New Byte Error Correcting Codes with Simple Decoding for Reliable Cache Design[1]

Lake Bu
Reliable Computing Laboratory
Electrical and Computer Engineering
Boston University
Boston, USA
bulake@bu.edu

Mark Karpovsky
Reliable Computing Laboratory
Electrical and Computer Engineering
Boston University
Boston, USA
markkar@bu.edu

Zhen Wang
MathWorks
Boston, USA
zhen.boston@gmail.com

***Abstract*—Most cache designs support single or double bit-level error detection and correction in cache lines. However, a single error may distort a whole byte or even more, resulting in much higher decoding complexity than that of bit-level distortions. Thereby this paper proposes a new group testing based error correcting code (GTB code) for byte-level error locating and correcting which provides much stronger protection for memories. This new class of non-binary GTB codes is generated from binary superimposed codes. Since it is encoded and decoded by binary matrices, no complicated Galois Field computations in *GF*(*Q*) such as multiplications and inversions are involved. Comparing with popular non-binary error correcting codes (ECC) such as Hamming, Reed-Solomon and interleaved codes, the GTB codes achieves up to 42% reduction of the decoding complexity (hardware cost × latency) for single-byte error correction, and up to 98% reduction for double-byte error correction. Moreover, given the length of codewords (e.g. 512 bits for cache lines), as the size of each *Q*-ary digit (byte) increases, the saving increases.**

Index terms—error correction codes; fault tolerance; cache; superimposed codes; encoding; decoding.

I. Introduction

The semiconductor industry has witnessed an explosive growth of the digital memories and portable smart devices, e.g. fast caches and smart phones. As digital devices become unprecedentedly widely used in people's daily life, it becomes extremely important to protect them against errors.

Error control codes are widely used in various applications to protect devices and signals against random soft errors [1]. In a memory that is byte-organized or word-organized, each digit contains b bits. A single error can affect many bits and it is common that multiple b-bit bytes can be distorted [2], [3]. Therefore much stronger protection than bit-level error correction are required for reliable design.

Moreover, for latency/power critical applications such as cache memories and digital communications in handset devices, lower decoding complexity is indispensable [4]. Generally speaking, the decoding complexity of non-binary error correcting codes (ECC) is determined by three factors their error correcting capability m (number of defected bytes to be located and corrected), the size of the Galois field $GF(Q)$, where $Q = 2^b$, and the number of bytes in each codeword N.

As m or Q increases, more computations are needed to successfully detect, locate and correct potential errors, which results in higher decoding complexity. Most caches nowadays have a cache line size of 512 bits. For byte-level error corrections it could need finite field computations over $GF(2^8)$, $GF(2^{16})$ or $GF(2^{32})$ for byte, double-byte or word-level error corrections respectively. Therefore this paper focuses on single and double error correction on codewords (cache lines) of 64×8-bit , 32×16-bit, and 16×32-bit information digits.

Based on the above criteria, in this paper we will propose a new class of a group testing based Q-ary error correcting codes: GTB codes. The check matrices of GTB codes are generated from superimposed codes. Since the check matrix is binary, all computations for the decoding of GTB codes are just bitwise XORs. As Q increases, the computation complexity only increases proportionally to $b = \log_2 Q$, which is the number of bits of each digit in a codeword. However, popular codes such as Reed-Solomon codes all involve in computations over non-binary finite fields $GF(Q)$, whose computation complexity increases proportionally to at least b^2 when field multiplications and inversions are needed. The GTB codes decoding latency is two clock cycles, while other non-binary popular ECCs takes much more. For example, the classical non-binary error correcting Reed-Solomon (RS) codes are based on Berlekamp-Massey algorithm and Chien search decoding [5], [6], takes $2m^2+9m+3+N$ clock cycles for m errors locating and correction in an N-byte codeword [7]. The GTB codes trade extra redundancy for low decoding complexity, similar to Orthogonal Latin Square Codes (OLSCs) [8]. And it even has lower decoding complexity than that of OLSCs. These are the most significant advantages of the proposed codes over other ECC codes.

The rest of the paper is organized as follows. In Section II, the properties and constructions of superimposed codes are introduced. In Section III-A we will introduce the proposed GTB codes and their parameters. In III-B the encoding algorithm will be explained. In Section IV, we will look into single error correcting GTB codes and then compare it with classical codes such as non-binary Hamming, RS and interleaved codes. Section V introduces double-error correcting GTB code compares it with RS and interleaved codes. GTB codes in both sections achieve significant savings in decoding.

[1] This work is sponsored by the NSF grant CNS 1012910.

II. Superimposed Codes

Superimposed codes are defined by the following properties.

