

Reliable MLC NAND Flash Memories Based on Nonlinear t -Error-Correcting Codes

Zhen Wang, Mark Karpovsky, Ajay Joshi
Reliable Computing Laboratory, Boston University
8 Saint Mary's Street, Boston, MA, USA
{lark,markkar,joshi}@bu.edu

Abstract

Multi-level cell (MLC) NAND flash memories are very popular storage media because of their power efficiency and big storage density. This paper proposes to use nonlinear t -error-correcting codes to replace linear BCH codes for error detection and correction in MLC NAND flash memories. Compared to linear BCH codes with the same bit-error correcting capability t , the proposed codes have less errors miscorrected by all codewords and nearly no undetectable errors. For example, the proposed (8281, 8201, 11) 5-error-correcting code has no errors of multiplicity six miscorrected by all codewords while the widely used (8262, 8192, 11) linear shortened BCH code has $\binom{11}{6} \times A_{11}$ errors in this class, where $A_{11} \approx 10^{14}$ is the number of codewords of Hamming weight eleven in the shortened BCH code. Moreover, in spite of the fact that the Hamming distance of the proposed code is $2t+1$, it can also correct some errors of multiplicity $t+1$ and $t+2$ requiring no extra hardware overhead and latency penalty. In this paper, the constructions and the error correction algorithm for the nonlinear t -error-correcting codes are presented. The architecture of the encoder and the decoder for the codes are shown. The error correcting capabilities, the hardware overhead, the latency and the power consumption for the encoder and the decoder will be analyzed and compared to that of the linear BCH codes to demonstrate the advantages of the proposed codes for error detection and correction in MLC NAND flash memories.

1 Introduction

The semiconductor industry witnesses an explosive growth of the NAND flash memory market in the past several years. Due to its high data transfer rate, low power consumption, big storage density and long mechanical durability, the NAND flash memories are widely used as storage

media for devices such as portable media players, digital cameras, cell phones and low-end netbooks.

The increase of the storage density and the reduction of the cost per bit of flash memories were traditionally achieved by the aggressive scaling of the memory cell transistor until the multi-level cell (MLC) technology was developed and implemented in 1997 [1]. MLC technology is based on the ability to precisely control the amount of charge stored into the floating gate of the memory cell for the purpose of setting the threshold voltage to a number of different levels corresponding to different logic values, which enables the storage of multiple bits per cell.

However, one of the main issues caused by the increased number of programming threshold voltage levels is that the reliability of the memories is degraded due to the reduced operational margin. The raw bit error rate of the MLC NAND flash memory is around 10^{-6} [2] and is at least two orders of magnitude worse than that of the single-level cell (SLC) NAND flash memory [3]. Moreover, the same reliability concerns as for SLC NAND flash memories may become more significant for MLC NAND flash memories, e.g. program/read disturb, data retention, programming/erasing endurance [4] and soft errors [5][6][7]. Thereby, a powerful error correcting code (ECC) that is able to correct at least 4-bit errors is required for the MLC NAND flash memories to achieve an acceptable application bit error rate, which is no larger than 10^{-11} [2].

Several works have investigated the potential usage of linear block codes to improve the reliability of MLC NAND flash memories. In [8], the author presented a high-throughput and low-power ECC architecture based on $(n = 4148, k = 4096, d = 9)$ BCH codes with $t = 4$. In [9], a 4Gb 2b/cell NAND flash memory chip incorporating a 250MHz BCH error correcting architecture was shown. The author of [10] demonstrated that the use of strong BCH codes (e.g. $t = 12, 15, 67, 102$) can effectively increase the number of bits per cell thus further increase the storage capacity of MLC NAND flash memories. In [11], an adaptive-rate ECC architecture based on BCH codes was

proposed. The design had four operation modes with different error correction capabilities. An ECC architecture based on Reed-Solomon codes of length 828 and 820 information symbols constructed over $GF(2^{10})$ was proposed in [12], which can correct all bit errors of multiplicity less or equal to four. The architecture achieves higher throughput, requires less area overhead for the encoder and the decoder but needs 32 more redundant bits than architectures based on BCH codes with the same error correcting capability. In [13], an architecture based on asymmetric limited-magnitude error correcting code was proposed, which can correct all asymmetric errors of multiplicities up to t .

All the above architectures are based on linear block codes and have a large number of undetectable errors. Indeed, for any linear code with k information bits, the number of undetectable errors is 2^k , which is a potential threat to the reliability of the memory systems. What aggravates the situation even more is the miscorrection of errors. Denote by e a binary error vector and $\|e\|$ the multiplicity of e . A multi-bit error e , $\|e\| > t$ is miscorrected by a t -error-correcting code if and only if it has the same syndrome as some e' , $\|e'\| \leq t$. It is easy to show that the number of errors miscorrected by all codewords of a (n, k, d) linear t -error-correcting code is $\sum_{i=1}^t \binom{n}{i} \times (2^k - 1)$.

Under the assumption that errors are independent whose distribution satisfies $P(e) = p^{\|e\|}(1-p)^{n-\|e\|}$, where p is the raw bit distortion rate and $P(e)$ is the probability of the occurrence of event e , the most harmful miscorrected errors are errors of multiplicity $t + 1$. Denote by A_{2t+1} the number of codewords of Hamming weight $2t + 1$. The number of errors of multiplicity $t + 1$ that are miscorrected by all codewords is $\binom{2t+1}{t} \times A_{2t+1}$.

For the commonly used $(n = 8262, k = 8192, d = 11)$ linear BCH codes with $t = 5$, the number of errors of multiplicity six miscorrected by all codewords is as large as $462 \times A_{11} \approx 10^{17}$. Such a big number of miscorrected errors of multiplicity six can not be neglected and should be taken into account when designing reliable MLC NAND flash memories.

To reduce the number of undetectable and miscorrected errors, nonlinear *minimum distance robust* and *minimum distance partially robust* codes have been proposed in [14][15][16]. An ECC architecture based on nonlinear single-error-correcting, double-error-detecting (SEC-DED) codes for the protection of memories against soft errors has been shown in [15].

In this paper, we first generalize the existing nonlinear perfect Hamming codes, i.e. Vasil'ev codes [17], to generate partially robust nonlinear t -error-correcting codes of any length and any Hamming distance. Then we propose an ECC architecture for MLC NAND flash memories based on the generalized nonlinear t -error-correcting codes. The proposed architecture has nearly no undetectable errors and

much less errors miscorrected by all codewords than that based on the linear BCH codes with the same error correcting capability t at the cost of only a little extra overhead in area, latency and power consumption. Moreover, the proposed architecture can also correct some errors of multiplicity $t + 1$ and $t + 2$ requiring no extra overhead resulting in further improvement of the reliability of the memories.

In addition to errors that are undetectable or miscorrected by all codewords, there are also some errors which are masked or miscorrected by a fraction of codewords of the proposed nonlinear t -error-correcting codes, which are called **conditionally detectable** and **conditionally miscorrected** errors. We note that the data-dependent error detection and correction properties of the proposed codes are useful for detecting and locating repeating errors, e.g. errors introduced by hardware malfunctions such as data retention and programming/erasing failure.

