Memory Reliability for Cells with Strong Bit-Coupling Interference

Kfir Mizrachi
Viterbi Department of EE
Technion - Israel Institute of Technology
Haifa, Israel
kfirster@gmail.com

Ilan Bloom
Viterbi Department of EE
Technion - Israel Institute of Technology
Haifa, Israel
ilanbloom1@gmail.com

Yuval Cassuto
Viterbi Department of EE
Technion - Israel Institute of Technology
Haifa, Israel
ycassuto@ee.technion.ac.il

ABSTRACT

Emerging memory technologies are offering unprecedented storage densities, alongside significant new reliability issues. One such issue this paper addresses is inter-cell interference between coupled pairs of cells. In the studied model there is strong interference between cells, in the sense that programming one cell to a high level changes the level of a second cell significantly. The particular type of interference we study is pair-wise coupling interference: where interference happens between disjoint pairs of cells, so every cell is affected by exactly one other cell.

Our results show that strong coupling interference can be effectively mitigated without need to add large amounts of redundancy beyond the simple Hamming codes common in low-latency memories. One of our techniques is using a soft decoder that can correct many more error combinations thanks to its knowledge of the interference model and parameters. Another technique introduces controlled intentional coupling between the cells at the write path, such that the undesired coupling can be neutralized at the read path with a clever choice of read levels. Overall the two schemes show promising reliability results compared to using the accepted read/write and decoding schemes. The schemes are applicable to a very general class of memories, and thus can help in the deployment of extremely dense emerging storage-class memory technologies that suffer from poor isolation between cells.

CCS CONCEPTS

• **Hardware → Non-volatile memory**; **Error detection and error correction**; *Hardware reliability*; *Analysis and design of emerging devices and systems*; *Memory and dense storage*; • **Information systems →** *Storage class memory*;

KEYWORDS

Inter-cell interference, coupling interference, non-volatile memory, soft decoding, storage-class memories.

MEMSYS US 2017, October 2017, Washington DC

ACM ISBN 123-4567-24-567/08/06…$15.00
https://doi.org/10.475/123_4

ACM Reference format:
Kfir Mizrachi, Ilan Bloom, and Yuval Cassuto . 2017. Memory Reliability for Cells with Strong Bit-Coupling Interference. In *Proceedings of ACM conference, Washington DC, October 2017 (MEMSYS US 2017),* 9 pages.
https://doi.org/10.475/123_4

1 INTRODUCTION

Non-Volatile Memories (NVM) suffer from a multitude of reliability impediments such as: noise, inter-cell interference, read/write disturbs, leakage mechanisms etc. These issues significantly reduce the storage reliability of NVMs. The reliability issues are expected to grow in severity as technology continues to scale aggressively, in particular inter-cell interference will likely become a limiting factor as cells are getting closer together with density.

In mature memory technologies like NAND-flash, inter-cell interference is well understood in the established array architecture. But the emergence of new extremely dense memory technologies calls for studying new and more severe interference regimes. For example, *resistive memory technologies* (also known as reRAM or RRAM) come in array architectures that provide much poorer isolation between cells than current technologies. This happens while low-latency requirements (in the *storage-class memory* paradigm) prevent throwing in powerful error-correcting codes (ECC) to solve the problem. When density is pushed to the physical limits, a first likely effect is introduction of strong coupling interference between *pairs of adjacent cells*. If the strong interference affects all pairs of adjacent cells, it is extremely challenging to achieve reliability at acceptable storage cost and complexity [1]. However, with careful design it is often possible to ensure that interference affects *disjoint pairs* of cells. That is, each cell C has one adjacent cell C' with which it interferes. In one plausible such setup we aggressively reduce the width of half of the inter-column (or inter-row) spaces, and gain significant density advantage with only this limited interference. A proof for the importance of the disjoint-pair coupling interference is its applicability to the existing and ubiquitous memory technology of *mirror-bit NOR Flash* [2]. In this technology density advantage is gained by packing two bits in (two sites of) a single cell, which causes these bits to suffer coupling interference, while bits of adjacent cells remain well isolated.

The results of this paper show that interference in the form of coupled pairs of cells can be managed effectively, and with low associated overheads. Our focus is schemes that use only simple low-order ECC, because the relevant memory technologies need to support small-block access that cannot afford much stronger ECC. Our first scheme uses the technique of *soft-decision decoding* (SDD),

in which our knowledge of the interference model and parameters translates the binary-measurement outcomes to likelihood functions. These likelihoods are then used by an SDD decoder of a standard Hamming code, and are shown to improve bit-error rate (BER) without adding code redundancy. Section 4 shows the SDD results for two setups: a simple one (Section 4.1), where a single static read level is used without shifting it according to the interference parameters, and a realistic one (Section 4.2), where the read level adapts to the interference parameter, and a second read level is added. In the latter more realistic setup the BER improvement of our scheme is more significant. Our second scheme, described and investigated in Section 5, uses intentional (but harmless) coupling in the write process that allows effective mitigation of the interference at read time. The scheme then cleverly uses three read levels, and improves the BER significantly over both the baseline and the previous scheme from Section 4. Here, thanks to the change in the write process we are able to reduce the BER by orders of magnitude compared to schemes that do not use this technique. Toward presenting these schemes we provide in Section 2 the interference model, and detail the SDD algorithm and likelihood calculations in Section 3. These calculations use probability theory to extract from the measurements likelihoods on individual code bits to be used by a SDD, given our knowledge on both cells in the pair and the coupling between them. Our results are general and can apply to any memory technology whose readout is done by binary measurements comparing the cell level to a chosen read level.

The novelty of this study is in being first to apply soft decoding and advanced read/write algorithms to the problem of strong pair-wise coupling interference, and proving that the problem can be mitigated with low-complexity and low-latency ECC schemes required in storage-class/embedded memories. From the circuit and device perspective the schemes only require implementing established techniques in memory design. For example, design of multiple read levels allowing instantaneous read operations is straightforward and known in the literature. By adding each sense amplifier a comparator per read level, chip area and chip power consumption are mildly affected. Similar techniques have been applied in the literature to other interference models in NVMs, most commonly in NAND-flash, but building upon much more powerful ECC schemes than afforded in our application of interest [3–8] (partial list).

