Improving STT-MRAM Density Through Multibit Error Correction

Brandon Del Bel, Jongyeon Kim, Chris H. Kim, and Sachin S. Sapatnekar
Department of ECE, University of Minnesota
{delbel, kimx2889, chriskim, sachin}@umn.edu

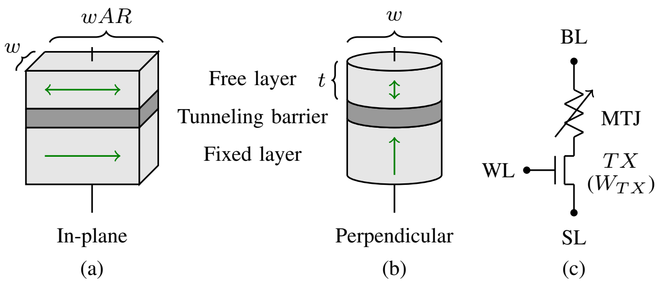
***Abstract*—STT-MRAMs are prone to data corruption due to inadvertent bit flips. Traditional methods enhance robustness at the cost of area/energy by using larger cell sizes to improve the thermal stability of the MTJ cells. This paper employs multibit error correction with DRAM-style refreshing to mitigate errors and provides a methodology for determining the optimal level of correction. A detailed analysis demonstrates that the reduction in nonvolatility requirements afforded by strong error correction translates to significantly lower area for the memory array compared to simpler ECC schemes, even when accounting for the increased overhead of error correction.**

I. Introduction

Spin-transfer torque magnetoresistive RAMs (STT-MRAMs) have received much attention in recent years as a replacement for SRAMs in cache and off-chip memory applications. STT-MRAMs are attractive because of their compact design, non-volatility, low leakage power, and their potential for scalability.

The principal component of an STT-MRAM is the magnetic tunnel junction (MTJ). MTJs come in two flavors, depending on the orientation of their anisotropy relative to the substrate. Initial work on STT-MRAMs focused on the use of *in-plane* MTJs (Fig. 1(a)). Recent work has also addressed the less mature technology based on *perpendicular* MTJs (Fig. 1(b)), which have lower switching current densities. In circuit applications, both types of devices are similarly deployed, but they differ in attributes such as volume, aspect ratio, switching current, and scalability in future technologies [5], [7].

Several architectures have been proposed for STT-MRAM cells, but fundamentally all consist of an MTJ, an access transistor, and contacts for the word line (WL), bit line (BL), and source line (SL). The differences arise from the way the MTJ is connected to the transistor, the number of transistors used, and the cell layout. A common design uses one access transistor TX and an MTJ (1T1MTJ) with the free layer connected to the bit line BL, as shown in Fig. 1(c) [14], [20].



In-plane
(a)

Perpendicular
(b)

(c)

Fig. 1. Simplified in-plane (a) and perpendicular (b) MTJ stacks and 1T1MTJ bit cell schematic (c).

The MTJ stores state using free layer polarization relative to a fixed layer. The two states – parallel and antiparallel – have different resistances, allowing for read operations with voltage sense amplifiers after passing a small current pulse through the access transistor TX. The fixed layer is also used to write the free layer by either passing a larger current or applying a longer pulse (as compared to the read operation) through TX.

A key attribute of an MTJ is the notion of *thermal stability*. Fig. 2 illustrates the stable states (parallel and antiparallel) of the MTJ free layer and shows the energy barrier, E, that separates them. In the absence of spin current, the actual state of the magnet is perturbed slightly from its current stable state due to random thermal fluctuations that have an average magnitude of k_BT, where k_B is the Boltzmann constant and T is the temperature in Kelvin. If the energy from these random fluctuations exceeds E, it can overcome the energy barrier and erroneously change state. The thermal stability factor, Δ, (often referred to informally as the thermal stability) is the height of the energy barrier relative to k_BT, and is related to the likelihood that a state is inadvertently flipped: the larger the value of Δ, the less likely the free layer is to spontaneously change state. The thermal stability factor is given by

$$\Delta = \frac{E}{k_BT} = \frac{H_kM_sV}{2k_BT} \qquad (1)$$

where H_k is the effective anisotropy field, M_s is the saturation magnetization, and V is the free layer volume [5].