Definition 2.1: Let be the element in row *i* and column *j* in a binary matrix *M* of size $A \times N$. The set of columns of *M* is called an *m*-superimposed code, if for any set *T* of up to *m* columns and any single column $h \notin T$, there exists a row *k* in the matrix *M*, for which $M_{k,h} = 1$ for column *h*, and $M_{k,j} = 0$ for all $j \in T$ [9], [10].

It follows that, for all the *N* columns in *M*, the Boolean ORs of up to any *m* columns are different [11].

A. Notations

Superimposed codes can be constructed from conventional error correcting codes such as Reed-Solomon codes. We define the notations below:

n_q: the length of codewords in a *q*-ary $(n_q, k_q, d_q)_q$ code C_q.

k_q: the number of information digits of codewords in C_q.

d_q: the distance between codewords in C_q.

A: the length of codewords in a superimposed code C_{SI}.

N: the size of a superimposed code C_{SI}.

d_{SI}: the distance between codewords of C_{SI}.

m: the number of errors to be corrected. In this paper we focus on the cases when *m* = 1 and *m* = 2.

l: the maximum Hamming weight of rows in *M*.

B. Construction of Superimposed Codes

Construction 2.1: Let C_q be a $(n_q, k_q, d_q)_q$ q-ary Reed-Solomon code and *q* is not a power of 2. Each digit of C_q in *GF*(*q*) is represented by a *q*-bit binary vector with Hamming weight 1. Construct C_{SI} by substituting every *q*-ary digit of codewords in C_q by its corresponding binary vector. The *m*-superimposed code C_{SI} has the following parameters [12]:

$$\begin{aligned} A &= qn_q; \\ N &= q^{k_q}; \\ l &= q^{k_q - 1}; \\ d_{SI} &= 2d_q; \\ m &= \left\lfloor \frac{n_q - 1}{n_q - d_q} \right\rfloor = \left\lfloor \frac{n_q - 1}{k_q - 1} \right\rfloor. \end{aligned} \tag{1}$$

All the codewords of the *m*-superimposed code C_{SI} form the columns of an $A \times N$ matrix *M*. In every row there are *l* 1's and every column n_q 1's.

Example 2.1: A ternary Reed-Solomon code has its parameters $(n_q, k_q, d_q)_q = (3, 2, 2)_3$. The codewords are:

$$C_q = \{(0,0,0),(0,1,2),(0,2,1),(1,0,2), (1,1,1),(1,2,0),(2,0,1),(2,1,0),(2,2,2)\}.$$

Suppose 0, 1, 2 are represented by 3-bit binary vectors (100), (010), (001) respectively. Then the 2-superimposed code C_{SI} consisting of the following 9 codewords can be listed as the columns of a 9×9 matrix *M*.

The codewords of C_{SI} forms the columns of *M* with length $A = 9$, Hamming distance $d_{SI} = 4$ and contains the same number of codewords as C_q. According to (1), $m = 2$ and the code C_{SI} is a 2-superimposed code.

$$M = \begin{array}{c|ccccccccc|} & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 4 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 5 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 6 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 7 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 8 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 9 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \end{array}$$

There are other ways to construct superimposed codes. When $m = 1$ the columns of a binary Hamming check matrix can be selected as codewords of a 1-superimposed code. These constructions lead to simple decoding algorithms of GTB codes. This will be explained in details in Sections IV and V.

III. GTB Codes

In this section we will propose a new non-binary linear code based on check matrix *M* formed by superimposed codes. This new class of codes is called Group Testing Based Codes (GTB codes). Since for a GTB code *V*, $v \in V$ if and only if $M \cdot v = 0$, where *M* is a binary check matrix, the encoding and decoding of GTB codes only involves binary XOR operations.

We define the notations below:

M: the matrix of superimposed codes of size $A \times N$.

$M_{i,j}$: the binary element of *M* in i^{th} row and j^{th} column.

$M_{i,*}$: the i^{th} row of *M*.

$M_{*,j}$: the j^{th} column of *M*.

N: the number of digits in a $(N, K, D)_Q$ GTB codeword.

K: the number of information digits in codewords of *V*.

R: the number of redundant digits in codewords of *V*.

b: the number of bits to represent the digits (bytes) of GTB codewords in *GF*(*Q*): $b = \log_2 Q$.

λ: the maximal number of 1's in common between any two columns in *M*: $|M_{*,i} \cdot M_{*,j}| \leq \lambda$ for $i, j \in 1, 2, ..., N$, where $\cdot$ is bitwise AND [13].

With these notations we will be able to define the new linear GTB codes and theirs properties.