The rest part of the paper is organized as follows. In Section 2, we briefly review the architecture of MLC NAND flash memories and explain the error model we use in the paper. In Section 3, we generalize the constructions of existing nonlinear perfect Hamming codes for any length and any distance and analyze their error detecting capabilities. In Section 4, the error correction algorithm for the proposed nonlinear t -error-correcting codes will be presented. The error correcting capability of the proposed codes will be evaluated. The hardware design of the encoder and the decoder for the proposed codes will be given in Section 5. The area overhead, the latency and the power consumption of the design will be estimated and compared to that of the linear BCH code and the advantage of the proposed architecture will be demonstrated. We conclude the paper in Section 6.

2 Overview of MLC NAND Flash Memories

2.1 Cell Threshold Voltage Distribution

MLC memory cell is able to store multiple bits by precisely controlling the threshold voltage level. In practice, the threshold voltage of the whole memory array satisfies a Gaussian distribution due to random manufacturing variations. Figure 2 illustrates the threshold voltage distribution of a multi-level cell which can store 2 bits. Denote by σ the standard deviation of the middle two Gaussian distributions. The standard deviations of the outer two distributions are approximately 4σ and 2σ . Each voltage range corresponds to a specific logic value represented as a 2-bit binary vector. Different schemes can be used for mapping the logic values to binary vectors. A direct mapping was used in [1] while the author in [12] proposed to use a Gray mapping to improve the reliability of the memory since it is more likely that a voltage level will be taken as one of its adjacent levels

during READ operation when error happens (Figure 2).

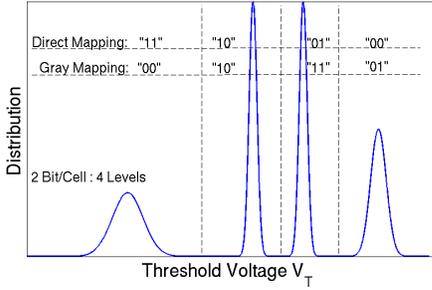


Figure 1: Threshold voltage distribution for 2bit/cell

2.2 Data Organization

The data of the NAND flash memory is organized in *blocks*. Each block consists of a number of *pages*. Each page stores K data bytes and R spare bytes. The spare area is physically the same as the rest of the page and is typically used for overhead functions such as ECC and wear-leveling [18]. The proportion of the spare bytes in the total number of bytes per page is usually 3%, e.g. 64 spare bytes for 2048 data bytes. More spare bytes may be required as the page size increases, e.g. 218 spare bytes for 4096 data bytes [2]. Due to the existence of spare bytes, the number of redundant bits of the error correcting codes used for NAND flash memories is not as critical as for other types of memories such as SRAM and DRAM where the area overhead is mostly determined by the number of redundant bits. This allows for a flexible design of more powerful error correcting codes for NAND flash memories.

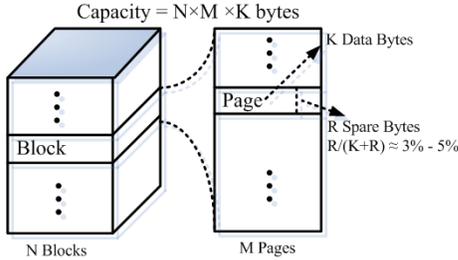


Figure 2: Data organization in NAND flash memories

2.3 Error Model for MLC NAND Flash Memories

Similar to SLC flash memories, the primary failure mechanisms for MLC NAND flash memories include threshold voltage distribution, program/read disturb, data retention, programming/erasing endurance and single event upset. However, while for SLC flash memories a lot of errors are asymmetric, e.g. errors introduced by program dis-

turb and data retention [13], for MLC NAND flash memories errors have no preferred symmetry [19]. Moreover, experimental results show that errors in MLC flash memories are more likely to occur uniformly within a page without any observable burstiness or local data dependency [19]. Thereby, throughout the paper we assume a random symmetric error model, where $P(e) = p^{|e|}(1-p)^{n-|e|}$, p is the raw bit distortion rate and $|e|$ is the multiplicity of the error. However, we want to emphasize that the proposed nonlinear t -error-correcting codes can also provide a guaranteed level of reliability in situations where the error model is unpredictable or multi-bit errors are more probable.

3 Constructions of Nonlinear t -Error-Correcting Codes

Throughout the paper we denote by \oplus the addition in binary field and (n, k, d) a n -bit binary code with k information bits and Hamming distance d . In general, the error correcting capability $t = \lfloor \frac{d-1}{2} \rfloor$.

The error detection properties of nonlinear codes are highly related to nonlinear functions. The nonlinearity of a function $f : GF(2^k) \rightarrow GF(2^r)$ can be measured by using derivatives $D_a f(x) = f(x \oplus a) \oplus f(x)$. The nonlinearity measure can be defined by (from [20])

$$P_f = \max_{0 \neq a \in GF(2^k)} \max_{b \in GF(2^r)} P(D_a f(x) = b), \quad (1)$$

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

3.1 Generalized Vasil'ev Codes

Vasil'ev code was first proposed in [17] in 1962 for $n = 2^r - 1$ and $t = 1$, where r is the number of redundant bits. We generalize the original construction to generate nonlinear partially robust codes with any length n and any Hamming distance d as shown in the following theorem.

Theorem 3.1 Let V be a (n_1, k_1, d) code and $\{(u, uP)\}$ be a $(n_2, k_2, d-1)$ code, where $u \in GF(2^{k_2})$, $k_2 \leq n_1$ and P is a $k_2 \times r_2$ binary encoding matrix (the last r_2 columns of the generator matrix of the code in standard form), $r_2 = n_2 - k_2$. Let $f : GF(2^{k_1}) \rightarrow GF(2^{r_2})$ be an arbitrary mapping such that $f(\mathbf{0}), \mathbf{0} \in GF(2^{k_1})$ is equal to zero and $f(y) \oplus f(y') \neq f(y \oplus y')$ for some $y, y' \in GF(2^{k_1})$. Denote by v_k the information bits of $v \in V$. The code defined by

$$C = \{(u, (u, \mathbf{0}) \oplus v, Pu \oplus f(v_k))\}, \mathbf{0} \in GF(2^{n_1 - k_2}) \quad (2)$$

is a $(n_1 + n_2, k_1 + k_2, d)$ partially robust code with 2^{k_2} undetectable errors. The remaining errors are detectable with

a probability of at least $1 - P_f$, where P_f is the nonlinearity of f .

Proof Let $c = (u, (u, \mathbf{0}) \oplus v, Pu \oplus f(v_k))$, $c' = (u', (u', \mathbf{0}) \oplus v', Pu' \oplus f(v'_k))$ be two codewords of C . The Hamming distance between c and c' is

$$\begin{aligned} \|c \oplus c'\| &= \|u \oplus u'\| + \|(u, \mathbf{0}) \oplus v \oplus (u', \mathbf{0}) \oplus v'\| \\ &+ \|Pu \oplus f(v_k) \oplus Pu' \oplus f(v'_k)\| \\ &\geq \|v \oplus v'\|. \end{aligned}$$

1. If $v \neq v'$, $\|c \oplus c'\| \geq d$ because the Hamming distance of V is d .
2. If $v = v'$, $\|c \oplus c'\| = 2 \times \|u \oplus u'\| + \|Pu \oplus Pu'\| \geq d$ because the Hamming distance of $\{(u, Pu)\}$ is $d - 1$.

