDR2-NVM, conventional STT-MRAM and SMART in Table 1

SMART does not increase the latency of serving a single read request or consecutive read requests issued at the `tCCD` interval but it still increases CL. This may increase overall read latency of serving multiple read requests issued at longer intervals than `tCCD`. However, our evaluation shows that the

[4]We leverage the ∼0.5ns difference, but some overlap between deactivating the previous WL and activating the new WL is acceptable because such overlap does not destroy the cell states.

performance degradation caused by the increased `CL` is outweighed by the performance increase by larger pages, higher bank-level parallelism, and lower row-buffer miss latency.

Since a row buffer holds data only for a previously accessed column, SMART cannot compare-before-write when a `WR` command is sent to a different column of the activated row. Comparing data before writing reduces write energy and improves the endurance by not overwriting the same data [43, 48]. However, the high endurance of STT-MRAM cells ($> 10^{15}$) guarantees practically unlimited write operations (Sec. 2.1). Moreover, cell write energy is not a major component in overall write energy. Therefore, such a technique has limited impact on giga-bit scale STT-MRAM (Sec. 4.3).

Unlike conventional shared SA designs [29, 35] that use extra address pins, SMART does not change the physical interface (DDR PHY). Moreover, changes to access timing (*e.g,*`tRCD`) do not require modification in memory controllers because programmable parameters are a built-in feature to support various DRAM's speed grades [57]. Although issuing the next `ACT` without `PRE` is prohibited in the traditional open-page policy, we expect minimal modification in the scheduler can enable skipping precharge for SMART.

Off-chip error correction code (ECC), where extra chips are added to store parity data on memory modules, is not dependent on memory cells; hence, the same ECC can be adopted in SMART. On-chip ECC proposed in LPDDR4 [53], can also be adopted because SMART keeps same access granularity, which determines the code word for ECC, as DRAM. However, placing parity columns and their SAs for on-chip ECC would require more area in STT-MRAM than in DRAM because of bigger memory cells and SAs.

Lastly, as this work focuses on re-architecting STT-MRAM for higher performance and lower energy, we do not discuss challenges related to its cells, such as thermal stability, write endurance, and read disturbance in detail [33, 58]. Nonetheless, recent studies have demonstrated small (sub-20nm) STT-MRAM cells that can offer fast switching time (sub-10ns) under low write current (sub-10uA), high write endurance ($> 10^{15}$), thermal stability and read disturbance [12, 41, 52].

4 DEVICE MODELING

To evaluate DRAM, STT-MRAM and SMART, we take DRAMSpec [40], a detailed timing, power, and area exploration tool which is originally developed for DRAM but can be adapted for other memory technologies such as STT-MRAM. For the baseline DRAM, we consider ×8 8Gb DRAM devices. Then, we adapt some parameters of DRAMSpec to obtain similar results for a baseline DRAM in 30 nm technology. While keeping the same chip floor-plan as DRAM, we adapt the bank architecture and interconnect models to model STT-MRAM and SMART with parameters taken from NVSim [11] and prior work [7, 8, 27, 47, 48] and then extrapolated to $30nm$ technology.

4.1 Area Model

We assume a $7.2F^2$ DRAM cell (2F×3.6F = WL×BL) provided by the DRAMSpec's $30nm$ technology model and a $10.9F^2$ (3F×3.6F) STT-MRAM cell. For conventional STT-MRAM, we take $N = 16$ which is from an industry STT-MRAM chip [48]. Following the JEDEC standard for DRAM, SMART has the same number of rows and columns as DRAM. However, SMART can implement any page size with the number of SAs equal to the number of bits per column access.

In accordance with baseline DRAM [36], conventional STT-MRAM and SMART are modeled as an 8-bank, 8Gb device. Each bank consists of 2048 mats comprised of 512 WLs and 1024 BLs. Because SMART has the same floor-plan and similar bank structure as DRAM, similar approaches to increase the capacity can be applied, such as increasing the number of banks and rows. We also assume 12 redundant WLs and BLs per mat for DRAM and SMART whereas we suppose 32 redundant BLs per mat for conventional STT-MRAM (Sec. 3.2). Prior work [54] demonstrated the layout area of various SAs for STT-MRAM, but it designed the SAs with a logic technology. Thus, we convert transistor sizes and design rules to those of a memory technology based on the ITRS roadmap to re-estimate the area [16].

Table 2: Area comparison

	DRAM	Conventional STT-MRAM	SMART
Cell size	$7.2F^2$	$10.9F^2$	$10.9F^2$
SA size	$1{,}213F^2$	$27{,}111F^2$	$27{,}111F^2$
# of SA per bank	1,048,576	1,024	128
Bank area (μm^2)	2,914×7,512	3,132×8,695	2,980×8,390
Chip area (mm^2)	185.65	224.19 (21% ↑)	211.48 (14% ↑)

We summarize the analyzed area of key memory components in Table 2. The total height of the BLSA blocks in DRAM is ∼20% of the total chip height in a $20nm$ 8Gb DRAM device [45]. Therefore, the number and size of SAs greatly affect the total chip size. SMART has 2× more sensing paths than conventional STT-MRAM, but it consumes ∼6% smaller space because it needs 8× fewer SAs. Albeit SMART uses 1.5× and 21.4× larger cells and SAs than DRAM, it uses 8,192× fewer SAs. This is because a bank has 128 sub-arrays and a pair of two sub-arrays shares 16,384 SA in DRAM. SMART is only 14% larger than DRAM whereas conventional STT-MRAM is 21% larger. Furthermore, the chip area of SMART is less sensitive to memory cell size than that of the conventional designs because of fewer redundant and reference BLs.

