Replacing Linear Hamming Codes by Robust Nonlinear Codes Results in a Reliability Improvement of Memories

Zhen Wang, Mark G. Karpovsky, Konrad J. Kulikowski
Boston University
Reliable Computing Laboratory
8 Saint Mary's Street,Boston, MA, USA
{*lark,markkar,konkul*}*@bu.edu*

Abstract

Linear SEC-DED codes used for protection of memories cannot detect and also miscorrect many errors with large Hamming weights. As multiple bit upsets become more probable for new technologies the reliability of memories protected by linear error correcting codes (ECC) can not be guaranteed. In this paper we propose to protect memory devices using a class of *nonlinear* SEC-DED codes called extended Vasil'ev codes. These nonlinear codes have a minimum distance of four and have fewer undetectable errors as well as errors that are always miscorrected than linear codes with the same parameters. The extended Vasil'ev codes can provide for higher reliability in the presence of repeating errors or high MBU rate with relatively low hardware overhead. The proposed approach for design of reliable memories can be applied to nearly all kinds of memories such as RAM, ROM, FLASH and disk memories.

1 Introduction

Memories play an important role in today's system-on-chip(SOC) designs. According to [30] and [34], 70% of the chip area in many of today's processors is taken by embedded memory and this number is expected to reach 90% by 2011. As memory occupies a big percentage of the area on a chip, it is especially vulnerable to single-event-upset (SEU) caused by single, energetic particles like high-energy neutrons and alpha particles. SEU temporarily alters the state of the devices and results in soft errors, which are non-destructive and appear as unwanted bit flips in memory cells and registers. With continuous scaling, SEU becomes more probable and the soft error rate increases. As the speed of the devices becomes higher the relative size of the clock transition timing window increases, which makes devices more sensitive to SEU [14]. The decrease of applied voltage also raises the soft error rate because bit inversion is more likely to occur when the electrical charge stored in the memory cell is low due to an decrease in the voltage [8].

Linear single error correction, double error detection codes (SEC-DED) are used in modern computer systems as a countermeasure against soft errors to increase the reliability of the system. These codes have Hamming distance 4 and are able to correct all single bit errors and detect all double bit errors. In the presense of multiple errors, however, the reliability of systems utilizing error protection schemes based on linear codes may be questionable. For any linear (n, k) SEC-DED error correcting codes, the number of undetectable multiple errors is 2^k. In addition to this, a huge number of multiple errors will be miscorrected as single bit errors. Since in many cases SEU results in multiple bit distortions, these codes may not be sufficient to provide for a high reliability.

In fact, anomaly of systems caused by multiple bit upset (MBU) was already reported. In [32], it was shown that the Cassini Solid-State Recorder was experiencing a very high rate of uncorrectable multiple bit errors. The author concluded that the MBU rate was architecture dependent and any architecture of DRAM-based designs must be scrutinized carefully to avoid unexpected high MBU rate. In [29], the reliability of systems protected by two types of single error correcting codes was examined. The error rate was reported and the author concluded that conventional ECC may not be sufficient to protect devices against multiple bit errors for certain configuration patterns.

The increase of MBU rate in deep submicron technologies deteriorates the situation even further. In 65nm triple-well SRAMs with a thin cell architecture, the rate of multiple bit errors caused by neutron induced SEU increases by a factor of 10 compared with that in 90nm technologies – nearly 55% of the errors due to neutron radiation were multiple bit errors [11]. Although there are mechanisms like bit interleaving [21] that can be used to minimize the error rate contribution of multiple bit errors, whether it is enough under such high MBU rate is

still unknown. Moreover, the advantage of bit interleaving comes at a price of more layout constrains, which may result in larger power consumptions and longer access times. Thereby, memory protection schemes which can provide better protection against multiple bit errors than that based on classical linear codes are in demand.

In this paper we propose a technique for design of reliable memories based on nonlinear SEC-DED partially robust codes. These codes have fewer undetectable errors and fewer multibit errors which are always miscorrected while requiring similar hardware overhead as the conventional linear SEC-DED codes. We propose that linear extended Hamming codes can be replaced by nonlinear extended Vasil'ev SEC-DED codes described in Section 4 resulting in improved reliability in the presence of multibit distortions.

The rest of the paper is organized as follows. In Section 2, previous work on error correcting codes for memory protection will be summarized. In Section 3, definitions of robust codes are given. In Section 4, we describe the construction methods for robust and partially robust codes with minimum distance. The error detection kernels of different codes are compared and the reason why the extended Vasil'ev code is selected is explained. In Section 5, the architecture utilizing extended Vasil'ev codes is described and the error correcting algorithm is shown in detail. The hardware overhead and error correcting and detecting properties of the extended Vasil'ev code are compared with the extended Hamming codes to demonstrate the advantage of the proposed approach.

2 Previous Work

Since the basic construction of SEC-DED codes was presented by Hamming in 1950 [12], a number of modifications have been proposed. In [13], a class of optimal minimum odd-weight-column SEC-DED codes was constructed for better performance, cost and reliability. To further simplify the encoding and decoding complexity, the author in [27] proposed a coding technique requiring less 1's in the parity check matrix than the code presented in [13]. In [2], a hardware efficient method has been proposed to construct SEC-DED-AUED systematic codes that can also detect all unidirectional errors. For protecting byte oriented memories, SEC-DED-SBD codes are proposed in [28], [4] and [6]. These codes are known as single-error-correcting, double-error detecting, single-byte-error-detecting codes and are able to detect all single byte errors. SEC-DED-SBD codes that are also able to correct any odd number of erroneous bits per byte were proposed in [23]. To enhance the error correction capability of SEC-DED codes, the author in [7] constructed single-error-correcting, double-error-detecting, double-adjacent-error-correcting (SEC-DED-DAEC) code by selectively avoiding certain types of linear dependencies in the parity check matrix. These codes use the same number of check bits and similar overhead to the other SEC-DED codes and have the advantage that it can correct all adjacent double errors. In [5], the author constructed single-byte-error-correcting, double-byte-error-detecting codes (SBC-DBD) which can provide complete single byte error correction capabilities. In [20], double-error-correcting and triple-error-detecting code was proposed to correct all double bit errors. The well known Reed-Solomon code, as another example, was utilized in Hubble Space Telescope to protect 16 Mbit DRAMs manufactured by IBM [36].

All the codes mentioned above are classical linear codes. They concentrate their error detection and correction capabilities on a specific type of errors (i.e. errors with small multiplicities or belonging to the same byte). The reliability of the system can not be guaranteed when MBU rate is high.