Thereby, the Hamming distance of C is d . We say that an error e is masked by a codeword c if $e \oplus c = c' \in C$. Let H be the parity check matrix of V . An error $e = (e_1, e_2, e_3)$ where $e_1 \in GF(2^{k_2})$, $e_2 \in GF(2^{n_1})$ and $e_3 \in GF(2^{r_2})$ is masked if and only if $H((e_1, \mathbf{0}) \oplus e_2)$ is zero and $f(\tilde{v}_k) \oplus f(v_k) \oplus p(e_1) \oplus e_3 = 0$, where \tilde{v}_k is the information part of $\tilde{v} = v \oplus e_1 \oplus e_2$. The errors can be divided into four classes as follows.

1. $(e_1, \mathbf{0}) = e_2$ and $Pe_1 = e_3$. The error will always be masked. The number of errors in this class is 2^{k_2} ;
2. $(e_1, \mathbf{0}) = e_2$ but $Pe_1 \neq e_3$. The error will always be detected. There are $2^{n_2} - 2^{k_2}$ errors belonging to this class;
3. $H((e_1, \mathbf{0}) \oplus e_2)$ is zero but $(e_1, \mathbf{0}) \neq e_2$. The error masking probability depends on the nonlinear function f . In the worst case, a specific error will be masked by $P_f \times |C|$ codewords. The number of errors in this class is $2^{n_2}(2^{k_1} - 1)$;
4. $H((e_1, \mathbf{0}) \oplus e_2)$ is not zero. The error will always be detected. The number of errors is $2^{n_2}(2^{n_1} - 2^{k_1})$. ■

Remark 3.1 *Vasil'ev code is a special case where $\{(u, Pu)\}$ is a linear parity code with minimum distance two and V is a perfect Hamming code.*

Some partially robust nonlinear BCH codes as good as linear BCH codes in terms of the number of redundant bits for the same distance and length can be generated based on the above construction.

Example 3.1 *In [21], it was shown that the largest possible k for binary codes with $n = 63$ and $d = 5$ is 52. Let V be a $(63, 52, 5)$ code. Let $\{(u, Pu)\}$ be a $(4, 1, 4)$ repetition code that contains only 2 codewords 0000 and 1111. Select f to be a quadratic perfect nonlinear function $f = s_1 \bullet s_2 \oplus s_3 \bullet s_4 \oplus \dots \oplus s_{13} \bullet s_{14}$ with $P_f = \frac{1}{8}$ where $s_i \in GF(2^3)$ and \bullet is the multiplication in $GF(2^3)$. A $(67, 53, 5)$ partially robust nonlinear BCH code can be constructed as described*

in Theorem 3.1. This code has Hamming distance five and only one undetectable nonzero error with the same number of redundant bits as linear BCH codes. All the other errors are detectable with a probability of at least 0.875.

4 Error Correction Algorithm for Nonlinear t -Error-Correcting Codes

4.1 Parameters of the Selected Code

For the protection of MLC NAND flash memories, we propose to use a $(8281, 8201, 11)$ nonlinear 5-error-correcting code with $n_1 = 8270$, $k_1 = 8200$, $n_2 = 11$, $k_2 = 1$. The detailed construction of the code is described below. All the algorithms and analysis in this section are also applicable to codes with other lengths and correcting capabilities.

Denote by $c = (x_1, x_2, x_3)$ a codeword of the nonlinear t -error-correcting codes constructed as in Theorem 3.1. From (2), we have

$$\begin{aligned} x_1 &= u, x_1 \in GF(2^{k_2}), \\ x_2 &= (u, \mathbf{0}) \oplus v, x_2 \in GF(2^{n_1}), \\ x_3 &= Pu \oplus f(v_k), x_3 \in GF(2^{r_2}). \end{aligned}$$

Let V be a $(8270, 8200, 11)$ linear BCH code. To simplify the encoding and decoding procedures, we select $\{(u, Pu)\}$ to be a repetition code with $k_2 = 1$ and $r_2 = 10$. $f : GF(2^{8200}) \rightarrow GF(2^{10})$ is chosen to be a quadratic perfect nonlinear function with $P_f = 2^{-10}$ for the purpose of maximizing the error detection and correction capabilities of the code. f is defined by

$$f(s) = s_1 \bullet s_2 \oplus s_3 \bullet s_4 \cdots s_{819} \bullet s_{820}, \quad (3)$$

where $s_i \in GF(2^{10})$, $1 \leq i \leq 820$ and \bullet is the multiplication in $GF(2^{10})$. The resulting nonlinear 5-error-correcting code constructed as in Theorem 3.1 is a $(8281, 8201, 11)$ partially robust code with only one undetectable nonzero error. In addition, there is a small fraction of errors which are detectable with a probability of at least $1 - 2^{-10}$.

Remark 4.1 *1. From Theorem 3.1, the Hamming distance of $\{(u, Pu)\}$ should be larger or equal to $d - 1$. For error correcting code with $t = 5$, $r_2 \geq 9$. We select $r_2 = 10$ because quadratic perfect nonlinear function from $GF(2^{8200})$ to $GF(2^9)$ does not exist.*

2. The pages in MLC NAND flash memories typically contain 2048 or 4096 data bytes. Multiple codes with separate redundant bits can be used to protect the whole page.

4.2 Algorithm and Properties

In this section we present the error correction algorithm for the case when $\{(u, Pu)\}$ is a repetition code with $k_2 = 1$ and $r_2 \geq d - 2$. We also assume that f is a perfect nonlinear function with $P_f = 2^{-r_2}$. The error will be corrected only if at least one of the information bits is distorted. If all errors are in the redundant bits, no correction will be attempted. Denote by $e = (e_1, e_2, e_3)$ the error vector and

Case	\hat{e}_2	S_1
No errors are detected	$\mathbf{0} \in GF(2^{n_1})$	0
Errors of multiplicity at most t are detected	Detected error vector	$\ \hat{e}_2\ $
Errors of multiplicity larger than t are detected	$\mathbf{0} \in GF(2^{n_1})$	-1

Table 1: The output of the linear BCH decoder

$\tilde{c} = (\tilde{x}_1, \tilde{x}_2, \tilde{x}_3)$ the distorted codeword, where $e_1, \tilde{x}_1 \in GF(2)$ ($k_2 = 1$), $e_2, \tilde{x}_2 \in GF(2^{n_1})$, $e_3, \tilde{x}_3 \in GF(2^{r_2})$ and $\tilde{x}_i = x_i \oplus e_i$, $1 \leq i \leq 3$. Denote by $\tilde{v} = (\tilde{x}_1, \mathbf{0}) \oplus \tilde{x}_2$ the distorted codeword in V and \tilde{v}_k the information part of \tilde{v} . After receiving the possibly distorted memory output $(\tilde{x}_1, \tilde{x}_2, \tilde{x}_3)$, compute $S_2 = P\tilde{x}_1 \oplus f(\tilde{v}_k) \oplus \tilde{x}_3$. Decode \tilde{v} using the standard decoder for the linear BCH code. The output of the linear BCH decoder contains two parts. One is the decoded error vector \hat{e}_2 and the other is the error flag signal S_1 . The values for these two signals in different cases are described in Table 1. The detailed error correction algorithm for the proposed nonlinear t -error-correcting code is as stated below.