2 MODELING STRONG BIT-COUPLING INTERFERENCE

In the scope of this work are memory technologies that suffer from strong bit-coupling interference, that is, the programmed level of one cell has a strong dependence on the bit written to a second cell. The models we propose for strong bit-coupling interference are general, and can be applied to multiple memory technologies and different methods of programming the cell levels. Two models are considered: 1) the *linear model* where the cell level is shifted by a constant α times the level programmed to the second cell, and 2) the *shift model* where the level shifts by a constant a if the second bit is written to $'1'$ (we choose the convention that logical bit $'1'$ is represented by the higher programmed level, but everything works identically if reversed.) The linear model is more physical as it captures the electric properties of the program process; the shift model is a simplification that allows a clearer mathematical analysis. Before specifying the interference models, we discuss cell programming distributions without coupling interference.

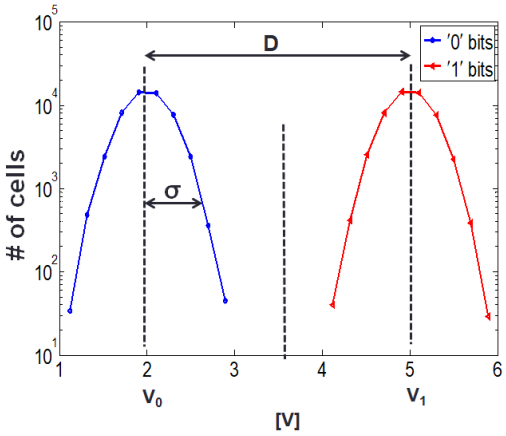

Figure 1: Programming levels as Gaussian distributions (log scale).

2.1 Program-level distributions without coupling

Due to cell variability, a logical bit value written to a cell does not result in a precise program level, but in a *distribution* around a target level. The canonical representation of such a distribution is as a Gaussian random variable with mean μ that equals to the target level, and variance σ^2 that depends on the programming accuracy. To first order the Gaussian model of (1) captures the distributions well.

$$f_{\mu,\sigma}(v) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(v-\mu)^2}{2\sigma^2}} \tag{1}$$

A graphical illustration of the Gaussian model is given in Fig. 1 with two distributions of means V_0,V_1, and equal variance σ^2 (marked on the distribution graph as the standard deviation σ). The Gaussians in Fig. 1 are shown in logarithmic scale, and normalized by the number of cells in the sampled population.

Throughout the paper we assume a read model whereby a *read level* is applied somewhere on the v axis, e.g. in the interval $[V_0, V_1]$, and a bit is returned per each measured cell telling whether the cell level is *above* or *below* the read level. This is the simplest and most common way to sense the programmed levels of non-volatile memories. The natural (and in this simple case optimal) way to choose the read level is as the center point between the two target levels: $(V_0 + V_1)/2$. Then an "above" read leads to the hypothesis of a $'1'$ bit and a "below" read leads to the $'0'$ bit hypothesis. The reliability of the read process depends on the margin $D \triangleq V_1 - V_0$ between the target levels, and on the variance σ^2 of the distributions. If either D is too small or σ is too large, a cell programmed as a bit $'0'$ (resp. $'1'$) may cross the read level to the right (resp. left), and be read erroneously as a $'1'$ (resp. $'0'$). Our knowledge of

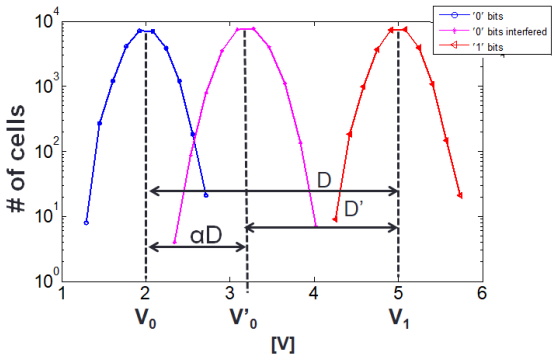

Figure 2: Level distribution with coupling interference in the *linear* model.

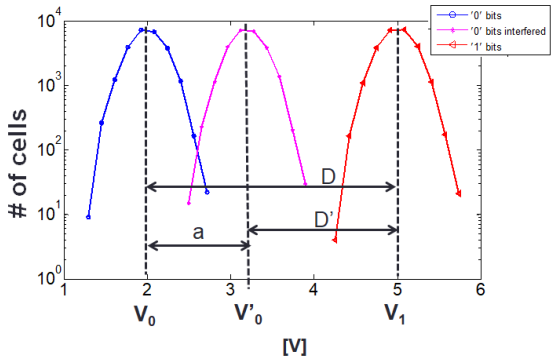

Figure 3: Level distribution with coupling interference in the *shift* model.

the program-level distributions (e.g. by sampling cells from the population) allows us to extract *soft information* from the above read-level measurements, i.e., not just a 1-bit hypothesis of whether the cell was written to $'0'$ or $'1'$, but also the posterior[1] probabilities that the cell is in each of these states.

2.2 Program-level distributions with coupling interference

When each cell is coupled to a second cell, the distributions change accordingly. In essence, the level of a cell with bit $'0'$ is shifted to the right if its second cell is written as bit $'1'$. The way the level shifts depends on whether we assume the linear or shift model described in the beginning of Section 2. Suppose $v0$ and $v1$ are the levels drawn for the $'0'$-bit and $'1'$-bit cells, respectively, in the plain Gaussian model without interference, and define $\Delta v = v1 - v0$. Then in the linear model we have the level of the $'0'$-bit cell

$$v0' = v0 + \alpha\Delta v, \tag{2}$$

and in the shift model we simply have

$$v0' = v0 + a. \tag{3}$$

The level of the $'1'$-bit cell is unchanged as $v1$ from the model without interference. Each model implies a different distribution of the $'0'$-bit cells, which are depicted in Figs. 2 and 3, respectively. We denote by V_0' the center of the distribution of $'0'$ bits interfered by a second cell written to the bit $'1'$. In Figs. 2,3 we normalized the distributions around V_0 ($'0'$ bits not interfered) and V_0' ($'0'$ bits interfered) to be the same height as the distribution around V_1 ($'1'$ bits) for visual appeal – in reality they should be each about half its height, as the $'0'$ population splits roughly half-way between interfered and not interfered. It can be seen that the linear and shift distributions are very similar, only that the linear model gives a slightly wider distribution due to the variance of Δv. We denote by D' the difference between the center of the $'1'$ level and the center of the interfered-$'0'$ level, that is

$$D' = V_1 - V_0' = V_1 - V_0 - \alpha D = V_1 - V_0 - a.$$

[1]posterior=after knowing the measurement outcome.