Robust codes have been proposed as solution to the limitation of minimum distance linear error detecting codes for detection of fault injection attacks [16]. These nonlinear codes are designed to provide equal protection against all errors thereby eliminating possible weak areas in the protection that can be exploited by an attacker. Several variants of robust codes have been used to protect both private and public cryptographic algorithms. These variants allow several tradeoffs in terms of robustness and hardware overhead for many architectures. Robust, partially robust, and minimum distance partially robust codes have been used for the protection for both private [15] [16] and public key cryptosystems [10].

Robust and partially robust codes with minimum distance larger than 2 are able to correct errors with small multiplicities and are promising alternatives to linear error correcting codes in applications where protection against multiple bit errors is important. These codes have smaller number of undetectable and miscorrected multiple errors than traditional linear error correcting codes (Section 4 and Section 5.2). We will overview several constructions for these codes in Section 4.

3 Definitions

Throughout the whole paper, we denote by "+" the componentwise addition of binary vectors.

Definition 3.1 *(**Kernels of the code**) For any error correcting code $C \subseteq GF(2^n)$, the **detection kernel** K_d is the set of errors that are masked for all codewords.*

$$K_d = \{e | e + c \in C, \forall c \in C\}. \quad (1)$$

It is easy to show that K_d is a linear subcode of C and if C is linear then $K_d = C$. Denote by A the correction

algorithm for code C and E the set of errors that A attempts to correct. The ***correction kernel*** K_c *is the set of errors outside E which have the same result of A as some* $e^{'} \in E$ *for all codewords.*

$$K_c = \{e | e \notin E, \forall c \in C, \exists e^{'} \in E, A(e,c) = A(e^{'},c)\}. \quad (2)$$

The detection kernels of different codes will be analyzed and compared in this section. The correction kernel, which is related to the error correction algorithms of the code, will be discussed in Section 5.2.

Example 3.1 ***(Kernels of Linear Hamming Codes)*** *A* $(n, n - \lceil log_2(n+1) \rceil, 3)$ *linear Hamming code* $C \subseteq GF(2^n)$ *has minimum distance 3 and is able to correct all single bit errors. Denote by H the parity check matrix of C. An error e is undetectable if and only if e is a codeword (*$He = 0$*). Thereby the detection kernel* K_d *of a Hamming code is C itself. For single error correcting codes* $E = \{e \big| ||e|| = 1\}$*, where* $||e||$ *is the number of ones in e. A multiple error* $e, ||e|| > 1$ *will be miscorrected if and only if it has the same syndrom as some single bit error. So the correction kernel of Hamming code is* $\{e | He = He_i\}$*, where* e_i *is an error vector with only one 1. Obviously,* K_d *and* K_c *are disjoint. For perfect linear Hamming code,* $K_d \bigcup K_c \bigcup E = GF(2^n)$.

A main characteristic of traditional linear error detecting codes is that they concentrate their error detecting power on a small subset of errors which are assumed to be the most likely to occur. Typically, such codes concentrate their error detection on errors of a small multiplicity. They are designed to guarantee detection of all errors with a multiplicity less than d. Error detection beyond the minimum distance of the code is typically not a part of the design criteria and can be unpredictable and ineffective. While for some classes of errors the codes provide 100% protection, for a very large class of errors linear codes offer no protection for all messages. In another word, traditional linear error detecting codes have large K_d.

Robust codes, on the other hand, are designed to provide for a guaranteed level of detection against all errors. These codes are characterized by their error masking probability $Q(e)$, which is the fraction of codewords that mask each error.

$$Q(e) = \frac{|\{c | c \in C, c + e \in C\}|}{|C|}. \quad (3)$$

Definition 3.2 *The code C is* ***robust*** *if* $\max_{e \neq 0} Q(e) < 1$ *or equivalently the detection kernel of the code contains only the zero vector* $K_d = \{0\}$.

Robust codes have no undetectable errors. In general, traditional robust codes do not have a minimum distance larger than one and can not be used to do error corrections. A possible variant of the traditional robust codes is to include a minimum distance into the design criteria.

Definition 3.3 *Let* $||e||$ *denote the multiplicity of an error e. A robust code where* $Q(e) = 0$ *for all errors where* $||e|| < d$ *is a* ***d-minimum distance robust code****.*

Minimum distance robust codes have no undetectable errors and the worst case error masking probability is bounded and predictable. Moreover, larger minimum distance makes them able to guarantee 100% detection for a predefined class of errors so that they can be useful for providing the highest protection against the most likely or most dangerous threat while maintaining a detection guarantee in case of an unexpected behavior or attack.

For some applications the error characteristics of robust codes can be considered too pessimistic. Variants of robust codes which fill the gap between the optimistic linear codes and pessimistic robust codes are possible. *Partially robust* codes and *minimum distance partially robust* codes allow for a trade off between robustness, encoding complexity, and overheard.

Definition 3.4 *A systematic* (n, k) *code with a detection kernel smaller than* 2^k *is a* ***partially robust code****. If the code also has a minimum distance greater than one it is referred to as a* ***minimum distance partially robust code****.*

Partially robust codes reduce the number of undetectable errors while preserving some structures of linear codes which can be exploited to build efficient prediction hardware that generates redundant bits of a message. Like linear codes, partially robust codes still have undetectable errors (hence they are not completely robust). The number of undetectable errors is reduced by many orders of magnitude. For practical partially robust constructions the number of undetectable errors can be reduced from 2^k to $2^{k-r}, r = n - k$ compared to a linear (n, k) code [18]. The probability of masking for the errors that are detectable is bounded as in robust codes.

For memory applications, we are mostly interested in minimum distance robust or partially robust codes that can be used to do error corrections. Compared with traditional linear error correcting codes, the advantage of these codes is that they can provide better protection against multiple errors due to the fact that they have less undetectable/miscorrected errors. Several constructions and examples of minimum distance robust/partially robust codes will be described in the next section.

4 Constructions of Codes

4.1 Minimum Distance Robust Codes

Systematic robust and partially robust codes are highly related to nonlinear functions. The nonlinearity of a function $f : GF(2^k) \rightarrow GF(2^s)$ can be measured by using derivatives $D_a f(x) = f(x+a) + f(x)$. The nonlinearity measure can be defined by (from [3])

$$P_f = \max_{0 \neq a \in GF(2^k)} \max_{b \in GF(2^s)} Pr(D_a f(x) = b), \quad (4)$$

where $Pr(E)$ denotes the probability of occurrence of event E. The smaller the value of P_f, the higher the corresponding nonlinearity of f. When $P_f = 2^{-s}$, f is a perfect nonlinear function

The simplest way to construct minimum distance robust codes is to append extra nonlinear redundant bits to codewords of an existing code with given distance d.