1. Compute \hat{e}_2, S_1 and S_2 .
2. If $S_1 = 0, S_2 = \mathbf{0}, \mathbf{0} \in GF(2^{r_2})$, no error is detected.
Undetectable Errors: Rewrite \tilde{v} and S_2 as follows.

$$\tilde{v} = (x_1, \mathbf{0}) \oplus x_2 \oplus (e_1, \mathbf{0}) \oplus e_2, \quad (4)$$

$$S_2 = f(\tilde{v}_k) \oplus f(v_k) \oplus Pe_1 \oplus e_3. \quad (5)$$

Denote by H the parity check matrix of the linear BCH code. Nonzero errors will be masked if and only if $S_1 = 0$ and $S_2 = \mathbf{0}$. If $(e_1, \mathbf{0}) = e_2$ and $Pe_1 = e_3$, S_1 and S_2 are always zero and the error is undetectable. The number of errors in this class is 2^{k_2} . Since in our case $k_2 = 1$, there is only one nonzero error that can not be detected.

3. If $S_1 = 0, S_2 = \mathbf{1}$, where $\mathbf{1}$ is the all one's vector in $GF(2^{r_2})$, the error is $e = (1, \mathbf{1}, \mathbf{0} \in GF(2^{n_1+n_2-2}))$.
Miscorrection : A nonzero errors e' will be miscorrected as e if and only if $S_1 = 0$ and $S_2 = \mathbf{1}$. The only error e' that will be miscorrected by all codewords has $e'_1 = 0, e'_2 = \mathbf{0} \in GF(2^{n_1})$ and $e'_3 = \mathbf{1} \in GF(2^{r_2})$. The multiplicity of e' is r_2 .
4. Suppose $S_1 = 0, S_2 \neq \mathbf{1}, S_2 \neq \mathbf{0}$.

- (a) If $\|S_2\| \geq r_2 - t + 2$, the error is $e = (1, \mathbf{1}, \mathbf{0}, \mathbf{1} \oplus S_2)$, where $\mathbf{0} \in GF(2^{n_1-1})$.

Miscorrection : Errors $e' = (0, \mathbf{0}, S_2)$ will be miscorrected as e by all codewords, where $\mathbf{0} \in GF(2^{n_1})$. The number of miscorrected errors in this class is $\sum_{i=r_2-t+2}^{r_2-1} \binom{r_2}{i}$. The multiplicity of $\|e'\|$ is at least $r_2 - t + 2$.

- (b) If $\|S_2\| < r_2 - t + 2$, an error in x_3 or an error of multiplicity larger than t is detected.

5. If $S_1 = -1$, an error of multiplicity larger than t is detected.
6. If $S_1 > 0$, let $\hat{x}_2 = \tilde{x}_2 \oplus \hat{e}_2, \hat{v} = (\tilde{x}_1, \mathbf{0}) \oplus \hat{x}_2$. Denote by \hat{v}_k the information bits of \hat{v} . Compute $\hat{S}_2 = P\hat{x}_1 \oplus f(\hat{v}_k) \oplus \tilde{x}_3 = f(\hat{v}_k) \oplus f(v_k) \oplus Pe_1 \oplus e_3$.

- (a) If $\hat{S}_2 = \mathbf{0}, \mathbf{0} \in GF(2^{r_2})$, the error is $e = (0, \hat{e}_2, \mathbf{0})$. If the located errors are all in the redundant bits of \tilde{v} , no correction will be attempted. Otherwise the data will be corrected by XORing \tilde{x}_2 with \hat{e}_2 .

Miscorrection : A nonzero error e' will be miscorrected as e if and only if $\hat{S}_2 = \mathbf{0}$. The only error e' that will be miscorrected by all codewords is

$$e' = (1, \mathbf{1} \oplus \hat{e}_2[n_1 - 1], \hat{e}_2[n_1 - 2 : 0], \mathbf{1}),$$

where $\hat{e}_2[n_1 - 1]$ is the most significant bit of \hat{e}_2 and $\hat{e}_2[n_1 - 2 : 0]$ is the $n - 1$ least significant bits of \hat{e}_2 . Since there are $\sum_{i=1}^t \sum_{j=0}^{t-i} \binom{k_1}{i} \binom{r_1}{j}$ possible \hat{e}_2 , the number of miscorrected errors in this class is $\sum_{i=1}^t \sum_{j=0}^{t-i} \binom{k_1}{i} \binom{r_1}{j}$. The multiplicity of e' is at least $r_2 + 1$.

- (b) If $\hat{S}_2 = \mathbf{1}$, where $\mathbf{1}$ is the all one's vector in $GF(2^{r_2})$, the error is

$$e = (1, \mathbf{1} \oplus \hat{e}_2[n_1 - 1], \hat{e}_2[n_1 - 2 : 0], \mathbf{0}).$$

Miscorrection : A nonzero error $e' = (0, \hat{e}_2, \mathbf{1})$, where $\mathbf{1} \in GF(2^{r_2})$ will be miscorrected as e by all codewords. The number of miscorrected errors in this class is $\sum_{i=1}^t \binom{n_1}{i}$. The multiplicity of e' is at least $r_2 + 1$.

- (c) If $\hat{S}_2 \neq \mathbf{0}$ and $\hat{S}_2 \neq \mathbf{1}$.
 - i. If $\|\hat{e}_2\| = t$, an error of multiplicity larger than t is detected.
 - ii. If $\|\hat{e}_2\| < t$ and $1 \leq \|\hat{S}_2\| \leq t - \|\hat{e}_2\|$, the error is $e = (0, \hat{e}_2, \hat{S}_2)$.

Miscorrection : Errors e' in the following format will be miscorrected by all codewords as e :

$$e' = (1, \mathbf{1} \oplus \hat{e}_2[n_1 - 1], \hat{e}_2[n_1 - 2 : 0], \mathbf{1} \oplus \hat{S}_2).$$

The number of miscorrected errors is $\sum_{i=1}^{t-1} \sum_{j=0}^{t-1-i} \sum_{l=1}^{t-i-j} \binom{k_1}{i} \binom{r_1}{j} \binom{r_2}{l}$. Since $1 \leq \|\hat{S}_2\| \leq t - \|\hat{e}_2\|$, we have

$$r_2 - t + \|\hat{e}_2\| \leq \|\mathbf{1} \oplus \hat{S}_2\| \leq r_2 - 1.$$

Thereby the multiplicity of e' is at least $r_2 - t + 2$.

iii. If $\|\hat{e}_2\| < t$ and either

$$\begin{cases} \hat{e}_2[n_1 - 1] = 1, \\ r_2 - t + \|\hat{e}_2\| \leq \|\hat{S}_2\| \leq r_2 - 1, \end{cases}$$

or

$$\begin{cases} \hat{e}_2[n_1 - 1] = 0, \\ r_2 - t + 2 + \|\hat{e}_2\| \leq \|\hat{S}_2\| \leq r_2 - 1, \end{cases}$$

is satisfied, the error is

$$e = (1, 1 \oplus \hat{e}_2[n_1 - 1], \hat{e}_2[n_1 - 2 : 0], \mathbf{1} \oplus \hat{S}_2).$$

Miscorrection : Errors $e' = (0, \hat{e}_2, \hat{S}_2)$ will be miscorrect as e by all codewords. The number of miscorrected errors in this class is $\sum_{i=0}^{t-2} \sum_{j=1}^{t-i-1} \binom{n_1-1}{i} \binom{r_2}{j} + \sum_{i=1}^{t-1} \sum_{j=1}^{t-i-2} \binom{n_1-1}{i} \binom{r_2}{j}$. The multiplicity of e' is at least $r_2 - t + 2$.