3 SOFT LEVEL MEASUREMENTS AND SOFT DECODING

To deal with strong bit-coupling interference without adding redundancy, a key tool we use is *soft information* extracted from the read measurements, and its use by the ECC decoder. By soft information we mainly refer to *likelihood functions*: probabilities to observe the read outcome given each of the hypotheses of the cell having been programmed to $'0'/'1'$. Computing the soft information uses the value of the read level, the measurement outcome (above/below), and our knowledge of the level-distributions and coupling-interference parameters. We show in this section how to extract this soft information, and how to use it in decoding the error-correcting code.

3.1 Soft-information probabilities

First let us consider the case without coupling interference. Given Gaussian cell-level distributions $v0 \sim N(V_0, \sigma^2)$ and $v1 \sim N(V_1, \sigma^2)$, a read level V_{RD} induces the following probabilities

$$P_0 = \int_{-\infty}^{V_{RD}} f_0(v)dv,\ P_1 = \int_{V_{RD}}^{\infty} f_1(v)dv,$$

where f_0 and f_1 are the probability density functions (pdf) of the distributions of $v0$ and $v1$, respectively. We use the function $normcdf(v, \mu, \sigma)$ to denote the probability that the Gaussian random variable with parameters μ,σ has a value at most v. Then $P_0 = normcdf(V_{RD}, V_0, \sigma)$ and $P_1 = 1 - normcdf(V_{RD}, V_1, \sigma)$. An example for such P_0 and P_1 is given in Fig. 4 for the case $V_{RD} = (V_0 + V_1)/2$. Define $w = 0$ (resp. $w = 1$) as the event that the cell is programmed to logical bit-value $'0'$ (resp. $'1'$). Also, define $r = 0$ (resp. $r = 1$) as the event that the read outcome is below (resp. above) V_{RD}. Then we have

$$P(r = 0|w = 0) = P_0,\ P(r = 1|w = 1) = P_1,$$

and their complements

$$P(r = 1|w = 0) = 1 - P_0,\ P(r = 0|w = 1) = 1 - P_1.$$

As our soft information we calculate for each cell the likelihood function as the pair

$$P(r|w = 0),\ P(r|w = 1), \tag{4}$$

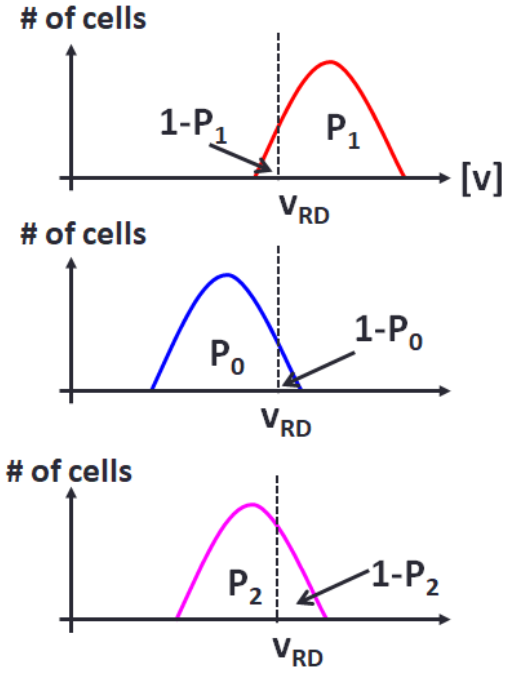

Figure 4: Pictorial illustration of the probabilities P_0, P_1, P_2.

where for r we substitute the binary value (above/below) obtained from that cell's measurement. We refer as a soft-decision decoder (SDD) to any decoding algorithm whose inputs are pairs $P(r|w = 0), P(r|w = 1)$ provided for all codeword bits (note that $P(r|w = 0), P(r|w = 1)$ in general do not sum to 1). If we have the pairs of likelihoods on the codeword bits we can easily get the binary inputs for a hard-decision decoder (HDD) as follows: given r, if $P(r|w = 0)$ is greater than $P(r|w = 1)$, then the input is taken as $'0'$ because it is the more likely written value given the read outcome, and if otherwise the input is taken as $'1'$.

Now let us add the strong-coupling interference. Our objective is to get to likelihoods on the individual code bits similar to (4) (so we can use any existing SDD with these inputs), but now taking into account our knowledge of the coupling between the pairs of cells. Throughout the derivations, we assume the simpler *shift* interference model, where a cell with the logical-$'0'$ bit value has its level shifted upward by a constant a when its second cell has the logical-$'1'$ value. Cells with the logical-$'1'$ value are *not* affected by the second cell. This shift interference introduces another level distribution for cells that are at logical $'0'$ and whose second cell is programmed to logical $'1'$. The level of these cells is distributed according to $N(V_0 + a, \sigma^2)$, which was shown in Fig. 3. For this distribution we define

$$P_2 = normcdf(V_{RD}, V_0 + a, \sigma),$$

and note that $P_2 < P_0$ when $a > 0$, due to the shift of the distribution rightwards. The three probabilities P_0, P_1, P_2 used in the calculation of soft information are illustrated pictorially in Fig. 4.

Table 1: Dependence of read value on write value and second-cell's write value.

r_f	Written values		$P(r_f\|w_f, w_s)$
	w_f	w_s	
0	0	0	P_0
1	0	0	$1 - P_0$
0	0	1	P_2
1	0	1	$1 - P_2$
1	1	*	P_1
0	1	*	$1 - P_1$

It is clear that with the coupling interference the probability to read a cell at outcome r depends on the logical write value of both that cell and the interfering second cell. Denote by w_f the logical bit value of the first cell, and by r_f the binary measurement outcome of that cell. We similarly define w_s and r_s to be these respective values for the second cell. It is immediate to obtain the dependence of r_f on w_f and w_s, which is given in Table 1. The symbol * represents either 0 or 1.

