CONSTRUCTION OF SELF-DUAL CODES OVER R_k AS LIFTS OF BINARY SELF-DUAL CODES

by

Refia AKSOY

A thesis submitted to

the Graduate School of Sciences and Engineering

of

Fatih University

in partial fulfillment of the requirements for the degree of

Master of Science

in

Mathematics

June 2014
Istanbul, Turkey

APPROVAL PAGE

This is to certify that I have read this thesis written by Refia AKSOY and that in my opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science in Mathematics.

Assoc. Prof. Dr. Suat KARADENİZ
Thesis Supervisor

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science in Mathematics.

Prof. Dr. Feyzi BAŞAR
Head of Department

Examining Committee Members

Assoc. Prof. Dr. Suat KARADENİZ _____

Assist. Prof. Dr. İsmail Gökhan KELEBEK _____

Assoc. Prof. Dr. Bülent KÖKLÜCE _____

It is approved that this thesis has been written in compliance with the formatting rules laid down by the Graduate School of Sciences and Engineering.

Assoc. Prof. Dr. Nurullah ARSLAN
Director

June 2014

CONSTRUCTION OF SELF-DUAL CODES OVER R_k AS LIFTS OF BINARY SELF-DUAL CODES

Refia AKSOY

M.S. Thesis – Mathematics
June 2014

Thesis Supervisor: Assoc. Prof. Dr. Suat KARADENİZ

ABSTRACT

The main purpose in this thesis is to find a new method to obtain self-dual codes of larger lengths from the known binary self-dual codes. In this work, the structure of the ring R_k and self-dual codes are studied. Lifts of self-dual codes over rings and the Gray images of codes are sought. The codes obtained via the Gray map are extended with the method given by Bouyuklieva and Bouyukliev and as a result of this study, new codes have been found.

Keywords: Linear codes over rings, self-dual codes, extremal binary codes, projections and lifts.

İKİLİ SELF-DUAL KODLARIN R_k HALKASINA TAŞINMASIYLA SELF-DUAL KODLARIN İNŞAASI

Refia AKSOY

Yüksek Lisans Tezi – Matematik
Haziran 2014

Tez Danışmanı: Doç. Dr. Suat KARADENİZ

ÖZ

Bu çalışmada temel amaç bilinen ikili self-dual kodlardan yola çıkarak daha büyük uzunlukta self-dual kodları elde etmeyi sağlayan bir method bulmaktır. Çalışmada self-dual kodların ve R_k halkasının yapısı incelenip, kodların halkalar üzerine taşınma metodları ve Gray dönüşümleri altındaki görüntülerine bakıldı. Gray dönüşümü ile elde edilen kodlar Bouyuklieva ve Bouyukliev'in genişletme yöntemi ile genişletilip yeni kodlar bulundu.

Anahtar Kelimeler: Halkalar üzerindeki lineer kodlar, self-dual kodlar, ekstrem ikili kodlar, izdüşümler ve taşınmalar.

To My Parents

ACKNOWLEDGEMENT

I would like to thank my thesis supervisor Assoc. Prof. Dr. Suat KARADENİZ for his guidance and encouragement. The experience I got by working alongside him will always be a valuable component of my future career.

I want to thank the rest of my committee Assist. Prof. İsmail Gökhan KELEBEK and Assoc. Prof. Dr. Bülent KÖKLÜCE. I would like to thank Prof. Dr. Feyzi BAŞAR since he encouraged me to study in Algebra. I also want to thank the Mathematics Department of Fatih University especially to Assoc. Prof. Dr. Bahattin YILDIZ, Assist. Prof. Abdullah Said ERDOĞAN, Ali Uğur SAZAKLIOĞLU, Abidin KAYA and Nesibe TÜFEKÇİ for their help with LaTeX.

I would like to send my special thanks to Assist. Prof. Dr. Zeynep ÖDEMİŞ ÖZGER due to her endless support, valuable comments and contributions.

Finally, I would like to thank my parents and brothers for their motivation and patience and to thank my friends especially to Mustafa DURAN, Kübra DURAN and Nazlı Deniz SAĞLAM because of their encouragement and support.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 HISTORY

Coding Theory arose in the late 1940's and was originated from the problems in the digital communication such as encoding and decoding. The main purpose of Coding Theory is detecting and correcting errors that might occur during information transmission.

In the early periods of Coding Theory, codes were generally studied over finite fields, particularly over the field of two elements, which are called binary codes. Linear codes have been the most widespread studied types of codes due to their convenience of constructing, encoding and decoding. Since the early nineties, rings have come out as an essential tool for Coding Theory.

In 1994, Hammons et al. published the paper (Hammons et al., 1994) in which they indicated that well-known binary codes could be obtained as images of linear codes over $\mathbb{Z}_4$ under a non-linear Gray map. This paper brought a new perspective to Coding Theory and since then, a great deal of research has done on codes over rings.

From the beginning of Coding Theory, a significant attention was on self-dual codes. These codes have been studied over different structures such as groups, rings and fields. Also these codes are found to be connected with many different fields of study such as group theory, combinatorial theory and cryptography. In 1999, Dougherty et al. investigated self-dual codes over $\mathbb{F}_2 + u\mathbb{F}_2$ in (Dougherty

et al., 1999). In 2010, Yildiz and Karadeniz studied linear codes over the ring $\mathbb{F}_2 + u\mathbb{F}_2 + v\mathbb{F}_2 + uv\mathbb{F}_2$ in (Yildiz and Karadeniz, 2010) which is not a chain ring nor a principal ideal ring. Then, Dougherty et al. considered the codes over R_k and defined Gray maps in (Dougherty et al., 2011).

1.2 BASIC DEFINITIONS

We start with some basic concepts of Coding Theory. We refer to (Ling and Xing, 2004) and (Huffman and Pless, 2003) for more.

Let $\mathbb{F}_q^n$ be a vector space of dimension n over the finite field $\mathbb{F}_q$, where q is a prime power. A subspace C of vector space $\mathbb{F}_q^n$ is called a *linear code* of length n. C is also said to be an (n, M) code over $\mathbb{F}_q$ where M is the size of C. The vectors in C are called as *codewords* and they are denoted by $\overline{c} \in C$. The code C is called *binary* if $q = 2$ and *ternary* if $q = 3$.

The *Hamming distance*, the most common distance used for codes over finite fields, between two vectors $x, y \in \mathbb{F}_q^n$ is the number of coordinates in which $x = (x_1, x_2, \ldots, x_n)$ and $y = (y_1, y_2, \ldots, y_n)$ differ, i.e.,

$$d_H(x, y) = |\{i : x_i \neq y_i\}|$$

The *Hamming weight* of x, denoted by $w_H(x)$, is defined to be the number of nonzero coordinates in x where $x = (x_1, x_2, \ldots, x_n) \in \mathbb{F}_q^n$, i.e.,

$$w_H(x) = |i : x_i \neq 0|$$

Notice that

$$d_H(x, y) = w_H(x - y), \ \forall x, y \in \mathbb{F}_q^n.$$

The *minimum distance* of a code C is the smallest distance between two distinct codewords, i.e., the minimum distance of a C is

$$d_{min}(C) = min\{d(x, y)|x, y \in C, \ x \neq y\}.$$

The *minimum weight* of a code C is defined as follows:

$$w(C) = min\{w(\overline{c})|\overline{c} \in C,\ \overline{c} \neq 0\}$$

Hereby, examples about Hamming distance and Hamming weight will be helpful to understand.

Example 1.1. *Let* $A = \{0, 1\}$ *and let* $x = 101001$, $y = 011011$ *and* $z = 111101$. *Then*

$$d(x, y) = 3, d(y, z) = 3, d(x, z) = 2$$

Example 1.2. *Consider the binary linear code* $C = \{0000, 1000, 0010, 1010\}$. *It is seen that*

$$w_H(1000) = 1,\ w_H(0010) = 1,\ w_H(1010) = 2$$

Therefore $d(C) = 1$.

The minimum distance of a code tells us what the error-correcting capacity of the code is, in other words, if the minimum distance is d, then the code can detect $d - 1$ errors and correct up to $e = \left[\frac{d-1}{2}\right]$ errors.

A code C of length n over $\mathbb{F}_q^n$, of dimension k with the minimum distance d is denoted by $[n, k, d]$-code. These are the most important parameters of a code.

1.3 OVERVIEW OF THE THESIS

In Chapter 2, basic information about self dual codes are given. At the end of the chapter, some extension methods which allow us to obtain codes of length $n+2$ from the codes of length n are mentioned.

In Chapter 3, the ring R_k is introduced and the properties of self-dual codes over the ring are investigated. The distance preserving natural Gray map from the ring to the binary field is defined. The elementary examples of the ring R_k such as R_1 and R_2 are examined.

In Chapter 4, we deal with the way to construct self-dual codes over R_k. Projection and lifting of codes are defined. A new method to construct self-dual codes over R_k as lifts of binary self-dual codes is introduced.

In Chapter 5, we list up the new codes that are obtained by using the lifting and extension method and give the complete list of the extremal binary self-dual codes of length 58 which are obtained by using the method we developed.

CHAPTER 2

SELF-DUAL CODES

In this chapter, we give some fundamental about linear codes especially self-dual codes and their extension methods.

A *linear* $[n, k]$-code over $\mathbb{F}_q$ is a k-dimensional subspace of $\mathbb{F}_q^n$. A benefit of working with linear codes is that the minimum distance is equal to the minimum weight of a linear code since $d(x, y) = w(x - y)$, $d(C) = w(C)$. As a result of this, the complexity of finding the minimum distance is reduced.

Definition 2.1. *(Huffman and Pless, 2003) A $k \times n$ matrix G whose rows form a basis for C is called a generator matrix for C.*

A generator matrix in standard form is shown as $G = [I_k|A]$ where I_k denotes the $k \times k$ identity matrix.

The vector space $\mathbb{F}_q^n$ has an inner product which is defined as

$$(x_1, x_2, ..., x_n).(y_1, y_2, ..., y_n) = x_1y_1 + x_2y_2 + ... + x_ny_n.$$

Let C be a linear $[n, k]$-code. Then its dual set $C^\perp$ is defined as

$$C^\perp = \{\overline{u} \in \mathbb{F}_q^n \mid \overline{u}.\overline{v} = 0, \forall \overline{v} \in C\}.$$

If $C \subseteq C^\perp$ then C is called *self-orthogonal*, and if $C = C^\perp$ then C is said to be *self-dual*.

Definition 2.2. *(Ling and Xing, 2004) The matrix whose rows form a basis for $C^\perp$ is called a parity-check matrix of C.*

The standard form of a parity-check matrix is shown as $H = [-A^T | I_{n-k}]$. It is obvious that $G.H^T = \overline{0}$.

If C is an $[n, k]$-code over $\mathbb{F}_q$ then its dual $C^\perp$ is a linear $[n, n-k]$-code.

Definition 2.3. *(Rains and Sloane, 1998) A self-dual code is called doubly-even (Type II) if all codewords have weight* $\equiv 0 \pmod 4$ *and singly-even (Type I) if some codewords have weight* $\equiv 2 \pmod 4$.

Codes over $\mathbb{F}_2$ are called *binary*. Note that binary self-dual codes exist only for even lengths. Moreover, doubly-even self-dual codes exist only for lengths $n \equiv 0 \pmod 8$.

The minimum weight d of a self-dual code of length n is bounded (Rains, 1998) as follows:

$$d \leq \begin{cases} 4\left[\frac{n}{24}\right] + 4, n \not\equiv 22 \pmod{24}, \\ 4\left[\frac{n}{24}\right] + 6, n \equiv 22 \pmod{24}. \end{cases}$$

A self-dual code meeting the bound is called *extremal*. A self-dual code which has the largest minimum weight among all self-dual codes of a given length is named *optimal* (Bouyuklieva et al., 2005).

Two codes are called *permutationally equivalent* if one can be obtained from the other by a suitable permutation of coordinates of all codewords (Bouyuklieva et al., 2005).

Let C be a linear code of length n and let A_i be the number of codewords of weight i. Then

$$A(x) := \sum_{i=0}^{n} A_i x^i$$

is called the *weight enumerator* of C. If C is an $[n, k]$-code with weight enumerator $A(x)$, then

$$B(x) := \frac{1}{|C|}(1+x)^n A\left(\frac{1-x}{1+x}\right)$$

is the weight enumerator of $C^\perp$. This identity is called *MacWilliams Identity*.

In general, the weight enumerator of C is

$$W_C(x,y) = \sum_{i=0}^{n} A_i x^{n-i} y^i$$

where x and y are indeterminates.

Recall that w_H is the Hamming weight and

$$W_C(x,y) = \sum_{\overline{u} \in C} x^{n-w_H(\overline{u})} y^{w_H(\overline{u})}$$

By taking x as 1, one can obtain the weight enumerator in one indeterminate y,

$$A(y) = W_C(1,y) = W_C(y) = \sum_{i=0}^{n} A_i y^i.$$

The weight enumerator of the dual code $C^{\perp}$ is

$$\sum_{i=0}^{n} A'_i x^{n-i} y^i = \sum_{\overline{u} \in C^{\perp}} x^{n-w_H(\overline{u})} y^{w_H(\overline{u})}$$

Theorem 2.1. *(MacWilliams and Sloane, 1977) If C is an $[n,k]$ binary linear code with dual $C^{\perp}$, then*

$$W_{C^{\perp}}(x,y) = \frac{1}{|C|} W_C(x+y, x-y) \tag{2.1}$$

where $|C|$ is the size of the code C. Equivalently,

$$\sum_{k=0}^{n} A'_i x^{n-k} y^k = \frac{1}{|C|} \sum_{i=0}^{n} A_i (x+y)^{n-i} (x-y)^i, \tag{2.2}$$

or

$$\sum_{\overline{u} \in C^{\perp}} x^{n-w_H(\overline{u})} y^{w_H(\overline{u})} = \frac{1}{|C|} \sum_{\overline{u} \in C} (x+y)^{n-w_H(\overline{u})} (x-y)^{w_H(\overline{u})}. \tag{2.3}$$

These equations (2.1)-(2.3) are called the MacWilliams identities.

Before we give extension methods, we complete this part with some examples of binary self-dual codes.

Example 2.1.

