Error Correction Improvement based on Weak-Bit-Flipping for Resistive Memories

Valentin Gherman, Lorenzo Ciampolini, Samuel Evain, Sébastien Ricavy

Error Correction Improvement based on Weak-Bit-Flipping for Resistive Memories

Valentin Gherman[1], Lorenzo Ciampolini[1], Samuel Evain[2] and Sébastien Ricavy[3]
[1]Université Grenoble Alpes, CEA, List, F-38000 Grenoble, France
[2]Université Paris-Saclay, CEA, List, F-91120, Palaiseau, France
[3]Université Grenoble Alpes, CEA, Leti, F-38000 Grenoble, France

***Abstract*—Resistive memories are affected by significant error rates tied to structural relaxation and wear out of the resistive memory devices. A way to reduce the need for strong error-correcting codes (ECCs) is to improve error correction based on the *weak* bits, i.e., potentially faulty bits, identified in sensed memory words. Here, it is formally proven that conventional ECC decoders reinforced with weak-bit-flipping may achieve similar error correction capability as theoretical generalized-minimum-distance decoders. It is shown that weak-bit-flipping may reduce the uncorrectable bit error rate (UBER) by orders of magnitude when applied in conjunction with single-error-correcting and double-error-detecting (SEC-DED) or double-error-correcting and triple-error-detecting (DEC-TED) codes. In particular, weak-bit-information extracted from a 2T2R memory and used to reinforce a DEC-TED code with a conventional decoder may enable an UBER that is one order of magnitude better than the UBER achieved with a triple-error-correcting (TEC) code and a conventional decoder.**

Keywords—resistive memories; error rate; error-correcting codes; weak bis; weak-bit-information; erasure-information

I. Introduction

Nowadays, different types of resistive memories compete with each other on parameters like: integration density, access latency, data retention capability, cycling endurance, power consumption, CMOS compatibility, price, etc. Magnetic, phase-change or oxide-based resistive memories are among the most promising ones [7][14][20]. Unfortunately, the relatively low data retention capability is still a limitation that needs to be overcome [7][20]. This challenge can be addressed by promoting the detection and exploitation of *weak bits*, i.e., bits that are susceptible to be erroneously sensed, in order to improve the error correction capability of a conventional ECC decoder [1][4][10]. Fully unreliable weak bits which offer no hint about their original program value are known as *erasures* [2][5][8][9][13][18].

A way to identify weak bits in resistive and flash memories is to perform multiple sense operations with different reference values [8][11][17][19]. The resulting information can be considered as a rudimentary type of the soft-information [1][4][10] usually used to boost the error correction capability of powerful ECCs such as low-density parity-check (LDPC) codes [16][17][19]. LDPC codes necessitate slow iterative decoding which makes them suitable for relatively slow storage systems with page-level access. Here, the focus is on resistive memories deployed at higher levels of the memory hierarchy that require word-level access and low access latency. In this case, a more appropriate error correction approach is to rely on light-weight ECCs that enable the correction of few errors per memory word based on fast non-iterative decoders.

This contribution concerns the error correction improvement that can be achieved by leveraging the weak-bit-information in conjunction with an i-error-correcting and $(i+1)$-error-detecting code and a conventional errors-only decoder, i.e., an ECC decoder that cannot explicitly handle weak bits or erasures. The considered procedure is to look for weak bits only upon indication of an uncorrectable error in a sensed memory word. Once identified, the weak bits are flipped and error correction is resumed [8][11]. It is shown and formally proven that a conventional errors-only decoder reinforced with weak-bit-flipping becomes at least as efficient as an erasures-and-errors or a theoretical generalized-minimum-distance decoder [9][10] if the goal is to correct up to $i+1$ erroneous bits per code word. As long as one cannot guarantee a quasi-complete $(i+1)$-bit error correction, it makes no sense to strive to correct higher rank errors as they remain rarer and, consequently, with a lower impact on the UBER.

To the best of our knowledge, this is the first time when the impact of weak-bit-information on the error correction capability of ECC decoders is estimated based on a mathematical formalism that is independent of the weak-bit-identification scheme.

1T1R and 2T2R memory configurations are briefly presented in section II. The approach to enhance the error correction capability of conventional errors-only decoders based on weak-bit-information is analyzed in section III. Two weak-bit-identification schemes are briefly discussed in sections IV and V. The resulting impacts on the UBER are analyzed in section VI. Latency and logic overheads are reported in section VII and conclusions are drawn in section VIII.

II. 1T1R versus 2T2R

1T1R and 2T2R memory cell configurations are sketched in Fig. 1. 2T2R configurations enable a much lower raw bit error rate (RBER) based on differential encoding where a logical 0/1 value may be encoded by programming a first/second resistor in a high resistance state (HRS) and a second/first resistor in a low resistance state (LRS). In order to visualize the

impact on the RBER, it will be assumed that the LRS, HRS and, implicitly, their difference are normally distributed [14] [20] as shown in Fig. 2 and Fig. 3. Further, assuming that these distributions affect the memory cells in a random way, one can express the RBER with the help of the complementary error function (erfc) according to annex I as follows:

$$RBER_{1T1R} = \frac{1}{2}\mathrm{erfc}\left(\frac{\mu_{HRS}-\mu_{LRS}}{\sqrt{2}(\sigma_{HRS}+\sigma_{LRS})}\right) \quad (1)$$

$$RBER_{2T2R} = \frac{1}{2}\mathrm{erfc}\left(\frac{\mu_{HRS}-\mu_{LRS}}{\sqrt{2(\sigma_{HRS}^2+\sigma_{LRS}^2)}}\right) \quad (2)$$

where *ref* in Fig 2 is considered to be ideally placed at the same number of σ_{LRS} and σ_{HRS} from μ_{LRS} and μ_{HRS}, respectively. In the case of 2T2R cell configurations, the distributions of the resistance differences R_{LRS}–R_{HRS} and R_{HRS}–R_{LRS} are always symmetrical with respect to zero, which acts as an implicit reference value. This symmetry is preserved with the increasing (a) age of the storage devices and (b) data storage time. Thus, in this case, one should not bother anymore with the selection and lifetime evolution of the most appropriate reference value [3]. (1) and (2) also hold when the LRS and HRS are log-normally distributed [14] if one considers the distribution of the electrical resistance logarithm.

The UBER is a metric for the occurrence rate of data errors and represents the number of data errors per bits read [12]. Assuming that each stored word is (a) subject to random errors according to a certain RBER and (b) protected by an *i*-error-correcting code, the UBER that can be achieved with a conventional errors-only decoder can be estimated with the expression below [15]:

$$UBER = \frac{1}{k}\left[1 - \sum_{t=0}^{i}\binom{n}{t} RBER^t (1-RBER)^{n-t}\right] \quad (3)$$

where:

- k and n represent the number of data-bits and total number of bits per code word, respectively;
- i stands for the maximum number of erroneous bits that can be corrected in a code word;
- the expression in square brackets represents the probability of unsuccessful correction of the errors present in a sensed memory word initially programmed as a code word.