The interesting part now with respect to calculating the soft information is that the likelihood function for a logical bit value w_f is no longer just a function of r_f, w_f as in (4), but it also depends on the value of r_s, which we also have after the read. Explicitly, for soft decoding with coupling interference we need to calculate

$$P(r_f|r_s, w_f = 1),\ P(r_f|r_s, w_f = 0), \tag{5}$$

where we substitute for r_f and r_s the binary read values for the first and second cell, respectively.

3.2 Calculating likelihoods with coupling interference

The technical but important calculation of the likelihoods in (5) now follows. In Table 2 we list the values of the likelihood function for all combinations of r_f, r_s, w_f. We explain here how to derive these values. First it will be convenient to express the likelihood function with additional conditioning on the write value w_s

$$\begin{aligned} P(r_f|r_s, w_f) = P(r_f|w_f, w_s = 0)P(w_s = 0|r_s, w_f)+ \\ P(r_f|w_f, w_s = 1)P(w_s = 1|r_s, w_f), \end{aligned} \tag{6}$$

where we used the fact that r_f is independent of r_s given w_f and w_s. The first term in each product at the right-hand side of (6) can be extracted from Table 1 for any combination of r_f, w_f, w_s. Now to get the second terms, we use elementary probability theory, with the assumption that written logical values are equiprobable $'0'/'1'$, and get

$$\begin{aligned} P(w_s = z|r_s, w_f) = \\ \frac{P(r_s|w_s = z, w_f)}{P(r_s|w_s = z, w_f) + P(r_s|w_s = 1 - z, w_f)}. \end{aligned} \tag{7}$$

We notice that all terms on the right-hand side of (7) can be extracted from Table 1 by exchanging the roles of the first and second cell. From this we get the likelihoods with coupling interference, shown in Table 2.

Table 2: Likelihood values with coupling interference.

r_f	r_s	w_f	$P(r_f \vert r_s, w_f)$
0	0	0	$(P_0^2 + P_2 - P_1P_2)/(1 + P_0 - P_1)$
1	0	0	$(P_0 - P_0^2 + (1-P_1)(1-P_2))/(1 + P_0 - P_1)$
0	1	0	$(P_0 - P_0^2 + P_1P_2)/(1 - P_0 + P_1)$
1	1	0	$(1 - 2P_0 + P_0^2 + P_1 - P_1P_2)/(1 - P_0 + P_1)$
1	*	1	P_1
0	*	1	$1 - P_1$

Table 3: Likelihood values with static read level $V_{RD} = \frac{V_0+V_1}{2}$.

r_f	r_s	w_f	$P(r_f \vert r_s, w_f)$
0	0	0	$P_2(1-P_1) + P_1^2$
1	0	0	$(1-P_1)(1-P_2+P_1)$
0	1	0	$P_1(1-P_1+P_2)$
1	1	0	$1 - P_1(1-P_1+P_2)$
1	*	1	P_1
0	*	1	$1 - P_1$

3.3 Implementing soft decoding

For simplicity and clarity we take the Hamming code as our test case for soft decoding with coupling interference. However, the same techniques can be extended to stronger ECCs. Let $\boldsymbol{c} = (c_0, \ldots, c_{n-1})$ be a binary codeword programmed to a block of n cells. The code dimension (the number of information bits) is denoted k. Let $\boldsymbol{r} = (r_0, \ldots, r_{n-1})$ be the received vector output from the binary read values of the n cells. $\boldsymbol{r}$ may differ from $\boldsymbol{c}$ due to noise and interference, and define the error vector $\boldsymbol{e}$ by $\boldsymbol{r} = \boldsymbol{c} + \boldsymbol{e}$, where summation is element-wise over the binary field (bit-wise exclusive or). For the HDD it is required to estimate the stored bits from the potentially corrupted vector $\boldsymbol{r}$. The first step of HDD is to calculate the syndrome $\boldsymbol{s}$ as follows

$$\boldsymbol{s} = \boldsymbol{r}H^T = (\boldsymbol{c} + \boldsymbol{e})H^T = \boldsymbol{e}H^T, \tag{8}$$

where H is the parity-check matrix of the code, and the arithmetic is carried out modulo 2. We then find the most likely (lowest Hamming weight) error vector $\widehat{\boldsymbol{e}}$ that gives the syndrome $\boldsymbol{s}$, and estimate $\boldsymbol{c}$ as $\widehat{\boldsymbol{c}} = \boldsymbol{r} + \widehat{\boldsymbol{e}}$. HDD does not take into account the coupling interference and will correct only one bit error.

In contrast, SDD factors in the coupling interference through the input likelihoods, and can correct the errors even in cases where more than one bit is flipped in the read values. An SDD with optimal BER is most efficiently implemented with the BCJR algorithm [9] over the code's *trellis*, but a more succinct (and practical for low-rate codes) specification of the optimal-BER SDD is given by [10]

$$\widehat{c}_m = 0 \text{ iff } \sum_{j=1}^{2^{n-k}} \prod_{l=0}^{n-1} \left(\frac{1-\Phi_l}{1+\Phi_l}\right)^{x_l^j \oplus \delta_{ml}} > 0 \qquad \text{and } \widehat{c}_m = 1 \text{ otherwise,} \tag{9}$$

where x_l^j denotes the l-th bit of the j-th word in the dual code; δ_{ml} is 1 iff $l = m$, and Φ_l is the likelihood ratio of the l-th bit. For SDD with coupling interference we substitute

$$\Phi_l = \frac{P(r_l \vert r_{l'}, w_l = 1)}{P(r_l \vert r_{l'}, w_l = 0)}, \tag{10}$$

where in the right-hand side we take the likelihoods from (5) and replace r_f by r_l and r_s by $r_{l'}$, and l' is taken as the index of the cell interfering with cell l.

4 DECODING PERFORMANCE WITH SOFT INFORMATION

After laying out the method to extract soft information and decode with it, in this section we realize this in two realistic scenarios of strong bit-coupling interference: 1) where the read level V_{RD} is statically fixed to the mid-point $(V_0 + V_1)/2$, and 2) with dynamic read level that is shifted given the interference parameter. In both cases we use the (fixed) shift model of coupling interference described by (3).