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{bmatrix}$$

This matrix in standard form generates $[8, 4, 4]$ *extended binary Hamming code.*

The following extended Hamming code generated by G *has sixteen codewords:*

$$C = \left\{ \begin{array}{cccc} 00000000 & 10001110 & 01000111 & 00101011 \\ 00011101 & 11001001 & 10100101 & 10010011 \\ 01101100 & 01011010 & 00110110 & 11110000 \\ 11010100 & 01110001 & 10111000 & 11111111 \end{array} \right\}$$

This code is obtained from $[7, 4, 3]$ *Hamming code by adding parity check bit to each codeword. Parity check bit makes the code even weighted.*

Example 2.2. *The* $[24, 12, 8]$ *extended binary Golay code can be generated by* 12×24 *matrix* $G = [I|A]$ *where* I *is the* 12×12 *identity matrix and* A *is the following matrix*

$$A = \left[\begin{array}{cccccccccccc} 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \end{array} \right]$$

Example 2.3. *The extended Quadratic Residue code is* $[48, 24, 12]$ *code. Moreover, it is doubly-even, extremal self-dual code.*

An important way to obtain a new code from known codes is using extension methods. Now, we give some of these methods that allow us to obtain a binary self-dual code of length $n + 2$ from a binary self-dual code of length n.

2.1 EXTENSION METHODS

2.1.1 Harada's Extension

Proposition 2.1. *(Harada, 1997) Let Ω be a subset of the set $\{1, 2, \ldots, n\}$ such that $|\Omega|$ is odd if $2n \equiv 0 \pmod 4$ and $|\Omega|$ is even if $2n \equiv 2 \pmod 4$. Let $G_0 = (I_n, A)$ be a generator matrix of a self-dual code C_0 of length $2n$, where I_n is the identity matrix of order n. Then the following matrix:*

$$G = \begin{bmatrix} 1 & 0 & x_1 & \ldots & x_n & 1 & \ldots & 1 \\ y_1 & y_1 & & & & & & \\ \vdots & \vdots & & I_n & & & A & \\ y_n & y_n & & & & & & \end{bmatrix},$$

where $x_i = 1$ if $i \in \Omega$ and $x_i = 0$ otherwise, and $y_i \equiv x_i + 1 \pmod 2$ $(1 \leq i \leq n)$, generates a self-dual code C of length $2n + 2$.

By using this construction method, Harada obtained a singly-even $[70, 35, 12]$ code which is not an extremal code. With the help of the same method, Boukliev and Buyuklieva obtained some new extremal self-dual codes with lengths 44, 50, 54 and 58 in (Boukliev and Buyuklieva, 1998).

This method enables to obtain a number of different generator matrices of self-dual codes of length $2n + 2$ from a self-dual code of length $2n$. Accordingly various different codes can be obtained. The method is efficient because we only need to determine the vector x and y_i's are automatically congruent to $x_i + 1$. The vector x can be chosen in 2^{n-1} different ways.

2.1.2 Kim's Extension

Theorem 2.2. *(Kim, 2001) Let S be a subset of the set $\{1, 2, \ldots, 2n\}$ of coordinate indices such that $|S|$ is odd. Let $G_0 = (L|R) = (l_i|r_i)$ be a generator matrix (may not be in standard form) of a self-dual code C_0 of length $2n$, where l_i and r_i are rows*

of L and R, respectively, for $1 \leq i \leq n$. Let

$$x = (x_1, \ldots, x_n, x_{n+1}, \ldots, x_{2n})$$

be the characteristic vector of S, i.e., $x_j = 1$ if $j \in S$ and $x_j = 0$ if $j \notin S$ for $1 \leq j \leq 2n$. Suppose that

$$y_i := (x_1, \ldots, x_n, x_{n+1}, \ldots, x_{2n}) \cdot (l_i | r_i)$$

for $1 \leq i \leq n$. Here $\cdot$ denotes the (scalar) inner product. Then the following matrix:

$$G = \begin{bmatrix} 1 & 0 & x_1 & \ldots & x_n & x_{n+1} & \ldots & x_{2n} \\ y_1 & y_1 & & & & & & \\ \vdots & \vdots & & L & & & R & \\ y_n & y_n & & & & & & \end{bmatrix}$$

generates a self-dual code C of length $2n + 2$.

Kim used this method to obtain new extremal self-dual binary codes of lengths 36,38 and 58 in (Kim, 2001). In this work, he showed that there were at least 14 inequivalent extremal self-dual $[36, 18, 8]$ codes and also there were at least 368 inequivalent extremal self-dual $[38, 19, 8]$ codes. He constructed 11 extremal self-dual $[58, 29, 10]$ codes whose weight enumerators were not previously known to exist.

The disadvantage of this extension is that it requires too many calculations in order to determine y_i's. Here, x can be selected in 2^{2n-1} different ways.

2.1.3 Melchor et al.'s Extension

(Melchor and Gaborit, 2008) Assume C_{n+2} is an $\left[n + 2, \frac{n}{2} + 1, d + 2\right]$ self-dual code with $d \geq 2$ then up to equivalence a generator matrix of C_{n+2} can be written as

$$G = \begin{bmatrix} 1 & 1 & \mathbf{y} \\ 0 & 0 & D \\ 0 & 1 & \mathbf{z} \end{bmatrix}$$

where y is a codeword of length n and weight d; D is a certain subcode $\left[n, \frac{n}{2} - 1\right]$ of C_{n+2}, truncated on its first two coordinates; and z is a certain truncated codeword of C_{n+2}.

We remark that the code with generator matrix $\begin{pmatrix} \mathbf{y} \\ D \end{pmatrix}$ is a self-dual $\left[n, \frac{n}{2}, d\right]$ code. This remark implies that a self-dual $\left[n, \frac{n}{2}, d\right]$ code can be associated to any self-dual $\left[n+2, \frac{n}{2}+1, d+2\right]$ code.

Now, conversely, suppose that one starts from a $\left[n, \frac{n}{2}, d\right]$ code C_n obtained as before by truncation of two columns and getting rid of an appropriate row of a $\left[n+2, \frac{n}{2}+1, d+2\right]$ code C_{n+2}. Then one can choose a generator matrix of C_n a matrix of the form: $\begin{pmatrix} \mathbf{y}' \\ E \end{pmatrix}$ for $\mathbf{y}'$ a word of weight d.

If one considers the set MC of all the $\left[n+2, \frac{n}{2}\right]$ codes C with generator matrices of the form

$$\begin{bmatrix} 1 & 1 & \mathbf{y}' \\ a_1 & a_1 & \\ \vdots & \vdots & E \\ a_{\frac{n}{2}-1} & a_{\frac{n}{2}-1} & \end{bmatrix}$$

where $a_i \in 0, 1$, one will necessarily construct a subcode of C_{n+2} of dimension n since any codeword of C_n extended either by 00 or by 11 is in C_{n+2}. To find C_{n+2} back it is then sufficient to complete all the codes C of MC by one of the three non-null elements of $C^{\perp}/C$.

Hence, we have proven that if one knows a subcode $\left[n, \frac{n}{2}, d\right]$ C_n of an $\left[n+2, \frac{n}{2}+1, d+2\right]$ code C_{n+2}, it is possible to rebuild the code C_{n+2} by this method, and this up to equivalence.

Therefore, if rather than starting from a unique $\left[n, \frac{n}{2}, d\right]$ code, one starts from the set of all inequivalent $\left[n, \frac{n}{2}, d\right]$ self-dual codes, it is possible to rebuild all the $\left[n+2, \frac{n}{2}+1, d+2\right]$ self-dual codes.

Now 2^{n-1} possibilities must be considered for the a_i. Meanwhile, by noticing that necessarily all the words of weight d of the $\left[n, \frac{n}{2}, d\right]$ code have to be extended in codewords of weight $d+2$, it is then possible to greatly fasten the algorithm by considering not only one vector of weight d but the code C_d generated by the vectors of weight d. Obviously, the dimension of k of C_d satisfies $k \leq n/2$. It is then

sufficient to consider 2^{n-k} possibilities for the a_i instead of 2^{n-1} and this, only once for each code.

We now sum up the algorithm to compute recursively all the $\left[n+2, \frac{n}{2}+1, d+2\right]$ self-dual codes from all the $\left[n, \frac{n}{2}, d\right]$ self-dual codes:

2.1.3.1 Recursive Algorithm

Input: S_n the set of $\left[n, \frac{n}{2}, d\right]$ self-dual codes up to permutation

Output: The set of $\left[n+2, \frac{n}{2}+1, d+2\right]$ self-dual codes

For each code C_n of S_n do:

(1) List all the words of weight d and construct the subcode C_d of dimension k generated by these words. Construct a generator matrix G_d of C_d composed only with words of weight d.

(2) Let E be a code of dimension $n-k$ with generator matrix G_E such that $C_n = C_d + E$, constructs the extended codes C with generator matrices

$$\begin{bmatrix} 1 & 1 & \\ \vdots & \vdots & G_d \\ 1 & 1 & \\ a_1 & a_1 & \\ \vdots & \vdots & G_E \\ a_{\frac{n}{2}-k} & a_{\frac{n}{2}-k} & \end{bmatrix}$$

such that $a_i \in \{0,1\}$, $(1 \leq i \leq \frac{n}{2}-k)$.

(3) Complete all the previous codes C by non-null elements of $C^{\perp}/C$ in order to obtain a self-dual code D and check for codes with minimum distance $d+2$. For codes with weight $d+2$, check for the equivalence with already obtained self-dual $\left[n+2, \frac{n}{2}+1, d+2\right]$ codes.

The purpose of this extension is to obtain new codes of minimum weight d+2 by

using the codes of minimum weight d. The advantage of this extension is narrowing the search space, but it is not effective as Harada's extension if the length is big.

Extremal Type I $[32, 16, 8]$ codes were classified by Conway and Sloane in (Conway and Sloane, 1990). A classification of Type I self-dual codes of length 32 and 34 was given in (Bilous and van Rees, 2002) and (Bilous, 2006). Harada and Munemasa gave a complete classification of binary self-dual codes of length 36 in (Harada and Munemasa, 2012). Melchor and Gaborit classified extremal $[36, 18, 8]$ binary self-dual codes from $[34, 17, 6]$ codes by using this algorithm. Using the $[36, 18, 6]$ codes, all extremal $[38, 19, 8]$ codes were classified in (Aquilar-Melchor et al., 2012) by Aguilar-Melchor et al. with the help of the same algorithm.

2.1.4 Bouyuklieva and Bouyukliev's Extension

Proposition 2.2. *(Bouyuklieva and Bouyukliev, 2012) If C is a binary self-dual $[n = 2k > 2, k, d]$ code then C is equivalent to a code with a generator matrix in the form*

$$G = \begin{pmatrix} x_1 & \dots & x_{k-1} & 0 & 0 & \dots & 0 & 1 & 0 \\ & & & & & & & x_1 & x_1 \\ & I_{k-1} & & & & A & & \vdots & \vdots \\ & & & & & & & x_{k-1} & x_{k-1} \end{pmatrix}$$

and the matrix $(I_{k-1}|A)$ generates a self-dual $[n-2, k-1]$ code.

Here, the weight of $(x_1, x_2, \dots, x_{k-1})$ must be odd.

This extension is similar to Harada's extension, but this is the smartest one. We only need a vector whose weight is odd in order to extend the codes of length n to the codes of length $n + 2$.

The doubly even self-dual codes of length 40 were classified in (Betsumiya et al., 2012). Moreover, using these codes, they classified the optimal self-dual $[38, 19, 8]$ codes. Using the construction method that we gave above, a complete classification of all self-dual codes of length 38 was given by Bouyuklieva and Bouyukliev in (Bouyuklieva and Bouyukliev, 2012).

CHAPTER 3

THE RING R_k AND SELF-DUAL CODES OVER R_k

In this chapter, we introduce the ring R_k and define self-dual codes over this ring. We begin with the definition of a linear code over a finite ring R.

Definition 3.1. *(Horimoto and Shiromoto, 2001) Let R be a finite commutative ring. A code over R of length n is a subset of R^n. A linear code over R of length n is an R-submodule of R^n.*

The ring R_k is defined in (Dougherty et al., 2011) as follows:

$$R_k = \mathbb{F}_2[u_1, u_2, ..., u_k]/\langle u_i^2 = 0,\ u_i u_j = u_j u_i \rangle.$$

The ring can also be defined recursively,

$$R_k = R_{k-1}[u_k]/\langle u_k^2 = 0, u_k u_j = u_j u_k \rangle = R_{k-1} + u_k R_{k-1}, \quad j = 1, 2, ..., k-1$$

For any subset $A \subseteq \{1, 2, \ldots, k\}$, u_A can be fixed as

$$u_A := \prod_{i \in A} u_i$$

with $u_\emptyset = 1$. Any element of R_k can be represented as

$$\sum_{A \subseteq \{1,\ldots,k\}} c_A u_A, \qquad c_A \in \mathbb{F}_2.$$

Lemma 3.1. *(Dougherty et al., 2011) The ring R_k is a commutative ring with $|R_k| = 2^{(2^k)}$.*

The ring R_k is a local ring with maximal ideal $\langle u_1, u_2, ..., u_k \rangle$ and it is a Frobenius ring. The ring is neither a principal ideal ring nor a chain ring when $k \geq 2$. In (Dougherty et al., 2011), it is shown that an element of R_k is a unit if and only if it has the term 1 and that each unit is also its own inverse. Therefore, we have the following:

$$\forall a \in R_k,\ a \cdot (u_1 u_2 \dots u_k) = \begin{cases} u_1 u_2 \dots u_k, \text{ if } a \text{ is a unit} \\ 0, \text{ otherwise.} \end{cases}$$

Also,

$$\forall a \in R_k,\ a^2 = \begin{cases} 1, \text{ if } a \text{ is a unit} \\ 0, \text{ otherwise.} \end{cases}$$

In (Dougherty et al., 2013), authors show that the inner product over R_k is defined as $[\mathbf{v}, \mathbf{w}]_k = \sum \mathbf{v}_i \mathbf{w}_i$. The dual set of C is $C^\perp = \{\mathbf{v} \in R_k^n \mid [\mathbf{v}, \mathbf{w}]_k = 0 \text{ for all } \mathbf{w} \in C\}$. Due to the results given in (Wood, 1999), we can say that for finite k, a linear code C over R_k of length n satisfies $|C||C^\perp| = |R_k|^n$. As stated earlier in Chapter 2, the code is self-orthogonal if $C \subseteq C^\perp$ and self-dual if $C = C^\perp$.

