We can now fill in the remaining values in the table:

\[
\begin{align*}
G(0) &= 0 \\
G(a) &= a \\
G(2a) &= a_1 a_2 \\
G(3a) &= a_1 a_2 a_3 \\
&\vdots \\
G(ja) &= a_1 a_2 a_3 \cdots a_j \\
&\vdots \\
G((j+s)a) &= b_1 b_2 b_3 \cdots b_j \\
G((j+s+1)a) &= b_2 b_3 b_j \cdots b_j \\
G((j+s+2)a) &= b_3 \cdots b_j \\
&\vdots \\
G((2j+s-1)a) &= b_j \\
G((2j+s)a) &= b_j
\end{align*}
\]

In particular, we have \(G((2j+s)a) = 0\), i.e., \((2j+s)a \equiv 0 \mod p^k\). So for the ideal \((a)\), we have

\[(a) = \{xa | x \in \mathbb{Z}_k\} = \{ia | i \in \mathbb{Z}\} = \{ia | i = 0, \ldots, 2j + s - 1\}.
\]

Let \(\mathcal{U} = \{x \in \mathbb{Z}_k | \text{wt}(x) = 1\}\). We want to determine how many elements are in \((a) \cap \mathcal{U}\) as \(G\) is weight-preserving

\[(a) \cap \mathcal{U} = \{x \in (a) | \text{wt}_G(x) = 1\} = 1.
\]

If \(j \geq 2\) then \((a) \cap \mathcal{U} = \{a, (2j + s - 1)a\}\). If \(j = 1\) then \((a) \cap \mathcal{U} = \{a, 2a, \ldots, (s+1)a\}\) and \(s+1 \leq (p-1)^k = p-1\). So \(|(a) \cap \mathcal{U}| \leq \max\{2, p-1\}\).

As \(a \in \mathcal{U}\) was chosen arbitrarily, we have proved that for any \(x \in \mathcal{U}\) there are at most \(\max\{2, p-1\}\) elements in \((a) \cap \mathcal{U}\).

Recall that any element \(x \in \mathbb{Z}_k\) can be written as \(x = p^u\) for some \(0 \leq u \leq k-1\) and \(u\) a unit in \(\mathbb{Z}_k\). The integer \(i\) is unique and will be denoted by \(\log_p x\). Choose an element \(b \in \mathcal{U}\) with \(\log_p b\) minimal. For any other element \(c \in \mathcal{U}\), \(\log_p c \geq \log_p b\), i.e., \(b|c\) and, therefore, \(c \in (b)\). Hence \(\mathcal{U} \subseteq (b)\). The number of elements of weight 1 in \(\mathbb{Z}_k\) and, therefore, in \(\mathbb{Z}_k\), is \(p^k-1\).

\[
k(p-1) = |k| \mid (b) \cap \mathcal{U} | \leq \max\{2, p-1\}.
\]

When \(k \geq 2\), the inequality \(|k| = \max\{2, p-1\}\) is satisfied only for \(p = k = 2\). (The other solution, \(k = 1\) and \(p\) arbitrary, corresponds to the trivial isometry between \(\mathbb{Z}_2\) and \(\mathbb{Z}_2\).)

We have proved, in particular, that for \(p = 2\) and \(k > 2\) none of the Gray maps is an isometry. Recall that a Gray map from \(\mathbb{Z}_k\) to \(\mathbb{Z}_2\) is a one-to-one map \(G\) having the property that \(G(x)\) and \(G(x+1)\) differ by exactly one bit. For weights on \(\mathbb{Z}_k\) with \(\text{wt}(1) = 1\) any isometry would in particular be a Gray map.

Acknowledgment

This correspondence was prompted by the question of C. Carlet (in his presentation of [2]) as to whether there are distance-preserving one-to-one maps from \(\mathbb{Z}_8\) to \(\mathbb{Z}_2\).

References


Decoding of Cyclic Codes Over \(\mathbb{F}_2 + u\mathbb{F}_2\)

Paramapalli Udaya, Member, IEEE, and Alexis Bonnecaze, Member, IEEE

Abstract—We give a simple decoding algorithm to decode linear cyclic codes of odd length over the ring \(R = \mathbb{F}_2 + u\mathbb{F}_2 = \{0, 1, u, \bar{u} = u + 1\}\), where \(u^2 = 0\). A spectral representation of the cyclic codes over \(R\) is given and a BCH-like bound is given for the Lee distance of the codes. The ring \(R\) shares many properties of \(\mathbb{Z}_4\) and \(\mathbb{F}_4\) and admits a linear “Gray map.”

Index Terms—Cyclic codes over rings, cyclic codes, Gray map, self-dual codes.

I. INTRODUCTION

Cyclic codes over rings have recently received a great deal of interest among coding theorists. Ring theory, as predicted earlier by McDonald [10], offers a general setting to define and study codes. The most recent success was with the ring of integers modulo four, \(\mathbb{Z}_4\), which explained the construction of certain good nonlinear codes, namely Kerdock, Preparata, and Goethals codes, through the Gray map from \(\mathbb{Z}_4\) to \(\mathbb{F}_2\) in a simple and natural way [7]. The ring \(\mathbb{F}_4\) has also been extensively studied which admits a linear Gray map but it does not lead to good binary codes. Recently, a new ring \(R = \mathbb{F}_2 + u\mathbb{F}_2\) has been considered which shares some good properties of both \(\mathbb{Z}_4\) and \(\mathbb{F}_4\) [2], [14], [6]. This alphabet is given by all binary polynomials in the indeterminate \(u\) of degree less than \(2\), and is closed under usual binary polynomial addition and multiplication modulo \(n^2\). The set of elements of \(R\) is \(\{0, 1, u, \bar{u} = u + 1\}\). It is easy to verify that \(R\) is a local ring with a maximal ideal given by \(\{0, u\}\). The multiplication and addition table for the ring is given in Table I. The multiplication table coincides with that of \(\mathbb{Z}_4\), when \(u\) and \(\bar{u}\) are replaced by, respectively, \(3\) and \(2\). In this sense, \(R\) is analogous to \(\mathbb{Z}_4\) and here \(u\) plays the role of 2. However, the addition table is different. The addition table is similar to that of the Galois field \(\mathbb{F}_4 = \{0, 1, \beta, \beta^2 = \beta + 1\}\), when \(u\) and \(\bar{u}\) are replaced, respectively, by \(\beta\) and \(\beta^2\). Note that from the definition, the characteristic of the ring is 2. Thus in the structure of alphabets, \(R\) lies between \(\mathbb{Z}_4\) and \(\mathbb{F}_4\). In [2] and [14] we described the structure of cyclic codes over \(R\) of odd length \(n\) and constructed some good self-dual codes. In this

Manuscript received July 9, 1998; revised February 1, 1999. The work of P. Udaya was supported under the ARC Grant A9701206. The material in this correspondence was presented in part at the IEEE International Symposium on Information Theory, MIT, Cambridge, MA, August 16–21, 1998.

P. Udaya is with the Department of Mathematics, City Campus, RMIT University, GPO Box 2476V, Melbourne VIC-3001, Australia.

A. Bonnecaze is with GECT, Université de Toulon et du Var, 83 400 La Garde, France.

Communicated by T. Kløve, Associate Editor for Coding Theory.

Publisher Item Identifier S 0018-9448/99/05871-X.
correspondence, we give a simple Bose–Chaudhuri–Hocquenghem-
lke (BCH-like) bound for the minimum Lee distance of these codes
by using a spectral representation on the lines of binary cyclic codes.
In this regard, we use the $r$th-degree Galois extension ring of $R$,
where $r$ is the least integer such that $n$ divides $2^r - 1$. The ring
admits a natural linear Gray map which takes a code of degree $u$
over a binary code of length $2n$. The Gray map leads to a construc-
tion of certain good nonlinear binary codes, the study of
some good binary linear codes. We give a decoding algorithm in
$R$ domain which resembles the Peterson–Gorenstein–Zierler algorithm
for binary BCH codes. This algorithm decodes up to minimum Lee
distance assured by the BCH-like bound. We show that the codes
so obtained are equivalent to $(u, u+v)$ constructed codes. The main
difference is that the component vectors $u$ and $u+v$ are interleaved in
the cyclic $R$ code construction whereas the vectors are concatenated
in $(u, u+v)$ construction. This is remarkable in the sense that certain
good $(u, u+v)$ constructed binary codes have a simple representation
as cyclic $R$ codes. Where the study of $Z_2$ cyclic codes revealed
the structure of certain good nonlinear binary codes, the study of
$R$ cyclic codes reveal the structure of good linear $(u, u+v)$ constructed
codes. In fact, some of the binary nonlinear codes obtained from
linear $Z_2$ cyclic codes can also be viewed as a generalization of
$(u, u+v)$ construction codes (see [8]).

The correspondence is organized as follows. Section II gives some
basic mathematical preliminaries and describes the structure of cyclic
codes over $R$. The spectral-domain representation and BCH-like of
bound on minimum Lee distance of $R$ cyclic codes are given in
Section III. A simple decoding algorithm for these BCH-type codes
over $R$ is given in Section IV. We conclude the correspondence
in Section V.