4.1 Soft decoding with static read level

Static read level is the simplest setup to show the benefits of soft decoding for correcting coupling interference errors. In that it serves as a good introduction for the more realistic dynamic read level setup following next. It is also interesting in its own right for memory technologies that do not allow easy adjustment of read levels. In that setup the read level is set at

$$V_{RD} = \frac{V_0 + V_1}{2},$$

which introduces the symmetry $P_0 = P_1$. This gives the likelihoods in Table 3. We take the $[127, 120]$ single-error correcting Hamming code, and shorten it to carry $k = 64$ information bits. 64 bits give a word size that fits fine-access low-latency applications, which are likely to use emerging non-volatile memories. Hence we get a $[n, k] = [71, 64]$ code. For each of the n code bits we calculate the likelihoods in the numerator and denominator of (10). For each likelihood we use the row in Table 3 whose (r_f, r_s) entry equals $(r_l, r_{l'})$. We decode the resulting likelihood ratios with the SDD algorithm, and compare to the standard HDD for Hamming code. We plot the results in Fig. 5 for a range of the parameter D'/σ (σ is fixed at 0.3 and D' is the independent variable). The results show the BER at the decoder output for both the SDD and HDD, also showing the raw BER without ECC. The SDD is able to improve the BER by significant percentages: at the lowest BER point ($D'/\sigma = 10$) we get improvement from $4.7 \cdot 10^{-4}$ to $3.2 \cdot 10^{-4}$, which amounts to 32% reduction.

Even in this simple setup significant BER improvements were attained simply by invoking a more powerful soft decoder, without increasing the redundancy or the complexity of the read process (no additional read levels were used). This motivates the use of soft decoders in the more realistic setups we study next, aiming at BER values that are more suitable to practical applications.

4.2 Soft decoding with dynamic read level and an auxiliary level

In a more realistic setup, we know the interference parameter a, and are allowed to shift the read level V_{RD} to the mid-point between V_1 and the *shifted distribution* of level '0': $V_0 + a$; V_{RD} is set as the level $(V_0' + V_1)/2$ in Fig. 3. This choice of read level saves many of

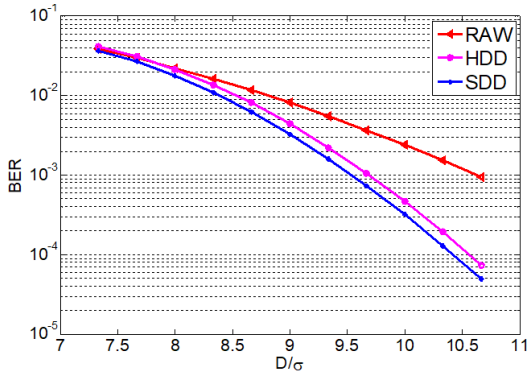

Figure 5: BER of SDD (star markers) with static read level in comparison to HDD (circle markers) and RAW/uncoded (triangle markers).

the interferred first cells from crossing over to be read as logical $'1'$. This can reduce the BER significantly compared to the static read level in Section 4.1. Our method to reduce the BER further with soft decoding is given in Algorithm 1, and described in the following. The key idea is to introduce a second read level for the SDD inputs, but do so only in instances where the HDD cannot correct the errors. For that we now take our ECC to be the $[128, 120]$ extended Hamming code, which is single-error correcting and double-error detecting (SEC/DED); we similarly shorten it to $[n, k] = [72, 64]$. We fix the second read level to be $V_{RD2} < V_{RD}$, and invoke it on all the cells in instances when the HDD of the SEC/DED code detects two bit errors. The rationale behind measuring the cells at a lower V_{RD2} is that knowing that the cell level is further to the left below V_{RD2} helps the soft decoder identify the cell as less likely to have crossed over from $'1'$ to $'0'$. The second read level V_{RD2} is used in instances where the HDD detects a 2-bit error (which it cannot correct). In these cases we invoke the soft decoder with likelihoods given in Table 4. The entries r_f, r_s in the table are obtained from the measurements of the first and second cell, respectively, according to the rule

$$r = \begin{cases} 0, & \text{if } v < V_{RD2} \\ 1, & \text{if } v > V_{RD} \\ 2, & \text{otherwise} \end{cases}$$

The likelihoods in Table 4 are calculated by a straightforward extension of the analysis in Section 3.2 to ternary read values. We use Q_i to denote the probability corresponding to P_i, but with V_{RD} replaced by V_{RD2}.

We plot the results in Fig. 6 for a range of the parameter D'/σ (σ is fixed at 0.3 and D' the independent variable). The results show the BER at the decoder output for both Algorithm 1 and HDD, also showing the raw BER without ECC. Algorithm 1 improves the BER significantly, more so than in the previous setup in Section 4.1. At the lowest BER point ($D'/\sigma = 7$) we get improvement from $2.14 \cdot 10^{-6}$ to $3.89 \cdot 10^{-7}$, close to an order of magnitude. This improvement is much more significant than in Section 4.1, and it is achieved for absolute BER values that are much lower.

Even more impressive than the BER improvement is that the SDD in Algorithm 1 succeeds in correcting the vast majority of 2-bit errors (recall that the extended Hamming code with HDD cannot correct any combination of 2-bit errors). The correction percentages for different values of D'/σ are shown in Fig. 7.

To better understand the performance of Algorithm 1, we want to examine how the BER is affected by the spacing between the read levels: $V_{RD} - V_{RD2}$. For that we vary the spacing between the read levels in the interval $[0.1, 0.5]$, and plot the resulting BER improvement (over HDD) in Fig. 8. It can be seen that the BER improvement reaches its peak at $V_{RD} - V_{RD2} = 0.3$, but also at other choices the BER savings are significant. In general the value that peaks the BER improvement depends on the value of σ.