A natural Gray map from R_k to $\mathbb{F}_2^{2^k}$ maps self-dual codes over R_k of length n to binary self-dual codes of length $2^k n$. For $\overline{c} \in R_k^n$, $\overline{c} = \overline{c}_1 + u_k \overline{c}_2$ with $\overline{c}_1, \overline{c}_2 \in R_{k-1}$, then the Gray map can be defined as

$$\phi_k(\overline{c}) = (\phi_{k-1}(\overline{c}_2), \phi_{k-1}(\overline{c}_1) + \phi_{k-1}(\overline{c}_2)).$$

Here ϕ_0 is the identity map on $\mathbb{F}_2$.

The Lee weight w_L of a codeword is the Hamming weight of the image of the codeword under ϕ_k. Then the Gray map is a linear weight preserving map from R_k^n to $\mathbb{F}_2^{2^k n}$.

Lemma 3.2. *(Karadeniz and Yıldız, 2013) If C is a self-dual code over R_k of length n, then $\phi_k(C)$ is a binary self-dual code of length $2^k n$. The Lee weight distribution of C and the Hamming weight distribution of $\phi_k(C)$ are the same. In particular, if C is Type I, then so is $\phi_k(C)$ and the same is true for Type II codes as well.*

By lemma, we can deduce that it is possible to obtain binary self-dual codes of length $2^k n$ from self-dual codes over R_k of length n.

Corollary 3.1. *(Dougherty et al., 2013) Let $d_L(n, I)$ and $d_L(n, II)$ denote the minimum distance of a Type I and Type II code over R_k of length n, respectively. Then for $k \geq 2$ we have*

$$d_L(n, I), d_L(n, II) \leq 4 \left[\frac{2^{k-2} n}{6} \right] + 4.$$

See (Karadeniz, 2011) for a general setting of the ring R_2.

Let us see the first two examples of R_k, which are R_1 and R_2.

3.1 THE RING R_1

The rings $R_0 = \mathbb{F}_2$ and $R_1 = \mathbb{F}_2 + u\mathbb{F}_2$ have been studied intensively in the literature of Coding Theory. R_1 which is a ring of characteristic 2 was identified to construct lattices in (Bachoc, 1997). This ring is a commutative chain ring with a nilpotent element u where $u^2 = 0$. The elements of the ring are $R_1 = \mathbb{F}_2 + u\mathbb{F}_2 = \{0, 1, u, 1 + u\}$. For $a, b, c, d \in \mathbb{F}_2$ addition and multiplication are defined as

$$(a + ub) + (c + ud) = (a + c) + u(b + d)$$

and

$$(a + ub)(c + ud) = ac + u(ad + bc).$$

The units of the ring are 1 and $1 + u$. (0),(u) and (1) are the ideals of the ring.

A linear Gray map from R_1^n to $\mathbb{F}_2^{2n}$ was defined in terms of vectors as

$$\phi_1(\overline{x} + u\overline{y}) = (\overline{y}, \overline{x} + \overline{y})$$

where $\overline{x}, \overline{y} \in \mathbb{F}_2^n$ which is a linear distance preserving map (Dougherty et al., 1999).

More specifically,

$$\phi_1(0) = (00), \phi_1(1) = (01), \phi_1(u) = (11), \phi_1(1 + u) = (10)$$

The Lee weight is determined as $w_L(0) = 0$, $w_L(1) = 1$, $w_L(u) = 2$, $w_L(1+u) = 1$ in $\mathbb{F}_2$. We can easily see that the Lee weight of a codeword is equal to the Hamming weight of the image of the codeword under the Gray map.

Example 3.1. *Let* $\overline{c} \in R_1^n = \overline{x} + u.\overline{y}$, *where* $\overline{x}, \overline{y} \in \mathbb{F}_2^n$.

$$\overline{c} = (1+u, u, 1, 0, u) = (1, 0, 1, 0, 0) + u(1, 1, 0, 0, 1).$$

Then the Gray image of $\overline{c}$ *is*

$$\begin{aligned} \phi(\overline{c}) &= (\overline{y}, \overline{x}+\overline{y}) = (1,1,0,0,1,0,1,1,0,1), \\ w_H(\phi(\overline{c})) &= 6 = w_L(\overline{c}) \end{aligned}$$

In (Dougherty et al., 1999), it was shown that any code over R_1 is permutation-equivalent to a code C with the generator matrix

$$G = \begin{pmatrix} I_{k_1} & A & B_1 + uB_2 \\ 0 & uI_{k_2} & uD \end{pmatrix}$$

where A, B_1, B_2 and D are matrices over $\mathbb{F}_2$. In this work, they correlate two binary codes: the residue code C_1 and the torsion code C_2 as follows:

$$C_1 = \{x \in \mathbb{F}_2^n | \ \exists y \in \mathbb{F}_2^n : \ x + uy \in C\}$$

and

$$C_2 = \{x \in \mathbb{F}_2^n | \ ux \in C\}$$

A generator matrix of C_1 is

$$G_1 = \begin{pmatrix} I_{k_1} & A & B_1 \end{pmatrix}$$

and a generator matrix of C_2 is

$$G_2 = \begin{pmatrix} I_{k_1} & A & B_1 \\ 0 & I_{k_2} & D \end{pmatrix}$$

The number of elements in C is calculated as

$$|C| = |C_1| \, . \, |C_2| = 2^{k_1} 2^{k_1+k_2} = 2^{2k_1+k_2} = 4^{k_1} 2^{k_2}.$$

3.2 THE RING R_2

The ring $R_2 = \mathbb{F}_2 + u\mathbb{F}_2 + v\mathbb{F}_2 + uv\mathbb{F}_2$ was firstly introduced by Yildiz and Karadeniz in (Yildiz and Karadeniz, 2010). This ring is an extension of R_1. It is defined as a characteristic 2 ring with the restrictions $u^2 = v^2 = 0$ and $uv = vu$. Thus R_2 is defined as follows:

$$R_2 = \mathbb{F}_2[u, v]/(u^2 = v^2 = 0, uv = vu).$$

It can also be described in terms of R_1 as

$$R_2 = R_1[v]/\langle v^2 = 0, uv - vu\rangle = R_1 + vR_1$$

The elements of the ring are $\{0, 1, u, v, uv, 1+u, 1+v, 1+uv, u+v, u+uv, v+uv, 1+u+v, u+v+uv, 1+u+uv, 1+v+uv, 1+u+v+uv\}$.

The addition and multiplication over the ring are defined as follows: For all $a, b, c, d, e, f, g, h \in \mathbb{F}$,

$$(a+ub+vc+uvd)+(e+uf+vg+uvh) = (a+e)+u(b+f)+v(c+g)+uv(d+h)$$

and

$$(a+ub+vc+uvd)(e+uf+vg+uvh) = ae+u(af+eb)+v(ag+ce)+uv(bg+cf+ah+de).$$

The units of the ring are $\{1, 1+u, 1+v, 1+u+v, 1+u+uv, 1+v+uv, 1+uv, 1+u+v+uv\}$. The ideals of R_2 can be written as

$$I_0 = \{0\} \subseteq I_{uv} = uv(R_2) = \{0, uv\} \subseteq I_u, I_v, I_{u+v} \subseteq I_{u,v} \subseteq I_1 = R_2$$

where $I_u = u(R_2) = \{0, u, uv, u+uv\}$, $I_v = v(R_2) = \{0, v, uv, v+uv\}$,

$I_{u+v} = (u+v)(R_2) = \{0, u+v, uv, u+v+uv\}$,

$I_{u,v} = \{0, u, v, u+v, uv, u+uv, v+uv, u+v+uv\}$.

R_2 is both Frobenius ring and local ring with the maximum ideal $I_{u,v}$ but it is not a chain ring.

In (Yildiz and Karadeniz, 2010), the map $\phi_2 : R_2^n \longrightarrow \mathbb{F}_2^{4n}$ given by

$$\phi_2(\overline{a} + u\overline{b} + v\overline{c} + uv\overline{d}) = (\overline{d}, \overline{c} + \overline{d}, \overline{b} + \overline{d}, \overline{a} + \overline{b} + \overline{c} + \overline{d})$$

as the Gray map from R_2^n to $\mathbb{F}_2^{4n}$.

Notice that ϕ is a linear map that takes a linear codes over $\mathbb{F}_2+u\mathbb{F}_2+v\mathbb{F}_2+uv\mathbb{F}_2$ of length n to a binary linear code of length $4n$. By using this map, the Lee weight can be defined as follows:

Definition 3.2. *Let* $\overline{a} + u\overline{b} + v\overline{c} + uv\overline{d}$ *be an element of* $\mathbb{F}_2 + u\mathbb{F}_2 + v\mathbb{F}_2 + uv\mathbb{F}_2$. $w_L(\overline{a} + u\overline{b} + v\overline{c} + uv\overline{d}) = w_H(\overline{d}, \overline{c} + \overline{d}, \overline{b} + \overline{d}, \overline{a} + \overline{b} + \overline{c} + \overline{d})$

As a consequent of this we have one element whose weight is 0, four elements whose weights are 1 which are 1, $1 + u$, $1 + v$, $1 + u + v + uv$, six element of weight 2 which are u, v, $u + v$, $u + uv$, $v + uv$, $u + v + uv$, four element whose weights are 3 which are $1 + uv$, $1 + u + uv$, $1 + v + uv$, $1 + u + v$ and one element of weight 4 which is uv.

Lemma 3.3. *(Yıldız and Karadeniz, 2010) If* C *is a linear code over* R_2 *of length* n*, size* 2^k *and minimum Lee distance* d*, then* $\phi_2(C)$ *is a binary* $[4n, k, d]$*-linear code.*

Theorem 3.1. *(Yıldız and Karadeniz, 2010) If* C *is a self-dual code over* R_2*, then* C *contains the all-*uv *vector.*

3.3 SELF-DUAL CODES OVER R_k

In this section, we give important properties of self-dual codes over R_k, mainly based on (Dougherty et al., 2013).

Lemma 3.4. *(Dougherty et al., 2011)The code* $\langle u_i \rangle$ *of length 1 is a self-dual code in* R_k *for all* $k \geq i$.

Before giving the theorem let us introduce direct product of codes over R_k.

If A and B are self-dual codes over R_k then it can be defined that $A \times B = \{(v, w)|v \in A, w \in B\}$. It is seen that this code is self-orthogonal and of the proper cardinality. Hence, the code is self-dual.

Theorem 3.2. *(Dougherty et al., 2013) Self-dual codes over R_k exist for all lengths and for all $k \geq 1$.*

Theorem 3.3. *(Dougherty et al., 2013) Type II codes over R_k of all lengths exist for any $k \geq 3$.*

Theorem 3.4. *(Dougherty et al., 2013) Type II codes exist over R_2 for all even lengths.*

For the section below, we refer to (Dougherty et al., 2013).

3.4 SELF-DUAL CODES OF LENGTH 1 AND 2

3.4.1 Length 1 Self-Dual Codes over R_k

Note that if a length 1 code C, generated by $a + u_k b$, with $a, b \in R_{k-1}$ is self-orthogonal, then we must have that a is a non-unit in R_{k-1}, because if a were a unit, then we would have $(a + u_k b)^2 = a^2 = 1 \neq 0$.

In order to prove the condition that while a is a non-unit and b is a unit, then $\langle a + u_k b \rangle$ is a self-dual code, we introduce the following map:

$$\Psi_k : R_k \longrightarrow R_{k-1}^2$$

defined by

$$\Psi_k(a + u_k b) = (b, a + b).$$

It is seen that Ψ_k is a linear bijection from R_k^n to R_{k-1}^{2n} and moreover it is a distance preserving map.

The following lemma helps us to clarify why a must be non-unit.

Lemma 3.5. *(Dougherty et al., 2013) If C is a length 1 code over R_k generated by $a + u_k b$ with $a, b \in R_{k-1}$, then $\Psi_k(C)$ is a length 2 code over R_{k-1} generated by $(b, a + b)$ and (a, a).*

Theorem 3.5. *(Dougherty et al., 2013) Let C be the length 1 code over R_k generated by $a + u_k b$ where a is a non-unit and b is a unit in R_{k-1}. Then C is self-dual.*

For the necessary conditions, by changing the indices of the u_i, we obtain next corollary by generalizing Theorem (3.5):

Corollary 3.2. *(Dougherty et al., 2013) Let C be a length 1 code over R_k generated by $a + u_i b$ for some i with $1 \leq i \leq k$, where a is a non-unit and b is a unit in R_k, such that neither au_i nor bu_i equal to 0. Then C is a self-dual code.*

By the corollary above, we can deduce that there is a large class of length 1 self-dual codes.

3.4.2 Length 2 Self-Dual Codes over R_k

Note that, for any $a \in R_k$, with $k \geq 1$, $a^2 = 1$ if a is a unit and $a^2 = 0$ otherwise. This tells us that every codeword in a length 2 self-dual code over R_k must be of the form (a_1, a_2) where a_i's are units or of the form (b_1, b_2) where b_i's are non-units.

Proposition 3.1. *(Dougherty et al., 2013) Let C be a linear code over R_k of length 2 generated by*
$(1, 1 + u_1 u_2 \dots u_k)$ with $k \geq 2$. Then C is a Type II code with minimum distance 4.

Proposition 3.2. *(Dougherty et al., 2013) Let C be a linear code over R_k of length 2 generated by (a, b) where a and b are units in R_k. Then C is a self-dual code.*

CHAPTER 4

CONSTRUCTION OF SELF-DUAL CODES OVER R_k

In this chapter we first deal with projections and lifts which enable us to translate codes between rings. Then we work through lifting of self-dual codes, particulary R_2 lifts of $[14, 7, 4]$ binary self-dual code.

4.1 PROJECTIONS AND LIFTS IN R_k

In (Karadeniz and Yıldız, 2013), $R_{k,i}$ was defined as follows: For $1 \leq i \leq k$, $R_{k,i} = R_k/\langle u_i \rangle$ that means

$$R_{k,i} = \mathbb{F}_2[u_1, u_2, \ldots, u_{i-1}, u_{i+1}, \ldots, u_k]/\langle u_1^2, \ldots, u_{i-1}^2, u_{i+1}^2, \ldots, u_k^2 \rangle.$$

Here $R_{k,k} = R_{k-1}$ and for any $i = 1, 2, \ldots, k$ the ring $R_{k,i}$ is isomorphic to R_{k-1}.

Now, we define $\pi_{k,i} : R_k \to R_{k,i}$, that is, the canonical projection where $\pi_{k,i}(a) \equiv a \pmod{u_i}$. This map can be extended to R_k^n. Therefore, if we have a linear code C over R_k of length n, then $\pi_{k,i}(C)$ is a linear code over $R_{k,i}$ of length n.