The next Corollary derived from the above error correction algorithm shows that the number of errors that are masked or miscorrected by all codewords of the nonlinear t -error-correcting code is drastically reduced compared to that of linear BCH codes.

Corollary 4.1 Let $\{(u, Pu)\}$ be a repetition code with $k_2 = 1$ and $r_2 \geq d - 2$. Let f be a perfect nonlinear function with $P_f = 2^{-r_2}$. The nonlinear t -error-correcting code constructed as in Theorem 3.1 has only one undetectable error. The smallest possible multiplicity of errors that are miscorrected by all codewords is $r_2 - t + 2$. The total number of errors that are miscorrected by all codewords is

$$\begin{aligned} & 1 + \sum_{i=1}^t \sum_{j=0}^{t-i} \binom{k_1}{i} \binom{r_1}{j} + \sum_{i=1}^t \binom{n_1}{i} + \sum_{i=r_2-t+2}^{r_2-1} \binom{r_2}{i} \\ & + \sum_{i=1}^{t-1} \sum_{j=0}^{t-1-i} \sum_{l=1}^{t-i-j} \binom{k_1}{i} \binom{r_1}{j} \binom{r_2}{l} \\ & + \sum_{i=0}^{t-2} \sum_{j=1}^{t-i-1} \binom{n_1-1}{i} \binom{r_2}{j} + \sum_{i=1}^{t-1} \sum_{j=1}^{t-i-2} \binom{n_1-1}{i} \binom{r_2}{j}. \end{aligned}$$

The number of errors of multiplicity $t + 1$ that are miscorrected by all codewords is $2 \binom{r_2}{t-1} + \binom{r_2}{t-2}$.

Remark 4.2 The above error correction algorithm can also correct some errors of multiplicity $t + 1$ and $t + 2$. Rewrite \tilde{v} as $\tilde{v} = (\tilde{x}_1, \mathbf{0}) \oplus \tilde{x}_2 = v \oplus (e_1, \mathbf{0}) \oplus e_2$. As long as $\|(e_1, \mathbf{0}) \oplus e_2\| \leq t$, the error can be located by the linear BCH code as $\hat{e}_2 = (e_1, \mathbf{0}) \oplus e_2$. If $e_1 = e_2[n_1 - 1] = 1$ and $\|e_2\| > 1$, then $\hat{S}_2 = \mathbf{1}$ and the error belongs to case 6(b). Obviously, errors of multiplicity $t + 1$ and $t + 2$ in this class can also be corrected by the above algorithm.

Example 4.1 In this example we show the encoding and decoding procedure of a $(32, 19, 5)$ nonlinear 2-error-correcting code. Let V be a $(28, 18, 5)$ linear BCH code whose generator polynomial is $g(x) = x^{10} + x^9 + x^8 + x^6 + x^5 + x^3 + 1$. Let $\{(u, Pu)\}$ be a repetition code, where $u \in GF(2)$, $Pu \in GF(2^3)$. Select f to be a quadratic perfect nonlinear function from $GF(2^{18})$ to $GF(2^3)$ defined by $f = s_1 \bullet s_2 \oplus s_3 \bullet s_4 \oplus s_5 \bullet s_6$. Let 1010110011110100111 be the 19-bit message that needs to be encoded. Then $u = 1$, $v_k = 110110011110100111$. The redundant bits for V is 0001111111 and $Pu \oplus f(v_k) = 101$. Thereby the entire codeword is 1010110011110100111000111111101. Suppose the four left-most bits are distorted. The distorted codeword is 0101110011110100111000111111101. $\tilde{v} = 1011100111101001110001111111$. The decoder will correct the 2-bit error $\hat{e}_2 = 0110000000000000000000000000$ in \tilde{v} . After this, we re-compute S_2 . $\hat{S}_2 = P\tilde{x}_1 \oplus f(\hat{v}_k) \oplus \tilde{x}_3 = 111$, which belongs to case 6.(b). The error is

$$\begin{aligned} e &= (1, 1 \oplus \hat{e}_2[n_1 - 1], \hat{e}_2[n_1 - 2 : 0], \mathbf{0}) \\ &= 11110000000000000000000000000000. \end{aligned}$$

Thereby, an error of multiplicity four is successfully corrected although the Hamming distance of the code is only five.

In Table 2, we compare the number of errors miscorrected by all codewords for the conventional $(8262, 8192, 11)$ linear BCH code and the proposed $(8281, 8201, 11)$ nonlinear 5-error-correcting code. Denote by A_i the number of codewords of multiplicity i belonging to the linear BCH code. For every codeword c of multiplicity eleven belonging to the linear BCH code, if $e \oplus e' = c$, $\|e\| = 5$ and $\|e'\| = 6$, $\|e'\|$ will be miscorrected by all codewords as e since they have the same syndrome. Thereby, the number of errors of multiplicity six miscorrected by all codewords of the linear BCH code is $A_{11} \binom{11}{6}$. (If the error is only corrected when there is at least one information bit is distorted, this number will be a little smaller.) Similarly, the number of errors of multiplicity seven miscorrected by all codewords of the linear BCH code is $A_{11} \binom{11}{7} + A_{12} \binom{12}{7}$. According to the results presented in [21], for a $(8262, 8192, 11)$ linear shortened BCH code, A_{11} and A_{12} can be roughly estimated by $A_{11} = \binom{8262}{11} / 2^{70} \approx 10^{14}$ and $A_{12} = \binom{8262}{12} / 2^{70} \approx 10^{17}$.

The proportion F of errors that are always miscorrected by the linear BCH code can be calculated as

$$F = \frac{(2^{k_1} - 1) \sum_{i=1}^t \binom{n_1}{i}}{2^{n_1}} \approx \frac{\sum_{i=1}^t \binom{n_1}{i}}{2^{r_1}}. \quad (6)$$

For the linear $(8262, 8192, 11)$ linear BCH code, $F \approx 10^{-4}$.

The smallest possible multiplicity for errors that are always miscorrected by the proposed nonlinear t -error-

correcting is $r_2 - t + 2$. For the (8281, 8201, 11) 5-error-correcting code, $r_2 = 10$. Hence no errors of multiplicity six will be miscorrected by all codewords of the code. From Corollary 4.1, the number of errors of multiplicity seven which are always miscorrected is only 540. The total number of errors that are always miscorrected is of the order of magnitude 10^{17} . Thereby, the proportion of errors miscorrected by all codewords of the proposed nonlinear code is very close to zero.

In addition to errors that are miscorrected by all the codewords, there are also errors which are conditionally miscorrected by the nonlinear t -error-correcting codes. However, these errors usually have high multiplicities. Moreover, most of the conditionally miscorrected errors are only miscorrected with a probability of $2^{-r_2} = 2^{-10}$. Thereby, the existence of conditionally miscorrected errors will not compromise the reliability of NAND flash memories protected by the nonlinear t -error-correcting codes.

