INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

Bell & Howell Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA

UMI®
800-521-0600
GENERALIZED $r$-FOLD WEIGHT ENUMERATORS FOR LINEAR CODES AND NEW LINEAR CODES WITH IMPROVED MINIMUM DISTANCES

DISSERTATION

Presented in Partial Fulfillment of the Requirements for the Degree Doctor of Philosophy in the Graduate School of the Ohio State University

By

Irfan Siap, M.S.

The Ohio State University
1999

Dissertation Committee:
Professor Dijen K. Ray-Chaudhuri, Adviser
Professor Thomas Dowling
Professor Joseph Ferrar

Approved by

Adviser
Department Of Mathematics
ABSTRACT

MacWilliams-type identities play an important role in coding theory. There has been several generalizations of the Hamming weight enumerator of a code and the identity proved by F. J. MacWilliams in [37]. We define a generalized r-fold weight enumerator of r linear codes over rings, and we obtain a MacWilliams-type identity for the generalized r-fold weight enumerators of these codes and their duals. This identity generalizes the previously known MacWilliams-type identities. It gives a unified treatment and combines most well-known MacWilliams-type identities as special cases. By specializing the variables of the generalized r-fold weight enumerators we are able to obtain other weight enumerators, such as the Hamming, Lee, complete, r-genus, r-ply and symmetric weight enumerators. Also, we define generalized r-fold symmetric and Hamming weight enumerators of r codes, and MacWilliams-type identities are obtained by specializing the variables in the generalized r-fold complete weight enumerators of these codes. Further, we consider codes (submodules) over rings $\mathbb{F}_3 + u\mathbb{F}_3$, $\mathbb{F}_5 + u\mathbb{F}_5$ and $\mathbb{F}_5 + u\mathbb{F}_5 + u^2\mathbb{F}_5$ with $u^2 = 1$ where $\mathbb{F}_5$ is a finite field with 5 elements. Using a Gray map we relate the codes over these rings with their images which are codes over $\mathbb{F}_3$ and $\mathbb{F}_5$. These "good" submodules give 19 new linear codes over $\mathbb{F}_5$ and one new linear code over $\mathbb{F}_3$ with improved minimum distances. By
applying puncturing and shortening to these new codes, one can get further improvements on lower bounds for minimum distances of many other codes. Finally, we apply invariant theory to investigate the ring of invariants of 2-ply weight enumerators of binary self-dual codes with lengths divisible by 4 or 8. We show that the 2-ply weight enumerators of binary self-dual codes with lengths divisible by 8 are related to the 2-ply weight enumerators of extended Hamming and Golay codes. We conclude the dissertation by establishing the ring of invariants of complete weight enumerators of codes over \( \mathbb{F}_2 + u\mathbb{F}_2 \) with \( u^2 = 1 \) whose images are Type I and II codes.
Dedicated to my father and my family
ACKNOWLEDGMENTS

I would like to express my gratitude to my adviser, Professor Dijen K. Ray-Chaudhuri who introduced me to coding theory and design theory. I am especially grateful to him for his intellectually stimulating discussions, suggestions, insights, support and patience.

I would like to thank the other members of my dissertation committee, Professor Thomas Dowling and Professor Joseph Ferrar. Especially, I would like to thank both Professor Joseph Ferrar and Professor Thomas Dowling for their reading and constant support in my graduate studies in The Ohio State University.

I would like to thank all my teachers who helped me through my education.

Also I would like to thank my friends in Columbus for their support. Especially, Nuh Aydin for his discussions, Bora Karayaka and Metin Demirci for their help in computer programming.

I could never have completed this project without the love and encouragement of my family.
VITA

September 6, 1970 ............................... Born in Tetova, Macedonia.

1992 ....................................................... B.S. in Mathematics, Istanbul University, Turkey.

1996 ....................................................... M.S. in Mathematics, The Ohio State University, Ohio.

PUBLICATIONS


FIELDS OF STUDY

Major field: Mathematics

Specialization: Combinatorics

Studies in
Algebraic Coding Theory
Design Theory
**TABLE OF CONTENTS**

Abstract ............................................................................................................................... ii
Dedication ............................................................................................................................ iii
Acknowledgments .............................................................................................................. v
Vita ...................................................................................................................................... vi
List of Figures .................................................................................................................. x

<table>
<thead>
<tr>
<th>CHAPTER</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1.1</td>
<td>4</td>
</tr>
<tr>
<td>1.2</td>
<td>8</td>
</tr>
<tr>
<td>1.3</td>
<td>10</td>
</tr>
<tr>
<td>1.3.1</td>
<td>14</td>
</tr>
<tr>
<td>1.3.2</td>
<td>17</td>
</tr>
<tr>
<td>1.4</td>
<td>22</td>
</tr>
<tr>
<td>2</td>
<td>25</td>
</tr>
<tr>
<td>2.1</td>
<td>29</td>
</tr>
<tr>
<td>2.2</td>
<td>37</td>
</tr>
<tr>
<td>2.3</td>
<td>50</td>
</tr>
<tr>
<td>2.4</td>
<td>55</td>
</tr>
<tr>
<td>2.5</td>
<td>70</td>
</tr>
<tr>
<td>2.6</td>
<td>84</td>
</tr>
<tr>
<td>2.6.1</td>
<td>84</td>
</tr>
<tr>
<td>2.6.2</td>
<td>85</td>
</tr>
</tbody>
</table>
3 NEW LINEAR CODES WITH IMPROVED BOUNDS ON MINIMUM DISTANCES ........................................ 91

3.1 Quasi Cyclic Codes .......................................................... 92
3.1.1 1-generator QC codes ...................................................... 100
3.2 Codes over rings of type $R + uR$ ........................................ 105
3.3 New Linear Codes Over $\mathbb{F}_3 + u\mathbb{F}_3$ and $\mathbb{F}_5 + u\mathbb{F}_5$ ................................ 113
3.3.1 New codes and their generator matrices ....................... 115
3.3.2 Gray weight enumerators of new codes ....................... 119
3.4 New Linear Codes Over $\mathbb{F}_5$ Obtained by Tripling Method 123
3.4.1 Codes over rings of type $R + uR + \cdots + u^{m-1}R$ .......... 123
3.4.2 New Linear Codes over $\mathbb{F}_5 + u\mathbb{F}_5 + u^2\mathbb{F}_5$ .......... 129

4 APPLICATIONS OF INVARIANT THEORY ............................... 134

4.1 Ring of invariants of 2-ply weight enumerators of binary self-dual linear codes ................ 134
4.2 Classification of complete weight enumerator of self-dual codes over $\mathbb{F}_2 + u\mathbb{F}_2$ .......... 143
4.2.1 Complete weight enumerators of codes over $\mathbb{F}_2 + u\mathbb{F}_2$ ......................... 143
4.2.2 Ring of invariants of codes over $\mathbb{F}_2 + u\mathbb{F}_2$ ................................. 147

Bibliography ....................................................................................... 154

ix
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>FIGURE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1 Transmission process of a message.</td>
<td>3</td>
</tr>
</tbody>
</table>
CHAPTER 1

INTRODUCTION

Coding Theory uses mathematical constructs to improve reliability of communication through a noisy channel by employing redundancy. The redundant bits are chosen as complex functions of the information bits. For a fixed number of redundant bits, one tries to optimize the error-correction ability (maximizing the information bits) of the code and give an algebraic construction of the code which leads to efficient encoding and decoding. Error-correcting codes have found applications in many real-life situations, in space-communication (for instance the Mariner Program), in noiseless reproduction of music on compact disks, error-free computer memory, computer networks, and cryptography.

Suppose that we wish to transmit a message and know that in the process of transmission there would be some altering of the message, due to weak signals, sporadic electrical bursts and other naturally occurring noise that affects the transmission process. The problem is to insure that the intended message is obtainable from whatever is actually received. One simple approach to this problem is what is called a repetition code. For instance, if we want to send the message NICE JOB. we can repeat each letter five times and send, say,

NNNNNNIIIIICCCCEEEEJJJJOOOOBBBBB.
A number of these letters might get changed in transmission. The intended message could be recovered from a received message such as

NNFNIJJNECCCFERREEEEIIJIIJNNOOBOBNBB.

by a process called majority decoding. In this case, for each block of 5 letters the intended letter is the one which appears most frequently in the block. Suppose, in our example, that the probability that a letter is altered in transmission process is $p = 0.1$ and so $q = 1 - p = 0.9$ is the probability that a letter is correctly received. Without any coding, the probability of our 7 letter message being correctly received is $q^7 = 0.48$. Using this repetition code, the probability of correctly decoding a given letter is $q^5 + 5q^4p + 10q^3p^2 = 0.99$ and so the probability of getting the correct message after decoding is $0.99^7 = 0.94$, clearly a great increase over the non-coded message. This code gives significant improvement in the probability of correct transmission of a message. But this improvement does have a cost. Unfortunately, as easy as the repetition code looks, it is not very efficient. The increase in the length of the transmitted code, and thus the increase in the time and energy required to transmit it, is substantial. Indeed the rate of transmission is only 1/5.

To increase the probability of decoding the correct message with this type of code we would have to increase the number of repetitions, which may not be desirable or even possible in certain situations. However, by using other coding schemes, it is possible to get a higher probability of correct transmission. So, one of the main problems in coding theory is to construct encoding schemes (or codes) that are more efficient. In digital communication, the messages are sent mostly as strings of 0's and
1's. The above example was given in letters in order to make the concept clearer. The finite set of symbols used for sending messages is called an alphabet, say $F$, the most important case being $F = \{0, 1\}$. The fields with $q$ elements will be denoted by $F_q$. A code $C$ of length $n$ over the alphabet $F$ is a subset of $F^n$ (the set of $n$-tuples with entries from $F$). The elements of $C$ are called codewords.

Let us examine the components involved in a communication process. Assume that we have a message $(1, 0, 1, 1) \in F_2^4$ and we are going to send this message to a receiver. Before we send it we encode the original message to $(1, 0, 1, 1, 0.1, .1) \in F_2^6$ in order to introduce some protection while transmitting the original message. Consider the following Figure 1.1:

![Figure 1.1: Transmission process of a message.](image)

Figure 1.1: Transmission process of a message.
The message from the source is sent to the encoder. The original message consists of 4 bits, called information bits. The encoder introduces three more bits, called check bits. Basically, this is a function that assigns a codeword to each source message. The codeword is then sent over a channel, where some bits may be changed, and errors may occur. The decoder accepts the output from the channel, and checks whether there is an error or not. If there is an error, then the decoder attempts to correct it. The decoder attempts to recover the original message and finally sends it to the receiver.

The rate of transmission is the ratio of the length of the message (source) to the length of the code (encoded message). In our example above, the rate of transmission is 4/7.

In 1948, Claude Shannon [47] first proved "The noisy coding theorem" which basically states that there exist "good" codes for large lengths. Unfortunately, neither his proof nor the others given later, are constructive. No one has found a way to construct the encoding schemes promised by Shannon's theorem. In order to find encoding schemes that are relatively easy to implement, i.e. encode, decode, etc., coding theorists have been led to search for codes that have algebraic or geometric structure.

1.1 Basics

We follow the definition of a ring $R$, given by Lang in [35]. A ring $R(+,\cdot)$ is a set, together with two law of composition, called multiplication (denoted by $\cdot$) and
addition (denoted by +), satisfying the following conditions:

1) \((R, +)\) is a commutative group,

2) multiplication is associative and \(R\) has a unit element,

3) right and left distributivity holds in \(R\). i.e. for all \(a, b, c \in R\) we have.

\[(a + b)c = ac + bc \quad \text{and} \quad a(b + c) = ab + ac.\]

We give two examples of noncommutative rings:

1) We can take \(n \times n\) matrices over any commutative ring. For instance, \(M_2(\mathbb{F}_2)\), 2 by 2 matrices over the binary field. Codes over these type of rings have been considered by Bachoc in [1].

2) Quaternions over rings. For instance, \(R = Z_4 + iZ_4 + jZ_4 + kZ_4\) where \(Z_4\) is the ring of integers modulo 4, and \(i^2 = j^2 = k^2 = -1, ij = k\).

Most work on codes has been done in codes over fields. Lately, codes over commutative rings have been considered. The most important work on codes over rings \((Z_4, \text{integers modulo } 4)\) was initiated by Hammons, Kumar, et al. in [27]. In this paper, they show that some very important binary nonlinear codes, such as Kerdock, Goethals can be viewed as linear codes (submodules) over \(Z_4\). This paper led to a better understanding of the structure of some important nonlinear binary codes. After it appeared, work on codes over rings intensified. Later, Bachoc considered codes over \(F_p + uF_p\) where \(u^2 = 0\) and \(p\) is prime. Recently, type II codes over \(\mathbb{F}_2 + u\mathbb{F}_2\) are considered by Dougherty et al. in [12]. Gulliver and Harada in [24] have considered codes over \(\mathbb{F}_3 + u\mathbb{F}_3\) with \(u^2 = 0\), including new linear ternary codes with improved minimum distances. We will explain this approach in detail in chapter 2.
Let \( R = \{ \beta_0, \beta_1, \ldots, \beta_{m-1} \} \) be a finite ring (not necessarily commutative) of size \( m > 1 \) with an identity. Let \( C \) be a left submodule of \( R^n \) and \( u = (u_1, \ldots, u_n) \), \( v = (v_1, \ldots, v_n) \in C \). The function

\[
d : R^n \times R^n \to \mathbb{N}_0
\]
called **Hamming distance** is defined by

\[
d(u, v) = |\{ i : u_i \neq v_i \}|
\]

where \( \mathbb{N}_0 = \mathbb{N} \cup \{0\} \) and \( \mathbb{N} \) is the set of positive integers. The Hamming distance is a metric in \( R^n \).

The **minimum Hamming distance** of a code \( C \) is defined by

\[
d(C) = \min_{u, v \in C, u \neq v} d(u, v).
\]

\( C \) is said to be an \((n, M)\)-linear code if and only if \( C \) is a submodule (we will always assume left submodule and say simply submodule in the noncommutative case) of \( R^n \) of size \( M \), and \( d(C) = d \). The parameter \( n \) is called the **length** of the code \( C \).

A **decoding** rule is a mapping \( D : R^n \to C \). If \( v \in C \) is received, then \( D(v) \) is interpreted as the transmitted message. **Minimum distance decoding** is defined by \( D(v) = u \) where \( u \) is an element such that for all \( u' \in C \), \( d(v, u) \leq d(v, u') \). A code \( C \) is said to be an **e-error correcting code**, \( e \) a positive integer, if there is a decoding rule for \( C \) such that \( D(v) = u \) whenever \( d(u, v) \leq e \) for all \( u \in C \) and

6
It is easy to prove that if \( d(C) = d \), then the minimum distance decoding rule corrects \( \left\lfloor \frac{d-1}{2} \right\rfloor \) error(s).

**Definition 1.1.1** An \([n, k, d]_q\) linear code is a vector subspace \( C \) of \( F_q^n \) of dimension \( k \) with \( d(C) = d \).

Let \( C \) be an \([n, k, d]_q\) linear code. Since \( C \) is a vector subspace of \( F_q^n \), every basis has \( k \) elements. A matrix \( G \) whose rows consist of a basis of \( C \) is called a **generator matrix** for \( C \). A generator matrix of the form \( G = (I_k | A) \) where \( I_k \) denotes the \( k \times k \) identity matrix, is said to be in standard form.

In this introductory chapter we describe some basic concepts and well known lemmas and theorems. In some instances, for the convenience of the reader, some short proofs are included.

The left inner product (i.e. inner product in the field case) of two codewords \( u \) and \( v \) is defined as follows

\[
\langle u, v \rangle := \sum_{i=1}^{n} u_i v_i. \tag{1.1}
\]

Note that \( \langle u + u', v \rangle = \langle u, v \rangle + \langle u', v \rangle \), and \( r \langle u, v \rangle = \langle u, rv \rangle \) for all \( u, u', v \in V \) and \( r \in R \). The **dual code** \( C^\perp \) of an \((n, M)\) linear code \( C \) is defined by

\[
C^\perp := \{ v \in V \mid \langle u, v \rangle = 0 \text{ for all } u \in C \}.
\]

Observe that \( C^\perp \) is also a submodule of \( R^n \).

Let \( C \) be an \([n, k, d]_q\) code and let \( G = (I_k, A) \) be a generator matrix of \( C \). The dual
of $C$ has dimension $n-k$, since $\dim(C) + \dim(C^\perp) = n$ over fields. If $H = (-A^T|I_{n-k})$, where superscript $T$ stands for the transpose, then

$$GHT = (I_k|A) \begin{pmatrix} -A \\ I_{n-k} \end{pmatrix} = -A + A = 0.$$  

Thus, the rows of $H$ are orthogonal to the rows of $G$, and since $\text{rank}(H) = n - k$, $H$ is a generator matrix for $C^\perp$. The matrix $H$ is called a parity check matrix for $C$. We see that a linear code $C$ can be uniquely determined by either a generator matrix or a parity check matrix.

### 1.2 Some Bounds on Codes

A good $(n,M,d)$-code should have a relatively large size so that it can be used to encode a large number of source messages and a relatively large minimum distance, so that it can be used to correct a large number of errors. However, these two goals conflict with each other. Finding an optimal solution to this problem is one of the main problems of coding theory.

**Definition 1.2.1** Let $d_q(n,k)$ denote the largest possible minimum distance for a linear code of fixed length $n$ and dimension $k$ over a finite field of order $q$. Also, $A_q(n,d)$ will denote the largest possible size $M$ of a $q$-ary code (a code over $F_q$) with minimum distance $d$.

One of the important problems of coding theory is to find a linear code over a finite field $F_q$ that has the largest possible minimum distance for fixed length $n$ and dimension $k$ and construct the corresponding codes algebraically.
The most current information for the values of $d_q(n,k)$ is available in the web page [4] maintained by A. E. Brouwer for $q = 2, 3, 5, 7, 9$. Also, a table of best known codes recently has been published in the Handbook of Coding Theory [43].

We will state some important theorems which give upper bounds for minimum Hamming distances of codes. Proofs can be found in the book "Information and Coding Theory" by Roman [45].

**Theorem 1.2.1 (The Gilbert Varshamov Bound)**

$$A_q(n,d) \geq \frac{q^n}{\sum_{j=0}^{d-1} \binom{n}{j}(q - 1)^j}$$

Further, there exists a linear $[n,k]$-code over $F_q$ with minimum distance at least $d$ provided that

$$q^k \leq \frac{q^n}{\sum_{i=0}^{d-2} \binom{n-1}{i}(q - 1)^i}.$$

Hence, if $k$ is the largest integer for which this inequality holds, then $A_q(n,d) \geq q^k$.

**Theorem 1.2.2 (The Singleton Bound)**

$$A_q(n,d) \leq q^{n-d+1}.$$

**Theorem 1.2.3 (The Sphere Packing Bound or the Hamming Bound)**

$$A_q(n,d) \leq \frac{q^n}{\sum_{j=0}^{t} \binom{n}{j}(q - 1)^j}, \quad t = \lfloor \frac{d - 1}{2} \rfloor.$$
Codes for which the equality holds in the sphere packing bound are called **perfect** codes.

**Theorem 1.2.4 (Plotkin Bound)** Let $\theta = q/(q - 1)$. If $d \geq \theta n$, then

$$A_q(n, d) \leq \frac{d}{d - \theta n}.$$ 

Note that the Plotkin bound is useful only when the minimum distance $d$ is rather large. This bound has been refined further for binary codes.

**Theorem 1.2.5 (The Elias Bound)** Let $\theta = (q - 1)/q$. If $r$ is a positive integer satisfying $r \leq \theta n$ and $r^2 - 2\theta nr + \theta nd > 0$, then

$$A_q(n, d) \leq \frac{\theta ndq^n}{(r^2 - 2\theta nr + \theta nd)V_q(n, r)}.$$ 

where

$$V_q(n, r) = \sum_{j=0}^{r} \binom{n}{j}(q - 1)^j.$$ 

1.3 **Cyclic Codes**

Let $c = (c(0), c(1), \ldots, c(n - 1))$ be a codeword in $C \subseteq F_q^n$. We may associate to each codeword $c$ a polynomial in $F_q[x]$ as follows:

$$\Omega : c = (c(0), c(1), \ldots, c(n - 1)) \rightarrow c(x) = c_0 + c_1x + \cdots + c_{n-1}x^{n-1}. \quad (1.2)$$

The map $\Omega$ is a vector space isomorphism from $C$ onto the subspace $\Omega(C)$ of $F_q[x]$. For the sake of brevity, $\Omega(C)$ is also written as $C$. 

10
**Definition 1.3.1** A linear code is cyclic if and only if

\[(c(0), c(1), \ldots, c(n - 1)) \in C \text{ implies } (c(n - 1), c(0), c(1), \ldots, c(n - 2)) \in C.\]

Viewing a codeword \(c\) as a polynomial \(c(x)\) in a cyclic code \(C\) implies that \(xc(x) \in C\) modulo \((x^n - 1)\). Thus, a code \(C\) is cyclic if \(C\) is an ideal of

\[R_n = \frac{F_q[x]}{(x^n - 1)}.\]

This relation enables us to use the algebraic structure of ideals in order to understand cyclic codes.

**Theorem 1.3.1** Let \(F_q\) be a finite field with \(q\) elements, and let \(f(x) \in F_q[x]\).

1) \(F_q[x]\) is a principal ideal domain.

2) \(F_q[x]/(f(x))\) is a principal ideal ring.

The following theorem gives important information regarding the structure of cyclic codes.

**Theorem 1.3.2** [45] Let \(C\) be an ideal in \(R_n\), i.e. a cyclic code of length \(n\).

1) There is a unique least degree monic polynomial in \(R_n\), say \(g(x)\) which generates \(C\), i.e. \(C = (g(x))\). This polynomial is a generator polynomial of \(C\).

2) \(g(x)\) divides \(x^n - 1\).

3) If \(\deg(g(x)) = r\), then \(C\) has dimension \(n - r\).
4) If \( g(x) = g_0 + g_1 x + \cdots + g_r x^r \), then \( g_0 \neq 0 \) and \( C \) has a generator matrix of the following form

\[
\begin{bmatrix}
g_0 & g_1 & g_2 & \cdots & g_r & 0 & 0 & \cdots & 0 \\
0 & g_0 & g_1 & g_2 & \cdots & g_r & 0 & \cdots & 0 \\
0 & 0 & g_0 & g_1 & \cdots & g_r & 0 & \cdots & 0 \\
\vdots & & & & & \ldots & \vdots & & 0 \\
0 & 0 & 0 & \cdots & 0 & g_0 & g_1 & g_2 & \cdots & g_r
\end{bmatrix}
\]

where each row is a right shift of the previous row.

**Lemma 1.3.1** \([39]\) Let \( C = \langle g(x) \rangle \) where \( g(x) \mid (x^m - 1) \). Then,

\[ C = \langle f(x)g(x) \rangle \]

for all \( f(x) \) such that \( (f(x), (x^m - 1)/g(x)) = 1 \).

**Proof:** It is clear that \( \langle f(x)g(x) \rangle \subseteq \langle g(x) \rangle \). Since \( (f(x), g(x)/(x^m - 1)) = 1 \)
there exist \( s(x), t(x) \in F_q[x]/(x^m - 1) \) such that \( s(x)f(x) + t(x)(x^m - 1)/g(x) = 1 \).
Hence, \( s(x)f(x)g(x) + t(x)(x^m - 1) = g(x) \), i.e. \( s(x)f(x)g(x) = g(x) \). Thus, \( g(x) \in \langle f(x)g(x) \rangle \), i.e. \( \langle g(x) \rangle \in \langle f(x)g(x) \rangle \). \( \square \)

We would like to point out an important fact about cyclic codes. As mentioned above, each cyclic code can be identified with a generator polynomial which divides \( x^n - 1 \). Assume that our code is over \( F_q \), \( (n, q) = 1 \), and we let \( \omega \) denote a primitive \( n^{th} \) root of unity over \( F_q \) i.e \( \omega \) is an \( n^{th} \) root of unity of order \( n \). Suppose that \( x^n - 1 = m_1(x)m_2(x) \cdots m_t(x) \) over \( F_q[x] \) where each \( m_i(x) \) is a monic irreducible polynomial in \( F_q[x] \). Then, \( g(x) \) is a product of some \( m_i(x) \)'s. A suitable power of \( \omega \).
say $\omega^j$ is a root of a $m_i(x)$ for some $i$. And the other roots of $m_i(x)$ are basically the powers of $\omega$ that belong into the same cyclotomic coset as $j_i$. Thus, we can uniquely identify a cyclic code through the powers of a primitive $n^{th}$ root of unity.

Next, we give the definition of Mattson-Solomon polynomials. Using these polynomials, we can establish an important bound (BCH bound) for cyclic codes.

Let $\omega$ be a primitive $n^{th}$ root of unity over $F_q$. For each polynomial $p(x) \in R_n$ we can associate another polynomial $p_{ms}(x)$ called the Mattson-Solomon polynomial for $p(x)$, which is defined as follows:

$$p_{ms}(x) = \sum_{i=0}^{n-1} p(\omega^{-i})x^i. \quad (1.3)$$

The equation (1.3) together with the fact that $\omega^n = 1$, leads to

$$p_{ms}(x) = \sum_{i=0}^{n-1} p(\omega^{n-i})x^i = \sum_{i=1}^{n} p(\omega^i)x^{n-i}.$$ 

Given the Mattson-Solomon polynomial $p_{ms}(x)$, we can recover the original polynomial $p(x)$:

**Theorem 1.3.3** [45] For any polynomial $p(x)$ in $R_n$, we have

$$p(x) = \frac{1}{n} \sum_{i=0}^{n-1} p_{ms}(\omega^i)x^i. \quad (1.4)$$

A few applications of the Mattson-Solomon polynomials are as follows:

**Corollary 1.3.1** [45] The Hamming weight of $p(x) \in R_n$ is equal to $n - s$ where $s$ is the number of zeros of $p_{ms}(x)$ among the $n^{th}$ roots of unity.

**Corollary 1.3.2** [45] The Hamming weight of $p(x) \in R_n$ is at least $n - \deg(p_{ms}(x))$. 

Corollary 1.3.3 [45](BCH bound) Let $\omega$ be a primitive $n$-th root of unity. Let $C$ be a cyclic code in $R_n$, with generator polynomial $g(x)$. Suppose that for some nonnegative integer $b$, $g(\omega^{b+i}) = 0$ for all $0 \leq i \leq d - 1$. Then, the minimum weight of $C$ is at least $d + 1$.

1.3.1 Some Invariant Theory

The **Hamming weight** of a codeword $u = (u_1, u_2, \ldots, u_n)$ is defined by

$$w(u) = |\{j | u_j \neq 0, 1 \leq j \leq n\}|,$$

i.e the number of the nonzero entries of $u$. The **Hamming weight enumerator** $W_C^H(x, y)$ of a linear $[n, k, d]$ code $C$ over a finite field $F_q$, is defined as follows:

$$W_C^H(x, y) = \sum_{u \in C} x^{n - w(u)} y^{w(u)} = \sum_{i} A_i x^{n-i} y^i$$

(1.5)

where $A_i = |\{u \in C | w(u) = i\}|$ i.e the number of codewords in $C$ with weights equal to $i$.

**Theorem 1.3.4** [39] The relation between the Hamming weight enumerators of $C$ and its dual $C^\perp$ is given by

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

(1.6)

**Definition 1.3.2** The map that sends $W_C^H(x, y)$ to $\frac{1}{|C|} W_C^H(x + (q - 1)y, x - y)$ is called the **MacWilliams (Krawtchouk) transform**.

**Definition 1.3.3** A code $C$ is called **self-dual** if and only if $C = C^\perp$. A code $C$ is called **formally self-dual** if and only if its Hamming weight enumerator coincides with its MacWilliams transform.
It is clear that a self-dual code is also formally self-dual.

**Definition 1.3.4** A binary self-dual code with all weights divisible by 4 is called a doubly-even or Type II code; otherwise it is called a singly-even or Type I code.

An extended Hamming code with parameters [8, 4, 4], extended Golay code with parameters [24, 12, 8], and other important extended quadratic residue codes are all Type II codes.

Self-dual codes form an important class of codes. It is also shown by Pless et al. in [44] that self-dual codes over a finite field $F_q$ satisfy a modified Varshamov bound.

It is of great importance to find out where the weight enumerators of self-dual codes live. Using invariant theory we are able to answer this question. Also, by applying invariant theory we get important information on self-dual codes.

Let $G$ be a finite group of linear transformations on $n$ (complex) variables $x_1, x_2, \ldots, x_n$. In other words $G$ is a multiplicative group of nonsingular complex $n \times n$ matrices. Let $g$ be the order of $G$ and let $I$ stand for the $n \times n$ identity matrix.

Let $f(x) = f(x_1, \ldots, x_n)$ and $A = (a_{ij})$ be an $n \times n$ matrix over complex numbers. We define

$$f(Ax) := f(\ldots + \sum_{j=1}^{n} a_{ij} x_j, \ldots)$$

(The symbol $:=$ refers to a defining equation.) In other words, $f(Ax)$ is a polynomial obtained by applying the transformation $A$ (viewing the matrix $A$ as a transformation) to $x = (x_1, \ldots, x_n)$.

**Definition 1.3.5** $f(x)$ is an invariant polynomial of $G$ if and only if $f(Ax) = f(x)$, for all $A \in G$. 15
By definition (1.3.5) we see that if \( f, g \) are invariants of \( G \) so are \( f + g \) and \( fg \). Hence, the invariants of \( G \) form a ring, say \( \mathcal{R}(G) \). In order to characterize \( \mathcal{R}(G) \) it is sufficient to characterize the invariants that are homogenous polynomials. since any invariant is a sum of homogenous invariants.

**Definition 1.3.6** Polynomials \( f_1(x), \ldots, f_m(x) \) are algebraically dependent if there is a polynomial \( P \) with complex coefficients, not all zero, such that \( P(f_1(x), \ldots, f_m(x)) \equiv 0 \). Otherwise, \( f_1(x), \ldots, f_m(x) \) are algebraically independent.

**Lemma 1.3.2** [30] Any \( n + 1 \) polynomials in \( n \) variables are algebraically dependent.

The following theorem provides an easy way to show whether polynomials are algebraically independent or not.

**Theorem 1.3.5** [15] Let \( f_1(x), \ldots, f_m(x) \in \mathbb{C}[x] \). Then \( f_1, f_2, \ldots, f_m \) are algebraically independent if and only if their Jacobian is not equal to zero, i.e.

\[
\left| \frac{\partial f_i(x)}{\partial x_i} \right| \neq 0.
\]

**Theorem 1.3.6** [38] If \( f(x) \) is any polynomial, then the average of \( f(x) \) over the group \( G \), \( h(x) = \frac{1}{g} \sum_{A \in G} f(Ax) \) is an invariant of \( G \).

Using the above theorem we will be able to construct homogenous polynomials which are invariant under all elements of the matrix group \( G \).
1.3.2 Ring of invariants of binary self-dual codes

Our goal in this section is to show how to determine the ring of invariants of a Type II code. We are going to apply this method to some weight enumerators in Chapter 4 to determine their ring of invariants. We are going to follow [39] closely. To achieve this goal, first we will determine which transformations leave the Hamming weight enumerator of a Type II code invariant. Next, we invoke a theorem of T. Molien which tells us the number of algebraically independent invariants of each degree. This theorem will suggest the degrees of homogenous invariant polynomials. Using Theorem 1.3.6, we will be able to find algebraically independent homogeneous invariants of the group. Hence, the ring of invariants will be determined.

MacWilliams identity (1.6) imposes a strong restriction on the weight enumerator of a self-dual binary code. In other words, since $C = C^\perp$ and $q = 2$, we have

$$W^H_C(x, y) = \frac{1}{|C|} W^H_C(x + y, x - y). \quad (1.7)$$

Since $\dim(C) + \dim(C^\perp) = n$, we get $\dim(C) = n/2$. This implies that $|C| = 2^{n/2}$. Hence,

$$W^H_C(x, y) = \frac{1}{2^{n/2}} W^H_C(x + y, x - y)$$

$$= W^H_C \left( \frac{x + y}{\sqrt{2}}, \frac{x - y}{\sqrt{2}} \right).$$

This means that the Hamming weight enumerator $W^H_C(x, y)$ of a Type II code is invariant under the transformation:
Another transformation that leaves $W_C^H(x, y)$ invariant comes from the fact that $C$ is of Type II, i.e all weights are divisible by 4. This implies that exponents of $y$ in $W_C^H(x, y)$ are always divisible by 4. Thus,

$$W_C^H(x, y) = W_C^H(x, iy)$$

where $i^2 = -1$.

In other words, $W_C^H(x, y)$ is invariant under the transformation:

$$T_2 : egin{bmatrix} x \\ y \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}.$$  

It is clear that $W_C^H(x, y)$ remains unchanged under any combinations of $T_1$ and $T_2$, such as $T_1T_2$, $T_1^2$, $T_2T_1$, etc. Hence, $W_C^H(x, y)$ is invariant under the group, say $G$, generated by

$$\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \text{ and } \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}.$$  

The order of $G$ is 192.

We would like to know the number of linearly independent homogenous invariants under this group $G$, or any group in general, of degree $\nu$. The following important result published in 1897 is due to T. Molien:

$$\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \text{ and } \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}.$$  

The order of $G$ is 192.

We would like to know the number of linearly independent homogenous invariants under this group $G$, or any group in general, of degree $\nu$. The following important result published in 1897 is due to T. Molien:
Theorem 1.3.7 [38] (Molien series) The number of linearly independent invariants of $G$ of degree $\nu$ is the coefficient of $\lambda^\nu$ in the expansion of

$$\Phi(\lambda) = \frac{1}{g} \sum_{A \in G} \frac{1}{|A - \lambda I|}$$

where $|A|$ stands for the determinant of $A$. Also $I$ denotes the identity matrix. $\phi(\lambda)$ is called the Molien series of $G$. A proof of this result can be found in [39].

The Hamming weight enumerators of the extended Hamming code and the extended Golay code with parameters $[8,4,4|2$ and $[24,12,8|2$ are respectively

$$W^H_H(x, y) := x^8 + 14x^4y^4 + y^8,$$  \hspace{1cm} (1.8)

and

$$W^H_G(x, y) := x^{24} + 759x^{16}y^8 + 2576x^{12}y^{12} + 759x^8y^{16} + y^{24}. \hspace{1cm} (1.9)$$

Theorem 1.3.8 [16] (Gleason's Theorem)

The Hamming weight enumerator $W^H_C(x, y)$ of a binary Type II code $C$ is a polynomial in $W^H_H(x, y)$ and $W^H_G(x, y)$, i.e.

$$W^H_C(x, y) \in \mathbb{C}[W^H_H(x, y), W^H_G(x, y)].$$

The degree of the Hamming weight enumerator of a code $C$ of length $n$ is equal to $n$. From the above theorem, we see that a weight enumerator of a Type II code $C$ of length $n$ is a polynomial formed by polynomials of degrees 8 and 24. This implies that $n = 8k + 24l$ for some $k, l \in \mathbb{N}$. Thus, we have

Corollary 1.3.4 The length of a Type II code $C$ is divisible by 8.
There exist theorems similar to the above one for weight enumerators of other types. These theorems are referred to as Gleason-type theorems. In Chapter 4 we are going to obtain Gleason-type theorems for 2-ply weight enumerators of self-dual binary codes and complete weight enumerators of some codes over $\mathbb{F}_2 + u\mathbb{F}_2$.