II. PRELIMINARIES

A. Vector Space Structure of $R$

This ring $R$ can be viewed as a vector space of dimension 2
over $F_2$. Moreover, the sets $\{0, 1\}$, $\{0, u\}$, and $\{0, \bar{u}\}$
form three subspaces in $R$ and the subspace $\{0, 1\}$ ($= F_2$) is a subring. Thus
any element of $R$ can be expressed in two different ways as

$$a = a_1 + a_2 u, \quad a_1, a_2 \in F_2$$

$$a = (a_1 + a_2) + \sigma a_2. \quad (1)$$

B. Galois Extension Ring of $R$

The method of constructing Galois rings over $R$ is similar to the
construction of Galois rings over $Z_4$. The general case of such rings
over $F_2[u]/(w(u)^k)$, $k > 1$, where $w(u)$ is an irreducible polynomial
of degree $m \geq 1$ over $F_2$ has been studied in [15]. The ring $F_2 + u F_2$
is a special case of these rings when $w(u) = u$ and $k = 2$. Let $R[x]$
be the ring of polynomials over $R$. We have a natural homomorphic
mapping from $R$ to its residue field $F_2$. For any $a \in R$, let $\bar{a}$
denote the polynomial modulo $u$. Now, define a polynomial
reduction mapping $\mu: R[x] \rightarrow F_2[x]$ in the obvious way

$$f(x) = \sum_{i=0}^{r} a_i x^i \rightarrow \sum_{i=0}^{r} \bar{a}_i x^i.$$

A monic polynomial $f$ over $R[x]$ is said to be a basic irreducible
polynomial if its projection $\mu(f)$ is irreducible over $F_2[x]$. The
Galois ring of $R$ denoted as $GR(R, r)$ is defined as $R[x]/(f(x))$, where
$f(x)$ is a basic monic irreducible polynomial of degree $r$
over $R$. Hence the ring $GR(R, r)$ is a module over $R$. The basic
monic irreducible polynomial of degree $r$ over $R$ can be lifted from
a monic irreducible polynomial over $F_2$. The trick is to consider a
monic irreducible polynomial over $F_2$ which is a subring of $R$. For
any polynomial $f(x) \in F_2[x]$, let $\bar{f}(x)$ denote the same polynomial
viewed as an element of $R[x]$. Since $F_2$ is a subring of $R$, we will
not make any distinction between $f$ and $\bar{f}$ if the context is clear. Any
irreducible polynomial over the subring is obviously irreducible over
the ring. Thus any monic irreducible polynomial $f(x)$ over $F_2$ is a
basic monic irreducible over $R$.

Note that this is not the situation in the $Z_4$ case where the
polynomial lift from the ground field $Z_4$ is nontrivial [7]. Like Galois
fields, $GR(R, r)$ is unique for a given $r$ [10]. The group of units of
$GR(R, r)$ denoted by $GR^*(R, r)$ is given by a direct product of two
groups:

$$GR^*(R, r) = G_C \times G_A$$

where $G_C$ is a cyclic group of order $2^r - 1$ and $G_A$ is an Abelian
group of order $2^r$ [15], [5].

**Lemma 1:** The set $G_C, 0$ is isomorphic to the residue field $F_{2^r}$
and is also a subspace of $GR(R, r)$. Thus the set is a subring over
$GR(R, r)$.

**Proof:** Since $G_C$ is cyclic, the set $G_C, 0$ satisfies the multi-
plicative axiom of a field. For every,

$$\alpha, \beta \in G_C, (\alpha + \beta)^{2^r} = (\alpha + \beta)$$

since the characteristic of $R$ is 2. This implies that $(\alpha + \beta)^{2^r-1} = 1$
and $(\alpha + \beta) \in \{G_C, 0\}$. Thus $G_C, 0$ is closed under addition
which proves the lemma.

Using Lemma 1, the elements of $G_A$ are given by the set

$$\{(1 + u \alpha), \alpha \in F_{2^r}\}.$$ The zero divisors of the ring $R$
are given by the elements of the maximal ideal generated by $u$, namely,

$$\{(u \alpha), \alpha \in F_{2^r}\}.$$ Thus we have

**Lemma 2:** The only ideals of $GR(R, r)$ are $(0), (1)$, and $(u)$.

Thus any element of $GR(R, r)$ can be uniquely represented as

$$a = a_1 + a_2 u, a_1, a_2 \in F_2.$$

This is analogous to the $p$-adic representation considered in [3].

The Galois automorphism group of $GR(R, r)$ is cyclic of order $r$
and is generated by the Frobenius map $\sigma$ defined by

$$\sigma(\alpha) = (\alpha^2 + u(\alpha u)^2), \quad \alpha \in GR(R, r)$$

where $\alpha$ is as in (2).

C. Cyclic Codes Over $R$

We first establish some terminology. The set $R^n$ of $n$-tuples from
$R$ is an $R$-module. By a linear code $C$ over $R$ (or a $R$-code),
we mean an additive submodule of $R^n$. Duality for codes is understood
with respect to the form $x \cdot y = \sum x_i y_i$, where $x = (x_1, x_2, \ldots, x_n)$
and $y = (y_1, y_2, \ldots, y_n)$. $C$ is said to be self-dual if $C = C^\perp$.

Two codes are equivalent if one can be obtained from the other by
permuting the coordinates and if necessary exchanging 1 and $\bar{u}$ in
certain coordinates. The Leight weight $w_C$, where $x_1, \ldots, x_n$ is
defined as $n_1(x) + 2 n_2(x)$, where $n_2(x)$ and $n_1(x)$ are, respectively,
the number of $u$ symbols and the number of 1 or $\bar{u}$ symbols in $x$. A
nonzero linear code $C$ over $\mathbb{R}$ has a generator matrix which after a suitable permutation of the coordinates can be written in the form

$$G = \begin{bmatrix} I_{r_1} & A & B \\ 0 & u & uD \end{bmatrix}$$

where $A$ and $B$ are matrices over $\mathbb{R}$ and $D$ is an $F_2$ matrix. The code $C$ then contains all codewords $[v_0, v_1]G$, where $v_0$ is a vector of length $k_1$ over $\mathbb{R}$ and $v_1$ is a vector of length $k_2$ over $F_2$. Thus $C$ contains a total of $4^{k_1}2^{k_2}$ codewords. The parameters of $C$ are given by $[n, 4^{k_1}2^{k_2}, d_{\text{min}}]$, where $d_{\text{min}}$ represents the minimum Lee distance of $C$. Following [6], we associate to the code $C$ two binary codes. The residue code $C_1$ defined as

$$C_1 = \{ x \in F_2^n | 3y \in F_2^n | x + uy \in C \}$$

and the torsion code $C_2$ defined as

$$C_2 = \{ x \in F_2^n | ux \in C \}.$$

A cyclic code of length $n$ over $\mathbb{R}$ is a linear code with the property that if $(c_0, c_1, \ldots, c_{n-1}) \in C$ then $(c_1, c_2, \ldots, c_n) \in C$. We assume that $n$ is odd and represent codewords by polynomials. Then cyclic codes are ideals in the ring $\mathbb{R}_n = \mathbb{R}[x]/(x^n - 1)$.

In [2] and [14] we have described the structure of the ideals in $\mathbb{R}_n$. The key idea which fixes the ideal structure of the ring $\mathbb{R}_n$ is the factorization of the polynomial $x^n - 1$ over $\mathbb{R}$. Since the the set $\{0, 1\}$ of $\mathbb{R}$ is a subring of $\mathbb{R}$, the factorization of $x^n - 1$ can be obtained by a trivial lift to $\mathbb{R}$. The following theorem characterizes all cyclic codes over $\mathbb{R}$.

**Theorem 1 [2]:** Suppose $C$ is a cyclic code of odd length $n$ over $\mathbb{R}$, then there are unique, monic polynomials $f, g, h$ such that $C = (f h, u f g)$, where $f g h = x^n - 1$ and $|C| = 4^{\deg f} 2^{\deg h}$ when $h = 1, C = (f)$ and $|C| = 2^{n-\deg f}$ when $g = 1, C = (u f)$ and $|C| = 2^{n-\deg f}$.

The proof of the above theorem follows on the lines exactly similar to the corresponding theorem for $\mathbb{Z}_n$ cyclic codes given in [12]. Hence, with any cyclic code over $\mathbb{R}$, the residue and torsion codes are given by

The residue code $C_1 = \mu(f h)$ of dimension $\deg(g)$

The torsion code $C_2 = \mu(f)$ of dimension $\deg(g) + \deg(h)$. (3)

Note that the code $C$ over $\mathbb{R}$ is completely determined from the residue and torsion codes, as the residue field obtained using the homomorphic mapping $\mu$ is a subring in $\mathbb{R}$. A code is free if and only if the dimension of the residue code is equal to the dimension of the torsion code.

**D. Gray Map and Binary Codes from Cyclic Codes Over $\mathbb{R}$**

An interesting aspect of $\mathbb{R}$ is that it admits a linear Gray map from $\mathbb{R}$ to $F_2^n$. For any element of $\mathbb{R}$ expressed as $x + uy$, we let

$$\phi(x + uy) = (y, x + y)$$

where $x, y \in F_2$. We extend this map in an obvious way to vectors over $\mathbb{R}$

$$\Phi(x + uy) = (y, x + y)$$

where $x, y \in F_2^n$ and $(x + uy) \in \mathbb{R}^n$. From the definitions of the Gray map and the Lee weights, the following lemma is easy to prove.

**Lemma 3 [6]:** If a code $C$ is linear or self-dual so is $\Phi(C)$. The minimum Lee weight of $C$ is equal to the minimum Hamming weight of $\Phi(C)$.