There is only one error which is undetectable by the proposed nonlinear 5-error-correcting code. All the other errors will be detected with a probability of at least $1 - 2^{-10}$ while most of them will always be detected. When error stays for several clock cycles, the probability that the error will be detected and successfully located will drastically increase.

We note that the enhanced capability of the proposed nonlinear 5-error-correcting code to detect and correct repeating errors is very helpful for MLC NAND flash memories for the protection of hardware malfunctions such as data retention and programming/erasing endurance failure [4]. Due to the decreased programming voltage margin, data retention is more likely to happen for MLC technology than for SLC technology. The problem of programming/erasing endurance also becomes more serious for MLC NAND flash memories, for which the typical number of supported program/erase cycles is fewer than 10000 [2]. Errors introduced by these hardware failures will never disappear or will only disappear after the next erasing or programming operation. Hence, the proposed nonlinear t -error-correcting code with stronger error detection and correction capability for repeating errors can be used together with other protection schemes to efficiently detect these failures and protect the devices against them.

5 Hardware Design of the Encoder and the Decoder for Nonlinear t -Error-Correcting Codes

5.1 Encoder Architecture

The encoders for linear BCH codes are conventionally implemented based on a linear feedback register (LFSR) architecture. Both the serial and the parallel architectures for LFSRs are well studied in the community. In general, the

serial LFSR needs k clock cycles while the parallel LFSR needs only $\lceil k/q \rceil$ clock cycles to finish the computation of the redundant bits at the cost of higher hardware complexity, where k is the number of information bits and q is the parallelism level of the LFSRs. Compared to the encoder

	The conventional (8262, 8192, 11) linear BCH code	The proposed (8281, 8201, 11) nonlinear code
$\ e\ = 6$	$A_{11} \binom{11}{6} \approx 10^{17}$	0
$\ e\ = 7$	$A_{11} \binom{11}{7} + A_{12} \binom{12}{7} \approx 10^{20}$	540
Fraction of miscorrected errors	$\approx 10^{-4}$	≈ 0

Table 2: Comparison of the number of miscorrected errors for the (8262, 8192, 11) linear BCH code and the proposed (8281, 8201, 11) nonlinear 5-error-correcting code

for linear BCH codes, the encoder for the proposed nonlinear t -error-correcting code mainly requires one more multiplier in $GF(2^{r_2})$ and two r_2 -bit registers. The architecture of the encoder for the nonlinear (8281, 8201, 11) 5-error-correcting code is shown in Figure 3. The design is based on the parallel LFSR proposed in [22]. The parallelism level of the design is ten. During each clock cycle, ten information bits are inputted to the encoder. The most significant bit (*msb*) of the message is inputted via a separate port. The first information bit for the linear BCH code is derived by XORing *msb* with the first bit of *msg* at the first clock cycle (when *cnt* = 0 as shown in the figure). The bottom half of the architecture is a parallel LFSR used to generate the redundant bits for linear BCH codes. D is a 10×70 binary matrix [22]. During each clock cycle, the ten most significant bits in the shift register is XORed with the new input and then multiplied by D . The output of the multiplier is XORed with the shifted data of the shift register to generate the input to the register. The top half of the architecture is for the computation of nonlinear redundant bits. During the even-numbered clock cycles, the 10-bit input is buffered. During the odd-numbered clock cycles, the buffered data is multiplied by the new input in $GF(2^{10})$ and then added to the output registers. A 10-bit mask is XORed with the data in the output register to generate the nonlinear redundant bits. For the (8281, 8201, 11) 5-error-correcting code, 820 clock cycles are required to complete the encoding of the message.

5.2 Decoder Architecture

The decoding of the nonlinear t -error-correcting codes requires the decoding of a linear BCH code with one less information bits. The standard decoder for the linear BCH

codes mainly contains three parts: the syndrome computation block, the error locator polynomial generation block and the Chien search block.

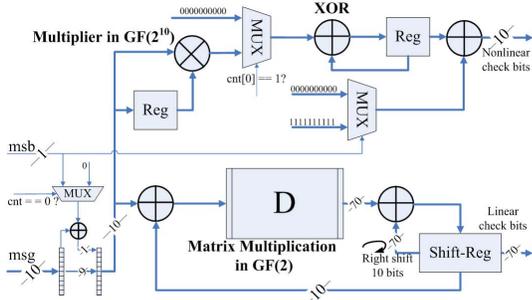


Figure 3: The architecture of the encoder for the (8281, 8201, 11) nonlinear 5-error-correcting code

5.2.1 Syndrome Computation

Without loss of generality, assume that the linear BCH code is a narrow-sense BCH code [21]. Denote by $\tilde{c} = (\tilde{x}_1, \tilde{x}_2 \cdots \tilde{x}_{n-1}, \tilde{x}_n)$ the received codeword. For a $(n, k, d = 2t + 1)$ linear t -error-correcting BCH codes, the syndromes are defined as $\hat{S}_i = \sum_{j=0}^{n-1} x_{j+1} \alpha^{ij}, 0 \leq i \leq 2t - 1$, where α is the primitive element of a finite field $GF(2^m)$. For binary linear BCH codes, $\hat{S}_{2i} = \hat{S}_i^2$. Thereby only odd-numbered \hat{S}_i needs to be computed from \tilde{c} . The other syndromes can be computed using a much simpler square circuit in $GF(2^m)$. To improve the throughput of the decoder, a parallel design can be applied to process multiple bits per clock cycle. Figure 4 shows the syndrome computation circuit with a parallelism level of q for one \hat{S}_i . t such structures are needed for the whole block. In our design the parallelism level is ten. The computation of syndromes for the linear BCH code will be finished in 827 clock cycles.

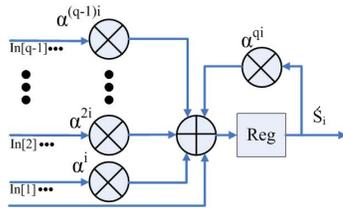


Figure 4: The syndrome computation block with a parallelism level of q for linear BCH codes

5.2.2 Error Locator Polynomial Generation

After the syndromes are computed, the error locator polynomial Λ will be generated using the Berlekamp-Massey (BM) algorithm. The hardware implementations of the BM algorithms have been well studied in the community [23] [24] [25] [26]. In our design a fully serial structure proposed in [23] is used to minimize the area overhead. The design

mainly requires three multipliers in $GF(2^m)$ and two FIFOs. $t(t + 3)/2$ clock cycles are required to generate the error locator polynomial Λ . When $t = 5$, the number of clock cycles needed is 20.