RBER and UBER values for binary systematic block SEC, DEC and TEC codes [5] are plotted in Fig. 4. All RBER and UBER values are monotonically decreasing with the increasing σ-distance between the LRS and HRS distributions. This is coherent with the expectation that the UBER is monotonically increasing with the RBER. Thus, for a given ECC and a target UBER, one needs to figure out the maximum tolerated RBER which gives the minimum tolerated σ-distance between LRS and HRS. This minimum may be reached in worn out devices after a maximum data storage time in worst-case conditions. It can be noticed that 2T2R memories enable

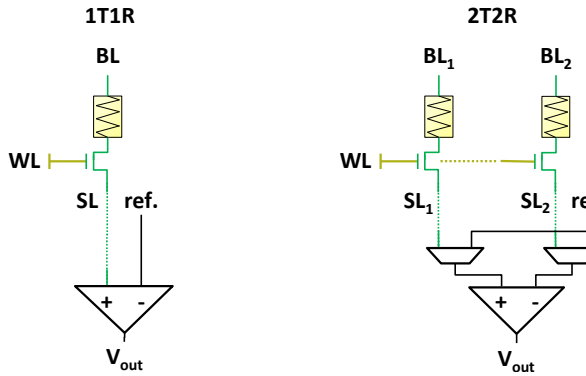

Fig. 1: 1T1R and 2T2R cell configurations with connections to word line (WL), bit lines (BL), source lines (SL) and sense amplifier. In 2T2R cell configurations, a *ref* value is used to verify in 1T1R mode the separation between the LRS and HRS values programed during memory write operations.

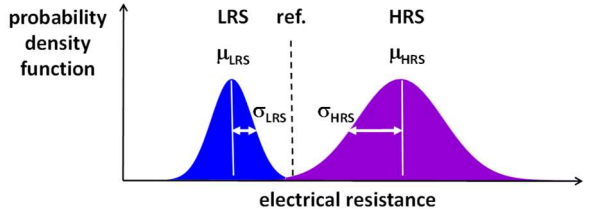

Fig. 2: Distribution of the electrical resistances in low resistance states (LRS) and high resistance states (HRS) for a 1T1R cell configuration with 1 bit per cell.

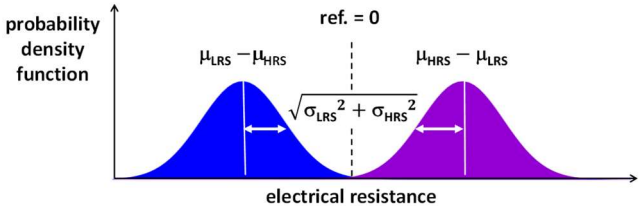

Fig. 3: Distribution of the differences between HRS and LRS used to encode logical values in a 2T2R cell configuration.

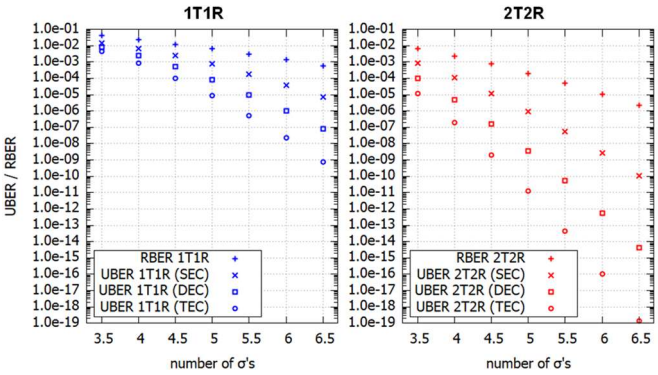

Fig. 4: RBER and UBERs vs. σ-distance (number of σ's) between LRS and HRS for 32-bit 1T1R and 2T2R memories. The σ-distance between LRS and HRS is defined according to annex I.

lower RBERs and UBERs which may result in a better retention capability and, potentially, a better wear out tolerance.

The UBER of a 1T1R memory is independent of the $\sigma_{HRS}/\sigma_{LRS}$ ratio provided that the σ-distance between LRS and HRS on the *x*-axis is calculated according to annex I. The UBERs reported in Fig. 4 for 2T2R memories correspond to a $\sigma_{HRS}/\sigma_{LRS}$ ratio equal to 1. For ratios much larger or much smaller than 1, a 1T1R memory with ideally selected *ref* values becomes equivalent to a 2T2R memory in terms of the UBER as (1) and (2) become equivalent.

For convenience and readability, in the following sections, the LRS and HRS will be assumed to be normally distributed [14][20]. Nevertheless, the proposed methods are not bound to this assumption. For instance, the reported figures can also be applied to 1T1R memories with LRS and HRS subject to log-normal distributions [14] if one considers the distribution of the electrical resistance logarithm.

III. *i*-Error-Correcting and (*i*+1)-Error-Detecting Codes Boosted with Weak-Bit-Information

Improving the UBER by selecting a stronger ECC comes with significant storage overhead, i.e., at least $\lfloor \log_2(k) \rfloor + 1$ additional check-bits per binary code word for each increment of the maximum number of correctable erroneous bits, where k is the number of data-bits per code word. A way to avoid this cost in flash and resistive memories is to (a) identify those memory cells that are in a weak state, i.e., likely to deliver a weak bit during a memory sense operation, and (b) exploit this weak-bit-information to improve error correction.

A noisy channel model equivalent to a binary memory with weak-bit-information is illustrated in Fig. 5. If the weak 1 and 0 values become indistinguishable they can be merged into a single erasure symbol and a conventional binary erasure channel model can be used [6]. According to Fig. 5, the bits sensed from a resistive memory can be classified as *weak* and *non-weak* (*reliable*). Non-weak bits may still be erroneous but their error probability is assumed to be much smaller.

Based on weak-bit-information, the error correction capability achievable with a binary ECC can be improved if, instead of the Hamming distance, one uses a more elaborated definition of the distance between a sensed memory word and an arbitrary code word. According to the *generalized distance* defined in [10], a sensed memory word can be corrected and mapped to a code word if the following relation is fulfilled:

$$2n_{E\cap\bar{w}} + (1-\alpha_w)n_{\bar{E}\cap w} + (1+\alpha_w)n_{E\cap w} < d_H \quad (4)$$

where:

- d_H is the minimum Hamming distance between the code words of the considered binary ECC;
- w and $\bar{w}$ represent events associated to sensing a weak or non-weak bit, respectively;
- E and $\bar{E}$ represent events associated to an erroneously or correctly sensed bit, respectively;

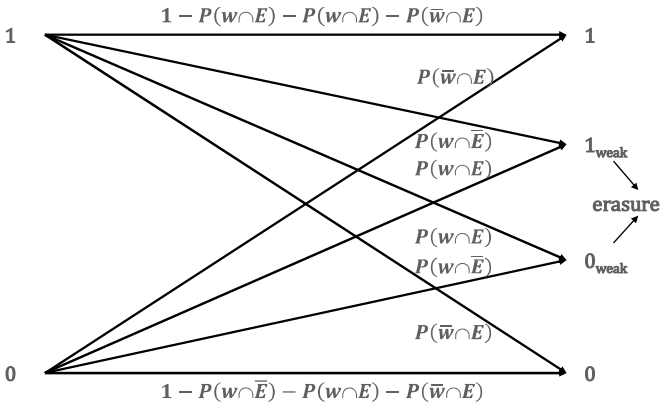

Fig. 5: Equivalent channel model with transition probabilities between programmed and sensed values. *E* and *w* represent events associated to an erroneous bit or an identified weak bit in a sensed memory word, respectively. An overline indicates a complementary event. The symbol '∩' stands for event intersection.