Thus a code $C = [n, 4^{k_1}2^{k_2}, d_C]$ over $\mathbb{R}$ of length $n$, $4^{k_1}2^{k_2}$ codewords with minimum Lee distance of $d_C$ gives rise to a binary code $\Phi(C) = [2n, 2k_1 + k_2, d = d_C]$. Let $C = (f h, u f g)$, where $f g h = x^n - 1$, then $\deg(h) = k_2$ and $\deg(g) = k_1$. Since $f, g,$ and $h$ are polynomials over $F_2$, the subring of $\mathbb{R}$, the code $\Phi(C)$ can be seen as equivalent to a $(u, u+\nu)$ constructed code with binary codes $C_1 = (fh)$, $n, k_1, k_2$, and $C_2 = (f) = [n, k_1 + k_2, d_C]$. The codes $C_1$ and $C_2$ are, respectively, residue and torsion codes of $C = (f h, u f g)$ (see (3)). From the definition of the Gray map and the vector space structure of $\mathbb{R}$, it is easy to see that any codeword in $\Phi(C)$ can be written as the interleaved version of $c_1$ and $(c_1 + c_2)$, where $c_1 \in C_1$ and $c_2 \in C_2$. Thus $\Phi(C)$ essentially can be obtained from the torsion and residue codes of $C$ through a $(u, u+\nu)$ construction. Note that in the $(u, u+\nu)$ construction, the component vectors $u$ and $u+\nu$ are concatenated whereas they are interleaved in our construction. Thus the cyclic codes over $\mathbb{R}$ gives another way of looking at some good classes of $(u, u+\nu)$ constructed codes. Consequently, to the authors’ best knowledge, the self-dual codes including extremal [14, 22, 8] and [64, 32, 12] codes described in [2] and [14] are new examples of codes through $(u, u+\nu)$ constructions. It is, in fact, the structure of the ring $\mathbb{R}$ which was helpful in revealing this construction. The ring description also leads to advantages in decoding with which we deal in subsequent sections.

**III. SPECTRAL REPRESENTATION OF CYCLIC CODES OVER $\mathbb{R}$**

In this section, we define on the lines of [1] a Galois ring Fourier transform for vectors over $\mathbb{R}$. Using the Galois ring Fourier transform, we give another description of cyclic codes over $\mathbb{R}$. A similar transform for vectors over more general ring $\mathbb{Z}_M$ has been used in [13] to study $\mathbb{Z}_M$ cyclic codes. The spectral domain approach makes the study of codes pedagogically simple.

**A. A Galois Ring Fourier Transform Over $\mathbb{R}$**

**Definition 1:** Let $\mathbf{v} = \{v_j\}_{0 \leq j \leq n-1}$ be a vector over $\mathbb{R}$, where $n$ divides $2^r - 1$ for some $r$, and let $\alpha$ be an element of $G_C \in F_{2^r}$ of order $n$. The Galois ring Fourier transform of the vector $\mathbf{v}$ is the vector $\mathbf{V} = \{V_j\}_{0 \leq j \leq n-1}$ given by

$$V_j = \sum_{i=0}^{n-1} \alpha^{-ij} v_i, \quad 0 \leq j \leq n-1. \quad (4)$$

Analogous to the finite-field Fourier transform [1], we call components $V_j$, $0 \leq j \leq n-1$, the transform coefficients of the vector $\mathbf{v}$ over $\mathbb{R}$. Since $n$ is relatively prime to $2^r - 1$, the order of $G_C$ in $F_{2^r}$, we have the following inversion formula for the transform coefficients:

$$v_i = \sum_{j=0}^{n-1} \alpha^{-ij} V_j, \quad 0 \leq i \leq n-1. \quad (5)$$

We can represent a vector $\mathbf{v}$ by a polynomial

$$v(x) = v_0 + v_1 x + \cdots + v_{n-1} x^{n-1}.$$  

Then using (4), $v(x)$ can be transformed into a polynomial

$$V(y) = V_0 + V_1 y + \cdots + V_{n-1} y^{n-1}.$$  

We call $V(y)$ the spectrum polynomial of $v(x)$.

**Theorem 2:** Let $a(x)$ and $f(x)$ be two polynomials corresponding to vectors $\mathbf{a}$ and $\mathbf{f}$ of length $n$ over $\mathbb{R}$, respectively. If

$$a(x)f(x) = c(x) \mod x^n - 1$$
then the transform polynomial
\[ C(y) = C_0 + C_1 y + \cdots + C_{n-1} y^{n-1} \]
is obtained using transform coefficients of \( a(x) \) and \( f(x) \) as
\[ C_{j} := F_{j} A_{j}, \quad 0 \leq j < n. \]

**Theorem 3:**

i) The polynomial \( v(x) \) has a zero at \( \alpha^{j} \) if and only if the \( j \)th frequency component \( V_{j} \) equals 0.

ii) The polynomial \( V(y) \) has a zero at \( \alpha^{-i} \) if and only if the \( i \)th component \( v_{i} \) equals 0.

As in the case of finite-field transforms [1], the transform coefficients satisfy what are called conjugacy constraints. Let \( GR(R, r) \) be the set of all vectors of length \( n \) over \( GR(R, r) \).

**Theorem 4:** Let \( V \) be a vector of length \( n \) over \( GR(R, r) \), where \( n \) divides \( 2^{r} - 1 \). Then, the inverse transform \( v \) is a vector over \( R \) if and only if the following relations are satisfied:
\[ V_{2j} = \sigma(V_{j}) = (V_{j,0})^2 + u(V_{j,1})^2; \]
where \( V_{j} = V_{j,0} + uV_{j,1}, \quad 0 \leq j < n. \)

The proofs of the Theorems 2–4 follow similarly to the proofs of their corresponding theorems in [1]. Because of the above relations, we can partition the set of integers modulo \( n \) into a collection conjugacy classes as in [1]
\[ A_{j} = \{ j, 2j, 2^{2}j, \cdots , 2^{e-1}j \} \]
where \( e \) is called the exponent, the smallest integer such that \( 2^{e}j = j \) mod \( n \) and is the order of the class \( A_{j} \).

**Example 1:** Conjugacy classes of \( n = 15 \): \( A_{0} = \{ 0 \}, A_{1} = \{ 1, 2, 4 \}, A_{2} = \{ 3, 6, 5 \}. \)

Let \( \{ A_{1}^{i}, A_{2}^{i}, \cdots , A_{t}^{i} \} \) be the set of conjugacy classes where superscripts indicate order of the conjugacy classes. The following lemma is a consequence of conjugacy relations in Theorem 4.

**Lemma 4:** Let \( v \) and \( V \) be transform pairs, then spectral coefficients belonging to a conjugacy class \( A_{d}^{i} \) can only take values from \( GR(R, e_{i}) \).

We shall define the structure \( R_{n} \) comprising of vectors over \( GR(R, r) \) of length \( n \) satisfying the conjugacy relations of Theorem 4 with two operations; pointwise addition and multiplication in \( GR(R, r) \). Now using Theorems 2–4 and Lemma 4, it is a simple exercise to prove that the ring \( R_{n} \), under usual polynomial addition and multiplication modulo \( x^{n} - 1 \), is isomorphic to the structure \( R_{n} \) under pointwise addition and multiplication. This fact leads to Theorem 5, whose proof can be done similarly to the corresponding theorem for the cyclic codes over ring \( Z_{2}^{k} \), \( p \) a prime, \( k \) a positive integer, in [13]. This isomorphism is the key to define cyclic codes on spectral domain.

**Theorem 5:** \( R_{n} \) is a direct sum of Galois rings given by \( R_{n} = \bigoplus_{t=1}^{t} GR(R, e_{i}) \), where \( t \) is the number of conjugacy classes and \( e_{i}, \quad i = 1, 2, \cdots , t \) are the exponents of the conjugacy classes.

From Lemma 2, we know that the only ideals of \( GR(R, e_{i}) \) are \( (0), (1), \) and \( (u) \). Thus in the transform domain a cyclic code can be defined as inverse fast Fourier transform (FFT) of a set of transform-domain vectors having a specified ideal \( [(0), (1), \) or \( (u)] \) in the various conjugacy classes. Thus each conjugacy class \( A_{d}^{i} \) can take values from \( (0) \) or \( uGR(R, e_{i}) \) or \( GR(R, e_{i}) \). Hence there are \( 3^{t} \) different possible ideals in \( R_{n} \) leading to distinct cyclic codes. We state the characterization in the following theorem.

**Theorem 6:** Any cyclic code is an inverse FFT of a set of elements in \( R_{n} \) with certain conjugacy classes taking values identically equal to 0 and certain other conjugacy classes taking values from \( uGR(R, e_{i}) \).

The above theorem gives a representation of a cyclic code leading to its parity-check matrix. In the standard polynomial representation, cyclic codes over \( R \) are defined as ideals in the ring \( \mathcal{R}_{n} = \mathcal{R}_{[x]/(x^{n} - 1)} \). In [2] we give a time-domain characterization of ideals in \( \mathcal{R}_{n} \). Any cyclic code over \( R \) has the following representation.
\[ C'(fh, ufg), \quad \text{where} \quad fgh = x^{n} - 1 \quad \text{and} \quad |C'| = 4^{\deg(x) / 2 \deg(h)}. \]
Using Theorem 6, we identify the connection between time domain representation and the transform domain representation as follows.