Theorem 4.1 *[19] Let V be a systematic (n, k, d) code and let $f : GF(2^k) \rightarrow GF(2^s)$ be nonlinear function with nonlinearity P_f. The code*

$$C = \{(x, \phi(x), f(x)) | (x, \phi(x)) \in V\}, \quad (5)$$

where ϕ is the encoding function for code V, is a $(n + s, k, d)$ minimum distance robust code where $\max_{e \neq 0} Q(e) \leq P_f$.

Example 4.1 ***(Shortened Robust Hamming)*** *Let $C = \{(x, Px)\}$ be a $(38, 32, 3)$ shortended Hamming code, where $x \in GF(2^{32})$, P is a (6×32) encoding matrix and $Px \in GF(2^6)$. Let $f : GF(2^{32}) \rightarrow GF(2)$ be a perfect nonlinear function defined by $f(x = (x_1, x_2, ..., x_{32})) = x_1x_2 + x_3x_4 + ... + x_{31}x_{32}$ [3] (non-repetitive quadratic function) . Then the code $C = \{(x, Px, f(x))\}$ is a robust code with minimum distance 3. For this code, $Q(e) = 0$ when $||e|| < 3$ and $Q(e) \leq 0.5$ when $||e|| \geq 3$.*

Shortened robust Hamming code has no undetectable errors. The only element in K_d is the zero vector. It is able to correct any single bit error and provide nearly equal protection to most of the multiple errors. However, the advantage of shortened robust Hamming code comes at the price of one more redundant bit. The code has only minimum distance 3 although it needs the same number of redundant bits as $(39, 32)$ SEC-DED codes which have minimum distance 4.

4.2 Minimum Distance Partially Robust Codes

Many classical constructions of nonlinear codes are partially robust minimum distance codes. They have a minimum distance larger than 1 and have much fewer undetectable errors than linear codes. Such codes can even be perfect with respect to the classical Hamming bound.

The first nonlinear perfect code was constructed by Vasil'ev in [35] and was generalized by Mollard in [22]. We first review the basic construction of Vasil'ev code.

Theorem 4.2 ***(Vasil'ev Code[35])*** *For $x \in GF(2^m)$, let $p(x) = ||x||$ mod 2. Let V be a perfect not necessarily linear Hamming code of length $m = 2^r - 1$ with $k_V = m - r$ information bits. Let $f : V \rightarrow \{0, 1\}$ be an arbitrary nonlinear mapping such that $f(\mathbf{0}) = 0$ and $f(v) + f(v') \neq f(v + v')$ for some $v, v' \in V$. The code C defined by*

$$C = \{(x, x + v, p(x) + f(v)) | x \in GF(2^m), v \in V\} \quad (6)$$

(where $+$ is over $GF(2)$) is a $(2m+1, 2m-r, 3)$ perfect nonlinear Hamming code.

Remark 4.1 *We note that the above construction can be generalized to generate robust codes with any given distance d. Denote by P a binary matrix. The code*

$$C = \{(x, x+v, Px+f(v)) | x \in GF(2^m), v \in V\} \quad (7)$$

is a nonlinear code with minimum distance d if V has distance d and the code composed of all vectors (x, Px) has distance $d-1$. Vasil'ev code is a special case where (x, Px) is a linear parity code with minimum distance 2. Some partially robust codes as good as BCH codes in terms of the number of redundant bits can be generated based on this construction.

Theorem 4.3 *[19] Vasil'ev code is a $(2m + 1, 2m - r, 3)$ partially robust code with $|K_d| = 2^m$ and $\max_{e \notin K_d} Q(e) = P_f$ where P_f is the nonlinearity of f, K_d is the detection kernel of the code and $m = 2^r - 1$.*

Vasil'ev codes are perfect single error correcting codes and have the same parameters as linear Hamming codes. The basic construction of Vasil'ev code can be further generalized as follows. The Theorem can be proved in a similar way to the proof of Theorem 4.2 presented in [19].

Theorem 4.4 ***(Shortened Vasil'ev Code)*** *For $x \in GF(2^a)$, let $p(x) = ||x||$ mod 2. Let V be a $(m, k_V, 3)$ not necessarily linear Hamming code with $r = m - k_V$ redundant bits. Without lost of generality, assume that the first k_V bits in any codeword are information bits. Denote by $v = (y, z), y \in GF(2^{k_V}), z \in GF(2^r)$ the codewords of V. Let $f : GF(2^{k_V}) \rightarrow \{0, 1\}$ be an arbitrary mapping such that $f(\mathbf{0}) = 0$ and $f(y) + f(y') \neq f(y+y')$ for some $y, y' \in GF(2^{k_V})$. The code C defined by*

$$C = \{(x, (x, \boldsymbol{0}) + v, p(x) + f(y))\}, \quad (8)$$

where $x \in GF(2^a), \boldsymbol{0} \in GF(2^{m-a}), 0 < a \leq m, v \in V$ *is a* $(a + m + 1, a + k_V, 3)$ *code with* $|K_d| = 2^a$*, and* $max_{e \notin K_d} Q(e) = P_f$*. Adding one more overall linear parity bit to C will result in a nonlinear SEC-DED code with the same* K_d *and* $max_{e \notin K_d} Q(e)$ *as C and minimum distance 4.*

The significance of Theorem 4.4 is twofold. First, it can generate robust SEC-DED codes of arbitrary lengths. These codes have the same number of redundant bits as best linear SEC-DED codes but much smaller number of undetectable multiple errors and are more suitable for applications where MBU rate is high. Second, it allows a tradeoff in terms of robustness and the hardware overhead. Generally speaking, the smaller *a* is, the more robust the code is and more hardware overhead is required for the encoder. By carefully selecting *a* and *m*, we can construct codes for situations that have different requirements for robustness and the hardware overhead.

Example 4.2 ***(Extended Vasil'ev Code)***

1. *Let* $a = 16$ *and* V *be a* $(21, 16, 3)$ *Hamming code. Select* f *to be the same nonrepetitive quadratic function as in Example 4.1. The extended Vasil'ev code constructed by adding one more overall parity bit to the generalized Vasil'ev construction described in Theorem 4.4 is a* $(39, 32, 4)$ *partially robust code with* $|K_d| = 2^{16}$*,* $d = 4$ *and* $max_{e \notin K_d} Q(e) = 0.5$*.*

2. *Alternatively let* $a = 6$ *and* V *be a* $(31, 26, 3)$ *perfect Hamming code. We can construct a* $(39, 32, 4)$ *partially robust code with* $|K_d| = 2^6$ *at the price of larger hardware overhead for the encoder.*

3. *For applications where hardware overhead is more critical, we can select* a *to be* 18 *and* V *to be a* $(19, 14, 3)$ *Hamming code. The resulting partially robust code will have* $|K_d| = 2^{18}$*, which is the biggest of the 3 discussed variants. However, the hardware overhead for the encoder of this implementation will be the smallest.*