- $n_{E\cap w}$ is the number of erroneous bits identified as weak (true positives) in a sensed memory word;
- $n_{E\cap\bar{w}}$ is the number of erroneous bits not identified as weak (false negatives) in a sensed memory word;
- $n_{\bar{E}\cap w}$ is the number of correct bits identified as weak (false positives) in a sensed memory word;
- α_w is a weight that represents the level of confidence in the sensed value of an identified weak bit.

Here, α_w is defined according to the expression below:

$$\alpha_w = P(\bar{E}/w) - P(E/w)$$

where $P(\bar{E}/w)$ and $P(E/w)$ stand for the conditional probabilities that an identified weak bit is correct and erroneous, respectively. Ways to compute $P(\bar{E}/w)$ and $P(E/w)$ are given in the annexes III and IV. α_w can be approximated to (a) 1 if all weak bits become *non-weak* (*reliable*) so that only the first and last terms on the left-hand side of (4) may be kept and merged and (b) 0 if all weak bits are fully unreliable, i.e., erasures. In the latter case, one may use a so-called erasures-and-errors decoder [9] whose error-correction capability can be evaluated by merging the last to terms on the left-hand side of (4) as follows [2][5][8][9][13][18]:

$$2n_{E\cap\bar{e}} + n_e < d_H \quad (5)$$

where a subscript e is used to represent an erasure instead of a weak bit and n_e is the number of erasures in a sensed memory word ($n_e = n_{E\cap e} + n_{\bar{E}\cap e}$).

The number of errors n_E that can be corrected in a sensed memory word with a generalized-minimum-distance decoder [10] based on (4) can be estimated as follows:

$$2n_E + (1-\alpha_w)(n_{\bar{E}\cap w} - n_{E\cap w}) < d_H \quad (6)$$

where $n_E = n_{E\cap w} + n_{E\cap \bar{w}}$.

Consider now a binary ECC with a conventional errors-only decoder. Further, assume that upon detection of an uncorrectable number of erroneous bits, one can flip all bits identified as weak and resume error correction [8][11]. The number of correctable erroneous bits with an errors-only decoder subsequent to weak-bit-flipping (n'_E) should satisfy the following relation:

$$2n'_E = 2(n_E + n_{\bar{E}\cap w} - n_{E\cap w}) < d_H \quad (7)$$

Relations (6) and (7) can be rearranged as follows:

$$n_{\bar{E}\cap w} - n_{E\cap w} < (d_H - 2n_E)/v \quad (8)$$

where v is equal to $1\text{-}\alpha_w$ and 2, respectively. It is assumed that $0 \le \alpha_w < 1$ so that $0 < 1\text{-}\alpha_w \le 1$.

The right-hand side of (8) becomes zero if one attempts to correct an $(i+1)$-bit error ($n_E = i + 1$) with an i-error-correcting and $(i+1)$-error-detecting code ($d_H = 2i + 2$). In this case, an errors-only decoder with weak-bit-flipping becomes equivalent to a theoretical generalized-minimum-distance decoder [10] or an erasures-and-errors decoder [9]. Restricted use of the weak-bit-flipping, i.e., only in the presence of detectable uncorrectable errors, guarantees the correction of all errors that affect at most i bits per sensed memory word with a conventional errors-only decoder. This proves that a conventional errors-only decoder with weak-bit-flipping is at least as optimal as a theoretical generalized-minimum-distance decoder or an erasures-and-errors decoder as long as one is concerned with the correction of errors that affect at most $i+1$ bits per sensed memory word. As long as the quality of the weak-bit-information does not guarantee quasi-full $(i+1)$-bit error correction, it makes no sense to strive to correct higher rank errors as they remain much rarer and, consequently, with a much lower impact on the UBER.

Besides weak-bit-flipping [8][11], ways to use the weak-bit-information in conjunction with conventional errors-only decoders are forcing all weak bits to either 0 or 1 [21]. Such an approach is deemed as less efficient and is not considered here.

The UBER that can be achieved with (a) an i-error-correcting and $(i+1)$-error-detecting code and (b) a conventional errors-only decoder reinforced with weak-bit-flipping can be evaluated by extending (3) as follows:

$$UBER = \frac{1}{k}\left[1 - \sum_{t=0}^{i}\binom{n}{t} RBER^t (1-RBER)^{n-t}\right] - \\ - \frac{1}{k} P_{weak\_bits/i+1}\binom{n}{i+1} RBER^{i+1}(1-RBER)^{n-i-1} \\ - \frac{R_{i+2}}{k} P_{weak\_bits/i+2}\binom{n}{i+2} RBER^{i+2}(1-RBER)^{n-i-2} \quad (9)$$

where:

- the additional two terms with respect to (3) account for detectable errors affecting $i+1$ and $i+2$ bits in a sensed memory word initially programmed as a code word;
- $P_{weak\_bits/i+1}$ and $P_{weak\_bits/i+2}$ represent the conditional probabilities to have at most i erroneous bits after weak-bit-flipping upon detection of $i+1$ or $i+2$ erroneous bits, respectively, in a sensed memory word, i.e., the resulting new word will contain a number of erroneous bits that can be corrected by a conventional errors-only decoder.
- R_{i+2} stands for the ratio of detectable errors affecting $i+2$ bits which is usually different from zero since:
  - the deployed ECC is imperfect, i.e., not all syndromes are used for the identification of errors affecting up to $i+1$ bits;
  - a conventional errors-only decoder can easily be modified to take advantage of such unused syndromes and enable partial $(i+2)$-bit error detection.
- binary SEC-DED and DEC-TED codes investigated in the following sections enable the detection of $R_3 = 40.7\%$ 3-bit errors and $R_4 = 78.5\%$ 4-bit errors, respectively.

In the case of a SEC-DED code, $P_{weak\_bits/2}$ can be expressed as follows:

$$P_{weak\_bits/2} = [P(\bar{w}/\bar{E})]^{n-2}\{P(w/E)[P(w/E) + 2\,P(\bar{w}/E)]\} + \\ + \{(n-2)\,P(w/\bar{E})[P(\bar{w}/\bar{E})]^{n-3}\}[P(w/E)]^2 \quad (10)$$

where:

- n represents the total number of bits per code word;
- the first term stands for the conditional probability that among $n-2$ correctly sensed bits none is labeled as weak while at least one of the erroneously sensed bits is identified as weak;
- the second term represents the conditional probability that among $n-2$ correctly sensed bits one is indicated as weak while both erroneously sensed bits are signaled as weak;
- $P(w/E)$ and $P(w/\bar{E})$ stand for the conditional probabilities that an erroneously or correctly sensed bit, respectively, is identified as weak;
- $P(\bar{w}/E)$ and $P(\bar{w}/\bar{E})$ are the conditional probabilities that an erroneously or correctly sensed bit, respectively, is not signaled as weak.

Expressions similar to (10) can be derived for the conditional probabilities $P_{weak\_bits/i+1}$ and $P_{weak\_bits/i+2}$ ($i \ge 1$) that occur in (9) based on the conditional probabilities $P(w/E)$, $P(w/\bar{E})$, $P(\bar{w}/E)$ and $P(\bar{w}/\bar{E})$. The latter probabilities depend on the distributions of the involved resistance states and the weak-bit-identification scheme.