We note that our choice in Algorithm 1 to run the SDD with two read levels only upon 2-bit error detection is for the purpose of saving the read complexity in the majority of reads that have < 2 errors. We have simulated a variant of Algorithm 1 that runs

Algorithm 1: Soft Decoding with Dynamic Read Levels

input : n cells with coupling interference
output: k estimated information bits
read with V_{RD} and hard-decision decoding:
read n cells with V_{RD}: $r_i = 0$ if $v_i < V_{RD}$ and $r_i = 1$ otherwise
decode $(r_0, \ldots, r_{n-1})$ with HDD
if not detected 2-bit error **then**
  return HDD output
else
  read with V_{RD2} and soft-decision decoding:
  read n cells with V_{RD2}: $r_i = 0$ if $v_i < V_{RD2}$, $r_i = 1$ if $v_i > V_{RD}$, and $r_i = 2$ otherwise
  decode $(r_0, \ldots, r_{n-1})$ with SDD using likelihoods in Table 4
  return SDD output
end

Table 4: Likelihood function of read bit given written bit and read values obtained from two read levels.

r_f	Given data		Probability to read this data
	r_s	w_f	
0	0	0	$\frac{Q_0^2+(1-Q1)Q2}{1+Q_0-Q_1}$
1	0	0	$1-P_2-\frac{(P_0-P_2)Q_0}{1+Q_0-Q_1}$
2	0	0	$\frac{(P_0-Q_0)Q_0+(1-Q_1)(P_2-Q_2)}{1+Q_0-Q_1}$
0	1	0	$\frac{P_1Q_2+(1-P_0)Q_0}{1-P_0+P_1}$
1	1	0	$\frac{(1-P_0)^2+P_1(1-P_2)}{1-P_0+P_1}$
2	1	0	$\frac{(1-P_0)(P_0-Q_0)+P_1(P_2-Q_2)}{1-P_0+P_1}$
0	2	0	$\frac{(P_0-Q_0)Q_0+(-P_1+Q_1)Q_2}{P_0-P_1-Q_0+Q_1}$
1	2	0	$1-P_2-\frac{(P_0-P_2)(P_0-Q_0)}{P_0-P_1-Q_0+Q_1}$
2	2	0	$\frac{(P_0-Q_0)^2+(-P_1+Q_1)(P_2-Q_2)}{P_0-P_1-Q_0+Q_1}$
0	*	1	$1-Q_1$
1	*	1	P_1
2	*	1	Q_1-P_1

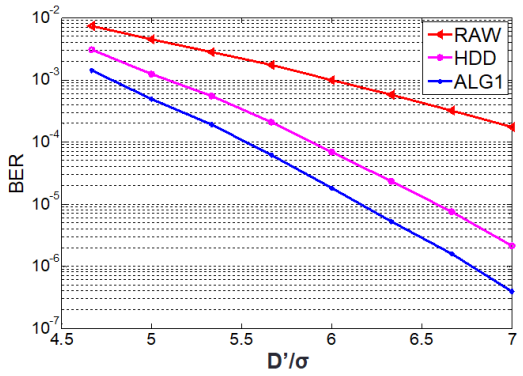

Figure 6: BER of Algorithm 1 (star markers) with $V_{RD2} = V_{RD} - 0.3$ in comparison to HDD (circle markers) and RAW/uncoded (triangle markers).

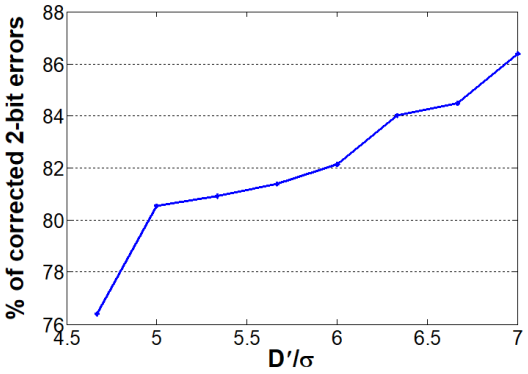

Figure 7: Percentage of correcting 2-bit error combinations by the SDD in Algorithm 1. With the HDD we get 0% for every D'/σ.

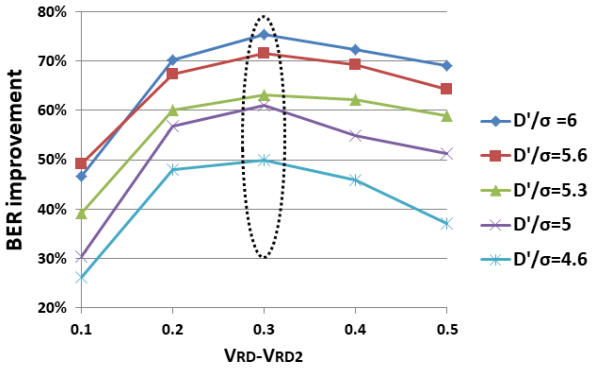

Figure 8: Algorithm 1 BER improvement vs. the difference $V_{RD} - V_{RD2}$.

the SDD in every instance (without first running the HDD for 1-bit correction), and the results are similar and in cases better.

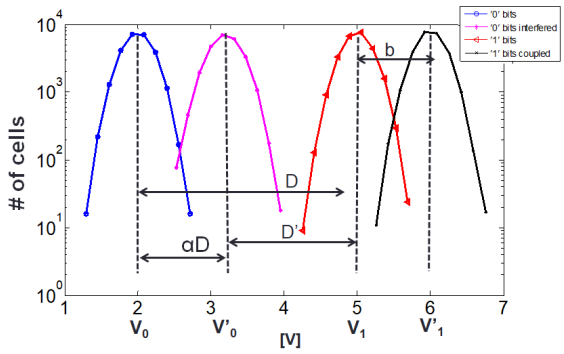

Figure 9: Level distribution with the linear interference model plus intentional shift coupling of $'1'$-bit cells.

5 IMPROVED PERFORMANCE BY COUPLED WRITING AND 3 READ LEVELS

In the fight against coupling interference, we showed in the previous section the power of soft decoding to reduce error rates. Now we propose the ultimative method to combat this type of interference. The key of this method is to add one simple ingredient: *intentional coupling of cell pairs written to* $'1'$. To the undesired coupling we have by the linear/shift models (2)/(3), we add intentional coupling in the form of shifting the level of $'1'$-bit cells when their second cell is also a $'1'$-bit cell. If $v1$ is the level drawn for the cell by the standard distribution $N(V_1, \sigma^2)$, then this shifting results in the level

$$v1' = v1 + b,$$

where b is some known design constant. In addition to this shift of $v1$ we assume the linear[2] interference model for $v0$ as in (2). The resulting cell-level distributions are plotted in Fig. 9. We first observe that this shifting does not add additional interference, because only pairs of cells of both $'1'$-bit are affected. As we show in the remainder of this section, this added coupling leads to effective mitigation of the undesired coupling, reaching extremely low BER levels.