Definition 3.1: Let *M* be an $A \times N$ binary matrix whose columns are all codewords of a m-superimposed code. *V* is called a GTB code if $V = \{v | M \cdot v = 0\}$, $v \in GF(Q^N)$. *M* is the check matrix of *V*.

When $m = 1$ this $(N, K, 3)_Q$ GTB code *V* has length *N*, information digit length $K = N - \lceil \log_2 N \rceil$, and distance $D = 3$ which is able to correct single errors. When $m = 2$ and *M* is constructed by *Construction 2.1*, this $(N, K, 5)_Q$ GTB code *V* has length *N*, information digit length $K = N - A + 2$, and distance $D = 5$ which is able to correct double-errors.

Remark 3.1: Since the check matrix *M* of a GTB code is a binary matrix, no matter how large $Q = 2^b$ is, the syndrome computation is always as simple as additions in $GF(2^b)$ which are bitwise XORs. No multiplications or inversions in finite filed $GF(Q)$ are involved in decoding procedures. This property makes it possible to have dramatic reduce on hardware and time cost in decoding of GTB codes.

Remark 3.2: For a GTB code *V*'s check matrix *M* of size $A \times N$, since there are *A* components of a syndrome and each components involves *l* additions in $GF(2^b)$, the syndrome computation requires totally $A \cdot l$ additions (XORs). Thus for a GTB code *V* with given *m* and *K*, to achieve low decoding complexity, it is critical to minimize $A \cdot l$. Therefore we will use $A \cdot l$ as the optimization criterion in the selection on q of C_q to construct GTB codes' check matrices M.

A. The Optimal Construction of the Check Matrix for GTB codes Minimizing $A \cdot l$

Corollary 3.1: When $m = 1$ an optimal $(N, K, 3)_Q$ GTB code V can be constructed from a Hamming check matrix H where $H = [H_1, H_2, ..., H_N]$ and each H_i is a binary representation of decimal number i. The redundancy $R = \lceil \log_2(N+1) \rceil$ and the distance $D = 2m + 1 = 3$.

When $m = 2$ an optimal $(N, K, D)_Q$ GTB code V can be constructed from a superimposed code by *Construction 2.1*.

Theorem 3.1: When $m = 2$ and K is given, the optimal GTB code and its check matrix with minimum $A \cdot l$ can be constructed by the following $(n_q, k_q, d_q)_q$ Reed-Solomon code:

$$n_q = 3, k_q = 2, d_q = 2$$

$$q = \left\lceil \frac{3 + \sqrt{4K+1}}{2} \right\rceil_{p^s} \tag{2}$$

Note: $\lceil \; \rceil_{p^s}$ is the smallest integer that is the power s of a prime number p larger than the value inside $\lceil \; \rceil$.

Proof: according to *Construction 2.1*, we have $N = K + R$, $N = q^{k_q}$, $R = A - 2$, $A = q \cdot n_q$. By substituting N, R and A to the first equation we have:

$$q^{k_q} - q \cdot n_q - K + 2 = 0.$$

Also, to calculate the minimum $A \cdot l$, we have:

$$\left. \begin{array}{l} A = q \cdot n_q \\ l = q^{k_q - 1} \end{array} \right\} \Rightarrow A \cdot l = n \cdot q^{k_q}$$

Since in Reed-Solomon codes $n_q \leq q - 1$, the smaller n_q is, the smaller q can be selected. Thus finding the minimum $A \cdot l$ comes down to the problem of finding the minimum n_q and k_q.

When $m = 2$, according to (1) the smallest n_q and k_q are:

$$m = 2 = \left\lfloor \frac{n_q - 1}{n_q - d_q} \right\rfloor = \left\lfloor \frac{n_q - 1}{k_q - 1} \right\rfloor \Rightarrow \begin{cases} k_q = 2; \\ n_q = 3. \end{cases}$$

By substituting them into $q^{k_q} - q \cdot n_q - K + 2 = 0$: and solving the quadratic equation, the optimal q to achieve the minimum $A \cdot l$ is (q has to be power of a prime number):

$$q = \left\lceil \frac{3 + \sqrt{4K+1}}{2} \right\rceil_{p^s}$$

Corollary 3.2: For a given K, a double-error correcting $(m = 2)$ GTB code V over $GF(Q)$ minimizing decoding complexity has $(N, K, D)_Q = (q^2, q^2 - 3q + 2, 5)_Q$ where q is defined in (2).

Remark 3.3: From (1) and (2) it is easy to have the rate of GTB codes $= K/N = K/q^2$ and when $K \to \infty$, rate $\to 1$.

B. Encoding

Single-error correcting GTB codes can be encoded by the following Corollary.