According to (9), $P_{weak\_bits/i+j}$ has to be as large as possible in order to minimize the UBER. This requires a maximum probability that an erroneous (correct) bit is (not) indicated as weak, i.e., maximum $P(w/E)$ and ($P(\bar{w}/\bar{E})$. These conditions can be checked on (10) by exploiting the fact that $P(\bar{w}/E)$ and $P(\bar{w}/\bar{E})$ are complementary to $P(w/E)$ and $P(w/\bar{E})$, respectively.

IV. Weak-Bit-Identification in 2T2R Memories based on Additional Sensing in 1T1R Mode

Consider a 2T2R memory cell configuration in which a logical 1 is stored by programming a first resistor (R_1) to LRS and a second resistor (R_2) to HRS as illustrated on the left-hand side of Fig. 6. During a memory write operation, the programmed resistance values may be verified in 1T1R mode against a reference value (*ref*), i.e., one verifies whether R_1<*ref* and R_2>*ref*. The program operations may be repeated if the verifications fail. Subsequently, with increasing storage time, R_1 and R_2 may drift in a way that makes the condition R_1<R_2 invalid and an incorrect logical 0 may be sensed in 2T2R (differential) mode during a memory read operation.

As shown on the right-hand side of Fig. 6, there are three possible scenarios according to which the condition R_1<R_2 becomes invalid. In the first scenario, R_1 and R_2 are still separated by *ref* while in the other two scenarios both R_1 and R_2 are either larger or smaller than *ref*. The latter two scenarios can be identified by sensing R_1 and R_2 separately in 1T1R mode and, potentially, using the same reference value as during the verification of program operations.

Having R_1 and R_2 on the same side of *ref* does not necessarily mean that the encoded bit will be erroneously sensed in differential mode but indicates that either R_1 or R_2 were subject to a significant drift which could result in an erroneously sensed value. On the other hand, errors may remain unveiled despite a significant drift of both R_1 and R_2 according to the first scenario on the right-hand side of Fig. 6.

V. Weak-Bit-Identification based on Sensing with Additional Reference Values

Another way of identifying weak bits in resistive (flash) memories is to compare the electrical resistance (threshold voltage) of a storage device against different reference values [3][8][11][17][19]. Any bit that changes its value when sensed with the additional reference values may be considered as weak. Two additional reference values may be placed around a conventional reference as shown in Fig. 7. The conventional reference may correspond to (a) the *ref*. value shown in Fig. 2 for a 1T1R cell configuration or (b) the implicit zero reference indicated in Fig. 3 for a 2T2R cell configuration. As illustrated in Fig. 8, the additional reference values may be implemented by providing an additional connection to either input of the sense amplifiers in Fig. 1. During a memory sense operation with one of the additional reference values, the connection to each input of the sense amplifier is selected in order to get a resistive bias (Δref.) that corresponds to the difference between the additional and conventional reference values.

The conditional probabilities involved in (10) that correspond to the weak-bit-identification schemes discussed in sections IV and V can be calculated as described in annexes II and III.

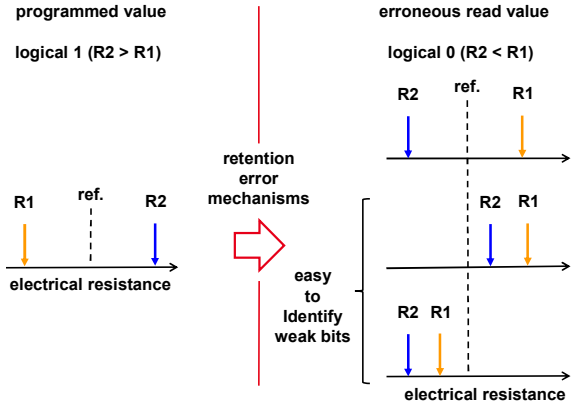

Fig. 6: Retention error mechanisms in a 2T2R cell configuration.

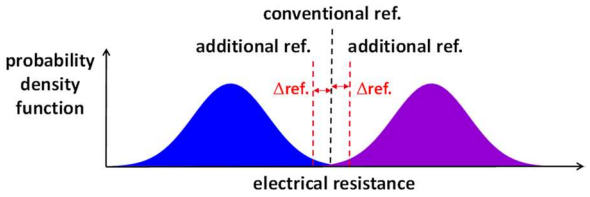

Fig. 7: Weak-bit-identification based on sensing with additional reference values. The logarithm of the electrical resistance may be considered on the X-axis for log-normal distributions of the electrical resistance states.

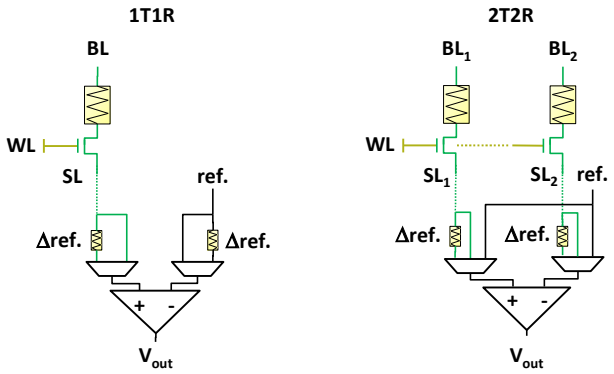

Fig. 8: Possible implementation of sensing with additional reference values as illustrated in Fig. 7.

VI. UBER Improvement

Fig. 9 to Fig. 12 report the impact of the weak-bit-information extracted according to sections IV and V on the UBER that can be achieved with a SEC-DED or DEC-TED code in conjunction with erasures-and-errors, generalized-minimum-distance and conventional errors-only decoders. Weak-bit-flipping is used to reinforce the conventional errors-only decoders and weak bits are assimilated to erasures in the case of erasures-and-errors decoders [9]. The conventional SEC-DED errors-only decoders were considered without partial 3-bit error detection since this feature did not bring a significant UBER improvement. On contrary, the conventional DEC-TED errors-only decoders were upgraded to provide 78.5% 4-bit error detection as enabled by the used parity-check matrix. SEC, DEC and TEC codes are considered as

references only in combination with conventional errors-only decoders and without weak-bit-flipping.

The results estimated for the weak-bit-identification scheme discussed in section IV are shown in Fig. 10 and Fig. 12 under the label *2×1T1R*. As compared to conventional SEC and DEC errors-only decoders, conventional SEC-DED and DEC-TED errors-only decoders reinforced with weak-bit-flipping enabled UBER improvements of more than one decade if the LRS and HRS are separated by at least 5.5×σ and 5×σ, respectively. The UBERs of erasures-and-errors decoders were estimated according to annex IV, but not shown as they were below the UBERs of conventional errors-only decoders without any kind of weak-bit-information.

Better UBER results can be obtained if the weak-bit-information is extracted as explained in section V. For each considered σ-distance between LRS and HRS, the reported UBER figures are the best obtained by varying the values of the additional references illustrated in Fig. 7. As compared to a conventional SEC errors-only decoder, in 1T1R memories, a SEC-DED errors-only decoder boosted with weak-bit-flipping may enable UBER improvements beyond one decade if the LRS and HRS are separated by at least 6×σ according to Fig. 9. For 2T2R memories, this is achieved at 4.5×σ between the LRS and HRS as shown in Fig. 10. The lower σ-distance is due to a better separation of the differential resistance distributions which reduces (a) the RBER and (b) the probability of errors induced by weak-bit-flipping.