4.2 Timing Model

Table. 3 summarizes the timing parameters and read access latency of DRAM, conventional STT-MRAM and SMART based on the memory clock frequency of 800MHz. DRAM gives the shortest latency for a single read access because its sensing speed is faster than that of STT-MRAM. However, the overall latency of multiple read accesses is determined not only by the sensing speed but also by other timing parameters

Table 3: Timing and latency comparison

	DRAM	Conventional STT-MRAM	SMART
tRCD (clock cycle)	11	17 (1)	8
tRAS (cc)	27	18 (19)	9
tWR (cc)	12	19 (19)	19
tRP (cc)	11	4 (4)	4
tRTP (cc)	6	1 (18)	9
tRRD (cc)	6	5 (5)	1
tFAW (cc)	32	24 (24)	4
CL (cc)	11	8 (25)	17
Latency for single RD	22	25 (26)	25
Latency for five RDs (different banks)	54	49 (50)	41
Latency for two RDs (row miss)	60	47 (48)	42

The numbers in ( ) are for the delaying ACT like comboAS [48, 61].

Table 4: Energy and current comparison

	DRAM	Conventional STT-MRAM	SMART
ACT (nJ)	1.28	0.45	0.09
Single RD (nJ)	0.27	0.26	0.31
Single WR (nJ)	0.28	0.35	0.34
Energy for 8B read (nJ)	1.54	0.71	0.39
Energy for 2KB read (nJ)	69.99	86.56	78.19
Energy for 8B write (nJ)	1.55	0.81	0.43
Energy for 2KB write (nJ)	72.79	96.80	87.40
IDD0 (mA)	67	64	43
IDD2P (mA)	14	17	16
IDD2N (mA)	36	39	38
IDD3N (mA)	51	39	38
IDD4R (mA)	122	121	127
IDD4W (mA)	122	132	131
IDD5 (mA)	245	-	-

such as tRRD, tFAW, tRAS and tRP, especially when different banks and rows need to be accessed.

4.3 Energy Model

In conventional STT-MRAM, activating fewer SAs reduces the energy consumption of a single ACT command. SMART, however, completely removes sensing energy from ACT and thus it gives much smaller ACT energy than conventional STT-MRAM, as shown in Table 4. Instead, SMART includes the sensing energy in RD and thus the RD energy is higher than DRAM and conventional STT-MRAM. The conventional STT-MRAM has the least RD energy because of the reduced read path. Besides, STT-MRAM consumes more WR energy than DRAM because of the higher write current per cell. However, considering the energy consumption to transfer data across the chip through long interconnects, the impact of the cell write energy on the total write energy is limited. Table 4 shows the dynamic energy consumption of a single memory device for a single column (8B) request and a single page (2KB) request. If all columns in a page are accessed, STT-MRAM consumes more energy because of its inherent higher sensing and cell write energies. However, a single read/write access in STT-MRAM consumes less energy than DRAM because of the low ACT energy consumption. Comparing the two STT-MRAM's energies, SMART is more energy-efficient in all cases. This is because there is no wasted energy for the single column access and there are fewer ACT commands for the 2KB access. In particular, because SMART does not include sensing energy in write requests, the gap between the two STT-MRAM designs in a 2KB write is larger than that in a 2KB read. In addition, we expect that the increased sensing energy by the longer sensing path in larger capacity is not directly reflected in total energy consumption of SMART because sensing energy is not a dominant component in SMART.

IDD2P is the power-down current and it is close to the sum of total transistor leakage. Thus, IDD2P is usually proportional to the total transistor width under the same technology. For simplicity, we increase IDD2P of STT-MRAM linearly with chip size. IDD2N and IDD3N are background current under precharge- and active-standby, respectively. The difference between IDD2P and IDD2N mainly results from the address/-clock buffer current components. Thus, we assume the same increment for STT-MRAM's IDD2N. However, IDD3N stems from DRAM's unique leakage component. When a row in a bank is activated, 32,768 BLs are fully charged or discharged, whereas all BLs are precharged to $V_{DD}/2$ level when the row is deactivated. If the BL length is 512, then 16,777,216 cells are connected to the 32,768 BLs. The increased voltage difference between BL to a cell transistor increases leakage current, which is mostly GIDL and junction leakage current [42]. Because of this large number, small leakage current changes cause huge increases in IDD3N. However, the BL/SL condition of STT-MRAM is different from that of DRAM during active-standby, because BLs/SLs are always discharged except during sensing and writing. Therefore, we assume IDD3N is the same as IDD2N in STT-MRAM.

5 EVALUATION

5.1 Evaluation Methodology

Table 5: Default system configuration

Component	Specification
Processor	single and quad core
Last Level Cache	2MB-8 way (single), 4MB-16 way (quad)
Memory Controller	FR-FCFS scheduling [49], open-page policy, Ch:Ra:Ro:Ba:Co and Ro:Co:RA:Ba:Ch mappings
Memory System	1∼4 channels (8GB/ch with 8Gb x8-DDR3L-1600)

We evaluate SMART using MARSSx86 [46] and DRAMSim2 [51]. The system configuration is shown in Table 5. Power-down mode is enabled to minimize standby power when there are no pending requests in the memory controller. For baseline DRAM, an 8GB memory channel (single rank) is composed using eight 8-bank, 8Gb x8 devices and 2KB page size [36]. We compare to two conventional STT-MRAM designs with the shared SA structure: Conv-Pin and