Corollary 3.3: When $m = 1$, if $H = [H_1, H_2, ..., H_N]$ is a binary check matrix for a $(N, K, 3)_Q$ GTB code V described in *Corollary 3.1*, then by permuting the columns of H, it can be transformed to the standard form $H' = [P|I_R]$, where I_R is an $R \times R$ identity matrix. Then for $v = (v_1, v_2, ..., v_N), v \in V$, it can be encoded by $P \cdot (v_1, ..., v_K)^{\perp} = (v_{K+1}, ..., v_N)$.

Encoding double-error correcting GTB codes involves transformation of the check matrix M. For convenience we define the concept of blocks in M:

Definition 3.2: For an $A \times N$ m-superimposed code constructed by *Construction 2.1*, it can be equally partitioned into n_q sub-matrices, such that each sub-matrix has exactly q rows. Each sub-matrix is called a block B_t which is a set of q rows where $B_t = \{M_{i,*} | \lceil i/q \rceil = t\}, t \in \{1, 2, ..., n_q\}$. Each row in a block has exactly l 1's and each column has exactly one 1.

Since the sum of all rows in every block is the same: an N-bit vector of all 1's. According to (2) every M constructed from a 2-superimposed code has 3 blocks, then by removing one row from any 2 blocks all $A - 2$ rows left will be linear independent and can be transformed to standard form.

Corollary 3.4: When $m = 2$, if M is a binary check matrix for a $(N, K, 5)_Q$ GTB code V, the $A - 2$ linear independent rows of M can be transformed to a standard form $M' = [P|I_R]$, where I_R is an $R \times R$ identity matrix. Then for $v = (v_1, v_2, ..., v_N), v \in V$, the redundancy is encoded by $P \cdot (v_1, ..., v_K)^{\perp} = (v_{K+1}, ..., v_N)$.

Example 3.1: A GTB code V over $GF(Q)$ has $K = 2$ information digits and is able to correct single and double-errors. According to *Corollary 3.2* it can be constructed from the $(n_q, k_q, d_q)_q = (3, 2, 2)_3$ Reed-Solomon code. Code V has the parameters of $N = 9, A = 9, R = 7, K = 2, m = 2, l = 3$ which are the same as *Example 2.1*.

After removing $m = 2$ rows from M and transforming it into standard form (I_R is a 7×7 identity matrix) we have the encoding matrix:

$$M' = [P|I_R] = \left| \begin{array}{cc|ccccccc} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array} \right|$$

IV. Single-byte Error Correcting GTB Codes

The concept of the support of syndromes is introduced to help with single-byte error correction.

Definition 4.1: For a GTB code $V = \{v | M \cdot v = 0\}$ over $GF(Q)$, where M is an $A \times N$ binary matrix, if a codeword $v = (v_1, v_2, ..., v_N)$ is distorted by an error e to $\tilde{v} = v \oplus e$, $\tilde{v}, v, e \in GF(Q)$, the syndrome $S = M \cdot \tilde{v} = M \cdot e$. There are A components in $S = \{S(i) | i = 1, 2, ..., A\}, S(i) \in GF(Q)$. Then the support of syndrome $S_{sup} = \{S_{sup}(i) | i = 1, 2, ..., A\}$ is defined as:

$$S_{sup}(i) = \begin{cases} 0, & \text{if } S(i) = 0; \\ 1, & \text{if } S(i) \neq 0. \end{cases}$$

Corollary 4.1: According to *Corollary 3.1*, a single error correcting GTB code V has the following parameters: $(N, K, D)_Q = (N, N - \lceil \log_2(N+1) \rceil, 3)_Q$. The support of the syndrome S_{sup} is the error location and $e = S(i), S(i) \neq 0$.

The major competitors of GTB codes when $m = 1$ are non-binary Hamming codes, Reed-Solomon (RS) codes and interleaved binary Hamming codes [14]. The GTB codes and interleaved Hamming codes have $R = A = \lceil \log_2(N+1) \rceil$

as their redundancy, while non-binary Hamming codes have $R = \lceil \log_Q [(Q-1)N+1] \rceil$, and Reed-Solomon codes always $R = 2$ [15].

Non-binary Hamming codes and Reed-Solomon codes both have less redundant digits (R) than GTB codes. However, decoding of non-binary Hamming codes and Reed-Solomon codes always requires finite field multiplications and divisions. In contrary, GTB codes check matrices are always binary and so the computation of syndromes are just bitwise XORs.

In our experiments each cache line (512 bits) is treated as a codeword's information part. The digits of each codewords can be bytes in $GF(2^8)$ field, double-bytes in $GF(2^{16})$ field, or words in $GF(2^{32})$ field. And so the number of information digits are correspondingly 64, 32, 16 and m is always 1. The experimental results based on the parameters below are collected on a Xilinx Virtex4 XC4VFX60 FPGA board.