With a DEC-TED code, weak-bit-flipping enables a conventional errors-only decoder to achieve UBER values that come close to or even exceed what can be attained with a TEC code as shown in Fig. 11 and Fig. 12. The UBER improvement with respect to a TEC code is due to (a) the correction of quasi-all 3-bit errors together with a non-negligible fraction of 4-bit errors and (b) a reduced number of check-bits which implies shorter memory words.

In all considered cases, erasures-and-errors decoding is outmatched by conventional errors-only decoding reinforced with weak-bit-flipping. Since both approaches have the same probability to correct $(i+1)$-bit errors according to (8), it comes out that erasures-and-errors decoding is unable to achieve full correction of lower rank errors due to the presence of false positives. This is related to the quality of the erasure-information extracted from normal resistance distributions.

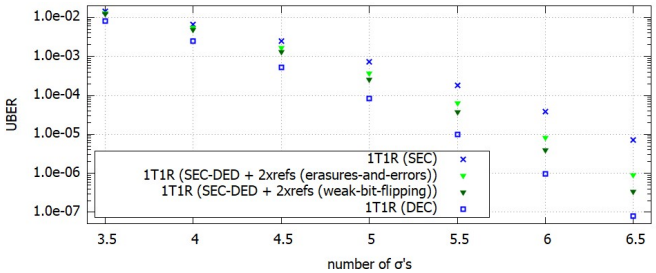

Fig. 9: UBER vs. σ-distance between LRS and HRS for a 32-bit 1T1R memory protected by a SEC-DED code with weak bits identified as shown in Fig. 7.

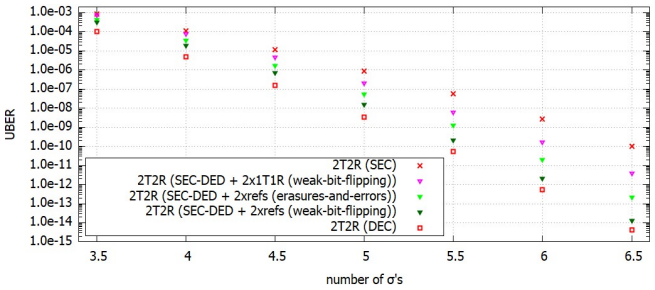

Fig. 10: UBER vs. σ-distance between LRS and HRS for a 32-bit 2T2R memory protected by a SEC-DED code with weak bits identified as illustrated in Fig. 6 and Fig. 7.

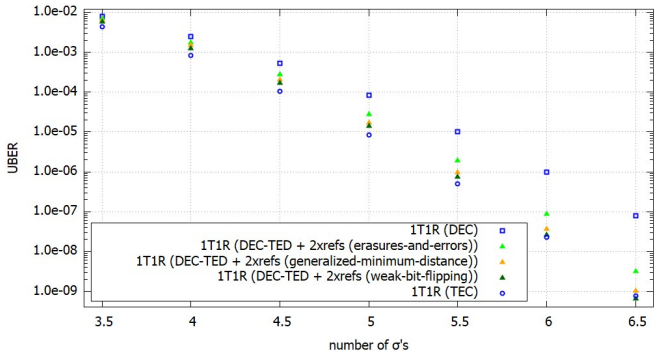

Fig. 11: UBER vs. σ-distance between LRS and HRS for a 32-bit 1T1R memory protected by a DEC-TED code with weak bits identified as shown in Fig. 7.

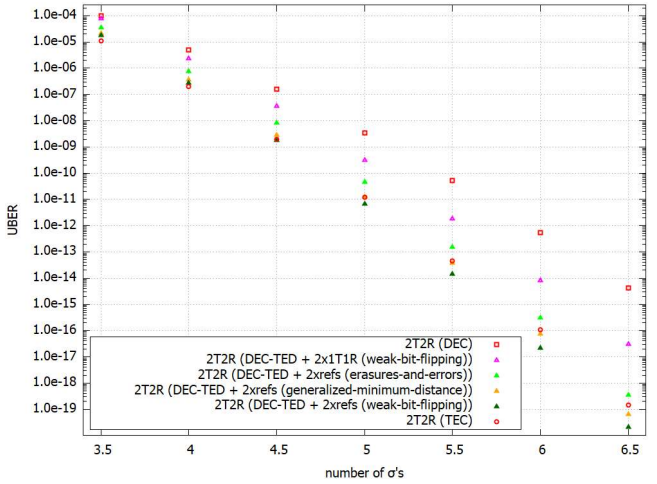

Fig. 12: UBER vs. σ-distance between LRS and HRS for a 32-bit 2T2R memory protected by a DEC-TED code with weak bits identified as illustrated in Fig. 6 and Fig. 7.

Generalized-minimum-distance decoding based on (4) has been evaluated according to annex IV. The resulting UBERs were similar to those obtained based on weak-bit-flipping and conventional errors-only decoders without partial *(i+2)*-bit error detection. This may imply the following:

- the correction of up to *i*-bit errors is guaranteed by generalized-minimum-distance decoding as with conventional errors-only decoding;

- identical ratios of *(i+1)*-bit errors can be corrected with both decoding approaches in agreement with (8);
- in the case of 2T2R memories protected by a DEC-TED code, generalized-minimum-distance decoding was unable to improve the UBER based on 4-bit error correction even if a better UBER could be obtained as compared to a TEC code with conventional errors-only decoding due to a reduced number of check-bits and, implicitly, shorter memory words. The absence of useful 4-bit error correction is due to the fact that:
  - quasi-full *3*-bit error correction is necessary in order to have a noticeable impact of the 4-bit error-correction on the UBER;
  - based on simulation results, quasi-full *3*-bit error correction requires α_w-weights larger than 0.95;
  - according to (4), the maximum number of erroneous bits correctable with a theoretical generalized-minimum-distance decoder is bounded by $d_H/(1+\alpha_w)$ which becomes smaller than 4 when quasi-full *3*-bit error correction is achieved with a DEC-TED code [$d_H/(1+\alpha_w) = 6/(1+0.95) < 4$];

(*i+1*)-bit errors that remain uncorrectable but still detectable after weak-bit-flipping can be detected and signaled by a conventional errors-only decoder. With the weak-bit-identification scheme in section V and a DEC-TED code, virtually all uncorrectable 3-bit errors remain detectable. For 2T2R memories protected by a SEC-DED code, all uncorrectable 2-bit errors remain detectable.

The UBERs reported here correspond to a $\sigma_{HRS}/\sigma_{LRS}$ ratio equal to 1. For normal LRS and HRS distributions and 2T2R memories, the impact of $\sigma_{HRS}/\sigma_{LRS}$ comes only via the influence on the raw bit error rate $RBER_{2T2R}$. For the weak-bit-identification scheme proposed in section IV, this is due to the fact the expressions in annex II only depend on the σ-distance between LRS and HRS while, for the weak-bit-identification scheme discussed in section V, this is due to the symmetry of the distributions in Fig. 3. In this case, the minimum UBER achievable for a 2T2R memory with an arbitrary $\sigma_{HRS}/\sigma_{LRS}$ ratio is equal to the UBER attainable for a 1T1R memory that has (a) a $\sigma_{HRS}/\sigma_{LRS}$ ratio equal to 1 and (b) the same RBER as the 2T2R memory. A more involved study of the $\sigma_{HRS}/\sigma_{LRS}$ impact on the UBER is beyond the scope of this paper.