Other constructions of perfect nonlinear codes which have small detection kernels can be found in [24],[25],[31],[1],[9]. Table 1 compares K_d, $max_{e \notin K_d} Q(e)$ and encoding/decoding complexities of these nonlinear codes with perfect linear Hamming code. As expected, linear Hamming code has the lowest encoding/decoding complexities but the largest number of undetectable errors. Large K_d makes it unsuitable for applications where protection against multiple bit errors is important. Phelps-Solov'eva code ([24]) has slightly larger K_d than Vasil'ev code. The encoding and decoding complexity of this code is high due to the fact that at least two matrix multiplication over $GF(2)$ need to be performed to compute the syndromes of the two parts of the codeword. Using the switching constructions [9], perfect nonlinear codes can be constructed with a detection kernel of dimension one. However, the maximum $Q(e)$ for these codes are close to one. Moreover, their encoding and decoding complexities are much larger than Vasil'ev code and this drastically limits their applications. We therefore propose the Vasil'ev code and the extended Vasil'ev code as alternatives for traditional linear single error correcting and SEC-DED codes for applications where multiple bit errors are un-negligible.

We note that constructions of minimum distance robust and partially robust codes described in this section can be easily generalized for nonbinary case.

5 Architecture

In order to demonstrate the advantage of utilizing minimum distance partially robust code to protect memory against soft errors, we compare the error detection/correction properties as well as the hardware overhead of a $(39, 32, 4)$ extended Vasil'ev code to the modification of an in-use $(39, 32, 4)$ extended Hamming code. The latter was presented in [33] to protect double data rate DIMM memory in a Virtex-II Pro device.

Figure 1 shows the general memory architecture with ECC functions based on systematic error correcting codes. During a WRITE operation, the redundant bits of the code are generated by the encoder and saved in the redundant memory block. During a READ operation, the ECC block computes the signature of the retrieved data and executes the error correction algorithm. If uncorrectable errors occur, ERR will be asserted and no correction will be attempted.

5.1 Memory protection architecture based on extended Hamming code

For traditional linear SEC-DED codes, the encoder performs matrix multiplication over GF(2) between the k-bit data and the encoding matrix P of the selected linear code. The $(39, 32)$ parity check matrix used to generate the Hamming code C in [33] is in standard form

$(2^m-1, 2^m-1-m, 3)$ Perfect Codes	Dimension of K_d	$max_{e \notin K_d} Q(e)$	Encoding & Decoding Complexity
Linear Hamming Code [12]	2^m-1-m	-	Low
Vasil'ev Code [35]	$2^{m-1}-1$	≥ 0.5	Medium
Phelps-Solov'eva Code [24]	2^{m-1}	Depending on α [24]	High
One Switching Code [9]	$2^{m-1}-1$	$1-2^{-2^{m-1}+m+1}$	High
Multiple Switching Code [26]	1	close to 1	Very High

Table 1: Comparison of different perfect single error correcting codes

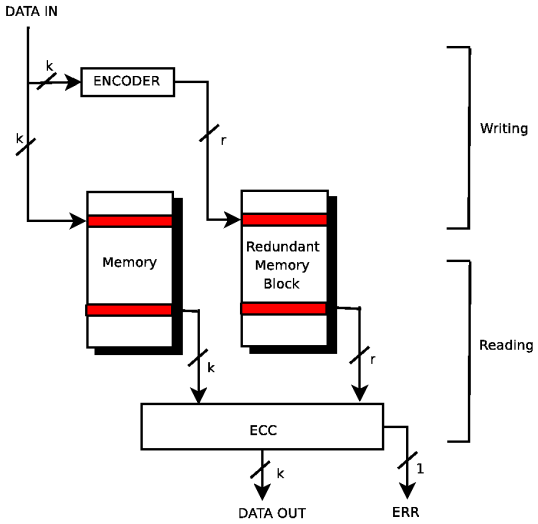


Figure 1: General Memory Architecture with ECC Functions

$H = (P|I)$, where I is the identity matrix and

$$P = \begin{pmatrix} 01010110101010101010110101011011 \\ 10011011001100110011011001101101 \\ 11100011110000111100011110001110 \\ 00000011111111100000011111110000 \\ 00000011111111100111110000000000 \\ 11111100000000000000000000000000 \\ 11111111111111111111111111111111 \end{pmatrix}.$$

The last parity bit of the design in [33] only covers the information part of the code. C is not a fully SEC-DED code and is only able to detect double errors occurring in the information part of the code. If at least one bit of the double error is in the redundant portion of C, the code may miscorrect it as a single bit error. To make C a fully SEC-DED code, we compute the last parity bit based on all bits of the codeword instead of only the information bits.

The check bits are generated and written in the memory along with the associated 32-bit data. During the memory READ stage, the data and the check bits are read simultaneously. Syndromes are calculated in a similar way to the check bit generation and are used to look for the error type and the error location. A 32-bit correction mask is created to correct single bit errors occurring to the information part of the code. When a single bit error is detected, the original data is XORed with the mask and the distorted bit is reversed. When there are no errors or multiple errors, all the mask bits are zeros. The data will go through the ECC block without any changes.

The disadvantage of memory protection scheme based on traditional linear SEC-DED codes is its large number of undetectable/miscorrected multiple errors. For any linear systematic code, $K_d = C, |K_d| = |C| = 2^k$. Thereby the number of undetectable errors for a $(39, 32, 4)$ extended Hamming code is 2^{32}.

It is easy to prove that any (n, k) linear systematic error correcting code C is able to correct up to $2^{n-k}-1$ errors. If N errors are corrected, $0 \leq N \leq 2^{n-k}-1$, the number of miscorrected errors is $N \cdot (2^k - 1)$.

For example, for the approach described in [33], only single errors occurring to the information part of the code will be corrected, $N = 32$. Thus the number of miscorrected multiple errors is $32(2^{32}-1)$.

5.2 Memory protection architecture based on extended Vasil'ev code

The codewords of a $(a+m+2, a+k_V, 4)$ extended Vasil'ev code is in the format of

$$(x, (x, \mathbf{0}) + v, p(x) + f(y), p(x) + p(v) + f(y)), \quad (9)$$

where $x \in GF(2^a)$, $\mathbf{0} \in GF(2^{m-a}), 0 < a \leq m$, $v \in V$ is the codeword of a $(m, k_V, 3)$ Hamming code, $y \in GF(2^{k_V})$ are the information bits of V, $f : GF(2^{k_V}) \rightarrow \{0, 1\}$ is a nonlinear mapping satisfying $f(\mathbf{0}) = 0$ and p is the linear parity function. In order to simplify the encoding and decoding complexities, we select V to be linear Hamming codes.