TABLE I. PARAMETERS OF GTB AND RS CODES ($m = 1$)

Code	K	R	K	R	K	R
GTB	64	7	32	6	16	5
RS		2		2		2
Hamming		2		2		2
Interleaved Hamming		7		N/A		5

The figure below shows that the GTB codes achieve less complexity over non-binary Hamming and Reed-Solomon codes because of its simple decoding computation which only requires XORs instead of finite field operations, although it costs more redundancy than Hamming and RS codes.

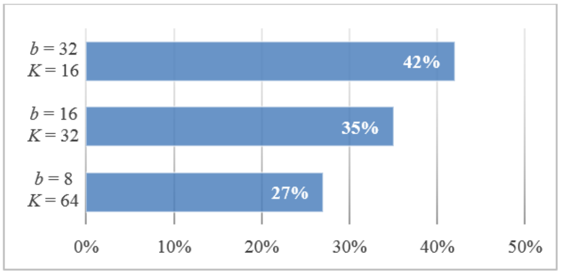

Fig. 1: Savings in complexity of decoding for GTB codes over Hamming or Reed-Solomon codes when $m = 1$. The larger the number of bits in each digit is, the greater the saving.

The GTB codes and interleaved Hamming codes both use XORs only in decoding and they have the same rate and redundancy, but the GTB codes still save more by having only one set of decoder while interleaved Hamming has to have b sets of decoders.

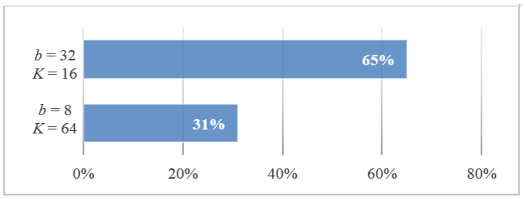

Fig. 2: Savings in complexity of decoding for GTB codes over interleaved Hamming codes when $m = 1$. The larger the number of bits in each digit is, the greater the saving is.

V. Double-byte Error Correcting GTB Codes

Double-byte error ("double-error correction" for simplicity) usually costs much more time and space to be located and corrected than single errors. However, GTB codes are very time and cost efficient in double-error corrections.

Let the columns of matrix M in size $A \times N$ be the set of all non-zero codewords of a 2-superimposed code constructed by *Construction 2.1* from a $(n_q, k_q, d_d)_q$ Reed-Solomon code C_q. Let S_{sup} be the A-bit binary vector representing the support of syndrome $S = M \cdot \tilde{v} = M \cdot e$, $e = (0, ..., 0, e_s, 0, ..., 0, e_t, 0, ..., 0)$, $s \neq t$. Let u be the error locating vector for GTB codes such that $u = S_{sup} \times M$, where $\times$ is the arithmetic multiplication and $u = (u_1, u_2, ..., u_N)$. Let U be the set of components in u and $U = \{u_i | u_i = n_q = 3\}$. Denote B_t as a block in M as in *Definition 3.2*. The GTB codes' double-error locating algorithm is introduced in the following theorem:

Theorem 5.1: If $u = 0$, $|U| = 0$, there is no error.

If $|U| = 1$, there is a single error and if $u_s \in U$ and $u_s = 3$, s is the location of the error.

If $|U| = 2$, there is a double-error and if $u_s, u_t \in U$ and $u_s = u_t = 3$, s and t are the locations of the double-error.

If $|U| > 2$, there are more than two errors.

If $u \neq 0$, $|U| = 0$, then there exists one and only one block $B_h, 1 \leq h \leq n_q$, where $\hat{S_h} = S_{sup}(q(h-1)+1) \vee S_{sup}(q(h-1)+2) \vee \cdots \vee S_{sup}(q(h-1)+q) = 0$. Find $W = \{i | u_i = 2\}$ and note that $|W| = 4$. Denote by $s, t \in W$ the indices that correspond to a pair of RS codewords in C_q whose h^{th} elements are identical. Then s and t are the error locations.

Proof: When $m = 2$, from *Definition 2.1* we know that no single column s will be covered by the bit-wise OR of up to 2 columns other than s. Moreover, the number of 1's in each column of M is exactly $n_q = 3$. We have $u_i \leq n_q$ and there are at most $m = 2$ indexes i such that $u_s = u_t = n_q = 3$. Then these two indices correspond to the error locations.