**An application of Gleason’s Theorem (Theorem 1.3.8):** We are going to use this theorem to determine the Hamming weight enumerator of the $[48, 24]$ quadratic residue code. By investigating the generator matrix of this code [39], we know that it is a self-dual code with minimum distance 12. Hence the weight enumerator of this code has the following form:

$$W(x, y) := x^{18} + A_{12}x^{36}y^{12} + \cdots$$

$A_{12}$ is the number of codewords of weight 12. To simplify the problem we make the following observation:

$$W'(x, y) := \frac{W_H^H(x, y)^3 - W_G^H(x, y)}{42} = x^4y^4(x^4 - y^4)^1.$$  

This implies that $\{W_H^H(x, y), W'(x, y)\}$ is also another polynomial basis for Type II codes. Since the quadratic residue code mentioned above is a Type II code, and its weight enumerator is homogenous of degree 48, by Theorem 1.3.8, we must have

$$W(x, y) = aW_H^H(x, y)^6 + bW_H^H(x, y)W'(x, y) + cW'(x, y)^3 \quad (1.10)$$

for some real numbers $a, b, c$. After expanding equation (1.10), and equating the coefficients of both sides, we get

$$a = 1, b = -84, \text{ and } c = 246.$$
Hence, \( W(x, y) \) is uniquely determined by the above values of \( a, b \) and \( c \). Indeed, by equation (1.10), we obtain the Hamming weight enumerator of this code:

\[
W(x, y) = x^{48} + 17296x^{36}y^{12} + 535095x^{32}y^{16} + 3995376x^{28}y^{20} \\
+ 768680x^{24}y^{24} + 39995376x^{20}y^{28} + 535095x^{16}y^{32} + 17296x^{12}y^{36} + y^{18}.
\]

As seen above the Gleason-type theorems play an important role in obtaining some results for self-dual codes.

**Definition 1.3.7** A polynomial basis for \( \mathcal{R}(G) \) (ring of invariants of \( G \)) consists of homogenous invariants \( f_1, \ldots, f_l \) \((l \geq m)\) where \( f_1, \ldots, f_m \) are algebraically independent and

\[
\mathcal{R}(G) = \mathbb{C}[f_1, \ldots, f_m] \text{ if } l = m,
\]

or if \( l \geq m \),

\[
\mathcal{R}(G) = \mathbb{C}[f_1, \ldots, f_m] \oplus f_{m+1}\mathbb{C}[f_1, \ldots, f_m] \oplus \cdots \oplus f_l\mathbb{C}[f_1, \ldots, f_m].
\]

The polynomials \( f_1, \ldots, f_m \) are "free" invariants and can be used as often as needed but \( f_{m+1}, \ldots, f_l \) are "transient" invariants and each can be used at most once. In the ring of invariants of binary Type II codes we have only two free polynomials, basically the weight enumerators of extended Hamming and Golay codes. We can easily see that the Molien series can be identified by looking at the degrees and the structure of a polynomial basis. Let \( \text{deg}(f_i) = d_i \) for all \( 1 \leq i \leq l \). Then,

\[
\phi_G(\lambda) = \frac{1}{\prod_{i=1}^{m}(1 - \lambda^{d_i})}, \text{ if } l = m,
\]

(1.11)
or
\[ \phi_G(\lambda) = \frac{1 + \sum_{j=m+1}^{l} \lambda^{d_j}}{\prod_{i=1}^{m} (1 - \lambda^{d_i})}, \text{ if } l > m. \] (1.12)

1.4 Overview of Dissertation

We now give a brief explanation for the chapters that follow:

In chapter 2 we generalize the complete weight enumerator of a code over a field to the generalized \( r \)-fold weight enumerator of \( r \) codes over a finite ring. We obtain a MacWilliams-type identity for the generalized \( r \)-fold weight enumerator of a code and its dual. We call this theorem the Main Theorem of this chapter. A MacWilliams-type identity combines almost all previously known MacWilliams-type identities as corollaries. Further we have generalized the \( \lambda \)-ply weight enumerator of a code over a field, introduced by Shiromoto in [48], to codes over rings. We also derive a MacWilliams-type identity for the \( \lambda \)-ply weight enumerators of codes over rings from the generalized \( r \)-fold (with \( r = \lambda \)) weight enumerator. This gives an alternative proof of the MacWilliams-type identity for \( \lambda \)-ply weight enumerators obtained in [48]. Next we define generalized \( r \)-fold symmetric and Hamming weight enumerators of \( r \) codes. The former is a generalization of symmetric weight enumerators of codes over \( \mathbb{Z}_4 \) and the latter is a generalization of the Hamming weight enumerator. The MacWilliams identities for these two generalized weight enumerators are obtained from the Main Theorem by specializing the variables of generalized \( r \)-fold weight enumerators in MacWilliams-type identities. Also, the MacWilliams-type identity for the \( r \)-genus weight enumerator for binary codes proven by Duke in [13] follows as
a corollary to our Main Theorem. We conclude this chapter with a generalization of
r-byte weight enumerator of binary codes introduced by Tadashi Wadayama et al. in
[51] to codes over any field.

In chapter 3, we go over the basic facts and definitions of quasi-cyclic codes. We
give independent proofs to some well known and very recent results by a different
approach. In the second section we generalize the method of obtaining codes over
$\mathbb{F}_3$ via a Gray map from codes over $\mathbb{F}_3 + u\mathbb{F}_3$ where $u^2 = 0$, introduced by Gulliver
and Harada in [24], to codes over $\mathbb{F}_5 + u\mathbb{F}_5$ with $u^2 = 1$. First we investigate the
relation between the codes over $R + uR$ where $R$ is a commutative ring and $u^2 = a$
for some $a \in R$ and codes over $R$ via a Gray map. Next we apply the theorems for
the special case $R = \mathbb{F}_5$. We obtain 9 new codes over $\mathbb{F}_5$ with improved minimum
distances. In the third section we generalize the results of the previous section to
codes over $R + uR + \cdots + u^{m-1}R$ where $u^m = a$ for some $a \in R$ and a positive integer
$m > 1$. As an application we take $R = \mathbb{F}_5$ and $m = 3$ and obtain 11 new codes over
$\mathbb{F}_5$ with improved minimum distances. These 20 new record-breaking codes (see [4])
yield other improvements on bounds of minimum distances by applying truncation,
puncturing and shortening on these codes.

In the last chapter we apply invariant theory to 2-ply weight enumerators of binary
self-dual codes with lengths divisible by 4 or 8. In both cases we obtain Gleason-type
theorems and give the corresponding ring of invariants. The 2-ply weight enumerators
of binary self-dual codes with lengths divisible by 8 are shown to be related to the
2-ply weight enumerators of extended Hamming and Golay codes. Finally, we relate
the 2-byte weight enumerators of binary codes with the complete weight enumerators
of codes over \( \mathbb{F}_2 + u\mathbb{F}_2 \). We establish the ring of invariants of codes over \( \mathbb{F}_2 + u\mathbb{F}_2 \) whose images under a Gray map are binary self-dual codes of Type I and II.

All results and proofs that are done by other authors are attributed to them. Otherwise, the results are proven by the author and they are believed to be new.
CHAPTER 2

GENERALIZED r-FOLD WEIGHT ENUMERATORS

Let \( F_q = \{0 = \alpha_0, \alpha_1, \ldots, \alpha_{q-1}\} \) be a finite field of order \( q \). The first generalization of Hamming weight enumerator (1.6) defined in the introduction is the complete weight enumerator of a code, which carries more information about the code than the Hamming weight enumerator. We define the composition of \( u = (u_1, \ldots, u_n) \), denoted by \( \text{comp}(u) \), as \( (s_0, \ldots, s_{q-1}) \) where \( s_i = s_i(u) \) is the number of components \( u_j, 1 \leq j \leq n \), equal to \( \alpha_i \) i.e \( s_i(u) = |\{j : u_j = \alpha_i\}| \). The complete weight enumerator of \( C \) is defined by

\[
W_C(z_0, \ldots, z_{q-1}) = \sum_{u \in C} z_0^{s_0(u)} \cdots z_{q-1}^{s_{q-1}(u)} = \sum_{\mathbf{A}(\mathbf{K})} z_0^{t_0} \cdots z_{q-1}^{t_{q-1}} \tag{2.1}
\]

where \( \mathbf{A}(\mathbf{K}) = |\{u \in C|s_i(u) = t_i, 0 \leq i \leq q-1\}| \), \( \mathbf{K} = (t_0, \ldots, t_{q-1}) \in \mathbb{N}_0^q \), and \( \sum_{i=0}^{q-1} t_i = n \). Here \( \mathbb{N}_0 \) denotes the set of nonnegative integers.

Note that Hamming and complete weight enumerators of a code coincide over a binary field \( F_2 \). The relation between the complete weight enumerator of a code \( C \) and its dual \( C^\perp \) is given by a well-known theorem in the book "The Theory of Error Correcting Codes" [39] by MacWilliams and Sloane:

25
Theorem 2.0.1 [39]

\[ W_{C^*}(z_0, \ldots, z_q) = \frac{1}{|C|} W_C(\sum_{i=0}^{q-1} \chi(\alpha_0 \alpha_i)z_i, \ldots, \sum_{i=0}^{q-1} \chi(\alpha_q \alpha_i)z_i) \text{.} \]

Further, the joint weight enumerator of two binary codes \( C_1 \) and \( C_2 \), as defined by MacWilliams, Mallows and Sloane in [38], is

\[ W_{(C_1, C_2)}^H(z_{00}, z_{01}, z_{10}, z_{11}) = \sum_{u \in C_1, v \in C_2} z_0^{i(u, v)} z_1^{j(u, v)} z_2^{k(u, v)} z_3^{l(u, v)} \]

where

\[ i(u, v) = \{|m|(u_m, v_m) = (0, 0), \quad 1 \leq m \leq n\}, \]

\[ j(u, v) = \{|m|(u_m, v_m) = (0, 1), \quad 1 \leq m \leq n\}, \]

\[ k(u, v) = \{|m|(u_m, v_m) = (1, 0), \quad 1 \leq m \leq n\}, \]

\[ l(u, v) = \{|m|(u_m, v_m) = (1, 1), \quad 1 \leq m \leq n\} \]

and \( u = (u_1, \ldots, u_n) \in C_1, v = (v_1, \ldots, v_n) \in C_2 \).

The relation between the joint weight enumerator of \( C_1 \) and \( C_2 \) and their duals is given by

Theorem 2.0.2 [38]

\[ W_{(C_1^*, C_2^*)}^H(z_{00}, z_{01}, z_{10}, z_{11}) = \frac{1}{|C_1||C_2|} W_{(C_1, C_2)}^H(z_{00} + z_{01} + z_{10} + z_{11}, z_{00} - z_{01} + z_{10} - z_{11},\]

\[ z_{00} + z_{01} - z_{10} - z_{11}, z_{00} - z_{01} - z_{10} + z_{11}) \]
Lately there have been some generalizations over rings. The relation between the Hamming weight enumerators of a code \( C \subseteq R^n \), where \( R = \mathbb{Z}_m \) is the ring of integers modulo \( m \), and its dual \( C^\perp \) is given by

\[
W_{C^\perp}^H(x, y) = \frac{1}{|C|} W_C^H(x + (m - 1)y, x - y).
\]  

**Theorem 2.0.3 [32]**

Our goal in this chapter is to generalize the joint weight enumerator to \( r \) codes over rings and obtain a MacWilliams-type identity, where \( r \) is any positive integer.

In this chapter, we let \( R = \{J_0, J_1, \ldots, J_{m-1}\} \) denote a finite ring (not necessarily commutative) of size \( m > 1 \) with an identity. However we mention that when \( R \) is a field, to conform with the standard notation, the size of \( R \) is denoted by \( q \). Let \( C \) be a left submodule of \( R^n \). By an additive character of \( R \), we mean a character of the additive group \((R, +)\). We will refer to the following lemmas later.

**Lemma 2.0.1** Let \( R \) be a finite ring with size \( > 1 \). There exits an additive character \( \chi \) which is nontrivial over all subgroups of \((R, +)\) with size \( > 1 \).

**Proof:** Since \( R \) is a finite ring, \((R, +)\) is a finite abelian group. Hence by the fundamental theorem of abelian groups, \( R \cong C_{m_1} \times C_{m_2} \times \cdots \times C_{m_s} \) for some \( s \in \mathbb{N} \), and each \( C_{m_i} \) is a cyclic subgroup of size \( m_i \), \( 1 \leq i \leq s \). We can assume that \( R \cong C_{m_1} \times C_{m_2} \) with the rest following by induction on \( s \). We take \( \chi_1 = e^{\frac{2\pi i}{m_1}} \), \( \chi_2 = e^{\frac{2\pi i}{m_2}} \) as characters of \( C_{m_1} \) and \( C_{m_2} \), respectively. It can easily shown that \( \chi = \chi_1 \chi_2 \) is a character of \( R \) and is nontrivial on all subgroups of \( R \) with size \( > 1 \). \( \square \)
Lemma 2.0.2 Let $R$ be a finite ring of size $|R|$, and $\chi$ an additive nontrivial (over all subgroups of $R$ with size $> 1$) character of $R$. Then, for fixed $b \in R$ and $b \neq 0$,

$$
\sum_{a \in R} \chi(ab) = \begin{cases} 
0, & \text{if } b \neq 0, \\
|R|, & \text{if } b = 0.
\end{cases}
$$

(2.7)

**Proof:** If $b = 0$, then it is clear. If $b \neq 0$, then, since $\chi$ is nontrivial over the additive group generated by $b$, $\chi(b) \neq 1$. We set $A = \sum_{a \in R} \chi(ab)$. Now, we consider

$$
A = \sum_{a \in R} \chi(ab) = \sum_{a \in R} \chi((a + 1)b) = \sum_{a \in R} \chi((ab) + b) = \sum_{a \in R} \chi(ab)\chi(b) = A\chi(b).
$$

This implies $A = 0$, since $\chi(b) \neq 1$. □

Lemma 2.0.3 Let $C \subset R^n$ be a linear code, $C^\perp$ its dual and $\chi$ an additive nontrivial (over all subgroups of $R$ with size $> 1$) character of $R$. Then, for fixed $v \notin C^\perp$,

$$
\sum_{u \in C} \chi(\langle u, v \rangle) = 0.
$$

(2.8)

**Proof:** Let $A = \sum_{u \in C} \chi(\langle u, v \rangle)$. Then there exists a $u' \in C$ such that $\beta = \langle u', v \rangle \neq 0$. First we note that since $\chi$ is nontrivial over the subgroup $H = \langle \beta \rangle$, $\chi(\beta) \neq 1$.

$$
A\chi(\langle u', v \rangle) = \sum_{u \in C} \chi(\langle u, v \rangle)\chi(\langle u', v \rangle) = \sum_{u \in C} \chi(\langle u + u', v \rangle) = A.
$$

Hence $A = 0$, since $\chi(\langle u', v \rangle) \neq 1$. □

Let $C_1, \ldots, C_r \in R^n$ be linear codes ($R$-submodules of $R^n$) of length $n$. We adopt the following notations for simplification,

$$
C := C_1 \times C_2 \times \cdots \times C_r \quad \text{and} \quad C^\perp := C_1^\perp \times C_2^\perp \times \cdots \times C_r^\perp
$$
and

$$|C| = |C_1| \times |C_2| \times \cdots \times |C_r|, \quad |C_r^+| = |C_1^+| \times \cdots \times |C_r^+|$$

where $r$ is a positive integer.

For convenience of the reader, first we are going to work out the generalized 2-fold complete weight enumerator of two codes $C_1$ and $C_2$ in detail. Next we will define the most general case and prove the necessary identities.

### 2.1 Generalized 2-fold complete weight enumerator

We define and denote the generalized 2-fold complete weight enumerator of two linear codes $C_1$ and $C_2$ over a finite ring $R = \{\beta_0, \ldots, \beta_{m-1}\}$ of length $n$ as follows:

$$W_C(z_0, \ldots, z_{i_1}, \ldots, z_{i_{m-1}}, \ldots, z_{m-1,m-1}) = \sum_{(u_1, u_2) \in C} z_0^{s_0}(u_1, u_2) \cdots z_{i_1}^{s_{i_1}}(u_1, u_2) \cdots z_{m-1,m-1}^{s_{m-1,m-1}}(u_1, u_2)$$

where

$$s_{i_1}(u_1, u_2) := |\{j|(u_{1j}, u_{2j}) = (\beta_{i_1}, \beta_{i_2})\}|$$

and $u_1 \in C_1, u_2 \in C_2$, where $u_1 = (u_{11}, u_{12}, \ldots, u_{1n})$, and $u_2 = (u_{21}, u_{22}, \ldots, u_{2n})$.

This is a very natural generalization of the complete weight enumerator of a code $C$ which is a special case when $r = 1$ and $R = F_q$. Also, when $r = 2$ and $R = F_2$ the 2-fold weight enumerator is called the joint weight enumerator.
Let \( u \in V := R^n \) be a fixed element and \( \chi \) a nontrivial additive character of \( R \).

We define \( \chi_u \), a mapping from \( V \) into complex numbers as follows

\[
\chi_u(v) = \chi((u, v)) \text{ for all } v \in V.
\]  

(2.9)

**Lemma 2.1.1**

\[
\chi_u(v + w) = \chi_u(v)\chi_u(w)
\]  

(2.10)

for all \( u, v, w \in C \) (i.e. \( \chi_u \) is an homomorphism from \((V, +)\) to \((C, \cdot)\)).

**Proof:** In the following proof, we are going to use the definitions (2.9) and (1.1),

\[
\chi_u(v + w) = \chi((u, v + w)) = \chi\left(\sum_{i=0}^{n}(u_i + w_i)u_i\right) \\
= \chi\left(\sum_{i=0}^{n}u_iu_i + \sum_{i=0}^{n}w_iu_i\right) = \chi\left(\sum_{i=0}^{n}u_iu_i\right)\chi\left(\sum_{i=0}^{n}w_iu_i\right) \\
= \chi((u, v))\chi((u, w)) = \chi_u(v)\chi_u(w). \quad \square
\]

Below we define a Fourier transform \( \hat{f} \) of a function

\[
f : C_1 \times C_2 \rightarrow G
\]

where \( G \) is an algebra over the complex numbers. For instance, \( G = C[x_1, x_2, \ldots, x_l] \), the algebra of polynomials in \( l \) variables over \( C \). We establish the connection between the Fourier transform \( \hat{f} \) and \( f \). Our next goal is to find a MacWilliams-type identity which relates the weight enumerator of these codes with their duals. We are going to use a technique similar to one used in the case \( r = 1 \) [39], with some generalizations.
Lemma 2.1.2 Let \( \chi \) be an additive character of \( R \) which is nontrivial over all subgroups of \( R \) of size \( > 1 \) and \( f \) be a function such that

\[
f: \mathbb{C} \to G
\]

where \( G \) is a commutative and associative algebra over complex numbers. For \( u, v \in V, 1 \leq i \leq 2 \) define

\[
\hat{f}((u_1, u_2)) = \sum_{(v_1, v_2) \in V^2} \chi_{u_1}(v_1)\chi_{u_2}(v_2)f((v_1, v_2)).
\]

Then

\[
\sum_{(v_1, v_2) \in G} f((v_1, v_2)) = \frac{1}{|G|} \sum_{(u_1, u_2) \in G} \hat{f}((u_1, u_2)). \tag{2.11}
\]

Proof: The right hand side of equation (2.11) is

\[
\sum_{(u_1, u_2) \in G} \hat{f}((u_1, u_2))
\]

\[
= \sum_{(u_1, u_2) \in G} \sum_{(v_1, v_2) \in V^2} \chi_{u_1}(v_1)\chi_{u_2}(v_2)f((v_1, v_2))
\]

\[
= \sum_{(u_1, u_2) \in G} \sum_{(v_1, v_2) \in G} \chi_{u_1}(v_1)\chi_{u_2}(v_2)f((v_1, v_2)) \tag{K}
\]

\[
+ \sum_{(u_1, u_2) \in G} \sum_{(v_1, v_2) \notin G} \chi_{u_1}(v_1)\chi_{u_2}(v_2)f((v_1, v_2)) \tag{L}
\]

We compute the sums \( K \), and \( L \) respectively.

\[
K = \sum_{(u_1, u_2) \in G} \sum_{(v_1, v_2) \in G} \chi((u_1, v_1))\chi((u_2, v_2))f((v_1, v_2)).
\]

31
Since \((u_j, v_j) = 0\) for all \(u_j \in C_j\) and \(v_j \in C_j^+\), \(1 \leq j \leq 2\) and \(\chi(0) = 1\), we have

\[
K = |C| \sum_{(v_1, v_2) \in C^+} f((v_1, v_2))
\]

and now we compute \(L\),

\[
L = \sum_{(u_1, u_2) \in C} \sum_{(v_1, v_2) \in C^+} \chi_{u_1}(v_1) \chi_{u_2}(v_2) f((v_1, v_2)).
\]

Changing the order of the sums,

\[
L = \sum_{(v_1, v_2) \in C^+} \sum_{(u_1, u_2) \in C} \chi_{u_1}(v_1) \chi_{u_2}(v_2) f((v_1, v_2)).
\]

If \((v_1, v_2) \notin C^+\), then \(v_{i_0} \notin C_{i_0}^+\) for some \(i_0 \in \{1, 2\}\). Without loss of generality let us assume that \(v_1 \notin C_1^+\). Then, for fixed \((v_1, v_2) \notin C^+\) we have

\[
M = f((v_1, v_2)) \sum_{(u_1, u_2) \in C} \chi_{u_1}(v_1) \chi_{u_2}(v_2)
\]

\[
= f((v_1, v_2)) \left( \sum_{u_1 \in C_1} \chi((u_1, v_1)) \right) \left( \sum_{u_2 \in C_2} \chi((u_2, v_2)) \right)
\]

\[= 0 \text{ by (2.8)}
\]

Therefore,

\[
\sum_{(u_1, u_2) \in C} \hat{f}((u_1, u_2)) = K
\]

\[
\sum_{(u_1, u_2) \in C} \hat{f}((u_1, u_2)) = |C| \sum_{(v_1, v_2) \in C^+} f((v_1, v_2)).
\]
If we divide both sides of the last equation by \(|C|\), we have the result. □

For \(\beta_i \in R, 0 \leq i \leq m - 1\) we define

\[
W_i(\beta_j) = \begin{cases} 
1, & \text{if } j = i, \\
0, & \text{otherwise}.
\end{cases}
\]  

(2.12)

and we observe that

\[
s_{1,2}(u_1, u_2) = \sum_{j=1}^{n} W_1(u_{1j}) W_2(u_{2j}).
\]  

(2.13)

**Theorem 2.1.1** Let \(\chi\) be an additive character of \(R\) which is nontrivial over all subgroups of \(R\) with size > 1. Using the above notation, the relation between the generalized complete 2-fold weight enumerator of linear codes \(C_1, C_2\), or \(C\) in short, and their duals \(C_1^+, C_2^+,\) or \(C^+\) in short, is given by

\[
W_{C^+}(z_{00}, \ldots, z_{11}, \ldots, z_{m-1,m-1}) = \frac{1}{|C|} W_C \left( \sum_{j_1,j_2=0}^{m-1} \chi(\beta_{j_1}, \beta_{j_2}) z_{j_1,j_2} \cdots \sum_{j_1,j_2=0}^{m-1} \chi(\beta_{j_1}, \beta_{m-1}) z_{j_1,j_2} \right).
\]  

(2.14)

**Proof:**

We take the function \(f(z_{00}, \ldots, z_{11}, \ldots, z_{m-1,m-1}) \in C[z_{00}, \ldots, z_{m-1,m-1}]\), defined by

\[
f((u_1, u_2)) = z_{00}^{s_{00}(u_1, u_2)} \cdots z_{11}^{s_{11}(u_1, u_2)} \cdots z_{m-1,m-1}^{s_{m-1,m-1}(u_1, u_2)}.
\]

33
Then, we apply Lemma 2.1.2:

\[ \hat{f}((u_1, u_2)) = \sum_{(v_1, v_2) \in V^2} \chi((u_1, v_1)) \chi((u_2, v_2)) \hat{f}((v_1, v_2)) \]

\[ = \sum_{(v_1, v_2) \in V^2} \chi((u_1, v_1)) \chi((u_2, v_2)) z_{00}^{s_{11}^{u_1, u_2}} \cdots z_{11}^{s_{11}^{u_1, u_2}} \cdots z_{m-1,m-1}^{s_{11}^{u_1, u_2}}. \]

We further expand the first sums over each coordinate and we use (2.13) and (1.1), and let \( \sum_j \) in the exponents, stand for the sum over \( j \), varying from 1 to \( n \). Then, the last expression is equal to

\[ = \sum_{v_1 \in R} \sum_{v_2 \in R} \cdots \sum_{v_1 \in R} \sum_{v_2 \in R} \cdots \sum_{v_2 \in R} \]

\[ \chi(v_{11} u_{11} + v_{12} u_{12} + \cdots + v_{1n} u_{1n}) \chi(v_{21} u_{21} + v_{22} u_{22} + \cdots + v_{2n} u_{2n}) \cdot \]

\[ \sum_{00}^{w_0(v_1,v_2)} \cdots \sum_{11}^{w_1(v_1,v_2)} \cdots \sum_{m-1,m-1}^{w_m(v_1,v_2)}. \]

Now we expand the sums that are powers of \( z \)'s, and also use the fact that \( \chi \) is a homomorphism. hence the previous expression

\[ = \sum_{v_11 \in R} \sum_{v_12 \in R} \cdots \sum_{v_1n \in R} \sum_{v_21 \in R} \cdots \sum_{v_2n \in R} \]

\[ \chi(v_{11} u_{11}) \cdots \chi(v_{1n} u_{1n}) \chi(v_{21} u_{21}) \cdots \chi(v_{2n} u_{2n}) \cdot \]

\[ \sum_{00}^{w_0(v_11,v_21)+w_0(v_12,v_22)+\cdots+w_0(v_1n,v_2n)}. \]

\[ \vdots \]

34
Finally, we rearrange the sums by collecting the similar terms. The previous expression

\[
\sum_{j_{11},j_{21}} \chi(j_{11},u_{11}) \chi(j_{21},u_{21}) z_{00}^{w_{0}(u_{11}) w_{0}(u_{21})} \cdots z_{11}^{w_{1}(u_{11}) w_{1}(u_{21})} \cdots z_{m-1,m-1}^{w_{m-1}(u_{11}) w_{m-1}(u_{21})}
\]  

By way of illustration, using the definition of \(w_j\) (2.12) we observe that when \(v_{11} = j_{11}, v_{21} = j_{21},\)

\[
\chi(v_{11},u_{11}) \chi(v_{21},u_{21}) z_{00}^{w_{0}(u_{11}) w_{0}(u_{21})} \cdots z_{11}^{w_{1}(u_{11}) w_{1}(u_{21})} \cdots z_{m-1,m-1}^{w_{m-1}(u_{11}) w_{m-1}(u_{21})} = \chi(\beta_{j_{11}},u_{11}) \chi(\beta_{j_{21}},u_{21}) z_{j_{11},j_{21}}.
\]

Then, we run the sums through the elements of \(R\), and get

\[
(2.15) = \left( \sum_{j_{11},j_{21}}^{m-1} \chi(\beta_{j_{11}},u_{11}) \chi(\beta_{j_{21}},u_{21}) z_{j_{11},j_{21}} \right) \left( \sum_{j_{1},j_{2}}^{m-1} \chi(\beta_{j_{1}},u_{12}) \chi(\beta_{j_{2}},u_{22}) z_{j_{1},j_{2}} \right) \cdots \\
\cdots \left( \sum_{j_{1},j_{2}}^{m-1} \chi(\beta_{j_{1}},u_{n}) \chi(\beta_{j_{2}},u_{n}) z_{j_{1},j_{2}} \right).
\]

35
Using (2.1), we recall that \( s_{112}(u_1, u_2) \) counts the vertical columns of \((u_1, u_2)\) which equal to \((\beta_{11}, \beta_{12})\). Hence, the previous expression is equal to

\[
\prod_{i_1,j_2 \in Q} \left( \sum_{j_1,j_2=0}^{m-1} \chi(\beta_{j_1} \beta_{i_1}) \chi(\beta_{j_2} \beta_{i_2}) z_{j_1,j_2} \right)^{s_{112}(u_1,u_2)}
\]

Thus,

\[
\tilde{f}((u_1, u_2)) = \prod_{i_1,j_2 \in Q} \left( \sum_{j_1,j_2=0}^{m-1} \chi(\beta_{j_1} \beta_{i_1}) \chi(\beta_{j_2} \beta_{i_2}) z_{j_1,j_2} \right)^{s_{112}(u_1,u_2)}
\]

We use equation (2.11) in Lemma 2.1.2 and apply the above result to get,

\[
\sum_{(u_1,u_2) \in C_1}^{s_{00}(u_1,u_2)} \ldots z_{i_1,j_2}^{s_{112}(u_1,u_2)} \ldots z_{m-1,m-1}^{s_{m-1,m-1}(u_1,u_2)}
\]

\[
= \frac{1}{|C|} \sum_{(u_1,u_2) \in C_1} \prod_{i_1,j_2 \in Q} \left( \sum_{j_1,j_2=0}^{m-1} \chi(\beta_{j_1} \beta_{i_1}) \chi(\beta_{j_2} \beta_{i_2}) z_{j_1,j_2} \right)^{s_{112}(u_1,u_2)}
\]

This completes the proof. □

The corollary below is a MacWilliams-type identity for the joint weight enumerator of two codes \(C_1\) and \(C_2\) over a ring \(R\). The case when \(R\) is a field is proved by MacWilliams et al in [38].

**Corollary 2.1.1**

\[
W_{C_1}(z_{00}, \ldots, z_{i_1,j_2}, \ldots, z_{m-1,m-1})
\]

\[
= \frac{1}{|C|} W_{C}(\sum_{j_1,j_2=0}^{m-1} \chi(\beta_{j_1} \beta_{0}) \chi(\beta_{j_2} \beta_{0}) z_{j_1,j_2}, \ldots, \sum_{j_1,j_2=0}^{m-1} \chi(\beta_{j_1} \beta_{i_1}) \chi(\beta_{j_2} \beta_{i_2}) z_{j_1,j_2}, \ldots, \sum_{j_1,j_2=0}^{m-1} \chi(\beta_{j_1} \beta_{m-1}) \chi(\beta_{j_2} \beta_{m-1}) z_{j_1,j_2}).
\]

36
Also for case $r = 1$, the relation between the complete weight enumerator of a code and its dual over a finite ring is obtained as

**Corollary 2.1.2**

\[
W_C(z_0, \ldots, z_{r-1}, \ldots, z_{m-1}) = \frac{1}{|C|} \left( \sum_{j_1=0}^{m-1} \gamma(\beta_{j_1}z_0) z_{j_1} \right) \left( \sum_{j_1=0}^{m-1} \gamma(\beta_{j_1}z_1) z_{j_1} \right) \left( \sum_{j_1=0}^{m-1} \gamma(\beta_{j_1}z_{r-1}) z_{j_1} \right).
\]

### 2.2 Generalized $r$-fold complete weight enumerator

Let $r$ be a positive integer. We define the generalized $r$-fold complete weight enumerator of $r$ codes

\[
W_C(z_{00}, \ldots, z_{r12}, \ldots, z_{r-m+1,m-1}, \ldots, z_{m-1,m-1}) \text{ of } C \text{ as follows}
\]

\[
W_C(z_{00}, \ldots, z_{r12}, \ldots, z_{r-m+1,m-1}, \ldots, z_{m-1,m-1}) =
\sum_{(u_1, \ldots, u_r) \in C} z_{00}^{s_{112}...r(u_1, u_2, \ldots, u_r)} \cdots z_{r12}^{s_{112}...r(u_1, u_2, \ldots, u_r)} \cdots z_{m-1,m-1}^{s_{112}...r(u_1, u_2, \ldots, u_r)}
\]

where

\[
s_{i_1i_2...i_r}(u_1, u_2, \ldots, u_r) :=
\]

\[
|\{j| (u_{i_1, j}, u_{i_2, j}, \ldots, u_{i_r, j}) = (\beta_{i_1}, \beta_{i_2}, \ldots, \beta_{i_r}), 1 \leq j \leq r}\}|
\]

and $u_1 \in C_1, u_2 \in C_2, \ldots, u_r \in C_r$ for all $u_1 = (u_{i_1}, u_{i_2}, \ldots, u_{i_n}), u_2 = (u_{i_1}, \ldots, u_{i_n}), \ldots, u_r = (u_{r1}, u_{r2}, \ldots, u_{rn})$ and $i_1, \ldots, i_r \in \{0, \ldots, m-1\}$.

Here, when computing $s_{i_1, \ldots, i_r}(u_1, \ldots, u_r)$ we are taking the vertical (cross sections) columns of $r$ vectors $u_1, \ldots, u_r$ and counting the $r$-tuples which are equal to $(\beta_{i_1}, \beta_{i_2}, \ldots, \beta_{i_r})$. 

37
Example: Let \( r = 3, q = 5, Z_5 = \{0, 1, 2, 3, 4\}, n = 4 \) and assume that \( C_1, C_2 \) and \( C_3 \) are codes over \( Z_5^4 \), and further
\[
\begin{align*}
\boldsymbol{u}_1 &= (2, 3, 4, 2) \in C_1, \\
\boldsymbol{u}_2 &= (1, 2, 3, 1) \in C_2, \quad \text{and} \\
\boldsymbol{u}_3 &= (1, 3, 1, 1) \in C_3.
\end{align*}
\]
Then, using the above definitions, we have
\[
s_{i_1i_2i_3}(\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{u}_3) = \begin{cases} 
2, & \text{if } (i_1, i_2, i_3) = (2, 1, 1) \\
1, & \text{if } (i_1, i_2, i_3) = (3, 2, 3) \\
1, & \text{if } (i_1, i_2, i_3) = (4, 3, 1) \\
0, & \text{otherwise.}
\end{cases}
\] (2.17)

We observe that
\[
\sum_{(i_1, i_2, \ldots, i_r) \in Q^r} s_{i_1i_2i_3}(\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_r) = n
\]
where \( Q = \{0, 1, 2, \ldots, m\} \). Note that if \( r = 1 \) and \( r = 2 \) then the definition of the generalized \( r \)-fold complete weight enumerator coincides with the usual complete weight enumerator (2.1) and joint weight enumerator (2.3) respectively.

Below we define a Fourier transform \( \hat{\boldsymbol{f}} \) of a function
\[
\hat{\boldsymbol{f}} : C_1 \times C_2 \times \cdots \times C_r \to G
\]
where \( G \) is an algebra over complex numbers. We establish the connection between the Fourier transform \( \hat{\boldsymbol{f}} \) and \( \boldsymbol{f} \).