In 1T1MTJ designs, the cell area is dominated by the size of the access transistor: for example, a reference 45nm design [12] uses a size that is 6× wider than the minimum feature size. Better integration densities for memory cells can be achieved by using smaller memory cells, but there is a tradeoff between cell area and error rate, both of which are related to the thermal stability of the free layer. Fig. 3 illustrates this tradeoff: as the thermal stability is increased, the error rate drops off since it becomes increasingly difficult to surmount the energy barrier. However, the area of the cell also increases correspondingly, for two reasons: first, the dimension of the MTJ must be increased

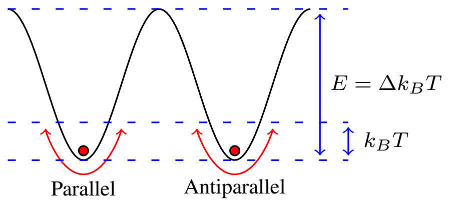


Fig. 2. Thermal fluctuations in an MTJ free layer [20].

This work was supported in part by C-SPIN, one of the six SRC STARnet Centers, sponsored by MARCO and DARPA.
978-3-9815370-2-4/DATE14/©2014 EDAA

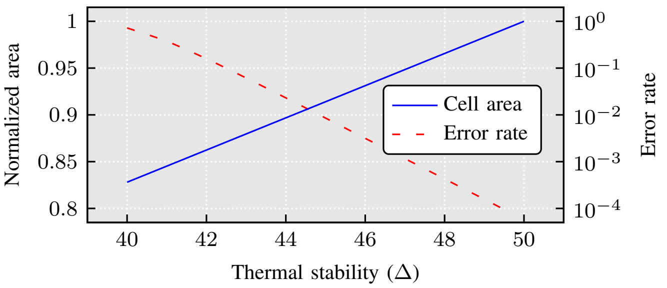


Fig. 3. Impact of thermal stability on cell area and error rate.

to ensure the desired thermal stability. Second, the write current required to change state increases, implying that a wider access transistor must be used to maintain the write time.

The objective of this work is to exploit the impact of this tradeoff on memory systems. Reducing Δ translates to reducing cell area, since both the MTJ and the access transistor become smaller, as well as the write energy. This allows for smaller bit cells and more compact memory arrays, at the expense of increased error probabilities as random perturbations are more likely to be able to surmount the reduced energy barrier. To compensate for the increased bit flip probability, we utilize multibit error correction coding (ECC) with refresh operations.

Our work is based on original derivations of analytical models for STT-RAM failure rates and models that relate STT-RAM area to the thermal stability factor (and hence, the failure rate) for both in-plane and perpendicular MTJ memories. We also provide a methodology for determining the optimal level of correction. To our knowledge, we are the first to demonstrate that the reduction in thermal stability without increased error rates afforded by multibit ECC can significantly reduce overall STT-MRAM area compared to single-bit correction.

Prior approaches have also explored the concept of permitting increased error rates in STT-MRAMs to reap the benefits of reducing cell areas [9], [15], [18], [19]. These works have at best used single-bit correction [15], leaving the full potential of ECC untapped. The only related work that considers multibit errors [22] focuses on variability issues without touching upon other benefits of thermal stability. Our work shows that multibit ECC can provide an additional 21% improvement in density for current STT-MRAM designs compared to single-bit correction.

Moreover, all of these approaches use simple models for retention times, without tying the retention time to error rate metrics such as failures-in-time (FITs). In Section III, we derive explicit generalized expressions that provide simple analytical models for the error rate in the presence of ECCs that correct c errors; plainly, when $c = 1$, our expressions can be applied to the single-bit case explored in prior work. In addition, our methodology is applicable to any use of STT-MRAM where data retention is critical, such as write-back caches or off-chip storage, thus covering a broader spectrum than prior work.

II. Thermal Stability and Area

A. Thermal Reversals and Data Retention

1) Errors in a Single MTJ: A low error rate is an important property of a reliable memory array. Although STT-RAMs are often informally described as nonvolatile, this property is not absolute. There is a statistical chance that random thermal fluctuations will cause the MTJ free layer magnetization to overcome the energy barrier in Fig. 2, resulting in a bit flip. The mean time between two reversals, known as the relaxation time, is $\tau = \tau_0 e^{\Delta}$, where the attempt period, τ_0, represents how frequently reversal attempts occur [17]. Typical values of τ_0 are on the order of a nanosecond.

There are three types of errors based on unintentional free layer magnetization reversal: *read errors* are caused by passing too much current while reading the state of the memory, *write errors* are caused by passing too little current while reading the state of the memory, and *random errors* occur in the absence of MTJ current and are caused by thermal noise. All failure modes are characterized by the probability of free layer reversal [20]:

$$P_{rev} = 1 - e^{-\frac{t_r}{\tau_0} e^{-\Delta\left(1-\frac{I}{I_{c0}}\right)}} \tag{2}$$

where t_r is the period over which the evaluation is conducted, I is the current that passes through the MTJ, and I_{c0} is the critical current of the MTJ, a concept discussed in greater detail in Section II-B. When $I = 0$, this expression is simplified to

$$P_{rev} = 1 - e^{-\frac{t_r}{\tau_0} e^{-\Delta}} \tag{3}$$

Ideally, free layer reversal should occur 100% of the time when the state of the MTJ is altered during a write operation, and 0% otherwise, including during read operations, idle states, or write operations that retain the previous state. Read and write errors show a departure from ideal behavior while accessing a memory cell, and the actual rates are given by the above equation. Several techniques can be used to mitigate errors, including using larger access transistors to provide more write current [22] and shorter read pulses to avoid accidental writes.