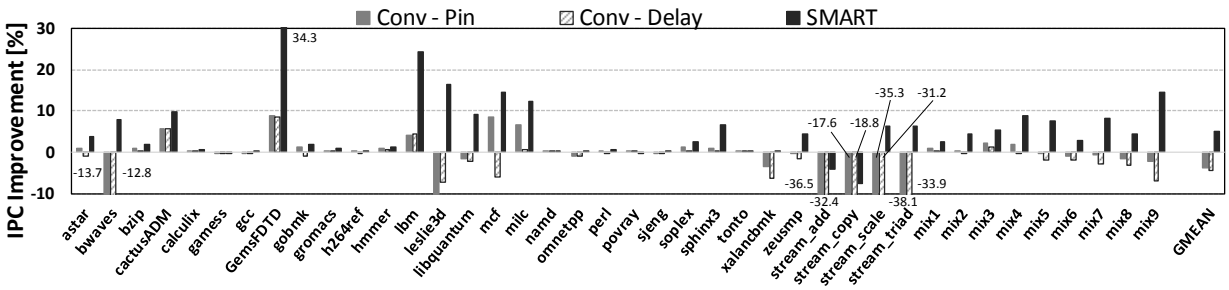


Figure 10: Performance improvement compared to DRAM.

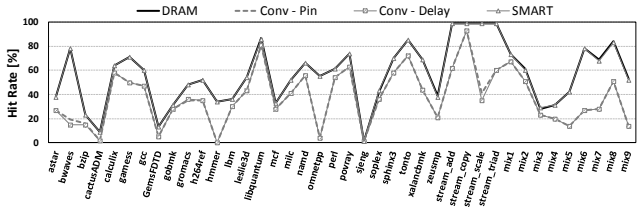


Figure 11: Row hit rate in various workloads.

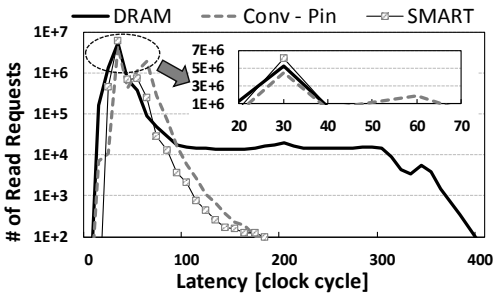


Figure 12: Read latency profile of mix6 workload.

Conv-Delay. Conv-Pin emulates the designs in [29, 35]. Although [29] is a PCM design, its bank structure and memory operation are applicable to most non-volatile memories. However, they have no consideration for the expanded address pins which result from the shared SA structure and are not compatible with JEDEC DDR. Conv-Delay, on the other hand, internally delays the activation instead of increasing the pin count [48, 61].

We employ two benchmark suites: SPEC2006 [13] and STREAM [34]. For multi-core simulations, multi-program workloads are composed using SPEC CPU2006 and STREAM Misses-per-kilo-instructions (MPKI) increases from mix1 to mix9. For all workloads, one billion instructions are simulated in their region of interest (ROI).

5.2 Performance

In SMART, we strive to implement large pages (2KB) with low cost. Memory performance is affected by row buffer hit rate and page size. Both DRAM and SMART can implement 2KB pages, while the conventional STT-MRAM can implement only 128B pages. The row buffer hit rates are shown in Fig. 11. Because our default address mapping scheme (Ro:Ba:Co) assigns LSB-side bits for column selection to exploit data locality, Fig. 11 exhibits good row hit, overall. In multi-program workloads, because workloads of different MPKI are mixed, high MPKI workloads (*e.g.,* STREAM and lbm) dominate overall memory accesses. SMART has a row hit rate within 1% of DRAM's, because page size, rather than column latency, mainly determines row hit rate under the same scheduling and memory address mapping scheme. However, both conventional designs have significantly lower hit rates, especially in STREAM workloads. Although all designs show low hit rate in a few workloads, the designs with the larger pages still perform better.

Fig. 12 shows the overall read latency distribution of the three devices under the mix6 workload. DRAM has a long tail latency because of refresh, but STT-MRAM has a short tail. In the Conv-Pin, we clearly observe two peaks in its distribution corresponding to read latency under row hits and misses, respectively. The Conv-Pin has a low hit rate of 27%. In SMART, there is no clear second peak because row misses are serviced faster and their read latency is partially overlapped with read latency under row hits. Although SMART has no data in 10∼19 cc range due to its long `CL`, it mostly falls within 30∼39 cc, implying that it is neither too quick nor too slow. DRAM also has most of its latency within 30∼39 cc, but its long tail negatively affects the overall latency. The average read latency of DRAM, Conv-Pin, and SMART is 52.6, 57.9 and 47.1 cc, respectively.

Fig. 10 shows system IPC improvement over DRAM. Conv-Pin and Conv-Delay degrade IPC on average by 3.7% and 4.3%. For some memory intensive workloads IPC degrades more than 30%. The biggest drawback in conventional STT-MRAM designs is the small page size. A substantial drop in row hit rate over DRAM (*e.g.*, bwaves, sphinx3 and STREAMs) significantly degrades their IPC. For workloads with similar row hit rate (*e.g.*, lbm and milc), IPC is better due to lack of refresh and restoration operations and the reduced precharge time. For non-memory-intensive workloads,