However when $m = 2$, since λ is defined as the maximal number of 1's in common between any 2 columns in M, by *Construction 2.1* and 2, $\lambda = n_q - d_q = n_q - (r_q + 1) = k_q - 1 = 1$. Thus for double-error $e = (0, ..., 0, e_s, 0, ..., 0, e_t, 0, ..., 0)$, $s \neq t$, there are chances that when $e_s = e_t$, there exists one and only one row $M_{h,*}$, such that $S(h) = M_{h,*} \cdot e = e_s \oplus e_t = 0$. This is referred to as error masking. Denote the block which contains $M_{h,*}$ as B_h, then this block can be identified according to *Definition 3.2*, where $S(i) = 0$ for all $M_{i,*} \in B_h$. Since by *Construction 2.1*, all blocks are generated from the elements of codewords in C_q, the error masking must happen between 2 codewords whose h^{th} elements are identical. Then the indices of these 2 codewords in C_q are the locations of e_s and e_t. ■

For any located double-error e_s and e_t, they always can be corrected by the algorithm in the following theorem.

Theorem 5.2: Let code v over $GF(Q)$ be distorted to $\tilde{v}$ by a double-error $e = (0, ..., 0, e_s, 0, ..., 0, e_t, 0, ..., 0)$, $s \neq t$. Also let S be the A-digit syndrome where $S = M \cdot \tilde{v} = M \cdot e$ and $S(i) \in \{S(1), S(2), ..., S(A)\}$, $S(i) \in GF(Q)$. Then for any e_s and e_t, there must exist row $M_{f,*}$ and $M_{h,*}$ in M such that:

$$\begin{vmatrix} M_{f,s} & M_{f,t} \\ M_{h,s} & M_{h,t} \end{vmatrix} = \begin{vmatrix} 1 & 0 \\ 0 & 1 \end{vmatrix}.$$

Thus $S(f) = M_{f,*} \cdot e = e_s$ and $S(h) = M_{h,*} \cdot e = e_t$. Codeword v can be corrected by $v_s = \tilde{v}_s \oplus S(f)$ and $v_t = \tilde{v}_t \oplus S(h)$.

Proof: according to *Definition 2.1*, in a matrix M whose columns are codewords of an m-superimposed code, within any set T of columns, $|T| \leq m+1$, for any column h, $h \in T$, there must exist a row k in M, where $M_{k,h} = 1$ in column h, and $M_{k,j} = 0$ for all $j \in T, j \neq h$. Since this is true for all columns in T, there exists an $(m+1) \times (m+1)$ identity sub-matrix for any given $m+1$ columns. Surely this property also applies to m columns or less. This $m \times m$ identity sub-matrix will provide the indices of the components in syndrome S which are affected by each single error only for all m errors. ■

Example 5.1: A Q-ary GTB code V has $Q = 2^3, (b = 3)$, length $N = 9$, information digits $K = 2$ and is capable to correct double-errors $(m = 2)$. Code V's check matrix is constructed by a $(n_q, k_q, d_q)_q = (3, 2, 2)_3$ Reed-Solomon code, which is the same M in *Example 2.1*.

By representing every 3-bit digit in octal, a legal message is:

$$v = (1, 2, 3, 3, 1, 2, 2, 3, 1).$$

Suppose v is distorted by a double-error at digits 4 and 5:

$$e = (0, 0, 0, 5, 7, 0, 0, 0, 0).$$

Then:

$$S = M \cdot \tilde{v} = (0, 2, 0, 5, 7, 0, 0, 7, 5);$$

$$S_{sup} = (0, 1, 0, 1, 1, 0, 0, 1, 1);$$

$$u = S_{sup} \times M = (1, 2, 1, 3, 3, 1, 2, 1, 1).$$

It is obvious that $u_4 = u_5 = n_q = 3$, $|U| = 2$, which indicates that digit 4 and 5 are distorted. To correct the double-error, according to *Theorem 5.2*, the identity matrix can be found in M:

$$\begin{vmatrix} M_{4,4} & M_{4,5} \\ M_{5,4} & M_{5,5} \end{vmatrix} = \begin{vmatrix} 1 & 0 \\ 0 & 1 \end{vmatrix}.$$

Thus we have $e_4 = S_4 = 5$ and $e_5 = S_5 = 7$. ■

For a double-error in bytes s and t, if $e_s = e_t$, there is a chance that e_s and e_t will mask each other in a component of syndrome. In this case the double-error can still be located by *Theorem 5.1*.