VII. Overhead

Weak-bit-flipping can be implemented with the help of glue logic placed in front of a conventional errors-only decoder as illustrated in Fig. 13. At the beginning of each read operation, the registers R and R' should be initialized such that a conventionally sensed word arrives unchanged at the inputs of the ECC decoder. In case of an uncorrectable error signaled by the ECC decoder, the conventionally sensed word is stored in R and two additional sense operations are launched according to section IV or V. R' is used to store the first of the two additionally sensed words. Corresponding bits in the additionally sensed words are compared using an *xnor* or *xor* gate depending on whether the weak bits are identified according to section IV or V, respectively. Based on the outcome of each comparison, an *xor* gate is used to flip the corresponding bit in the conventionally sensed word stored in R. Potential errors in the resulted word with flipped weak bits may still be corrected by the ECC decoder.

The considered weak-bit-identification schemes do not require ECC encoder modifications and have no impact on the latency of memory write instructions but, as mentioned, may require supplementary sense operations during memory read instructions. In order to evaluate this overhead, it is assumed that two additional memory sense and ECC decode operations (even if only one additional ECC decode operation is actually needed) are performed each time an uncorrectable error is signaled. As illustrated in Fig. 14.a, the resulting operation overhead is lower for (a) 2T2R memories due to better RBERs and (b) stronger ECCs. There is virtually no operation overhead beyond 4×σ and 5.5×σ between LRS and HRS for a DEC-TED or SEC-DED code, respectively. Relating this to the monotonic evolution of the UBER with the σ-distance between LRS and HRS, one may state that the considered weak-bit-identification schemes have virtually no impact on the

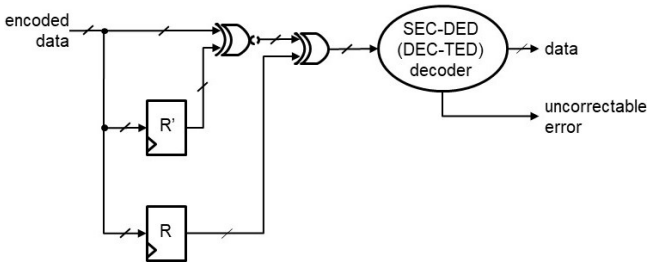

Fig. 13: Glue logic that may be used to implement weak-bit-flipping in front of a conventional errors-only decoder.

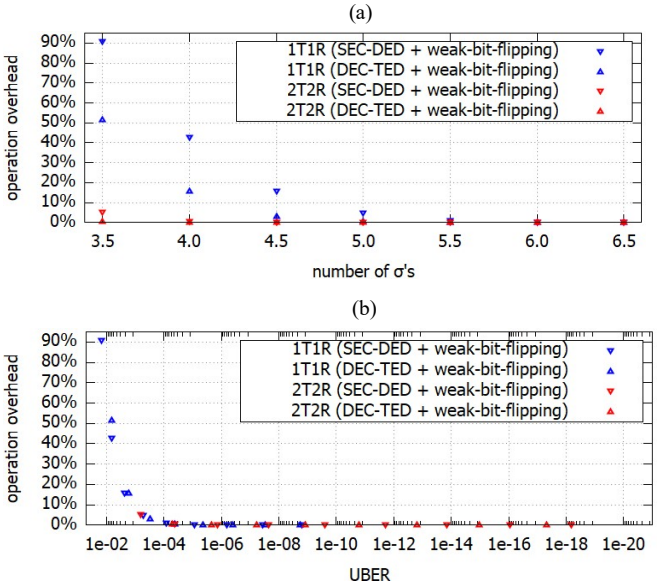

Fig. 14: Memory sense and ECC decode overhead vs. (a) σ-distance between LRS and HRS and (b) achieved UBER.

memory access latency at UBERs relevant to client (UBER ≤ 10^{-15}) or industrial (UBER ≤ 10^{-16}) applications [12] as shown in Fig. 14.b. This is due to the fact that such small UBERs are obtained at very small RBERs so that an uncorrectable error may occur very seldom.

Besides the presence of glue logic, modifications of conventional errors-only decoders may be required to implement partial detection of higher rank errors as in the case of the DEC-TED decoder with partial 4-bit error detection. Synthesis results obtained with a 130nm CMOS technology from STMicroelectronics are reported in Table I. The latency and combinational area overhead is considered with respect to a conventional errors-only SEC decoder without weak-bit-flipping. It comes out that a SEC-DED or DEC-TED code boosted with weak-bit-flipping enable sensible reductions in terms of storage area, latency and combinational area as compared to a DEC or TEC code, respectively.

VIII. Conclusions

This work proves that weak-bit-information enables important UBER improvements in resistive memoires protected by light-weight ECCs with conventional errors-only decoders. For instance, in 2T2R memories, a DEC-TED code reinforced with weak-bit-flipping may provide similar or even better UBERs than a TEC code. In this way, weak-bit-flipping can be used to reduce the storage, latency and logic overhead of error correction in resistive memories.

Reported UBER figures may be directly applied to resistive memories where the electrical resistance states can be assumed to follow a normal distribution. This study can easily be adapted to arbitrary distribution laws of the electrical resistance states as long as corresponding *error functions* can be determined.

TABLE I. Storage Overhead, Decoder Latency and Combinational Area for the Considered ECCs.

ECC for 32 data-bits	check-bits		decoder overhead w.r.t. a SEC code	
	number	storage overhead	latency increase	combinational area
SEC	6	+19%	-	-
SEC-DED	7	+22%	0%	×1
SEC-DED with weak-bit-flipping	7	+22%	+13%	×2
DEC	12	+38%	+38%	×9
DEC-TED	13	+41%	+38%	×14
DEC-TED with weak-bit-flipping	13	+41%	+63%	×16
DEC-TED with partial 4-bit error detection and weak-bit-flipping	13	+41%	+75%	×17
TEC	18	+56%	+100%	×80

A significant outcome of this study is a formal proof that, given an i-error-correcting and $(i+1)$-error-detecting code, a conventional errors-only decoder used in combination with weak-bit-flipping performs at least as good as an erasures-and-errors or a theoretical generalized-minimum-distance decoder as long as the goal is to correct up to $i+1$ erroneous bits per code word.

Annex I

In order to estimate the raw bit error rate $RBER_{1T1R}$ of 1T1R memories, it is assumed that the *ref* value in Fig 2 is ideally placed at the same σ-distance from the LRS and HRS mean values. The following relation should hold:

$$\frac{ref-\mu_{LRS}}{\sigma_{LRS}} = \frac{\mu_{HRS}-ref}{\sigma_{HRS}} \tag{11}$$

which allows to obtain an expression for *ref*. When the assumption of the ideal selection of *ref* is not true, one may pessimistically consider the σ-distance to either μ_{LRS} or μ_{HRS} that results in the worst RBER value.