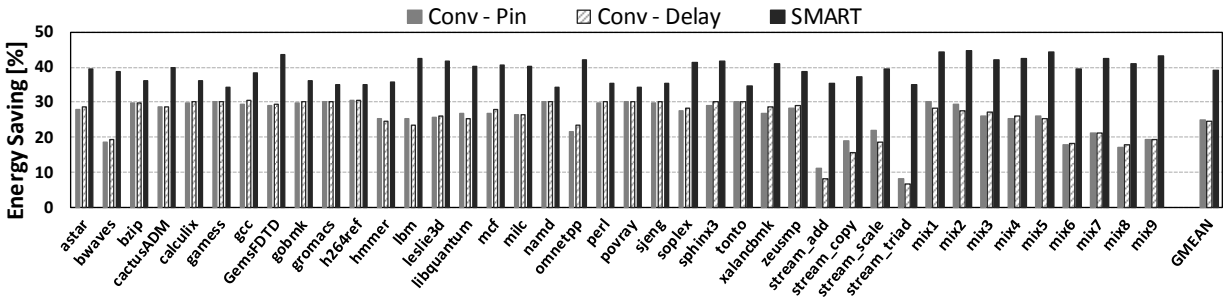


Figure 13: Energy savings over DRAM.

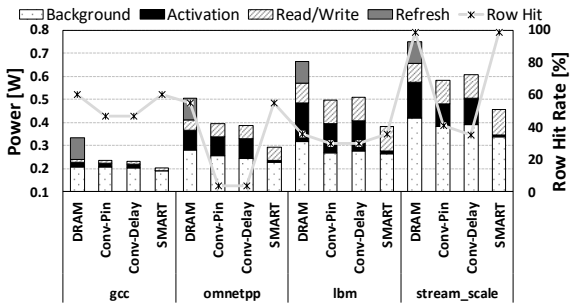
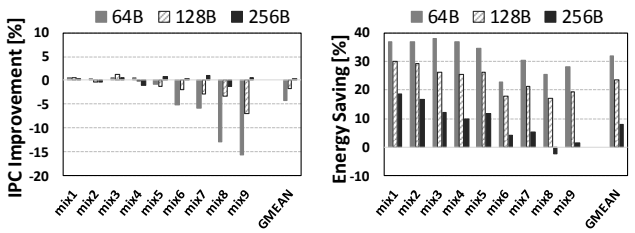


Figure 14: Breakdown of average memory power.



(a) Performance improvement **(b) Energy saving**

Figure 15: Normalized performance and energy of Conv-Delay to the baseline DRAM with various page size.

regardless of hit rate, the IPC difference is negligible. Between the two conventional designs, Conv-Delay shows worse overall performance than Conv-Pin because of its long `CL`.

SMART, on the other hand, improves IPC on average by 5.1% over DRAM. Because SMART has the same page size as DRAM, it achieves high row hit rates for applications having sequential memory accesses. In addition, applications having random memory accesses (low row hit rate) sees better row-miss latency and bank-level parallelism. As a result, MPKI is correlated to the IPC difference. In memory intensive workloads with (MPKI >15) (*e.g.*, lbm and libquantum), there is up to a 34% IPC improvement over DRAM. Non-memory intensive workloads with (MPKI < 1) (*e.g.*, gamess, and namd), show no significant IPC improvement.

5.3 Energy

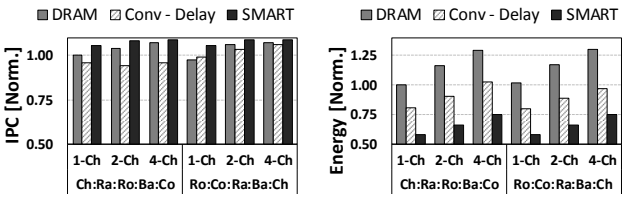
SMART has three energy advantages over DRAM. First, `ACT` energy is extremely low because sensing, which was the main energy contributor, was moved to `RD`. Second, cell leakage current is eliminated while the bank is activated (low `IDD3N`). Last, STT-MRAM cells are non-volatile and have no refresh. While both conventional designs enjoy similar advantages over DRAM, they saved `ACT` energy by reducing the page size and impacted the row hits.

Fig. 14 breaks down average memory power. `ACT` power is higher than `RD/WR` power in DRAM. The average refresh power is 14∼29% of the total memory average power and its relative portion increases in non-memory intensive workloads (*e.g.*, gcc). Although `ACT` power in the conventional STT-MRAM is less than in DRAM, the difference decreases when the conventional designs have low row hit rates (*e.g.*, omnetpp and stream_scale). In these workloads, because memory has fewer chances to enter power-down mode due to row misses, the background power becomes higher than in DRAM in spite of the lower `IDD3N`. `RD/WR` power is also higher than DRAM because of higher cell write current. In contrast, in SMART, the `ACT` power does not dominate total memory power. Although its `RD/WR` power is the highest among the four devices, the total dynamic power, which is the sum of `ACT` and `RD/WR` power, is the lowest. In addition, because SMART enters power-down mode as often as DRAM, background power stays low.

Fig. 13 shows the energy savings over DRAM. On average, the Conv-Pin and Conv-Delay save 24.9% and 24.5% of energy and SMART saves 38.9%. Due to small page size, energy savings of the conventional STT-MRAM are more sensitive to the row hit rate difference than SMART.

5.4 Sensitivity Analysis

Page size. As we discussed, page size of DRAM affects both performance and energy consumption in memory-intensive workloads. Fig. 15 shows that large pages improve performance and small pages save energy for various page sizes in conventional STT-MRAM. Overall, large pages improve performance while small pages save energy. 256B page slightly outperforms DRAM while saving less than 10% of energy.



(a) Normalized IPC **(b) Normalized energy**

Figure 16: Normalized IPC and energy to DRAM with various configurations (average of all mix workloads).