For example we can write that

$$\begin{aligned}
R_{4,1} &= \mathbb{F}_2 + u_2\mathbb{F}_2 + u_3\mathbb{F}_2 + u_4\mathbb{F}_2 + u_2u_3\mathbb{F}_2 + u_2u_4\mathbb{F}_2 + u_3u_4\mathbb{F}_2 + u_2u_3u_4\mathbb{F}_2, \\
R_{4,2} &= \mathbb{F}_2 + u_1\mathbb{F}_2 + u_3\mathbb{F}_2 + u_4\mathbb{F}_2 + u_1u_3\mathbb{F}_2 + u_1u_4\mathbb{F}_2 + u_3u_4\mathbb{F}_2 + u_1u_3u_4\mathbb{F}_2, \\
R_{4,3} &= \mathbb{F}_2 + u_1\mathbb{F}_2 + u_2\mathbb{F}_2 + u_4\mathbb{F}_2 + u_1u_2\mathbb{F}_2 + u_1u_4\mathbb{F}_2 + u_2u_4\mathbb{F}_2 + u_1u_2u_4\mathbb{F}_2,
\end{aligned}$$

$$R_{4,4} = \mathbb{F}_2 + u_1\mathbb{F}_2 + u_2\mathbb{F}_2 + u_3\mathbb{F}_2 + u_1u_2\mathbb{F}_2 + u_1u_3\mathbb{F}_2 + u_2u_3\mathbb{F}_2 + u_1u_2u_3\mathbb{F}_2$$

The projection acts on the ring as follows:

$$\begin{aligned}
\pi_{4,1}(1 + u_1 + u_4 + u_2u_3 + u_1u_2u_4 + u_1u_2u_3u_4) &= 1 + u_4 + u_2u_3 \\
\pi_{4,2}(1 + u_1 + u_4 + u_2u_3 + u_1u_2u_4 + u_1u_2u_3u_4) &= 1 + u_1 + u_4 \\
\pi_{4,3}(1 + u_1 + u_4 + u_2u_3 + u_1u_2u_4 + u_1u_2u_3u_4) &= 1 + u_1 + u_4 + u_1u_2u_4 \\
\pi_{4,4}(1 + u_1 + u_4 + u_2u_3 + u_1u_2u_4 + u_1u_2u_3u_4) &= 1 + u_1 + u_2u_3
\end{aligned}$$

Definition 4.1. *(Karadeniz and Yıldız, 2013) Let C be a linear code over R_k and D be a linear code over $R_{k,i}$ such that $\pi_{k,i}(C) = D$ for some i. Then we say D is a projection of C, and C is a lift of D.*

Theorem 4.1. *(Karadeniz and Yıldız, 2013) If C is a self-dual code over R_k of length n, then $\pi_{k,i}(C)$ is self-orthogonal for any $i = 1, 2, \ldots, k$.*

The projection of a self-dual code over R_k may not be self-dual over $R_{k,i}$. But if C is a free code then we have the following about its projection:

Corollary 4.1. *(Karadeniz and Yıldız, 2013) If C is a self-dual code generated over R_k by a matrix of the form $\left[I_{n/2}|A\right]$, then the projections of C are self-dual in $R_{k,i}$ for all $i = 1, 2, \ldots, k$.*

Lemma 4.1. *(Karadeniz and Yıldız, 2013) Let $\overline{c}_1, \overline{c}_2 \in R_k^n$ be such codes that satisfy $\langle \pi_{k,i}(\overline{c}_1), \pi_{k,i}(\overline{c}_2) \rangle_{k-1} = 0$ in $R_{k,i}$ for all $i = 1, 2, \ldots, k$. Then $\langle \overline{c}_1, \overline{c}_2 \rangle_k = 0$ or $u_1u_2 \cdots u_k$.*

The following theorem explains that for k elements, one in each $R_{k,i}$ for $i = 1, 2, \ldots, k$, the case that these elements have a common lift.

Theorem 4.2. *(Karadeniz and Yıldız, 2013) Let $a_1, a_2, \ldots, a_k$ be elements in $R_{k,1}, R_{k,2}, \ldots, R_{k,k}$ respectively. Then there exists $a \in R_k$ such that $\pi_{k,i}(a) = a_i$ for $i = 1, 2, \ldots, k$ if and only if for any $0 \leq j < k$ and for any $\{i_1, i_2, \cdots, i_j\} \subset \{1, 2, \cdots, k\}$, the term $u_{i_1}u_{i_2} \ldots u_{i_j}$ appears in either none of a_i's or in exactly $k - j$ of the a_i's. Here $j = 0$ corresponds to the term 1.*

The following theorem explains that how many different lifts can exist.

Theorem 4.3. *(Karadeniz and Yıldız, 2013) Suppose that $a_i \in R_{k,i}$ for $i = 1, 2, \ldots, k$ are given such that they have a common lift in R_k. Then there exist exactly two lifts of $a_1, a_2, \ldots, a_k$ to R_k, denoted by a and a' with $a' = a + u_1 u_2 \ldots u_k$.*

Note that the lifts of a self-dual code are also self-dual but all projections may not be self-dual.

4.2 LIFTING BINARY SELF-DUAL CODES TO R_k

We start this section with some preliminary definitions and lemmas which serve us as a fundamental tool for the lifting method.

Definition 4.2. *A matrix $[I_n|A]$ which generates a self-dual code is called a LRM (lift-ready-matrix) if each upper left $k \times k$ square submatrix of A denoted by A_k is invertible, i.e., has determinant a unit.*

Next we will show that R_k-lifts of a LRM always give us self-dual codes.

Lemma 4.2. *Let $G = [I_n|A]$ be a generating matrix of a binary self-dual code. By permuting certain columns, G can be put into lift ready form.*

Proof. Since $G = [I_n|A]$ generates a binary self-dual code, $AA^T = I_n$. A is an invertible $n \times n$ matrix with rank n, hence any upper left $k \times n$ submatrix of A $(k = 1, \ldots, n)$ has rank k. By using the equality of rowrank and columnrank, we can construct the lift ready matrix inductively as follows:

Denote by A_k, upper left $k \times k$ submatrix of A. First row of A contains a unit and permuting that column with the first one we get A_1, that is, 1×1 invertible matrix. Assume A_k is invertible. Take $(k+1) \times n$ submatrix of A, because A_k is invertible first k-columns are linearly independent. Knowing that columnrank is $k+1$, there is at least one column not in the span of first k-columns, by permuting $(k+1)^{th}$ column with any of them we obtain A_{k+1} which is invertible. □

Proposition 4.1. *Let A be an $n \times n$ binary matrix and A' be a lift of A to the ring R_k. If A is invertible, then A' is also invertible.*

Proof. Let $\pi : R_k \to \mathbb{F}_2$ be the ring homomorphism defined as follows:

$$\pi(a) \equiv a \mod (u_1, u_2, \ldots, u_k)$$

We need to show that $det(A')$ is a unit in R_k. It is given that $\pi(A') = A$ and A is an invertible binary matrix implies $det(A) = 1$. Note that $det(A) = det(\pi(A')) = \pi(det(A')) = 1$. Since π is a ring homomorphism, $det(A') \in R_k^*$. So A' is invertible. □

4.2.1 Cramer's Rule

Consider a system of n linear equations for n unknowns, represented in matrix multiplication form as follows

$$Ax = b$$

where the n by n matrix A is invertible, and the vector $x = (x_1, \ldots, x_n)^T$ is the column vector of unknowns. Then the rule states that in this case the system has a unique solution, whose individual values for the unknowns are given by

$$x_i = \frac{\det A_i}{\det A} \; i = 1, \ldots, n$$

where A_i is the matrix formed by replacing the i^{th} column of A by the column vector b. The rule holds for systems of equations with coefficients and unknowns in any ring.

4.2.2 Lifting Method

Let $G = [I|A]$ be the generating matrix of a binary self-dual code in lift ready form (LRM) and $G' = [I|A']$ be a lift of G over R_k. By (3), it is known that each row of G' is orthogonal to itself. Next we show that it is always possible to generate a self-dual code from R_k lifts of LRM by imposing some additional conditions on

entries of A'. Let $A' = \pi^{-1}(A)$, be a lift of the binary matrix A, where

$$A' = \begin{bmatrix} x_{11} & x_{12} & x_{13} & \dots & x_{1n} \\ y_{21} & x_{22} & x_{23} & \dots & x_{2n} \\ y_{31} & y_{32} & x_{33} & \dots & x_{3n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ y_{n1} & y_{n2} & y_{n3} & \dots & x_{nn} \end{bmatrix} \in M_n(R_k).$$

Firstly, initialize all x_{ij}'s where $i \leq j$, i.e., upper triangular part of A. Note that 1's and 0's are lifted to units and non-units respectively in R_k. The question now is how to determine unknowns y_{ij}'s $(i > j)$ so that orthogonality of rows is preserved. To do so, we make iterative use of Cramer's rule which gives us a unique solution for each initialization of x_{ij}'s.

Denote i^{th} row of A' by A'_{R_i} and upper left $k \times k$ submatrix of A' by A'_k. Since A is a LRM, so is A' by 4.1 which implies that A'_k is invertible for each $k = 1, 2, \dots, n$.

We can determine the unknown y_{21} by using the condition

$$\langle A'_{R_1}, A'_{R_2} \rangle_k = x_{11}.y_{21} + \sum_{i=2}^{n} x_{1i}x_{2i} = 0.$$

So, $y_{21} = x_{11}.\sum_{i=2}^{n} x_{1i}x_{2i}$. Similarly, to find the unknowns y_{31} and y_{32}, we use

$$\langle A'_{R_1}, A'_{R_3} \rangle_k = x_{11}.y_{31} + x_{12}.y_{32} + \sum_{i=3}^{n} x_{1i}x_{3i} = 0.$$

$$\langle A'_{R_2}, A'_{R_3} \rangle_k = y_{21}.y_{31} + x_{22}.y_{32} + \sum_{i=3}^{n} x_{2i}x_{3i} = 0.$$

which is equivalent to the linear system of equations.

$$\begin{bmatrix} x_{11} & x_{12} \\ y_{21} & x_{22} \end{bmatrix} \begin{bmatrix} y_{31} \\ y_{32} \end{bmatrix} = \begin{bmatrix} \sum_{i=3}^{n} x_{1i}x_{3i} \\ \sum_{i=3}^{n} x_{2i}x_{3i} \end{bmatrix}$$

Since $A'_2 = \begin{bmatrix} x_{11} & x_{12} \\ y_{21} & x_{22} \end{bmatrix}$ is invertible, the unknowns y_{31}, y_{32} can be found

uniquely by Cramer's rule.

In general, to determine $y_{k1}, y_{k2}, \ldots, y_{k(k-1)}$ we need to solve the system $A'_{k-1}Y{=}b$

where $Y = \begin{bmatrix} y_{k1} \\ y_{k2} \\ \vdots \\ y_{k(k-1)} \end{bmatrix}$ and $b = \begin{bmatrix} \sum_{i=k}^{n} x_{1i}x_{ki} \\ \sum_{i=k}^{n} x_{2i}x_{ki} \\ \vdots \\ \sum_{i=k}^{n} x_{(k-1)i}x_{ki} \end{bmatrix}, 1 < k < n.$

Let $(A'_k)_j$ be the matrix obtained from (A'_k) by replacing j^{th} column by b. Then

$$y_{kj} = \frac{|(A'_k)_j|}{|(A'_k)|} = det(A'_k)_j.det(A'_k)$$

All unknowns can be found by the same way. Finally, we end up with $G' = [I|A']$ which generates a self-dual code over R_k. Hence, a unique R_k self-dual code corresponds to each different choice of upper triangular part of A'.

Corollary 4.2. *Let C be a binary self-dual code of length $2n$. Then it has $\left(2^{2^k-1}\right)^{\frac{n(n+1)}{2}}$ different possible lifts.*

Proof. If C is a binary self-dual code of length $2n$ with a generator matrix $[I_n|A]$, then upper triangular part of A has $\frac{n(n+1)}{2}$ entries. R_k has 2^{2^k} elements but half of them are units, the others are nonunits. Since we lift units in $\mathbb{F}_2$ to the units in R_k, there are 2^{2^k-1} possible lifts for each entry. Hence, C has $\left(2^{2^k-1}\right)^{\frac{n(n+1)}{2}}$ different possible lifts. □

Here, we give two examples about self-dual lifts. First one is obtained by lifting the extended binary Hamming code to the ring R_1 and second one is obtained by lifting the extended binary Hamming code to the ring R_2.

Example 4.1. *We started with the $[8, 4, 4]$ extended binary Hamming code and by using the method, we obtained $(8, 2^8, 4)$ R_1 self-dual code.*

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1+u & 1+u \\ 0 & 1 & 0 & 0 & 0 & 1 & 1+u & 1+u \\ 0 & 0 & 1 & 0 & 1+u & 1+u & 1 & 0 \\ 0 & 0 & 0 & 1 & 1+u & 1+u & 0 & 1 \end{bmatrix}$$

Example 4.2. *We again used the same Hamming code to obtain* $[32, 16, 8]$ *codes. We first lifted this matrix to the ring* R_1 *then to the ring* R_2*.*

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 1+u+v+uv & v & 1+u+v+uv & 1+uv \\ 0 & 1 & 0 & 0 & u+v+uv & 1+uv & 1+u+v & 1+u+uv \\ 0 & 0 & 1 & 0 & 1+u+v+uv & 1+u & 1+u & uv \\ 0 & 0 & 0 & 1 & 1+u+v & 1 & u & 1+u+v+uv \end{bmatrix}$$

After taking the image under the Gray map, we obtain the code of length 32 with minimum weight 8.