**Lemma 2.2.1 (The Fundamental Lemma)** Let \( \chi \) be an additive character of \( R \) which is nontrivial over all subgroups of \( R \) and \( \boldsymbol{f} \) be a function such that
\[
\boldsymbol{f} : \mathbb{C} \to G
\]
where \( G \) is a commutative and associative algebra over complex numbers and \( \chi_u \) is defined in (2.9). Define

\[
\hat{f}((u_1, u_2, \ldots, u_r)) = \sum_{(v_1, v_2, \ldots, v_r) \in V^r} \chi_{u_1}(v_1)\chi_{u_2}(v_2) \cdots \chi_{u_r}(v_r) f((v_1, v_2, \ldots, v_r)).
\]

Then

\[
\sum_{(v_1, v_2, \ldots, v_r) \in \mathbb{C}^r} f((v_1, v_2, \ldots, v_r)) = \frac{1}{|\mathbb{C}|} \sum_{(u_1, u_2, \ldots, u_r) \in \mathbb{C}^r} \hat{f}((u_1, u_2, \ldots, u_r)). \tag{2.18}
\]

Proof:

\[
\sum_{(u_1, u_2, \ldots, u_r) \in \mathbb{C}^r} \hat{f}((u_1, u_2, \ldots, u_r)) \]

\[
= \sum_{(u_1, u_2, \ldots, u_r) \in \mathbb{C}^r} \sum_{(v_1, v_2, \ldots, v_r) \in V^r} \chi_{u_1}(v_1)\chi_{u_2}(v_2) \cdots \chi_{u_r}(v_r) f((v_1, v_2, \ldots, v_r))
\]

\[
= \sum_{(u_1, u_2, \ldots, u_r) \in \mathbb{C}^r} \sum_{(v_1, v_2, \ldots, v_r) \in \mathbb{C}^r} \chi_{u_1}(v_1)\chi_{u_2}(v_2) \cdots \chi_{u_r}(v_r) f((v_1, v_2, \ldots, v_r))
\]

\[
+ \sum_{(u_1, u_2, \ldots, u_r) \in \mathbb{C}^r} \sum_{(v_1, v_2, \ldots, v_r) \in \mathbb{C}^r} \chi_{u_1}(v_1)\chi_{u_2}(v_2) \cdots \chi_{u_r}(v_r) f((v_1, v_2, \ldots, v_r)).
\]

We compute the sums \( K \), and \( L \) respectively.

\[
K = \sum_{(u_1, \ldots, u_r) \in \mathbb{C}^r} \sum_{(v_1, v_2, \ldots, v_r) \in \mathbb{C}^r} \chi((u_1, v_1)) \cdots \chi((u_r, v_r)) f((v_1, \ldots, v_r)).
\]

Since \( (u_j, v_j) = 0 \) for all \( u_j \in C_j \) and \( v_j \in C_j^+ \), \( 1 \leq j \leq r \) and \( \chi_0 = 1 \), we have

\[
K = |\mathbb{C}| \sum_{(v_1, v_2, \ldots, v_r) \in \mathbb{C}^r} f((v_1, v_2, \ldots, v_r))
\]

39
and now we compute $L$, 

$$L = \sum_{(u_1, \ldots, u_r) \in C} \sum_{(v_1, \ldots, v_r) \in C^+} \chi_{u_1}(v_1) \cdots \chi_{u_r}(v_r) f((v_1, \ldots, v_r)).$$

Change the order of the sums.

$$L = \sum_{(v_1, \ldots, v_r) \in V^r \times C^+} \sum_{(u_1, \ldots, u_r) \in C} \chi_{u_1}(v_1) \cdots \chi_{u_r}(v_r) f((v_1, \ldots, v_r)).$$

First we will show that for every fixed $(v_1, v_2, \ldots, v_r) \notin C^+$, $M = 0$ and hence, the sum over all such $(v_1, v_2, \ldots, v_r) \notin C^+$ will equal to zero, i.e. $L = 0$. Since $(v_1, v_2, \ldots, v_r) \notin C^+$, $u_{i_0} \notin C_{i_0}^\perp$ for some $i_0 \in \{1, 2, \ldots, r\}$. Then,

$$M = \sum_{(u_1, u_2, \ldots, u_r) \in C} \chi_{u_1}(v_1) \chi_{u_2}(v_2) \cdots \chi_{u_r}(v_r) f((v_1, v_2, \ldots, v_r))$$

$$= f((v_1, \ldots, v_{i_0}, \ldots, v_r)) \left( \sum_{u_1 \in C_{i_1}} \chi((u_1, v_1)) \right) \cdots \left( \sum_{u_{i_0} \in C_{i_0}} \chi((u_{i_0}, v_{i_0})) \right)$$

$$= 0 \text{ by (2.8)}$$

$$= 0.$$

Therefore,

$$\sum_{(u_1, u_2, \ldots, u_r) \in C} \hat{f}((u_1, u_2, \ldots, u_r)) = K$$

$$\sum_{(u_1, u_2, \ldots, u_r) \in C} \hat{f}((u_1, u_2, \ldots, u_r)) = |C| \sum_{(v_1, v_2, \ldots, v_r) \in C^+} f((v_1, v_2, \ldots, v_r)).$$
Now dividing both sides of the last equation by $|\mathcal{C}|$, we have the result. □

Using the definition (2.12), we easily see that

$$s_{i_1 i_2 \ldots i_r}(u_1, u_2, \ldots, u_r) = \sum_{j=1}^{n} w_{i_1}(u_{1j}) w_{i_2}(u_{2j}) \cdots w_{i_r}(u_{rj}). \quad (2.19)$$

**Lemma 2.2.2** For fixed $b_1, b_2, \ldots, b_r$ belonging to $R$ and $r$ a positive integer, we have

$$\sum_{a_1 a_2 \cdots a_r \in R} \chi(a_1 b_1) \chi(a_2 b_2) \cdots \chi(a_r b_r) z^{w_0(a_1)w_0(a_2)\cdots w_0(a_r)}_{0   0   0   0} \cdots z^{w_{m-1}(a_1)w_{m-1}(a_2)\cdots w_{m-1}(a_r)}_{m-1 m-1 m-1}$$

$$= \left( \sum_{j_1=0}^{m-1} \chi(\beta_{j_1} b_1) \left( \sum_{j_2=0}^{m-1} \chi(\beta_{j_2} b_2) \cdots \left( \sum_{j_r=0}^{m-1} \chi(\beta_{j_r} b_r) z_{j_1 j_2 \cdots j_r} \right) \cdots \right) \right). \quad (2.20)$$

**Proof**: We apply induction on $r$. For $r = 1$, using the equation (2.12), we have

$$\sum_{a_1 \in R} \chi(a_1 b_1) z^{w_0(a_1)}_{0} \cdots z^{w_{m-1}(a_1)}_{m-1} = \sum_{j_1=0}^{m-1} \chi(\beta_{j_1} b_1) z_{j_1}. \quad (2.21)$$

Hence, (2.20) holds for $r = 1$. Now, we suppose (2.20) holds for $r < s$, and we will show that (2.20) holds for $r = s$. We start with

$$\sum_{a_s \in R} \cdots \sum_{a_3 \in R} \sum_{a_2 \in R} \sum_{a_1 \in R} \chi(a_1 b_1) \chi(a_2 b_2) \cdots \chi(a_s b_s) z^{w_0(a_1)w_0(a_2)\cdots w_0(a_s)}_{0   0   0   0} \cdots z^{w_{m-1}(a_1)w_{m-1}(a_2)\cdots w_{m-1}(a_s)}_{m-1 m-1 m-1}.$$
Expanding the sum involving $a_1$ over the elements of $R$, we see that the previous expression is equal to

$$
\sum_{a_1 \in R} \cdots \sum_{a_3 \in R} \chi(\beta_0 b_1) \chi(a_2 b_2) \chi(a_3 b_3) \cdots \chi(a_s b_s) z_{00} \cdots 0 \cdots z_{m-1, m-1, \ldots, m-1} + \\
\sum_{a_1 \in R} \cdots \sum_{a_3 \in R} \sum_{a_2 \in R} \chi(\beta_1 b_1) \chi(a_2 b_2) \chi(a_3 b_3) \cdots \chi(a_s b_s) z_{00} \cdots 0 \cdots z_{m-1, m-1, \ldots, m-1} + \\
\vdots
$$

We then factor out the term $\chi(\beta_j b_1)$ for $0 \leq j \leq m - 1$ and check that the last expression

$$
= \chi(\beta_0 b_1) \sum_{a_2 \in R} \cdots \sum_{a_3 \in R} \chi(a_2 b_2) \chi(a_3 b_3) \cdots \chi(a_s b_s) z_{00} \cdots 0 \cdots z_{m-1, m-1, \ldots, m-1} + \\
\chi(\beta_1 b_1) \sum_{a_2 \in R} \cdots \sum_{a_3 \in R} \chi(a_2 b_2) \chi(a_3 b_3) \cdots \chi(a_s b_s) z_{00} \cdots 0 \cdots z_{m-1, m-1, \ldots, m-1} + \\
\vdots
$$
+ \chi(\beta_{m-1} b_1) \sum_{a_2 \in R} \cdots \sum_{a_3 \in R} \sum_{a_2 \in R} \chi(a_2 b_2) \chi(a_3 b_3) \cdots \chi(a_s b_s) z_{0 \cdot 1 \cdot 0}^{(\beta_{m-1}) \cdots w_0(\beta_{m-1}) \cdots w_0(\beta_{m-1})}.

= \chi(\beta_0 b_1) \sum_{a_3 \in R} \cdots \sum_{a_2 \in R} \chi(a_2 b_2) \chi(a_3 b_3) \cdots \chi(a_s b_s) z_{0 \cdot 0 \cdot 0}^{w_0(a_2) \cdots w_0(a_s)}.

By induction hypothesis, (2.20) holds for r < s. Here \( z_{0 \cdot 1 \cdot 1 \cdot 0} \), can be viewed as a new variable, say \( y_1 \cdot y_r \) for the purpose of applying the induction hypothesis to derive the next equality. Hence if we apply the induction hypothesis to the above sums respectively. The last expression

\[
+ \chi(\beta_{m-1} b_1) \sum_{a_2 \in R} \cdots \sum_{a_3 \in R} \sum_{a_2 \in R} \chi(a_2 b_2) \chi(a_3 b_3) \cdots \chi(a_s b_s) z_{1 \cdot 0 \cdot 0}^{w_0(a_2) \cdots w_0(a_s)}.
\]

\[
= \chi(\beta_0 b_1) \left( \sum_{j_2=0}^{m-1} \chi(\beta_{j_2} b_2) \left( \sum_{j_3=0}^{m-1} \chi(\beta_{j_3} b_3) \cdots \left( \sum_{j_s=0}^{m-1} \chi(\beta_{j_s} b_s) z_{0 \cdot j_2 \cdot \ldots j_s} \right) \right) \right)
\]

\[
+ \chi(\beta_1 b_1) \left( \sum_{j_2=0}^{m-1} \chi(\beta_{j_2} b_2) \left( \sum_{j_3=0}^{m-1} \chi(\beta_{j_3} b_3) \cdots \left( \sum_{j_s=0}^{m-1} \chi(\beta_{j_s} b_s) z_{1 \cdot j_2 \cdot \ldots j_s} \right) \right) \right) + \cdots
\]

\[
+ \chi(\beta_{m-1} b_1) \left( \sum_{j_2=0}^{m-1} \chi(\beta_{j_2} b_2) \left( \sum_{j_3=0}^{m-1} \chi(\beta_{j_3} b_3) \cdots \left( \sum_{j_s=0}^{m-1} \chi(\beta_{j_s} b_s) z_{m-1 \cdot j_2 \cdot \ldots j_s} \right) \right) \right).
\]
Finally, we collect the $\beta_i$'s under a single sum and the last expression

$$
= \sum_{j_1=0}^{m-1} \chi(\beta_{j_1}b_1) \left( \sum_{j_2=0}^{m-1} \chi(\beta_{j_2}b_2) \left( \sum_{j_3=0}^{m-1} \chi(\beta_{j_3}b_3) \cdots \left( \sum_{j_r=0}^{m-1} \chi(\beta_{j_r}b_r) z_{j_1j_2\ldots j_r} \right) \right) \right). \quad \square
$$

**Theorem 2.2.1 (The Main Theorem)** Let $\chi$ be an additive character of a finite ring $R$ which is nontrivial over all subgroups of $R$ with size $> 1$. Using the above notation, the relation between the generalized complete r-fold weight enumerator of linear codes $C_1, C_2, \ldots, C_r$, or $C$ in short, and their duals $C_1^\perp, \ldots, C_r^\perp$, or $C^\perp$ in short, is given by

$$
W_{C^\perp}(z_{00\ldots 0}, \ldots, z_{j_1j_2\ldots j_r}, \ldots, z_{m-1,m-1\ldots m-1})
= \frac{1}{|C|} W_C \left( \sum_{j_1=0}^{m-1} \chi(\beta_{j_1}b_0) \sum_{j_2=0}^{m-1} \chi(\beta_{j_2}b_1) \cdots \sum_{j_r=0}^{m-1} \chi(\beta_{j_r}b_{r-1}) z_{j_1j_2\ldots j_r}, \ldots, \sum_{j_r=0}^{m-1} \chi(\beta_{j_r}b_{m-1}) z_{j_1j_2\ldots j_r} \right).
$$

(2.21)

**Proof of the Theorem:** We first take

$$
f((u_1, \ldots, u_r)) = z_{00\ldots 0}^{u_1,\ldots, u_r} \cdots z_{j_1\ldots j_r}^{u_1,\ldots, u_r} \cdots z_{m-1,m-1\ldots m-1}^{u_1,\ldots, u_r}.
$$

Then we compute

$$
\hat{f}((u_1, u_2, \ldots, u_r)) = \sum_{(v_1, v_2, \ldots, v_r) \in V^r} \chi((u_1, v_1)) \chi((u_2, v_2)) \cdots \chi((u_r, v_r)) f((v_1, v_2, \ldots, v_r)).
$$

44
\[
= \sum_{(u_1, u_2, \ldots, u_r) \in V^r} \chi((u_1, v_i)) \chi((u_2, v_i)) \cdots \chi((u_r, v_i)) \cdot
\]

\[
\sum_{z_00 \cdots 0 \to z_{s_{11} \cdots t_r}} \sum_{z_{m-1, m-1, \ldots, m-1} \to z_{m-1, m-1, \ldots, m-1}}
\]

\[
= \sum_{v_1 \in V} \sum_{v_2 \in V} \cdots \sum_{u_r \in V} \chi((u_1, v_i)) \chi((u_2, v_i)) \cdots \chi((u_r, v_i)) \cdot
\]

The last expression is obtained by expanding the first sum over each coordinate. We further expand the first sums, we use (2.19), and (1.1), and we let \( \sum_j \) denote the sum over \( j \) ranging from 1 to \( n \). Then, we have

\[
\tilde{f}((u_1, u_2, \ldots, u_r))
\]

\[
= \sum_{v_{11} \in R} \sum_{v_{12} \in R} \cdots \sum_{v_{1n} \in R} \sum_{v_{21} \in R} \cdots \sum_{v_{2n} \in R} \cdots \sum_{v_{rn} \in R} \chi(v_{11} u_{11} + v_{12} u_{12} + \cdots + v_{1n} u_{1n}) \chi(v_{21} u_{21} + v_{22} u_{22} + \cdots + v_{2n} u_{2n}) \cdots
\]

\[
\chi(v_{r1} u_{r1} + v_{r2} u_{r2} + \cdots + v_{rn} u_{rn}) \sum_j w_0(v_{1j} w_0(v_{2j}) \cdots w_0(v_{rj})) \cdots
\]

\[
\sum_{z_{112 \cdots t_r}} \sum_{z_{m-1, m-1, \ldots, m-1}}
\]

Now we expand the sums that are powers of \( z \)'s, and also we use the fact that \( \chi \) is a homomorphism. Then, the previous expression

\[
= \sum_{v_{11} \in R} \sum_{v_{12} \in R} \cdots \sum_{v_{1n} \in R} \sum_{v_{21} \in R} \cdots \sum_{v_{2n} \in R} \cdots \sum_{v_{rn} \in R} \chi(v_{11} u_{11}) \cdots \chi(v_{1n} u_{1n}) \chi(v_{21} u_{21}) \cdots \chi(v_{2n} u_{2n}) \cdots \chi(v_{rn} u_{rn}) \cdot
\]

\[
\sum_{z_{00 \cdots 0}} w_0(v_{11})w_0(v_{21}) \cdots w_0(v_{r1}) + w_0(v_{12})w_0(v_{22}) \cdots w_0(v_{r2}) + \cdots + w_0(v_{1n})w_0(v_{2n}) \cdots w_0(v_{rn})
\]

\[
\cdot
\]

45
Finally, we rearrange the sums by collecting the similar terms. The last expression

\[ \sum_{v_{12} \in R} \sum_{v_{22} \in R} \cdots \sum_{v_{rr} \in R} \chi(v_{11}) \chi(v_{21}) \cdots \chi(v_{rr}) \]

where we take

\[ b_1 = u_{11}, b_2 = u_{21}, \ldots, b_r = u_{rr} \]

for \( 1 \leq i \leq n \) respectively. The last expression

\[ \sum_{j_1=0}^{m-1} \sum_{j_2=0}^{m-1} \sum_{j_r=0}^{m-1} \chi(\beta_{j_1} u_{11}) \chi(\beta_{j_2} u_{21}) \cdots \chi(\beta_{j_r} u_{rr}) z_{j_1 \ldots j_r} \]

where we take

\[ b_1 = u_{11}, b_2 = u_{21}, \ldots, b_r = u_{rr} \]

for \( 1 \leq i \leq n \) respectively. The last expression

\[ \sum_{j_1=0}^{m-1} \sum_{j_2=0}^{m-1} \sum_{j_r=0}^{m-1} \chi(\beta_{j_1} u_{12}) \chi(\beta_{j_2} u_{22}) \cdots \chi(\beta_{j_r} u_{rr}) z_{j_1 \ldots j_r} \]

Finally, we rearrange the sums by collecting the similar terms. The last expression

\[ = \left( \sum_{v_{11} \in R} \sum_{v_{21} \in R} \cdots \sum_{v_{rr} \in R} \chi(v_{11}) \chi(v_{21}) \cdots \chi(v_{rr}) \right) \]

where we take

\[ b_1 = u_{11}, b_2 = u_{21}, \ldots, b_r = u_{rr} \]

for \( 1 \leq i \leq n \) respectively. The last expression

\[ = \left( \sum_{v_{11} \in R} \sum_{v_{21} \in R} \cdots \sum_{v_{rr} \in R} \chi(v_{11}) \chi(v_{21}) \cdots \chi(v_{rr}) \right) \]

Finally, we rearrange the sums by collecting the similar terms. The last expression

\[ = \left( \sum_{v_{11} \in R} \sum_{v_{21} \in R} \cdots \sum_{v_{rr} \in R} \chi(v_{11}) \chi(v_{21}) \cdots \chi(v_{rr}) \right) \]

Finally, we rearrange the sums by collecting the similar terms. The last expression

\[ = \left( \sum_{v_{11} \in R} \sum_{v_{21} \in R} \cdots \sum_{v_{rr} \in R} \chi(v_{11}) \chi(v_{21}) \cdots \chi(v_{rr}) \right) \]

Finally, we rearrange the sums by collecting the similar terms. The last expression

\[ = \left( \sum_{v_{11} \in R} \sum_{v_{21} \in R} \cdots \sum_{v_{rr} \in R} \chi(v_{11}) \chi(v_{21}) \cdots \chi(v_{rr}) \right) \]

Finally, we rearrange the sums by collecting the similar terms. The last expression

\[ = \left( \sum_{v_{11} \in R} \sum_{v_{21} \in R} \cdots \sum_{v_{rr} \in R} \chi(v_{11}) \chi(v_{21}) \cdots \chi(v_{rr}) \right) \]
\[
\cdots \left( \sum_{j_1=0}^{m-1} \chi(\beta_{j_1}u_{1n}) \sum_{j_2=0}^{m-1} \chi(u_{2n}\beta_{j_2}) \cdots \sum_{j_r=0}^{m-1} \chi(u_{rn}\beta_{j_r}) z_{j_1j_2\ldots j_r} \right).
\]

Using (2.16), and recalling that \(s_{i_1\ldots i_r}(u_1, \ldots, u_r)\) counts the number of columns (vertical sections) of codewords \(u_1, \ldots, u_r\) equal to \((\alpha_1, \ldots, \alpha_r)\), we get

\[
\hat{f}(\langle u_1, u_2, \ldots, u_r \rangle) = \prod_{i_1, \ldots, i_r \in Q} \left( \sum_{j_1=0}^{m-1} \chi(\beta_{j_1}\beta_{i_1}) \cdots \sum_{j_r=0}^{m-1} \chi(\beta_{j_r}\beta_{i_r}) z_{j_1\ldots j_r} \right)^{s_{i_1\ldots i_r}(u_1, \ldots, u_r)}.
\]

Thus,

\[
\hat{f}(\langle u_1, u_2, \ldots, u_r \rangle) = \prod_{i_1, \ldots, i_r \in Q} \left( \sum_{j_1=0}^{m-1} \chi(\beta_{j_1}\beta_{i_1}) \cdots \sum_{j_r=0}^{m-1} \chi(\beta_{j_r}\beta_{i_r}) z_{j_1\ldots j_r} \right)^{s_{i_1\ldots i_r}(u_1, \ldots, u_r)}.
\]

We use equation (2.18) in Lemma 2.2.1 and apply the above result to get

\[
\sum_{(u_1, \ldots, u_r) \in \mathbb{G}^+} z_{0\ldots 0}^{s_{0\ldots 0}(u_1, \ldots, u_r)} \cdots z_{i_1\ldots i_r}^{s_{i_1\ldots i_r}(u_1, \ldots, u_r)} \cdots z_{m-1\ldots m-1}^{s_{m-1\ldots m-1}(u_1, \ldots, u_r)}
\]

\[
= \frac{1}{|\mathbb{G}|} \sum_{(u_1, \ldots, u_r) \in \mathbb{G}^+} \prod_{i_1, \ldots, i_r \in Q} \left( \sum_{j_1=0}^{m-1} \chi(\beta_{j_1}\beta_{i_1}) \cdots \sum_{j_r=0}^{m-1} \chi(\beta_{j_r}\beta_{i_r}) z_{j_1\ldots j_r} \right)^{s_{i_1\ldots i_r}(u_1, \ldots, u_r)}.
\]

A special case of a generalized \(r\)-fold complete weight enumerator is obtained when \(C_1 = C_2 = \cdots = C_r = C\), where it is called the \(r\)-fold complete weight.
enumerator of a code \( C \). We will denote the \( r \)-fold complete weight enumerator of a code \( C \) by \( W_{C^r}(z_{00}...0, \cdots, z_{11...1}, \cdots, z_{m-1,m-1,...,m-1}) \). Thus.

**Theorem 2.2.2** (The \( r \)-fold weight enumerator) Let \( \chi \) be an additive character of \( R \) which is nontrivial over all subgroups of \( R \) with size \( > 1 \). The relation between an \( r \)-fold complete weight enumerator of a linear code \( C \) and its dual \( C^\perp \) is given by

\[
W_{(C^\perp)^r}(z_{00}...0, \cdots, z_{11...1}, \cdots, z_{m-1,m-1,...,m-1}) = \frac{1}{|C|^r} W_{C^r} \left( \sum_{\alpha_1=0}^{m-1} \chi(\beta_1, \beta_0) \sum_{\alpha_2=0}^{m-1} \chi(\beta_2, \beta_0) \cdots \sum_{\alpha_r=0}^{m-1} \chi(\beta_r, \beta_0) z_{11...1}, \cdots, \right.
\]

\[
\left. \sum_{\alpha_1=0}^{m-1} \chi(\beta_1, \beta_{m-1}) \sum_{\alpha_2=0}^{m-1} \chi(\beta_2, \beta_{m-1}) \cdots \sum_{\alpha_r=0}^{m-1} \chi(\beta_r, \beta_{m-1}) z_{11...1}, \cdots, \right)
\]

Recently, for \( r = 2 \) 2-fold complete weight enumerators (or simply joint weight enumerators) of codes over Frobenius rings have been considered by Dougherty in [11], and for \( r = 1 \), 1-fold complete weight enumerators (or simply complete weight enumerators of codes) over rings have been considered by Wood in [53]. In both cases relations of these weight enumerators between codes and their duals are obtained. In the above papers, the dual of a linear code has been taken as general as possible but this generalization forced the rings to be taken as Frobenius rings. In our case for \( r = 2 \) or \( r = 1 \), duality is restricted to the most applicable one with no restriction on rings. Also very recently, \( r \)-fold complete enumerators of codes over \( Z_{2k} \) (integers modulo \( 2k \)) are considered in [2] by Bannai et al. In this paper these weight enumerators are
called $r$-genus complete weight enumerators and the relation between these weight enumerators and some lattices has been investigated.

**Remark:** The generalized $r$-fold weight enumerator and consequently the MacWilliams-type identity obtained in this section can be generalized further to codes over different $r$ rings $R_1, R_2, \ldots, R_r$. 
2.3 λ-ply weight enumerator of codes

Let $R = \{\beta_0, \ldots, \beta_{m-1}\}$ be finite ring of size $m$. First, we give the definition of λ-ply weight enumerator ($\lambda \in \mathbb{N}$) of linear codes $C_1, \ldots, C_\lambda \subset R^n$, or $\mathcal{C} = C_1 \times \cdots \times C_\lambda$ in short, which is an extension to the definition given in [48] for a code over a field.

\[ W_{\mathcal{C}}^{(\lambda)}(x, y) := \sum_{(u_1, u_2, \ldots, u_\lambda) \in \mathcal{C}} x^{n - s(u_1, \ldots, u_\lambda)} y^{s(u_1, \ldots, u_\lambda)} \quad (2.23) \]

where

\[ s(u_1, \ldots, u_\lambda) := |\{i \in N | u_{ji} \neq 0 \text{ for some } j \in \{1, 2, \ldots, \lambda\}\}| \quad (2.24) \]

and $N = \{1, 2, \ldots, n\}$.

We are going to prove a MacWilliams-type identity for λ-ply weight enumerator of $\lambda$ codes over a ring. Shiromoto, [48], defined the λ-ply weight enumerator of a linear code over a finite field $F_q$ and proved algebraically the relation between the λ-ply weight enumerator of a linear code $C$ and its dual $C^\perp$. We derive a MacWilliams-type identity for λ-ply weight enumerators of $\lambda$ codes over a ring as a corollary to Theorem 2.2.1 and hence we also obtain the identity given in [48].

**Corollary 2.3.1** (The λ-ply weight enumerator of $\lambda$ codes) Let $\mathcal{C}$ be $\lambda$ codes over a finite ring $R$ of size $m$. Then,

\[ W_{\mathcal{C}^\perp}^{(\lambda)}(x, y) = \frac{1}{|\mathcal{C}|} W_{\mathcal{C}}^{(\lambda)}(x + (m^\lambda - 1)y, x - y). \quad (2.25) \]
Proof: We will apply Theorem 2.2.2 with \( r = \lambda \). We set \( z_{00 \ldots 0} = x \) and \( z_{i_1 i_2 \ldots i_\lambda} = y \) for all \((i_1, i_2, \ldots, i_\lambda) \neq (0, 0, \ldots, 0)\) in Theorem 2.2.1, equation (2.21). On comparing (2.24) and (2.16), we observe that

\[
s(u_1, u_2, \ldots, u_\lambda) = \sum_{(i_1, \ldots, i_\lambda) \neq (0, \ldots, 0)} s_{i_1 \ldots i_\lambda}(u_1, \ldots, u_\lambda). \tag{2.26}
\]

Clearly, \( s(u_1, u_2, \ldots, u_\lambda) = n - s_{00 \ldots 0}(u_1, u_2, \ldots, u_\lambda) \).

Then the left-hand side of (2.21) in Theorem 2.2.1 is equal to

\[
\sum_{(u_1, u_2, \ldots, u_\lambda) \in \mathcal{L}} x_{n-s(u_1, u_2, \ldots, u_\lambda)} y_{s(u_1, u_2, \ldots, u_\lambda)}
\]

which equals to the left-hand side of (2.25), by definition. Let \( Q = \{0, 1, \ldots, m-1\} \subset \mathbb{N} \).

The right-hand side of (2.21)

\[
\frac{1}{|\mathcal{C}|} \sum_{(u_1, u_2, \ldots, u_\lambda) \in \mathcal{L}} \prod_{(i_1, \ldots, i_\lambda) \in Q} \left( \sum_{j_1=0}^{m-1} \chi(\beta_{j_1} \beta_{i_1}) \cdots \sum_{j_\lambda=0}^{m-1} \chi(\beta_{j_\lambda} \beta_{i_\lambda}) z_{j_1 \ldots j_\lambda} \right)
\]

\[
= \frac{1}{|\mathcal{C}|} \sum_{(u_1, \ldots, u_\lambda) \in \mathcal{L}} \left( \sum_{j_1=0}^{m-1} \chi(\beta_{j_1} \beta_0) \cdots \sum_{j_\lambda=0}^{m-1} \chi(\beta_{j_\lambda} \beta_0) z_{j_1 \ldots j_\lambda} \right) \prod_{(i_1, \ldots, i_\lambda) \in Q^{\lambda}} \left( \sum_{j_1=0}^{m-1} \chi(\beta_{j_1} \beta_{i_1}) \cdots \sum_{j_\lambda=0}^{m-1} \chi(\beta_{j_\lambda} \beta_{i_\lambda}) z_{j_1 \ldots j_\lambda} \right)
\]

where \( Q^{\lambda} = Q^{\lambda} \setminus (0, \ldots, 0) \) and \( Q^{\lambda} = Q \times \cdots \times Q \), \( \lambda \) times.

51
Hence, \((i_1', i_2', \ldots, i_{\lambda}' ) \in \{ (i_1, i_2, \ldots, i_{\lambda}) | (i_1, i_2, \ldots, i_{\lambda}) \neq (0, 0, \ldots, 0) \) where 0 \leq i_s \leq m for 1 \leq s \leq \lambda \}. Since \(d_0 = 0\) then, \(\chi(0 \cdot \lambda_{j_s}) = 1\) for 1 \leq s \leq \lambda. The last expression

\[
\frac{1}{|G|} \sum_{(u_1, u_2, \ldots, u_{\lambda}) \in G} \left( \sum_{j_1=0}^{m-1} \sum_{j_2=0}^{m-1} \cdots \sum_{j_{\lambda}=0}^{m-1} z_{j_1 j_2 \ldots j_{\lambda}} \right)^{s_{00...0}(u_1, u_2, \ldots, u_{\lambda})} \prod_{(i_1', \ldots, i'_{\lambda}) \in P_{\lambda}} \left( \sum_{j_1=0}^{m-1} \chi(\beta_{j_1} \beta'_{i_1'}) \cdots \sum_{j_{\lambda}=0}^{m-1} \chi(\beta_{j_{\lambda}} \beta'_{i_{\lambda}'}) z_{j_1 \ldots j_{\lambda}} \right)^{s_{i_1' \ldots i'_{\lambda}}(u_1, \ldots, u_{\lambda})}.
\]

(2.27)

First we are going to compute \(D\) and then \(E\):

\[
D = z_{00...0} + \sum_{(j_1', j_2', \ldots, j'_{\lambda}) \in P_{\lambda}} z_{j_1' j_2' \ldots j'_{\lambda}}
\]

where again the \(j'_s\) have the same meaning as above. Now we substitute \(z_{00...0} = x\) and \(z_{j_1 j_2 \ldots j_{\lambda}} = y\) for all \((j_1, j_2, \ldots, j_{\lambda}) \neq (0, 0, \ldots, 0)\) and we have,

\[
D = x + \sum_{(j_1', j_2', \ldots, j'_{\lambda}) \in P_{\lambda}} y = x + (m_{\lambda} - 1)y \quad \text{and,}
\]

52
\[ E = z_{0 \ldots 0} + \sum_{(j_1', j_2', \ldots, j_\lambda') \in \mathbb{Q}^\lambda} \chi(\beta_{j_1}, \beta_{j_1'}) \chi(\beta_{j_2}, \beta_{j_2'}) \cdots \chi(\beta_{j_\lambda}, \beta_{j_\lambda'}) z_{j_1' j_2' \ldots j_\lambda'} \]
\[ = x + \sum_{(j_1', j_2', \ldots, j_\lambda') \in \mathbb{Q}^\lambda} \chi(\beta_{j_1}, \beta_{j_1'}) \chi(\beta_{j_2}, \beta_{j_2'}) \cdots \chi(\beta_{j_\lambda}, \beta_{j_\lambda'}) y \]
\[ = x - y + \sum_{(j_1, j_2, \ldots, j_\lambda) \in \mathbb{Q}^\lambda} \chi(\beta_{j_1}, \beta_{j_1}) \chi(\beta_{j_2}, \beta_{j_2}) \cdots \chi(\beta_{j_\lambda}, \beta_{j_\lambda}) y. \]

Since \((i_1', i_2', \ldots, i_\lambda') \neq (0, 0, \ldots, 0)\), there exists some \(i_h' \neq 0\) which implies that \(\sum_{j_h=0}^{m-1} \chi(\beta_{j_h}, \beta_{i_h'}) = 0\) by (2.7) and hence.

\[ E = x - y + \left( \sum_{(j_1', \ldots, j_{h-1}' j_{h+1}' \ldots j_\lambda') \in \mathbb{Q}^{\lambda-1}} \chi(\beta_{j_1}, \beta_{i_1'}) \cdots \chi(\beta_{j_{h-1}}, \beta_{i_{h-1}'}) \chi(\beta_{j_{h+1}}, \beta_{i_{h+1}'}) \cdots \chi(\beta_{j_\lambda}, \beta_{i_\lambda'}) \right) \sum_{j_h=0}^{m-1} \chi(\beta_{j_h}, \beta_{i_h'}) y \]
\[ = x - y. \]

Now if we put the above expressions for \(D\) and \(E\) into the equation (2.27), the right-hand side of (2.22).

\[ \frac{1}{|C|} \sum_{(u_1, u_2, \ldots, u_\lambda) \in \mathbb{C}} \prod_{i_1, i_2, \ldots, i_\lambda} (x + f(u_1, u_2, \ldots, u_\lambda)) \]
\[ = \frac{1}{|C|} \sum_{(u_1, u_2, \ldots, u_\lambda) \in \mathbb{C}} (x + (m^\lambda - 1) y)^{z_{00 \ldots 0}} \cdot f(u_1, u_2, \ldots, u_\lambda) \]
\[ \prod_{(i_1', i_2', \ldots, i_\lambda')} (x - y)^{s_{i_1' i_2' \ldots i_\lambda'}} \]
\[ = \frac{1}{|C|} \sum_{(u_1, u_2, \ldots, u_\lambda) \in \mathbb{C}} (x + (m^\lambda - 1) y)^{z_{00 \ldots 0}} \cdot f(u_1, u_2, \ldots, u_\lambda) \cdot (x - y)^{\sum (s_{i_1' i_2' \ldots i_\lambda'})} \]

53
Since \( s(u_1, u_2, \ldots, u_\lambda) = \sum_{(i_1, i_2, \ldots, i_\lambda)} s_{i_1 i_2 \ldots i_\lambda}(u_1, u_2, \ldots, u_\lambda) \), by 2.26,
where \((i_1, \ldots, i_\lambda) \neq (0, \ldots, 0)\) and \(0 \leq i_j \leq m - 1\) for \(1 \leq j \leq \lambda\), and \(n - s(u_1, u_2, \ldots, u_\lambda) = s_{00\ldots0}(u_1, u_2, \ldots, u_\lambda)\). After simplification, the last expression

\[
= \frac{1}{|C|} \sum_{(u_1, u_2, \ldots, u_\lambda) \in C} (x + (m^\lambda - 1)y)^{n - s_{(u_1, u_2, \ldots, u_\lambda)}}(x - y)^{s_{(u_1, u_2, \ldots, u_\lambda)}}.
\]

By comparing the left-hand side and the right-hand side, we have the result. □

Let \( C^\lambda \) denote \( C \times C \times \cdots \times C \). If we take \( C_1 = \cdots = C_\lambda = C \) in the previous corollary, then we obtain a MacWilliams-type identity for \( \lambda \)-ply weight enumerator of a code \( C \) over a finite ring of size \( m \).