**Spectral Domain Representation of Cyclic Codes over \( R \):** Let
\[ Z_{f} = \{ a^{1}, a^{2}, \cdots , a^{f} \} \]
\[ Z_{h} = \{ a^{h}, a^{2h}, \cdots , a^{gh} \} \]
and
\[ Z_{g} = \{ a^{g}, a^{2g}, \cdots , a^{ng} \} \]
be sets of roots of polynomials \( f, h, \) and \( g \), respectively, where \( fgh = x^{n} - 1 \). Then a cyclic code \( C' = (fh, ufg) \) has the following parity-check matrix:
\[
\begin{bmatrix}
1 & a^{1} & a^{2} & \cdots & a^{(n-1)}f \\
1 & a^{2} & a^{2} & \cdots & a^{(n-1)}f \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
1 & a^{f} & a^{2f} & \cdots & a^{(n-1)g} \\
u a^{1} & u a^{2} & u a^{2} & \cdots & u a^{(n-1)g} \\
u a^{2} & u a^{2} & u a^{2} & \cdots & u a^{(n-1)g} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
u a^{h} & u a^{2h} & u a^{2h} & \cdots & u a^{(n-1)g} \\
\end{bmatrix}
\]

where \( \alpha \) is a primitive element of order \( n \) in \( GR(R, r), r \) is the least integer such that \( n \) divides \( 2^{r} - 1 \).

**B. BCH-Like Bound for Cyclic Codes Over \( R \)**

The elements of \( R \) are given by \( \{ 0, 1, u, \pi = (1 + u) \} \). The Lee weight of 1, \( \pi \) is equal to 1 and that of \( u \) is equal to 2. The Lee weight of a vector over \( R \) is defined as the sum of Lee weights of its components.

In the next lemma, we give a BCH-like bound on minimum Lee weight of a code \( C \).

**Theorem 7:** Let
\[ Z_{1} = \{ a^{1}, a^{i+1} \cdots , a^{i+1} \cdots \} \]
be \( t_{1} \) consecutive roots of the polynomial \( f \) and let
\[ Z_{2} = \{ a^{2}, a^{2+i+1}, \cdots , a^{2+i+1} \cdots \} \]
be \( (t_{1} + t_{2}) \) consecutive roots of the polynomial \( fh \). Then, the Lee distance of the code \( C \) is given by
\[ d_{L} \leq \min\{ t_{1} + t_{2} + 1, 2(t_{1} + 1) \}. \]

**Proof:** Proof follows since the code is equivalent to \( \{ (u, u+i) \} \) constructed code from two binary codes \( \{ \mu(fh) \} \) and \( \{ \mu(f) \} \).

We have given a BCH-like bound mainly to decode \( R \) cyclic codes. In general, the minimum Lee distance of an \( R \) cyclic code is given by \( \min\{ d_{R}, 2d_{T} \} \), where \( d_{R} \) and \( d_{T} \) are, respectively, the minimum distances of the residue and torsion codes of the cyclic \( R \) code. We present some examples of cyclic codes over \( R \) with good minimum distance properties in Tables II and III. In Table II, \( d_{L} \leq \min \) corresponds to the exact minimum Lee distance. We have used the results from minimum distance tables of binary cyclic codes from [11] to compute the exact \( d_{L} \). The results in Table III have been computed by the BCH-like bound in Theorem 7.
### Table II
Examples of Cyclic Codes Over $F_2 + uF_2$

<table>
<thead>
<tr>
<th>$n$</th>
<th>Zeros of $f$</th>
<th>Zeros of $h$</th>
<th>Number of codewords</th>
<th>$t_1$</th>
<th>$(t_1 + t_2)$</th>
<th>$d_{Lee}$</th>
<th>Binary Gray Image</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>1</td>
<td>0</td>
<td>4$^3$ 2</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>[14, 7, 4]</td>
</tr>
<tr>
<td>15</td>
<td>1</td>
<td>3</td>
<td>4$^{10}$ 2$^4$</td>
<td>2</td>
<td>4</td>
<td>5</td>
<td>[30, 18, 5]</td>
</tr>
<tr>
<td>15</td>
<td>3, 5</td>
<td>1</td>
<td>4$^5$ 2$^4$</td>
<td>2</td>
<td>6</td>
<td>7</td>
<td>[30, 14, 7]</td>
</tr>
<tr>
<td>15</td>
<td>1, 3</td>
<td>0, 7</td>
<td>4 2$^5$</td>
<td>4</td>
<td>9</td>
<td>10</td>
<td>[30, 9, 10]</td>
</tr>
<tr>
<td>15</td>
<td>1, 3, 5</td>
<td>7</td>
<td>4 2$^4$</td>
<td>6</td>
<td>14</td>
<td>14</td>
<td>[30, 6, 14]</td>
</tr>
<tr>
<td>31</td>
<td>1, 0</td>
<td>3, 5</td>
<td>4$^{15}$ 2$^{10}$</td>
<td>3</td>
<td>7</td>
<td>8</td>
<td>[62, 40, 8]</td>
</tr>
<tr>
<td>31</td>
<td>1, 3</td>
<td>5, 7</td>
<td>4$^{11}$ 2$^{10}$</td>
<td>4</td>
<td>10</td>
<td>10</td>
<td>[62, 32, 10]</td>
</tr>
<tr>
<td>31</td>
<td>0, 1, 3</td>
<td>5, 7</td>
<td>4$^{10}$ 2$^{10}$</td>
<td>5</td>
<td>11</td>
<td>12</td>
<td>[62, 30, 12]</td>
</tr>
<tr>
<td>31</td>
<td>1, 3, 5</td>
<td>7, 11</td>
<td>4$^6$ 2$^{10}$</td>
<td>6</td>
<td>14</td>
<td>14</td>
<td>[62, 22, 14]</td>
</tr>
<tr>
<td>31</td>
<td>0, 1, 3, 5</td>
<td>7, 11</td>
<td>4$^5$ 2$^{10}$</td>
<td>7</td>
<td>15</td>
<td>16</td>
<td>[62, 20, 16]</td>
</tr>
<tr>
<td>63</td>
<td>1</td>
<td>3</td>
<td>4$^{51}$ 2$^6$</td>
<td>2</td>
<td>4</td>
<td>5</td>
<td>[126, 108, 5]</td>
</tr>
<tr>
<td>63</td>
<td>0, 1, 3</td>
<td>5, 7, 9</td>
<td>4$^{35}$ 2$^{15}$</td>
<td>5</td>
<td>11</td>
<td>12</td>
<td>[126, 85, 12]</td>
</tr>
<tr>
<td>63</td>
<td>0, 1, 31</td>
<td>3, 5, 9</td>
<td>4$^{35}$ 2$^{15}$</td>
<td>5</td>
<td>9</td>
<td>12</td>
<td>[126, 85, 12]</td>
</tr>
<tr>
<td>63</td>
<td>1, 5, 9, 21</td>
<td>3, 7</td>
<td>4$^{34}$ 2$^{12}$</td>
<td>3</td>
<td>10</td>
<td>11</td>
<td>[126, 80, 11]</td>
</tr>
<tr>
<td>63</td>
<td>1, 5, 9, 21</td>
<td>0, 3, 7</td>
<td>4$^{33}$ 2$^{12}$</td>
<td>3</td>
<td>11</td>
<td>12</td>
<td>[126, 78, 12]</td>
</tr>
<tr>
<td>63</td>
<td>1, 3, 5</td>
<td>7, 9, 11</td>
<td>4$^{30}$ 2$^{15}$</td>
<td>6</td>
<td>12</td>
<td>13</td>
<td>[126, 75, 13]</td>
</tr>
<tr>
<td>63</td>
<td>1, 3, 5</td>
<td>0, 7, 9, 11</td>
<td>4$^{29}$ 2$^{16}$</td>
<td>6</td>
<td>13</td>
<td>14</td>
<td>[126, 74, 14]</td>
</tr>
<tr>
<td>63</td>
<td>0, 1, 3</td>
<td>5, 7, 9, 11</td>
<td>4$^{29}$ 2$^{21}$</td>
<td>5</td>
<td>13</td>
<td>12</td>
<td>[126, 79, 12]</td>
</tr>
<tr>
<td>63</td>
<td>0, 1, 3, 5</td>
<td>7, 9, 11, 13</td>
<td>4$^{23}$ 2$^{21}$</td>
<td>7</td>
<td>15</td>
<td>16</td>
<td>[126, 67, 16]</td>
</tr>
<tr>
<td>63</td>
<td>0, 1, 3, 5</td>
<td>7, 9, 11, 21</td>
<td>4$^{27}$ 2$^{17}$</td>
<td>7</td>
<td>13</td>
<td>16</td>
<td>[126, 71, 16]</td>
</tr>
<tr>
<td>63</td>
<td>1, 3, 5, 7</td>
<td>9, 11, 13, 15</td>
<td>4$^{18}$ 2$^{21}$</td>
<td>8</td>
<td>20</td>
<td>18</td>
<td>[126, 57, 18]</td>
</tr>
</tbody>
</table>

**Remark 1:** The above theorem holds in the $\mathbb{Z}_4$-cyclic codes context as well. In the $\mathbb{Z}_4$ context, the polynomials $f$, $g$, and $h$ are factors of $x^n - 1$ over $\mathbb{Z}_4$. Then, $d_{Lee}$ of the $\mathbb{Z}_4$-cyclic code is lower-bounded by $\min \{t_1 + t_2 + 1, 2(t_1 + 1)\}$. The $\mathbb{Z}_4$ versions of Kerdock, Preparata, Goethals, and Calderbank–McGuire codes are termed exceptional since the Lee distance of these codes is far greater than the lower bound.