During the memory WRITE stage, all the check bits are generated by the encoder and saved in the redundant memory block. The redundant portion of the extended Vasil'ev code contains three parts. The first part is the redundant bits of V which can be generated by a linear XOR network performing matrix multiplication over $GF(2)$. The second and the third part are nonlinear (see 9). The encoder for these two parts needs to perform the

linear parity predictions $p(x), p(v)$ as well as the nonlinear mapping $f : GF(2^{k_V}) \rightarrow \{0,1\}$. When k_V is even, we can select f to be the non-repetive quadratic function (Example 4.1) for the purpose of minimizing $max_{e \notin K_d} Q(e)$.

$$f = v_1v_2 + v_3v_4 + v_5v_6 \cdots + v_{k_V-3}v_{k_V-2} + v_{k_V-1}v_{k_V}. \quad (10)$$

The error correction algorithm performed in the decoder of the extended Vasil'ev code is slightly more complex. Different from traditional linear error correcting codes, nonlinear codes do not have parity check matrix H. Hence the classical syndrome He that can be used to locate and correct errors for linear codes do not work for nonlinear error correcting codes. Before we describe the error correction algorithm for the extended Vasil'ev code, the signature S for locating and correcting errors need to be defined. Denote by $c = (c_1, c_2, c_3, c_4)$ the codeword of the extended Vasil'ev code, $e = (e_1, e_2, e_3, e_4)$ the error vectors and $\tilde{c} = (\tilde{c}_1, \tilde{c}_2, \tilde{c}_3, \tilde{c}_4)$ the distorted codeword.

$$\begin{aligned} c_1 &= x, \\ c_2 &= (x, \mathbf{0}) + v, \\ c_3 &= p(x) + f(y), \\ c_4 &= p(x) + p(v) + f(y). \end{aligned}$$

Let H be the parity check matrix of the linear code V and $\tilde{y}$ be the distorted information bits of V. The signature can be defined as $S = (S_1, S_2, S_3)$, where

$$S_1 = H((\tilde{c}_1, \mathbf{0}) + \tilde{c}_2)), \quad (11)$$
$$S_2 = p(\tilde{c}_1) + f(\tilde{y}) + \tilde{c}_3, \quad (12)$$
$$S_3 = p(\tilde{c}_1) + p(\tilde{c}_2) + p(\tilde{c}_3) + p(\tilde{c}_4). \quad (13)$$

The error correction algorithm is as stated below. Similar to the design described in [33],only single errors in the information part of the code will be corrected. If single errors in the redundant portion or multiple errors are detected, ERR will be asserted but no correction will be attemped.

1. Compute by (11),(12),(13) the signature of the code $S = (S_1, S_2, S_3)$, where $S_1 \in GF(2^{\lceil log_2(m+1) \rceil})$ and $S_2, S_3 \in GF(2)$.
2. If S is the all zero vector, then no error is detected. Otherwise one or more errors are detected.
3. If $S_3 = 0$ and at least one of S_1, S_2 is nonzero, errors with even multiplicities are detected and ERR will be raised. Errors in this class are uncorrectable because all of them are multiple errors.
4. If $S_3 = 1$ and $S_1 = \mathbf{0}$, a single bit error occurs to the last two redundant bits of the code. ERR will be asserted and the data will go through ECC without any correction because only single bit errors in the information part need to be corrected.
5. If $S_3 = 1, S_1 \neq \mathbf{0}$ and S_1 does not match any columns of the parity check matrix H, an uncorrectable multiple error of odd multiplicities is detected and ERR will be raised.
6. If $S_3 = 1$ and $S_1 = h_i$, where h_i is the i_{th} column of H of the linear code V, a single bit error in the first two parts of the code or multiple errors are detected. Without lost of generality, we assume that the first $m - \lceil log_2(m+1) \rceil$ bits of V are information bits. Let $k_V = m - \lceil log_2(m+1) \rceil$.
   - If $a \leq k_V$ and $1 \leq i \leq a$, flip the i_{th} bit of c_1, recalculate S_2. If $S_2 = 0$, the single error is in the i_{th} bit of c_1 and is successfully corrected. Otherwise the single error occurs in the i_{th} bit of c_2.
   - If $a \leq k_V$ and $a < i \leq k_V$ flip the i_{th} bit of c_2, recalculate S_2. If $S_2 = 0$, the single error is in the i_{th} bit of c_2 and is successfully corrected. Otherwise multiple errors with odd multiplities are detected.
   - If $a \leq k_V$ and $i > k_V$, the error occurs to the redundant bits of V and does not need to be corrected. ERR will be asserted and no correction will be attempted.
   - Similar procedures can be applied to the case when $a > k_V$.

Example 5.1 *In this example we show the encoding and decoding procedure for a* $(39, 32, 4)$ *extended Vasil'ev code* C *with* $a = 6$ *(see Theorem 4.4). The format of codewords is as stated in (9). When* $a = 6$, $x \in GF(2^6), v \in GF(2^{31}), k_V = 26$. *Let* $(11111001011011000110010111001111)$ *be the message that needs to be encoded.* $x = (111110)$, y *can be computed by XOR* $(x, \mathbf{0}), \mathbf{0} \in GF(2^{|a-k_V|})$ *with the other* $k_V = 26$ *bits of the message. Thus* $y = (10100011000110010111001111)$. *Let*

$$H = \begin{pmatrix} 11111011101101001111000000010000 \\ 11110111011010101000111000001000 \\ 11101110110110010100100111000100 \\ 11011101110001110010010101000010 \\ 10111100001111110001001011000001 \end{pmatrix}$$

be the parity check matrix of V. *Then the redundant bits of* $v \in V$ *are* (00101). *Let* f *be the nonrepetitive quadratic function as described in Example 4.1, then* $p(x) = 1, p(v) = 0, f(y) = 0$. *The last two*