Assuming normal LRS and HRS distributions, (1) can be obtained by replacing the expression of *ref* obtained according to (11) into either of the complementary error functions (erfc) below:

$$RBER_{1T1R} = \frac{1}{2}\mathrm{erfc}\left(\frac{ref-\mu_{LRS}}{\sqrt{2}\sigma_{LRS}}\right) = \frac{1}{2}\mathrm{erfc}\left(\frac{\mu_{HRS}-ref}{\sqrt{2}\sigma_{HRS}}\right)$$

For 2T2R memories, (2) can be obtained out of (1) if σ_{LRS} and σ_{HRS} are replaced by $\sqrt{\sigma_{HRS}^2 + \sigma_{LRS}^2}$ while μ_{LRS} and μ_{HRS} are replaced by μ_{LRS}-μ_{HRS} and μ_{HRS}-μ_{LRS}, respectively.

For either 1T1R or 2T2R memories, the number of σ's (σ-distance) between LRS and HRS is calculated according to the expression below:

$$\sigma_{LRS\leftrightarrow HRS} = \frac{ref-\mu_{LRS}}{\sigma_{LRS}} + \frac{\mu_{HRS}-ref}{\sigma_{HRS}} = 2\frac{\mu_{HRS}-\mu_{LRS}}{\sigma_{LRS}+\sigma_{HRS}} \tag{12}$$

where *ref* is assumed to be at the same number of σ_{LRS} and σ_{HRS} from μ_{LRS} and μ_{HRS}, respectively.

For 1T1R memories, the variability of the implemented *ref* value can be taken into account by transferring it into the LRS and HRS distributions via the convolution products of the probability density functions of the *ref* variable and the LRS or HRS variable, respectively. In the end, this can be handled as a 2T2R configuration with asymmetric distributions.

Annex II

This annex evaluates the conditional probabilities used in (10) according to the weak-bit-identification scheme discussed in section IV for 2T2R memory cell configurations. The events involved in (10) are the following:

- E stands for the event of erroneously sensing a bit;
- w denotes the event of identifying a sensed bit as weak;

- $\bar{E}$ and $\bar{w}$ are the events complementary to E and w, respectively.

Assuming normally distributed LRS and HRS [14][20], the joint probability of E and $\bar{w}$ can be expressed as follows:

$$P(E \cap \bar{w}) = \frac{\int_{ref}^{\infty} exp\left[-\frac{(x-\mu_{LRS})^2}{2\sigma_{LRS}^2}\right]dx \int_{-\infty}^{ref} exp\left[-\frac{(y-\mu_{HRS})^2}{2\sigma_{HRS}^2}\right]dy}{2\pi\sigma_{LRS}\sigma_{HRS}}$$

$$= \frac{1}{4}\left[erfc\left(\frac{\mu_{HRS}-\mu_{LRS}}{\sqrt{2}(\sigma_{HRS}+\sigma_{LRS})}\right)\right]^2 = \frac{1}{4}\left[erfc\left(\frac{\sigma_{LRS\leftrightarrow HRS}}{2\sqrt{2}}\right)\right]^2$$

under the assumption that *ref* is ideally selected at the same number of σ_{LRS} and σ_{HRS} from μ_{LRS} and μ_{HRS}, respectively, which enables (12) and provides the minimum $P(E \cap \bar{w})$.

The conditional probability $P(w/E)$ can be inferred as below:

$$P(w/E) = \frac{P(E \cap w)}{P(E)} = \frac{P(E) - P(E \cap \bar{w})}{P(E)}$$

where *P(E)* is equal to $RBER_{2T2R}$.

Based on Fig. 2 and (12), the probability *P(w)* can be expressed as follows:

$$P(w) = \frac{2\int_{-\infty}^{ref} exp\left[-\frac{(x-\mu_{LRS})^2}{2\sigma_{LRS}^2}\right]dx \int_{-\infty}^{ref} exp\left[-\frac{(y-\mu_{HRS})^2}{2\sigma_{HRS}^2}\right]d}{2\pi\sigma_{LRS}\sigma_{HRS}}$$

$$= erfc\left(\frac{\sigma_{LRS\leftrightarrow HRS}}{2\sqrt{2}}\right)\left[1 - erfc\left(\frac{\sigma_{LRS\leftrightarrow HRS}}{2\sqrt{2}}\right)/2\right]$$

under the assumption that *ref* is placed at the same number of σ_{LRS} and σ_{HRS} from μ_{LRS} and μ_{HRS}, respectively.

The conditional probability $P(w/\bar{E})$ can be derived based on the Bayes' theorem:

$$P(w/\bar{E}) = \frac{P(\bar{E}/w)P(w)}{P(\bar{E})} = \frac{[1-P(E/w)]P(w)}{1-P(E)}$$

where $P(E/w)$ can be inferred as follows:

$$P(E/w) = \frac{P(w/E)P(E)}{P(w)}$$

Conditional probabilities $P(\bar{w}/E)$ and $P(\bar{w}/\bar{E})$ are complementary to $P(w/E)$ and $P(w/\bar{E})$, respectively.

For other types of distributions, the same approach can be followed provided that appropriate probability density functions are considered.

ANNEX III

This annex evaluates the conditional probabilities used in (10) for the weak-bit-identification scheme discussed in section V. Fig. 15 illustrates the electrical resistance domains and their associated probabilities defined by the conventional and additional reference values with respect to the left-hand side distribution in Fig. 7. Assuming a normal distribution of

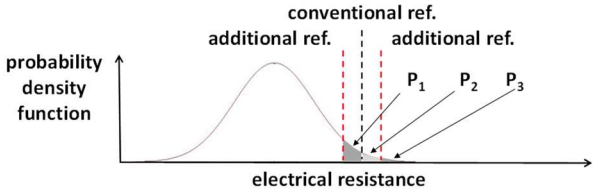

Fig. 15: Graphical representation of the probabilities *P1*, *P2*, *P3* for the left-hand side distribution in Fig. 7.

the resistance value (or of its logarithm) [14][20], the probabilities P_1, P_2 and P_3 can be derived with the help of the complementary error function (erfc) based on the σ-distance of the conventional and additional reference values from the mean value. Similarly, P_1, P_2 and P_3 probabilities can be defined and calculated for the right-hand side distribution in Fig. 7.

Based on P_1, P_2 and P_3, one can compute the following probabilities:

$$P(E) \stackrel{\text{def}}{=} RBER = P_2 + P_3$$

$$P(w/E) = \frac{P_2}{P_2+P_3} = \frac{P_2}{RBER}$$

$$P(\bar{w}/\bar{E}) = \frac{1-P_1-P_2-P_3}{1-P_2-P_3} = 1 - \frac{P_1}{1-RBER}$$

$$P(E/w) = \frac{P_2}{P_1+P_2}$$

Conditional probabilities $P(\bar{w}/E)$, $P(w/\bar{E})$ and $P(\bar{E}/w)$ are complementary to $P(w/E)$, $P(\bar{w}/\bar{E})$ and $P(\mathrm{E}/w)$, respectively.

$P(\bar{w}/\bar{E})$ and $P(w/E)$ need to be as large as possible for a maximum error-correction improvement based on weak-bit-flipping. Consequently, for a given RBER, P_1 (P_2) should be as small (large) as possible. Decreasing (increasing) P_1 (P_2) can be achieved by pushing the additional references in Fig. 15 to the right. Unfortunately, this will have the opposite effect on the right-hand side distribution in Fig. 7. Placing the additional references at different distances from the conventional reference may be considered when the left-hand and right-hand side distributions in Fig. 7 are different or have different σ's.