Read and write operations to memory are relatively rare. For an overwhelming majority of the time, the STT-MRAM cells are simply expected to retain their data, i.e., MTJ polarizations should ideally remain unchanged from their previously written states and there should be a 0% chance of reversal with $I = 0$.

A typical specification on an STT-MRAM is the *retention time*, i.e., the desired duration for which no bit flips should occur. The error probability during standby in a single MTJ cell is obtained from Eq. (3) by setting t_r to the retention time.

2) Errors in an STT-RAM Array: For an STT-MRAM with m bits, the probability that none of the m cells experiences an error is given by the m MTJs, $(1-P_{rev})^m$. We ignore the case where a cell flips an even nonzero number of times, ending back at the correct value, because the probability of this event is negligible. For a given value of the retention time, t_r, the probability of failure, λ, for a memory array with m cells is given by the complement of this value. In standby mode where $I = 0$, this value is computed using Eq. (3) as:

$$\lambda = 1 - \left(e^{-\frac{t_r}{\tau_0} e^{-\Delta}}\right)^m = 1 - e^{-m\frac{t_r}{\tau_0} e^{-\Delta}} \tag{4}$$

An analysis of this equation leads to the following insights. *Thermal stability*: The failure rate can be reduced by increasing Δ. However, there is a trade-off between thermal stability and cell area, described in Fig. 3 and quantified in Section II-B. *Memory size*: The memory array size is important in affecting the design of individual MTJ cells. A typical memory failure specification may be expressed in units of FIT (failure in time), where 1 FIT (failure in time) corresponds to one failure per billion [devices × operational hours], translating to a failure rate of 0.00876% for one device over 10 years. To achieve a failure rate of 1 FIT for a single MTJ, solving Eq. (4) for Δ gives a required thermal stability factor of $50k_BT$. For a gigabit array, this number becomes $70k_BT$, which translates to overheads in memory area/packing density as well as read/write power. The goal of using error correction is to reduce these overheads by softening the stringent requirement on Δ.

Refresh operations: The longer t_r is, the more likely errors are, and the use of refresh operations can be used to ensure data integrity. A similar concept is used in DRAM arrays, which periodically read and rewrite data back to cells to compensate for temporal degradation in voltage levels. However, as explained in Section III-B, STT-MRAM errors are stochastic and are different from DRAM errors, which are used to correct deterministic errors caused by the charge degradation over time in DRAM cells. If a DRAM-like refresh operation were to be used on an STT-RAM cell, it would read an inadvertently flipped bit and write back the incorrect value, thus providing no relief in lowering the error rate. Therefore any refresh operation must be accompanied by error correction.

B. Relating Bit Cell Area to Thermal Stability

From Eq. (1), MTJs with larger Δ values must be larger in size. Larger MTJs require a larger switching current, resulting in the need for larger access transistors and thus increasing the cell size. The critical MTJ switching current density, J_{c0}, for current-driven magnetization reversal is given by [2], [8]:

$$J_{c0} = \frac{1}{\eta}\frac{2\alpha e}{\hbar} M_s t \left(2\pi M_s + H_k\right) \tag{5}$$

where η is the spin transfer torque efficiency, α is the damping constant, e is the charge of an electron, $\hbar$ is the reduced Planck constant, and t is the free layer thickness.

For an in-plane magnet, the term H_k is based on shape anisotropy and depends on t/w and is typically dominated by the $2\pi M_s$ term [2] in the expression for J_{c0} above. For a perpendicular magnet, H_k is determined by bulk crystalline anisotropy and is independent of device dimensions. Therefore, for either case, one may conclude that J_{c0} is proportional to t.