4.3 R_2 LIFTS OF $[14, 7, 4]$ BINARY SELF-DUAL CODE

4.3.1 The weight enumerators for extremal self-dual codes of length 58

It was described in (Conway and Sloane, 1990) that there are two types of weight enumerators for extremal self-dual codes of length 58:

$$W_{58,1} = 1 + (165 - 2\beta)y^{10} + (5078 + 2\beta)y^{12} + \cdots$$

where $0 \leq \beta \leq 82$, and

$$W_{58,2} = 1 + (319 - 24\beta - 2\gamma)y^{10} + (3132 + 152\beta + 2\gamma)y^{12} + \cdots$$

where $0 \leq \beta \leq 11$ and $0 \leq \gamma \leq 159 - 2\beta$. In (Bouyuklieva and Bouyukliev, 1998), (Harada and Kimura, 1995), (Kim et al., 2011), (Tsai and Jiang, 1998), (Yankov and Russeva, 2011) and (Karadeniz and Kaya, 2012), new extremal self-dual codes of length 58 are obtained. Recently, in (Yankov and Lee, 2013), the authors indicate the known binary self-dual codes of length 58 and they obtain new ones. Together with the ones added from (Karadeniz and Kaya, 2012) and (Yankov and Lee, 2013),

the existence of such codes is known for $\beta = 55$ in $W_{58,1}$ and for

$$\beta = 0 \text{ with } \gamma \in \{2m | m = 0, 1, 5, 6, 8, 9, 10, 11, 13, 68, 71, 79 \text{ or } 15 \leq m \leq 65\},$$
$$\beta = 1 \text{ with } \gamma \in \{2m | m = 13, 18, 53, 58, 63 \text{ or } 21 \leq m \leq 57\},$$
$$\beta = 2 \text{ with } \gamma \in \{2m | m = 0, 16, 18, 19, 20, 21, 22, 46, 49, 50, 55 \text{ or } 24 \leq m \leq 44\}$$

in $W_{58,2}$.

4.3.2 Lifting $[14, 7, 4]$ binary self-dual codes to the ring R_2

In this section, as an application of the lifting method, we look for R_2 self-dual lifts of $[14, 7, 4]$ binary self-dual code. $G = [I_7|A]$ where

$$A = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \end{bmatrix}$$

is the generator matrix of the $[14, 7, 4]$ code. It is obvious that G is in lift ready form. By Corollary 4.2, we know that there are 2^{84} possible self-dual lifts of G to R_2. That is why, a randomized search algorithm is used. Here, we sum up the algorithm to compute self-dual lifts of G.

Input: $[I_7|A]$
Convert G in LRM (if necessary)
Initialize a_{ij}, $i \leq j$
Calculate a_{ij}, $i > j$ by Cramer's rule.
Output: R_2 self-dual lift of G.

We run the randomized search algorithm on a usual PC for about a day and find the following. There are at least 61 R_2 self-dual $(14, 2^{28}, 10)$ codes with different weight enumerators as lifts of the $[14, 7, 4]$ binary code. As the Gray images of these

codes via the map ϕ_2, we obtained $[56, 28, 10]$ binary self-dual codes. The generator matrices G_i, $i \in \{1, 2, \ldots, 61\}$ are available at (Karadeniz and Aksoy). And then by applying the extension method given in (Bouyuklieva and Bouyukliev, 2012) by Bouyuklieva and Bouyukliev, we got binary self-dual codes of length 58 for $\beta = 55$ in $W_{58,1}$ and for

$$\beta = 0 \text{ with } \gamma \in \{2m | m = 14 \text{ or } 17 \leq m \leq 56\},$$
$$\beta = 1 \text{ with } \gamma \in \{2m | 16 \leq m \leq 50\},$$
$$\beta = 2 \text{ with } \gamma \in \{2m | m = 17, 47, 48 \text{ or } 19 \leq m \leq 45\}$$

in $W_{58,2}$.

More importantly, ten new $[58, 29, 10]$ binary self-dual codes are found with unknown weight enumerators.

CHAPTER 5

RESULTS AND CONCLUSIONS

5.1 RESULTS

We first started with the $[14, 7, 4]$ binary self-dual code and lifted this code to the ring R_2. As the Gray images of these codes via the map ϕ_2, we obtained $[56, 28, 10]$ binary self-dual codes over $\mathbb{F}_2$ with different weight enumerators and minimum distance 10. Then by extending them with suitable vectors, we obtained extremal $[58, 29, 10]$ binary codes. Ten of them are new that are for $\beta = 0$ with $\gamma = 28$, $\beta = 1$ with $\gamma = 32, 34, 38, 40$ and $\beta = 2$ with $\gamma = 34, 46, 90, 94, 96$ in $W_{58,2}$.

i	X	β	γ
G_{55}	01101011011000111011010 11011	0	28
G_{57}	00011010011111111001100000100	1	32
G_{55}	10000000001111101111010100 01	1	34
G_{47}	00011000110001110101100 11001	1	38
G_{29}	11001100101100000011101001 10	1	40
G_{57}	00001011010110010110010 11111	2	34
G_{36}	01010101001000001011000 11100	2	46
G_{32}	11101010001111001010100 11010	2	90
G_{54}	01101000100001010000111 11110	2	94
G_{61}	00010010011010100010101 01010	2	96

Table 5.1 New $[58, 29, 10]$ binary self-dual codes

In Table 5.1, G_i's are binary generator matrices that are obtained as Gray images of R_2-lifts for $i \in \{1, 2, \ldots, 61\}$. The extension of Bouyuklieva and Bouyukliev is applied to G_i with binary vector X and extremal codes of length 58 with new enumerators are found. All the results were obtained by using Magma Package (Bosma et al., 1997).

Example 5.1. *If we extend* $G_{57} = [I_{28}|A]$ *generator matrix with the vector* $X = 00001011010110010110010111111$ *we obtained the code whose* $\beta = 2$ *and* $\gamma = 34$ *where* A *is*

$$A = \begin{bmatrix}
0&1&0&1&0&0&0&0&0&0&0&1&1&0&1&0&1&0&1&0&1&0&1&0&1&1&0&0\\
1&1&0&1&0&0&0&1&1&0&0&0&1&0&0&1&1&0&0&1&1&0&0&1&1&0&1&0\\
0&0&0&1&0&1&1&0&1&1&1&1&0&0&1&1&1&1&0&0&1&0&1&1&0&0&0&1\\
1&1&1&1&0&1&1&0&0&0&1&1&1&0&0&0&1&0&1&1&1&0&1&0&0&0&1&0\\
0&0&0&0&0&0&0&0&1&0&1&1&0&0&1&1&0&0&1&0&0&0&1&0&0&0&1&1&0\\
0&0&1&1&0&0&0&0&1&1&0&0&1&1&0&1&0&1&0&1&0&1&0&1&0&0&0&0\\
0&0&1&1&0&0&0&1&1&1&0&1&0&0&1&1&1&1&0&0&0&1&1&1&1&0&1&0\\
0&1&0&0&1&0&1&0&0&1&1&0&1&1&0&0&0&0&1&1&1&0&1&0&0&1&0&1\\
0&1&1&0&0&1&1&0&1&1&0&0&0&0&1&0&1&1&0&0&1&1&1&1&0&1&1&0\\
0&0&1&0&1&1&1&1&1&1&1&0&0&0&1&1&0&0&1&1&0&1&1&0&0&1&1&1\\
0&0&1&1&1&0&0&1&0&1&0&1&1&1&1&1&0&0&0&0&1&0&0&0&0&1&1&0\\
1&0&1&1&0&0&1&0&0&1&1&0&0&0&1&0&0&0&1&0&0&1&1&0&1&1&1&0\\
1&1&0&1&0&1&0&1&0&0&1&0&1&1&0&1&0&1&1&0&1&0&1&1&1&1&1&0\\
0&0&0&0&1&1&0&1&0&0&1&0&1&1&0&0&1&0&0&0&0&1&0&1&0&0&0&0\\
1&0&1&0&1&0&1&0&1&0&1&1&0&0&0&1&0&1&0&0&0&0&0&0&0&1&1&0\\
0&1&1&0&0&1&1&0&0&1&1&0&1&0&1&1&0&1&0&0&0&1&1&0&0&0&1&0\\
1&1&1&1&0&0&1&0&1&1&0&0&0&1&0&0&0&1&0&1&1&0&1&1&1&1&0&0\\
0&0&1&0&1&1&1&0&1&0&0&0&1&0&1&1&1&1&0&1&1&0&0&0&1&1&1&0\\
1&0&0&1&0&0&0&0&1&0&0&0&1&1&0&0&0&0&0&0&0&0&1&0&1&1&0&1\\
0&1&0&1&0&1&0&1&0&1&0&0&0&0&0&0&0&1&1&0&0&0&0&0&1&1&0&0&1&1\\
1&1&1&1&0&0&0&1&1&1&1&0&1&0&0&0&1&1&0&0&0&1&1&1&0&1&0&0\\
0&0&0&0&1&1&1&0&1&0&0&1&0&1&0&1&0&0&1&0&1&0&0&1&1&0&1&1\\
1&0&1&1&0&0&1&1&1&1&0&1&1&0&0&1&1&0&0&1&1&0&1&1&0&0&0&0\\
0&1&1&0&0&1&1&0&1&1&0&0&1&1&0&0&1&0&1&1&1&1&1&1&1&1&0&0\\
1&1&0&0&0&0&1&0&0&0&0&1&1&0&0&0&1&1&1&0&0&1&0&1&0&1&1&1\\
1&0&0&0&1&0&0&1&1&0&1&1&1&0&1&0&1&1&0&0&1&0&0&1&1&0&0&0\\
0&1&0&1&1&0&1&0&1&1&1&1&1&0&1&1&0&1&0&1&0&1&0&0&1&0&1&1\\
0&0&1&0&0&0&0&1&0&1&0&0&0&0&0&0&0&0&1&1&0&1&0&0&1&0&1&1
\end{bmatrix}$$

Figure 5.1 The Matrix A of G_{57}

As a result, the set of known codes of length 58 with different weight enumerators is expanded.

Theorem 5.1. *There exist binary self-dual* $[58, 29, 10]$ *codes with weight enumerator for* $\beta = 55$ *in* $W_{58,1}$ *and*

$$\begin{aligned} \text{for } \beta &= 0 \text{ with } \gamma \in \{2m | m = 0, 1, 5, 6, 8, 9, 10, 11, 13, 68, 71, 79 \text{ or } 13 \leq m \leq 65\}, \\ \beta &= 1 \text{ with } \gamma \in \{2m | m = 13, 63 \text{ or } 16 \leq m \leq 58\}, \\ \text{and } \beta &= 2 \text{ with } \gamma \in \{2m | m = 0, 55 \text{ or } 16 \leq m \leq 50\} \text{ in } W_{58,2}. \end{aligned}$$

5.1.1 The Generator Matrices

The generator matrices that we obtained by lifting $[14, 7, 4]$ binary self-dual code to the ring R_2 are listed below. In order to fit the matrices to the page, we represent the elements of R_2 as follows:

$$0 \longrightarrow 0$$

$$1 \longrightarrow 1$$

$$u \longrightarrow 2$$

$$1 + u \longrightarrow 3$$

$$v \longrightarrow 4$$

$$1 + v \longrightarrow 5$$

$$u + v \longrightarrow 6$$

$$1 + u + v \longrightarrow 7$$

$$uv \longrightarrow 8$$

$$1 + uv \longrightarrow 9$$

$$u + uv \longrightarrow 10$$

$$v + uv \longrightarrow 11$$

$$1 + u + uv \longrightarrow 12$$

$$1 + v + uv \longrightarrow 13$$

$$u + v + uv \longrightarrow 14$$

$$1 + u + v + uv \longrightarrow 15$$

$G_1 = [I_7|A]$ where

$$A = \begin{bmatrix} 5 & 3 & 9 & 10 & 10 & 7 & 7 \\ 12 & 8 & 15 & 5 & 8 & 14 & 11 \\ 4 & 9 & 3 & 15 & 14 & 13 & 3 \\ 1 & 1 & 4 & 1 & 2 & 15 & 7 \\ 10 & 11 & 8 & 8 & 13 & 9 & 13 \\ 4 & 9 & 2 & 8 & 9 & 12 & 10 \\ 14 & 15 & 14 & 4 & 13 & 8 & 9 \end{bmatrix},$$

$G_2 = [I_7|A]$ where

$$A = \begin{bmatrix} 15 & 3 & 5 & 14 & 14 & 15 & 9 \\ 9 & 10 & 7 & 12 & 11 & 8 & 0 \\ 10 & 9 & 13 & 9 & 6 & 15 & 7 \\ 9 & 15 & 14 & 15 & 10 & 15 & 12 \\ 10 & 11 & 2 & 11 & 7 & 1 & 13 \\ 6 & 15 & 11 & 11 & 13 & 1 & 14 \\ 11 & 5 & 2 & 8 & 1 & 0 & 7 \end{bmatrix},$$

$G_3 = [I_7|A]$ where

$$A = \begin{bmatrix} 3 & 7 & 9 & 14 & 8 & 9 & 15 \\ 1 & 10 & 3 & 9 & 11 & 2 & 6 \\ 11 & 12 & 1 & 9 & 11 & 9 & 9 \\ 15 & 12 & 11 & 13 & 10 & 1 & 5 \\ 8 & 4 & 10 & 2 & 1 & 1 & 9 \\ 2 & 9 & 11 & 2 & 13 & 13 & 11 \\ 4 & 7 & 14 & 10 & 9 & 6 & 3 \end{bmatrix},$$

$G_4 = [I_7|A]$ where

$$A = \begin{bmatrix} 13 & 15 & 1 & 8 & 6 & 13 & 9 \\ 7 & 0 & 15 & 3 & 10 & 0 & 11 \\ 10 & 7 & 3 & 9 & 11 & 12 & 15 \\ 9 & 15 & 14 & 7 & 14 & 9 & 15 \\ 11 & 6 & 11 & 14 & 9 & 5 & 1 \\ 6 & 1 & 14 & 10 & 7 & 9 & 11 \\ 0 & 3 & 10 & 11 & 7 & 6 & 15 \end{bmatrix},$$

$G_5 = [I_7|A]$ where

$$A = \begin{bmatrix} 1 & 7 & 7 & 14 & 10 & 13 & 9 \\ 12 & 2 & 5 & 3 & 4 & 2 & 14 \\ 0 & 12 & 7 & 15 & 2 & 7 & 1 \\ 9 & 5 & 8 & 15 & 10 & 13 & 13 \\ 11 & 4 & 10 & 14 & 3 & 13 & 1 \\ 14 & 15 & 10 & 8 & 9 & 12 & 0 \\ 6 & 15 & 4 & 10 & 15 & 6 & 15 \end{bmatrix},$$

$G_6 = [I_7|A]$ where

$$A = \begin{bmatrix} 7 & 1 & 9 & 10 & 11 & 5 & 13 \\ 5 & 6 & 12 & 9 & 4 & 4 & 0 \\ 14 & 7 & 3 & 5 & 10 & 15 & 15 \\ 5 & 5 & 6 & 13 & 2 & 3 & 1 \\ 11 & 14 & 6 & 2 & 7 & 15 & 15 \\ 2 & 9 & 11 & 8 & 9 & 13 & 2 \\ 11 & 9 & 8 & 14 & 12 & 0 & 5 \end{bmatrix},$$