Example 5.2: A GTB code V with the same parameters and the same legal codeword v as *Example 5.1*. It is now distorted by a double-error at digit 4 and 5 that causes error masking:

$$e = (0, 0, 0, 7, 7, 0, 0, 0, 0).$$

Then:

$$S = M \cdot \tilde{v} = (0, 0, 0, 7, 7, 0, 0, 7, 7);$$

$$S_{sup} = (0, 0, 0, 1, 1, 0, 0, 1, 1);$$

$$u = S_{sup} \times M = (1, 2, 1, 2, 2, 1, 2, 1, 1).$$

Since $\hat{S}_1 = S_{sup}(1) \vee S_{sup}(2) \vee S_{sup}(3) = 0$ in B_1, the error masking happens in the 1^{st} element between 2 codewords in C_q. Also since $W = \{2, 4, 5, 7\}$, we find out that codewords $(1, 0, 2)_3$ and $(1, 1, 1)_3$'s 1^{st} elements are the same. Therefore the double-error is located at e_4 and e_5. Similarly to *Example 5.1*, they can be corrected by the identity matrix listed in *Theorem 5.2*. ■

The major competitors of GTB codes when $m = 2$ are Reed-Solomon codes and interleaved Orthogonal Latin Square codes (OLSC) [16]. For double-error correction, RS codes always have $R = 4$, and interleaved OLSCs have $R = 2m\sqrt{K} = 4\sqrt{k}$, while GTB codes' redundancy is $R = 3q - 2$ by *Corollary 3.2* and q is given by 2. Usually when K is small, interleaved OLSCs have less redundancy. When K is large, GTB codes have better rate.

In our experiments each cache line (512 bits) is treated as a codeword's information part (K). It can be partitioned to 64×8-bit bytes, 32×16-bit double-bytes, and 16×32-bit words.

TABLE II. PARAMETERS OF GTB AND RS CODES ($m = 2$)

Code	K	R	K	R	K	R
GTB	64	31	32	25	16	19
RS		4		4		4
Interleaved OLSC		32		N/A		16

The experimental results based on the above parameters are collected on a Xilinx Virtex4 XC4VFX60 FPGA board.

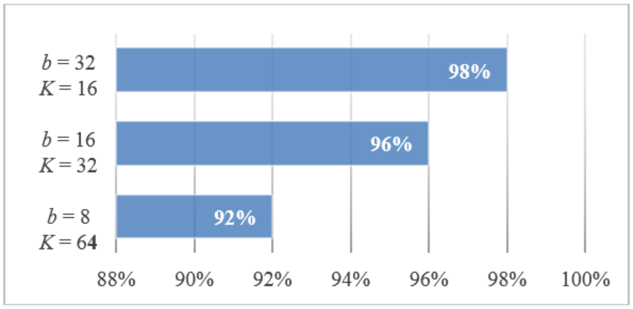

Fig. 3: Savings in complexity of decoding for GTB codes over Reed-Solomon codes when $m = 2$.

From the figure above, GTB codes considerably reduce over 98% decoding complexity from RS codes. The larger the byte size b is, the more complicated the finite field computation will be. Thus the saving increases as b increases.

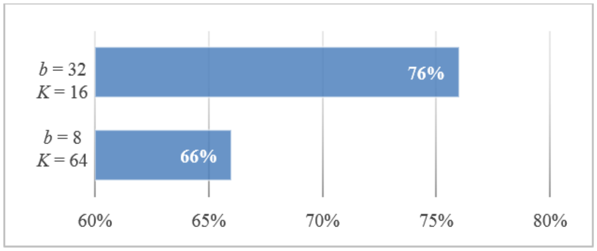

Fig. 4: Savings in complexity of decoding for GTB codes over interleaved OLSCs when $m = 2$.

The interleaved OLSCs use majority voting for error locating and the syndrome is computed by XORs only. The GTB codes also use XORs only for syndrome computation, and a matrix multiplication between two $(1 \times A)$ and $(A \times N)$ low-density binary matrices, together with a few simple controls for error locating and correcting. The overall decoding complexity of GTB codes is still much less than interleaved OLSCs. The larger the K is, the more the saving will be.

When $m = 2$, the Reed-Solomon codes have the best rate, while the interleaved OLSCs the second and GTB codes the lowest. However, in terms of the decoding complexity, the GTB codes rank the top and save significantly. Interleaved OLSCs is the second and Reed-Solomon is the worst in decoding complexity since it involves finite field computations.

VI. Conclusion

As multiple bit upsets which result in multiple byte errors become more probable with new and fast memories, stronger protection against byte-level distortions is highly demanded. Therefore we have introduced this new non-binary group testing based byte-level error correcting codes (GTB codes) and its application single and double error corrections. For codewords with digits in Galois field $GF(Q)$, the proposed new codes' decoding does not require any multiplications or inversions in Galois fields. Although the achievement is made at the cost of larger redundancy, through experiments still costs much less hardware and time than the popular codes such as Hamming, Reed-Solomon and bit-interleaved codes.

It is notable that the GTB codes are also decoded in a much faster speed than other popular codes. For instance, in double-error corrections, GTB codes usually take 2 clock cycles while Reed-Solomon codes have to take at least $29 + N$ cycles.