### IV. A Decoding Algorithm

A BCH-like bound given in the last section ensures the minimum Lee weight of a cyclic $\mathbb{R}$ code. Unlike codes over $\mathbb{Z}_4$, free cyclic codes (when $h = 1, t_2 = 0$) over $\mathbb{R}$ are not interesting because of their poor minimum Lee distance. The codes are only interesting when $t_1 + t_2$ is approximately equal to $2t_1$, where $t_1$ and $t_2$ are the lengths of consecutive roots as given in Theorem 7. There are many examples where this situation is possible. On applying the Gray map, $\mathbb{R}$ cyclic codes give rise to binary linear codes with minimum Hamming distance equal to the minimum Lee distance of $\mathbb{R}$ codes. Tables II and III gives some examples of good binary codes obtained from cyclic codes over $\mathbb{R}$. In this section we decode these codes in the $\mathbb{R}$ domain.

The decoding problem: Given a received vector $r = e + e'$, the problem is to estimate either $e$ or $e'$ given that the Lee weight of $e$ is within the decoding capability of the code. To do so, we use only the traditional concept of error-location polynomials in algebraic decoding schemes and bypass error evaluation. Note that a similar decoding problem for cyclic codes over $\mathbb{Z}_4$ is much more complex [4]. Let $e(x)$ be the error polynomial associated with an error vector $e$. Here we assume that $n$ divides $2^r - 1$ and $a \in G_c$ is of multiplicative order $n$. In $\mathbb{R}$, the set $\{G_c, 0\}$ is isomorphic to the Galois field of order $2^r$ (refer to Lemma 1). Let $l_1, l_2, \ldots, l_r$ be the nonzero
TABLE III
EXAMPLES OF CYCLIC CODES OVER $\mathbb{F}_2 + u\mathbb{F}_2$ OF LENGTH 127

<table>
<thead>
<tr>
<th>$n$</th>
<th>Zeros of $f$</th>
<th>Zeros of $h$</th>
<th>Number of codewords</th>
<th>$t_1$</th>
<th>$(t_1 + t_2)$</th>
<th>$d_{\text{Lee}} \geq$</th>
<th>Binary Gray Image</th>
</tr>
</thead>
<tbody>
<tr>
<td>127</td>
<td>1</td>
<td>3</td>
<td>$4^{113}$ $2^7$</td>
<td>2</td>
<td>4</td>
<td>5</td>
<td>[254, 233, 3]</td>
</tr>
<tr>
<td>127</td>
<td>1</td>
<td>0, 3</td>
<td>$4^{112}$ $2^8$</td>
<td>2</td>
<td>5</td>
<td>6</td>
<td>[254, 233, 6]</td>
</tr>
<tr>
<td>127</td>
<td>1, 3</td>
<td>0, 5, 7</td>
<td>$4^{98}$ $2^{15}$</td>
<td>4</td>
<td>9</td>
<td>10</td>
<td>[254, 211, 10]</td>
</tr>
<tr>
<td>127</td>
<td>0, 1, 3, 5</td>
<td>5, 7, 9</td>
<td>$4^{91}$ $2^{21}$</td>
<td>5</td>
<td>11</td>
<td>12</td>
<td>[254, 203, 12]</td>
</tr>
<tr>
<td>127</td>
<td>1, 3, 5, 7, 9</td>
<td>0, 7, 9, 11</td>
<td>$4^{84}$ $2^{22}$</td>
<td>6</td>
<td>12</td>
<td>13</td>
<td>[254, 190, 13]</td>
</tr>
<tr>
<td>127</td>
<td>0, 1, 3, 5, 7</td>
<td>7, 9, 11, 13</td>
<td>$4^{77}$ $2^{28}$</td>
<td>7</td>
<td>15</td>
<td>16</td>
<td>[254, 182, 16]</td>
</tr>
<tr>
<td>127</td>
<td>0, 1, 3, 5, 7</td>
<td>9, 11, 13, 15</td>
<td>$4^{70}$ $2^{28}$</td>
<td>9</td>
<td>19</td>
<td>20</td>
<td>[254, 168, 20]</td>
</tr>
<tr>
<td>127</td>
<td>1, 3, 5, 7, 9</td>
<td>0, 11, 13, 15</td>
<td>$4^{65}$ $2^{29}$</td>
<td>10</td>
<td>21</td>
<td>22</td>
<td>[254, 155, 22]</td>
</tr>
<tr>
<td>127</td>
<td>0, 1, 3, 5, 7</td>
<td>11, 13, 15, 19</td>
<td>$4^{56}$ $2^{35}$</td>
<td>11</td>
<td>23</td>
<td>24</td>
<td>[254, 147, 24]</td>
</tr>
<tr>
<td>127</td>
<td>0, 1, 3, 5, 7</td>
<td>7, 9, 11, 13</td>
<td>$4^{49}$ $2^{35}$</td>
<td>13</td>
<td>27</td>
<td>28</td>
<td>[254, 133, 28]</td>
</tr>
<tr>
<td>127</td>
<td>1, 3, 5, 7, 9</td>
<td>13, 15, 19, 23</td>
<td>$4^{42}$ $2^{36}$</td>
<td>14</td>
<td>29</td>
<td>30</td>
<td>[254, 120, 30]</td>
</tr>
<tr>
<td>127</td>
<td>0, 1, 3, 5, 7</td>
<td>7, 9, 11, 13</td>
<td>$4^{42}$ $2^{35}$</td>
<td>15</td>
<td>31</td>
<td>32</td>
<td>[254, 112, 32]</td>
</tr>
<tr>
<td>127</td>
<td>1, 3, 5, 7, 9</td>
<td>15, 19, 21, 23</td>
<td>$4^{35}$ $2^{42}$</td>
<td>18</td>
<td>42</td>
<td>38</td>
<td>[254, 100, 38]</td>
</tr>
<tr>
<td>127</td>
<td>0, 1, 3, 5, 7</td>
<td>9, 11, 13, 15</td>
<td>$4^{28}$ $2^{42}$</td>
<td>19</td>
<td>43</td>
<td>40</td>
<td>[254, 98, 40]</td>
</tr>
<tr>
<td>127</td>
<td>0, 1, 3, 5, 7</td>
<td>11, 13, 15, 19</td>
<td>$4^{28}$ $2^{35}$</td>
<td>21</td>
<td>43</td>
<td>44</td>
<td>[254, 91, 44]</td>
</tr>
<tr>
<td>127</td>
<td>1, 3, 5, 7, 9</td>
<td>21, 23, 27, 29</td>
<td>$4^{22}$ $2^{35}$</td>
<td>22</td>
<td>46</td>
<td>46</td>
<td>[254, 79, 46]</td>
</tr>
<tr>
<td>127</td>
<td>0, 1, 3, 5, 7</td>
<td>11, 13, 15, 19, 21</td>
<td>$4^{21}$ $2^{35}$</td>
<td>23</td>
<td>47</td>
<td>48</td>
<td>[254, 77, 48]</td>
</tr>
<tr>
<td>127</td>
<td>1, 3, 5, 7, 9</td>
<td>13, 15, 19, 21</td>
<td>$4^{14}$ $2^{35}$</td>
<td>27</td>
<td>55</td>
<td>56</td>
<td>[254, 63, 56]</td>
</tr>
<tr>
<td>127</td>
<td>0, 1, 3, 5, 7</td>
<td>15, 19, 21, 23</td>
<td>$4^{7}$ $2^{28}$</td>
<td>31</td>
<td>63</td>
<td>64</td>
<td>[254, 42, 64]</td>
</tr>
</tbody>
</table>

positions of $e$. Then $e(x)$, by using the vector representation of $R$ over GF$(2)$, can be written as

$$e(x) = (e_{1,0} + u e_{1,4}) x^1 + (e_{2,0} + u e_{2,4}) x^2 + \cdots + (e_{t,0} + u e_{t,4}) x^t$$

(6)

where $e_{i,j}, i = 1, \ldots, t; j = 0, 1$ belongs to GF$(2)$. Then the error locator polynomial $\Lambda(y)$ corresponding to errors in positions $t_1, t_2, \ldots, t$ is defined as

$$\Lambda(y) = 1 + \Lambda_{1y} y + \cdots + \Lambda_{t-1y} y^{t-1} + \Lambda_{ty} y^t$$

$$= (1 - y^{\alpha^1})(1 - y^{\alpha^2}) \cdots (1 - y^{\alpha^t}).$$

(7)

Note that $\Lambda(y)$ is a polynomial over $\mathbb{F}_2[x]$, the subring of GR$(R, r)$. Also, the definition of $\Lambda(y)$ does not depend on the type of errors. Note that the degree of $\Lambda(y)$ need not be equal to the Lee weight of the error, whereas in the binary case, the degree of the locator polynomial is equal to the number of errors. In fact, the degree of $\Lambda(y)$ depends on the type of errors. We circumvent this problem by defining three binary error vectors for any $e(x)$ over $R$ in the following way. For any polynomial $e(x) = \sum_{i=0}^{n-1} e_i x^i$ over $R$, let $e_{1,\pi}(x)$ be the binary polynomial such that for all $i = 0, 1, \ldots, n-1$, the coefficient of $x^i$ in $e_{1,\pi}(x)$ is equal to 1 if $e_i$ is either 1 or $\pi$ and is equal to 0 otherwise. Similarly, the polynomials $e_{s,\pi}(x)$ and $e_{u,\pi}(x)$ are defined. The polynomials can be expressed as follows using (6):

$$e_{1,\pi}(x) = e_{1,0} x^1 + e_{2,0} x^2 + \cdots + e_{t,0} x^t + e_{2,4} x^{2t} + \cdots + e_{t,4} x^{tt}$$

$$e_{s,\pi}(x) = (e_{1,0} + e_{1,4}) x^1 + (e_{2,0} + e_{2,4}) x^{2t} + \cdots + (e_{t,0} + e_{t,4}) x^{tt}$$

$$e_{u,\pi}(x) = e_{1,4} x^1 + e_{2,4} x^{2t} + \cdots + e_{t,4} x^{tt}.$$  

(8)

By using the vector space representation of $R$, it is easy to see that

$$e(x) = e_{1,\pi}(x) + u e_{s,\pi}(x) + e_{u,\pi}(x).$$

(9)

Note that, in general, the number of nonzero positions of the binary error polynomials defined above is less than or equal to that of $e(x)$. We have the following lemma which can be proved using the definition of Lee weight over $R$. 