The switching current, I_{c0} is the product of J_{c0} with the cross-sectional area of the magnet, which is $\frac{\pi}{4}w^2$ for perpendicular devices and $w^2 AR$ for in-plane devices, where AR is the aspect ratio. The waveform of the switching current, I_c, is determined by I_{c0} and the pulse width [6], [21], and when the pulse duration is on the order of a few nanoseconds, $I_c \approx I_{c0}$. Since the amount of current that a transistor can drive is directly proportional to its width W_{TX}, it follows that

$$W_{TX} \propto I_c \propto tw^2 \tag{6}$$

The dependence of Δ on the free layer thickness and width is captured by its proportionality to H_kV. For in-plane and perpendicular devices, this can be expressed as [2], [8]:

$$\Delta_{=} \propto t^2 w, \quad \Delta_{\perp} \propto tw^2 \tag{7}$$

where "=" and "$\perp$" represent the in-plane and perpendicular cases, respectively. The difference between the two expressions above arises from the H_k dependence on t/w for in-plane devices and the independence of H_k on t and w for perpendicular devices. For perpendicular devices, Eqs. (6) and (7) imply:

$$W_{TX} \propto \Delta \tag{8}$$

For in-plane MTJs, several scaling scenarios may be considered:

- By scaling $\Delta \to \gamma\Delta$ as $t \to \sqrt[3]{\gamma}\, t$ and $w \to \sqrt[3]{\gamma}\, w$, W_{TX} also scales by γ and $W_{TX} \propto \Delta$
- If only t is scaled, then $W_{TX} \propto \sqrt{\Delta}$.
- If only w is scaled, then $W_{TX} \propto \Delta^2$.

For now, we assume that both scale equally so that our analysis is equally valid for both in-plane and perpendicular devices. Alternative scaling schemes are briefly explored in Section V-D.

Next, we consider the impact of changing Δ on the area of an STT-RAM bit cell. Although the exact dimensions of the cell are layout-dependent, the size is dominated by the access transistor. The ratio of transistor width to cell width $R = W_{TX}/W_{cell}$ can be as high as $W_{TX}/\left(W_{TX} + F\right)$, where F is the process feature size [20]. Reducing Δ through free layer scaling can substantially reduce bit cell area and increase memory density as long as W_{TX} is not reduced beyond F.

III. Error Correction

From Eqs. (3) and (4), a reduction in the thermal stability, Δ, of the MTJ will result in an increase in P_{rev}, the error probability for an individual cell, and in the memory array failure rate, λ. To achieve the benefits of reducing Δ while leaving λ unchanged, some form of error recovery is essential. Despite the overhead of extra bit cells required for ECC and the codec area, we show that the reduction in cell area afforded by decreased thermal stability can yield denser STT-MRAMs.

We consider two scenarios for error correction:

- For smaller memories (e.g., on-chip memories), we perform DRAM-style periodic refreshes, but with error correction, as well as error correction when data is accessed, as discussed in Section II-A2.
- For larger memories (e.g., off-chip memories), error correction is performed only when the data is accessed.

The first case clearly allows for lower retention time specifications than the second, and in both cases, the specification is lower than the case where no ECC is used.

In this work, we use block error correcting codes. In addition to their ability to correct for multibit errors, they function by transforming a fixed-size data block into another fixed-size data block. This is appropriate for memory applications because data widths are constant, and the memory organization is fixed.

An error-correcting code (ECC) can be characterized by the number of symbols prior to encoding, k, the number of symbols after encoding, n, and the number of correctable symbols for the encoded data, c. For binary codes, each bit is a symbol. As c increases, more additional symbols are required for correction, so n increases [13]. The three most well-known options for block codes are Hamming, Bose–Chaudhuri–Hocquenghem (BCH), and Reed–Solomon. Of these, binary Hamming codes are the most simple but are only able to correct a single error. Binary BCH codes are more complicated but can be designed to correct any number of errors. Reed–Solomon can be thought of as BCH codes extended to nonbinary symbols, and afford the ability to correct burst errors with lower overhead because the errors tend to be isolated to a small number of symbols.

We focus on BCH codes because there is need for multibit correction, and, due to the random nature of MTJ free layer reversals, nonbinary symbols are not helpful. Strictly, BCH codes require that $n = 2^a - 1,\ a \in \mathbb{Z}^+$, but by using systematic encoding where the encoded data is the same as the unencoded data with the addition of correction bits, unused bits can be omitted. This technique is known as shortening.

A. Finding the Error Rate with Error Correction

Consider a memory with m blocks, each of length k, with a block size of m/k. Typically, m/k is an integer because data blocks correspond to the fundamental elements of a memory, such as words, lines, and pages. Each such fundamental element consists of k data bits and $(n-k)$ ECC bits that are used to correct up to c errors in the block.

In standby, the error probability, P_{rev}, in an MTJ is given by Eq. (3). The probability of i incorrect bits and $n-i$ error-free bits in an n-bit block is $P_{rev}^{i}(1-P_{rev})^{n-i}$. The probability of error-free operation with up to c correctable errors is:

$$\lambda = 1 - \left(\sum_{i=0}^{c} \binom{n}{i} \left(1 - e^{-\frac{t_r}{\tau_0} e^{-\Delta}}\right)^{i} \left(e^{-\frac{t_r}{\tau_0} e^{-\Delta}}\right)^{n-i} \right)^{\frac{m}{k}} \quad (9)$$

The first term of the summand accounts for the number of ways in which i errors can be manifested within an n-bit block. The probability that all data in the memory is error-free or correctable is the likelihood that all m/k blocks in memory have this property: this accounts for the exponent of m/k. Finally, the formula complements this probability to represent the probability, λ, of one or more data blocks being *uncorrectable*, i.e., the probability of an error in the memory in the presence of error correction.