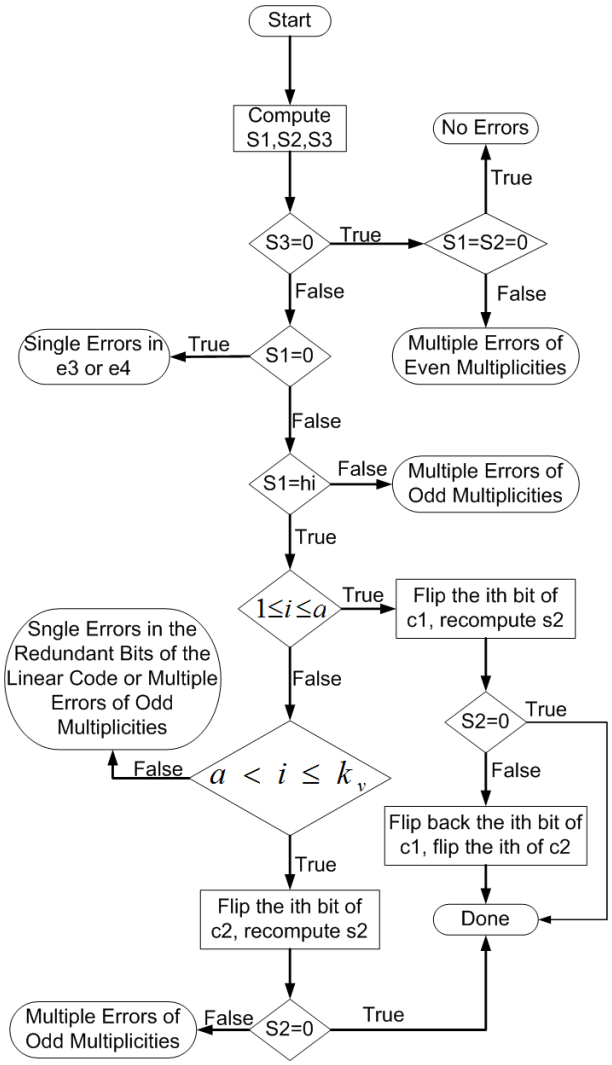


Figure 2: Error Correction Algorithm for the Extended Vasil'ev Code ($a \leq k_V$)

nonlinear redundant bits are 11. *The entire codeword is* $(11111001011011000110010111001111001010111)$. *Suppose a single bit error occurs to the* 9_{th} *bit of the codeword. After receiving the distorted codeword,* S_1, S_2 *and* S_3 *can be computed according to (11),(12),(13). We have* $S_1 = h_3 = (11101), S_2 = 0, S_3 = 1$. *The* 3_{rd} *bit of* x *is flipped and* S_2 *is recomputed. The new value of* S_2 *is 1. So the error is at the* 3_{rd} *bit of the second part i.e.* 9_{th} *bit of the entire codeword.*

The entire error correction procedure is shown in Figure 2. The sizes of K_d and K_c for extended Vasil'ev code can be computed according to the next theorem.

Theorem 5.1 *For* $(a + m + 2, a + k_V, 4)$ *extended Vasil'ev codes, where* $k_V = m - \lceil log_2(m + 1) \rceil$, *denote* $t = min\{a, k_V\}$. *There are* 2^a *undetectable errors and* $2^{a+1}(2^{k_V} - 1)$ *conditionally detectable errors. If only errors occurring to the information part of the code are corrected, the number of miscorrected errors is* $2t(2^{a+k_V} - 1) + (2^t - 1)|a - k_V|$. *The number of conditionally miscorrected errors is* $2|a - k_V|(2^{a+k_V} - 2^t)$. *The probability of error masking for conditionally detectable errors and the probability of miscorrection for conditionally miscorrected errors are bounded by* P_f, *which is the nonlinearity of* f *defined by (4) (see Theorem 4.4).*

Proof The signature of the code can be re-written as follows.

$$\begin{aligned} S_1 &= H((\tilde{c}_1, \mathbf{0}) + \tilde{c}_2) = H((e_1, \mathbf{0}) + e_2), \\ S_2 &= p(\tilde{c}_1) + f(\tilde{y}) + \tilde{c}_3 \\ &= f(\tilde{y}) + f(y) + p(e_1) + e_3 \\ S_3 &= p(\tilde{c}_1) + p(\tilde{c}_2) + p(\tilde{c}_3) + p(\tilde{c}_4) \\ &= p(e_1) + p(e_2) + p(e_3) + p(e_4). \end{aligned}$$

1. $K_d = \{e | S_1 = \mathbf{0} \in GF(2^{\lceil log_2(m+1) \rceil}), S_2 = S_3 = 0 \in GF(2), \forall x \in C\}$. $S_1 = H((e_1, \mathbf{0}) + e_2) = \mathbf{0} \Rightarrow (e_1, \mathbf{0}) + e_2$ is a codeword of the linear code V. Because $f : GF(2^{k_V})$ is a nonlinear function, the only possibility to guarantee $S_2 = 0, \forall x \in C$ is that $(e_1, \mathbf{0}) = e_2, p(e_1) = e_3$. $S_3 = 0 \Rightarrow e_4 = e_3 = p(e_1)$. So the detection kernel of the code contains all error vectors $e = (e_1, e_2, e_3, e_4)$ such that $(e_1, \mathbf{0}) = e_2, e_3 = e_4 = p(e_1)$. The number of errors in this class is 2^a;

2. If $(e_1, \mathbf{0}) + e_2$ is a nonzero codeword of V and $e_4 = p(e_1) + p(e_2) + p(e_3)$, then $S_1 = \mathbf{0}, S_3 = 0, \forall x \in C$. S_2 can be either 1 or 0 depending on the information part of the code. These errors will be conditionally detected. The error masking probability is bounded by P_f. If f is a perfect nonlinear function, these errors will be detected with probability 0.5. The number of errors in this class is $2^{a+1}(2^{k_V} - 1)$.

3. Multiple error e will be miscorrected as single bit errors occurring in the information part of the code if and only if $S_3 = 1, S_1 = h_i, 1 \leq i \leq max\{a, k_V\}$. Let $t = min\{a, k_V\}$.

   - If $1 \leq i \leq t$, e will always be miscorrected as a single error in the i_{th} bit of either c_1 or c_2. The number of pairs of e_1, e_2 satisfying $S_1 = H((e_1, \mathbf{0}), e_2) = h_i, 1 \leq i \leq t$ is $t \cdot 2^{a+k_V}$. e_3 can be either 1 or 0. $e_4 = p(e_1) + p(e_2) + p(e_3) + 1$. So there are $2t \cdot 2^{a+k_V}$ errors that satisfy $S_3 = 1, S_1 = h_i, 1 \leq i \leq t$. $2t$ of them are correctly corrected. The number of miscorrected errors in this class is $2t(2^{a+k_V} - 1)$.

- If $t < i \leq max\{a, k_V\}$, the number of errors satisfying $S_3 = 1, S_1 = h_i$ is $2|a - k_V| \cdot 2^{a+k_V}$. After flipping the i_{th} bit of either $\tilde{c_1}$ or $\tilde{c_2}$, S_1 and S_3 become zero. Denote by e_1^*, e_2^* the new error vectors after flipping the bit for the first two parts of the codewords.
  - If $(e_1^*, \mathbf{0}) + e_2^* = 0$ and $e_3 = p(e_1^*)$, S_2 is always zero. The number of errors in this class is $2^t \cdot |a - k_V|$ and $|a - k_V|$ of them are correctly corrected. The number of miscorrected errors is $(2^t - 1) \cdot |a - k_V|$.
  - If $(e_1^*, \mathbf{0}) + e_2^* = 0$ and $e_3 \neq p(e_1^*)$, S_2 is always one. Errors in this class are always detectable. The number of them is $2^t \cdot |a - k_V|$.
  - If $(e_1^*, \mathbf{0}) + e_2^* \neq 0$, then S_2 can be either 0 or 1 depending on the information bits of the code. Errors in this class will be conditionally miscorrected. The probability of miscorrection is bounded by P_f. If f is a perfect nonlinear function, the probability of miscorrection is 0.5. The number of errors in this class is $2|a - k_V|(2^{a+k_V} - 2^t)$.