$G_7 = [I_7|A]$ where

$$A = \begin{bmatrix} 3 & 12 & 13 & 10 & 11 & 7 & 9 \\ 15 & 14 & 1 & 3 & 11 & 10 & 6 \\ 8 & 15 & 13 & 13 & 14 & 12 & 3 \\ 12 & 13 & 0 & 5 & 6 & 15 & 13 \\ 14 & 14 & 4 & 2 & 12 & 5 & 3 \\ 4 & 15 & 10 & 6 & 9 & 13 & 4 \\ 11 & 9 & 0 & 0 & 5 & 6 & 5 \end{bmatrix},$$

$G_8 = [I_7|A]$ where

$$A = \begin{bmatrix} 12 & 1 & 1 & 10 & 6 & 5 & 15 \\ 3 & 6 & 15 & 15 & 11 & 10 & 8 \\ 10 & 15 & 3 & 13 & 10 & 5 & 12 \\ 12 & 7 & 11 & 15 & 11 & 7 & 13 \\ 11 & 8 & 6 & 11 & 12 & 12 & 13 \\ 14 & 7 & 10 & 11 & 1 & 3 & 2 \\ 4 & 12 & 11 & 10 & 5 & 10 & 9 \end{bmatrix},$$

$G_9 = [I_7|A]$ where

$$A = \begin{bmatrix} 9 & 1 & 15 & 6 & 2 & 5 & 7 \\ 12 & 11 & 13 & 7 & 11 & 14 & 4 \\ 14 & 1 & 1 & 12 & 4 & 7 & 3 \\ 3 & 3 & 14 & 7 & 14 & 9 & 12 \\ 2 & 14 & 6 & 10 & 1 & 7 & 5 \\ 4 & 7 & 10 & 8 & 15 & 1 & 14 \\ 14 & 12 & 14 & 11 & 5 & 8 & 7 \end{bmatrix},$$

$G_{10} = [I_7|A]$ where

$$A = \begin{bmatrix} 9 & 9 & 1 & 10 & 0 & 7 & 1 \\ 7 & 6 & 12 & 7 & 10 & 0 & 0 \\ 0 & 12 & 12 & 1 & 4 & 5 & 9 \\ 12 & 12 & 4 & 1 & 8 & 15 & 7 \\ 11 & 0 & 11 & 2 & 3 & 9 & 15 \\ 14 & 15 & 0 & 2 & 5 & 1 & 14 \\ 8 & 12 & 10 & 14 & 13 & 10 & 12 \end{bmatrix},$$

$G_{11} = [I_7|A]$ where

$$A = \begin{bmatrix} 12 & 9 & 1 & 4 & 10 & 13 & 12 \\ 5 & 2 & 12 & 13 & 0 & 6 & 11 \\ 14 & 5 & 12 & 12 & 11 & 12 & 1 \\ 5 & 15 & 10 & 7 & 2 & 15 & 15 \\ 4 & 6 & 8 & 6 & 3 & 5 & 3 \\ 0 & 9 & 8 & 11 & 7 & 5 & 2 \\ 11 & 7 & 6 & 4 & 3 & 8 & 13 \end{bmatrix},$$

$G_{12} = [I_7|A]$ where

$$A = \begin{bmatrix} 5 & 1 & 3 & 6 & 6 & 9 & 3 \\ 1 & 11 & 5 & 9 & 11 & 10 & 2 \\ 11 & 15 & 9 & 5 & 2 & 3 & 15 \\ 3 & 3 & 14 & 9 & 0 & 9 & 7 \\ 14 & 4 & 6 & 11 & 12 & 7 & 15 \\ 14 & 12 & 10 & 14 & 1 & 15 & 14 \\ 10 & 7 & 4 & 11 & 5 & 14 & 3 \end{bmatrix},$$

$G_{13} = [I_7|A]$ where

$$A = \begin{bmatrix} 3 & 12 & 7 & 11 & 14 & 1 & 9 \\ 3 & 14 & 7 & 12 & 4 & 14 & 4 \\ 4 & 5 & 7 & 3 & 8 & 3 & 5 \\ 7 & 13 & 11 & 7 & 8 & 13 & 7 \\ 2 & 11 & 6 & 2 & 12 & 1 & 7 \\ 8 & 7 & 14 & 2 & 12 & 15 & 2 \\ 8 & 5 & 2 & 11 & 3 & 0 & 12 \end{bmatrix},$$

$G_{14} = [I_7|A]$ where

$$A = \begin{bmatrix} 15 & 3 & 12 & 11 & 2 & 1 & 1 \\ 3 & 6 & 15 & 12 & 4 & 8 & 2 \\ 6 & 3 & 12 & 5 & 14 & 13 & 15 \\ 13 & 13 & 6 & 7 & 14 & 9 & 7 \\ 4 & 14 & 8 & 2 & 3 & 12 & 3 \\ 0 & 15 & 2 & 0 & 12 & 1 & 4 \\ 11 & 5 & 10 & 6 & 5 & 8 & 12 \end{bmatrix},$$

$G_{15} = [I_7|A]$ where

$$A = \begin{bmatrix} 13 & 15 & 1 & 0 & 10 & 15 & 5 \\ 1 & 2 & 3 & 12 & 6 & 6 & 2 \\ 10 & 12 & 5 & 1 & 4 & 1 & 9 \\ 5 & 12 & 11 & 5 & 6 & 5 & 15 \\ 6 & 2 & 6 & 2 & 9 & 3 & 9 \\ 8 & 15 & 14 & 4 & 3 & 13 & 14 \\ 11 & 7 & 8 & 10 & 9 & 4 & 7 \end{bmatrix},$$

$G_{16} = [I_7|A]$ where

$$A = \begin{bmatrix} 13 & 1 & 12 & 0 & 11 & 13 & 12 \\ 12 & 8 & 9 & 12 & 14 & 0 & 4 \\ 2 & 12 & 7 & 3 & 8 & 9 & 12 \\ 7 & 15 & 14 & 5 & 11 & 12 & 7 \\ 0 & 6 & 10 & 8 & 5 & 15 & 1 \\ 14 & 13 & 4 & 11 & 3 & 13 & 2 \\ 8 & 15 & 11 & 14 & 7 & 11 & 1 \end{bmatrix},$$

$G_{17} = [I_7|A]$ where

$$A = \begin{bmatrix} 3 & 5 & 13 & 4 & 14 & 13 & 15 \\ 15 & 6 & 1 & 9 & 0 & 2 & 8 \\ 8 & 5 & 1 & 5 & 0 & 3 & 9 \\ 12 & 15 & 0 & 9 & 11 & 12 & 7 \\ 14 & 0 & 4 & 8 & 3 & 7 & 9 \\ 2 & 12 & 11 & 10 & 7 & 3 & 8 \\ 6 & 13 & 6 & 6 & 3 & 2 & 12 \end{bmatrix},$$

$G_{18} = [I_7|A]$ where

$$A = \begin{bmatrix} 7 & 9 & 7 & 8 & 6 & 12 & 1 \\ 9 & 11 & 7 & 5 & 14 & 4 & 6 \\ 14 & 5 & 12 & 1 & 11 & 1 & 13 \\ 15 & 7 & 8 & 5 & 11 & 15 & 3 \\ 8 & 2 & 8 & 11 & 12 & 3 & 5 \\ 6 & 7 & 10 & 10 & 5 & 9 & 6 \\ 10 & 1 & 6 & 0 & 15 & 2 & 9 \end{bmatrix},$$

$G_{19} = [I_7|A]$ where

$$A = \begin{bmatrix} 12 & 7 & 12 & 0 & 14 & 1 & 13 \\ 5 & 2 & 3 & 12 & 8 & 11 & 8 \\ 11 & 3 & 3 & 9 & 2 & 7 & 12 \\ 5 & 5 & 10 & 15 & 4 & 12 & 9 \\ 0 & 10 & 0 & 11 & 3 & 7 & 7 \\ 4 & 3 & 8 & 10 & 9 & 5 & 10 \\ 6 & 15 & 4 & 8 & 7 & 8 & 1 \end{bmatrix},$$

$G_{20} = [I_7|A]$ where

$$A = \begin{bmatrix} 5 & 3 & 15 & 8 & 6 & 12 & 15 \\ 13 & 8 & 3 & 12 & 10 & 11 & 0 \\ 10 & 13 & 13 & 3 & 4 & 9 & 12 \\ 5 & 1 & 6 & 9 & 14 & 12 & 12 \\ 2 & 11 & 2 & 14 & 13 & 9 & 7 \\ 4 & 7 & 0 & 2 & 5 & 13 & 8 \\ 11 & 3 & 2 & 4 & 1 & 8 & 7 \end{bmatrix},$$

$G_{21} = [I_7|A]$ where

$$A = \begin{bmatrix} 3 & 13 & 5 & 4 & 4 & 15 & 12 \\ 7 & 6 & 12 & 9 & 8 & 4 & 4 \\ 4 & 5 & 9 & 9 & 14 & 15 & 7 \\ 15 & 5 & 14 & 1 & 6 & 3 & 9 \\ 11 & 11 & 6 & 2 & 7 & 5 & 12 \\ 11 & 12 & 0 & 6 & 15 & 1 & 0 \\ 10 & 7 & 11 & 4 & 12 & 8 & 12 \end{bmatrix},$$

$G_{22} = [I_7|A]$ where

$$A = \begin{bmatrix} 9 & 7 & 9 & 11 & 6 & 3 & 9 \\ 12 & 14 & 3 & 13 & 8 & 8 & 2 \\ 2 & 5 & 1 & 1 & 14 & 3 & 12 \\ 7 & 3 & 2 & 15 & 4 & 3 & 12 \\ 11 & 14 & 14 & 0 & 1 & 15 & 1 \\ 14 & 13 & 2 & 2 & 5 & 1 & 11 \\ 0 & 12 & 11 & 8 & 1 & 14 & 1 \end{bmatrix},$$

$G_{23} = [I_7|A]$ where

$$A = \begin{bmatrix} 15 & 1 & 7 & 8 & 14 & 13 & 1 \\ 7 & 0 & 12 & 15 & 4 & 14 & 10 \\ 2 & 13 & 15 & 13 & 2 & 1 & 3 \\ 12 & 5 & 6 & 9 & 11 & 15 & 5 \\ 11 & 4 & 2 & 6 & 7 & 3 & 3 \\ 6 & 12 & 6 & 10 & 9 & 5 & 11 \\ 4 & 12 & 11 & 14 & 5 & 8 & 13 \end{bmatrix},$$

$G_{24} = [I_7|A]$ where

$$A = \begin{bmatrix} 9 & 5 & 9 & 11 & 6 & 7 & 9 \\ 15 & 4 & 12 & 15 & 10 & 14 & 2 \\ 6 & 3 & 15 & 13 & 0 & 15 & 3 \\ 5 & 9 & 2 & 7 & 10 & 1 & 1 \\ 4 & 0 & 10 & 8 & 9 & 1 & 7 \\ 14 & 15 & 0 & 14 & 5 & 7 & 10 \\ 11 & 13 & 14 & 14 & 12 & 14 & 12 \end{bmatrix},$$

$G_{25} = [I_7|A]$ where

$$A = \begin{bmatrix} 12 & 5 & 15 & 4 & 6 & 1 & 7 \\ 7 & 0 & 12 & 7 & 4 & 6 & 2 \\ 6 & 13 & 5 & 12 & 10 & 1 & 7 \\ 15 & 13 & 14 & 5 & 2 & 3 & 12 \\ 2 & 11 & 11 & 6 & 15 & 13 & 15 \\ 4 & 7 & 6 & 8 & 1 & 5 & 10 \\ 6 & 1 & 10 & 2 & 5 & 8 & 1 \end{bmatrix},$$

$G_{26} = [I_7|A]$ where

$$A = \begin{bmatrix} 15 & 7 & 7 & 2 & 4 & 3 & 7 \\ 5 & 14 & 13 & 13 & 4 & 8 & 4 \\ 0 & 3 & 1 & 3 & 14 & 3 & 7 \\ 7 & 3 & 8 & 13 & 11 & 13 & 7 \\ 6 & 14 & 10 & 11 & 1 & 5 & 15 \\ 8 & 3 & 11 & 14 & 13 & 9 & 6 \\ 2 & 15 & 0 & 11 & 1 & 6 & 7 \end{bmatrix},$$

$G_{27} = [I_7|A]$ where

$$A = \begin{bmatrix} 1 & 13 & 5 & 4 & 0 & 5 & 5 \\ 12 & 11 & 9 & 15 & 6 & 4 & 10 \\ 8 & 13 & 5 & 1 & 2 & 12 & 7 \\ 15 & 3 & 6 & 15 & 8 & 15 & 13 \\ 2 & 2 & 14 & 11 & 9 & 15 & 9 \\ 10 & 13 & 2 & 14 & 1 & 7 & 10 \\ 6 & 1 & 0 & 6 & 5 & 10 & 5 \end{bmatrix},$$

$G_{28} = [I_7|A]$ where

$$A = \begin{bmatrix} 13 & 7 & 15 & 4 & 4 & 12 & 3 \\ 7 & 10 & 7 & 3 & 0 & 11 & 0 \\ 0 & 1 & 5 & 15 & 14 & 3 & 12 \\ 15 & 9 & 10 & 15 & 2 & 7 & 13 \\ 8 & 14 & 14 & 2 & 13 & 1 & 5 \\ 2 & 15 & 2 & 14 & 5 & 3 & 4 \\ 11 & 1 & 14 & 8 & 5 & 10 & 9 \end{bmatrix},$$

$G_{29} = [I_7|A]$ where

$$A = \begin{bmatrix} 13 & 1 & 15 & 6 & 6 & 12 & 15 \\ 13 & 2 & 7 & 7 & 4 & 10 & 14 \\ 2 & 13 & 13 & 9 & 6 & 9 & 7 \\ 12 & 7 & 2 & 5 & 11 & 1 & 1 \\ 14 & 4 & 2 & 11 & 5 & 3 & 9 \\ 2 & 13 & 2 & 11 & 12 & 12 & 2 \\ 2 & 15 & 4 & 14 & 1 & 2 & 12 \end{bmatrix},$$