**Corollary 2.3.2** (The \( \lambda \)-ply weight enumerator of a code) Let \( C \) be a code over a finite ring \( R \) of size \( m \). Then,

\[
W_{C^\lambda}(x, y) = \frac{1}{|C|^\lambda} W_C^{(\lambda)}(x + (m^\lambda - 1)y, x - y). \quad (2.28)
\]

Moreover, if \( R = F_q \) is a finite field with \( q \) elements then,

**Corollary 2.3.3** \([48]\)

\[
W_{C^\lambda}(x, y) = \frac{1}{|C|^\lambda} W_C^{(\lambda)}(x + (q^\lambda - 1)y, x - y). \quad (2.29)
\]

54
2.4 Generalized $r$-fold Symmetric Weight Enumerator

In this section we are going to give a similar theorem for generalized $r$-fold symmetric weight enumerators, to be defined later, and derive this theorem as a corollary to the main theorem, Theorem 2.2.1. At the same time we will also derive the MacWilliams-type identities for both the Lee weight enumerators defined by MacWilliams, Sloane and Goethals in [40], and the symmetric weight enumerators of $Z_4$ codes defined by Klemm in [32]. We begin with the definition and the theorem which relates the usual Lee weight enumerator of a code and its dual over a finite field $F_q = \{0 = \alpha_0, \alpha_1, \alpha_{-1}, \ldots, \alpha_d, \alpha_{-d}\}$, $q$ an odd prime power, where $\alpha_{-i} = -\alpha_i$ for all $1 \leq i \leq d$ and $d = \frac{1}{2}(q - 1)$. The Lee weight of a vector $v$ in $F_q^n$ is defined by

$$\text{Lee}(v) = (l_0, l_1, \ldots, l_d)$$

(2.30)

where $l_i = l_i(v)$ for all $1 \leq i \leq d$ is the number of coordinates of $v$ equal to $\alpha_i$ or $\alpha_{-i}$, i.e. if $v = (v_1, v_2, \ldots, v_n)$, then

$$l_i = |\{j|v_j = \alpha_i \text{ or } v_j = \alpha_{-i}\}|.$$  

(2.31)

If we compare this definition with the definition of $\text{comp}(v)$ in the complete weight enumerator case, then we see that

$$l_0(v) = s_0(v)$$

and

$$l_i(v) = s_i(v) + s_{-i}(v) \text{ for } i = 1, 2, \ldots, d.$$  

The Lee weight enumerator of a code $C$ is defined by

$$W^L_C(z_0, z_1, \ldots, z_d) = \sum_{v \in C} z_0^{l_0(v)} z_1^{l_1(v)} \cdots z_d^{l_d(v)}.$$  

(2.32)
and the relation between the Lee weight enumerators of $C$ and its dual $C^\perp$ will be obtained later as a corollary to our generalized version.

The symmetric weight enumerator of a code $C \subseteq \mathbb{Z}_4^n$ is defined as follows:

$$swe_C(z_0, z_1, z_2) = \sum_{u \in C} z_0^{n_0(u)} z_1^{n_1(u)} z_2^{n_2(u)} \quad (2.33)$$

where $u = (u_1, u_2, \ldots, u_n)$,

$$n_i(u) = |\{j | u_j = \pm i, 1 \leq j \leq n\}|,$$

and $i \in \mathbb{Z}_4$.

We next generalize the symmetric weight enumerator over an arbitrary finite ring and similar to the previous section, we introduce the generalized $r$-fold symmetric (generalization of symmetric weight enumerator for $\mathbb{Z}_4$ codes) weight enumerator.

Let $\mathbb{R}$ denote a finite ring. Since $(\mathbb{R}, +)$ is an abelian group we can order the ring elements of $\mathbb{R}$ with respect to additive inverses such that $\beta_i \neq -\beta_i$ for $1 \leq i \leq d$, and $\beta_{d+j} = -\beta_{d+j}$ for $1 \leq j \leq s$. We will write $-\beta_i$ as $\beta_{-i}$ for $1 \leq i \leq d+s$.

Let $\mathbb{R} = \{0, \beta_0, \beta_1, \beta_{-1}, \ldots, \beta_d, \beta_{-d}, \beta_{d+1}, \beta_{d+2}, \beta_{d+s}\}$ be a finite ring with $2d+s+1$ elements. Occasionally we are going to use the following notations:

$$d' := d+s, \quad (2.34)$$

and,

$$D := \{0, 1, 2, \ldots, d'\}. \quad (2.35)$$
Similar to (2.16), we define the **generalized r-fold symmetric weight** of $r$ vectors $v_1 = (v_{11}, v_{12}, \ldots, v_{1n}), \ldots, v_r = (v_{r1}, v_{r2}, \ldots, v_{rn}) \in R^n$ as follows:

$$l_{i_1i_2\ldots i_r}(v_1, v_2, \ldots, v_r) :=$$

$$\left| \{ j | v_{1j} = \epsilon \beta_{i_1}, v_{2j} = \epsilon \beta_{i_2}, \ldots, v_{rj} = \epsilon \beta_{i_r}, 1 \leq j \leq n \} \right|$$

where $\epsilon = \pm 1, v_{kj} = \epsilon \beta_{i_k}$ means $v_{kj} = \beta_{i_k}$ or $v_{kj} = -\beta_{i_k}$, and $i_k \in D, 1 \leq k \leq r$.

**Example**: Let $R = Z_4 = \{0,1,2,3\}, r = 3, n = 4$, and $u_1 = (0, 1, 1, 3), u_2 = (1, 3, 2, 1), u_3 = (2, 1, 2, 3)$, then

$$l_{111}(u_1, u_2, u_3) = \begin{cases} 
1. & \text{if } (i_1, i_2, i_3) = (0, 1, 2) \\
2. & \text{if } (i_1, i_2, i_3) = (1, 1, 1) \\
1. & \text{if } (i_1, i_2, i_3) = (1, 2, 2) \\
0, & \text{otherwise.} 
\end{cases}$$

Basically, the above definition is almost the same as the generalized $r$-fold complete weight (2.16) of $r$ vectors, except here we do not distinguish the entries if they differ by a sign. Also note that we usually use the positive indices for simplification. For instance, in the above example we used $i_1 = 1$ instead of $i_1 = 3$ or $i_1 = -1$.

In the above example we see that

$$l_{111}(u_1, u_2, u_3) = s_{131}(u_1, u_2, u_3) + s_{311}(u_1, u_2, u_3),$$

which holds in general. Hence, the following relation between the generalized $r$-fold
complete and the generalized $r$-fold symmetric weights holds:

$$l_{00\ldots 0}(v_1, v_2, \ldots, v_r) = s_{00\ldots 0}(v_1, v_2, \ldots, v_r)$$

$$(2.37)$$

$$l_{11\ldots 1r}(v_1, v_2, \ldots, v_r) = \sum_{(j_1, j_2, \ldots, j_r) \in D^r} s_{j_1j_2\ldots jr}(v_1, v_2, \ldots, v_r)$$

where the sum is over $\epsilon_i \in \{\pm 1\}, 1 \leq i \leq r$, and $0 \leq i_k \leq d$. $1 \leq k \leq r$.

The generalized $r$-fold symmetric weight enumerator

$$W_L^r(z_0, 0, \ldots, z_{112\ldots i}, \ldots, z_d^r \ldots d')$$

of codes $C$ is defined as follows

$$W_L^r(z_0, 0, \ldots, z_{112\ldots i}, \ldots, z_d^r \ldots d') = \sum_{(u_1, u_2, \ldots, u_r) \in G} z_{00\ldots 0}^{l_0}(u_1, \ldots, u_r) \cdots z_{112\ldots 1r}^{l_{112\ldots 1r}}(u_1, \ldots, u_r) \cdots z_{d^r \ldots d'}^{l_{d^r \ldots d'}}(u_1, \ldots, u_r)$$

When the underlying ring $R$ is $\mathbb{Z}$ and $r = 1$, our 1-fold symmetric weight enumerator is the symmetric weight enumerator (2.33) introduced in [32]. On the other hand when the underlying ring is a field and $r = 1$ our 1-fold symmetric weight enumerator is the Lee weight enumerator (2.32) introduced in [40]. The generalized $r$-fold symmetric weight enumerator will be called the generalized $r$-fold Lee weight enumerator if the underlying ring is a field. We will use the following notation:

$$\chi'(\alpha_i \beta) := \begin{cases} 
\chi(\alpha_i \beta) + \chi(-\alpha_i \beta), & 1 \leq i \leq d, \\
\chi(\alpha_i \beta), & d + 1 \leq i \leq d + s 
\end{cases}$$

(2.38)

where $\alpha, \beta \in R$.

In order to prove the next theorem we will use the following auxiliary lemmas:

**Lemma 2.4.1** Let $R = \{0 = \beta_0, \beta_1, \beta_1, \ldots, \beta_d, \beta_{d+1}, \beta_{d+2}, \beta_{d+s}\}$, and let $\chi$ be an additive character of $R$. Here we denote $-\beta_j$ as $\beta_{-j}$. Let $\beta_1, \ldots, \beta_{r-1}, \beta_r$ be fixed
elements of R. Then with the specialization \( z_{j_1 j_2 \ldots j_r} = z_{\varepsilon_{j_1} \varepsilon_{j_2} \ldots \varepsilon_{j_r}} \) where \( \varepsilon_s = \pm 1 \) for all \( 1 \leq s \leq n \) the following holds:

\[
\sum_{j_1, j_2, \ldots, j_r = 1}^{2d + s} \chi(\beta_{j_1}, \beta_{j_1}) \chi(\beta_{j_2}, \beta_{j_2}) \cdots \chi(\beta_{j_r}, \beta_{j_r}) z_{j_1 j_2 \ldots j_r} = \sum_{j_1, j_2, \ldots, j_r = 1}^{d + s} \chi'(\beta_{j_1}, \beta_{j_1}) \chi'(\beta_{j_2}, \beta_{j_2}) \cdots \chi'(\beta_{j_r}, \beta_{j_r}) z_{j_1 j_2 \ldots j_r}. \tag{2.39}
\]

**Proof:** We will prove (2.39) by induction on \( r \). If \( r = 1 \), then

\[
\sum_{j_1 = 1}^{2d + s} \chi(\beta_{j_1}, \beta_{j_1}) z_{j_1} = \sum_{j_1 = 1}^{d} \chi(\beta_{j_1}, \beta_{j_1}) z_{j_1} + \sum_{j_1 = d + 1}^{d + s} \chi(\beta_{j_1}, \beta_{j_1}) z_{j_1} = \sum_{j_1 = 1}^{d} \chi(\beta_{j_1}, \beta_{j_1}) z_{j_1} + \sum_{j_1 = d + 1}^{d + s} \chi(\beta_{j_1}, \beta_{j_1}) z_{j_1}.
\]

Since \( z_{j_1} = z_{-j_1} \), the previous expression

\[
= \sum_{j_1 = 1}^{d} \chi(\beta_{j_1}, \beta_{j_1}) z_{j_1} + \sum_{j_1 = 1}^{d} \chi(-\beta_{j_1}, \beta_{j_1}) z_{j_1} + \sum_{j_1 = d + 1}^{s} \chi(\beta_{j_1}, \beta_{j_1}) z_{j_1} = \sum_{j_1 = 1}^{d} \left( \chi(\beta_{j_1}, \beta_{j_1}) z_{j_1} + \chi(-\beta_{j_1}, \beta_{j_1}) \right) z_{j_1} + \sum_{j_1 = d + 1}^{s} \chi(\beta_{j_1}, \beta_{j_1}) z_{j_1} = \sum_{j_1 = 1}^{d + s} \chi'(\beta_{j_1}, \beta_{j_1}) \quad \text{by (2.38).}
\]
Now suppose that (2.39) holds for all \( k < r \). Then

\[
\sum_{j_1, j_2, \ldots, j_r = 1}^{2d+s} \chi(\beta_{j_1}, \beta_{i_1}) \chi(\beta_{j_2}, \beta_{i_2}) \cdots \chi(\beta_{j_r}, \beta_{i_r}) z_{j_1, j_2, \ldots, j_r} = \sum_{j_1 = 1}^{2d+s} \sum_{j_2, \ldots, j_r = 1}^{2d+s} \chi(\beta_{j_1}, \beta_{i_1}) \chi(\beta_{j_2}, \beta_{i_2}) \cdots \chi(\beta_{j_r}, \beta_{i_r}) z_{j_1, j_2, \ldots, j_r}.
\]

We expand the first sum over \( j_1 \). The last expression

\[
= \chi(\beta_1, \beta_{i_1}) \sum_{j_2, \ldots, j_r = 1}^{2d+s} \chi(\beta_{j_2}, \beta_{i_2}) \cdots \chi(\beta_{j_r}, \beta_{i_r}) z_{j_2, \ldots, j_r} + \chi(\beta_2, \beta_{i_1}) \sum_{j_2, \ldots, j_r = 1}^{2d+s} \chi(\beta_{j_2}, \beta_{i_2}) \cdots \chi(\beta_{j_r}, \beta_{i_r}) z_{j_2, \ldots, j_r} \\
+ \chi(\beta_{2d+s}, \beta_{i_1}) \sum_{j_2, \ldots, j_r = 1}^{2d+s} \chi(\beta_{j_2}, \beta_{i_2}) \cdots \chi(\beta_{j_r}, \beta_{i_r}) z_{2d+s, j_2, \ldots, j_r}.
\]

Applying the induction hypothesis to the above \( 2d + s \) sums, the last expression

\[
= \chi(\beta_1, \beta_{i_1}) \sum_{j_2, \ldots, j_r = 1}^{d+s} \chi'(\beta_{j_2}, \beta_{i_2}) \cdots \chi'(\beta_{j_r}, \beta_{i_r}) z_{j_2, \ldots, j_r} + \chi(\beta_2, \beta_{i_1}) \sum_{j_2, \ldots, j_r = 1}^{d+s} \chi'(\beta_{j_2}, \beta_{i_2}) \cdots \chi'(\beta_{j_r}, \beta_{i_r}) z_{j_2, \ldots, j_r} \\
+ \chi(\beta_{2d+s}, \beta_{i_1}) \sum_{j_2, \ldots, j_r = 1}^{d+s} \chi'(\beta_{j_2}, \beta_{i_2}) \cdots \chi'(\beta_{j_r}, \beta_{i_r}) z_{2d+s, j_2, \ldots, j_r}.
\]
Now, we collect the first terms under one sum again. The previous expression

\[
\begin{align*}
\sum_{j_1=1}^{2d+s} \chi(\beta_{j_1}, \beta_{l_1}) & \sum_{j_2=1}^{d+s} \chi'(\beta_{j_2}, \beta_{l_2}) \cdots \chi'(\beta_{j_r}, \beta_{l_r}) z_{j_1j_2 \ldots j_r} \\
= & \sum_{j_1=1}^{d} \chi(\beta_{j_1}, \beta_{l_1}) \sum_{j_2=1}^{d+s} \chi'(\beta_{j_2}, \beta_{l_2}) \cdots \chi'(\beta_{j_r}, \beta_{l_r}) z_{j_1j_2 \ldots j_r} \\
& + \sum_{j_1=-1}^{-d} \chi(\beta_{j_1}, \beta_{l_1}) \sum_{j_2=1}^{d+s} \chi'(\beta_{j_2}, \beta_{l_2}) \cdots \chi'(\beta_{j_r}, \beta_{l_r}) z_{j_1j_2 \ldots j_r} \\
& + \sum_{j_1=d+1}^{d+s} \chi(\beta_{j_1}, \beta_{l_1}) \sum_{j_2=1}^{d+s} \chi'(\beta_{j_2}, \beta_{l_2}) \cdots \chi'(\beta_{j_r}, \beta_{l_r}) z_{j_1j_2 \ldots j_r} \\
= & \sum_{j_1=1}^{d} \chi(\beta_{j_1}, \beta_{l_1}) \sum_{j_2=1}^{d+s} \chi'(\beta_{j_2}, \beta_{l_2}) \cdots \chi'(\beta_{j_r}, \beta_{l_r}) z_{j_1j_2 \ldots j_r} \\
& + \sum_{j_1=1}^{d} \chi(-\beta_{j_1}, \beta_{l_1}) \sum_{j_2=1}^{d+s} \chi'(\beta_{j_2}, \beta_{l_2}) \cdots \chi'(\beta_{j_r}, \beta_{l_r}) z_{j_1j_2 \ldots j_r} \\
& + \sum_{j_1=d+1}^{d+s} \chi(\beta_{j_1}, \beta_{l_1}) \sum_{j_2=1}^{d+s} \chi'(\beta_{j_2}, \beta_{l_2}) \cdots \chi'(\beta_{j_r}, \beta_{l_r}) z_{j_1j_2 \ldots j_r} \\
= & \sum_{j_1=1}^{d} \left( \chi(\beta_{j_1}, \beta_{l_1}) + \chi(-\beta_{j_1}, \beta_{l_1}) \right) \sum_{j_2=1}^{d+s} \chi'(\beta_{j_2}, \beta_{l_2}) \cdots \chi'(\beta_{j_r}, \beta_{l_r}) z_{j_1j_2 \ldots j_r} \\
& + \sum_{j_1=d+1}^{d+s} \chi(\beta_{j_1}, \beta_{l_1}) \sum_{j_2=1}^{d+s} \chi'(\beta_{j_2}, \beta_{l_2}) \cdots \chi'(\beta_{j_r}, \beta_{l_r}) z_{j_1j_2 \ldots j_r}
\end{align*}
\]

61
\[
\begin{align*}
&= \sum_{j_1=1}^{d+s} \chi'(\beta_{j_1}\beta_1) \sum_{j_2, \ldots, j_r=1}^{d+s} \chi'(\beta_{j_2}\beta_{i_2}) \cdots \chi'(\beta_{j_r}\beta_{i_r}) z_{j_1j_2\ldots j_r} \\
&= \sum_{j_1, j_2, \ldots, j_r=1}^{d+s} \chi'(\beta_{j_1}\beta_1)\chi'(\beta_{j_2}\beta_{i_2}) \cdots \chi'(\beta_{j_r}\beta_{i_r}) z_{j_1j_2\ldots j_r}.
\end{align*}
\]

Also here, for convenience, we define
\[
\theta((\beta_j, \beta)) := \begin{cases} 
1, & \text{if } j = 0 \\
\chi'(\beta_j), & \text{if } j \neq 0.
\end{cases} \tag{2.41}
\]

where \( \beta_j, \beta \in R \) for all \( 0 \leq j \leq 2d + s \).

**Lemma 2.4.2** Let \( \beta_{i_1}, \beta_{i_2}, \ldots, \beta_{i_r} \in R \) be fixed. If we assume the same notation as in the previous lemma, then
\[
\sum_{j_1, j_2, \ldots, j_r=0}^{2d+s} \chi(\beta_{j_1}\beta_{i_1})\chi(\beta_{j_2}\beta_{i_2}) \cdots \chi(\beta_{j_r}\beta_{i_r}) z_{j_1j_2\ldots j_r} = \sum_{j_1, j_2, \ldots, j_r=0}^{d+s} \theta((\beta_{j_1}, \beta_{i_1}))\theta((\beta_{j_2}, \beta_{i_2})) \cdots \theta((\beta_{j_r}, \beta_{i_r})) z_{j_1\ldots j_r}. \tag{2.42}
\]

**Proof:** We will prove the identity (2.42) by applying induction on \( r \) and using the previous lemma. If \( r = 1 \), then
\[
\begin{align*}
\sum_{j_1=0}^{2d+s} \chi(\beta_{j_1}\beta_{i_1}) z_{j_1} &= z_0 + \sum_{j_1=1}^{2d+s} \chi(\beta_{j_1}\beta_{i_1}) z_{j_1} \\
&= z_0 + \sum_{j_1=1}^{d+s} \chi'(\beta_{j_1}\beta_{i_1}) z_{j_1} \quad \text{(by Lemma 2.4.1)} \\
&= \sum_{j_1=0}^{d} \theta((\beta_{j_1}, \beta_{i_1})) z_{j_1} \quad \text{(by (2.41)).}
\end{align*}
\]
Now we assume that the equation (2.42) holds for all \( k < r \). Then

\[
\sum_{j_1, j_2, \ldots, j_r = 0}^{2d+s} \chi(\beta_{j_1}, \beta_{j_1}) \chi(\beta_{j_2}, \beta_{j_2}) \cdots \chi(\beta_{j_r}, \beta_{j_r}) z_{j_1, j_2, \ldots, j_r} = \sum_{j_1 = 0}^{2d+s} \chi(\beta_{j_1}, \beta_{j_1}) \sum_{j_2, \ldots, j_r = 0}^{2d+s} \chi(\beta_{j_2}, \beta_{j_2}) \cdots \chi(\beta_{j_r}, \beta_{j_r}) z_{j_1, j_2, \ldots, j_r}.
\]

We expand the first sum over \( j_1 \), and get the last expression

\[
= \chi(\beta_0, \beta_1) \sum_{j_2, \ldots, j_r = 0}^{2d+s} \chi(\beta_{j_2}, \beta_{j_2}) \cdots \chi(\beta_{j_r}, \beta_{j_r}) z_{0 j_2 \ldots j_r} + \chi(\beta_1, \beta_1) \sum_{j_2, \ldots, j_r = 0}^{2d+s} \chi(\beta_{j_2}, \beta_{j_2}) \cdots \chi(\beta_{j_r}, \beta_{j_r}) z_{1 j_2 \ldots j_r} + \cdots + \chi(\beta_{2d+s}, \beta_1) \sum_{j_2, \ldots, j_r = 0}^{2d+s} \chi(\beta_{j_2}, \beta_{j_2}) \cdots \chi(\beta_{j_r}, \beta_{j_r}) z_{2d+s j_2 \ldots j_r}.
\]

We apply the induction hypothesis to the above \( 2d + s + 1 \) sums successively, and get that the previous expression

\[
= \chi(\beta_0, \beta_1) \left( \sum_{j_2, \ldots, j_r = 0}^{d+s} \theta((\beta_{j_2}, \beta_{j_2})) \cdots \theta((\beta_{j_r}, \beta_{j_r})) z_{0 j_2 \ldots j_r} \right) + \chi(\beta_1, \beta_1) \left( \sum_{j_2, \ldots, j_r = 0}^{d+s} \theta((\beta_{j_2}, \beta_{j_2})) \cdots \theta((\beta_{j_r}, \beta_{j_r})) z_{1 j_2 \ldots j_r} \right) + \cdots + \chi(\beta_{2d+s}, \beta_1) \left( \sum_{j_2, \ldots, j_r = 0}^{d+s} \theta((\beta_{j_2}, \beta_{j_2})) \cdots \theta((\beta_{j_r}, \beta_{j_r})) z_{2d+s j_2 \ldots j_r} \right).
\]

63
Next, we collect the first terms under one sum. The last

\[ = \sum_{j_1=0}^{2d+s} \chi(\beta_{j_1} \beta_{1_1}) \sum_{j_2, \ldots, j_r=0}^{d+s} \theta((\beta_{j_2}, \beta_{j_2})) \cdots \theta((\beta_{j_r}, \beta_{j_r})) z_{j_1, j_2, \ldots, j_r}. \]

Similar to case \( r = 1 \), the last expression

\[ = \sum_{j_1=0}^{d+s} \chi(\beta_{0} \beta_{1_1}) \sum_{j_2, \ldots, j_r=0}^{d+s} \theta((\beta_{j_2}, \beta_{j_2})) \cdots \theta((\beta_{j_r}, \beta_{j_r})) z_{0, j_2, \ldots, j_r}, \]

\[ + \sum_{j_1=1}^{d} \chi(\beta_{j_1} \beta_{1_1}) \sum_{j_2, \ldots, j_r=0}^{d+s} \theta((\beta_{j_2}, \beta_{j_2})) \cdots \theta((\beta_{j_r}, \beta_{j_r})) z_{j_1, j_2, \ldots, j_r}, \]

\[ + \sum_{j_1=1}^{d} \chi(- \beta_{j_1} \beta_{1_1}) \sum_{j_2, \ldots, j_r=0}^{d+s} \theta((\beta_{j_2}, \beta_{j_2})) \cdots \theta((\beta_{j_r}, \beta_{j_r})) z_{j_1, j_2, \ldots, j_r}, \]

\[ + \sum_{j_1=2d+1}^{2d+s} \chi(\beta_{j_1} \beta_{1_1}) \sum_{j_2, \ldots, j_r=0}^{d+s} \theta((\beta_{j_2}, \beta_{j_2})) \cdots \theta((\beta_{j_r}, \beta_{j_r})) z_{j_1, j_2, \ldots, j_r}. \]

We collect the sums by using the definition of \( \theta \) given in (2.41). Finally, the previous expression

\[ = \sum_{j_1, j_2, \ldots, j_r=0}^{d+s} \theta((\beta_{j_1}, \beta_{1_1})) \theta((\beta_{j_2}, \beta_{j_2})) \cdots \theta((\beta_{j_r}, \beta_{j_r})) z_{j_1, j_2, \ldots, j_r}. \]

\[ \square \]

**Theorem 2.4.1** (The generalized \( r \)-fold symmetric weight enumerator) Let \( R \) be a finite ring, and \( \chi \) be a nontrivial, over all subgroups with size \( > 1 \), additive character of \( R \). We use the notation of the previous theorem. Then the relation between the generalized \( r \)-fold symmetric weight enumerator of codes \( C \) and their duals \( C^* \) is given by

\[ W_{C^*}(z_{00}, \ldots, z_{11}, \ldots, z_{d}, \ldots d') = \frac{1}{|C|} W_C^{*}(\sum_{j_1, j_2, \ldots, j_r=0}^{d+s} \theta((\beta_{j_1}, \beta_{0})) \theta((\beta_{j_2}, \beta_{0})) \cdots \theta((\beta_{j_r}, \beta_{0})) z_{j_1, j_2, \ldots, j_r}, \]

64
\[ \ldots, \sum_{j_1, j_2, \ldots, j_r = 0}^{d+s} \theta((\beta_{j_1}, \beta_{i_1})) \theta((\beta_{j_2}, \beta_{i_2})) \cdots \theta((\beta_{j_r}, \beta_{i_r})) z_{j_1, j_2, \ldots, j_r}, \ldots, \tag{2.43} \]

\[ \ldots, \sum_{j_1, j_2, \ldots, j_r = 0}^{d+s} \theta((\beta_{j_1}, \beta_{i_1})) \theta((\beta_{j_2}, \beta_{i_2})) \cdots \theta((\beta_{j_r}, \beta_{i_r})) z_{j_1, j_2, \ldots, j_r} \]

where \((i_1, i_2, \ldots, i_r) \in D^r\) and \(D = \{0, 1, 2, \ldots, d' = d + s\}\).

**Proof of the Theorem**: We use Theorem 2.2.1. In equation (2.21) we specialize \(z_{t_1 t_2 \ldots t_r} = z_{t_1 t_2 t_3 \ldots}, \) where \(\epsilon_s = \pm 1\) for all \(1 \leq s \leq r\). The left-hand side of (2.21) because of specialization of variables becomes:

\[ \sum_{(u_1, \ldots, u_r) \in \mathbb{C}^\perp} z_{0 \ldots 0} (u_1 \ldots u_r) \cdots z_{t_1 \ldots t_r} (u_1 \ldots u_r) \cdots z_{d' \ldots d'} (u_1 \ldots u_r) \]

where the sums in the exponents are over all \(\epsilon_j \in \{\pm 1\}, 1 \leq j \leq r\), and using (2.37) it reduces to

\[ \sum_{(u_1, u_2, \ldots, u_r) \in \mathbb{C}^\perp} z_{0 \ldots 0} (u_1 \ldots u_r) \cdots z_{t_1 t_2 \ldots t_r} (u_1 \ldots u_r) \cdots z_{d' \ldots d'} (u_1 \ldots u_r) \]

which is the generalized \(r\)-fold symmetric weight enumerator of \(\mathbb{C}^\perp\).

Now, we recall that the right-hand side of (2.21) is equal to

\[ \frac{1}{|\mathbb{C}|} \sum_{(u_1, \ldots, u_r) \in \mathbb{C}^{t_1 \ldots t_r}} \prod_{j_1, \ldots, j_r \in Q} \left( \sum_{j_1, \ldots, j_r = 0}^{2d+s} \chi((\beta_{j_1}, \beta_{i_1}) \cdots \chi((\beta_{j_r}, \beta_{i_r}), z_{j_1 \ldots j_r}) \right)^{s_{t_1 \ldots t_r} (u_1 \ldots u_r)} \]

Here \(Q = \{0, \pm 1, \ldots, \pm d, d + 1, \ldots, d + s\}\). We apply the equation (2.42) to the inner sum of the last term. Thus, the last term

\[ \frac{1}{|\mathbb{C}|} \sum_{(u_1, \ldots, u_r) \in \mathbb{C}^{t_1 \ldots t_r}} \prod_{j_1, \ldots, j_r \in Q} \left( \sum_{j_1, j_2, \ldots, j_r = 0}^{d+s} \theta((\beta_{j_1}, \beta_{i_1})) \theta((\beta_{j_2}, \beta_{i_2})) \cdots \theta((\beta_{j_r}, \beta_{i_r})) z_{j_1 \ldots j_r} \right)^{s_{t_1 \ldots t_r} (u_1 \ldots u_r)} \]
Finally, we observe that for any index \((e_1 i_1, e_2 i_2, \ldots, e_r i_r)\), the inner sums have the same value, i.e.,

\[
\sum_{j_1, j_2, \ldots, j_r = 0}^{d+s} \theta((\beta_{j_1}, \beta_{i_1}))\theta((\beta_{j_2}, \beta_{i_2})) \cdots \theta((\beta_{j_r}, \beta_{i_r})) z_{j_1 \cdots j_r} = \sum_{j_1, j_2, \ldots, j_r = 0}^{d+s} \theta((\beta_{j_1}, \beta_{i_1}))\theta((\beta_{j_2}, \beta_{i_2})) \cdots \theta((\beta_{j_r}, \beta_{i_r})) z_{j_1 \cdots j_r}
\]

since \(\theta((\alpha, J)) = \theta((-\alpha, J))\) by definition (2.41).

Hence, we can collect the similar terms and get that the right-hand side of (2.21)

\[
\frac{1}{|C|} \sum_{(u_1, \ldots, u_r) \in C^*_{1, \ldots, r}} \prod (\sum_{j_1, j_2, \ldots, j_r = 0}^{d+s} \theta((\beta_{j_1}, \beta_{i_1}))\theta((\beta_{j_2}, \beta_{i_2})) \cdots \theta((\beta_{j_r}, \beta_{i_r})) z_{j_1 \cdots j_r})
\]

(2.44)

Thus, by applying (2.37) to the exponent, we have

\[
\sum_{(i_1, i_2, \ldots, i_r) \in D^r} s_{e_{1}, e_{2}, \ldots, e_{r}}(u_1, \ldots, u_r) = l_{i_1 \ldots i_r}(u_1, \ldots, u_r).
\]

Therefore.

\[
(2.44) = \frac{1}{|C|} \sum_{(u_1, \ldots, u_r) \in C^*_{1, \ldots, r}} \prod (\sum_{j_1, j_2, \ldots, j_r = 0}^{d+s} \theta((\beta_{j_1}, \beta_{i_1}))\theta((\beta_{j_2}, \beta_{i_2})) \cdots \theta((\beta_{j_r}, \beta_{i_r})) z_{j_1 \cdots j_r})^{l_{i_1 \ldots i_r}(u_1, \ldots, u_r)}
\]

which is the right-hand side of the theorem. The equation (2.21) obtained in Theorem 2.2.1 implies the result. □
We can define an $r$-fold symmetric weight enumerator of a code $C$ by letting $C_1 = C_2 = \cdots = C_r = C$ in the generalized $r$-fold symmetric weight enumerator of $C$. Hence, we obtain the following theorem:

**Theorem 2.4.2** (The $r$-fold symmetric weight enumerator) Let $R$ be a finite ring, and $\chi$ be a nontrivial, over all subgroups with size $>1$, additive character of $R$. We use the notation of the last theorem. Then the relation between the $r$-fold symmetric weight enumerator of code $C$ and its dual $C^\perp$ is given by

$$W_{C^\perp}^r(z_0, \ldots, z_{i_1 j_1}, \ldots, z_{i_r j_r})$$

$$= \frac{1}{|C|^r} \sum_{d+3} \theta((\beta_{j_1}, \beta_0)) \theta((\beta_{j_2}, \beta_0)) \cdots \theta((\beta_{j_r}, \beta_0)) z_{j_1 j_2 \cdots j_r},$$

$$\ldots, \sum_{d+3} \theta((\beta_{j_1}, \beta_{i_2})) \theta((\beta_{j_2}, \beta_{i_2})) \cdots \theta((\beta_{j_r}, \beta_{i_2})) z_{j_1 j_2 \cdots j_r}, \ldots$$

$$\ldots, \sum_{d+3} \theta((\beta_{j_1}, \beta_{i_r})) \theta((\beta_{j_2}, \beta_{i_r})) \cdots \theta((\beta_{j_r}, \beta_{i_r})) z_{j_1 j_2 \cdots j_r}.$$  \hspace{1cm} (2.45)

where $(i_1, i_2, \ldots, i_r) \in D^r$ and, $D = \{0, 1, 2, \ldots, d' = d + s\}$.

The relation between the well-known Lee weight enumerator of a code $C$ over a finite field with odd cardinality and its dual is also is obtained in the corollary below.