The size of K_d and K_c are functions of a and m. For any $(n, k, 4)$ extended Vasil'ev code, we have

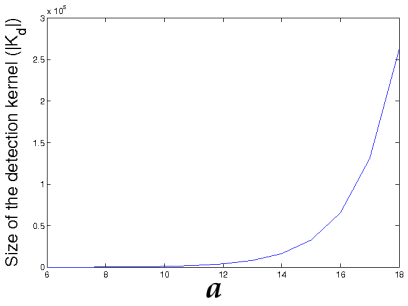
$$\begin{aligned} k &= a + k_V = a + m - \lceil log_2(m+1) \rceil, \\ n &= a + m + 2, \\ a &\leq m. \end{aligned}$$

Hence $n - 2^{n-k-2} - 1 \leq a \leq \lfloor \frac{n-2}{2} \rfloor$. When $n = 39, k = 32$, $6 \leq a \leq 18$. Figure 3 shows $|K_d|$ and $|K_c|$ of $(39, 32)$ extended Vasil'ev codes for different a. The minimum values of $|K_d|$ and $|K_c|$ are 2^6 and $12(2^{32} - 1) + 20(2^6 - 1)$ respectively, both of which are achieved when $a = 6$.

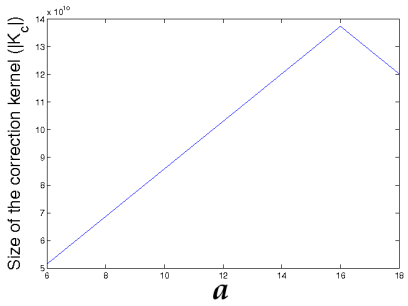
Different from traditional linear error detecting codes, extended Vasil'ev codes have conditionally undetectable/miscorrected errors. For $(39, 32, 4)$ extended Vasil'ev code with $a = 6$, the numbers of errors which are masked or miscorrected with probability 0.5 are $2(2^{32} - 2^6)$ and $40(2^{32} - 2^6)$.

5.3 Comparison of extended Hamming and extended Vasil'ev codes

Both extended Hamming code and extended Vasil'ev code are able to correct all single bit errors and detect all double bit errors. Under the assumption that all errors with higher multiplicities are equiprobable, extended Vasil'ev codes can provide better error protection than extended Hamming codes.



(a)



(b)

Figure 3: Kernel of $(39, 32, 4)$ exnteded Vasil'ev codes as a function of "a" (a) detection kernel (b) correction kernel

Table 2 shows the number of undetectable/miscorrected errors with multiplicities 3 to 6 for $(39, 32, 4)$ extended Hamming code and extended Vasil'ev code ($a = 6$). As expected, both codes have no undetectable single, double and triple errors. We note that 1583 quadruple errors will be masked by extended Hamming code while only 21 will be masked by extended Vasil'ev code. Only errors with odd multiplicities are miscorrected. The number of miscorrected errors with $||e|| \leq 6$ for the extended Vasil'ev code is less than one half of the corresponding number for the extended Hamming code.

Smaller kernels make extended Vasil'ev codes much more powerful for detection of repeating errors than the extended Hamming codes. Traditional linear error correcting code do not have conditionally undetectable/miscorrected errors. All errors are either 100% protected or not protected at all. As a result, $|K_d|$ and $|K_c|$ of linear codes are in general much larger than for nonlinear codes. Large K_d and K_c are bad for detection of repeating errors, since an error e will always be masked/miscorrected as long as it is masked/miscorrected for one single message. Detection of conditionally undetectable/miscorrected errors for extended Vasil'ev codes, however, is message-dependent. The more messages the error affects, the smaller the er-

	Code	$\|\|e\|\| = 3$	$\|\|e\|\| = 4$	$\|\|e\|\| = 5$	$\|\|e\|\| = 6$
Undetectable	Extended Hamming	0	1583	0	51744
	Extended Vasil'ev	0	21	0	0
Miscorrected	Extended Hamming	5176	0	254432	0
	Extended Vasil'ev	1635	0	108993	0

Table 2: Detection and Correction Kernel

ror masking probability is. Thereby, in "lazy channels" [17] where errors tend to repeat themselves, extended Vasil'ev codes have more advantages. Repeating errors can occur in many situations. If a SEU lasts for several consecutive read/write cycles of the memory, it is possible that different messages written into the same memory cell are affected by the same error pattern. Another example of repeating errors in memory is the hard error. Hard errors are caused by permanent faults of the device and are unrecoverable by re-writing the memory cells thus tend to repeat themselves until the memory is replaced. For all applications where repeating errors exist, extended Vasil'ev codes can be a promising alternative to the classical extended Hamming codes.

The encoders and the error correction circuits of both the extended Hamming code and the extended Vasil'ev code (a=6) are synthesized in Synopsys design compiler for the purpose of comparing the hardware overheads. For $(39, 32, 4)$ extended Hamming code the encoder performs matrix multiplication over $GF(2)$ and can be implemented using 72 2-input XOR gates. The decoder can be implemented using 450 2-input logic cells and inverters. The hardware overhead for extended Vasil'ev code is slightly higher. 106 and 538 2-input cells and inverters are needed for the encoder and decoder of $(39, 32, 4)$ extended Vasil'ev code respectively $(a = 6)$. The small difference in overheads between the two codes in many cases is not very important due to the fact that hardware overhead of encoder and decoder for memory protection circuit counts only a very small portion of the whole device.

6 Conclusion

In this paper, a partially robust code with minimum distance 4 is proposed to replace the traditional linear extended Hamming codes to protect memories for situations where MBU rate is high or errors tend to repeat themselves. The numbers of undetectable and miscorrected multiple errors for the proposed code are much smaller than for traditional linear error correcting codes. In the presence of multiple bit distortion, our codes can provide much better protection against soft errors with only a small increase in hardware overhead. Different from linear codes, robust and partially robust codes have conditionally undetectable/miscorrected errors. The detection/correction of errors are message-dependent. This makes robust codes useful to detect/correct repeating errors, i.e. hard errors caused by permanent faults.