5.1 Decoupled reading with 3 read levels

We now show how to read a cell distributed as in Fig. 9 such that the coupling interference is minimized. Looking at the distributions in Fig. 9 it is not clear how we can read the cell more reliably, and what good it added to couple the $'1'$ cells. However, when seeing the distributions in two dimensions, things become clearer. In Fig. 10 we plot the same distributions of Fig. 9 but adding the second cell in the y-axis. The plot then shows the joint distributions of the two coupled cells according to their logical values: $w_f = 0, w_s = 0$ (left lower), $w_f = 1, w_s = 0$ (right lower), $w_f = 0, w_s = 1$ (left upper), and $w_f = 1, w_s = 1$ (right upper).

Distinguishing between levels originating from $w_f = 0$ and from $w_f = 1$ can be done by using the three read levels marked on Fig. 10. By measuring the first cell with read levels V_{RDa} and V_{RDb}, and the second cell with V_{RDc}, we can dissect the two-dimensional plane

[2]The linear model is more physical than the shift model, so we prefer it when there is no analytic treatment to benefit from the simplicity of the latter.

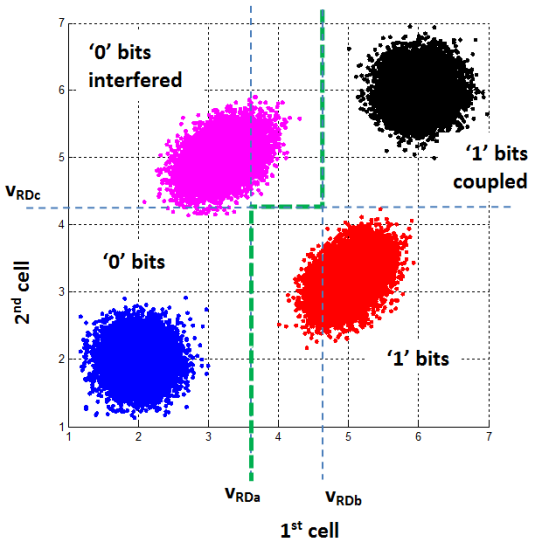

Figure 10: Two-dimensional view of the distributions of coupled cells with shifting the distribution of $('1','1')$ cell pairs.

Table 5: Read rule with 3 read levels $V_{RDa}, V_{RDb}, V_{RDc}$.

	$v_s < V_{RDc}$	$v_s \geq V_{RDc}$
$v_f < V_{RDa}$	0	0
$V_{RDa} \leq v_f < V_{RDb}$	1	0
$v_f \geq V_{RDb}$	1	1

with minimal cross over due to interference. Now it becomes clear why the intentional coupling we introduced is useful: it allows to distance cells with $w_f = 1$ (and $w_s = 1$) from the separating line of V_{RDb}, and thus better distinguishing them from cells with $w_f = 0$ (and $w_s = 1$). Formally, denote by (v_f, v_s) the pair of levels of the (first,second) cells. Then the read rule is taken as specified in Table 5. Note that every cell can be read this way by applying two read levels on it, plus one on its second cell. If the first and second cells are both read in the same memory word, we need total of three measurements of every cell. However, this method has the advantage that it can read the first cells only with the *same total number of measurements* even in setups when the second cells belong to a different memory word. That is, reading the two cells requires in total 6 read levels, whether they are read together ($V_{RDa}, V_{RDb}, V_{RDc}$ in each), or separately (V_{RDa}, V_{RDb} in the read cell and V_{RDc} in its second cell; then reversed for the other word-read).

5.2 BER performance

We now plot the BER results of the decoupled reading scheme with 3 read levels. In Fig. 11 we show the BER results of the proposed scheme in comparison to the standard scheme that shifts the read level to the mid-point of V_0' and V_1 without changing the write. The results of the new scheme without and with (HDD) ECC are shown in the lower two plots. The single read level scheme without and

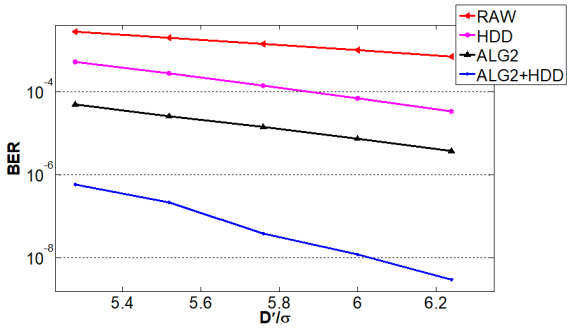

Figure 11: BER of proposed read scheme with shifted write and 3 read levels (two lower curves). Upper curves show the baseline of the standard read/write scheme.

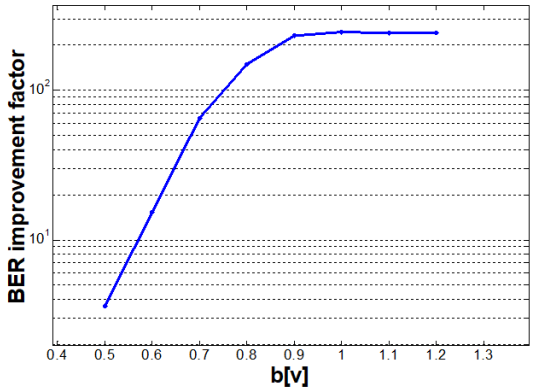

Figure 12: BER improvement factor as a function of the write shifting parameter b applied when the two cells are written to $'1'$, for $D'/\sigma = 5.28$.

with (HDD) ECC is plotted in the upper two curves. It can be seen that the advantage of the new scheme is huge. In particular, even with the simple Hamming code it succeeds in lowering the BER to orders that can be used in commercial products.