ANNEX IV

Based on (4), the UBER enabled by erasures-and-errors or theoretical generalized-minimum-distance decoders can be estimated as follows:

$$UBER = \frac{1}{k}\left[1 - \sum_{n_w=0}^{N_w}\binom{n}{n_w}P(w)^{n_w}(1-P(w))^{n-n_w} \times \sum_{n_{E/w}=0}^{N_{E/w}}\binom{n_w}{n_{E/w}}P(E/w)^{n_{E/w}}(1-P(E/w))^{n_w-n_{E/w}} \times \sum_{n_{E/\bar{w}}=0}^{N_{E/\bar{w}}}\binom{n-n_w}{n_{E/\bar{w}}}P(E/\bar{w})^{n_{E/\bar{w}}}(1-P(E/\bar{w}))^{n-n_w-n_{E/\bar{w}}}\right]$$

where:

- d_H is the minimum Hamming distance between the code words of the considered binary ECC;

- k and n represent the number of data-bits and total number of bits per code word, respectively;
- n_w, $n_{E/w}$ and $n_{E/\bar{w}}$ represent the numbers of weak bits, erroneous weak bits and erroneous non-weak bits, respectively;
- N_w, $N_{E/w}$ and $N_{E/\bar{w}}$ are the upper limits of n_w, $n_{E/w}$ and $n_{E/\bar{w}}$, respectively, and can be estimated based on (4) if one replaces $n_{E\cap w}$ and $n_{E\cap \bar{w}}$ by $n_{E/w}$ and $n_{E/\bar{w}}$, respectively;
- $P(w)$, $P(E/w)$ and $P(E/\bar{w})$ are the probability that a sensed bit is identified as weak and the conditional probabilities that a weak or non-weak bit is erroneous, respectively;
- each term of the first addition operator represents the probability of identifying n_w weak bits in a code word;
- each term of the second addition operator stands for the conditional probability of having $n_{E/w}$ erroneous bits among the identified weak bits;
- each term of the third addition operator stands for the conditional probability of having $n_{E/\bar{w}}$ erroneous bits among the non-weak bits;
- if the weak bits are assimilated to erasures, the second addition operator is equal to 1 as α_w is equal to 0 and the upper limit $N_{E/w}$ becomes equal to n_w.

REFERENCES

[1] M. Balser and R.A. Silverman, "Coding for constant-data-rate systems-Part II. Multiple-error-correcting codes," Proc. IRE, vol. 43, 1955, DOI: 10.1109/JRPROC.1955.278085.

[2] E. R. Berlekamp, "Algebraic Coding Theory," McGraw-Hill Book Co., Inc., New York, 1968.

[3] Y. Cai et al., "Data retention in MLC NAND flash memory: characterization, optimization, and recovery," International Symposium on High Performance Computer Architecture, 2015, pp. 551–563, DOI: 10.1109/HPCA.2015.7056062.

[4] D. Chase, "A class of algorithms for decoding block codes with channel measurement information," IEEE Trans. on Inf. Theory, vol. IT-18, no. 1, Jan. 1972, DOI: 10.1109/TIT.1972.1054746.

[5] C.L. Chen and M.Y. Hsiao, "Error-correcting codes for semiconductor memory applications: a state of the art review," Reliable Computer Systems - Design and Evaluation, Digital Press, 1992, pp. 124–134, DOI: 10.1147/rd.282.0124.

[6] P. Elias, "Coding for two noisy channels," in Information Theory, 3rd London Symposium, 1955, pp. 61–76.

[7] Y. Emre, C. Yang, K. Sutaria, Y. Cao and C. Chakrabarti, "Enhancing the reliability of STT-RAM throught Circuit and System Level Techniques," IEEE Workshop on Signal Processing Systems, 2012, pp. 125–130, DOI: 10.1109/SiPS.2012.11.

[8] S. Evain, V. Savin and V. Gherman, "Error correction schemes with erasure information for fast memories," JETTA, volume 30, issue 2, April 2014, pp. 183–192, DOI: 10.1007/s10836-014-5440-1.

[9] G.D. Forney, Jr., "On decoding BCH codes," IEEE Trans. on Information Theory, vol. IT-11, pp. 549-557, October 1965, DOI: 10.1109/TIT.1965.1053825.

[10] G.D. Forney, Jr., "Generalized minimum distance decoding," IEEE Trans. on Information Theory, vol. IT–12, no. 2, 1966, pp. 125–131, DOI: 10.1109/TIT.1966.1053873.

[11] B. Godard, J.-M. Daga, L. Torres and G. Sassatelli, "Evaluation of Design for Reliability Techniques in Embedded Flash Memories," IEEE Design Automation And Test in Europe, 2007, pp. 1593-1598, DOI: 10.1109/DATE.2007.364529.

[12] JEDEC Standard, "Solid-state drive (SSD) requirements and endurance test method," JESD218A, February 2011.

[13] M.N. Kaynak, P.R. Khayat and S. Parthasarathy, "Classification Codes for Soft-information Generation from Hard Flash Reads," IEEE Journal on Selected Areas in Communications, vol. 32, no. 5, May 2014, pp. 892–899, DOI: 10.1109/JSAC.2014.140509.

[14] B.Q. Le, A. Grossi, E. Vianello, T. Wu, G. Lama, E. Beigne, H.-S. P. Wong and S. Mitra, "Resistive RAM with multiple bits per cell: array-level demonstration of 3 bits per cell," IEEE Trans. on El. Dev., vol. 66, issue 1, January 2019, DOI: 10.1109/TED.2018.2879788.

[15] N. Mielke, T. Marquart, N. Wu, J. Kessenich, H. Belgal, E. Schares, F. Trivedi, E. Goodness and L.R. Nevill, "Bit error rate in NAND flash memories," IEEE Annual International Reliability Physics Symposium, 2008, pp. 9–19, DOI: 10.1109/RELPHY.2008.4558857.

[16] S. Ning, "Improving resistive RAM hard and soft decision correctable BERs by using improved-LLR and reset-check-reverse-flag concatenating LDPC code," Trans. on Circ. and Syst.—II: Express Briefs, vol. 67, no. 10, Oct. 2020, DOI: 10.1109/TCSII.2019.2960484.

[17] B. Peleato, R. Agarwal, J.M. Cioffi, M. Qin and P.H. Siegel, "Adaptive Read Thresholds for NAND Flash," IEEE Trans. on Communications, vol. 63, no. 9, Sep. 2015, pp. 3069–3081, DOI: 10.1109/TCOMM.2015.2453413.

[18] W. W. Peterson and E. J. Weldon, Jr., "Error Correcting Codes," 2nd edition, MIT Press, Cambridge, MA, 1972.

[19] J. Wang, T. Courtade, H. Shankar, R.D. Wesel, "Soft-information for LDPC decoding in flash: mutual-information optimized quantization," Global Telecom. Conf., 2011, DOI: 10.1109/GLOCOM.2011.6134417.

[20] C. Yang, Y. Emre, Y. Cao and C. Chakrabarti, "Improving reliability of non-volatile memory technologies through circuit level techniques and error control coding," EURASIP Journal on Advances in Signal Processing, 2012:211, 2012, DOI: 10.1186/1687-6180-2012-211.

[21] C. Zhang and T. Yoshihara, "Word error control algorithm through multi-reading for NAND Flash memories," IEEE International Conference on ASIC, 2011, DOI: 10.1109/ASICON.2011.6157165.