$G_{30} = [I_7|A]$ where

$$A = \begin{bmatrix} 7 & 1 & 9 & 10 & 8 & 13 & 12 \\ 9 & 10 & 15 & 13 & 6 & 2 & 10 \\ 2 & 12 & 13 & 7 & 2 & 5 & 13 \\ 5 & 1 & 10 & 1 & 11 & 1 & 5 \\ 8 & 14 & 2 & 4 & 15 & 13 & 7 \\ 14 & 13 & 14 & 6 & 13 & 5 & 11 \\ 11 & 9 & 0 & 10 & 9 & 4 & 12 \end{bmatrix},$$

$G_{31} = [I_7|A]$ where

$$A = \begin{bmatrix} 1 & 13 & 5 & 6 & 11 & 12 & 3 \\ 3 & 14 & 3 & 13 & 2 & 11 & 8 \\ 0 & 13 & 12 & 15 & 8 & 13 & 5 \\ 7 & 13 & 11 & 7 & 2 & 12 & 15 \\ 4 & 11 & 2 & 8 & 1 & 12 & 5 \\ 6 & 3 & 10 & 11 & 3 & 13 & 0 \\ 6 & 15 & 14 & 11 & 13 & 2 & 7 \end{bmatrix},$$

$G_{32} = [I_7|A]$ where

$$A = \begin{bmatrix} 13 & 13 & 9 & 14 & 8 & 9 & 1 \\ 3 & 0 & 15 & 5 & 10 & 11 & 10 \\ 11 & 12 & 15 & 7 & 4 & 15 & 13 \\ 7 & 15 & 11 & 12 & 0 & 5 & 15 \\ 14 & 6 & 2 & 0 & 13 & 5 & 15 \\ 0 & 9 & 2 & 11 & 1 & 3 & 11 \\ 11 & 13 & 14 & 10 & 7 & 6 & 5 \end{bmatrix},$$

$G_{33} = [I_7|A]$ where

$$A = \begin{bmatrix} 1 & 3 & 12 & 4 & 11 & 1 & 1 \\ 13 & 0 & 3 & 15 & 14 & 11 & 11 \\ 0 & 12 & 13 & 1 & 14 & 15 & 5 \\ 15 & 7 & 11 & 1 & 2 & 13 & 7 \\ 10 & 14 & 6 & 2 & 15 & 15 & 1 \\ 11 & 15 & 14 & 8 & 13 & 1 & 2 \\ 14 & 5 & 11 & 0 & 9 & 11 & 5 \end{bmatrix},$$

$G_{34} = [I_7|A]$ where

$$A = \begin{bmatrix} 13 & 15 & 7 & 0 & 2 & 7 & 9 \\ 15 & 6 & 13 & 1 & 0 & 4 & 2 \\ 11 & 5 & 3 & 12 & 4 & 3 & 15 \\ 13 & 13 & 4 & 15 & 2 & 13 & 13 \\ 8 & 2 & 2 & 4 & 15 & 9 & 5 \\ 10 & 3 & 10 & 10 & 5 & 7 & 14 \\ 2 & 5 & 14 & 8 & 5 & 0 & 5 \end{bmatrix},$$

$G_{35} = [I_7|A]$ where

$$A = \begin{bmatrix} 3 & 15 & 7 & 14 & 8 & 12 & 1 \\ 5 & 2 & 12 & 5 & 8 & 10 & 11 \\ 6 & 9 & 15 & 12 & 11 & 15 & 12 \\ 5 & 3 & 11 & 5 & 6 & 3 & 1 \\ 8 & 6 & 11 & 2 & 12 & 9 & 9 \\ 2 & 5 & 6 & 4 & 1 & 3 & 14 \\ 11 & 13 & 6 & 4 & 13 & 10 & 3 \end{bmatrix},$$

$G_{36} = [I_7|A]$ where

$$A = \begin{bmatrix} 7 & 9 & 3 & 11 & 4 & 13 & 15 \\ 9 & 2 & 7 & 3 & 6 & 14 & 2 \\ 14 & 7 & 13 & 1 & 0 & 3 & 12 \\ 9 & 1 & 11 & 1 & 2 & 7 & 3 \\ 2 & 10 & 11 & 11 & 12 & 9 & 12 \\ 6 & 15 & 14 & 2 & 7 & 5 & 2 \\ 4 & 12 & 8 & 10 & 15 & 6 & 12 \end{bmatrix},$$

$G_{37} = [I_7|A]$ where

$$A = \begin{bmatrix} 15 & 13 & 12 & 2 & 11 & 3 & 15 \\ 9 & 11 & 15 & 15 & 14 & 14 & 10 \\ 0 & 13 & 15 & 7 & 6 & 1 & 12 \\ 1 & 15 & 10 & 13 & 4 & 15 & 15 \\ 14 & 4 & 14 & 10 & 7 & 1 & 13 \\ 2 & 1 & 4 & 10 & 7 & 12 & 6 \\ 10 & 7 & 14 & 11 & 3 & 4 & 9 \end{bmatrix},$$

$G_{38} = [I_7|A]$ where

$$A = \begin{bmatrix} 9 & 12 & 15 & 2 & 4 & 12 & 1 \\ 1 & 11 & 1 & 7 & 11 & 10 & 2 \\ 10 & 7 & 7 & 7 & 6 & 9 & 7 \\ 3 & 15 & 11 & 13 & 11 & 12 & 9 \\ 11 & 4 & 2 & 2 & 7 & 7 & 3 \\ 14 & 7 & 14 & 2 & 15 & 7 & 11 \\ 6 & 1 & 2 & 14 & 12 & 6 & 13 \end{bmatrix},$$

$G_{39} = [I_7|A]$ where

$$A = \begin{bmatrix} 3 & 7 & 15 & 4 & 10 & 9 & 5 \\ 12 & 8 & 5 & 7 & 11 & 14 & 8 \\ 4 & 1 & 13 & 13 & 14 & 1 & 1 \\ 13 & 15 & 6 & 1 & 6 & 9 & 1 \\ 2 & 4 & 2 & 10 & 15 & 5 & 15 \\ 14 & 12 & 8 & 4 & 7 & 9 & 8 \\ 11 & 7 & 2 & 10 & 13 & 14 & 7 \end{bmatrix},$$

$G_{40} = [I_7|A]$ where

$$A = \begin{bmatrix} 3 & 12 & 7 & 6 & 6 & 12 & 3 \\ 12 & 14 & 15 & 7 & 14 & 11 & 11 \\ 2 & 7 & 1 & 13 & 2 & 12 & 13 \\ 3 & 13 & 10 & 12 & 6 & 12 & 1 \\ 11 & 4 & 4 & 14 & 5 & 15 & 13 \\ 10 & 3 & 4 & 10 & 5 & 15 & 11 \\ 8 & 13 & 10 & 6 & 13 & 2 & 12 \end{bmatrix},$$

$G_{41} = [I_7|A]$ where

$$A = \begin{bmatrix} 7 & 13 & 3 & 14 & 11 & 5 & 12 \\ 5 & 0 & 5 & 3 & 11 & 6 & 4 \\ 6 & 1 & 13 & 15 & 8 & 3 & 15 \\ 9 & 1 & 2 & 15 & 6 & 15 & 7 \\ 10 & 14 & 0 & 4 & 15 & 3 & 5 \\ 14 & 9 & 4 & 8 & 13 & 15 & 8 \\ 2 & 15 & 4 & 6 & 7 & 11 & 7 \end{bmatrix},$$

$G_{42} = [I_7|A]$ where

$$A = \begin{bmatrix} 12 & 5 & 13 & 8 & 10 & 7 & 7 \\ 12 & 14 & 15 & 12 & 2 & 10 & 6 \\ 2 & 1 & 9 & 5 & 6 & 13 & 7 \\ 5 & 13 & 0 & 15 & 6 & 3 & 13 \\ 4 & 0 & 8 & 10 & 13 & 3 & 5 \\ 6 & 15 & 11 & 11 & 12 & 5 & 0 \\ 11 & 13 & 2 & 4 & 12 & 2 & 1 \end{bmatrix},$$

$G_{43} = [I_7|A]$ where

$$A = \begin{bmatrix} 3 & 1 & 12 & 4 & 4 & 7 & 7 \\ 12 & 14 & 1 & 12 & 8 & 11 & 11 \\ 8 & 3 & 7 & 12 & 10 & 13 & 15 \\ 9 & 12 & 10 & 5 & 8 & 5 & 12 \\ 2 & 0 & 4 & 14 & 13 & 7 & 13 \\ 6 & 7 & 11 & 0 & 13 & 15 & 0 \\ 8 & 1 & 6 & 11 & 5 & 6 & 12 \end{bmatrix},$$

$G_{44} = [I_7|A]$ where

$$A = \begin{bmatrix} 7 & 9 & 12 & 10 & 4 & 7 & 9 \\ 12 & 6 & 13 & 13 & 2 & 8 & 14 \\ 10 & 13 & 5 & 15 & 14 & 7 & 3 \\ 13 & 13 & 0 & 3 & 10 & 3 & 1 \\ 6 & 10 & 4 & 14 & 7 & 1 & 12 \\ 2 & 3 & 10 & 4 & 13 & 5 & 11 \\ 8 & 15 & 6 & 4 & 12 & 14 & 3 \end{bmatrix},$$

$G_{45} = [I_7|A]$ where

$$A = \begin{bmatrix} 7 & 1 & 12 & 2 & 11 & 7 & 15 \\ 13 & 4 & 7 & 3 & 14 & 10 & 2 \\ 6 & 12 & 13 & 12 & 8 & 15 & 7 \\ 3 & 9 & 0 & 1 & 0 & 7 & 9 \\ 4 & 6 & 10 & 8 & 12 & 3 & 7 \\ 11 & 15 & 14 & 2 & 9 & 15 & 8 \\ 10 & 9 & 14 & 11 & 13 & 2 & 5 \end{bmatrix},$$

$G_{46} = [I_7|A]$ where

$$A = \begin{bmatrix} 15 & 13 & 1 & 4 & 11 & 15 & 3 \\ 9 & 2 & 5 & 15 & 6 & 11 & 8 \\ 8 & 1 & 7 & 3 & 14 & 5 & 15 \\ 12 & 5 & 10 & 9 & 6 & 3 & 13 \\ 2 & 0 & 14 & 14 & 7 & 12 & 15 \\ 14 & 12 & 11 & 10 & 12 & 9 & 14 \\ 0 & 15 & 6 & 2 & 3 & 2 & 9 \end{bmatrix},$$

$G_{47} = [I_7|A]$ where

$$A = \begin{bmatrix} 7 & 7 & 12 & 0 & 0 & 3 & 13 \\ 5 & 2 & 15 & 1 & 4 & 4 & 0 \\ 10 & 5 & 9 & 12 & 4 & 1 & 5 \\ 3 & 1 & 11 & 5 & 10 & 5 & 13 \\ 11 & 4 & 6 & 2 & 5 & 9 & 1 \\ 0 & 1 & 6 & 11 & 12 & 9 & 10 \\ 10 & 1 & 0 & 4 & 9 & 8 & 1 \end{bmatrix},$$

$G_{48} = [I_7|A]$ where

$$A = \begin{bmatrix} 3 & 5 & 9 & 10 & 8 & 7 & 9 \\ 3 & 4 & 9 & 3 & 2 & 2 & 4 \\ 2 & 3 & 3 & 1 & 6 & 7 & 5 \\ 15 & 3 & 10 & 15 & 14 & 5 & 1 \\ 8 & 11 & 6 & 10 & 13 & 12 & 7 \\ 6 & 5 & 4 & 14 & 15 & 15 & 2 \\ 2 & 7 & 14 & 6 & 1 & 6 & 9 \end{bmatrix},$$

$G_{49} = [I_7|A]$ where

$$A = \begin{bmatrix} 5 & 1 & 7 & 4 & 14 & 9 & 7 \\ 1 & 0 & 1 & 3 & 14 & 4 & 2 \\ 10 & 9 & 13 & 12 & 8 & 12 & 1 \\ 5 & 1 & 8 & 1 & 4 & 1 & 3 \\ 6 & 2 & 10 & 11 & 9 & 5 & 3 \\ 8 & 5 & 6 & 11 & 12 & 15 & 2 \\ 10 & 9 & 4 & 2 & 15 & 2 & 13 \end{bmatrix},$$

$G_{50} = [I_7|A]$ where

$$A = \begin{bmatrix} 15 & 1 & 5 & 2 & 0 & 13 & 9 \\ 12 & 4 & 3 & 12 & 6 & 14 & 2 \\ 4 & 9 & 15 & 3 & 2 & 9 & 9 \\ 1 & 1 & 6 & 7 & 10 & 3 & 5 \\ 2 & 4 & 4 & 4 & 12 & 3 & 13 \\ 0 & 13 & 2 & 6 & 7 & 5 & 6 \\ 14 & 15 & 10 & 4 & 1 & 10 & 1 \end{bmatrix},$$

$G_{51} = [I_7|A]$ where

$$A = \begin{bmatrix} 3 & 3 & 12 & 14 & 0 & 13 & 12 \\ 5 & 6 & 7 & 1 & 6 & 0 & 10 \\ 6 & 5 & 9 & 5 & 6 & 9 & 12 \\ 7 & 13 & 11 & 12 & 11 & 9 & 15 \\ 0 & 6 & 14 & 10 & 7 & 9 & 7 \\ 6 & 9 & 4 & 10 & 5 & 5 & 8 \\ 11 & 12 & 10 & 4 & 3 & 11 & 9 \end{bmatrix},$$

$G_{52} = [I_7|A]$ where

$$A = \begin{bmatrix} 5 & 3 & 5 & 11 & 4 & 1 & 13 \\ 12 & 8 & 12 & 12 & 4 & 14 & 10 \\ 6 & 12 & 3 & 9 & 8 & 12 & 12 \\ 15 & 1 & 6 & 5 & 2 & 15 & 9 \\ 11 & 2 & 0 & 4 & 5 & 9 & 7 \\ 0 & 9 & 4 & 14 & 13 & 7 & 4 \\ 14 & 9 & 11 & 11 & 1 & 0 & 9 \end{bmatrix},$$

$G_{53} = [I_7|A]$ where

$$A = \begin{bmatrix} 13 & 7 & 7 & 14 & 6 & 9 & 3 \\ 9 & 8 & 1 & 15 & 6 & 4 & 0 \\ 6 & 7 & 13 & 9 & 0 & 1 & 9 \\ 5 & 3 & 4 & 3 & 11 & 12 & 15 \\ 6 & 14 & 6 & 2 & 12 & 9 & 3 \\ 6 & 13 & 4 & 0 & 12 & 13 & 10 \\ 4 & 7 & 10 & 0 & 5 & 14 & 13 \end{bmatrix},$$