It is easy to verify that when there is no error correction, $c = 0$, $n = k$, and this equation is equivalent to Eq. (4).

B. Impact of Refresh

As stated in Section II-A2, the the integrity of data can be maintained through periodic refreshes with error correction. This is feasible and worthwhile for on-chip STT-RAM caches since the refresh operation permits a relaxation in the retention times, which translates into increased memory density as Δ is relaxed. Under our ECC scheme, these refresh operations help prevent an uncorrectable number of errors from accumulating. If t_f is the refresh period, the memory failure rate is

$$\lambda = 1 - \left(\sum_{i=0}^{c} \binom{n}{i} \left(1 - e^{-\frac{t_f}{\tau_0} e^{-\Delta}}\right)^{i} \left(e^{-\frac{t_f}{\tau_0} e^{-\Delta}}\right)^{n-i} \right)^{\frac{m t_r}{k t_f}} \quad (10)$$

Here the summation term is the same as in Eq. (9) but represents the probability of a correctable data block for a single refresh period, instead of for the operational time of the chip. Error-free operation implies that there are no errors in any of the (t_r/t_f) periods between refreshes, and this brings about an additional t_r/t_f term in the exponent of the summation term.

Again, this is equivalent to Eq. (4) when $c = 0$, proving that, unlike with DRAM, refreshing has no impact on the memory failure rate in the absence of ECC. When $t_f = t_r$, there is effectively no refresh, and Eqs. (9) and (10) are also identical.

Determining the optimal value of t_f requires memory access characterization, similar to [9], and, consequently, is outside the scope of this work. Our experiments have shown, however, that the optimal memory area achievable by multibit correction at a given refresh rate cannot always be matched by using single-bit correction in conjunction with a higher refresh rate.

```
Input: Initial memory parameters: $\lambda$, $\Delta$, $m$, $k$, $R$, $Area_{cell}$
$c \leftarrow 0$
$Area_{(c)} \leftarrow m Area_{cell}$
$\Delta_{(c)} \leftarrow \Delta$
while $c = 0$ or $Area_{(c)} < Area_{(c-1)}$ do
    $c \leftarrow c + 1$
    Generate and synthesize codec to get $n$ and $Area_{codec}$
    Solve Eq. (10) for $\Delta_{(c)}$
    $Area_{(c)} \leftarrow m \frac{n}{k} Area_{cell} \left(1 - R\left(1 - \frac{\Delta_{(c)}}{\Delta}\right)\right) + Area_{codec}$
end while
$c \leftarrow c - 1$
return $c$, $Area_{(c)}$, $\Delta_{(c)}$
```

Fig. 4. Algorithm for finding the optimal level of correction

Furthermore, even in cases where identical area savings are attainable, the required refresh rate may be unrealistic for a given processor clock rate. For this reason, multibit correction is a valuable option to consider when designing STT-MRAMs.

The error rate analysis discussed thus far does not include the impact of errors caused by the read operations. We assume that for a reliable memory design, this error rate is negligible. However, should this not be the case, the additional error term can be incorporated into the model. The effect of this is that the required thermal stability factors will increase from those calculated using our simplified model, but the relative area savings of multibit ECC should remain similar.

IV. Finding the Optimal Level of Correction

Stronger error correction allows for smaller STT-MRAM bit cells, but there is a tradeoff between the cell area and the area overhead of the codec and the extra bits required for ECC. An appropriate level of correction must be used to minimize the total area of the memory and the codec. A fully analytical solution is precluded since the codec area cannot be obtained as a closed-form function of c and must be obtained by synthesizing the circuit through a standard design flow. Therefore, our approach explores the space of c using a linear search. Other forms of search could possibly be used, e.g., using the monotonicity properties of the MTJ and codec areas, but they could be expensive. The BCH codec becomes increasingly time consuming to synthesize as c increases, and overshooting the value of c may make the search technique costly.

The algorithm in Fig. 4 determines which, if any, level of error correction results in the greatest area reduction relative to a design with no error correction. As the level of correction increases, the correction properties are recalculated through generation and synthesis of the codec, and the thermal stability factor is updated to maintain the error rate. These new values determine the updated combined memory and codec area. If the area is less than it was at the previous step, the procedure continues until increasing c no longer decreases the total area, at which point the locally optimal c has been found.