The proposed protection scheme is not targeted for any special memory architecture. It can be applied to nearly all types of memories such as RAM, ROM, FLASH and disk memories.

The constructions of binary minimum distance robust/partially codes shown in Section 4 can be easily generalized for non-binary case.

References

[1] BAUCER, H., GANTER, B., AND HERGERT, F. Algebraic techniques for nonlinear codes. In *Combinatoria* (1983), vol. 3, pp. 21–33.

[2] BHATTACHARRYYA, D., AND NANDI, S. An efficient class of sec-ded-aued codes. In *Third International Symposium on Parallel Architectures, Algorithms, and Networks* (1997).

[3] CARLET, C., AND DING, C. Highly nonlinear mappings. *Journal of Complexity 20*, 2-3 (2004).

[4] CHEN, C. L. Error-correcting codes with byte error-detection capability. *Computers, IEEE Transantions on C-32* (July 1983), 615–621.

[5] CHEN, C. L. Symbol error correcting codes for memory applications. In *Proceedings of the The Twenty-Sixth Annual International Symposium on Fault-Tolerant Computing (FTCS '96)* (1996).

[6] DUNNING, L. A. Sec-bed-ded codes for error control in byte-organized memory systems. *IEEE Transactions on Computer 34* (1985), 557–562.

[7] DUTTA, A., AND TOUBA, N. A. Multiple bit upset tolerant memory using a selective cycle avoidance based sec-ded-daec code. In *25th IEEE VLSI Test Symmposium (VTS'07)* (2007).

[8] ETO, A., HIDAKA, M., OKUYAMA, Y., KIMURA, K., AND HOSONO, M. Impact of neutron flux on soft errors in mos memories. In *Electron Devices Meeting* (1998).

[9] ETZION, T., AND VARDY, A. Perfect binary codes: Constructions, properties, and enumeration. In *IEEE Trans. on Information Theory* (1994), vol. 40, pp. 754–763.

[10] GAUBATZ, G., SUNAR, B., AND KARPOVSKY, M. G. Nonlinear residue codes for robust public-key arithmetic. In *Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC '06)* (2006).

[11] GEORGAKOS, G., HUBER, P., OSTERMAYR, M., AMIRANTE, E., AND RUCKERBAUER, F. Investigation of increased multi-bit failure rate due to neutron induced seu in advanced embedded srams. In *Symposium on V/LSI Circuits Digest of Technical Paper* (2007).

[12] HAMMING, R. W. Error correcting and error detecting codes. *The Bell System Technical Journal* (1950).

[13] HSIAO, M. Y. A class of optimal minimum odd-weight-column sec-ded codes. *IBM Journal of Research and Development 14* (1970), 395–401.

[14] JOHNSTON, A. H. Scaling and technology issues for soft error rates. 4th Annual Research Conference on Reliability.

[15] KARPOVSKY, M., KULIKOWSKI, K., AND A.TAUBIN. Differential fault analysis attack resistant architectures for the advanced encryption standard. Proc. IFIP World Computing Congress, Cardis, pp. 177–193.

[16] KARPOVSKY, M., KULIKOWSKI, K., AND A.TAUBIN. Robust protection against fault-injection attacks on smart cards implementing the advanced encryption standard. Proc. Int. Conference on Dependable Systems and Networks (DNS 2004).

[17] KARPOVSKY, M. G., KULIKOWSKI, K., AND Z, W. Robust error detection in communication and computation channels. In *Int. Workshop on Spectral Techniques* (2007).

[18] KARPOVSKY, M. G., AND TAUBIN, A. A new class of nonlinear systematic error detecting codes. *IEEE Trans Info Theory 50*, 8 (2004), 1818–1820.

[19] K.KULIKOWSKI, Z.WANG, AND M.G.KARPOVSKY. Comparative analysis of fault attack resistant architectures for private and public key cryptosystems. In *Proc of Int. Workshop on Fault-tolerant Cryptographic Devices* (2008).

[20] LALA, P. An adaptive double error correction scheme for semiconductor memory systems. *Digital processes 4* (1978), 237–243.

[21] MAIZ, J., HARELAND, S., ZHANG, K., AND ARMSTRONG, P. Characterization of multi-bit soft error events in advanced srams. In *IEEE Int'l Electronic Device Meeting* (December 2003), pp. 519–522.

[22] MOLLARD, M. A generalized parity function and its use in the construction of perfect codes. In *SIAM J. Alg. Disc. Meth* (1986), vol. 7, pp. 113–115.

[23] PENZO, L., SCIUTO, D., AND SILVANO, C. Construction techniques for systematic sec-ded codes with single byte error detection and partial corrrection capability for computer memory systems. *IEEE Transactions on Information Theory 41*, 2 (March 1995).

[24] PHELPS, K. T. A combinatorial construction of perfect codes. In *SIAM J. Alg. disc Meth.* (1983), vol. 4, pp. 398–403.

[25] PHELPS, K. T. A general product construction for error-correcting codes. In *SIAM J. Alg. disc Meth.* (1984), vol. 5, pp. 224–228.

[26] PHELPS, K. T., AND LEVAN, M. Kernels of nonlinear hamming codes. *Designs, Codes and Cryptography 6* (1995).

[27] P.K.LALA. A single error correcting and double error detecting coding scheme for computer memory systems. In *Proceedings of the 18th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems* (2003).

[28] REDDY, S. A class of linear codes for error control in byte-per-card organized digital systems. *Computer, IEEE Transanctions on C-27* (May 1978), 455–459.

[29] SATOH, S., TOSAKA, Y., AND WENDER, S. A. Geometric effect of multiple-bit soft errors induced by cosmic ray neutrons on drams, June 2000.

[30] S.K.MOORE. Masters of memory. *IEEE Spectrum 44*, 1 (Jan 2007), 45–49.

[31] SOLOV'EVA, F. I. On binary nongroup codes. In *Methodi Diskr. Analiza* (1981), vol. 37, pp. 65–76.

[32] SWIFT, G. M. In-flight observations of multiple-bit upset in drams. *IEEE Trans. Nuclear Science 47* (2001).

[33] TAM, S. *Application Note: Single Error Correction and Double Error Detection*. XILINX, 2006.

[34] T.R.HALFHIL. Z-ram shrinks embedded memory. Tech. rep., Microprocessor Report, Oct 2005.

[35] VASIL'EV, J. L. On nongroup close-packed codes. In *Probl.Kibernet.* (1962), vol. 8, pp. 375–378.

[36] WHITAKER, S., K.CAMERON, G.MAKI, J.CANARIS, AND P.OWSLEY. Vlsi reed-solomon processor for the hubble space telescope. In *VLSI Signal Processing IV IEEE Press* (1991).