**Corollary 2.4.1** [38] (Lee weight enumerator) Let $F_q = \{0 = \alpha_0, \alpha_1, \alpha_{-1}, \ldots, \alpha_d, \alpha_{-d}\}$, $q$ is an odd prime, be a finite field and $\chi$ be a nontrivial additive character of $F_q$. Then
the relation between the Lee weight enumerator of a code $C$ and its dual $C^\perp$ is given by

$$W_L^{C^\perp}(z_0, z_1, \ldots, z_d) =$$

$$\frac{1}{|C|}W_L^C(z_0 + \sum_{i=1}^d (\chi(\alpha_i \alpha_0) + \chi(-\alpha_i \alpha_0))z_i, z_0 + \sum_{i=1}^d (\chi(\alpha_i \alpha_1) + \chi(-\alpha_i \alpha_1))z_i, \ldots, z_0 + \sum_{i=1}^d (\chi(\alpha_i \alpha_d) + \chi(-\alpha_i \alpha_d))z_i).$$

(2.46)

Proof: Let $r = 1$ in the previous corollary, in equation (2.45). □

If $r = 2$, then by Theorem 2.4.2, the relation between 2-fold symmetric weight enumerators, “symmetric biweight enumerators”, of $C$ and $C^\perp$ is as follows:

**Corollary 2.4.2 (The 2-fold symmetric weight enumerator)**

Let $R = \{0 = J_0, J_1, J_{-1}, \ldots, J_d, J_{-d}, J_{d+1}, \ldots, J_{2d+1}\}$ be a finite ring and $\chi$ be a nontrivial additive character over all subgroups of $R$ of size $> 1$. Then the relation between the 2-fold symmetric weight enumerator of a code $C$ and its dual $C^\perp$ is given by

$$W_L^{(C^\perp)^{2}}(z_{00}, z_{11}, \ldots, z_{d'd'})$$

$$= \frac{1}{|C|^2}W_L^C\left(\sum_{j_1,j_2=0}^d \theta((\beta_{j_1}, J_0))\theta((\beta_{j_2}, J_0))z_{j_1j_2}, \ldots, \sum_{j_1,j_2=0}^d \theta((\beta_{j_1}, J_{i}))\theta((\beta_{j_2}, J_{i}))z_{j_1j_2}, \ldots, \sum_{j_1,j_2=0}^d \theta((\beta_{j_1}, J_{d'})\theta((\beta_{j_2}, J_{d'}))z_{j_1j_2}\right).$$

(2.47)
Corollary 2.4.3 (Symmetric weight enumerator) The relation between symmetric weight enumerator of a code $C$ and its dual $C^\perp$ over $Z_4$ is given by

$$sw_{w,C}(z_0, z_1, z_2) = \frac{1}{|C|}sw_{w,C}(z_0 + 2z_1 + z_2, z_0 - z_2, z_0 - 2z_1 + z_2). \quad (2.48)$$

Proof: Let $r = 1$ in Theorem 2.4.2 and $R = Z_4 = \{0, 1, 2, 3\}$. Here $d = 1$ and $s = 1$. Then,

$$W_{w,C^\perp}^r(z_0, z_1, z_2)$$

$$= \frac{1}{|C|}W_{w,C}^r\left( \sum_{j_1=0}^{3} \theta((,j_1, 0))z_{j_1}) \right) \cdot \sum_{j_1=0}^{3} \theta((,j_1, 1))z_{j_1} \cdot \sum_{j_1=0}^{3} \theta((,j_1, 2))z_{j_1}.$$

We take $\chi(j) = i^j$ where $i^2 = -1$ and $j \in Z_4$. Then,

$$\sum_{j_1=0}^{3} \theta((,j_1, 0))z_{j_1} = z_0 + (\chi(0) + \chi(0))z_1 + \chi'(0)z_2 \quad \text{(by (2.41) and (2.38))}$$

$$= z_0 + 2z_1 + z_2.$$  

$$\sum_{j_1=0}^{3} \theta((,j_1, 1))z_{j_1} = z_0 + (\chi(1) + \chi(-1))z_1 + \chi'(2)z_2 \quad \text{(by (2.41) and (2.38))}$$

$$= z_0 + (i + i^{-1})z_1 + i^2z_2 = z_0 - z_2.$$  

$$\sum_{j_1=0}^{3} \theta((,j_1, 2))z_{j_1} = z_0 + \chi'(2)z_1 + \chi'(0)z_2 \quad \text{(by (2.41) and (2.38))}$$

$$= z_0 + (\chi(2) + \chi(2))z_1 + \chi(0)z_2 = z_0 - 2z_1 + z_2,$$

which is the result. □
2.5 Generalized $r$-fold Hamming Weight Enumerator

In this section we prove another generalized MacWilliams-type identity and derive both the usual Hamming weight and biweight enumerators of a binary linear code as corollaries.

We change the notation to a more suitable one as follows:

$$s_r(\mathbf{i}, \mathbf{u}) := s_{u_1\ldots u_r}(u_1, \ldots, u_r)$$

where $\mathbf{i} = (i_1, \ldots, i_r), \mathbf{u} = (u_1, \ldots, u_r)$ and $i_k \in Q = \{0, 1, 2, \ldots, m - 1\}$ for all $k \in \{1, 2, \ldots, r\}$ and $u_i \in C, 1 \leq i \leq r$.

We define the maps

$$\nu : Q \to \{0, 1\} \quad \text{and} \quad \nu^r : Q^r \to \{0, 1\}^r$$

where

$$\nu(i_k) = \begin{cases} 0, & \text{if } i_k = 0 \\ 1, & \text{if } i_k \neq 0 \end{cases} \quad (2.49)$$

and

$$\nu^r(\mathbf{i}) = (\nu(i_1), \nu(i_2), \ldots, \nu(i_r)). \quad (2.50)$$

We let $\mathbf{I} = (l_1, l_2, \ldots, l_r) \in \{0, 1\}^r$ and define

$$L(\mathbf{I}) := \{\mathbf{j} = (j_1, \ldots, j_r) \in Q^r | \nu^r(\mathbf{j}) = \mathbf{I}\}. \quad (2.51)$$

and

$$h_r(\mathbf{i}; \mathbf{u}) = h_{u_1\ldots u_r}(u_1, \ldots, u_r) = \sum_{\mathbf{j} \in L(\mathbf{I})} s_r(\mathbf{j}; \mathbf{u}). \quad (2.52)$$
Note that if \( \mathbf{i}, \mathbf{j} \in L(l) \), then \( \nu^r(\mathbf{i}) = \nu^r(\mathbf{j}) \). To clarify the definitions above, we give an example:

**Example:** Let \( r = 3, n = 5 \) and \( \mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3 \in C \subset F_3^n \), where \( F_3 = \{J_0, J_1, J_2\} \).

Let \( \mathbf{u}_1 = (J_1, J_2, J_0, J_2, J_2), \mathbf{u}_2 = (J_2, J_1, J_0, J_1, J_1) \) and \( \mathbf{u}_3 = (J_0, J_1, J_2, J_2, J_0) \). Then, by definition, \( Q = \{0, 1, 2\} \) and

\[
L((1, 1, 0)) = \{(1, 2, 0), (2, 1, 0), (1, 1, 0), (2, 2, 0)\},
\]

\[
L((0, 0, 1)) = \{(0, 0, 1), (0, 0, 2)\}.
\]

\[
L((1, 1, 1)) = \{(1, 1, 1), (2, 1, 1), (1, 2, 1), (2, 2, 1), (1, 1, 2), (2, 1, 2), (2, 2, 2)\}.
\]

In other words, the set \( L(l) \) is the set of all possible \( r \)-tuples \( j \in Q^r \) where the zero places of \( j \) coincide with the zero places of \( l \).

Further, we observe that

\[
s_{j_1j_2j_3}(\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3) = \begin{cases} 
1, & \text{if } (j_1, j_2, j_3) = (1, 2, 0) \\
1, & \text{if } (j_1, j_2, j_3) = (2, 1, 0) \\
1, & \text{if } (j_1, j_2, j_3) = (2, 2, 1) \\
1, & \text{if } (j_1, j_2, j_3) = (2, 1, 1) \\
1, & \text{if } (j_1, j_2, j_3) = (0, 0, 2) \\
0, & \text{otherwise.}
\end{cases}
\]
Using (2.52), we have
\[ h_{110}(u_1, u_2, u_3) = \sum_{j \in L(1,1,0)} s(j; u) \]
\[ = s_{120}(u_1, u_2, u_3) + s_{210}(u_1, u_2, u_3) + s_{110}(u_1, u_2, u_3) + s_{220}(u_1, u_2, u_3) = 1 + 1 + 0 + 0 = 2. \]

Similarly, we get
\[ h_{121}(u_1, u_2, u_3) = \begin{cases} 2, & \text{if } (l_1, l_2, l_3) = (1, 1, 0) \\ 1, & \text{if } (l_1, l_2, l_3) = (0, 0, 1) \\ 2, & \text{if } (l_1, l_2, l_3) = (1, 1, 1) \\ 0, & \text{otherwise}. \end{cases} \]

We observe that
\[ \sum_{j \in \{0,1\}^r} h_r(j; u) = n. \]

For fixed \( l \in \{0,1\}^r \) and \( i \in Q^r \) we define
\[ E(l; i) = |\{ k | \nu(i_k) = 1, \nu(l_k) = 1, 1 \leq k \leq r \}| \]
\[ N(l; i) = |\{ k | \nu(i_k) = 0, \nu(l_k) = 1, 1 \leq k \leq r \}|. \]

**Example:** Let \( l = (0,1,1,1,0) \in \{0,1\}^5 \) and \( i = (2,0,1,2,0) \in Q^5 \) where \( Q = \{0,1,2\} \). Then,
\[ E(l; i) = 2 \text{ and } N(l; i) = 1. \]
In other words, first we consider the index subset of $i$ which is identified by the support set (index set of nonzero entries) of $\mathcal{L}$. Among these entries of $\mathcal{L}$, the number of zero entries is equal to $N$, and the number of the nonzero entries is equal to $E$.

We introduce $i' = (i_1, i_2, \ldots, i_{r-1})$ where $r > 1$, and $i \in Q^r$, i.e. $i'$ is the restriction of $i$ to the first $r - 1$ coordinates.

We point out the following properties which are going to be used in the proof of the next lemma. These properties follow directly from the definitions.

\[ N((\mathcal{L}; 0); i) = N(\mathcal{L}; i') \quad \text{and} \quad E((\mathcal{L}; 0); i) = E(\mathcal{L}; i) \]  \hspace{1cm} (2.54)

Also if, $i_r = 0$ then

\[ N((\mathcal{L}; 1); i) = N(\mathcal{L}; i') + 1 \quad \text{and} \quad E((\mathcal{L}; 1); i) = E(\mathcal{L}; i'). \]  \hspace{1cm} (2.55)

and if $i_r \neq 0$, then

\[ N((\mathcal{L}; 1); i) = N(\mathcal{L}; i') \quad \text{and} \quad E((\mathcal{L}; 1); i) = E(\mathcal{L}; i') + 1. \]  \hspace{1cm} (2.56)

**Lemma 2.5.1** Let $\mathcal{R} = \{0 = \beta_0, \beta_1, \ldots, \beta_{m-1}\}$ be a finite ring and $\chi$ be a character which is nontrivial over all additive subgroups of size $> 1$ of $\mathcal{R}$. Let $\beta_i, \beta_j, \ldots, \beta_r$ be fixed elements of $\mathcal{R}$ and let $i, j \in Q^r$. Then,

\[ \sum_{i \in \{0,1\}^r} \sum_{j \in L(\mathcal{L})} \chi(\beta_j, \beta_i) \cdots \chi(\beta_r, \beta_i) z_{i_1 \ldots i_r} = \sum_{i \in \{0,1\}^r} (m-1)^{N(\mathcal{L})} (-1)^{E(\mathcal{L})} z_{i_1 \ldots i_r}. \]  \hspace{1cm} (2.57)
**Proof**: We will prove the equation (2.57) by induction on \( r \). First we prove it for \( r = 1 \), so we have \( i = (i_1) \). We need to show that

\[
\sum_{l \in \{0,1\}} \sum_{j \in L(l)} \chi(j_j,j_{i_1}) z_l = \sum_{l \in \{0,1\}} (m - 1)^{N(i_1)} (-1)^{E(i_1)} z_l.
\]

Observe that \( L(0) = \emptyset \) and \( L(1) = \{1, 2, \ldots, m-1\} \). Hence the left-hand side of the above equation becomes

\[
\sum_{l=0}^{1} \sum_{j \in L(l)} \chi(j_j,j_{i_1}) z_l = \sum_{j \in L(0)} \chi(j_j,j_{i_1}) z_0 + \sum_{j \in L(1)} \chi(j_j,j_{i_1}) z_1
\]

\[
= \chi(j_0,j_{i_1}) z_0 + \sum_{j=1}^{m-1} \chi(j_j,j_{i_1}) z_1 = z_0 + \sum_{j=1}^{m-1} \chi(j_j,j_{i_1}) z_1
\]

\[
= z_0 + \begin{cases} 
(m - 1)z_1, & \text{if } i_1 = 0 \\
-z_1, & \text{if } i_1 \neq 0
\end{cases}
\]

\[
= (m - 1)^{N(0; i)} (-1)^{E(0; i)} z_0 + (m - 1)^{N(1; i)} (-1)^{E(1; i)} z_1
\]

since \( N(0; i) = 0, E(0; i) = 0 \), for \( i_1 \in \{0, 1, \ldots, m-1\} \) and \( N(1; i) = 0 \) for all \( i \neq 0 \). \( N(1; 0) = 1, E(1; 0) = 0 \), and \( E(1; i) = 1 \) for all \( i \neq 0 \). Note that the last expression is exactly the right-hand side of the identity in the lemma for \( r = 1 \).

Now, we assume that the equation (2.57) holds for all \( s < r \).
We start with the left-hand side of (2.57).

\[
\sum_{l \in \{0,1\}^*} \sum_{l \in L(l)} \chi(\beta_{j_1}, \beta_{l_1}) \cdots \chi(\beta_{j_r}, \beta_{l_r}) z_{l_{i_1} \ldots l_r}
\]

\[
= \sum_{(l', \omega) \in \{0,1\}^r} \sum_{l \in L(l', \omega)} \chi(\beta_{j_1}, \beta_{l_1}) \cdots \chi(\beta_{j_{r-1}}, \beta_{l_{r-1}}) \chi(\beta_{j_r}, \beta_{l_r}) z_{l_{i_1} \ldots l_r, 0}
\]

\[+ \sum_{(l', \omega) \in \{0,1\}^r} \sum_{l \in L(l', \omega)} \chi(\beta_{j_1}, \beta_{l_1}) \cdots \chi(\beta_{j_{r-1}}, \beta_{l_{r-1}}) \chi(\beta_{j_r}, \beta_{l_r}) z_{l_{i_1} \ldots l_r, 1}
\]

\[= \sum_{(l', \omega) \in \{0,1\}^r} \sum_{l \in L(l', \omega)} \chi(\beta_{j_1}, \beta_{l_1}) \cdots \chi(\beta_{j_{r-1}}, \beta_{l_{r-1}}) \chi(\beta_{j_r}, \beta_{l_r}) z_{l_{i_1} \ldots l_r, 0}
\]

\[+ \left( \sum_{j_r=1}^{m-1} \chi(\beta_{j_r}, \beta_{l_r}) \right) \sum_{(l', \omega) \in \{0,1\}^r} \sum_{l \in L(l', \omega)} \chi(\beta_{j_1}, \beta_{l_1}) \cdots \chi(\beta_{j_{r-1}}, \beta_{l_{r-1}}) z_{l_{i_1} \ldots l_r, 1}
\]

Viewing \(z_{l_{i_1} \ldots l_{r-1}, 0}\) (or \(z_{l_{i_1} \ldots l_{r-1}, 1}\)) as a new variable with \(r - 1\) subindices, we can apply the induction hypothesis to both sums, and the previous expression

\[= \sum_{(l', \omega) \in \{0,1\}^r} \sum_{l' \in \{0,1\}^r} (m - 1)^{N(l', \omega)} (-1)^{E(l', \omega)} z_{l_{i_1} \ldots l_{r-1}, 0}
\]

\[+ \sum_{j_r=1}^{m-1} \chi(\beta_{j_r}, \beta_{l_r}) \sum_{(l', \omega) \in \{0,1\}^r} \sum_{l' \in \{0,1\}^r} (m - 1)^{N(l', \omega)} (-1)^{E(l', \omega)} z_{l_{i_1} \ldots l_{r-1}, 1}
\]

\[= \sum_{(l', \omega) \in \{0,1\}^r} \sum_{l' \in \{0,1\}^r} (m - 1)^{N(l', \omega)} (-1)^{E(l', \omega)} z_{l_{i_1} \ldots l_{r-1}, 0}
\]

\[+ \begin{cases}
(m - 1) \sum_{l' \in \{0,1\}^r} (m - 1)^{N(l', \omega')} (-1)^{E(l', \omega')} z_{l_{i_1} \ldots l_{r-1}, 1}, & \text{if } i_r = 0

(-1) \sum_{l' \in \{0,1\}^r} (m - 1)^{N(l', \omega')} (-1)^{E(l', \omega')} z_{l_{i_1} \ldots l_{r-1}, 1}, & \text{if } i_r \neq 0.
\end{cases}
\]
Applying the properties (2.54), (2.55) and (2.56) and since \( \mathcal{N}(\mathbf{i}', 0) = \mathcal{N}(\mathbf{i}') \) and \( E(\mathbf{i}', 0) = E(\mathbf{i}') \), the last expression

\[
= \sum_{\{\mathbf{i}, \mathbf{k} \in \mathbb{Q}' \}} (m - 1)^{\mathcal{N}(\mathbf{i}; \mathbf{k})} (-1)^{E(\mathbf{i}; \mathbf{k})} z_{i_1 l_2 \ldots l_r} \quad \Box
\]

**Corollary 2.5.1** Let \( \mathbf{i}, \mathbf{k} \in \mathbb{Q}' \) satisfy \( \mathcal{N}(\mathbf{i}) = \mathcal{N}(\mathbf{k}) \). Then

\[
\sum_{\{\mathbf{i}, \mathbf{k} \in \mathbb{Q}' \}} \sum_{\mathbf{j} \in L(\mathbf{i})} \chi(\beta_{j_1}, \beta_{k_1}) \cdot \chi(\beta_{j_r}, \beta_{k_r}) z_{i_1 l_2 \ldots l_r} = \sum_{\{\mathbf{i}, \mathbf{k} \in \mathbb{Q}' \}} \sum_{\mathbf{j} \in L(\mathbf{i})} \chi(\beta_{j_1}, \beta_{k_1}) \cdot \chi(\beta_{j_r}, \beta_{k_r}) z_{i_1 l_2 \ldots l_r}. \quad (2.58)
\]

**Proof:** Since \( \mathcal{N}(\mathbf{i}) = \mathcal{N}(\mathbf{k}) \), by definition, \( \mathcal{N}(\mathbf{i}; \mathbf{k}) = \mathcal{N}(\mathbf{i}; \mathbf{k}) \) and \( E(\mathbf{i}; \mathbf{k}) = E(\mathbf{i}; \mathbf{k}) \). Hence, using the equation (2.57) in the previous lemma we have the result. \( \Box \)

We let \( \mathbf{i} = (1, 1, \ldots, 1) \) and \( \mathbf{0} = (0, 0, \ldots, 0) \) and we define the **generalized r-fold Hamming weight enumerator**, \( W^H_C(z_0 \ldots 0, \ldots, z_{l_1 l_2 \ldots l_r}) \) of a linear code \( C \) as follows:

\[
W^H_C(z_0 \ldots 0, \ldots, z_{l_1 l_2 \ldots l_r}) = \sum_{(u_1, u_2, \ldots, u_r) \in \mathbb{C}} z_{l_0}^{h_0(\mathbf{0}; \mathbf{u})} \cdot z_{l_1}^{h_r(\mathbf{1}; \mathbf{u})} \cdot \ldots \cdot z_{l_r}^{h_r(\mathbf{1}; \mathbf{u})}
\]

where \( l_i \in \{0, 1\}, 1 \leq k \leq r \).

Now we state the main theorem of this section:

**Theorem 2.5.1** (The generalized r-fold Hamming weight enumerator) Let \( C \) be a linear code of length \( n \) over a finite ring \( R \) and \( C^\perp \) its dual. Also, let the functions
\( N \) and \( E \) be defined as in equation (2.53). Then the relation between the generalized \( r \)-fold Hamming weight enumerators of \( C \) and \( C^\perp \) is as follows:

\[
W_{C^\perp}^H(z_0 \ldots 0, z_{i_1 i_2 \ldots i_r} \ldots z_{i_1 \ldots i_r}) = \frac{1}{|C|} W_C^H \left( \sum_{k \in \{0,1\}^r} (m-1)^{N(k;0)} (-1)^{E(k;0)} z_{k_1 \ldots k_r} \ldots \right)

\[
= \sum_{k \in \{0,1\}^r} (m-1)^{N(k;0)} (-1)^{E(k;0)} z_{k_1 \ldots k_r} \ldots \sum_{k \in \{0,1\}^r} (m-1)^{N(k;0)} (-1)^{E(k;0)} z_{k_1 \ldots k_r}
\]

where \( l \in \{0,1\}^r \).

**Proof**: To prove this theorem we are going to use the identity (2.21) in Theorem 2.2.1. In (2.21), we specialize \( z_{i_1 i_2 \ldots i_r} = z_{i_1 i_2 \ldots i_r} \) if and only if \( \nu^r(l) = \nu^r(l) \) where \( l = (i_1, i_2, \ldots, i_r) \in Q^r \) and \( l = (l_1, l_2, \ldots, l_r) \in \{0,1\}^r \) and \( \nu^r \) was defined at (2.50).

First we compute the left-hand side of (2.21):

\[
W_{C^\perp}^H(z_0 \ldots 0, z_{i_1 i_2 \ldots i_r} \ldots z_{m-1,m-1 \ldots m-1}) = \sum_{(u_1, \ldots, u_r) \in C} \sum_{00 \ldots 0} z_{u_1 u_2 \ldots u_r} \ldots z_{i_1 i_2 \ldots i_r} \ldots z_{m-1,m-1 \ldots m-1}
\]

\[
= \sum_{(u_1, \ldots, u_r) \in C} z_{00 \ldots 0} \cdot z_{i_1 i_2 \ldots i_r} \ldots z_{m-1,m-1 \ldots m-1} \quad \text{(by (2.51))}
\]

\[
= \sum_{(u_1, \ldots, u_r) \in C} h_r(u;0) \ldots h_r(u;0) \ldots z_{i_1 i_2 \ldots i_r} \ldots z_{m-1,m-1} \quad \text{(by (2.52))}
\]

which is the left-hand side of equation (2.59) to be proven.

The right-hand side of the identity (2.21) is

\[
\frac{1}{|C|} W_C^H \left( \sum_{j_1=0}^{m-1} \chi(\beta_{j_1} \beta_0) \sum_{j_2=0}^{m-1} \chi(\beta_{j_2} \beta_0) \cdots \sum_{j_r=0}^{m-1} \chi(\beta_{j_r} \beta_0) z_{j_1 j_2 \ldots j_r} \ldots \right)
\]

77
\[
\sum_{j_1=0}^{m-1} \chi(\beta_1, \beta_1) \sum_{j_2=0}^{m-1} \chi(\beta_2, \beta_2) \cdots \sum_{j_r=0}^{m-1} \chi(\beta_r, \beta_r) z_{j_1, j_2, \ldots, j_r},
\]
\[
\cdots \sum_{j_1=0}^{m-1} \chi(\beta_1, \beta_m) \sum_{j_2=0}^{m-1} \chi(\beta_2, \beta_m) \cdots \sum_{j_r=0}^{m-1} \chi(\beta_r, \beta_m) z_{j_1, j_2, \ldots, j_r}.
\]
\[
= \frac{1}{|\mathcal{C}|} \sum_{u \in \mathcal{C}} \prod_{i \in \mathcal{I}} \left( \sum_{j_1, \ldots, j_r=0}^{m-1} \chi(\beta_{j_1}, \beta_{j_1}) \cdots \chi(\beta_{j_r}, \beta_{j_r}) z_{j_1, j_2, \ldots, j_r} \right)^{s(\mathcal{I}; i)} \quad (2.60)
\]

First we compute \(S\) by making the substitution and using the sets \(L(\xi)\). Note that
\[
S = \sum_{\xi \in \{0, 1\}^r} \sum_{\xi \in L(\xi)} \chi(\beta_{j_1}, \beta_{j_1}) \cdots \chi(\beta_{j_r}, \beta_{j_r}) z_{j_1, j_2, \ldots, j_r}.
\]

By the identity (2.57) in Lemma 2.5.1,
\[
S = \sum_{\xi \in \{0, 1\}^r} (m - 1)^{N(\xi)} (-1)^{E(\xi)} z_{j_1, j_2, \ldots, j_r}.
\]

We take the above formula for \(S\) and substitute it back to get
\[
(2.60) = \frac{1}{|\mathcal{C}|} \sum_{u \in \mathcal{C}} \prod_{i \in \mathcal{I}} \left( \sum_{\xi \in \{0, 1\}^r} (m - 1)^{N(\xi)} (-1)^{E(\xi)} z_{j_1, j_2, \ldots, j_r} \right)^{s(\mathcal{I}; i)} \quad (2.60)
\]

Finally we use the identity (2.58) in Corollary 2.5.1 to get,
\[
(2.60) = \frac{1}{|\mathcal{C}|} \sum_{u \in \mathcal{C}} \left( \sum_{\xi \in \{0, 1\}^r} (m - 1)^{N(\xi)} (-1)^{E(\xi)} z_{j_1, j_2, \ldots, j_r} \right)^{s(\mathcal{I}; i)} \ldots
\]
\[
\left( \sum_{\xi \in \{0, 1\}^r} (m - 1)^{N(\xi)} (-1)^{E(\xi)} z_{k_1, k_2, \ldots, k_r} \right)^{s(\mathcal{I}; i)} \ldots
\]

78
\[
\cdots \left( \sum_{\mathbf{k} \in \{0,1\}^r} (m - 1)^{N(\mathbf{k}; \mathbf{1})} (-1)^{E(\mathbf{k}; \mathbf{1})} z_{k_1 \ldots k_r} \right)^{s(\mathbf{1}; \mathbf{1})} \\
= \frac{1}{|\mathcal{C}|} W^H_{\mathcal{C}} \left( \sum_{\mathbf{k} \in \{0,1\}^r} (m - 1)^{N(\mathbf{k}; \mathbf{1})} (-1)^{E(\mathbf{k}; \mathbf{1})} z_{k_1 \ldots k_r}, \ldots \right) \\
\sum_{\mathbf{k} \in \{0,1\}^r} (m - 1)^{N(\mathbf{k}; \mathbf{1})} (-1)^{E(\mathbf{k}; \mathbf{1})} z_{k_1 \ldots k_r}, \ldots , \\
\cdots \sum_{\mathbf{k} \in \{0,1\}^r} (m - 1)^{N(\mathbf{k}; \mathbf{1})} (-1)^{E(\mathbf{k}; \mathbf{1})} z_{k_1 \ldots k_r} \\
= \text{ right-hand side of (2.59).}
\]

So the proof is complete by equation (2.21). \(\square\)

When \(r = 2\), the generalized 2-fold Hamming weight enumerator is called the joint weight enumerator of \(\mathcal{C} = C_1 \times C_2\), \([39]\), and the relation between the joint weight enumerator of a code and its dual is:

**Corollary 2.5.2** (The generalized 2-fold Hamming weight enumerator) Assuming the above notation, we have,

\[
W^H_{\mathcal{C}_2}(z_{00}, z_{01}, z_{10}, z_{11}) \\
= \frac{1}{|\mathcal{C}|} W^H_{\mathcal{C}}(z_{00} + (m - 1)z_{01} + (m - 1)z_{10} + (m - 1)^2z_{11}, \quad (2.61) \\
z_{00} - z_{01} + (m - 1)z_{10} + (1 - m)z_{11}, z_{00} + (m - 1)z_{01} - z_{10} + (1 - m)z_{11}, \\
z_{00} - z_{01} - z_{10} + z_{11}).
\]

79
Proof: We basically use the main result (2.59) of this section for \( r = 2 \). We have.

\[
W_{c_4}^H(z_{00}, z_{01}, z_{10}, z_{11}) =
\frac{1}{|C|} W_{c_4}^H \left( \sum_{k \in \{0,1\}^r} ((m - 1)^{N(k,0)}(-1)^{E(k,0)}) z_{k_1 k_2}, \sum_{k \in \{0,1\}^r} ((m - 1)^{N(k,1)}(-1)^{E(k,1)}) z_{k_1 k_2} \right)
\]

where \( l \in \{0,1\}^r \). Now we compute the inner sums by using the definitions of \( E \) and \( N \) given in (2.53), and also observing that these sums run through \( k \in \{ (0,0), (0,1), (1,0), (1,1) \} \). Hence,

\[
(N(k; (0,0)), E(k; (0,0))) = \begin{cases} (0,0), & \text{if } k = (0,0) \\ (1,0), & \text{if } k = (0,1) \\ (1,0), & \text{if } k = (1,0) \\ (2,0), & \text{if } k = (1,1). \end{cases}
\]

\[
(N(k; (0,1)), E(k; (0,1))) = \begin{cases} (0,0), & \text{if } k = (0,0) \\ (0,1), & \text{if } k = (0,1) \\ (1,0), & \text{if } k = (1,0) \\ (1,1), & \text{if } k = (1,1). \end{cases}
\]
\[(N(k; (1, 0)), E(k; (1, 0))) = \begin{cases} 
(0, 0), & \text{if } k = (0, 0) \\
(1, 0), & \text{if } k = (0, 1) \\
(0, 1), & \text{if } k = (1, 0) \\
(1, 1), & \text{if } k = (1, 1). 
\end{cases}\]

\[(N(k; (1, 1)), E(k; (1, 1))) = \begin{cases} 
(0, 0), & \text{if } k = (0, 0) \\
(0, 1), & \text{if } k = (0, 1) \\
(0, 1), & \text{if } k = (1, 0) \\
(0, 2), & \text{if } k = (1, 1). 
\end{cases}\]

Using the above results we compute the sums,

\[
\sum_{k \in \{0,1\}^r} ((m - 1)^N(k; (0,0))(-1)^E(k; (0,0))) z_{k_1k_2}
\]

\[= z_{00} + (m - 1)z_{01} + (m - 1)z_{10} + (m - 1)^2z_{11}\]

\[
\sum_{k \in \{0,1\}^r} ((m - 1)^N(k; (1,0))(-1)^E(k; (0,1))) z_{k_1k_2}
\]

\[= z_{00} - z_{01} + (m - 1)z_{10} + (1 - m)z_{11}\]

\[
\sum_{k \in \{0,1\}^r} ((m - 1)^N(k; (1,0))(-1)^E(k; (1,0))) z_{k_1k_2}
\]

\[= z_{00} + (m - 1)z_{01} - z_{10} + (1 - m)z_{11}\]
\[
\sum_{\xi \in \{0, 1\}^r} ((m - 1)^{N(\xi; (1, 1))} (-1)^{E(\xi; (1, 1))}) z_{k_1 k_2} = z_{00} - z_{01} - z_{10} + z_{11}.
\]

Using the above equalities in equation (2.61), we get the final statement of the theorem. □

When \( C_1 = C_2 = \cdots = C_r = C \), the generalized \( r \)-fold Hamming weight enumerator is called the \( r \)-fold \textbf{Hamming weight enumerator} of \( C \). By setting \( C_1 = C_2 = \cdots = C_r = C \) in Theorem 2.5.1, we obtain the following MacWilliams-type identity for \( r \)-fold Hamming weight enumerators:

**Corollary 2.5.3** (The \( r \)-fold Hamming weight enumerator) Let \( C \) be a linear code of length \( n \) over a finite ring \( R \) and \( C^\perp \) its dual. Also, let the functions \( N \) and \( E \) be defined as in equation (2.53). Then the relation between the \( r \)-fold Hamming weight enumerator of a code \( C \) and its dual \( C^\perp \) is as follows:

\[
W_{(C^\perp)^r}^H(z_0, \ldots, z_{l_1 l_2 \ldots l_r}, \ldots, z_{11 \ldots 1}) = \frac{1}{|C|^r} W_C^H \left( \sum_{\xi \in \{0, 1\}^r} (m - 1)^{N(\xi; l)} (-1)^{E(\xi; l)} z_{k_1 \ldots k_r}, \ldots \right.
\]

\[
\sum_{\xi \in \{0, 1\}^r} (m - 1)^{N(\xi; l)} (-1)^{E(\xi; l)} z_{k_1 \ldots k_r}, \ldots, \sum_{\xi \in \{0, 1\}^r} (m - 1)^{N(\xi; l)} (-1)^{E(\xi; l)} z_{k_1 \ldots k_r})
\]

where \( l \in \{0, 1\}^r \).

Further, if we take \( C_1 = C_2 \) in Corollary 2.5.3, then we obtain the MacWilliams-type identity for 2-fold Hamming weight enumerator (or biweight enumerator) of a
code $C$. When $R = F_2$ the $r$-fold Hamming weight enumerator is called $r$-genus weight enumerator which is considered by Duke in [13]. The MacWilliams-type identity obtained for $r$-genus weight enumerators of codes follows as a corollary to Theorem 2.5.1.
2.6 On $r$-byte weight enumerators

The definition of $r$ byte weight enumerator of a binary code and the relation between 2-byte weight enumerator of a binary code and its dual is given by Wadayama et al in [51]. In [51], they use this approach to investigate some properties of the weight distribution for the Euclidean image of binary linear codes. In this section, we are going to extend the definition to any finite ring $R$, and obtain a MacWilliams-type identity for these weight enumerators. In the last chapter we are going to relate these weight enumerators to the complete weight enumerators of codes over $R + uR$. In this section we will assume that the length of a codeword, say $n$, is always divisible by $r$, i.e. $n = rs$ for some $s$.

2.6.1 Byte representation of a codeword

Let $u = (u_1, u_2, \ldots, u_n) \in C$ where $n = rs$. We define

$$u^1 = (u_1, u_2, \ldots, u_r) \in R^r$$
$$u^2 = (u_{r+1}, u_{r+2}, \ldots, u_{2r}) \in R^r$$
$$u^s = (u_{r(s-1)+1}, u_{r(s-1)+2}, \ldots, u_{rs}) \in R^r.$$  

In other words, $u^1, u^2, \ldots, u^s$ are the $r$-segments of a codeword starting from the first entry of the codeword. The $s$ tuple $(u^1, u^2, \ldots, u^s)$ is called the $r$-byte representation of a codeword $c$ of length $n = rs$. 

84
Example:

Let $R = F_3$ and $n = 6$ and $u = (2, 1, 0, 1, 2, 0) \in F_3^6$. Then, $u^1 = (2, 1), u^2 = (0, 1), u^3 = (2, 0)$ and the 2-byte representation of $u$ is $((2, 1), (0, 1), (2, 0))$. Actually, using the notation adopted in [51], the 2-byte representation of $u$ is given as $(21, 01, 20)_2$. Here, we are going to use the former representation which is more convenient for our purposes.

2.6.2 $r$-byte weight enumerator

Let $C$ be a code of length $n = rs$ over a ring $R = \{ \mathcal{J}_0, \mathcal{J}_1, \ldots, \mathcal{J}_{m-1} \}$. To each $r$-tuple in $R^r$ we associate the following weight function

$$
\eta_{i_1 i_2 \ldots i_r}(i_{j_1}, i_{j_2}, \ldots, i_{j_r}) = \begin{cases} 
1, & \text{if } (i_1, i_2, \ldots, i_r) = (j_1, j_2, \ldots, j_r) \\
0, & \text{otherwise.}
\end{cases}
$$

(2.65)

The $r$-byte weight enumerator of $C$ is given by

$$
W_C^{rb}(z_{0 \ldots 0}, \ldots, z_{1 \ldots 1}, \ldots, z_{m-1 \ldots m-1}) = 
\sum_{u \in C} z_{\mu_0(u)}^{\mu_0(u)} \ldots z_{\mu_{i_1 i_2 \ldots i_r}(u)}^{\mu_{i_1 i_2 \ldots i_r}(u)} \ldots z_{\mu_{m-1 \ldots m-1}(u)}^{\mu_{m-1 \ldots m-1}(u)}
$$

(2.66)

where

$$
\mu_{i_1 i_2 \ldots i_r}(u) = \sum_{j=1}^s \eta_{i_1 i_2 \ldots i_r}(u^j).
$$

(2.67)

(Note that the number of variables in an $r$-byte weight enumerator is $m^r$ and the superscript $rb$ in $W$ stands for r-byte.)
Example: Let \( u = (1, 0, 0, 1, 1, 1, 1, 1, 0) \in F_2^{10} \). Then,

\[
\mu_{i_1i_2}(u) = \begin{cases} 
2, & \text{if } (i_1, i_2) = (1, 0), \\
1, & \text{if } (i_1, i_2) = (0, 1), \\
2, & \text{if } (i_1, i_2) = (1, 1), \\
0, & \text{otherwise}.
\end{cases}
\]

Observe that the \( r \)-byte weight enumerator of a code is a multivariable polynomial with homogenous degree \( s \) where \( n = sr \).