However, 256B (N=8) is not a practical page size considering the large, power-hungry SAs. In prior STT-MRAM chip demonstrations, N ranges from 16 to 128 [7, 24, 44, 48, 59]. N=16 was conservatively selected for the baseline STT-MRAM.

Address mapping and channels. Fig. 16 shows two mapping schemes with 1∼4 channels. In general, Ch:Ra:Ro:Ba:Co and Ro:Co:Ra:Ba:Ch are good for sequential and random accesses, respectively. DRAM performs the worst with a single channel because although the total overhead of refresh is unchanged regardless of the number of channels, the memory system is completely blocked during refresh in a single channel. Generally, more channels increase the total bandwidth, but they also increase memory chips and energy.

Conventional STT-MRAM is sensitive to the address mapping scheme. It performs better under Ro:Co:Ra:Ba:Ch because of its small pages. In general, the page size and row buffer locality are important to the Ch:Ra:Ro:Ba:Co mapping and the number of banks, ranks and channels plays a primary role under the Ro:Co:Ra:Ba:Ch mapping.

In contrast, SMART is less sensitive to address mapping and channels, because it has large pages, short row-miss latency, and better bank-level parallelism. In addition, there is no significant energy sensitivity to the number of channels, because SMART reduces both dynamic and static energies.

6 RELATED WORK

Asifuzzaman *et al.* analyzes the impact of a slow STT-MRAM main memory on high performance computing (HPC) [2]. They add 20% slowdown, estimated from industry, to the main memory over DRAM and evaluate it on the HPC applications. Their evaluation results yield that 20% slower main memory has negligible impact on overall system performance due to the limited role of the main memory on the system and out-of-order pipelining. Kultursay *et al.* evaluate STT-MRAM as a main memory with optimizations such as partial write and write bypass [28]. They show comparable performance with DRAM and a 60% reduction in memory dynamic energy. Wang *et al.* investigate the design challenges of shared SAs such as small pages and pin compatibility [61]. With memory-architectural study, they propose three optimizations, comboAS, DynLat, and EarlyPA. Although these studies solve the compatibility problem and compensate for the reduced performance, the root cause of small pages and low chip yield remains unsolved.

LPDDR2-SX was designed for DRAM and its counterpart LPDDR2-NVM was designed for non-volatile devices with long write latency and large read/write circuits such as PCM [19, 31]. LPDDR2-NVM introduces a new command to deliver column selection rather than using additional pins. Although it can be a good candidate for STT-MRAM, its inherent performance is worse than LPDDR2-SX because of its three-phase addressing and software managed indirect write. Recently, 4Gb STT-MRAM has been demonstrated with LPDDR2-SX but not LPDDR2-NVM [8, 48].

SALP-1 [26] and EarlyPA [61] are similar to our skipping precharge in terms of alleviating `PRE` overhead. Unlike our technique, however, SALP-1 can only overlap `PRE` and next `ACT` when their sub-arrays are different. In EarlyPA, the precharge operation is automatically performed immediately after the sensing operation. Although this technique can efficiently hide the precharging time when the following command is `RD`, WL must be reactivated when the following command is `WR`. In contrast, our technique can hide `PRE` latency for both `RD` and `WR`.

7 CONCLUSION

We proposed SMART, a new cost-effective STT-MRAM architecture. We showed that by performing the sensing operation after a `RD` command instead of an `ACT` command, several benefits result. They include: larger pages, fewer sense amps, lower activation power, higher bank-level parallelism, shorter latency, fewer address pins, and more efficient column repair over conventional design. These benefits not only reduce energy but also improve performance compared to both DRAM and conventional STT-MRAM. In addition to the improvements in energy consumption and performance, SMART saves area in comparison to conventional STT-MRAM.

ACKNOWLEDGMENTS

This work is supported in part by ARM Ltd. and an NSF grant (CNS-1705047).

REFERENCES

[1] Syed M Alam, Thomas Andre, and Dietmar Gogl. 2016. Memory controller and method for interleaving DRAM and MRAM accesses. US Patent 9,418,001.

[2] Kazi Asifuzzaman, Milan Pavlovic, Milan Radulovic, David Zaragoza, Ohseong Kwon, Kyung-Chang Ryoo, and Petar Radojković. 2016. Performance Impact of a Slower Main Memory: A Case Study of STT-MRAM in HPC. In *International Symposium on Memory Systems (MEMSYS)*.

[3] Meng-Fan Chang, Shin-Jang Shen, Chia-Chi Liu, Che-Wei Wu, Yu-Fan Lin, Ya-Chin King, Chorng-Jung Lin, Hung-Jen Liao, Yu-Der Chih, and Hiroyuki Yamauchi. 2013. An offset-tolerant fast-random-read current-sampling-based sense amplifier for small-cell-current nonvolatile memory. *IEEE Journal of Solid-State Circuits (JSSC)* (2013).

[4] Meng-Fan Chang, Shyh-Shyuan Sheu, Ku-Feng Lin, Che-Wei Wu, Chia-Chen Kuo, Pi-Feng Chiu, Yih-Shan Yang, Yu-Sheng Chen, Heng-Yuan Lee, Chen-Hsin Lien, et al. 2013. A high-speed 7.2-ns read-write random access 4-mb embedded resistive ram (reram) macro using process-variation-tolerant current-mode read schemes. *IEEE Journal of Solid-State Circuits (JSSC)* (2013).

[5] Youngdon Choi, Ickhyun Song, Mu-Hui Park, Hoeju Chung, Sanghoan Chang, Beakhyoung Cho, Jinyoung Kim, Younghoon Oh, Duckmin Kwon, Jung Sunwoo, et al. 2012. A 20nm 1.8 V 8Gb PRAM with 40MB/s program bandwidth. In *International Solid-State Circuits Conference (ISSCC)*.