5.2.3 Chien Search

Denote by α the primitive element in $GF(2^m)$. The Chien search algorithm exhaustively tests whether α^i is a root of the error locator polynomial Λ . If $\Lambda(\alpha^i) = 0$, the error location is $2^m - 1 - i$. Rewrite $\Lambda(\alpha^i)$ as:

$$\begin{aligned} \Lambda(\alpha^i) &= \lambda_0 \oplus \lambda_1 \alpha^i \oplus \lambda_2 \alpha^{2i} \oplus \cdots \oplus \lambda_t \alpha^{ti} \\ &= \lambda_{0,i} \oplus \lambda_{1,i} \alpha \oplus \lambda_{2,i} \alpha^2 \oplus \cdots \oplus \lambda_{t,i} \alpha^t. \end{aligned}$$

The computation complexity is reduced based on the fact that $\lambda_{j,i+1} = \lambda_{j,i} \alpha^j, 0 \leq j \leq t$. The algorithm can also be paralleled to test multiple positions per clock cycle. A typical implementation of the algorithm with a parallelism level of q contains t m -bit multiplexers and registers, $q \times t$ multipliers for multiplication by a constant and q adders in $GF(2^m)$ [27]. In [28], a strength-reduced parallel Chien search architecture is proposed. The author showed that by a simple transformation of the error locator polynomial, most of the Galois field multiplications can be replaced by shift operations resulting in much lower hardware complexity (Figure 5). For the detail of the architecture, please refer to [28]. We implement the strength-reduced Chien search architecture with a parallelism level of ten. 827 clock cycles are required to complete the error locating procedure.

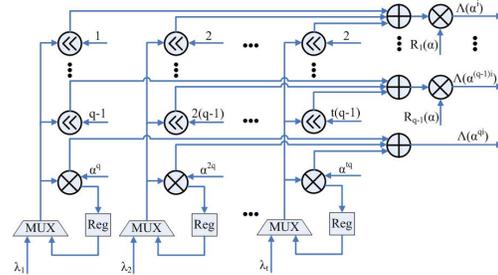


Figure 5: Strength-reduced Chien search architecture with a parallelism level of q

5.2.4 Decoder Architecture for the Nonlinear 5-Error-Correcting Code

The detailed architecture of the decoder for the proposed (8281, 8201, 11) nonlinear 5-error-correcting code is shown in Figure 6. The whole decoding procedure requires 1675 clock cycles assuming a parallelism level of ten. During the first 827 cycles, S_2 and the syndrome of the linear BCH code are computed. If no errors are detected by the linear BCH code, the decoding procedure will be completed at the 828th clock cycle. Depending on the value of S_2 , either

the first two information bits will be flipped or ERR will be pulled down by the ERR generating circuit which indicates that there are no errors occurring to the information bits of the code. The latency for the error locator polynomial generation and the Chien search will be only incurred when errors are detected by the linear BCH code, which can effectively reduce the average decoding latency.

If errors are detected by the linear BCH code, the Berlekamp-Massey algorithm will take another 20 clock cycles to generate the error locator polynomial Λ . After this the Chien search block will exhaustively test all possible error locations. If $\Lambda(\alpha^i) = 0$, then the error location is $2^m - 1 - i$. Since a (8270, 8200, 11) shortened BCH code is used, only $\Lambda(\alpha^i)$, $8114 \leq i \leq 16383$ ($m = 14$) need to be computed. The original strength-reduced Chien search architecture is slightly modified for the decoding of shortened linear BCH codes. The constant inputs to the bottom t Galois field multipliers in Figure 5 are set to be α^{-10i} instead of α^{10i} , $1 \leq i \leq t$ ($q = 10$).

\hat{S}_2 is initialized to be S_2 and is serially updated during the Chien search stage. Starting from the 848th clock cycle, the 10-bit FIFO output x_i (possibly distorted codeword) and the decoded 10-bit error vector e_i will be buffered in two 10-bit registers. At each odd-numbered clock cycle, \hat{S}_2 is updated as follows.

$$\hat{S}_2 = \hat{S}_2 \oplus x_{i-1} \bullet x_i \oplus (x_{i-1} \oplus e_{i-1}) \bullet (x_i \oplus e_i). \quad (7)$$

At the 1675 clock cycle, the value of \hat{S}_2 is used to re-check whether the most significant two bits are successfully corrected. A 2-bit error mask will be generated to make adjustment to these two bits according to the check results.

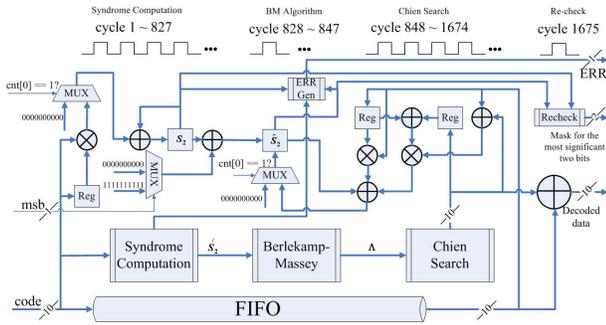


Figure 6: Decoder architecture for the proposed (8281, 8201, 11) nonlinear 5-error-correcting code

5.2.5 Area Overhead, Latency and Power Consumption

The (8262, 8192, 11) linear BCH code and the proposed (8281, 8201, 11) nonlinear code are modeled in Verilog and synthesized in cadence RTL compiler using the Nangate 45nm open cell library [29]. Both of the designs are placed and routed in cadence Encounter. In Table 3 we compare the

post-routed area overhead, latency and power consumption of the encoders and the decoders for the above two codes under the typical operation condition with a voltage supply of 1.1V and a temperature of 25C. All the designs have a parallelism level of ten. Both of the encoders can operate at 1GHz resulting in a encoding throughput of 10Gb/s. The decoders can operate at 400MHz resulting in a decoding throughput of 4Gb/s. Compared to the (8262, 8192, 11) linear BCH code, the proposed (8281, 8201, 11) nonlinear code requires the same number of clock cycles for the encoding and only one more clock cycle for the decoding. Thereby, the proposed code has the same encoding latency and only a little worse decoding latency than the linear BCH code.

For the encoder and decoder for the proposed nonlinear code, the total area overhead and power consumption are increased by 14.0% and 14.7% respectively compared to the linear BCH code, which is acceptable given the fact that the reliability of the MLC NAND flash protected by the proposed code will be largely improved compared to that protected by the linear BCH code (Table 2).

	(8262, 8192, 11) linear BCH code		(8281, 8201, 11) nonlinear code	
	Encoder	Decoder	Encoder	Decoder
Parallelism Level	10	10	10	10
Clock Speed(Hz)	1G	400M	1G	400M
Throughput	10Gb/s	4Gb/s	10Gb/s	4Gb/s
Latency (Cycles)	820	1674	820	1675
Latency (μ s)	0.82	4.185	0.82	4.1875
Area (μ m ²)	1674.6	19182.2	2765.3	21017.1
Power(mW)	1.294	5.329	2.158	5.439

Table 3: Comparison of the area, the latency and the power consumption of the (8262, 8192, 11) linear BCH code and the proposed (8281, 8201, 11) nonlinear code (Voltage = 1.1V, Temperature = 25C)

6 Conclusions

The design of reliable MLC NAND flash memories based on nonlinear t -error-correcting codes is proposed. The proposed code has much less undetectable and miscorrected errors than the conventional linear BCH codes (Table 2). The selected (8281, 8201, 11) nonlinear 5-error-correcting code has no miscorrected errors of multiplicity six, which are the most dangerous errors assuming an independent error model. Moreover, in spite of the fact that the Hamming distance of the proposed code is eleven, it can also correct some errors of multiplicity six and seven without extra hardware overhead and decoding latency resulting in further improvement of the reliability of the MLC NAND flash memories. We present the error correction algorithm, describe architectures for the encoder and decoder, implement the encoder and decoder in hardware and compare the

area overhead, the time latency and the power consumption of architectures based on the proposed code to that based on a (8262, 8192, 11) linear BCH code. The results show that our codes can achieve a much better reliability with nearly the same time latency and only a little more area overhead and power consumption than the linear BCH code (Table 3).