5.3 Setting design parameters

To get the best results in this scheme, there are two interesting parameters we should set when we design the memory. Here we want to examine and better understand them. The first parameter is b: the amount of shift we write to the pairs of cells at the $'1'$ value. b has *cost* ramifications, because a high shift implies higher delay/power/wear in the write path. Clearly we want to minimize these costs in a real device. In Fig. 12 we plot the dependence of the BER multiplicative improvement factor (without ECC) as a function of b, for the value $D'/\sigma = 5.28$. We can see that up to about 0.8[V] we gain by increasing b, but from that point further there is no point for higher values. Other values of D'/σ give similar behavior.

The second parameter we examine is the spacing $V_{RDb} - V_{RDa}$. Its value needs to balance between two properties that are useful for low BER. We can see in Fig. 10 that shifting V_{RDa} to the right will

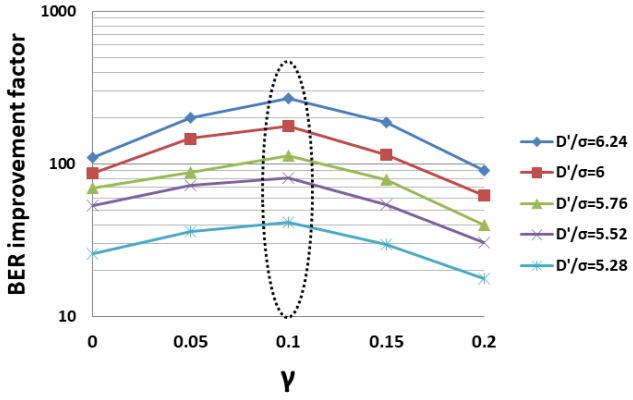

Figure 13: BER improvement factor as a function of the parameter γ setting the spacing between the read levels V_{RDa}, V_{RDb}.

cause cells from the right lower ($'1',' 0'$) distribution to cross to the left. On the flip side, shifting V_{RDa} to the left will cause cells from the left upper ($'0',' 1'$) distribution to cross to the right (near the corner $V_{RDa} \perp V_{RDc}$). The dependence of the BER improvement on the spacing $V_{RDb} - V_{RDa}$ is shown in Fig. 13. We set the baseline values of the read levels as $B_{RDa} = (V_0 + V_1)/2$ (the mid-point of the non-shifted levels), and $B_{RDb} = V_0' + (V_1 - V_0)/2$ (symmetric to B_{RDa} with respect to the mid-point $(V_0' + V_1)/2$). Then we vary the spacing between the read levels as $V_{RDa} = B_{RDa} + \gamma$ and $V_{RDb} = B_{RDb} - \gamma$, where γ is the independent variable in Fig. 13. (This implies that the mid-point between V_{RDa} and V_{RDb} is fixed at $(V_0' + V_1)/2$). The value that maximizes the BER improvement is $\gamma = 0.1$, which corresponds to $V_{RDa} = 3.6$ and $V_{RDb} = 4.6$. These values are marked in Fig. 10. Most importantly, the same γ is optimal for all tested values of D'/σ.

6 CONCLUSION

In this paper we proposed two techniques to reduce the BER with strong cell-coupling interference. The advantage of these techniques is that they work with the accepted architecture and ECC for low-latency memories, and manage to get significant BER reduction by changing the ECC decoder and read/write algorithms. Applying the new techniques in commercial memories does not entail any prohibitive implementation cost. For the first scheme one needs to implement a SDD, which for low-order ECC like Hamming or BCH codes is quite straightforward. For the second scheme one needs to change the cell-program algorithm to shift the levels by b in case both cells are $'1'$. This technique is standard, and has been implemented in commercial technologies like NAND-flash (for different purposes). The cost of this scheme in extra time/power grows gracefully with the parameter b, which can be set as a compromise between performance and cost.

Clearly the results shown here are just the starting point for these schemes showing their promise. In realistic setups involving emerging storage-class memories they will need to be enhanced by complementary techniques to get to the desired BER levels in the target applications. The most interesting future-research direction is how to design a stronger ECC to best combat coupling interference. It is not clear that going the standard direction of correcting more hard errors (e.g. by BCH codes) is the optimal route.

7 ACKNOWLEDGMENT

This work was supported in part by the Israel Science Foundation and by the US-Israel Binational Science Foundation.

REFERENCES

[1] M. Asadi, X. Huang, A. Kavcic and N.P. Santhanam, Optimal detector for multilevel NAND flash memory channels with intercell interference. IEEE Journal on Selected Areas in Communications, Vol.32, No.5, pp.825-835, May 2014.
[2] I. Bloom, A. Givant, E. Lusky, A.Shappir, M. Janai and B. Eitan. NROM/MirrorBit technology for Non-Volatile Memories. Reference module in materials science and materials engineering, Oxford: Elsevier, P.1-10, 2016.
[3] J. Xiao-bo, T. Xue-qing and H. Wei-pei. Novel ECC structure and evaluation method for NAND flash memory. 28th System-On-Chip Conference (SOCC), 2015.
[4] L. Shyue-Kung, Z. Shang-Xiu and H. Masaki. Adaptive ECC techniques for yield and reliability enhancement of flash memories. 25th Asian Test Symposium (ATS), 2016 IEEE.
[5] Y. Xiao, Y. Xie and D. Niu. Low power memristor-based ReRAM design with error correcting code. 17th Asia and South Pacific Design Automation Conference (ASP-DAC), 2012.
[6] W. Zhao and H. Sun. Improving min-sum LDPC decoding throughput by exploiting intra-cell bit error characteristic in MLC NAND flash memory. 30th IEEE Symposium on Mass Storage Systems and Technologies (MSST), 2014.
[7] G. Dong, N. Xie and T. Zhang. Enabling NAND flash memory use soft-decision error correction codes at minimal read latency overhead. IEEE Transactions on Circuits and Systems I: regular papers, Vol.60, No.9, Sept 2013.
[8] C. Argyrides, P. Reviriego and J. Antonio Maestr. Using single error correction codes to protect against isolated defects and soft errors. IEEE Transactions on Reliability, Vol.62, No.1, March 2013.
[9] B. Honary and G. Markarian. Low-complexity trellis decoding of Hamming codes. IEEE Electronics Letters, Vol.29, No.12, June 1993.
[10] A. Ashikhmin and S. Litsyn. Simple MAP decoding of first-order Reed-Muller and Hamming codes. IEEE Transanctions on Information Theory, Vol.50, No.8, August 2004.