$G_{54} = [I_7|A]$ where

$$A = \begin{bmatrix} 12 & 9 & 13 & 2 & 14 & 12 & 3 \\ 15 & 4 & 3 & 9 & 10 & 2 & 14 \\ 14 & 7 & 5 & 1 & 11 & 7 & 5 \\ 1 & 13 & 11 & 3 & 11 & 1 & 1 \\ 4 & 11 & 6 & 11 & 13 & 5 & 13 \\ 0 & 13 & 4 & 0 & 1 & 12 & 6 \\ 4 & 12 & 2 & 14 & 13 & 11 & 5 \end{bmatrix},$$

$G_{55} = [I_7|A]$ where

$$A = \begin{bmatrix} 3 & 12 & 13 & 8 & 11 & 13 & 5 \\ 15 & 10 & 12 & 7 & 6 & 6 & 6 \\ 11 & 15 & 12 & 1 & 0 & 15 & 1 \\ 12 & 9 & 14 & 7 & 11 & 1 & 13 \\ 2 & 6 & 11 & 0 & 5 & 7 & 3 \\ 11 & 9 & 0 & 6 & 3 & 3 & 4 \\ 8 & 12 & 6 & 4 & 7 & 2 & 13 \end{bmatrix},$$

$G_{56} = [I_7|A]$ where

$$A = \begin{bmatrix} 9 & 7 & 7 & 11 & 11 & 7 & 1 \\ 9 & 11 & 5 & 3 & 2 & 4 & 14 \\ 2 & 13 & 9 & 12 & 11 & 3 & 7 \\ 13 & 1 & 8 & 9 & 11 & 5 & 13 \\ 6 & 2 & 2 & 11 & 5 & 5 & 1 \\ 6 & 12 & 4 & 8 & 7 & 15 & 2 \\ 2 & 1 & 11 & 6 & 3 & 4 & 1 \end{bmatrix},$$

$G_{57} = [I_7|A]$ where

$$A = \begin{bmatrix} 7 & 3 & 3 & 14 & 11 & 15 & 12 \\ 7 & 10 & 15 & 5 & 14 & 10 & 2 \\ 2 & 1 & 15 & 5 & 0 & 1 & 7 \\ 9 & 9 & 6 & 15 & 11 & 15 & 7 \\ 2 & 2 & 0 & 11 & 15 & 15 & 7 \\ 11 & 7 & 6 & 11 & 15 & 12 & 14 \\ 8 & 12 & 4 & 10 & 9 & 4 & 7 \end{bmatrix},$$

$G_{58} = [I_7|A]$ where

$$A = \begin{bmatrix} 7 & 5 & 5 & 11 & 11 & 3 & 5 \\ 5 & 10 & 5 & 3 & 14 & 2 & 14 \\ 4 & 3 & 13 & 1 & 6 & 7 & 15 \\ 5 & 15 & 6 & 3 & 10 & 5 & 1 \\ 6 & 8 & 6 & 2 & 7 & 12 & 1 \\ 11 & 13 & 6 & 11 & 9 & 1 & 11 \\ 2 & 5 & 11 & 11 & 1 & 4 & 12 \end{bmatrix},$$

$G_{59} = [I_7|A]$ where

$$A = \begin{bmatrix} 9 & 15 & 13 & 10 & 0 & 9 & 13 \\ 3 & 8 & 12 & 12 & 4 & 6 & 0 \\ 14 & 9 & 12 & 12 & 10 & 15 & 15 \\ 12 & 13 & 10 & 7 & 14 & 9 & 5 \\ 4 & 6 & 4 & 2 & 7 & 13 & 7 \\ 6 & 5 & 8 & 4 & 1 & 7 & 2 \\ 8 & 5 & 10 & 14 & 3 & 0 & 13 \end{bmatrix},$$

$G_{60} = [I_7|A]$ where

$$A = \begin{bmatrix} 12 & 9 & 12 & 2 & 6 & 1 & 1 \\ 5 & 0 & 1 & 3 & 10 & 10 & 11 \\ 2 & 15 & 9 & 15 & 11 & 5 & 1 \\ 7 & 15 & 6 & 1 & 2 & 15 & 9 \\ 14 & 4 & 2 & 0 & 3 & 15 & 1 \\ 4 & 3 & 0 & 11 & 1 & 15 & 6 \\ 11 & 15 & 2 & 8 & 9 & 10 & 13 \end{bmatrix},$$

and $G_{61} = [I_7|A]$ where

$$A = \begin{bmatrix} 11 & 13 & 13 & 0 & 11 & 4 & 1 \\ 2 & 7 & 2 & 3 & 13 & 13 & 15 \\ 15 & 12 & 6 & 3 & 13 & 15 & 6 \\ 8 & 12 & 10 & 2 & 10 & 15 & 9 \\ 0 & 9 & 5 & 7 & 12 & 9 & 11 \\ 14 & 6 & 8 & 5 & 11 & 13 & 3 \\ 14 & 2 & 1 & 3 & 6 & 6 & 15 \end{bmatrix}.$$

5.2 CONCLUSION

In this thesis, we aimed to find an improved extension method to obtain self-dual codes starting with known binary self-dual codes.

In Chapter 2, basic notions about codes over rings are defined. Some properties of self-dual codes are given. Extended binary Hamming code, extended binary Golay code and extended Quadratic Residue code are given as examples. After these, some extension methods of self-dual codes are considered. Also, current status of classification of binary self-dual codes up to certain lengths is given.

In Chapter 3, the ring R_k has been defined and R_1 and R_2 are explained in detail. Their units, ideal structure and Gray maps are identified. The Lee weight of a codeword is defined to be the Hamming weight of the image of the codeword under the Gray map which is a distance preserving map.

In Chapter 4, the notion of projection and lift have been defined. The new method which enables us to construct binary self-dual codes of length $2^k.n$ from binary self-dual codes of length n is described.

As a result of the new method, 10 extremal binary self-dual codes with new weight enumerators have been obtained.

Our limited computational power did not let us to investigate lifts of bigger

size codes. But we believe that the technique might lead to more interesting results.

Also, what we did here can be applied to different kind of rings, for example $\mathbb{Z}_4$. But in $\mathbb{Z}_4$ case initial binary self-dual codes to be lifted must be doubly-even(Type II) since projections of self-dual $\mathbb{Z}_4$ codes are doubly-even.

REFERENCES

Aguilar-Melchor, C., Gaborit, P., Kim J.L., Sok, L., and Solé, P., "Classification of extremal and s-extremal binary self-dual codes of length 38", *IEEE Trans. Inform. Theory*, Vol. 58, No. 4, pp. 2253–2262, 2012.

Bachoc, C., "Applications of coding theory to the construction of modular lattices", *Journal Combin. Theory Ser. A*, Vol. 78, pp. 92–119, 1997.

Betsumiya, K., Harada, M., and Munemasa, A., "A complete classification of doubly even self-dual codes of length 40", *Electron. J. Combin.*, Vol. 19, No. 3, p. 18, 2012.

Bilous, R.T., "Enumeration of the binary self-dual codes of length 34", *J. Combin. Math. Combin. Comput.*, Vol. 59, pp. 173–211, 2006.

Bilous, R.T. and van Rees, G.H.J., "An enumeration of self-dual codes of length 32", *Des. Codes Cryptogr.*, Vol. 26, pp. 61–86, 2002.

Bosma, W., Cannon, J. and Playoust, C., "The Magma algebra system. I. The user language", *J. Symbolic Comput.*, Vol.24, pp. 235–265, 1997.

Boukliev, I. and Buyuklieva, S., "Some new extremal self-dual codes with lengths 44,50,54, and 58", *IEEE Trans. Inform. Theory*, Vol. 44, No. 2, pp. 809-812, 1998.

Bouyuklieva, S. and Bouyukliev, I., "An algorithm for classification of binary self-dual codes", *IEEE Trans. Inform. Theory*, Vol. 58, No. 6, pp. 3933–3940, 2012.

Bouyuklieva, S, and Bouyukliev, I., "Extremal self-dual codes with an automorphism of order 2", *IEEE Trans. Inform. Theory*, Vol. 44, No. 1, pp. 323–328, 1998.

Bouyuklieva, S., Russeva, R., and Yankov, N., "On the structure of binary self-dual codes having an automorphism of order a square of an odd prime", *IEEE Trans. Inform. Theory*, Vol. 51, No. 10, pp. 3678–3686, 2005.

Conway, J.H. and Sloane, N.J.A., "A new upper bound on the minimal distance of self-dual codes", *IEEE Trans. Inform. Theory*, Vol. 36, No. 6, pp. 1319–1333, 1990.

Dougherty, S.T., Gaborit, P., Harada, M., and Solé, P., "Type II codes over $\mathbb{F}_2 + u\mathbb{F}_2$", *IEEE Trans. Inform. Theory*, Vol. 45, No. 1, pp. 32–45, 1999.

Dougherty, S.T., Kim, J.L., and Kulosman, H., "MDS codes over finite principle ideal rings", *Des. Codes Cryptogr.*, Vol. 50, No. 1, pp. 77–92, 2009.

Dougherty, S.T., Yildiz, B., and Karadeniz, S., "Codes over R_k, Gray maps and their binary images", *Finite Fields Appl.*, Vol. 17, No. 3, pp. 205–219, 2011.

Dougherty, S.T., Yıldız, B., and Karadeniz, S., "Self-dual codes over R_k and binary self-dual codes", *Eur. J. Pure Appl. Math.*, Vol. 6, No. 1, pp. 89–106, 2013.

Hammons, A.R., Kumar, V., Calderbank A.R., Sloane, N.J., and Solé, P., "The Z_4 linearity of Kerdock, Preparata, Goethals and related codes", *IEEE Trans. Inform. Theory*, Vol. 40, pp. 301–319, 1994.

Harada, M., "The existence of a self-dual $[70, 35, 12]$ code and formally self-dual codes", *Finite Fields Appl.*, Vol. 3, pp. 131–139, 1997.

Harada, M. and Kimura, H., "On extremal self-dual codes", *Math. J. Okayama Univ.*, Vol. 37, pp. 1–14, 1995.

Harada, M. and Munemasa, A., "Classification of self-dual codes of length 36", *Advances Math. Commun.*, Vol. 6, pp. 229–235, 2012.

Horimoto, H., and Shiromoto, K., "On generalized Hamming weights for codes over finite chain rings", *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes Lecture Notes in Computer Science*, Vol. 2227, pp. 141–150, 2001.

Huffman, W.C. and Pless, V. (editors), " Fundamentals of Error-Correcting Codes", *Cambridge University Press* , New York, 2003.

Karadeniz, S., "Linear codes over the ring $\mathbb{F}_2 + u\mathbb{F}_2 + v\mathbb{F}_2 + uv\mathbb{F}_2$", *Ph.D Thesis, Fatih University* , 2011.

Karadeniz, S. and Aksoy, R., "$www.fatih.edu.tr/ \sim aserdogan/61matrices.txt$".

Karadeniz, S. and Kaya, A., "New extremal binary self-dual codes of length 58 as R_3-lifts of the shortened binary $[8, 4, 4]$ Hamming code", *J. Franklin Inst.*, Vol. 349, No. 9, pp. 2824–2833, 2012.

Karadeniz, S. and Yıldız, B., "New extremal binary self-dual codes of length 64 from R_3-lifts of the extended binary Hamming code", *Des. Codes Cryptogr.*, doi:10.1007/s10623-013-9884-6, 2013.

Karadeniz, S. and Yıldız, B., "New extremal binary self-dual codes of length 68 from R_2-lifts of binary self-dual codes", *Advances Math. Commun.*, Vol. 7, No. 2, pp. 219–229, 2013.

Kim, J.L., "New extremal self-dual codes of lengths 36,38, and 58", *IEEE Trans. Inform. Theory*, Vol. 47, No. 1, pp. 386–393, 2001.

Kim H.J., Lee, H., and Lee, J.B., "Construction of self-dual codes with an automorphism of order p", *Advances Math. Commun.*, Vol. 5, No. 1, pp. 23–26, 2011.

Ling, S. and Xing, C., "Coding Theory: A First Course", *Cambridge University Press*, New York, 2004.

MacWilliams, F.J. and Sloane, N.J.A., "The Theory of Error-Correcting Codes", *Amsterdam*, North-Holland, 1977.

Melchor, C.A. and Gaborit, P., "On the classification of extremal $[36, 18, 8]$ binary self-dual codes", *IEEE Trans. Inform. Theory*, Vol. 54, No. 10, pp. 4743–4750, 2008.

Rains, E.M., "Shadow bounds for self-dual codes", *IEEE Trans. Inform. Theory*, Vol. 44, No. 1, pp. 134–139, 1998.

Rains, E.M. and Sloane, N.J.A., "Self-Dual Codes, The Handbook of Coding Theory, (eds.) Pless V. and Huffman W.C.", *North-Holland*, New York, 1998.

Tsai, H.P. and Jiang, Y. J., "Some new extremal self-dual $[58, 29, 10]$ codes", *IEEE Trans. Inform. Theory*, Vol. 44, pp. 813–814, 1998.

Wood, J. A., "Duality for modules over finite rings and applications to coding theory", *Amer. J. Math.*, Vol. 121, pp. 555–575, 1999.

Yankov, N. and Lee, M.H., "New binary self-dual codes of lengths $50 - 60$", *Des. Codes Cryptogr.*, doi 10.1007/s10623-013-9839-y, 2013.

Yankov, N. and Russeva, R., "Binary self-dual codes of lengths 52 to 60 with an automorphism of order 7 or 13", *IEEE Trans. Inform. Theory*, Vol. 57, No. 11, pp. 7498–7506, 2011.

Yildiz, B. and Karadeniz, S., "Linear codes over $\mathbb{F}_2 + u\mathbb{F}_2 + v\mathbb{F}_2 + uv\mathbb{F}_2$", *Des. Codes Cryptogr.*, Vol. 54, No. 1, pp. 61–81, 2010.

Yıldız, B. and Karadeniz, S., "Self-dual codes over $\mathbb{F}_2 + u\mathbb{F}_2 + v\mathbb{F}_2 + uv\mathbb{F}_2$", *J. Franklin Inst.*, Vol. 347, No. 10, pp. 1888–1894, 2010.