Lemma 5: Let \( e(x) \) be an error polynomial over \( R \) and \( e_{u,v}, e_{u,v}, e_{u,v}, e_{u,v} \) be its associated binary error polynomials as defined above. Let \( W_t(e(x)) \leq t \), then we have for the Hamming weights of corresponding binary associated polynomials:

- \( W_t(e_1,\pi(x)) \leq t \).
- Either \( W_{2t}(e_u,\pi(x)) \leq \lfloor t/2 \rfloor \) or \( W_t(e_1,\pi(x)) \leq \lfloor t/2 \rfloor \), where \( \lfloor t \rfloor \) represents the largest integer less than or equal to \( t \).

Now corresponding to the above binary error polynomials we define error locator polynomials as in (7). Let \( \Lambda_u(y) \) be the error locator polynomial for errors of type 1, \( u \in e \). Similarly, we define \( \Lambda_u,\pi(y) \) and \( \Lambda_1,\pi(y) \). In the sequel we give procedures to compute \( \Lambda_{u,v}(y), \Lambda_{u,\pi}(y), \) and \( \Lambda_1,\pi(y) \).

**Syndromes:** Without loss of generality we can assume that for a code \( C = \{ f h, u f g \} \) the sets

\[
Z_1 = \{ \alpha^{i_1}, \alpha^{i_1+1}, \ldots, \alpha^{i_t+1-1} \}
\]

and

\[
Z_2 = \{ \alpha^{i_2}, \alpha^{i_2+1}, \ldots, \alpha^{i_t+t-1} \}
\]

are the sets of consecutive roots of polynomials \( f \) and \( fh \), respectively. Then the parity-check matrix for the code takes the form of the expression at the bottom this page.

Similar to (8), for any codeword polynomial \( c(x) \) over \( R \) we can associate three binary words \( c_1,\pi(x), c_{u,v}(x), \) and \( c_{u,\pi}(x) \). Again, by the vector space representation of \( R \), we have

\[
e(x) = c_1,\pi(x) + u c_{u,\pi}(x).
\]

The parity-check matrix given above implies the following:

\[
c_1,\pi(\alpha^{i+1}) = 0, \quad 1 \leq i \leq t + 2
\]

\[
c_{u,\pi}(\alpha^{i+1}) = c_{u,v}(\alpha^{i+1}), \quad 1 \leq i \leq t.
\]

Thus we can compute the syndromes using the equation

\[
S = r H^T = (e + e) H^T = e H^T.
\]

The syndromes alternatively can be evaluated by obtaining the received word at the roots given in the sets \( Z_1 \) and \( Z_2 \) as follows:

\[
S_1 = r(\alpha^{i+1}) = e(\alpha^{i+1}), \quad 1 \leq i \leq t + 1
\]

\[
u S_1 = r(\alpha^{i+1}) = e(\alpha^{i+1}), \quad 1 \leq i \leq t + 1.
\]

Using (9) the above syndromes can be written as

\[
S_1 = e_1,\pi(\alpha^{i+1}), \quad 1 \leq i \leq t + 1
\]

\[
S_1 = e_{u,\pi}(\alpha^{i+1}), \quad 1 \leq i \leq t + 1
\]

\[
\hat{S}_1 = e_1,\pi(\alpha^{i+1}) + e_{u,\pi}(\alpha^{i+1}), \quad 1 \leq i \leq t + 1.
\]

Note that the error polynomials \( e_1,\pi, e_{u,\pi}, \) and \( c_{u,v} \) are binary polynomials and, respectively, they are related to the three sets of syndromes defined above through the respective error locator polynomials. Let \( \Lambda_1,\pi(y) \) be the error locator polynomial for the errors of type 1 and \( \pi \) by

\[
\Lambda_1,\pi(y) = 1 + \Lambda_1 y^{1} + \Lambda_1 y^{1} + \Lambda_1 y^{1} = (1 - y^{1}y^{1} + \ldots + y^{1}y^{1}).
\]

By following the procedure of Peterson–Gorenstein–Zierler [1] to decode binary cyclic codes, the syndromes \( \{ S_{u,v}, 1 \leq u \leq t + 2 \} \) are related by the equation

\[
S_{u,v} = -\sum_{i=1}^{t} \Lambda_{u,v}(y), \quad j = 1, 2, \ldots, t. \tag{12}
\]

Thus calculating error locator polynomials resembles the decoding procedure for binary cyclic codes. The error locator polynomial can be computed by solving the following key equation over \( F_{2^n} \):

\[
(\hat{S}_1, \hat{S}_1, \hat{S}_1, \hat{S}_1, \hat{S}_1) \Lambda_1,\pi(y) = \Lambda_1,\pi(y) \mod y^{1+1}, \quad \deg (\Lambda_1,\pi(y)) < \deg (\Lambda_1,\pi(y)). \tag{13}
\]

Similar relations hold for the other two sets of syndromes corresponding to polynomials \( \Lambda_{u,v}(y) \) and \( \Lambda_{u,\pi}(y) \). Hence the polynomials \( \Lambda_{u,\pi} \) and \( \Lambda_{u,\pi}(y) \) can be obtained by solving the following two key equations:

\[
(\hat{S}_1, \hat{S}_1, \hat{S}_1, \hat{S}_1, \hat{S}_1) \Lambda_{u,\pi}(y) = \Lambda_{u,\pi}(y) \mod y^{1+1}, \quad \deg (\Lambda_{u,\pi}(y)) < \deg (\Lambda_{u,\pi}(y)). \tag{14}
\]

\[
(\hat{S}_1 + \hat{S}_2 + \hat{S}_1 + \hat{S}_1, \hat{S}_1) \Lambda_{u,v}(y) = \Lambda_{u,v}(y) \mod y^{1+1}, \quad \deg (\Lambda_{u,v}(y)) < \deg (\Lambda_{u,v}(y)). \tag{15}
\]

Now we give the basis of decoding algorithm by giving a result similar to [1, Theorem 7.2.2]. Let \( M(S_1, S_2, \ldots S_{\mu}) \) be a Hankel matrix over \( F_{2^n} \) of GR(\( R, r \)) as follows:

\[
M(S_1, S_2, \ldots S_{\mu}) = \begin{bmatrix}
S_1 & S_2 & \cdots & S_{\mu} \\
S_2 & S_3 & \cdots & S_{\mu+1} \\
\vdots & \vdots & \ddots & \vdots \\
S_{\mu} & S_{\mu+1} & \cdots & S_{2\mu-1}
\end{bmatrix}
\]

Further let

\[
M_{1,\pi} = M(S_{1,0}, S_{2,0}, \ldots S_{\mu,0})
\]

\[
M_{u,\pi} = M(S_{1,u}, S_{2,u}, \ldots S_{\mu,u})
\]

and

\[
M_{1,u} = M(\hat{S}_{1,1}, \hat{S}_{2,1}, \ldots \hat{S}_{2\mu,1})
\]

**Theorem 8:** The matrix \( M_{1,\pi} \) over \( F_{2^n} \) of syndromes, is nonsingular, if \( \mu \) is equal to \( t \), the number of 1 or \( \pi \) errors have actually occurred. The matrix is singular if \( \mu \) is greater than \( t \). The theorem applies to matrices \( M_{u,\pi} \) and \( M_{1,u} \) also except that \( t \) refers to the number of actual errors which have occurred are of type \( u \), \( \pi \) and 1, \( \pi \), respectively.

**Proof:** Similar to [1, Proof of Theorem 7.2.2].

The above theorem gives the basis for decoding \( R \)-cyclic codes. The syndrome equations can be solved using the Berlekamp–Massey algorithm [9] or Euclid’s algorithm. Equation (13) gives the correct result if the number of errors of type 1, \( \pi \) is less than or equal to
Then the inverse of the roots of \( \Lambda_1, \pi(y) \) gives the position of 1, \( \pi \) errors as in the case of decoding cyclic codes [1]. That is, if \( \alpha^{-1} \) is a root of the \( \Lambda_1, \pi(y) \), then at \( i \) th position there is an error 1 or \( \pi \). Let \( P_{1, \pi} \) be the positions of 1 or \( \pi \) errors. The roots of \( \Lambda_{u, \pi}(y) (P_{u, \pi}) \) and \( \Lambda_{1, u}(y) (P_{1, u}) \) gives, respectively, the positions of \( u, \pi \) and 1, \( u \) errors. Thus if we could determine any two of the sets \( P_{1, \pi}, P_{u, \pi}, \) and \( P_{1, u} \), we can complete the decoding. The situation is best explained in Fig. 1. For example, if \( P_{1, \pi} \) and \( P_{u, \pi} \) are determined, the positions of \( \pi \) errors are given by the intersection of the positions of \( P_{1, \pi} \) and \( P_{u, \pi} \). Then \( P_{1, \pi} \setminus P_{u, \pi} \) and \( P_{u, \pi} \setminus P_{1, \pi} \) gives, respectively, the positions of 1 and \( u \) errors, where \( \setminus \) represents the set-theoretic subtraction. The maximum possible Lee errors we can correct in this case is \( \left[ \frac{t_1 + t_2}{2} \right] \). Recall that the Lee weight of symbols 1 and \( \pi \) is 1 and that of \( u \) is 2. Hence, we can correct at most \( \left[ \frac{t_1 + t_2}{2} \right] \) errors of type 1, \( \pi \) and \( \left[ \frac{t_1 + t_2}{2} \right] \) errors of type \( u \). We have the following result.