References

- [1] G. Atwood, A. Fazio, D. Mills, and B. Reaves, "Intel StrataFlash™ memory technology overview," *Intel Technology Journal*, 1997.
- [2] J. Cooke, "The inconvenient truths about NAND flash memory," Micron MEMCON'07 presentation, 2007.
- [3] R. Dan and R. Singer, "White paper: Implementing MLC NAND flash for cost-effective, high capacity memory," M-Systems, 2003.
- [4] R. Bez, E. Camerlenghi, A. Modelli, and A. Visconti, "Introduction to flash memory," *Proceedings of the IEEE*, vol. 91, no. 4, pp. 489–502, April 2003.
- [5] G. Cellere, S. Gerardin, M. Bagatin, A. Paccagnella, A. Visconti, M. Bonanomi, S. Beltrami, R. Harboe-Sorensen, A. Virtanen, and P. Roche, "Can atmospheric neutrons induce soft errors in NAND floating gate memories?" *Electron Device Letters, IEEE*, vol. 30, no. 2, pp. 178–180, Feb. 2009.
- [6] M. Bagatin, G. Cellere, S. Gerardin, A. Paccagnella, A. Visconti, S. Beltrami, and M. Maccarrone, "Single event effects in 1Gbit 90nm NAND flash memories under operating conditions," in *On-Line Testing Symposium, 2007. IOLTS 07. 13th IEEE International*, July 2007, pp. 146–151.
- [7] F. Irom and D. Nguyen, "Single event effect characterization of high density commercial NAND and NOR nonvolatile flash memories," *Nuclear Science, IEEE Transactions on*, vol. 54, no. 6, pp. 2547–2553, Dec. 2007.
- [8] W. Liu, J. Rho, and W. Sung, "Low-power high-throughput BCH error correction VLSI design for multi-level cell NAND flash memories," in *Signal Processing Systems Design and Implementation, 2006. SIPS '06. IEEE Workshop on*, Oct. 2006, pp. 303–308.
- [9] R. Micheloni, R. Ravasio, A. Marelli, E. Alice, V. Altieri, A. Bovino, L. Crippa, E. Di Martino, L. D'Onofrio, A. Gambardella, E. Grillea, G. Guerra, D. Kim, C. Missiroli, I. Motta, A. Prisco, G. Ragone, M. Romano, M. Sangalli, P. Sauro, M. Scotti, and S. Won, "A 4Gb 2b/cell NAND flash memory with embedded 5b BCH ECC for 36mb/s system read throughput," in *Solid-State Circuits Conference, 2006. ISSCC 2006. Digest of Technical Papers. IEEE International*, Feb. 2006, pp. 497–506.
- [10] F. Sun, S. Devarajan, K. Rose, and T. Zhang, "Design of on-chip error correction systems for multilevel NOR and NAND flash memories," *IET Circuits, Devices and Systems*, vol. 1, no. 3, pp. 241–249, 2007.
- [11] T.-H. Chen, Y.-Y. Hsiao, Y.-T. Hsing, and C.-W. Wu, "An adaptive-rate error correction scheme for NAND flash memory," in *VLSI Test Symposium, 2009. VTS '09. 27th IEEE*, May 2009, pp. 53–58.
- [12] B. Chen, X. Zhang, and Z. Wang, "Error correction for multi-level NAND flash memory using Reed-Solomon codes," in *Signal Processing Systems, 2008. SIPS 2008. IEEE Workshop on*, Oct. 2008, pp. 94–99.
- [13] Y. Cassuto, M. Schwartz, V. Bohossian, and J. Bruck, "Codes for multi-level flash memories: Correcting asymmetric limited-magnitude errors," in *Information Theory, 2007. ISIT 2007. IEEE International Symposium on*, June 2007, pp. 1176–1180.
- [14] M. Karpovsky and A. Taubin, "New class of nonlinear systematic error detecting codes," *Information Theory, IEEE Transactions on*, vol. 50, no. 8, pp. 1818–1819, Aug. 2004.
- [15] Z. Wang, M. Karpovsky, and K. Kulikowski, "Replacing linear hamming codes by robust nonlinear codes results in a reliability improvement of memories," in *Dependable Systems and Networks, 2009. DSN '09. IEEE/IFIP International Conference on*, 29 2009–July 2 2009, pp. 514–523.
- [16] K. Kulikowski, Z. Wang, and M. Karpovsky, "Comparative analysis of robust fault attack resistant architectures for public and private cryptosystems," in *Fault Diagnosis and Tolerance in Cryptography, 2008. FDTC '08. 5th Workshop on*, Aug. 2008, pp. 41–50.
- [17] J. L. Vasil'ev, "On nongroup close-packed codes," in *Probl.Kibernet.*, vol. 8, 1962, pp. 375–378.
- [18] *Wear-leveling techniques in NAND flash devices*, Micron, 2008.
- [19] E. Yaakobi, J. Ma, A. Caulfield, L. Grupp, S. Swanson, P. H. Siegel, and J. K. Wolf, "Error correcting coding for flash memories," Center for Magnetic Recording Research (CMRR), Tech. Rep., 2009.
- [20] C. Carlet and C. Ding, "Highly nonlinear mappings," *J. Complex.*, vol. 20, no. 2-3, pp. 205–244, 2004.
- [21] F. J. MacWilliams and N. J. A. Sloane, *The theory of error correcting codes*. North-Holland Pub., 1977.
- [22] T.-B. Pei and C. Zukowski, "High-speed parallel CRC circuits in VLSI," *Communications, IEEE Transactions on*, vol. 40, no. 4, pp. 653–657, Apr 1992.
- [23] H. Burton, "Inversionless decoding of binary BCH codes," *Information Theory, IEEE Transactions on*, vol. 17, no. 4, pp. 464–466, Jul 1971.
- [24] D. V. Sarwate and N. R. Shanbhag, "High-speed architectures for Reed-Solomon decoders," *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 9, no. 5, pp. 641–655, 2001.
- [25] K. Seth, K. Viswajith, S. Srinivasan, and V. Kamakoti, "Ultra folded high-speed architectures for Reed Solomon decoders," in *VLSI Design, 2006. Held jointly with 5th International Conference on Embedded Systems and Design., 19th International Conference on*, Jan. 2006, pp. 4 pp.–.
- [26] S. Rizwan, "Retimed decomposed serial Berlekamp-Massey (BM) architecture for high-speed Reed-Solomon decoding," in *VLSI Design, 2008. VLSID 2008. 21st International Conference on*, Jan. 2008, pp. 53–58.
- [27] Y. Chen and K. Parhi, "Small area parallel Chien search architectures for long BCH codes," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 12, no. 5, pp. 545–549, May 2004.
- [28] J. Cho and W. Sung, "Strength-reduced parallel Chien search architecture for strong BCH codes," *Circuits and Systems II: Express Briefs, IEEE Transactions on*, vol. 55, no. 5, pp. 427–431, May 2008.
- [29] "Nangate 45nm open cell library," <http://www.nangate.com>.