[6] Hoeju Chung, Byung Hoon Jeong, ByungJun Min, Youngdon Choi, Beak-Hyung Cho, Junho Shin, Jinyoung Kim, Jung Sunwoo, Joon-min Park, Qi Wang, et al. 2011. A 58nm 1.8 v 1gb pram with 6.4 mb/s program bw. In *International Solid-State Circuits Conference (ISSCC)*.

[7] Suock Chung, K-M Rho, S-D Kim, H-J Suh, D-J Kim, H-J Kim, S-H Lee, J-H Park, H-M Hwang, S-M Hwang, et al. 2010. Fully integrated 54nm STT-RAM with the smallest bit cell dimension for high density memory application. In *International Electron Devices Meeting (IEDM)*.

[8] S-W Chung, T Kishi, JW Park, M Yoshikawa, KS Park, T Nagase, K Sunouchi, H Kanaya, GC Kim, K Noma, et al. 2016. 4Gbit density STT-MRAM using perpendicular MTJ realized with compact cell structure. In *International Electron Devices Meeting (IEDM)*.

[9] Elliott Cooper-Balis and Bruce Jacob. 2010. Fine-grained activation for power reduction in DRAM. *IEEE Micro* (2010).

[10] Vinodh Cuppu, Bruce Jacob, Brian Davis, and Trevor Mudge. 1999. A performance comparison of contemporary DRAM architectures. In *International Symposium on Computer Architecture (ISCA)*.

[11] Xiangyu Dong, Cong Xu, Yuan Xie, and Norman P Jouppi. 2012. NVSim: A Circuit-Level Performance, Energy, and Area Model for Emerging Nonvolatile Memory. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* (2012).

[12] Cécile Grezes, Hochul Lee, Albert Lee, Shaodi Wang, Farbod Ebrahimi, Xiang Li, Kin Wong, Jordan A Katine, Berthold Ocker, Jürgen Langer, et al. 2017. Write Error Rate and Read Disturbance in Electric-Field-Controlled Magnetic Random-Access Memory. *IEEE Magnetics Letters* 8 (2017), 1–5.

[13] John L Henning. 2006. SPEC CPU2006 benchmark descriptions. *ACM SIGARCH Computer Architecture News* (2006).

[14] Masahi Horiguchi, Jun Etoh, Masakazu Aoki, K Itoh, and T Matsumoto. 1991. A flexible redundancy technique for high-density DRAMs. *IEEE Journal of Solid-State Circuits (JSSC)* (1991).

[15] Masashi Horiguchi and Kiyoo Itoh. 2011. *Nanoscale memory repair*. Springer Science & Business Media.

[16] ITRS. 2013. http://www.itrs2.net/2013-itrs.html.

[17] Bruce Jacob, Spencer Ng, and David Wang. 2010. *Memory systems: cache, DRAM, disk*. Morgan Kaufmann.

[18] JEDEC. 2009. DDR3 SDRAM Specification. www.jedec.org/sites/default/files/docs/JESD79-3E.pdf.

[19] JEDEC. 2009. Low Power Double Data Rate 2 (LPDDR2). http://www.jedec.org/sites/default/files/docs/JESD209-2B.pdf.

[20] JEDEC. 2013. High Bandwidth Memory (HBM) DRAM. https://www.jedec.org/sites/default/files/docs/JESD235A.pdf.

[21] JEDEC. 2013. Low Power Double Data Rate 3 (LPDDR3). http://www.jedec.org/sites/default/files/docs/JESD209-3C.pdf.

[22] Mihail Jefremow, Thomas Kern, Wolf Allers, Christian Peters, Jan Otterstedt, Othmane Bahlous, Karl Hofmann, Robert Allinger, Stephan Kassenetter, and Doris Schmitt-Landsiedel. 2013. Time-differential sense amplifier for sub-80mV bitline voltage embedded STT-MRAM in 40nm CMOS. In *International Solid-State Circuits Conference (ISSCC)*.

[23] JJ Kan, C Park, C Ching, J Ahn, L Xue, R Wang, A Kontos, S Liang, M Bangar, H Chen, et al. 2016. Systematic validation of 2x nm diameter perpendicular MTJ arrays and MgO barrier for sub-10 nm embedded STT-MRAM with practically unlimited endurance. In *International Electron Devices Meeting (IEDM)*.

[24] Chankyung Kim, Keewon Kwon, Chulwoo Park, Sungjin Jang, and Joosun Choi. 2015. 7.4 A covalent-bonded cross-coupled current-mode sense amplifier for STT-MRAM with 1t1mtj common source-line structure array. In *International Solid-State Circuits Conference (ISSCC)*.

[25] Doo-Gon Kim and Ki-Tae Park. 2011. Semiconductor memory device with three-dimensional array and repair method thereof. US Patent 8,031,544.

[26] Yoongu Kim, Vivek Seshadri, Donghyuk Lee, Jamie Liu, and Onur Mutlu. 2012. A case for exploiting subarray-level parallelism (SALP) in DRAM. (2012).

[27] E Kitagawa, S Fujita, K Nomura, H Noguchi, K Abe, K Ikegami, T Daibou, Y Kato, C Kamata, S Kashiwada, et al. 2012. Impact of ultra low power and fast write operation of advanced perpendicular MTJ on power reduction for high-performance mobile CPU. In *International Electron Devices Meeting (IEDM)*.