**Theorem 9:** It is possible to completely decode an error \( e(x) \) if the \( W_L(e(x)) \) is less than or equal to \( \left[ \frac{t_1 + t_2}{2} \right] \).

**Proof:** Since the degree of the syndrome polynomial in (13) is \( t_1 + t_2 \), the solution of (13) is guaranteed and the roots of the error locator polynomial \( \Lambda_{1, \pi}(y) \) give the positions of 1 or \( \pi \) errors. But, we cannot yet distinguish between 1 and \( \pi \) errors. The solutions of (14) and (15) are guaranteed only if the degree of the corresponding error locator polynomials is less than or equal to \( \left[ \frac{t_1 + t_2}{2} \right] \). From Lemma 5, this condition is satisfied for one of (14) or (15). Thus in fact we can know the error positions of 1, \( \pi \) errors and error positions of 1, \( u \) or \( \pi \) errors. In any case, the error positions of 1 and \( \pi \) can be distinguished, which completes the decoding.

Note that even though an error vector is within the decoding capability of the code, the degree of one of the error locator polynomials \( \Lambda_{u, \pi}(y) \) and \( \Lambda_{1, u}(y) \) can be more than \( t_1 \). For example, for an error of type \( 1^t \pi^u \), we have degree \( \left( \Lambda_{1, u}(y) \right) > t_1 \) and hence \( \Lambda_{1, u}(y) \) cannot be determined from (15). To test the validity of theselocator polynomials, we use the familiar trick for checking the validity of binary error locator polynomials. It is known that all the known syndromes are recursively related according to (12), for \( j = 0, 1, \ldots, n - 1 \). Thus after solving any of the key equations (refer to (13)-(15)) for the \( \Lambda(y) \), some known syndromes other than those used in (13)-(15) could be checked using (12). If the check fails we declare that the particular \( \Lambda(y) \) is not valid. We summarize the decoding below.

**Decoding Procedure:**

1. Calculation of \( t_1 + t_2 \) syndromes \( S_0 = \{ S_{i, 0}, 1 \leq i \leq t_1 + t_2 \} \) and \( t_1 \) syndromes \( S_1 = \{ S_{i, 1}, 1 \leq i \leq t_1 \} \) according to (11).
2. Calculation of error locator polynomials: Find \( \Lambda_{1, \pi}(y) \) using (13) and \( \Lambda_{u, \pi}(y) \) using (14).
3. Check the validity of \( \Lambda_{1, \pi}(y) \) and \( \Lambda_{u, \pi}(y) \) according to (12).
4. If \( \Lambda_{1, \pi}(y) \) is not valid then declare that more number of errors have occurred and stop.
5. At this point at least any two of \( \Lambda_{1, \pi}(y), \Lambda_{u, \pi}(y), \) and \( \Lambda_{1, u}(y) \) are valid. Compute sets of error positions corresponding to any two valid error locator polynomials. This means that any two sets among the sets \( P_{1, \pi}, P_{u, \pi}, \) and \( P_{1, u} \) can be determined. The positions of 1, \( u \), and \( \pi \) are computed using the appropriate equations given below

   \[
   P_{1} = P_{1, \pi} \cap P_{1, u} = P_{1, \pi} \setminus P_{u} = P_{1, u} \setminus P_{u} \\
   P_{\pi} = P_{1, \pi} \cap P_{u, \pi} = P_{1, \pi} \setminus P_{1, u} = P_{u, \pi} \setminus P_{1, u} \\
   P_{u} = P_{1, u} \cap P_{u, \pi} = P_{1, u} \setminus P_{1, \pi} = P_{u, \pi} \setminus P_{1, \pi} \tag{16}
   \]

   where the symbols \( \cap \) and \( \setminus \) respectively, denote set intersection and subtraction.

   Apart from the errors with Lee weight less than or equal to \( t_1 + t_2 / 2 \), the code by virtue of its structure can correct some errors with Lee weight exceeding the bound. The proofs of the following two lemmas run similar to that of Theorem 9.

**Lemma 6:** The code can correct a maximum of \( \left[ \frac{t_1 + t_2}{2} \right] \) errors of type 1, \( \pi \) and additionally can correct a maximum of \( \left[ \frac{t_1 + t_2}{2} \right] \) or \( \pi \) type \( u \) errors, where \( n_u \) is the number of actual \( u \) errors.

**Lemma 7:** If \( n_u \) is the actual number of \( u \) errors, then the code can correct a maximum of \( \left[ \frac{t_1 + t_2}{2} \right] \) errors of type 1, \( \pi \) provided either \( n_{\pi} \leq \left[ \frac{t_1 + t_2}{2} \right] - n_u \) or \( n_{\pi} \leq \left[ \frac{t_1 + t_2}{2} \right] - n_u \).

We explain the procedure through an example.

**Example 2:** A cyclic code of length 31 over \( R = \langle f, h, u f g \rangle \) is given by

   \[
   f = f_{1} f_{2} f_{3}, \quad g = f_{1} f_{3} f_{0}, \quad h = f_{2} f_{3}, \quad f_{1} h = x^{31} - 1, \quad f_{1} = x^{2} + x^{2} + 1, \quad f_{2} = x^{2} + x^{2} + x^{2} + x^{3} + 1, \quad f_{3} = x^{2} + x^{2} + x^{2} + x^{2} + x^{4} + 1, \quad f_{2} = x^{2} + x^{2} + x^{2} + x^{4} + x^{4} + x + 1, \quad f_{1} = x^{2} + x^{2} + x^{4} + x^{4} + 1, \quad f_{0} = x + 1.
   \]

   The sets of consecutive roots of the code are given by

   \[
   Z_{1} = \{ \alpha^{1}, \alpha^{2}, \alpha^{3}, \alpha^{4} \} \quad Z_{2} = \{ \alpha^{1}, \alpha^{2}, \ldots, \alpha^{10} \}.
   \]

   Here \( t_1 = 4 \) and \( t_1 + t_2 = 10 \). The minimum Lee distance of the code is equal to 10. The calculations for different set of error vectors are shown below. We solve the key equations (13)-(15) in this case using either Euclidean or Berlekamp–Massey algorithm. We use the symbol \# when there is no solution to the key equation. To find out the actual error positions of 1, \( u \), and \( \pi \) errors, it is sufficient to obtain the positions of any two of the sets \( P_{1, u}, P_{u, \pi}, \) and \( P_{1, \pi} \). Table IV illustrates the working of the algorithm for errors indicated in Theorem 9 and Lemmas 6 and 7.

   The algorithm can be extended to any cyclic codes over the ring \( F_{p}[\pi]/\pi^{2} \), where \( F_{p} \) is a prime field. In this case, apart from finding the component error locator polynomials we need to evaluate the component error magnitudes. Distinguishing error positions in the basis 1 and \( u \) remains the same as that explained in the case of \( R \).

**Remark 2:** Cyclic codes over \( F_{p} + u F_{p} \) are naturally suited to correct burst errors. For example, errors of type \( 1^{t_1} \pi^{t_2} u \) of Lee weight \( \left[ \frac{t_1 + t_2}{2} \right] \) is a burst error of length \( t_1 + t_2 \). Moreover, Lemmas 6 and 7 assure correction of much longer error patterns.
TABLE IV  
WORKING OF DECODING ALGORITHM FOR EXAMPLE 2

<table>
<thead>
<tr>
<th>Sl.No</th>
<th>1.</th>
<th>2.</th>
</tr>
</thead>
<tbody>
<tr>
<td>$e(x)$</td>
<td>$1 + x^2 + x^3 + u(x^4 + x^5)$</td>
<td>$1 + x + x^2 + x^3 + u(x^4 + x^5)$</td>
</tr>
<tr>
<td>$S_0$</td>
<td>${\alpha^2, \alpha^4, \alpha^8, \alpha^{21}, \alpha^{17}, \alpha^{10}, \alpha^{16}, \alpha^{13}, \alpha^{11}}$</td>
<td>${\alpha^{23}, \alpha^{15}, \alpha^{25}, \alpha^{30}, \alpha^5, \alpha^{19}, \alpha^4, \alpha^{29}, \alpha^{17}, \alpha^{13}}$</td>
</tr>
<tr>
<td>$S_1$</td>
<td>${\alpha^9, \alpha^{18}, \alpha^{23}, \alpha^6}$</td>
<td>${\alpha^{23}, \alpha^{13}, \alpha^{10}, \alpha^{26}}$</td>
</tr>
<tr>
<td>$S$</td>
<td>${\alpha^{24}, \alpha^{17}, \alpha^{22}, \alpha^3}$</td>
<td>${\alpha^{24}, \alpha^{17}, \alpha^{22}, \alpha^3}$</td>
</tr>
<tr>
<td>$\Lambda_{1,\bar{u}}(y)$</td>
<td>$\alpha^y y^4 + \alpha^{26} y^3 + \alpha^{17} y^2 + \alpha^3 y + 1$</td>
<td>$\alpha^y y^4 + \alpha^{26} y^3 + \alpha^{17} y^2 + \alpha^3 y + 1$</td>
</tr>
<tr>
<td>$P_{1,\bar{u}} : {\text{Roots}}$</td>
<td>${\alpha^0, \alpha^5}$</td>
<td>${\alpha^0, \alpha^5}$</td>
</tr>
<tr>
<td>$\Lambda_{u,\bar{u}}(y)$</td>
<td>$\alpha^{23} y^2 + \alpha^2 y + 1$</td>
<td>$\alpha^{23} y^2 + \alpha^2 y + 1$</td>
</tr>
<tr>
<td>$P_{u,\bar{u}} : {\text{Roots}}$</td>
<td>${\alpha^{23}}$</td>
<td>${\alpha^{23}}$</td>
</tr>
<tr>
<td>$\Lambda_{1,\bar{u}}(y)$</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>Positions of 1 error</td>
<td>0, 5</td>
<td>0, 1, 2, 3</td>
</tr>
<tr>
<td>Positions of $\bar{u}$ error</td>
<td>No error</td>
<td>No error</td>
</tr>
<tr>
<td>Positions of $u$ error</td>
<td>9</td>
<td>4, 5</td>
</tr>
</tbody>
</table>