Experimental results suggest that this local minimum is also the global minimum (e.g., Fig. 5). Proving a global minimum is difficult because there is no closed form equation for the area as a function of c; it has to be calculated at each iteration. A closed form solution of Eq. (10) for Δ in terms of c could only determine the STT-MRAM bit cell area, and the codec area can only be accurately obtained through detailed synthesis.

This algorithm makes a few assumptions. First, it may not be possible to scale w or t, and thus Δ, beyond their initial values

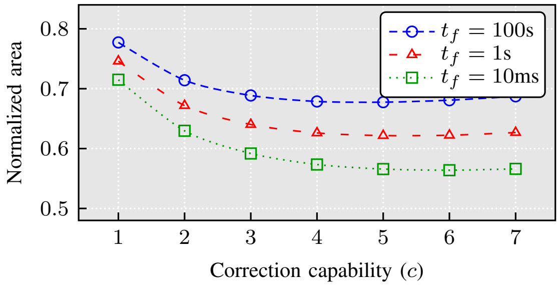


Fig. 5. Total area (STT-MRAM + codec) for cache, normalized to $c = 0$.

or during the recursive step due to technology limitations. A similar limitation exists for W_{TX} in that it cannot scale beyond F. These issues can be avoided by adding conditions to the algorithm to stop it early. Second, it is assumed that W_{TX} can scale continuously. In reality, it may be restricted to integer multiples of F. The algorithm can be modified to increase $\Delta_{(c)}$ such that $W_{TX}\Delta_{(c)}/\Delta$ satisfies this condition. Simulations show that this effect does not have a large impact on the result, and this detail is omitted for simplicity.

V. Results

This section demonstrates the effect of error correction on an example STT-MRAM design, following the methodology described in Section IV.

To estimate the area overhead of the codec, we developed a parametric Verilog generator for shortened systematic BCH codes. Given k and c, this produces the hardware description of the codec components. The designs were synthesized using Synopsys Design Compiler with the NanGate FreePDK45 cell library. The architectures for the encoders and decoders are fully-parallel and unrolled to minimize latency [1], [3]. There are many different types of codes and architectures, some better for specific correction levels than others, but we used a single correction scheme across all levels for simplicity [13].

A. Example STT-MRAM Cache

The effect of multibit correction is evaluated using a 45nm in-plane 1T1MTJ STT-MRAM design based on [10], [12]. For this analysis, a 32Mb last-level cache with three possible refresh periods is utilized, and any error correction is done using 64B line-level encoding. We change Δ from 51 to 67 for the baseline design, representing a failure rate of 1 FIT with no ECC, and the cell dimensions are adjusted accordingly.

We do not consider lower level cache applications because their low latency requirement is incompatible with the assumption in Section II-B that switching occurs on the order of a few nanoseconds. Last level caches, on the other hand, can tolerate relatively higher latencies, and in the case of STT-MRAM, this is actually desirable. By using longer switching pulses, write energy can be drastically reduced [8].

Fig. 5 shows the effect that increasing the level of error correction has on the combined STT-MRAM and codec area. The maximum reduction in area, 44% compared to no ECC and 21% compared to single-bit ECC, is achieved with 6-bit error correction and a 10ms refresh period. However, this reduction in area comes with the penalty of increased latency, as shown in Fig. 6. We show that this latency is quite manageable.

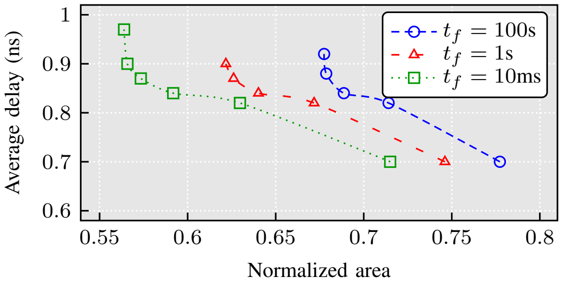


Fig. 6. Decoder delay vs. normalized area. The decode delays are a weighted sum of the detection and correction delays based on the line error rate. The correction level increases as normalized area decreases.

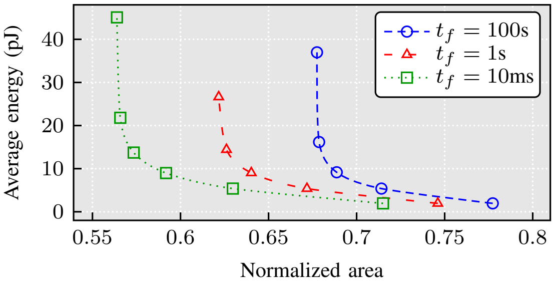


Fig. 7. Decoder energy vs. normalized area. The decode energies are a weighted sum of the detection and correction energies based on the line error rate. The correction level increases as normalized area decreases.