[28] Emre Kültürsay, Mahmut Kandemir, Anand Sivasubramaniam, and Onur Mutlu. 2013. Evaluating STT-RAM as an energy-efficient main memory alternative. In *Performance Analysis of Systems and Software (ISPASS)*.

[29] Benjamin C Lee, Engin Ipek, Onur Mutlu, and Doug Burger. 2009. Architecting phase change memory as a scalable dram alternative. In *International Symposium on Computer Architecture (ISCA)*.

[30] Dong Uk Lee, Kang Seol Lee, Yongwoo Lee, Kyung Whan Kim, Jong Ho Kang, Jaejin Lee, and Jun Hyun Chun. 2015. Design considerations of HBM stacked DRAM and the memory architecture extension. In *Custom Integrated Circuits Conference (CICC)*.

[31] Zhongqi Li, Ruijin Zhou, and Tao Li. 2013. Exploring high-performance and energy proportional interface for phase change memory systems. In *International Symposium on High Performance Computer Architecture (HPCA)*.

[32] Kyu-Nam Lim, Woong-Ju Jang, Hyung-Sik Won, Kang-Yeol Lee, Hyungsoo Kim, Dong-Whee Kim, Mi-Hyun Cho, Seung-Lo Kim, Jong-Ho Kang, Keun-Woo Park, et al. 2012. A 1.2 V 23nm 6F 2 4Gb DDR3 SDRAM with local-bitline sense amplifier, hybrid LIO sense amplifier and dummy-less array architecture. In *International Solid-State Circuits Conference (ISSCC)*.

[33] CJ Lin, SH Kang, YJ Wang, K Lee, X Zhu, WC Chen, X Li, WN Hsu, YC Kao, MT Liu, et al. 2009. 45nm low power CMOS logic compatible embedded STT MRAM utilizing a reverse-connection 1T/1MTJ cell. In *Electron Devices Meeting (IEDM), 2009 IEEE International*. IEEE, 1–4.

[34] John D McCalpin. 1995. A survey of memory bandwidth and machine balance in current high performance computers. *IEEE TCCA Newsletter* (1995).

[35] Justin Meza, Jing Li, and Onur Mutlu. 2012. Evaluating row buffer locality in future non-volatile main memories. (2012).

[36] Micron. 2015. 8Gb DDR3L, MT41K1G8.

[37] Kyeong-Sik Min and Jin-Yong Chung. 2001. A fast pump-down V BB generator for sub-1.5-V DRAMs. *IEEE Journal of Solid-State Circuits (JSSC)* (2001).

[38] Yongsam Moon, Yong-Ho Cho, Hyun-Bae Lee, Byung-Hoon Jeong, Seok-Hun Hyun, Byung-Chul Kim, In-Chul Jeong, Seong-Young Seo, Jun-Ho Shin, Seok-Woo Choi, et al. 2009. 1.2 V 1.6 Gb/s 56nm 6F 2 4Gb DDR3 SDRAM with hybrid-I/O sense amplifier and segmented sub-array architecture. In *International Solid-State Circuits Conference (ISSCC)*.

[39] Taehui Na, Jisu Kim, Jung Pill Kim, Seung H Kang, and Seong-Ook Jung. 2014. An offset-canceling triple-stage sensing circuit for deep submicrometer STT-RAM. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* (2014).

[40] Omar Naji, Christian Weis, Matthias Jung, Norbert Wehn, and Andreas Hansson. 2015. A high-level DRAM timing, power and area exploration tool. In *Embedded Computer Systems: Architectures, Modeling, and Simulation (SAMOS)*. IEEE.

[41] Janusz J Nowak, Ray P Robertazzi, Jonathan Z Sun, Guohan Hu, Jeong-Heon Park, JungHyuk Lee, Anthony J Annunziata, Gen P Lauer, Raman Kothandaraman, Eugene J OâĂŹSullivan, et al. 2016. Dependence of voltage and size on write error rates in spin-transfer torque magnetic random-access memory. *IEEE Magnetics Letters* 7 (2016), 1–4.

[42] Byoungchan Oh, Nilmini Abeyratne, Jeongseob Ahn, Ronald G Dreslinski, and Trevor Mudge. 2016. Enhancing DRAM Self-Refresh for Idle Power Reduction. In *International Symposium on Low Power Electronics and Design (ISLPED)*.

[43] Byoung-Chan Oh, Ji-Hyae Bae, Katsuyuki Fujita, and Yutaka Shirai. 2014. Electronic device including semiconductor memory and operation method thereof. US Patent 6,442,585.

[44] C Park, JJ Kan, C Ching, J Ahn, L Xue, R Wang, A Kontos, S Liang, M Bangar, H Chen, et al. 2015. Systematic optimization of 1 Gbit perpendicular magnetic tunnel junction arrays for 28 nm embedded STT-MRAM and beyond. In *International Electron Devices Meeting (IEDM)*.

[45] J Park, D-H Shin, Y-H Cho, and K-W Kwon. 2016. Inverted bit-line sense amplifier with offset-cancellation capability. *Electronics Letters* (2016).

[46] Avadh Patel, Furat Afram, Shunfei Chen, and Kanad Ghose. 2011. MARSS: a full system simulator for multicore x86 CPUs. In *Design Automation Conference (DAC)*.

[47] Arijit Raychowdhury, Dinesh Somasekhar, Tanay Karnik, and Vivek De. 2009. Design space and scalability exploration of 1T-1STT MTJ memory arrays in the presence of variability and disturbances. In *International Electron Devices Meeting (IEDM)*.