**Theorem 2.6.1** (The \( r \)-byte weight enumerator) Let \( C \) be a code with length \( n = sr \) over a finite ring \( R \). Let \( \chi \) be a nontrivial additive character of \( R \) over all subgroups of \( R \) with size \( > 1 \). The relation between the \( r \)-byte weight enumerator of \( C \) and its dual is given by

\[
W_{C_\perp}^{rb}(z_0...z_r, \ldots, z_{m-1...m-1}) = \frac{1}{|C|} W_C^{rb}(\sum_{j_1,...,j_r=0}^{m-1} \chi(\beta_{j_1} \beta_0) \cdots \chi(\beta_{j_r} \beta_0) z_{j_1...j_r}, \ldots \sum_{j_1,...,j_r=0}^{m-1} \chi(\beta_{j_1} \beta_{i_1}) \cdots \chi(\beta_{j_r} \beta_{i_r}) z_{j_1...j_r}, \ldots \sum_{j_1,...,j_r=0}^{m-1} \chi(\beta_{j_1} \beta_{m-1}) \cdots \chi(\beta_{j_r} \beta_{m-1}) z_{j_1...j_r}).
\]

**Proof:** We recall Lemma 2.2.1 with \( r = 1 \):

\[
\sum_{u \in C_\perp} f(u) = \frac{1}{|C|} \sum_{u \in C} \tilde{f}(u)
\]
where
\[
\hat{f}(u) = \sum_{v \in V} \chi_u(v)f(v).
\]

Let \( V = \mathbb{R}^n \). For fixed \( u \in C \) we take
\[
f(u) = \sum_{\mu_0 \ldots \mu_r} \chi_{u}(v) f(v).
\]

Then,
\[
\sum_{u \in C} \hat{f}(u) = \sum_{u \in C} \sum_{v \in V} \chi_u(v)f(v) = \sum_{u \in C} \sum_{v \in V} \chi_u(v)z_{\mu_0 \ldots \mu_r}(v) \cdots z_{\mu_{m-1} \ldots m-1}(v).
\]

For fixed \( u \in C \), first we compute \( \hat{f}(u) \). Using the definition of \( \mu_{i_1 \ldots i_r} \) in (2.67), we have
\[
\hat{f}(u) = \sum_{v \in V} \chi_u(v)z_{\mu_0 \ldots \mu_r}(v) \cdots z_{\mu_{m-1} \ldots m-1}(v).
\]

We rewrite each vector \( v \) and \( u \) in their \( r \)-byte form, and we observe that \( \chi_u(v) = \chi_{u^1}(v^1) \cdots \chi_{u^r}(v^r) \). Then,
\[
\hat{f}(u) = \sum_{v^1, \ldots, v^r \in \mathbb{R}^r} \chi_{u^1}(v^1) \cdots \chi_{u^r}(v^r).
\]
Collecting each $v^i$, $1 \leq i \leq s$ under a single sum, we get

$$
\tilde{f}(u) = \left( \sum_{v^1 \in R^r} \chi_{u^1}(v^1)z_{0...0}^{\eta_{0...0}(v^1)} \ldots z_{1...1}^{\eta_{1...1}(v^1)} \ldots z_{m-1...m-1}(v^1) \right) \\
\left( \sum_{v^2 \in R^r} \chi_{u^2}(v^2)z_{0...0}^{\eta_{0...0}(v^2)} \ldots z_{1...1}^{\eta_{1...1}(v^2)} \ldots z_{m-1...m-1}(v^2) \right) \\
\vdots \\
\left( \sum_{v^s \in R^r} \chi_{u^s}(v^s)z_{0...0}^{\eta_{0...0}(v^s)} \ldots z_{1...1}^{\eta_{1...1}(v^s)} \ldots z_{m-1...m-1}(v^s) \right).
$$

We use the definition of $\chi_u$, (2.9), and rewrite each $v^i$ more explicitly, with the observation that $\eta_{i_1...i_r}((\beta_{j_1}, \ldots, \beta_{j_r})) = 0$ unless $(i_1 \ldots i_r) = (j_1 \ldots j_r)$. Hence, we get

$$
\tilde{f}(u) = \left( \sum_{\beta_{j_1}, \ldots, \beta_{j_r} \in R^r} \chi((\beta_{j_1}, \ldots, \beta_{j_r})u^1)z_{j_1...j_r}^{\eta_{j_1...j_r}((\beta_{j_1}, \ldots, \beta_{j_r}))} \right) \\
\left( \sum_{\beta_{j_1}, \ldots, \beta_{j_r} \in R^r} \chi((\beta_{j_1}, \ldots, \beta_{j_r})u^2)z_{j_1...j_r}^{\eta_{j_1...j_r}((\beta_{j_1}, \ldots, \beta_{j_r}))} \right) \\
\vdots \\
\left( \sum_{\beta_{j_1}, \ldots, \beta_{j_r} \in R^r} \chi((\beta_{j_1}, \ldots, \beta_{j_r})u^s)z_{j_1...j_r}^{\eta_{j_1...j_r}((\beta_{j_1}, \ldots, \beta_{j_r}))} \right).
$$

By definition of $\eta_{j_1...j_r}$ in (2.65), the exponents of $z_{j_1...j_r}$ are equal to one. So

$$
\tilde{f}(u) = \left( \sum_{\beta_{j_1}, \ldots, \beta_{j_r} \in R^r} \chi((\beta_{j_1}, \ldots, \beta_{j_r})u^1)z_{j_1...j_r} \right) \\
\left( \sum_{\beta_{j_1}, \ldots, \beta_{j_r} \in R^r} \chi((\beta_{j_1}, \ldots, \beta_{j_r})u^2)z_{j_1...j_r} \right) \\
\vdots
$$

88
Recalling that \( \mu_{i_1, \ldots, i_r}(u) \) counts the number of \( r \)-bytes (\( r \)-segments) of the codeword \( u \) that are equal to the \( r \)-tuple \((\beta_1, \ldots, \beta_r)\), we get

\[
\hat{f}(u) = \prod_{i_1, \ldots, i_r \in \mathbb{Q}} \left( \sum_{\beta_1, \ldots, \beta_r \in \mathbb{R}} \chi((\beta_1, \ldots, \beta_r)(\beta_{i_1}, \ldots, \beta_{i_r}))z_{j_1 \ldots j_r}\right)^{\mu_{i_1, \ldots, i_r}(u)}.
\]

Now applying the equation (2.69) given at the beginning of the proof, we obtain the result. \( \Box \)

**Corollary 2.6.1** [51] (The 2-byte weight enumerator) Let \( C \) be a binary code of even length. Then the relation between a 2-byte weight enumerator of \( C \) and its dual is given by

\[
W_{C^\perp}^{2b}(z_{00}, z_{01}, z_{10}, z_{11}) = \frac{1}{|C|} W_C^{2b}(z_{00} + z_{01} + z_{10} + z_{11}, \quad (2.70)
\]

\[
z_{00} - z_{01} + z_{10} - z_{11}, z_{00} + z_{01} - z_{10} - z_{11}, z_{00} - z_{01} - z_{10} + z_{11}.
\]

**Proof:** The left-hand side is clear. To compute the right-hand side we take an additive character of \( \mathbb{F}_2 \), \( \chi(x) = e^{\pi i x} = (-1)^x \), and let

\[
A := \sum_{j_1, j_2 = 0}^{1} \chi(\beta_{j_1}, \beta_{j_2})\chi(\beta_{j_1}, \beta_{j_2})z_{j_1 j_2}
\]

\[
= \sum_{j_1, j_2 = 0}^{1} (-1)^{j_1 + j_2}z_{j_1 j_2}
\]

\[
= z_{00} + (-1)^{j_2}z_{01} + (-1)^{j_1}z_{10} + (-1)^{j_1 + j_2}z_{11}.
\]
Now, we compute

\[ A = \begin{cases} 
  z_{00} + z_{01} + z_{10} + z_{11}, & \text{if } (i_1, i_2) = (0, 0) \\
  z_{00} - z_{01} + z_{10} - z_{11}, & \text{if } (i_1, i_2) = (0, 1) \\
  z_{00} + z_{01} - z_{10} - z_{11}, & \text{if } (i_1, i_2) = (1, 0) \\
  z_{00} - z_{01} - z_{10} + z_{11}, & \text{if } (i_1, i_2) = (1, 1). 
\end{cases} \]

Hence, the result. □
CHAPTER 3
NEW LINEAR CODES WITH IMPROVED BOUNDS ON
MINIMUM DISTANCES

Recently, there has been much research on quasi-cyclic codes. These codes are a very natural generalization of cyclic codes. Some of the important facts that have motivated the researches are the following:

1. Quasi-cyclic codes meet a modified version of the Gilbert-Varshamov bound unlike many other classes of codes, (Kasami)[31].

2. Some of the best quadratic residue codes and Pless symmetry codes are quasi cyclic, [39].

3. They enjoy a simpler algebraic structure than most linear codes.

4. A large number of new linear codes with improved minimum distances are quasi-cyclic codes. Among these, there is a significant number of optimal codes (with the best possible minimum distance that a code can achieve).

5. They are natural generalizations of cyclic codes.

Due to the facts mentioned above and many more, researchers have worked on quasi-cyclic codes and discovered new linear codes with improved minimum distances
over finite fields of orders 2, 3, 5, 7, 8, and 9. Most of the work can be found in papers of Gulliver, Bhargava, Henk Van Tilborg, Dasklov ([25], [26], [21], [23], [18], [52], [3], [49], [19], [9], [10], [20], [17], [22], [6]).

In this chapter we are going to give the definition and some known properties of quasi-cyclic codes. Especially, we are going to provide some new and recent results on quasi-cyclic codes obtained by Lally and Fitzpatrick in [34] with a different approach. Next, we will generalize the methods introduced by Gulliver and Harada in [24] and obtain new linear codes over $\mathbb{F}_3$ and $\mathbb{F}_5$ with improved minimum distances.

### 3.1 Quasi Cyclic Codes

Quasi cyclic codes are generalization of cyclic codes. Let $C$ be a linear code over a finite ring $R$ of length $n$. Let $l \in \mathbb{N}$ and define

$$\mu_l : C \rightarrow R^n$$

$$\mu_l((c(0), \ldots, c(i), \ldots, c(n - 1))) = (c(0 - l), \ldots, c(i - l), \ldots, c(n - 1 - l))$$

where $i - l = i - l \mod n$ for $0 \leq i \leq n - 1$.

**Definition 3.1.1** A linear code $C$ over a field $F$ is called an $l$-quasi-cyclic $(l$-$QC)$ code if and only if $C$ is invariant under $\mu_l$ i.e. $\mu_l(C) = C$.

In other words a right cyclic shift of any codeword by $l$ positions is still a codeword. In the sequel, we will assume that $l$ divides the length of the code, say $n = ml$ for some $m$. Otherwise, if $n$ and $l$ are coprime, then the code $C$ is a cyclic code. Also, in the above definition if we let $l = 1$ then we have the definition of a cyclic code.
Further work in the algebraic structure of the QC codes can be found in [5] by Chen and Peterson, in [7] by Conan and Seguin, in [33] by Koshy, and recently in [34] by Lally and Fitzpatrick.

Let

\[ B := circ_k(b_0, b_1, \ldots, b_{m-1}) = \begin{bmatrix}
    b_0 & b_1 & b_2 & \cdots & b_{m-1} \\
    b_{m-1} & b_0 & b_1 & \cdots & b_{m-2} \\
    b_{m-2} & b_{m-1} & b_0 & \cdots & b_{m-3} \\
    \vdots & \vdots & \vdots & \ddots & \vdots \\
    b_{m-k+1} & b_{m-k+2} & b_{m-k+3} & \cdots & b_{m-k}
\end{bmatrix}_{k \times m} \]  

(3.1)

A \((k \times m)\) matrix of the type \(B\) is called a **nearly \((k \times m)\) circulant matrix**. If \(k = m\), then \(B\) is called a **circulant matrix** of order \(m\) and the matrix will be denoted by \(circ(b_0, b_1, \ldots, b_{m-1})\).

To each circulant matrix of order \(m\), say \(B\), we can associate a polynomial that is identified by the first row of this matrix. For instance, to the matrix \(B\) we can associate the polynomial \(b(x) = b_0 + b_1x + \cdots + b_{m-1}x^{m-1}\). This polynomial is called an **identifying polynomial** of \(B\). It is clear that from the first row we can recover the circulant matrix uniquely. The following rows of this matrix are single right shifts of the first. The rows of the matrix in an obvious way correspond to the polynomials \(b(x), xb(x), x^2b(x), \ldots, x^{m-1}b(x)\).

Let

\[ \sigma : \{0, 1, \ldots, n - 1\} \to \{0, 1, \ldots, n - 1\} \]

be a bijection, also called permutation. For a code \(C\), define
\[ \sigma(C) = \{(c(\sigma(0)), c(\sigma(1)), \ldots, c(\sigma(n-1)))\} \text{ for all } (c(0), c(1), \ldots, c(n-1)) \in C \}. \]

**Definition 3.1.2** Let \( C_1, C_2 \) be linear codes of length \( n \). \( C_2 \) is said to be (permutation) equivalent to \( C_1 \) if and only if \( \sigma(C_1) = C_2 \) for some permutation \( \sigma \).

Before stating and proving the following lemma, we shall give an example which will illuminate the proof of the next lemma.

**Example:** Let \( C \) be a 2-dimensional 3-QC code of length 6. Let the generator matrix \( D \) of \( C \) be given by

\[
D := \begin{bmatrix}
c_0 & c_1 & c_2 & c_3 & c_4 & c_5 \\
d_0 & d_1 & d_2 & d_3 & d_4 & d_5
\end{bmatrix}
\]

where \((c_0, c_1, c_2, c_3, c_4, c_5)\) and \((d_0, d_1, d_2, d_3, d_4, d_5)\) are a basis of \( C \).

Since \( C \) is 3-QC code, the following matrix \( D^* \) is also a generator of \( C \):

\[
D^* := \begin{bmatrix}
c_0 & c_1 & c_2 & c_3 & c_4 & c_5 \\
c_3 & c_4 & c_5 & c_0 & c_1 & c_2 \\
d_0 & d_1 & d_2 & d_3 & d_4 & d_5 \\
d_3 & d_4 & d_5 & d_0 & d_1 & d_2
\end{bmatrix}
\]

Now we apply the permutation \( \sigma = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 \\ 0 & 5 & 2 & 1 & 4 & 3 \end{pmatrix} \) to all codewords of \( C \). Especially, the following matrix which consists of permuted basis codewords

94
is a generator matrix for $\sigma(C)$.

**Remark:** To obtain the matrix $\sigma(D^*)$, since the code is a 3-QC code, the first column and fourth (3 units apart from each other), the second and the fifth and the third and the sixth are put adjacent to each other.

We observe that $\sigma(D^*)$ consists of six $2 \times 2$ circulant matrices. This property is generalized to all QC codes in the following lemma which establishes an important fact about the generator matrix of a quasi-cyclic code studied in [33]. We are going to give an independent proof of this lemma based on the above example.

**Lemma 3.1.1 [33]** Let $C$ be an $[n, k, d]$ l-QC code. Then, $C$ is equivalent to a linear code with generator matrix

\[
\sigma(D^*) = \begin{bmatrix}
c_0 & c_3 & c_2 & c_5 & c_4 & c_1 \\
c_3 & c_0 & c_5 & c_2 & c_1 & c_4 \\
d_0 & d_3 & d_2 & d_5 & d_4 & d_1 \\
d_3 & d_0 & d_5 & d_2 & d_1 & d_4
\end{bmatrix}
\]

where $a(h, k)$ denotes the $(h, k)$th entry of $A$ and each $A_{ji} = (a_{ji}(l_1, l_2))$ is a circulant matrix of size $m \times m$, $1 \leq h \leq sm, 1 \leq k \leq n, 1 \leq i \leq l, 1 \leq j \leq s, 1 \leq l_1, l_2 \leq m$. Further, a code with a generator matrix of the above form is an l-QC code.
Proof: Assume that $n = ml$. Let $c_1, c_2, \ldots, c_k \in F^n$ be a basis for $C$ where $c_j = (c_j(0), c_j(1), \ldots, c_j(n - 1))$ for $1 \leq j \leq k$. We consider the matrix whose rows are formed by this basis vectors. In other words, consider

$$D = \begin{bmatrix}
  c_1(0) & c_1(1) & \cdots & c_1(n - 1) \\
  c_2(0) & c_2(1) & \cdots & c_2(n - 1) \\
  \vdots & \vdots & \ddots & \vdots \\
  c_k(0) & c_k(1) & \cdots & c_k(n - 1)
\end{bmatrix}_{k \times n}$$

(3.3)

Since the code $C$ is $l$-quasi-cyclic, it is clear that the matrix

$$D^* = \begin{bmatrix}
  (c_1(0), c_1(1), \ldots, c_1(m - 1)), \cdots, c_1(m(l - 1)), \cdots, c_1(n - 1)) \\
  \mu_l(c_1(0), c_1(1), \ldots, c_1(m - 1)), \cdots, c_1(m(l - 1)), \cdots, c_1(n - 1)) \\
  \vdots \\
  \mu_l^{m-1}(c_1(0), c_1(1), \ldots, c_1(m - 1)), \cdots, c_1(m(l - 1)), \cdots, c_1(n - 1)) \\
  \mu_l(c_2(0), c_2(1), \ldots, c_2(m - 1)), \cdots, c_2(m(l - 1)), \cdots, c_2(n - 1)) \\
  \vdots \\
  \mu_l^{m-1}(c_2(0), c_2(1), \ldots, c_2(m - 1)), \cdots, c_2(m(l - 1)), \cdots, c_2(n - 1)) \\
  (c_k(0), c_k(1), \ldots, c_k(m - 1)), \cdots, c_k(m(l - 1)), \cdots, c_k(n - 1)) \\
  \mu_l(c_k(0), c_k(1), \ldots, c_k(m - 1)), \cdots, c_k(m(l - 1)), \cdots, c_k(n - 1)) \\
  \vdots \\
  \mu_l^{m-1}(c_k(0), c_k(1), \ldots, c_k(m - 1)), \cdots, c_k(m(l - 1)), \cdots, c_k(n - 1)) \\
\end{bmatrix}_{nk \times n}$$

(3.4)
is also a generating matrix for $C$. Now, we apply the following permutation $P$ to the columns of $D^*$:

$$P : c_j \rightarrow (c_j(0), c_j(l), \ldots, c_j(l(m - 1)), c_j(1), c_j(1 + l),$$

$$\ldots, c_j(1 + l(m - 1)), \ldots, c_j(l - 1), c_j(2l - 1), \ldots, c_j(lm - 1))$$

for $1 \leq j \leq k$. We consider the following blocks of the permuted matrix $D^*$

$$A_{ji} = \begin{bmatrix}
    c_j(i - 1) & c_j(i - 1 + l) & \ldots & c_j(i - 1 + l(m - 1)) \\
    c_j(i - 1 + l(m - 1)) & c_j(i - 1) & \ldots & c_j(i - 1 + l(m - 2)) \\
    \vdots & \vdots & \ddots & \vdots \\
    c_j(i - 1 + l) & c_j(i - 1 + 2l) & \ldots & c_j(i - 1)
\end{bmatrix} \quad (3.5)$$

where $1 \leq i \leq l$. Letting $k = s$, this leads us to the desired form (3.2). Hence, $C$ is (permutation) equivalent to a code with generator matrix of the form (3.2). Conversely, from the construction, it is clear that a code with a generator matrix of form (3.2) is permutation equivalent to an $l$-QC code. □

Since a QC code is permutation equivalent to a linear code with a generator matrix of type (3.2), in the rest of this chapter we will only consider QC codes that are generated by matrices of type (3.2).

**Definition 3.1.3** A linear code with a generator matrix of the form (3.2) where the submatrices $A_{j1}, A_{j2}, \ldots, A_{js}$ are nearly $k_j \times m$ circulant matrices for all $1 \leq j \leq s$, is called a $(k_1, k_2, \ldots, k_s)$ nearly quasi-cyclic code.
We have introduced Definition 3.1.3 because some of the new codes found in the following sections, are of this type. It is clear that quasi-cyclic codes are nearly quasi-cyclic.

To each codeword of an \( l \)-QC code with length \( n = ml \), we can associate an element of \( (F_q(x)/(x^m - 1))^l \) as follows:

\[
\begin{align*}
\{(c(0), c(1), \ldots, c(m - 1), \ldots, c((l - 1)m), \ldots, c(ml - 1)) & \rightarrow (c(0) + c(1)x + \cdots + c(m - 1)x^{m-1}, \ldots, c((l - 1)m) + \cdots + c(ml - 1)x^{m-1}).
\end{align*}
\]

Under the above identification, we can view \( C \) as an \( F_q[x]/(x^m - 1) \)-submodule of \( (F_q[x]/(x^m - 1))^l \). Note the similarity between QC codes and cyclic codes, especially in case \( l = 1 \), a QC code is a cyclic code.

We adopt the following notation for brevity:

\[
F := F_q[x]/(x^m - 1).
\]

The submatrix

\[
[A_{j1} \ A_{j2} \ \ldots \ A_{jl}]
\]

will be called the \( j^{th} \) horizontal block of \( A \). Corresponding to the first row of each horizontal block of \( A \), we get an element of \( F^l \). The set of \( s \) elements of \( F^l \) corresponding to the \( s \) horizontal blocks of \( A \) will be called an \( s \)-generator set of an \( l \)-QC code.

Note that this does not imply that \( C \) has dimension \( s \). Later we will prove a result relating the dimension and \( s \).
Definition 3.1.4 Let $R$ be a finite commutative ring. A set of codewords $\{v_1, \ldots, v_k\}$ is said to be linearly independent over $R$ if and only if for all $(\alpha_1, \ldots, \alpha_k) \in R^k$,
\[
\sum_{i=0}^{k} \alpha_i v_i = 0 \implies \alpha_i = 0, 1 \leq i \leq k.
\]

In the rest of this chapter, we are going to view a code $C$ as an $F$-submodule of $F^l$ with an $s$-generator set. Also $\langle g(x) \rangle$ will denote the ideal generated by $g(x)$.

Let $1 \leq i \leq l$. For fixed $i$ consider the following $i^{th}$ restriction map:

$$\Pi_i : (c(0), \ldots, c(ml - 1)) \to \langle c((i - 1)m), (1 + (i - 1)m), \ldots, c(m - 1 + (i - 1)m) \rangle.$$  \hspace{1cm} (3.7)

Clearly $\Pi_i(C)$ is a linear code for all $i$.

Lemma 3.1.2 Let $C$ be an $l$-QC code with the above notation. $\Pi_i(C)$ is a cyclic code.

Proof: First observe that $\Pi_i(C) = \langle a_{1i}(x), a_{2i}(x), \ldots, a_{ni}(x) \rangle$ where $a_{1i}(x), \ldots, a_{ni}(x)$ are respectively the identifying polynomials of the submatrices $A_{1i}, \ldots, A_{ni}$ for $1 \leq i \leq l$ of $A$ (3.2). Therefore, it easily follows that $\Pi_i(C) = \langle g_i(x) \rangle$ where $g_i(x) = \gcd(a_{1i}(x), a_{2i}(x), \ldots, a_{ni}(x))$ for some $g_i(x) \in F$. $\square$

Lemma 3.1.3 Let $g'_1(x), g'_2(x), \ldots, g'_s(x) \in (F_q(x)/(x^m - 1))^l$ be an $s$ generator set for an $l$-QC code $C$ with length $n = ml$. Let $g'_j(x) = (g'_j_1(x), g'_j_2(x), \ldots, g'_j_l(x))$. Then, for some polynomial $g_i(x)$, $g'_j_i(x) = f_{j_i}(x)g_i(x)$ i.e

$$g'_j(x) = (f_{j_1}(x)g_1(x), f_{j_2}(x)g_2(x), \ldots, f_{j_l}g_l(x))$$  \hspace{1cm} (3.8)

where $g_i(x)|(x^m - 1), f_{j_i}(x) \in F(x)/(x^m - 1)$ for $1 \leq i \leq l$, and $1 \leq j \leq s$. 

99
Proof: Using the map $\Pi_t$ defined above, we have $\Pi_t(C) = \langle g_i(x) \rangle$. Without loss of
generality we may assume that $g_i(x)|(x^m - 1)$ for $1 \leq i \leq l$. Thus, $g'_j(x) = f_j(x)g_i(x)$
for some $f_j(x), g_i(x) \in F(x)/(x^m - 1)$ where $g_j(x)|(x^m - 1)$. □

3.1.1 1-generator QC codes

Let $C$ be a 1-generator $l$-QC code of length $n = ml$. So a generator matrix $A$ has
only one horizontal block, i.e

$$A = \begin{bmatrix}
A_{11} & A_{12} & \cdots & A_{1l}
\end{bmatrix}$$

or more explicitly,

$$A = \begin{bmatrix}
circ(a_{10}, a_{11}, \ldots, a_{1,m-1}), circ(a_{20}, a_{21}, \ldots, a_{2,m-1}), \ldots, circ(a_{l0}, a_{l1}, \ldots, a_{l,m-1})
\end{bmatrix}.$$  \hspace{1cm} (3.9)

Viewing $C$ as an $F$-submodule of $F^l$, we observe that

$$(a_1(x), a_2(x), \ldots, a_l(x))$$  \hspace{1cm} (3.10)

where $a_i(x) = a_{i0} + a_{i1}x + \cdots + a_{i,m-1}x^{m-1}$ for $1 \leq i \leq l$ is a 1-generating set for
$C$. A 1-generating set for an $l$-QC code, by Lemma 3.1.3, has the following form:

$$(f_1(x)g_1(x), f_2(x)g_2(x), \ldots, f_l(x)g_l(x))$$  \hspace{1cm} (3.11)

where $g_i(x)|(x^m - 1)$ and further, by renaming the polynomials, we may assume that
$(f_i(x), h_i(x)) = 1, h_i(x) = (x^m - 1)/g_i(x)$ for $1 \leq i \leq l$. Note that after renaming,
g_i(x)'s may not be generators for the cyclic code $\Pi_l(C), 1 \leq i \leq l$. 

100
Lemma 3.1.4 [46] Let \( C \) be an \( l \)-QC code with a \( 1 \)-generating set

\[
\overline{f(x)} := (f_1(x)g_1(x), f_2(x)g_2(x), \ldots, f_l(x)g_l(x))
\]

where \( g_i(x) | (x^m - 1) \) and \( (f_i(x), h_i(x)) = 1 \), \( h_i(x) = (x^m - 1)/g_i(x) \) for \( 1 \leq i \leq l \). Then, the dimension of \( C \) is equal to \( m - \deg(gcd(g_1(x), g_2(x), \ldots, g_l(x), x^m - 1)) \).

Proof: First we observe that if \( g(x) = gcd(g_1(x), g_2(x), \ldots, g_l(x), x^m - 1) \) and \( h(x) = (x^m - 1)/g(x) \), then for any polynomial \( p(x) \neq 0 \) with \( \deg(p(x)) < \deg(h(x)) \) we have

\[
p(x)(f_1(x)g_1(x), f_2(x)g_2(x), \ldots, f_l(x)g_l(x)) \neq 0.
\] (3.12)

We prove (3.12) by contradiction. Assume that \( p(x)f_i(x)g_i(x) = 0 \) in \( (F_q[x]/(x^m - 1))^i \) for \( 1 \leq i \leq l \) i.e \( (x^m - 1)/p(x)f_i(x)g_i(x) \) in \( F_q[x] \) for \( 1 \leq i \leq l \). Since \( g_i(x)h_i(x) = x^m - 1 \), we have \( h_i(x)|p(x)f_i(x) \) for \( 1 \leq i \leq l \). Further, since \( (f_i(x), h_i(x)) = 1 \) we get \( h_i(x)|p(x) \) for \( 1 \leq i \leq l \). Hence, \( \text{lcm}_{i=1,\ldots,l}(h_i(x))|p(x) \).

Now, we claim that \( \text{lcm}_{i=1,\ldots,l}(h_i(x)) = h(x) \). Let \( x^m - 1 = \Pi_{i=1}^l p_i(x) \) where \( p_i(x) \) are irreducible polynomials in \( F_q[x] \). (Note that we always assume that the characteristic of the field and \( m \) are coprime, hence \( x^m - 1 \) can not have multiple roots in its splitting field. This implies that \( x^m - 1 \) can not have multiple factors.) Let \( g_i(x) = \Pi_{j \in A_i} p_j(x) \) where \( A_i \subset S := \{1, 2, \ldots, s\} \). Then, \( h_i(x) = \Pi_{i \in \mathcal{A}_i} p_i(x) \) where \( \mathcal{A}_i \) denotes the complement set of \( A_i \) in \( S \). Also, \( g(x) = \Pi_{j \in \bigcap_i A_i} p_j(x) \) and \( h(x) = \Pi_{j \in \bigcap_i \mathcal{A}_i} p_j(x) \) and \( \text{lcm}_{i=1,\ldots,l}(h_i(x)) = \Pi_{j \in \bigcup_{i \in \mathcal{A}_i}} p_j(x) \), which shows that \( \text{lcm}_{i=1,\ldots,l}(h_i(x)) = h(x) \).

Thus, \( h(x)|p(x) \) which is a contradiction to the fact that \( \deg(p(x)) < \deg(h(x)) \).
Since \( p(x)f_i(x)g_i(x) \) is not divisible by \( x^m - 1 \) for all \( i \), the elements of the set

\[
B := \{ \overline{f(x)}, x\overline{f(x)}, x^2\overline{f(x)}, \ldots, x^{\deg(h(x))-1}\overline{f(x)} \}
\]

are linearly independent.

Now we show that \( B \) spans \( C \). Since \( g(x) = \gcd(g_1(x), g_2(x), \ldots, g_l(x), x^m - 1) \) we have \( g_i(x) = g_i^*(x)g(x) \) for some \( g_i^*(x) \). Let \( e(x)\overline{f(x)} = e(x)(f_1(x)g_1(x), \ldots, f_l(x)g_l(x)) \) be an arbitrary element of \( C \). Then,

\[
e(x)\overline{f(x)} = (e(x)f_1(x)g_1^*(x)g(x), \ldots, e(x)f_l(x)g_l^*(x)g(x)).
\]

Clearly, \( \deg(e(x)f_i(x)g_i^*(x)) < \deg(h(x)) \) in \( F_q[x]/(x^m - 1) \) for all \( i \). Therefore, \( e(x)\overline{f(x)} \) is a linear combination of elements of \( B \). \( \Box \)

The following theorems have been introduced by Lally and Fitzpatrick in [34]. Here, we give independent proofs to these theorems.

**Theorem 3.1.1** [34] Let \( C \) be an \( l \)-QC code of length \( n = ml \) with a \( 1 \)-generating set

\[
\overline{f(x)} := (f_1(x)g_1(x), f_2(x)g_2(x), \ldots, f_l(x)g_l(x))
\]

where \( g_i(x)|(x^m - 1) \), \( (f_{i_0}, g_{i_0}) = 1 \) for some \( i_0 \), and \( (f_i(x), h_i(x)) = 1 \), \( h_i(x) = (x^m - 1)/g_i(x) \) in \( F_q[x] \) for \( 1 \leq i \leq l \). Indeed \( C \) is equivalent to a code generated by

\[
(k_1(x), r_2(x)k_2(x), \ldots, r_l(x)k_l(x))
\]

where \( k_i(x)|(x^m - 1) \), \( (r_i(x), x^m - 1)/k_i(x)) = 1 \), \( 1 \leq i \leq l \).
Proof: Without loss of generality we may take \( i_0 = 1 \) because we can permute the coordinates and obtain an equivalent code. So, we have \((f_1(x), x^m - 1) = 1\). There exist \(d_1(x), d_2(x) \in F_q[x]\) such that \(d_1(x)f_1(x) + d_2(x)(x^m - 1) = 1\). Thus, \(d_1(x)f_1(x) = 1\) in \((F_q[x]/(x^m - 1))^l\). Now consider

\[
d_1(x)(f_1(x)g_1(x), f_2(x)g_2(x), \ldots, f_l(x)g_l(x)) = (g_1(x), d_1(x)f_2(x)g_2(x), \ldots, d_1(x)f_l(x)g_l(x)) \text{ in } F_q[x]/(x^m - 1).
\]

Clearly, \((g_1(x), d_1(x)f_2(x)g_2(x), \ldots, d_1(x)f_l(x)g_l(x)) \in C\). Since \((d_1(x), x^m - 1) = 1\), and \(\gcd(f_1(x)g_1(x), \ldots, f_l(x)g_l(x), x^m - 1) = \gcd(g_1(x), \ldots, d_1(x)f_l(x)g_l(x), x^m - 1)\), both \(\overline{f(x)}\) and \(d_1(x)\overline{f(x)}\) generate a code with the same dimension, so they generate the same code. □

By consecutive roots of \(g(x)\) we mean the roots of \(g(x)\) which are powers of a primitive \(n^{th}\) root of unity where the exponents of the primitive \(n^{th}\) root are consecutive.

Theorem 3.1.2 [34] Let \(C\) be a \(l\)-generating \(l\)-QC code of length \(n = ml\) with the following generator matrix:

\[
(g(x), f_2(x)g(x), \ldots, f_l(x)g(x))
\]

where \(g(x)|(x^m - 1), g(x), f_i(x) \in F[x]/(x^m - 1), \text{ and } (f_i(x), \frac{x^m - 1}{g(x)}) = 1 \text{ for } 1 \leq i \leq l\). Then,

\[
l|\{(\# \text{ of consecutive roots of } g(x)) + 1\}| \leq d(C),
\]

and dimension of \(C\) is equal to \(n - \deg(g(x))\).
Proof: Observe that $\Pi_i(C)$ is a cyclic code generated by $< f_i(x)g(x) >$ where $1 \leq i \leq l$. Since $(f_i(x), \frac{x^m-1}{g(x)}) = 1$, we have $< f_i(x)g(x) >= < g(x) >$ by Lemma 1.3.1. Also we note that the zero codeword of $C$ occurs if and only if each codeword of $\Pi_i(C)$ is zero. This implies that if $c$ is a nonzero codeword of $C$ then $\Pi_i(c) \neq 0$. Since $\Pi_i(C)$ is a cyclic code with generator polynomial $g(x)$, every nonzero codeword has weight strictly larger than the number of consecutive roots of $g(x)$ by Corollary 1.3.3 (BCH bound). Hence, a nonzero codeword in $C$ has a weight larger or equal to $l((\# \text{ consecutive roots of } g(x)) + 1)$. Moreover, by Lemma 3.1.4 the dimension of $C$ is equal to $n \text{ - } \deg(g(x))$. □
3.2 Codes over rings of type $R + uR$

Let $R$ be a finite commutative ring with identity. From this point on, we will assume that $R$ is a commutative ring. We recall from the introduction that linear codes over commutative rings are submodules. Further if $C$ is a $k$-free (with a basis of $k$ elements) submodule with minimum distance $d$ and length $n$, then $C$ is called an $[n, k, d]$-linear code. Note that in the field case we use the notation $[n, k, d]_q$ and in the ring case we drop the subscript $q$. We consider the quotient $R_a = R[u]/(u^2 - a)$, where $a \in R$ and $u$ is an indeterminate, i.e.

$$R_a = \{ r + us | u^2 = a \text{ and } r, s \in R \}.$$ 

In the sequel we will consider $R$ as a subring of $R_a$ in an obvious way. We are going to consider only linear codes, however we can view nonlinear codes over $R_a$ simply as subsets of $R_a^n$. Similar work has been done by Gulliver and Harada in [24] for the special case $R = F_2$ and $a = 0$. We note that if $R = F_2$ then $R_0$ and $R_1$ are isomorphic rings via the map $u \rightarrow u + 1$ and $r \rightarrow r$ for all $r \in F_2$, but if $R = F_3$ they are not.