The comparison among GTB codes and others popular ECC can be concluded as the following:

When $m = 1$, the code's rate:

$$\text{RS} \geq \text{Hamming} > \text{Interleaved Hamming} = \text{GTB};$$

The saving on decoding complexity:

$$\text{GTB} \gg \text{Interleaved Hamming} > \text{RS} \approx \text{Hamming}.$$

When $m = 2$, the code's rate:

$$\text{RS} > \text{Interleaved OLSC} \approx \text{GTB};$$

The saving on decoding complexity:

$$\text{GTB} \gg \text{Interleaved OLSC} > \text{RS}.$$

Based on the GTB codes' fast and low complexity decoding, we suggest that this class of codes can serve as replacement of the current popular error correcting codes in reliable memory designs requiring small latency and low decoding complexity, e.g. SRAMs for cache, EEPROM and Flashes for cryptographic devices [17].

Moreover, GTB codes' can be easily generalized for locating and correcting multi-byte m errors for $m > 2$. Different from other ECCs whose decoding complexity increases significantly as m increases, the GTB codes always adopts the simple algorithm. This makes it promising to be further researched and implemented.

References

[1] Z. Wang and M. Karpovsky, "Reliable and secure memories based on algebraic manipulation detection codes and robust error correction," *Proc. Int. Depend Symp*, 2013.

[2] S. Kaneda and E. Fujiwara, "Single byte correcting and double byte detecting ecc for memory," *IEEE Transactions on Computers*, vol. c-31, no. 7, July 1982.

[3] W. Zhen, M. Karpovsky, and K. J. Kulikowski, "Replacing linear hamming codes by robust nonlinear codes results in a reliability improvement of memories," *Dependable Systems & Networks, DSN'09. IEEE/IFIP International Conference*, 2009.

[4] Z. Wang and M. Karpovsky, "New error detecting codes for the design of hardware resistant to strong fault injection attacks," *Proc. Int. Conference on Security and management, SAM*, 2012.

[5] Y. Wu, "New list decoding algorithms for reed-solomon and bch codes," *Information Theory, 2007. ISIT 2007. IEEE International Symposium*, 2007.

[6] J. Jeng and T. Truong, "On decoding of both errors and erasures of a reed-solomon code using an inverse-free berlekamp-massey algorithm," *IEEE Transactions on Communications*, no. 47.10, pp. 1488–1494, 1999.

[7] Xilinx, *LogiCORE IP Reed-Solomon Decoder*, v8.0, ds862 ed., October 19, 2011.

[8] M. Y. Hsiao, D. C. Bossen, and R. T. Chien, "Orthogonal latin square codes," *IBM Journal of Research and Development*, no. 14.4, pp. 390–394, 1970.

[9] A. G. D'yachkov, A. J. Macula, and V. V. Rykov, "On optimal parameters of a class of superimposed codes and designs," *IEEE International Symposium on Information Theory*, 1998.

[10] A. M. [10] A. G. Dyachkov and V. Rykov, "New applications and results of superimposed code theory arising from the potentialities of molecular biology," *Numbers, Information and Complexity*, pp. 265–282, 2000.

[11] W. Kautz and R. Singleton, "Nonrandom binary superimposed codes," *IEEE Transactions on Information Theory*, no. 10.4, pp. 363–377, 1964.

[12] P. Luo, A. Lin, W. Zhen, and M. Karpovsky, "Hardware implementation of secure shamir's secret sharing scheme," *High-Assurance Systems Engineering (HASE), IEEE 15th International Symposium on.*, 2014.

[13] A. G. D'yachkov and V. V. Rykov, "Optimal superimposed codes and designs for renyi's search model," *Journal of Statistical Planning and Inference*, no. 100.2, pp. 281–302, 2002.

[14] Y. Cui and X. Zhang, "Research and implemention of interleaving grouping hamming code algorithm," *IEEE International Conference on Signal Processing, Communication and Computing (ICSPCC)*, 2013.

[15] H. S. Shapiro and D. L. Slotnick, "On the mathematical theory of error-correcting codes," *IBM Journal of Research and development*, no. 3.1, pp. 25–34, 1959.

[16] G. Yalcin and et al, "Exploiting a fast and simple ecc for scaling supply voltage in level-1 caches," *IEEE On-Line Testing Symposium (IOLTS)*, 2014.

[17] S. Ge, Z. Wang, P. Luo, and M. Karpovsky, "Secure memories resistant to both random errors and fault injection attacks using nonlinear error correction codes," *ACM Proceedings of the 2nd International Workshop on Hardware and Architectural Support for Security and Privacy*, 2013.