Because correction is only necessary in the presence of an error, the average decoder delay is the sum of the detection delay and the weighted correction delay. The weight used is λ from Eq. (4) with n and t_f substituted for m and t_r, respectively. This weight corresponds to the probability of a cache line requiring correction during a refresh operation. When $c = 6$ and $t_f = 10$ms, there is approximately a 0.94ns delay for detection and a 4.9ns delay for correction. Correction is only necessary for 0.58% of refresh operations, so the average decoder delay is 0.97ns. Because the encoder is much simpler than the decoder, its latency is always lower. At the system level, a $<$1ns overhead in last level cache latency is minor, particularly in cases where the 0.7ns overhead associated with single-bit correction is tolerable.

Fig. 7 shows the average energy overhead of line decoding based on power and timing estimates from Synopsys Design Compiler. Because the correction circuit is only activated when its inputs change, the decoding energy is weighted similarly to as described for delay. When $c = 6$ and $t_f = 10$ms, the average decoding energy is approximately 45pJ.

The energy required to read the all of the bit cells in a cache line is around 90pJ without ECC, based on figures reported in [12], but can be decreased to around 40pJ when $c = 6$ and $t_f = 10$ms. Consequently, the total energy to read a cache line remains the same, despite the decoder overhead. Similar analysis for the write overhead shows that the encoder overhead is mitigated by the decreased thermal stability.

Figs. 6 and 7 also suggest that decreasing the correction factor by one can yield near optimal memory area with significant reduction in codec overhead. For example, using $c = 5$ with $t_f = 10$ms is only 0.3% worse than $c = 6$ in terms of area but 7% better in terms of average codec delay and 52% better in terms of average codec energy.

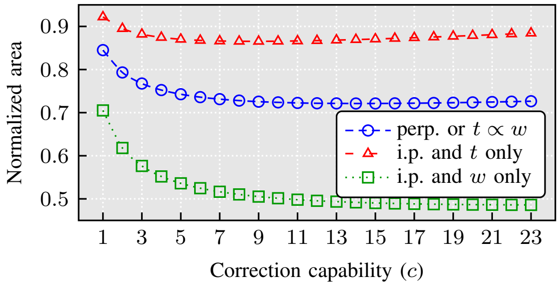


Fig. 8. STT-MRAM area for off-chip memory, not including codec area. The effect of alternative in-plane MTJ scaling schemes is also shown.

B. Off-chip Storage

The methodology described in Section IV is not exclusively applicable to caches. For example, consider a 32Gb memory for off-storage applications, similar to flash. If $k = 4096$ sector-level encoding is used, and $\Delta = 73.9$ (for 1 FIT with no ECC), then a 28% area savings can be achieved with 14-bit correction. The tradeoff in Fig. 8 shows that the potential for area reduction is similar to that in the previous example, albeit favoring stronger codes. Unlike the previous example, we no longer utilize refreshing, as the memory is expected to retain data even without power. By eliminating refresh, the area savings provided by error correction is drastically reduced, but the larger memory size and k value mitigate this effect somewhat. These results do not include the area overhead for the codec because off-chip memory applications typically use much more compact iterative codecs, making the codec area negligible compared to the memory area at the expense of increased latency and energy [4], [16].

C. Effect of Target Error Rate

Until now, we have assumed that the target error rate is 1 FIT, but it is worth considering how things change when this requirement is relaxed or tightened. Experiments show that changing the target error rate has little impact on the overall area reduction achievable relative to the baseline 1 FIT design. However, changing the required error rate changes the correction circuitry activity factor, impacting the average decoder energy for larger values of c. For example, relaxing the requirement to 1000 FIT in the on-chip case reduces the area improvement from 44% to 40% when $c = 6$ and $t_f = 10$ms, but it more than doubles the probability of reading an error, increasing the average decoding energy by 82%.

D. In-plane MTJ Scaling

So far, we have focused on equal scaling of both parameters to ensure the linear relationship between W_{TX} and Δ described in Section II-B for in-plane devices. Fig. 8 shows that only scaling t, making $W_{TX} \propto \sqrt{\Delta}$, reduces the potential area savings by roughly half whereas only scaling w, making $W_{TX} \propto \Delta^2$, roughly doubles it. It is important to keep in mind that this does not apply to perpendicular MTJs because both W_{TX} and Δ are proportional to tw^2.

VI. Conclusion

We have demonstrated that multibit error correction has STT-MRAM area benefits beyond those provided by single-bit correction, both for on-chip caches with refreshing and for off-chip storage. The reduction in nonvolatility afforded by stronger ECC allows for significant reductions in bit cell area that, even when considering the area overhead of the codec, enhances memory density. We also provided a methodology for finding the optimal level of correction required to meet these goals. These results make STT-MRAM even more attractive as a cache technology replacement and are applicable to other levels of the memory hierarchy as well.