The **Gray weight** of an element $x + uy \in R_a$ is denoted and defined as follows:

$$w_G(x + uy) = \begin{cases} 
0, & \text{if } (x, y) = (0, 0) \\
1, & \text{if exactly one of } x \text{ or } y \text{ is nonzero} \\
2, & \text{if both } x \text{ and } y \text{ are nonzero.} 
\end{cases} \quad (3.15)$$

The Gray weight of an $n$-tuple $c = (c_1, c_2, \ldots, c_n) \in R_a^n$ is the sum of the Gray weights of each component i.e.

$$w_G(c) = \sum_{i=1}^{n} w_G(c_i).$$

Also, the **Gray distance**
between two codewords $c$ and $e$ is $d_G(c, e) = w_G(c - e)$. It can be easily checked that this Gray distance is a metric. The minimum Gray weight of a code is the smallest nonzero Gray weight among all its codewords. In the case of linearity the **minimum Gray distance** $d_G(C)$ equals to the minimum Gray weight of the code $C$.

Now we relate the elements of $R_a$ and $R^2$ by the following map

$$\alpha : R_a \rightarrow R^2$$

$$\alpha(r + us) = (\alpha_1(r + us), \alpha_2(r + us))$$

where $\alpha_1(r + us) = s, \alpha_2(r + us) = r$. We will also denote $R_a$ by $R + uR$.

We can easily see that the map $\alpha$ which was defined in [24] is a special case of the above map (3.16) where $R_a = \mathbb{F}_3 + u\mathbb{F}_3, R = \mathbb{F}_3$ and $a = 0$.

We define a **Gray map** $\phi : R^n_a \rightarrow R^{2n}$ by

$$\phi(c) = (\alpha_1(c_1), \alpha_2(c_1), \alpha_1(c_2), \alpha_2(c_2), \ldots, \alpha_1(c_n), \alpha_2(c_n))$$

$$= (\alpha(c_1), \alpha(c_2), \ldots, \alpha(c_n))$$

where $c = (c_1, c_2, \ldots, c_n)$. We observe that $\phi$ is an injective map.

We naturally define an extension mapping

$$\phi : M_{k \times n}(R_a) \rightarrow M_{k \times 2n}(R)$$

$$\phi \left( \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_k \end{bmatrix}_{(k \times n)} \right) = \begin{bmatrix} \phi(c_1) \\ \phi(c_2) \\ \vdots \\ \phi(c_k) \end{bmatrix}_{k \times 2n}$$

106
where $\mathcal{M}_{l \times n}(R')$ is the set of $l \times n$ matrices over a ring $R'$.

**Definition 3.2.1** A multiset on $R_a^n$ is a mapping

$$m : R_a^n \rightarrow \mathbb{N}_0.$$  

The quantity $\sum_{x \in R_a^n} m(x)$ is called the size of the multiset. If $m(x) \in \{0, 1\}$ for all $x \in R_a^n$, then $m$ can be viewed as a subset. A multiset of size $p$ can be also written as a tuple $(r_1, r_2, \ldots, r_p), r_i \in R_a^n$ where $r_i$'s need not be all distinct.

The **Gray weight enumerator** of a multiset $C$ of $R_a^n$ is defined as follows

$$GW_C(y) = \sum_{c \in C} y^{w_G(c)}.$$  \hfill (3.18)

The Hamming weight of a codeword $c$, $w_H(c)$, is the number of nonzero entries of $c$. Also, the Hamming weight of a codeword can be defined to be the sum of the weights of its components, say $w_H(c) = \sum_{i=1}^n w_H(c_i)$ where $w_H(c_i)$ is zero if $c_i = 0$ and 1 otherwise. The Hamming distance of two codewords $c, e$, denoted by $d_H(c, e)$, is the Hamming weight of their difference, i.e. $d_H(c, e) = w_H(c - e)$.

**Remark**: If $r + us \in R_a$ then, we have

$$w_G(r + us) = w_H(r) + w_H(s).$$  \hfill (3.19)

**Lemma 3.2.1** The Gray map $\phi$ is a distance preserving map from

$$(R_a^n, \text{Gray distance})$$

to $$(R_{2n}^n, \text{Hamming distance})$, i.e.

$$d_G(c, e) = d_H(\phi(c), \phi(e))$$  \hfill (3.20)

where $c, e \in R_a^n$.  

107
Proof: Let \( c, e \in R^n_a \). Then, \( c = (c_1, \ldots, c_n) \) where \( c_i = r_i + u s_i \in R_a \) for some \( r_i, s_i \in R \), and \( e = (e_1, \ldots, e_n) \) where \( e_i = g_i + u h_i \in R_a \) for some \( g_i, h_i \in R \) and \( 1 \leq i \leq n \). Then, by the definitions above, the left hand side of (3.20),

\[
d_G(c, e) = w_G(c - e) = \sum_{i=1}^{n} w_G(r_i - g_i + u(s_i - h_i))
\]

By the definition of \( \phi \) we have

\[
\phi(c) = (\alpha(r_1 + u s_1), \alpha(r_2 + u s_2), \ldots, \alpha(r_n + u s_n))
\]

\[
= (s_1, r_1, s_2, r_2, \ldots, s_n, r_n) \in R^{2n}
\]

\[
\phi(e) = (\alpha(g_1 + u h_1), \alpha(g_2 + u h_2), \ldots, \alpha(g_n + u h_n))
\]

\[
= (h_1, g_1, h_2, g_2, \ldots, h_n, g_n) \in R^{2n}
\]

and the right hand side of (3.20),

\[
d_H(\phi(c), \phi(e)) = d_H((s_1, r_1, \ldots, s_n, r_n), (h_1, g_1, \ldots, h_n, g_n))
\]

\[
= w_H(s_1 - h_1, r_1 - g_1, \ldots, s_n - h_n, r_n - g_n)
\]

\[
= \sum_{i=1}^{n} (w_H(s_i - h_i) + w_H(r_i - g_i))
\]

which is equal to the left hand side, and hence the result. □

Theorem 3.2.1 Let \( C \) be an \((n, M)\)-linear code over \( R_a \) with \( d_G(C) = d \). Then, \( \phi(C) \) is a \((2n, M)\)-linear code over \( R \) with \( d_H(C) = d \).

Proof: By Lemma 3.2.1, we easily see that \( d_H(\phi(C)) = d_G(C) \). Also by the definition of \( \phi \), elements of \( C \) have length \( 2n \). We show that \( \phi(C) \) is an \( R \)-submodule
of $R^{2n}$. Let $c, e \in R_a$, and if we use the same notation as in the above Lemma 3.2.1, then we have

$$\phi(c + e) = \phi((c_1 + e_1, \ldots, c_n + e_n)) = (\alpha(c_1 + e_1), \ldots, \alpha(c_n + e_n))$$
$$= (\alpha(r_1 + g_1 + u(s_1 + h_1)), \ldots, \alpha(r_n + g_n + u(s_n + h_n)))$$
$$= (s_1 + h_1, r_1 + g_1, \ldots, s_n + h_n, r_n + g_n)$$
$$= (s_1, r_1, \ldots, s_n, r_n) + (h_1, g_1, \ldots, g_n, h_n)$$
$$= (\alpha(c_1), \ldots, \alpha(c_n)) + (\alpha(e_1), \ldots, \alpha(e_n))$$
$$= \phi(c) + \phi(e).$$

And if $y \in R, c \in C$, then

$$\phi yc = \phi(y(c_1, c_2, \ldots, c_n)) = \phi((y(r_1 + us_1), \ldots, y(r_n + us_n)))$$
$$= (ys_1, yr_1, ys_2, yr_2, \ldots, ys_n, yr_n) = y(s_1, r_1, s_2, r_2, \ldots, s_n, r_n)$$
$$= y\phi((c_1, c_2, \ldots, c_n)) = y\phi(c).$$

Since $\phi$ is injective, $\phi(C)$ has the same size as $C$. □

**Lemma 3.2.2** If $G_{k \times n}$ is a generator matrix of a code $C$ of full rank $k$ over $R_a$, then

$$\begin{bmatrix}
\phi(G) \\
\phi(uG)
\end{bmatrix}_{2k \times 2n}$$

(3.21)

is a generator matrix for $\phi(C)$, where $uG$ is a matrix obtained by multiplying the rows of $G$ by $u$ and $\phi$ was defined by (3.17).
Proof: Let \( v_1, v_2, \ldots, v_k \) be the row vectors of \( G \) which are linearly independent over \( \mathbb{R}_a \). We claim that
\[
\phi(v_1), \ldots, \phi(v_k), \phi(uv_1), \ldots, \phi(uv_k)
\]
are linearly independent over \( \mathbb{R} \). On the contrary assume not, then there exist \( \alpha_i \in \mathbb{R} \) for \( 1 \leq i \leq 2k \), not all zeros, such that
\[
\alpha_1 \phi(v_1) + \cdots + \alpha_k \phi(v_k) + \alpha_{k+1} \phi(uv_1) + \cdots + \alpha_{2k} \phi(uv_k) = 0.
\]
Since \( \phi \) is \( \mathbb{R} \)-linear, as shown in the proof of Theorem 3.2.1, we have
\[
\phi(\alpha_1 v_1 + \cdots + \alpha_k v_k + u\alpha_{k+1} v_1 + \cdots + u\alpha_{2k} v_k) = 0
\]
and since \( \phi \) is injective, we get
\[
\alpha_1 v_1 + \cdots + \alpha_k v_k + u\alpha_{k+1} v_1 + \cdots + u\alpha_{2k} v_k = 0
\]
\[
(\alpha_1 + u\alpha_{k+1})v_1 + (\alpha_2 + u\alpha_{k+2})v_2 + \cdots + (\alpha_k + u\alpha_{2k})v_k = 0.
\]
Now since \( v_1, \ldots, v_k \) are linearly independent over \( \mathbb{R}_a \), we must have \( \alpha_i + u\alpha_i = 0 \) for \( 1 \leq i \leq k \) and therefore \( \alpha_i = 0.1 \leq i \leq 2k \). This is a contradiction.

Hence \( \phi(v_1), \ldots, \phi(v_k), \phi(uv_1), \ldots, \phi(uv_k) \) generate a module over \( \mathbb{R}^{2n} \) which is a submodule of \( \phi(C) \) of equal size. Therefore, the matrix (3.21) which consists of these rows generates \( \phi(C) \). \( \square \)

Corollary 3.2.1 If \( C \) is an \([n, k, d]\) linear code over \( \mathbb{R}_a \) with respect to \( d_G \), then \( \phi(C) \) is a \([2n, 2k, d]\) linear code over \( \mathbb{R} \) with respect to \( d_H \).
Proof: We only need to show that $\phi(C)$ is a $2k$-free submodule. Since $C$ has a generator matrix of full rank $k$, then by Lemma 3.2.2, $\phi(C)$ will be generated by a matrix of full rank $2k$ and size $2k \times 2n$. Hence, $\phi(C)$ is $2k$-free.

Moreover if $R = F_q$ is a field, then we have the following:

Corollary 3.2.2 If $C$ is an $(n, |F_q|^{2k}, d)$ linear code over $R_u = F_q + uF_q$ with respect to $d_G$, then $\phi(C)$ is a $[2n, 2k, d]$ linear code over $F_q$ with respect to $d_H$.

Proof: We need only to show that $\phi(C)$ has dimension $2k$ but this is clear from Corollary 3.2.1 since $\phi(C)$ is a $2k$-free module over a field $F_q$. Hence, $\phi(C)$ has dimension $2k$. □

A special case, $q = 3$, of the above corollary is proven in [24].

By Theorem 3.2.1, $\phi(C)$ is a linear code and by Lemma 3.2.1 we have the following corollary:

Corollary 3.2.3 The Hamming weight enumerator of $\phi(C)$ is the same as the Gray weight enumerator of $C$.

Definition 3.2.2 Let $C$ be an $(n, M)$-linear code over a ring $R$ and $I_n = \{1, 2, \ldots, n\}$ (index set of coordinate positions of codewords). We think of a codeword $c$ as a mapping

$$c : I_n \rightarrow R.$$ 

For fixed $i \in I_n$, the punctured code $C_i$ of $C$ at the position $i$ has $I_n \setminus \{i\}$ as the set of coordinate positions and the new code $C_i$ is given by

$$C_i = \{c' : I_n \setminus \{i\} \rightarrow R, \text{ where } c'(j) = c(j) \text{ for all } j \in I_n \setminus \{i\}\}$$
and here $c = (c(1), c(2), \ldots, c(n)) \in C$. Intuitively, we just delete the $i^{th}$ coordinate position of the former code to get a new punctured code $C_i$.

The following Lemma is straightforward.

**Lemma 3.2.3** (Puncturing) Let $C$ be an $(n, M)$-linear code with $\rho(C) = d \geq 3$. Then the punctured code $C^*$ has the following parameters:

$$d^* = \begin{cases} d^* = d & \text{or } d^* = d - 1, \\ d^* = d & \text{or } d^* = d - 1 \text{ or } d^* = d - 2. \end{cases} \quad \text{if } \rho = d_H \quad \text{if } \rho = d_G$$

i.e. the minimum Hamming (Gray) distance of the punctured code is less at most 1 (2).

**Definition 3.2.3** We assume the same notations as in the previous lemma. Let $C$ be an $(n, M)$-linear code over a ring $R$ with $\rho(C) = d$ and coordinate positions $I_n$. Let $j \in I_n$ be such that there exists a codeword $e \in C$, where $e(j)$ is a unit (invertible element) of $R$. Then we define

$$C^0_j := \{ c \in C | c(j) = 0 \}$$

and let $C^j := (C^0_j)_j$. We call $C^j$ a shortening of $C$ at the position $j$. Note that not all $j \in I_n$ may afford a shortened code.

For $u \in R^n$, $$u + C := \{ u + c | c \in C \}.$$
Lemma 3.2.4 (Shortening) If $C$ is an $(n, M)$-linear code over a ring $R$ with $|R| = r$ and $\rho(C) = d$ and $|C'^0| > 1$, then $C'$, defined above, is an $(n', M')$-linear code with the following parameters

\[ n' = n - 1 \quad M' = M/r \quad d' \geq d. \]

Proof: By definition there exists a code $e \in C$ where $e(j)$ is a unit in $R := \{ \beta_0 = 0, \beta_1 = 1, \ldots, \beta_{r-1} \}$. Since $C'^0$ is a subgroup of $C$, $C$ can be partitioned into cosets each of equal size. We claim that $C = C'^0 \cup (e + C'^0) \cup (\beta_2 e + C'^0) \cup \cdots \cup (\beta_{r-1} e + C'^0)$ where the cosets are disjoint. If $\beta_i e + C'^0 = \beta_k e + C'^0$, then $\beta_i e - \beta_k e \in C'^0$ and $\beta_i e(j) = \beta_k e(j)$ so $\beta_i = \beta_k$. Also for an arbitrary codeword $c \in C \setminus C'^0$, we have $c(j) = \beta_s e(j)$ for some $s \neq 0$ which implies that $\beta_s e - c \in C'^0$. Thus, $|C'^0| = M/r$. Also, it is clear that the minimum distance is at least $d$ due to the definition of $C'^0$. Since $C'$ is obtained by deleting $j^{th}$ coordinate of $C'^0$ which is always zero, $C'$ has the same parameters as $C'^0$ except the length is reduced by 1. □

Note that Definition 3.2.3 we ask for a codeword having a unit element in the $j^{th}$ position in order to be able to determine the exact size of the shortened code.

3.3 New Linear Codes Over $\mathbb{F}_3 + u\mathbb{F}_3$ and $\mathbb{F}_5 + u\mathbb{F}_5$

To simplify the representation of the elements of $R + uR$ and generator matrices of quasi and nearly quasi cyclic codes we introduce the following notations and maps:

\[ lsu := l + us \text{ where } l, s \in R \text{ and hence } lsu \in Ra = R + uR. \quad (3.22) \]

113
Define

$$\mu_{k,m} : \mathcal{M}_{1 \times m}(R_a) \to \mathcal{M}_{k \times m}(R_a)$$

$$\mu_{k,m}((a_0 \ldots a_{m-1})) = \text{citr}_k(a_0, \ldots, a_{m-1}) = A_0$$

and

$$\sigma_{k,m,l}((a_0 a_1 \ldots a_{m-1}) \ldots (a_{l-1}m \ldots a_{m-1}))$$

$$= [\mu_{k,m}((a_0 \ldots a_{m-1})), \ldots, \mu_{k,m}((a_{l-1}m \ldots a_{m-1})])$$

$$= [A_0, A_1, \ldots, A_l] = A$$

Note that $\sigma_{k,m,l}$ maps the first row of $A$ to $A$.

**Example:**

We take the following generator matrix over $\mathbb{F}_3 + u\mathbb{F}_3$ where $u^2 = 1$ of a linear code say $C_{20}$:

$$G_{20} = \begin{bmatrix}
10u & 00u & 00u & 00u & 00u & 01u & 21u & 11u & 10u & 00u \\
00u & 10u & 00u & 00u & 00u & 00u & 01u & 21u & 11u & 10u \\
00u & 00u & 10u & 00u & 00u & 10u & 00u & 01u & 21u & 11u \\
00u & 00u & 00u & 10u & 00u & 11u & 10u & 00u & 01u & 21u \\
00u & 00u & 00u & 00u & 10u & 21u & 11u & 10u & 00u & 01u
\end{bmatrix}$$

(3.25)

Using notation (3.24), we can represent the matrix above, as follows:

$$G_{20} = \sigma_{5,5,2}((10u 00u 00u 00u 00u), (01u 21u 11u 10u 00u)).$$

114
This generator matrix generates a linear code which is equivalent to a $2$-$QC$ code. The Gray weight enumerator of the code generated by $G_{20}$ over $\mathbb{F}_3 + u\mathbb{F}_3$ is

$$W(y) = 1 + 200y^7 + 720y^8 + 1340y^9 + 3204y^{10} + 5520y^{11} + 8880y^{12} + 11040y^{13} + 10800y^{14} + 8400y^{15} + 5200y^{16} + 2700y^{17} + 900y^{18} + 120y^{19} + 24y^{20}.$$  

Using Corollary 3.2.3, we have a $[20,10,7]_3$ code, which is optimal.

Now we give the generator matrices of new codes over $\mathbb{F}_l + u\mathbb{F}_l$ where $l \in \{3,5\}$, which generate a code over $\mathbb{F}_l + u\mathbb{F}_l$ and by Gray map we obtain new linear codes that have better minimum distances.

### 3.3.1 New codes and their generator matrices

If $C$ is a QC code, then so is $\phi(C)$, where $\phi$ is the map defined in (3.2). So, the new codes that we found are equivalent to QC codes or nearly QC codes.

Similar to the example we are going to use the notation (3.24) for generator matrices which are given over $\mathbb{F}_l + u\mathbb{F}_l$ where $l = 3, 5$. However, by Lemma 3.2.2, we can easily construct the generator matrices of these codes over $\mathbb{F}_5$ or $\mathbb{F}_3$. Also we can use this lemma to find the generator matrices over $\mathbb{F}_3$ of new codes introduced in [24]. For instance, let us consider the generator matrix given in the above example. According to Lemma 3.2.2, the generator matrix of the ternary code $\phi(C_{20})$ is as follows:

$$\begin{bmatrix}
\phi(G_{20}) \\
\phi(uG_{20})
\end{bmatrix}_{10 \times 20}$$
New codes with improved minimum distances and their generator matrices:

1. A $[40, 10, 21]_5$ linear code:

   $\sigma_{5,5,4}((00u 24u 01u 03u 40u), (43u 21u 21u 23u 22u),
   (22u 33u 10u 30u 41u), (13u 31u 24u 43u 11u)).$

2. A $[48, 10, 25]_5$ linear code:

   $\sigma_{5,6,4}((00u 04u 22u 40u 34u 12u), (10u 11u 22u 34u 24u 22u),
   (11u 10u 20u 10u 34u 23u), (34u 34u 03u 02u 12u 11u)).$

3. A $[50, 10, 28]_5$ linear code:

   $\sigma_{5,5,5}(
   (04u 00u 04u 22u 40u), (34u 03u 33u 11u 22u), (34u 24u 22u 11u 10u),
   (20u 10u 34u 23u 00u), (34u 34u 02u 12u 11u)).$
4. A $[60, 10, 35]_5$ linear code:

\[ \sigma_{5,5,6}( \\
(11u 41u 12u 11u 01u), (03u 01u 34u 11u 12u), (23u 02u 20u 44u 03u), \\
(10u 40u 23u 21u 21u), (20u 30u 34u 12u 10u), (12u 20u 20u 34u 32u) ). \]

5. A $[70, 10, 42]_5$ linear code:

\[ \sigma_{5,5,7}( \\
(01u 00u 12u 11u 01u), (01u 00u 00u 11u 12u), (12u 02u 20u 11u 23u), \\
(10u 10u 11u 21u 21u), (20u 20u 10u 01u 10u), (12u 20u 04u 31u 23u), \\
(34u 02u 02u 01u 23u) ). \]

6. A $[48, 12, 24]_5$ linear code:

\[ \sigma_{5,6,4}((00u 04u 22u 40u 34u 12u), (00u 11u 22u 34u 24u 22u), \\
(11u 10u 12u 10u 34u 23u), (34u 34u 03u 02u 03u 11u)). \]

7. A $[42, 10, 21]_5$ linear code:

\[ \sigma_{5,7,3}((11u 00u 13u 24u 03u 34u 00u), (32u 22u 01u 11u 01u 40u 10u), \\
(04u 34u 12u 03u 02u 12u 12u)). \]

8. A $[80, 10, 49]_5$ linear code:

\[ \sigma_{5,5,8}( \\
(00u 02u 11u 01u 01u), (01u 12u 12u 11u 01u), (01u 00u 42u 11u 12u), \\
(12u 02u 20u 11u 03u), (10u 10u 11u 24u 21u), (20u 20u 00u 12u 13u), \\
(12u 20u 04u 31u 04u), (34u 02u 02u 01u 23u)). \]
9. A $[56, 14, 23]_3$ linear code:

$$\sigma_{4,7,4}((11u 12u 12u 10u 01u 01u 00u), (10u, 11u 12u 11u 12u 02u 10u), (12u 20u 11u 22u 01u 02u 22u), (02u 12u 11u 10u 00u 11u 12u)).$$

Using the above results, we have the following theorem:

**Theorem 3.3.1**

$$21 \leq d_5(40, 10) \leq 25, \quad 21 \leq d_5(42, 10) \leq 26, \quad 28 \leq d_5(50, 10) \leq 33.$$  
$$35 \leq d_5(60, 10) \leq 40, \quad 42 \leq d_5(70, 10) \leq 49, \quad 24 \leq d_5(56, 14) \leq 28.$$  
$$49 \leq d_5(80, 10) \leq 57, \quad and \quad 23 \leq d_5(56, 14) \leq 28.$$  

We will use the notation $P^iS^j \rightarrow m$ for puncturing $i$ times and then shortening $j$ times the new codes obtained in the previous section with number $m = \{1, 2, \ldots, 9\}$.

Now using the previous theorem and Lemma 3.2.3 (puncturing) and Lemma 3.2.4 (shortening) where $R = \mathbb{F}_5, \mathbb{F}_3$ successively, we obtain the following improvements in the bounds:

**Corollary 3.3.1**

$$20 \leq d_5(39, 10) \leq 24 \quad by\ P \rightarrow 1 \quad 19 \leq d_5(38, 10) \leq 24 \quad by\ P^2 \rightarrow 1.$$  
$$18 \leq d_5(37, 10) \leq 23 \quad by\ P^3 \rightarrow 1 \quad 27 \leq d_5(49, 10) \leq 32 \quad by\ P \rightarrow 3.$$  
$$26 \leq d_5(48, 10) \leq 31 \quad by\ P^2 \rightarrow 3 \quad 25 \leq d_5(47, 10) \leq 30 \quad by\ P^3 \rightarrow 3.$$  
$$24 \leq d_5(46, 10) \leq 30 \quad by\ P^4 \rightarrow 3 \quad 23 \leq d_5(45, 10) \leq 29 \quad by\ P^5 \rightarrow 3.$$  

118
3.3.2 Gray weight enumerators of new codes

We wrote a search program in C++ to search for good generator matrices over $\mathbb{F}_1 + u\mathbb{F}_1, l = \{3, 5\}$ and to compute the Gray weight enumerators of these codes. In this
process we made sure that the generator matrices generate a free submodule. In the search, the following observation also played an important role:

**Lemma 3.3.1** Let $GW_C(y) = \sum_i A_i y^i$ be the Gray weight enumerator of a set (multiset) $C$ generated (all possible linear combinations of the rows of $G$ over $R_n$) by $G$, a $k \times n$ matrix over $R_n$. $G$ generates a $k$-free submodule if and only if $A_0 = 1$.

**Proof:** $G$ generates a $k$-free submodule if and only if the rows of $G$ are linearly independent. This is possible if and only if $A_0 = 1$. \[\square\]

Let $W_C(y) = \sum_i B_i y^i$ be the Hamming weight enumerator of a code $C$. By Corollary 3.2.3, $A_i = B_i$ for all $i$.

The weight enumerators of new codes:

1. The weight enumerator of $[56,14,23]_3$:

$$W(y) = 1 + 308y^{23} + 672y^{24} + 1904y^{25} + 4186y^{26} + 8428y^{27} + 19392y^{28} + 35728y^{29} + 66318y^{30} + 110320y^{31} + 168406y^{32} + 247856y^{33} + 333928y^{34} + 428680y^{35} + 491036y^{36} + 537096y^{37} + 531580y^{38} + 492912y^{39} + 415828y^{40} + 325948y^{41} + 234320y^{42} + 153160y^{43} + 87892y^{44} + 48860y^{45} + 23940y^{46} + 9016y^{47} + 3864y^{48} + 928y^{49} + 392y^{50} + 56y^{51} + 14y^{52}.$$

For simplification weight enumerators, $W(y)$, of codes will be presented by bases and exponents where bases will correspond to possible weights of codewords and exponents will correspond to the number of codewords having that particular weight.
2. The weight enumerator of $[70,10,42]_5$:

$$W(y) = y^{42} 42^{680} 43^{2360} 44^{3880} 45^{8960} 46^{19960} 47^{11040} 48^{80260} 49^{140100} 50^{236240}$$

$$51^{366120} 52^{542100} 53^{739280} 54^{928920} 55^{1081328} 56^{1157880} 57^{1132840} 58^{1023980} 59^{830000}$$

$$60^{603396} 61^{400480} 62^{229480} 63^{115640} 64^{53500} 65^{19200} 66^{6260} 67^{1160} 68^{480}.$$ 

3. The weight enumerator of $[42,10,21]_5$:

$$W(y) = y^{10} 21^{88} 22^{336} 23^{1456} 24^{41300} 25^{12216} 26^{32224} 27^{75864} 28^{164700}$$

$$29^{314312} 30^{530556} 31^{49040} 32^{159636} 33^{415112} 34^{96836} 35^{967632}$$

$$36^{1064136} 37^{691312} 38^{364032} 39^{147752} 40^{142220} 41^{8712} 42^{1152}.$$ 

4. The weight enumerator of $[80,10,49]_5$:

$$W(y) = y^{50} 49^{560} 50^{960} 51^{2840} 52^{7020} 53^{14880} 54^{30820} 55^{58240} 56^{97880} 57^{168080} 58^{266600}$$

$$59^{391480} 60^{555964} 61^{728800} 62^{900000} 63^{1016680} 64^{1083960} 65^{1055888}$$

$$66^{973100} 67^{811160} 68^{620020} 69^{32160} 70^{276332} 71^{19280} 72^{76960} 73^{33320}$$

$$74^{13020} 75^{12240} 76^{1100} 77^{280}.$$ 

5. The weight enumerator of $[48,12,24]_5$:

$$W(y) = y^{50} 49^{816} 25^{2146} 26^{8616} 27^{28672} 28^{82388} 29^{225312} 30^{57912} 31^{133596} 32^{2876400} 33^{3512208}$$

$$34^{2180576} 35^{1503912} 36^{2621212} 37^{29300496} 38^{33098496} 39^{34829648} 40^{31378356} 41^{2412944}$$

$$42^{16338720} 43^{9109632} 44^{134648} 45^{1469440} 46^{81336} 47^{65712} 48^{6676}.$$ 

6. The weight enumerator of $[40,10,21]_5$:

$$W(y) = y^{10} 21^{520} 22^{2260} 23^{7720} 24^{18780} 25^{8528} 26^{11000} 27^{233680} 28^{432260} 29^{712640} 30^{1050080}$$

$$31^{1360120} 32^{1521100} 33^{1472280} 34^{1218300} 35^{837448} 36^{49760} 37^{200560} 38^{6480} 39^{12680} 40^{1068}.$$
7. The weight enumerator of $[48, 10, 25]_5$:

$$W(y) = 0^1 25^{120} 26^{340} 27^{1168} 28^{3536} 29^{9216} 30^{23350} 31^{52960} 32^{115120} 33^{220416} 34^{392400} 35^{623080}$$

$$36^{904536} 37^{1175360} 38^{1363116} 39^{1389032} 40^{1255996} 41^{976648} 42^{653628} 43^{361456}$$

$$44^{166752} 45^{58788} 46^{15716} 47^{2624} 48^{276}.$$ 

8. The weight enumerator of $[50, 10, 28]_5$:

$$W(y) = 0^1 28^{900} 29^{1880} 30^{6604} 31^{15360} 32^{36900} 33^{79280} 34^{159920} 35^{295240} 36^{482560} 37^{754800}$$

$$38^{1012820} 39^{1237240} 40^{1371024} 41^{1326320} 42^{1145300} 43^{850560} 44^{541120} 45^{284076} 46^{127680}$$

$$47^{12720} 48^{10840} 49^{1440} 50^{200}.$$ 

9. The weight enumerator of $[60, 10, 35]_5$:

$$W(y) = 0^1 35^{840} 36^{1080} 37^{14760} 38^{11980} 39^{27720} 40^{58840} 41^{111480} 42^{196560} 43^{341280}$$

$$44^{526880} 45^{71968} 46^{968440} 47^{1145000} 48^{1251080} 49^{1226200} 50^{1073532} 51^{844840} 52^{584720}$$

$$53^{353440} 54^{183900} 55^{78360} 56^{27160} 57^{8760} 58^{1820} 59^{200} 60^{84}.$$ 

122
3.4 New Linear Codes Over $\mathbb{F}_5$ Obtained by Tripling Method

In this section, we further generalize the method introduced in the previous section and obtain new linear codes over $\mathbb{F}_5$ which improve the bounds of the best known linear codes [4], [43]. We want to point out that with the methods introduced in both [24] and the previous section we were able to attack the problem of finding new linear codes over $\mathbb{F}_q$ for even length ($> 2$) codes only. Here, with a new generalization, we are able to search for linear codes with arbitrary lengths except when the length of a code is a prime number.

3.4.1 Codes over rings of type $R + uR + \cdots + u^{m-1}R$

Our notation will follow closely the previous section. Let $R$ be a commutative finite ring. Now we consider the quotient $R_a = R[u]/(u^m - a)$, where $a \in R, m \in \mathbb{N}$ and $u$ is an indeterminate, i.e.

$$R_a = \{r_0 + r_1 u + \cdots + r_{m-1} u^{m-1} | u^m = a \text{ and } r_i \in R, 0 \leq i \leq m - 1\}.$$

We will use the following short-hand notation:

$$u = r_0 + r_1 u + \cdots + r_{m-1} u^{m-1}.$$

We define a Gray weight of an element $r \in R_a$ and denote it as follows:

$$w_G(r) = \sum_{i=0}^{m-1} w_H(r_i) \quad (3.26)$$
where

\[
w_H(x) = \begin{cases} 
1, & \text{if } x \neq 0 \\
0, & \text{otherwise.}
\end{cases} \tag{3.27}
\]

for all \( x \in R \). The Gray weight of an \( n \)-tuple \( c = (c_1, c_2, \ldots, c_n) \in \mathbb{R}_n^m \) is the sum of the Gray weights of each component i.e \( w_G(c) = \sum_{i=1}^{n} w_G(c_i) \). Also, the Gray distance between two codewords \( c \) and \( e \) is \( d_G(c, e) = w_G(c - e) \). In this case also it can be easily checked that Gray distance is a metric. The minimum Gray weight of a code is the smallest nonzero Gray weight among all its codewords. For a linear code the minimum Gray distance denoted by \( d_G(C) \) equals to the minimum Gray weight of the code \( C \).

Now we relate the elements of \( R_a \) and \( \mathbb{R}^n \) by the following map

\[
\alpha : R_a \rightarrow \mathbb{R}^m
\]

\[
\alpha(r) = (\alpha_0(r), \alpha_1(r), \ldots, \alpha_{m-1}(r)) \tag{3.28}
\]

where \( \alpha_i(r) = r_i \) where \( 0 \leq i \leq m - 1 \). Note that \( \alpha_i \) is a projection map. We will also denote \( R_a \) by \( R + uR + \cdots + u^{m-1}R \).

We define a Gray map \( \phi : \mathbb{R}_a^m \rightarrow \mathbb{R}^{mn} \) as

\[
\phi(c) = (\alpha_0(c_1), \ldots, \alpha_{m-1}(c_1), \ldots, \alpha_0(c_n), \ldots, \alpha_{m-1}(c_n)) = (\alpha(c_1), \alpha(c_2), \ldots, \alpha(c_n)) \tag{3.29}
\]

where \( c = (c_1, c_2, \ldots, c_n) \). We observe that \( \phi \) is an injective map.
We naturally define an extension mapping

$$\phi : M_{k \times n}(R_a) \to M_{k \times mn}(R)$$

(3.30)

$$\begin{bmatrix}
    c_1 \\
    c_2 \\
    \vdots \\
    c_k \\
\end{bmatrix}_{(k \times n)} \mapsto \begin{bmatrix}
    \phi(c_1) \\
    \phi(c_2) \\
    \vdots \\
    \phi(c_k) \\
\end{bmatrix}_{k \times mn}$$

where $M_{l \times n}(R')$ is the set of $l \times n$ matrices over a ring $R'$.

**Lemma 3.4.1** The Gray map $\phi$ is a distance preserving map from

$$(R^n_a, \text{Gray distance})$$

to $$(R^{mn}, \text{Hamming distance})$$, i.e.

$$d_G(c, e) = d_H(\phi(c), \phi(e))$$

(3.31)

where $c, e \in R^n_a$.

**Proof:** Proof follows from definitions, as in the previous section. □

**Theorem 3.4.1** Let $C$ be an $(n, M)$-linear code over $R_a$ with $d_G(C) = d$. Then, $\phi(C)$ is an $(mn, M)$-linear code over $R$ with $d_H(C) = d$.