V. CONCLUSIONS

We described a simple decoding algorithm for cyclic codes over the ring $F_2 + uF_2$. The algorithm extends ideas in Peterson–Gorenstein–Zierler decoder for BCH codes to correct Lee errors in BCH-type codes over $F_2 + uF_2$. We show that the linear Gray images of these codes are equivalent to $(u, u+v)$ constructed codes where both $u$ and $v$ parts belong to cyclic codes. Hence a code of length $n$ over $F_2 + uF_2$ gives rise to a linear binary code of length $2n$ and some classes of these codes lead to good binary linear cyclic codes in terms of minimum distance. The main difference is that the component vectors $u$ and $u+v$ are interleaved in the cyclic $R$ code construction whereas the vectors are concatenated in the $(u, u+v)$ construction. This makes it possible to correct certain bursts.

The main advantage of studying these codes is from the decoding point of view. The natural structure of these Gray mapped codes allows for them to be decoded in the $R$ domain. Hence the complexity of decoding a codeword of length $2n$ boils down to the complexity of decoding a minimum of two or at most the three binary codewords of length $n$. This gives rise to significant advantage in practical implementation.

Finally a perspective from studying codes over rings. Recently, the ring $Z_4$ was used to rediscover the structure of some notorious nonlinear binary codes. Our ring $F_2 + uF_2$ throws light on the structure of certain good $(u, u+v)$ constructed binary linear codes. The use of ring theory to improve the decoding of linear binary codes seems to be new. The ideas developed in this correspondence apply also to cyclic codes over $F_p + uF_p$, where $p$ is a prime number.

REFERENCES

On Minimum Lee Weights of Hensel Lifts of Some Binary BCH Codes

Hao Chen

Abstract—Motivated by the paper of Calderbank, McGuire, Kumar, and Helleseth, we prove the following result: for any given positive integer \( l \geq 3 \), the minimum Lee weights of Hensel lifts (to \( Z_4 \)) of binary primitive BCH codes of length \( 2^m - 1 \) and designed distance \( 2^l - 1 \) is just \( 2^l - 1 \) when \( a \) or \( b \) is sufficiently large. For Hensel lifts of binary primitive BCH codes of arbitrary designed distance \( \delta \geq 4 \), we also prove that their minimum Lee weight \( d_{\min} \leq 2^{l+1} \), so that when \( m \) is sufficiently large. Moreover, a result about minimum Lee weights of certain \( Z_4 \) codes defined by Galois rings, which is similar to the result in Calderbank et al., is proved.

Index Terms—Binary BCH codes, Hensel lifts, minimum Lee weight.

I. INTRODUCTION

It is shown in [2] that the famous nonlinear binary codes, Kerdock, Preparata, Goethals, Delsarte–Goethals, are the images under the Gray map of some linear codes over \( Z_7 \). There has been a great deal of research about linear codes over \( Z_4 \). Corresponding to the use of the Galois field \( GF(2^m) \) in the theory of binary cyclic codes, Galois rings \( GF(4^m) \), which are extensions of \( Z_4 \) of degree \( m \) containing a \( (2^m - 1) \)st root of unity (see [2]), played a very important role in the theory of linear codes over \( Z_4 \). In the binary case it is well known that a primitive Bose–Chaudhuri–Hocquenghem (BCH) code with length \( n \geq 2^m - 1 \) and designed distance \( \delta = 2^l - 1 \) has true minimum distance \( \delta \) (e.g., see [3]). The paper of A. R. Calderbank et al. [1] studied the minimum Lee weight of Hensel lifts of (extended) two-error correcting and three-error correcting BCH codes. Motivated by giving another interpretation of the results in that interesting paper we prove the following main result.

Main Result: Suppose that \( C \) is the Hensel lift of a binary primitive BCH code of length \( n = 2^m - 1 \) and designed distance \( \delta = 2^l - 1 \). Then one of the following cases is true: either

1) \( m \) can be divided by \( l \); or

2) \( m \geq A(l) \), where \( A(l) \) is a constant only depending on \( l \).

The minimum Lee weight of \( C \) is \( \delta \).

We also give results on the minimum Lee weights of some \( Z_4 \) codes defined over Galois rings which are very similar to the \( Z_4 \) codes \( C \) considered in [1, Corollary 2.2 and Theorem 3.8].

II. PRELIMINARIES

Since the Galois ring \( GF(4^m) \) is the main tool of our correspondence, we recall the main facts about it. For details we refer to [2]. To begin, let \( h_2(x) \in Z_2[x] \) be a primitive irreducible polynomial of degree \( m \), and the Galois field of \( 2^m \) elements \( F_{2^m} = Z_2[x]/(h_2(x)) \). From Hensel’s lemma we know there exists a unique monic polynomial \( h_1(x) \) of degree \( m \), such that \( h_1(x) = h_2(x) \) and \( h_1(x) \) divides \( x^{2^m} - 1 \) in \( Z_2[x] \). Let \( \xi \) be a root of \( h_1(x) \), so that \( \xi^{2^m} - 1 = 1 \). The Galois ring \( GF(4^m) \) is defined to be \( Z_4[\xi] = Z_4[x]/(h_1(x)) \). Every element \( e \in GF(4^m) \) has a unique 2-adic representation \( e = a + 2b \), where \( a, b \) are taken from the subset \( T = \{ 0, 1, \xi, \ldots, \xi^{2^m-2} \} \) of \( GF(4^m) \). We also have \( a = r(e) = \epsilon^2 \). There is a ring automorphism \( \epsilon : GF(4^m) \rightarrow GF(4^m) \) called the Frobenius map, defined to be \( \epsilon = a + 2b = \epsilon^2 = a^2 + 2b^2 \). We know that if \( h(x) \in Z_4[x] \) and \( e \in GF(4^m) \) is a root of \( h(x) \), \( \epsilon^2 \) is also a root of \( h(x) \). It is obvious that there is a natural map \( \mu : GF(4^m) \rightarrow F_{2^m} \) by \( (mod \ 2) \). We have \( \mu(\epsilon) = \mu(a) \). It is clear we have the following identity:

\[
(a_1 + \cdots + a_t) \epsilon^2 = \left( a_1^2 + \cdots + a_t^2 + \sum_{i < j} a_i a_j \right) \left( \epsilon^2 \right)^{2n-1} - \left( a_1 + \cdots + a_t \right) \epsilon^2
\]

\[
= a_1^2 + \cdots + a_t^2 + 2\sum_{i < j} (a_i a_j) \epsilon^{2n-1} \quad \text{in } GF(4^m).
\]

(2.1)

Let \( C_2 \) be a binary cyclic code of length \( n = 2^m - 1 \) generated by a polynomial \( g_2(x) \in Z_2[x] \) where we have a polynomial \( g(x) \) such that \( g(x)g(\xi) = x^{2^m - 1} - 1 \) in \( Z_2[x] \). Let \( g_2(x) \in Z_4[x] \) be the Hensel lift of \( g(x) \), then \( C_4 \), the ideal generated by \( g_2(x) \) in \( Z_4[x]/(x^{2^m} - 1) \), is called the Hensel lift of \( C_2 \). It is a cyclic linear code over \( Z_4 \).

Lemma 2.1: If \( \mu(\xi)^l \in F_{2^m} \) is a zero point of the binary cyclic code \( C_2 \), then \( \xi^l \) is a zero point of the \( Z_4 \)-cyclic code \( C_4 \).

Proof: We have the following commutative diagram:

\[
\begin{align*}
Z_4[x]/(x^{2^m} - 1) & \quad \text{GF}(4^m) \\
\downarrow & \quad \downarrow \\
Z_2[x]/(x^{2^m} - 1) & \quad F_{2^m}
\end{align*}
\]

(2.2)

where the upper horizontal map is defined to be the evaluation of \( x \) by \( \xi \), the lower horizontal map is defined to be the evaluation of \( x \) by \( \mu(\xi) \) (note that \( \xi \) and \( \mu(\xi) \) are \( (2^m - 1) \)st roots of unity over \( Z_4 \) and \( Z_2 \), respectively), the left vertical map is \( \text{mod } 2 \) and the right vertical map is \( \mu \). From this commutative diagram we can check the conclusion easily. Q.E.D.