[48] Kwangmyoung Rho, Kenji Tsuchida, Dongkeun Kim, Yutaka Shirai, Jihyae Bae, Tsuneo Inaba, Hiromi Noro, Hyunin Moon, Sungwoong Chung, Kazumasa Sunouchi, et al. 2017. 23.5 A 4Gb LPDDR2 STT-MRAM with compact 9F2 1T1MTJ cell and hierarchical bitline architecture. In *International Solid-State Circuits Conference (ISSCC)*.

[49] Scott Rixner, William J Dally, Ujval J Kapasi, Peter Mattson, and John D Owens. 2000. Memory access scheduling. In *International Symposium on Computer Architecture (ISCA)*.

[50] ND Rizzo, D Houssameddine, R Janesky, J chand Whig, FB Mancoff, ML Schneider, M DeHerrera, JJ Sun, K Nagel, S Deshpande, et al. 2013. A fully functional 64 Mb DDR3 ST-MRAM built on 90 nm CMOS technology. *IEEE Transactions on Magnetics* (2013).

[51] Paul Rosenfeld, Elliott Cooper-Balis, and Bruce Jacob. 2011. DRAMSim2: A cycle accurate memory system simulator. *IEEE Computer Architecture Letters* (2011).

[52] Daisuke Saida, Saori Kashiwada, Megumi Yakabe, Tadaomi Daibou, Miyoshi Fukumoto, Shinji Miwa, Yoshishige Suzuki, Keiko Abe, Hiroki Noguchi, Junichi Ito, et al. 2017. 1x- to 2x-nm perpendicular MTJ Switching at Sub-3-ns Pulses Below 100uA for High-Performance Embedded STT-MRAM for Sub-20-nm CMOS. *IEEE Transactions on Electron Devices* 64, 2 (2017), 427–431.

[53] SK hynix. 2014. Evolutionary Migration from LPDDR3 to LPDDR4. https://www.jedec.org/sites/default/files/Minho_SK%20hynix_CES_14_new.pdf.

[54] Byungkyu Song, Taehui Na, Jisu Kim, Jung Pill Kim, Seung H Kang, and Seong-Ook Jung. 2015. Latch offset cancellation sense amplifier for deep submicrometer STT-RAM. *IEEE Transactions on Circuits and Systems I (TCAS I)* (2015).

[55] Kshitij Sudan, Niladrish Chatterjee, David Nellans, Manu Awasthi, Rajeev Balasubramonian, and Al Davis. 2010. Micro-pages: increasing DRAM efficiency with locality-aware data placement. In *Architectural Support for Programming Languages and Operating Systems (ASPLOS)*.

[56] Amoghavarsha Suresh, Pietro Cicotti, and Laura Carrington. 2014. Evaluation of emerging memory technologies for HPC, data intensive applications. In *International Conference on Cluster Computing (CLUSTER)*.

[57] Texas Instruments. 2015. Keystone Architecture DDR3 Memory Controller. http://www.ti.com/lit/ug/sprugv8e/sprugv8e.pdf.

[58] Luc Thomas, Guenole Jan, Jian Zhu, Huanlong Liu, Yuan-Jen Lee, Son Le, Ru-Ying Tong, Keyu Pi, Yu-Jen Wang, Dongna Shen, et al. 2014. Perpendicular spin transfer torque magnetic random access memories with high spin torque efficiency and thermal stability for embedded applications. *Journal of Applied Physics* (2014).

[59] Kenji Tsuchida, Tsuneo Inaba, Katsuyuki Fujita, Yoshihiro Ueda, Takafumi Shimizu, Yoshiaki Asao, Takeshi Kajiyama, Masayoshi Iwayama, Kuniaki Sugiura, Sumio Ikegawa, et al. 2010. A 64Mb MRAM with clamped-reference and adequate-reference schemes. In *International Solid-State Circuits Conference (ISSCC)*.

[60] Aniruddha N Udipi, Naveen Muralimanohar, Niladrish Chatterjee, Rajeev Balasubramonian, Al Davis, and Norman P Jouppi. 2010. Rethinking DRAM design and organization for energy-constrained multi-cores. In *International Symposium on Computer Architecture (ISCA)*.

[61] Jue Wang, Xiangyu Dong, and Yuan Xie. 2014. Enabling high-performance LPDDRx-compatible MRAM. In *International Symposium on Low Power Electronics and Design (ISLPED)*.

[62] DC Worledge, G Hu, PL Trouilloud, DW Abraham, S Brown, MC Gaidis, J Nowak, EJ O'Sullivan, RP Robertazzi, JZ Sun, et al. 2010. Switching distributions and write reliability of perpendicular spin torque MRAM. In *International Electron Devices Meeting (IEDM)*.

[63] Tien-Chun Yang, Yue-Der Chih, and Shang-Hsuan Liu. 2012. Redundancy circuits and operating methods thereof. US Patent 8,238,178.

[64] Jei-Hwan Yoo, Chang Hyun Kim, Kyu Chan Lee, Kye-Hyun Kyung, Seung-Moon Yoo, Jung Hwa Lee, Moon-Hae Son, Jin-Man Han, Bok-Moon Kang, Ejaz Haq, et al. 1996. A 32-bank 1 Gb DRAM with 1 GB/s bandwidth. In *International Solid-State Circuits Conference (ISSCC)*.

[65] Tao Zhang, Ke Chen, Cong Xu, Guangyu Sun, Tao Wang, and Yuan Xie. 2014. Half-DRAM: a High-bandwidth and Low-power DRAM Architecture from the Rethinking of Fine-grained Activation. In *International Symposium on Computer Architecture (ISCA)*.