References

[1] A. R. Alameldeen et al., "Energy-efficient cache design using variable-strength error-correcting codes," *Proceedings of the 38th Annual International Symposium on Computer Architecture*, pp. 461–472, 2011.

[2] D. Apalkov et al., "Comparison of scaling of in-plane and perpendicular spin transfer switching technologies by micromagnetic simulation," *IEEE Transactions on Magnetics*, vol. 46, pp. 2240–2243, 2010.

[3] H. Burton, "Inversionless decoding of binary BCH codes," *IEEE Transactions on Information Theory*, vol. 17, pp. 464–466, 1971.

[4] H. Choi et al., "VLSI implementation of BCH error correction for multilevel cell NAND flash memory," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 18, pp. 843–847, 2010.

[5] K. C. Chun et al., "A scaling roadmap and performance evaluation of in-plane and perpendicular MTJ based STT-MRAMs for high-density cache memory," *IEEE Journal of Solid-State Circuits*, vol. 48, pp. 598–610, 2013.

[6] Z. Diao et al., "Spin-transfer torque switching in magnetic tunnel junctions and spin-transfer torque random access memory," *Journal of Physics: Condensed Matter*, vol. 19, pp. 165209, 2007.

[7] R. Dorrance et al, "Scalability and design-space analysis of a 1T-1MTJ memory cell for STT-RAMs," *IEEE Transactions on Electron Devices*, vol. 59, pp. 878-887, 2012.

[8] Y. Huai, "Spin-transfer torque MRAM (STT-MRAM): challenges and prospects," *AAPPS Bulletin*, vol. 18, pp. 33-40, 2008.

[9] A. Jog et al, "Cache revive: architecting volatile STT-RAM caches for enhanced performance in CMPs," *Proceedings of the 49th Annual Design Automation Conference*, pp. 243–252, 2012.

[10] J. P. Kim et al., "A 45nm 1Mb embedded STT-MRAM with design techniques to minimize read-disturbance," *VLSI Circuits (VLSIC), 2011 Symposium on*, pp. 296–297, 2011.

[11] Z. Li and S. Zhang, "Thermally assisted magnetization reversal in the presence of a spin-transfer torque," *Physical Review B*, vol. 69, pp. 134416, 2004.

[12] C. J. Lin et al., "45nm low power CMOS logic compatible embedded STT MRAM utilizing a reverse-connection 1T/1MTJ cell," *2009 IEEE International Electron Devices Meeting (IEDM)*, pp. 1–4, 2009.

[13] R. H. Morelos-Zaragoza, *The Art of Error Correcting Coding*, John Wiley & Sons, Ltd, 2006.

[14] S. P. Park et al., "Future cache design using STT MRAMs for improved energy efficiency," *Proceedings of the 49th Annual Design Automation Conference*, pp. 492–497, 2012.

[15] C. W. Smullen et al., "Relaxing non-volatility for fast and energy-efficient STT-RAM caches," *2011 IEEE 17th International Symposium on High Performance Computer Architecture*, pp. 50–61, 2011.

[16] D. Strukov, "The area and latency tradeoffs of binary bit-parallel BCH decoders for prospective nanoelectronic memories," *2006 Fortieth Asilomar Conference on Signals, Systems and Computers*, pp. 1183–1187, 2006.

[17] H. J. Suh et al., "Attempt frequency of magnetization in nanomagnets with thin-film geometry," *Physical Review B*, vol. 78, pp. 064430, 2008.

[18] Z. Sun et al., "Multi retention level STT-RAM cache designs with a dynamic refresh scheme," *Proceedings of the 44th Annual IEEE/ACM International Symposium on Microarchitecture*, pp. 329–338, 2011.

[19] K. Swaminathan et al., "When to forget: A system-level perspective on STT-RAMs," *17th Asia and South Pacific Design Automation Conference*, pp. 311–316, 2012.

[20] R. Takemura et al., "A 32-Mb SPRAM with 2T1R memory cell, localized bi-directional write driver and '1'/'0' dual-array equalized reference scheme," *IEEE Journal of Solid-State Circuits*, vol. 45, pp. 869–879, 2010.

[21] C. Xu et al., "Device-architecture co-optimization of STT-RAM based memory for low power embedded systems," *2011 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pp. 463–470, 2011.

[22] W. Xu et al., "Improving STT MRAM storage density through smaller-than-worst-case transistor sizing," *Proceedings of the 46th Annual Design Automation Conference*, pp. 87–90, 2009.