**Proof:** By Lemma 3.4.1, we easily see that $d_H(\phi(C)) = d_G(C)$. Also by the definition of $\phi$, elements of $C$ have length $mn$. We show that $\phi(C)$ is an $R$-submodule
of \( R^{mn} \). Let \( c = (c_1, \ldots, c_n), e = (e_1, \ldots, e_n) \in R_a \), where \( e_j = e_{j0} + e_{j1}u + \cdots + e_{(m-1)j}u^{m-1}, 0 \leq j \leq n \), then we have

\[
\phi(c + e) = \phi((c_1 + e_1, \ldots, c_n + e_n)) = (\alpha(c_1 + e_1), \ldots, \alpha(c_n + e_n))
\]

\[
= \alpha(\sum_{i=0}^{m-1} (c_{i1} + e_{i1})u^i), \ldots, \alpha(\sum_{i=0}^{m-1} (c_{i1} + e_{i1})u^i))
\]

\[
= (c_{10} + e_{10}, \ldots, c_{1m-1} + e_{1m-1}, \ldots, c_{n0} + e_{n0}, \ldots, c_{nm-1} + e_{nm-1})
\]

\[
= (c_{10}, \ldots, c_{1m-1}, \ldots, c_{n0}, \ldots, c_{n0}, \ldots, c_{nm-1}) + (e_{10}, \ldots, e_{1m-1}, \ldots, e_{n0}, \ldots, e_{nm-1})
\]

\[
= (\alpha(c_1), \ldots, \alpha(c_n)) + (\alpha(e_1), \ldots, \alpha(e_n))
\]

\[
= \phi(c) + \phi(e).
\]

And if \( y \in R, c \in C \), then

\[
\phi(yc) = \phi((yc_1, \ldots, yc_n)) = \phi(\sum_{i=0}^{m-1} yc_{i1}, \ldots, \sum_{i=0}^{m-1} yc_{i1})
\]

\[
= (yc_{10}, \ldots, yc_{1m-1}, \ldots, yc_{n0}, \ldots, yc_{nm-1})
\]

\[
= yc_{10}, \ldots, c_{1m-1}, \ldots, c_{n0}, \ldots, c_{n0}, \ldots, c_{nm-1})
\]

\[
= y\phi((c_1, c_2, \ldots, c_n)) = y\phi(c).
\]

Since \( \phi \) is injective, \( \phi(C) \) has the same size as \( C \). □

**Lemma 3.4.2** If \( G_{k \times n} \) is a generator matrix of a code \( C \) of full rank \( k \) over \( R_a \), then

\[
\begin{bmatrix}
\phi(G) \\
\phi(uG) \\
\vdots \\
\phi(u^{m-1}G)
\end{bmatrix}_{mk \times mn}
\]

(3.32)
is a generator matrix for $\phi(C)$, where $u^iG$ is a matrix obtained by multiplying the rows of $G$ by $u^i$, $0 \leq i \leq m - 1$ and $\phi$ was defined by (3.28) and (3.29).

**Proof:** Let $v_1, v_2, \ldots, v_k$ be the row vectors of $G$ which are linearly independent over $R_a$. We claim that

$$
\phi(v_1), \ldots, \phi(v_k), \phi(uv_1), \ldots, \phi(uv_k), \ldots, \phi(u^{m-1}v_1), \ldots, \phi(u^{m-1}v_k)
$$

are linearly independent over $R$. On the contrary assume not, then there exist $\alpha_i \in R$ for $1 \leq i \leq mk$, not all zero, such that

$$
\alpha_1 \phi(v_1) + \cdots + \alpha_k \phi(v_k) + \cdots + \alpha_{(m-1)k+1} \phi(u^{m-1}v_1) + \cdots + \alpha_{mk} \phi(u^{m-1}v_k) = 0.
$$

Since $\phi$ is $R$-linear, as shown in the proof of Theorem 3.4.1, we have

$$
\phi(\alpha_1 v_1 + \cdots + \alpha_k v_k + \cdots + u^{m-1} \alpha_{(m-1)k+1} v_1 + \cdots + u^{m-1} \alpha_{mk} v_k) = 0
$$

and since $\phi$ is injective, we get

$$
\alpha_1 v_1 + \cdots + \alpha_k v_k + \cdots + u^{m-1} \alpha_{(m-1)k+1} v_1 + \cdots + u^{m-1} \alpha_{mk} v_k = 0
$$

$$
\left( \sum_{j=0}^{m-1} u^j \alpha_{jk+1} \right) v_1 + \left( \sum_{j=0}^{m-1} u^j \alpha_{jk+2} \right) v_2 + \cdots + \left( \sum_{j=0}^{m-1} u^j \alpha_{jk+k} \right) v_k = 0.
$$

Now since $v_1, \ldots, v_k$ are linearly independent over $R_a$, we must have

$$
\sum_{j=0}^{m-1} u^j \alpha_{jk+1} = 0 \text{ for all } 1 \leq i \leq k,
$$

and on equating the coefficients of $u^j$ we get

$$
\alpha_{jk+i} = 0 \text{ for } 0 \leq j \leq m-1, 1 \leq i \leq k.
$$

Therefore $\alpha_i = 0, 1 \leq i \leq mk$. This is a contradiction.
Hence \( \phi(v_1), \ldots, \phi(v_k), \ldots, \phi(u^{m-1}v_1), \ldots, \phi(u^{m-1}v_k) \) generate a module over \( R^{mn} \) which is a submodule of \( \phi(C) \) of equal size. Therefore, the matrix (3.32) which consists of these rows generates \( \phi(C) \). □

**Corollary 3.4.1** If \( C \) is an \([n, k, d]\) linear code over \( R \) with respect to \( d_C \), then \( \phi(C) \) is an \([mn, mk, d]\) linear code over \( R \) with respect to \( d_H \).

**Proof:** We only need to show that \( \phi(C) \) is an \( mk \)-free submodule. Since \( C \) has a generator matrix of full rank \( k \), then by Lemma 3.4.2, \( \phi(C) \) will be generated by a matrix of full rank \( mk \) and size \( mk \times mn \). Hence, \( \phi(C) \) is \( mk \)-free. □

Moreover if \( R = F_q \) is a field, then we have the following:

**Corollary 3.4.2** Let \( C \) be an \((n, |F_q|^{mk}, d)\) linear code over \( R = F_q + uF_q + \cdots + u^{m-1}F_q \) with respect to \( d_C \), then \( \phi(C) \) is an \([mn, mk, d]\) linear code over \( F_q \) with respect to \( d_H \).

**Proof:** We need only to show that \( \phi(C) \) has dimension \( 2k \) but this is clear from Corollary 3.4.1 since \( \phi(C) \) is a \( mk \)-free module over a field \( F_q \). Hence, \( \phi(C) \) has dimension \( mk \). □

By Theorem 3.4.1, \( \phi(C) \) is a linear code and by Lemma 3.4.1 we have the following corollary:

**Corollary 3.4.3** The Hamming weight enumerator of \( \phi(C) \) is the same as the Gray weight enumerator of \( C \).
3.4.2 New Linear Codes over $\mathbb{F}_5 + u\mathbb{F}_5 + u^2\mathbb{F}_5$

Our new results are obtained from codes over $\mathbb{F}_5 + u\mathbb{F}_5 + u^2\mathbb{F}_5$ where $u^3 = 1$. Hence we develop the following notations for $m = 3$ only.

To simplify the representation of the elements of $R + uR + u^2R$ we introduce the following notation:

$$r_0r_1r_2 := r_0 + r_1u + r_2u^2$$

where $r_0, r_1, r_2 \in R$. (3.33)

Note that if $C$ is a QC code, then so is $\phi(C)$, where $\phi$ is the map defined in (3.29). So, the new codes that we found in this section are equivalent to QC codes or are nearly QC codes.

In this research we wrote a search program in C++ similar to one that we used in the previous section.

The generator matrices and the weight enumerators of new codes over $\mathbb{F}_5 + u\mathbb{F}_5 + u^2\mathbb{F}_5$ are given below. Applying the Gray map we obtain new linear codes over $\mathbb{F}_5$ that have better minimum distances. We want to point out that by Lemma 3.4.2, we can construct the generator matrices of these codes over $\mathbb{F}_5$.

To represent the generator matrices of these new codes we use the functions $\mu_{k,m}$ (3.23) and $\sigma_{k,m,l}$ (3.24) introduced in the previous section.

The generator matrices and weight enumerators of new codes:

1. A $[36, 9, 20]_5$ linear code:

$$\sigma_{3,4,12}((000 \ 200 \ 401), (120 \ 031 \ 240), (310 \ 013 \ 430), (324 \ 431 \ 000)),$$
\[ W(y) = 0^1 20^{1332} 21^{4068} 22^{9288} 23^{21204} 24^{45336} 25^{94032} 26^{150336} 27^{229660} 28^{292608} \\
   29^{322164} 30^{302364} 31^{233100} 32^{145152} 33^{71688} 34^{424300} 35^{5796} 36^{696} . \]

2. A \([42, 9, 23]_5\) linear code:
\[
\sigma_{3,7,14} \left(\begin{array}{c}
(000 214 403 000 442 031333), \\
(340 320 341 320 042 30024)\
\end{array}\right),
\]

\[ W(y) = 0^1 23^{384} 24^{4856} 25^{2508} 26^{6084} 27^{15496} 28^{32820} 29^{63024} 30^{109616} 31^{171156} 32^{232980} \\
   33^{279592} 34^{300888} 35^{271740} 36^{214360} 37^{139116} 38^{72804} 39^{2632} 40^{3024} 41^{1800} 42^{208} . \]

3. A \([45, 9, 26]_5\) linear code:
\[
\sigma_{3,3,15} \left(\begin{array}{c}
(210 222 403), (340 042 030), (200 240 320), \\
(340 130 012), (430 012 441)\
\end{array}\right),
\]

\[ W(y) = 0^1 26^{684} 27^{2536} 28^{6048} 29^{12528} 30^{27684} 31^{53352} 32^{8064} 33^{144816} 34^{207144} 35^{259776} \\
   36^{286380} 37^{280260} 38^{2347720} 39^{170484} 40^{102888} 41^{47184} 42^{18720} 43^{5616} 44^{1074} 45^{76} . \]

4. A \([48, 9, 27]_5\) linear code:
\[
\sigma_{3,4,16} \left(\begin{array}{c}
(403 214 403 002), (442 032 214 240), \\
(321 303 320 331), (432 012 403 320)\
\end{array}\right),
\]

\[ W(y) = 0^1 27^{308} 28^{672} 29^{1800} 30^{4684} 31^{10680} 32^{22884} 33^{45220} 34^{76740} 35^{125016} 36^{181168} 37^{23168} \\
   38^{273204} 39^{278132} 40^{250828} 41^{195300} 42^{1293112} 43^{73296} 44^{33708} 45^{1500} 46^{1156} 47^{520} 48^{1} . \]

5. A \([54, 9, 32]_5\) linear code:
\[
\sigma_{3,3,18} \left(\begin{array}{c}
(111 411 403), (000 222 030), (422 240 320), \\
(333 320 204), (430 012 404), (020 012 043)\
\end{array}\right).
\]

130
\[ W(y) = 0^1 32 396 33 1500 34 3708 35 8280 36 15748 37 31896 38 53640 39 82556 40 140436 41 186156 42 234432 43 261792 44 259884 45 230088 46 180324 47 125604 48 72096 49 35352 50 13320 51 4872 52 972 53 72. \]

6. A \([63, 9, 38]_5\) linear code:

\[
\sigma_{3,3,21} (((000 223 104), (240 000 013), (330 230 110), (104 221 012), (210 014 202), (202 221 040), (240 402 330)),
\]

\[ W(y) = 0^1 38 33 996 40 1914 41 1752 42 2168 43 19548 44 31569 45 58092 46 89640 47 127980 48 174936 49 213156 50 243144 51 244188 52 221436 53 189360 54 136708 55 99532 56 52740 57 25689 58 10224 59 3780 60 912 61 180 62 36. \]

7. A \([72, 9, 45]_5\) linear code:

\[
\sigma_{3,3,24} (((000 111 403), (300u 222 110), (000 240 320), (333 320 331), (430 012 404), (023 212 030), (432 002 141), (400 432 444)),
\]

\[ W(y) = 0^1 45 684 46 1188 47 2736 48 5868 49 12168 50 21960 51 35868 52 58320 53 86580 54 121120 55 165708 56 196128 57 229392 58 234036 59 214956 60 184412 61 145080 62 107136 63 65988 64 38088 65 19260 66 6936 67 2304 68 28 69 688 70 12. \]

8. A \([75, 9, 46]_5\) linear code:

\[
\sigma_{3,3,25} (((210 111 403 300u 222), (030 000 240 320 333), (320 012 430 012 404), (023 312u 030 432 002), (141 212 400 432 444)),
\]

131
9. A \([81, 9.51]_5\) linear code:

\[
\sigma_{3,3,27}((213u 333 202), (100 402 014), (000 210 120),
(221 340 012), (203 012 000), (212 042 104),
(112 100 203), (330 032 222)),
\]

\[
W(y) = 0^1 46^264 47^528 48^1080 49^2328 50^4872 51^9656 52^17376 53^340564 54^50844 55^78984 56^108924
57^145880 58^182484 59^208572 60^221220 61^222696 62^200868 63^162792 64^121572 65^92836 66^51032
67^27540 68^12768 69^4984 70^1812 71^528 72^108 73^12.
\]

10. A \([90, 9.57]_5\) linear code:

\[
\sigma_{3,3,30}((130 141 103), (304 212 200), (031 240 120),
(000 320 144), (432 012 404), (023 240 032),
(432 302 141), (440 402 310), (320 234 302),
(411 403 012)),
\]

\[
W(y) = 0^1 51^516 52^972 53^1836 54^3828 55^7524 56^12744 57^23928 58^38232 59^58641 60^53100 61^115488
62^149904 63^183064 64^203580 65^215856 66^212880 67^188718 68^156636 69^118110 70^80316 71^50724
72^24504 73^13032 74^5508 75^1968 76^648 77^144.
\]
11. A $[99, 9, 65]_8$ linear code:

\[
\sigma_{3,3,3}((130 043 034), (000 140 403), (304 214 310),
(301 120 340), (412 012 401), (420 122 032),
(342 302 141), (401 212 340), (420 214 304),
(422 403 012), (114 401 334)).
\]

Using the above results, we have the following theorem.

**Theorem 3.4.2**

\[
20 \leq d_5(36, 9) \leq 23, \quad 23 \leq d_5(42, 9) \leq 23, \quad 26 \leq d_5(45, 9) \leq 30.
\]

\[
27 \leq d_5(48, 9) \leq 27, \quad 32 \leq d_5(54, 9) \leq 37, \quad 38 \leq d_5(63, 9) \leq 44.
\]

\[
45 \leq d_5(72, 9) \leq 51, \quad 46 \leq d_5(75, 9) \leq 51, \quad 51 \leq d_5(81, 9) \leq 59.
\]

\[
57 \leq d_5(90, 9) \leq 65, \quad and \quad 65 \leq d_5(99, 9) \leq 73.
\]

We point out that by shortening and puncturing these new codes further improvement on minimum distances are obtained.
CHAPTER 4
APPLICATIONS OF IN Variant THEORY

4.1 Ring of invariants of 2-ply weight enumerators of binary self-dual linear codes

In this section we are going to classify the 2-ply weight enumerator of a binary self-dual code. Similar classifications are done for Hamming weight enumerators [39], Lee weight enumerators [39], complete weight enumerators over $GF(3)$ [41], biweight enumerators of binary codes [29]. Self-dual codes over integers modulo 4 are investigated by Conway and Sloane in [8]. Also, extensive applications of invariant theory for several families is done in [43]. Recently, the ring of invariants of Hamming weight enumerators of codes over $Z_k$ (integers modulo k) was investigated by Harada and Oura [28].

We assume that $C$ is an $[n, k, d]$ binary linear code. Using the same notation as in the previous sections, we first state the identity (2.25) between the 2-ply weight enumerator of a code $C$ and its dual:

$$W^{(2)}_{C^\perp}(x, y) = \frac{1}{|C|^2}W^{(2)}_C(x + 3y, x - y)$$

(4.1)

where $W^{(2)}_C(x, y) = \sum_{u,v \in C} x^{n-s(u,v)}y^{s(u,v)}$ and
Let us recall the following notations: $F_2 = \{ \alpha_0 = 0, \alpha_1 = 1 \}$

$$s_{ij}(u, v) = |\{i|(u_i, v_i) = (\alpha_i, \alpha_j)\}, 1 \leq i \leq n\}$$ \hspace{1cm} (4.2)

where $\alpha_i, \alpha_j \in F_2, \quad 1 \leq i, j \leq 2$.

We observe that

$$s(u, v) = s_{01}(u, v) + s_{10}(u, v) + s_{11}(u, v). \quad (4.3)$$

**Example:** The 2-ply weight enumerator of the extended Hamming $[8, 4, 4]$ and the extended Golay code $[24, 12, 8]$ are given, respectively, as follows:

$$h_8(x, y) = x^8 + 42x^4y^4 + 168x^2y^6 + 45y^8. \quad (4.4)$$

and,

$$g_{24}(x, y) = x^{24} + 2277x^{16}y^8 + 29248x^{12}y^{12} + 3895947x^8y^{16} + 37631y^{24}$$
$$+ 1020096x^{10}y^{14} + 6120576x^6y^{18} + 4462920x^4y^{20} + 1020096x^2y^{22}. \quad (4.5)$$

In both $h_8(x, y)$ and $g_{24}(x, y)$, we see that the exponents of the variables $x$ and $y$ are even numbers. This is in general true for all 2-ply weight enumerators of binary self-dual codes.

**Lemma 4.1.1** If $C$ is an $[n, k, d]$ binary self-dual code then

$$s_{00}(u, v) \equiv s_{01}(u, v) \equiv s_{10}(u, v) \equiv s_{11}(u, v) \equiv 0 \quad (\text{mod } 2)$$

for all $u, v \in C$. 

135
Proof: Since $C = C^\perp$, $n \equiv 0 \pmod{2}$ and $\langle u, v \rangle \equiv 0 \pmod{2}$ for all $u, v \in C$. By definitions (1.1) and (4.2), we have $s_{11}(u, v) = \langle u, v \rangle \equiv 0 \pmod{2}$ for all $u, v \in C$. Also, note that $(u, u) = \sum u_i^2 = \text{wt}(u) \equiv 0 \pmod{2}$ for all $u \in C$. Since $\text{wt}(u) = s_{10}(u, v) + s_{11}(u, v)$, we have $s_{10}(u, v) \equiv 0 \pmod{2}$. Similar to the previous argument we get $s_{01}(u, v) \equiv 0 \pmod{2}$. Finally, note that $n = s_{00}(u, v) + s_{01}(u, v) + s_{10}(u, v) + s_{11}(u, v)$, hence $s_{00}(u, v) \equiv 0 \pmod{2}$. □

Corollary 4.1.1 If $C$ is a binary self-dual code then

$$s(u, v) \equiv n - s(u, v) \equiv 0 \pmod{2}$$

for all $u, v \in C$.

Proof: Note that by the above observation (4.3) and using Lemma 4.1.1, and the fact that $n \equiv 0 \pmod{2}$ we have the result. □

We will use the following notations:

$$p_2(x, y) := x^2 + 3y^2$$
$$p_6(x, y) := 33y^6 + 15x^4y^2 + 15x^2y^4 + x^6$$
$$p_4(x, y) := x^4 + 6x^2y^2 + 9y^4$$

\begin{equation}
\begin{aligned}
p_8(x, y) := & \quad 28x^2y^6 + 28x^6y^2 + 70x^4y^4 + 129y^8 + x^8 \\
p_{12}(x, y) := & \quad x^{12} + 6x^2y^{10} + 495x^8y^4 + 924x^6y^6 + 495x^4y^8 + 2049y^{12}.
\end{aligned}
\end{equation}

Note that the subindices correspond to the homogenous degrees of the polynomials.

Theorem 4.1.1 Let $C$ be a self-dual binary code, i.e $C = C^\perp$, then

$$W_C^{(2)}(x, y) \in \mathbb{C}[p_2, p_6].$$
Moreover, if the length of code is divisible by 4, i.e. \( n \equiv 0 \mod 4 \), then

\[
W^{(2)}_C(x, y) \in \mathbb{C}[p_4, p_{12}] \oplus p_6 \mathbb{C}[p_4, p_{12}].
\]

**Proof:** For the first part the only assumption is that the code is self-dual. So,

\[
W^{(2)}_C(x, y) = \frac{1}{|C|^2} W^{(2)}_C(x + 3y, x - y)
\]

since \( C = C^\perp \). Moreover, \(|C| = 2^{n/2}\), and

\[
W^{(2)}_C(x, y) = W^{(2)}_C \left( \frac{x + 3y}{2}, \frac{x - y}{2} \right). \tag{4.7}
\]

Hence, the 2-ply weight enumerator of self-dual code is invariant under the following transformation matrix, say

\[
M = \begin{pmatrix}
1/2 & 3/2 \\
1/2 & -1/2
\end{pmatrix}
\]

and also, using Corollary 4.1.1, we see that changing the variables \( x \) and \( y \) by \( \pm x \) and \( \pm y \) respectively, leaves the 2-ply weight enumerator invariant, i.e the 2-ply weight enumerator is also invariant under the following transformation matrices:

\[
\begin{pmatrix}
\pm 1 & 0 \\
0 & \pm 1
\end{pmatrix}
\]

Thus, the 2-ply weight enumerator of a binary linear code is invariant under the group, say \( G_1 \), generated by the above matrices. Using the GAP (software package
for group theoretical computations) [50], we found that $G_1$ is a group of order 12.

Carrying symbolic computations in MAPLE we found that the Molien series is

$$
\Phi_1(\lambda) = \frac{1}{g} \sum_{\lambda \in G_1} \frac{1}{|\lambda - \lambda I|} = \frac{1}{(1 - \lambda + \lambda^2)(1 + \lambda + \lambda^2)(1 + \lambda)^2(\lambda - 1)^2},
$$

After some algebraic manipulations, we get

$$
\Phi_1(\lambda) = \frac{1}{(1 - \lambda^6)(1 - \lambda^2)}
$$

which is a desired form of Molien series. By looking at the Molien series and using (1.3.7), we see that if we can find two algebraically independent polynomials and both invariant under $G_1$ of degrees 2 and 6, then we can identify $R(G_1)$.

Applying Theorem (1.3.6) to the function $f(x, y) = y^2$, i.e. averaging $f(x, y)$ over $G_1$, we get an invariant polynomial of degree 2 which is $2x^2 + 6y^2$. Since we can take any scalar multiple, for simplicity we take $x^2 + 3y^2$. that is $p_2(x, y)$ defined above. Taking the polynomial $(x + y)^6$ and averaging it, leads to another degree 6 invariant polynomial $254y^6 + 120x^4y^2 + 120x^2y^4 + 8x^6$ and dividing this by 8 gives $p_6(x, y)$. Now, in order to check whether $p_2(x, y)$ and $p_6(x, y)$ are algebraically independent we use (1.3.5). The Jacobian of the above polynomials is

$$
\begin{vmatrix}
\frac{\partial p_2(x, y)}{\partial x} & \frac{\partial p_6(x, y)}{\partial x} \\
\frac{\partial p_2(x, y)}{\partial y} & \frac{\partial p_6(x, y)}{\partial y}
\end{vmatrix} = 216xy^5 + 24x^5y - 240x^3y^3 + \ldots
$$

which is not zero, hence the above polynomials are algebraically independent. Thus, $R(G_1) = \mathbb{C}[p_2, p_6]$ which is the first part of the theorem.
In the second part of the theorem we further assume that \( n \equiv 0 \pmod{4} \). This implies that the 2-ply weight enumerator is homogenous of degree \( 4m \), for some positive integer \( m \). If \( s(u, v) = 0 \), then \( n - s(u, v) = n \) and is divisible by 4 or vice versa. If the exponents are never zero, after substituting both \( ix \) and \( iy \) simultaneously for \( x \) and \( y \) respectively, the 2-ply weight enumerator still remains the same. In other words, the 2-ply weight enumerator is invariant under the following transformation:

\[
J = \begin{pmatrix}
i & 0 \\
0 & i
\end{pmatrix}
\]

where \( i = \sqrt{-1} \). Now, the 2-ply weight enumerator is invariant under the group \( G_1 \) and the above matrix \( J \) which generate another group, say \( G_2 \), of order 24. The Molien series of \( G_2 \) is

\[
\Phi_2(\lambda) = \frac{1}{g} \sum_{A \in G_2} \frac{1}{|A - \lambda I|} = \frac{1 + \lambda^8}{(1 - \lambda^4)(1 - \lambda^{12})}.
\]

The form of the Molien series for \( G_2 \) suggests to look for 3 homogenous invariant polynomials of degrees 4, 12, and 8 where the ones with degrees 4 and 12 must be algebraically independent.

To get the following three invariants

\[
9x^4 + 81y^4 + 54x^2y^2, \quad 448x^6y^2 + 1120x^4y^4 + 448x^2y^6 + 16x^8 + 2064y^8, \quad \text{and}
\]

\[
16x^{12} + 1056y^{10}x^2 + 1056y^2x^{10} + 7920y^4x^8 + 1478y^6x^6 + 7920y^8x^4 + 32784y^{12}
\]

139
we averaged the polynomials

\[ x^4, \ (x - y)^8, \ \text{and} \ (x - y)^{12} \]

respectively, over \( G_2 \). If we divide the first by 9 and the last two polynomials by 16 we get \( p_4(x, y), p_8(x, y) \) and \( p_{12}(x, y) \) respectively. The Jacobian of \( p_4 \) and \( p_{12} \) is

\[
\left| \begin{array}{cc}
\frac{\partial p_4(x, y)}{\partial x} & \frac{\partial p_{12}(x, y)}{\partial x} \\
\frac{\partial p_4(x, y)}{\partial y} & \frac{\partial p_{12}(x, y)}{\partial y}
\end{array} \right| = -384x^{13}y + \cdots
\]

which is nonzero.

Hence, by (1.3.5), \( p_4 \) and \( p_{12} \) are algebraically independent. Using (1.3.7), we can conclude that \( \mathcal{R}(G_2) = \mathbb{C}[p_4, p_{12}] \oplus p_8 \mathbb{C}[p_4, p_{12}] \).

Further if we impose another restriction on the length \( n \) of binary self-dual codes, say \( n \equiv 0 \mod 8 \), then we obtain the ring of invariants of 2-ply weight enumerators of such codes in terms of 2-ply weight enumerators of extended Hamming and extended Golay codes. The class of self-dual codes with lengths divisible by 8 is an important class. Type II codes belong to this class.

**Theorem 4.1.2** Let \( C \) be a binary self-dual code with its length divisible by 8. Then

\[ W^{(2)}_C(x, y) \in p_8 \mathbb{C}[h_8, g_{24}] \oplus p_{10} \mathbb{C}[h_8, g_{24}] \oplus p_{24} \mathbb{C}[h_8, g_{24}] \oplus \mathbb{C}[h_8, g_{24}], \]
where $h_8$ and $g_{24}$ are the 2-ply weight enumerators of extended Hamming and Golay codes. The polynomial $p_8$ is given in equation (4.6) and

$$p_{16}(x, y) = 32769x^{16} + 472864392x^8y^{10} + 967222620x^4y^{12} + 57395680x^2y^{14} + 43046721y^{16},$$

$$p_{24}(x, y) = 94143178827y^{24} + 2790203x^{24} + 2887057484028y^{12}x^2 + 3126875549688y^{14}x^{10} + 17381749379148y^{18}x^6 + 479033122932y^{12}x^{12} + 828y^2x^{22} + 286902y^4x^{20} + 32706828y^6x^{18} + 1608475077y^8x^{16} + 38603401848y^{10}x^{14} + 10553204980197y^{16}x^8 + 12350190348342.$$

**Proof:** If $C$ is a binary self-dual code with length divisible by 4, then from the proof of the previous theorem we know that the 2-ply weight enumerator of such codes is invariant under the group $G_2$ of size 24. In this case, we have further assumed that the length of such codes is divisible by 8. The 2-ply weight enumerator is invariant under the following matrices:

$$\begin{pmatrix}
\pm e^{2\pi i/8} & 0 \\
0 & \pm e^{2\pi i/8}
\end{pmatrix}.$$

Hence, the 2-ply weight enumerator is invariant under a new group, say $G_3$ of order 48, which is generated by $G_2$ and the matrices given above.

The Molien series for $G_3$ is

$$\Phi_3(\lambda) = \frac{1}{g} \sum_{A \in G_3} \frac{|A|}{|A - \lambda I|} = \frac{1 + \lambda^8 + \lambda^{16} + \lambda^{24}}{(1 - \lambda^8)(1 - \lambda^{24})}.$$
Next we would like to have 2 free polynomials of degrees 8 and 24 and 3 transient polynomials of degrees 8, 16 and 24 by Theorem 1.3.7. The 2-ply weight enumerators of extended Hamming $h_8$ and Golay $g_{24}$ codes are left invariant under $G_3$. Further, the Jacobian of $g_{24}$ and $h_8$ is nonzero, hence, by Theorem 1.3.5 they are algebraically independent. Also, the polynomial $p_8$ is invariant under $G_3$. In order to find the other two transient polynomials $p_{16}$ and $p_{24}$, we have averaged the polynomials $x^{16}$ and $x^{24}$ respectively. □
4.2 Classification of complete weight enumerator of self-dual codes over $\mathbb{F}_2 + u\mathbb{F}_2$

4.2.1 Complete weight enumerators of codes over $\mathbb{F}_2 + u\mathbb{F}_2$

Commutative rings of order 4 are $\mathbb{F}_4$ (Galois field of order 4), $\mathbb{Z}_4$ (integers modulo 4), $\mathbb{F}_2[u]/(u^2 - 1)$, and $\mathbb{F}_2[u]/(u^2 - u)$. It is clear that the ring $\mathbb{F}_2[u]/(u^2 - 1)$ is isomorphic to the ring $\mathbb{F}_2[u]/(u^4)$ via the map $u \mapsto u + 1$. Most researches have considered codes over the rings of type $\mathbb{F}_2 + u\mathbb{F}_2$ with $u^2 = 0$. We are going to assume that our ring is $\mathbb{F}_2 + u\mathbb{F}_2$ with $u^2 = 1$ in order to be in accordance with the previous chapter.

For convenience let us use the following notation:

$$R := \mathbb{F}_2[u]/(u^2 - 1). \quad (4.8)$$

So a code $C$ of length $n$ over $R$ is an $R$ submodule of $R^n$. In the previous section we have considered the codes over $\mathbb{F}_3 + u\mathbb{F}_3$ and $\mathbb{F}_5 + u\mathbb{F}_5$ and via a Gray map we discovered new linear codes over $\mathbb{F}_3$ and $\mathbb{F}_5$ with improved minimum distances. Hence, it is important to establish further connections between a code over these rings and their images under Gray map. The very first problem is to identify the ring of invariants of complete weight enumerator of self-dual codes over $R$. This is going to be the main subject of this section.

The complete weight enumerator of a code $C$ of length $n$ over $R$ is given by

$$W_C^u(z_{00}, z_{01}, z_{10}, z_{11}) = \sum_{u \in C} z_{00}(u) z_{01}(u) z_{10}(u) z_{11}(u) \quad (4.9)$$

143
where

\[ \delta_{ij}(u) = |\{ s | u_s = i + u_j, 1 \leq s \leq n \}|, \]  

and \( 0 \leq i, j \leq 1. \)

In the previous chapter we defined two Gray maps. One of them was a swap map which takes each element \( a + ub \in \mathbb{F}_p + u\mathbb{F}_p, p = 3, 5 \) to \((b, a)\) and then we extended it to any codeword. The other Gray map defined in the next section maps \( a + ub \in \mathbb{F}_p + u\mathbb{F}_p \) to \((a, b)\) and we extended it to any codeword. The images of the two maps differ by a permutation of coordinates of the codewords. So, essentially the images of both Gray maps produce equivalent codes. Hence, we may consider the latter Gray map \( \phi \) which is simpler to work with. Let \( u = (a_1 + ub_1, a_2 + ub_2, \ldots, a_n + ub_n) \in C \). Then,

\[ \phi(u) = (a_1, b_1, a_2, b_2, \ldots, a_n, b_n). \]  

We recall that, \( \phi(C) \) is a binary linear code of length \( 2n \). Since, \( \phi(C) \) has length \( 2n \) we recall the definition of its 2-byte weight enumerator:

\[ W^{2b}_{\phi(C)}(z_{00}, z_{01}, z_{10}, z_{11}) = \sum_{u \in \phi(C)} z_{00}^{\mu_{00}(u)} z_{01}^{\mu_{01}(u)} z_{10}^{\mu_{10}(u)} z_{11}^{\mu_{11}(u)} \]  

where \( u = (u_1, u_2, \ldots, u_{2n-1}, u_{2n}) \in \phi(C) \), and

\[ \mu_{ij}(u) = |\{(u_l, u_{l+1}) = (i, j)| l = 1, 3, 5, \ldots, 2n - 1\}| \]  

for all \( i, j \in \mathbb{F}_2 \).

**Example:** We give an example of a self-dual code \( C \) over \( R \) with generator matrix:

\[
G := \begin{bmatrix}
1 & 0 & 1 + u & u \\
0 & 1 & u & 1 + u
\end{bmatrix}.
\]
The complete weight enumerator of $C$ is

$$W_C^u(z_{00}, z_{01}, z_{10}, z_{11}) = z_{00}^2 + z_{01}^2 + z_{10}^2 + z_{11}^2 + 2z_{00}^2z_{11}^2 + 2z_{01}^2z_{10}^2 + 8z_{00}z_{01}z_{10}z_{11}.$$ 

In order to establish a MacWilliams-type identity between the complete weight enumerators of a code $C$ over $R$ and its dual, we are going to consider the image code $\phi(C)$ and their 2-byte weight enumerators. In Chapter II we obtained the MacWilliams identity for 2-byte weight enumerators (2.70). Using this identity and the Gray map $\phi$, we will obtain a MacWilliams-type identity for complete weight enumerators of codes over $R$.

Using the definitions of 2-byte and complete weight enumerators together with the definition of $\phi$, we have the following lemma:

**Lemma 4.2.1** Let $C$ be a code over $R$ of length $n$. Let $\phi$ be the Gray map defined above in (4.11). Then,

$$W_C^u(z_{00}, z_{01}, z_{10}, z_{11}) = W_{\phi(C)}^u(z_{00}, z_{01}, z_{10}, z_{11}).$$  \hspace{1cm} (4.14)

Another important observation is the following lemma:

**Lemma 4.2.2** Let $C$ be a code over $R$. Then,

$$(\phi(C))^\perp = \phi(C^\perp).$$  \hspace{1cm} (4.15)
Proof: By definition, we have

$$\left( \phi(C) \right)^\perp = \{ v \in \mathbb{F}_2^n | \forall c \in C \Rightarrow \phi(c) \}.$$ 

Let $\phi(x) \in \phi(C^\perp)$where $x = (a_1 + u b_1, a_2 + u b_2, \ldots, a_n + u b_n) \in C^\perp$. Let $v = (c_1 + u d_1, c_2 + u d_2, \ldots, c_n + u d_n)$ be an arbitrary element of $C$. Then,

$$\langle \phi(x), \phi(v) \rangle = \langle (a_1, b_1, \ldots, a_n, b_n), (c_1, d_1, \ldots, c_n, d_n) \rangle$$

$$= \sum_{i=1}^{n} a_i c_i + \sum_{i=1}^{n} b_i d_i = 0$$

since $0 = \langle x, v \rangle = \sum_{i=0}^{n}(a_i c_i + b_i d_i) + u \sum_{i=0}^{n}(a_i d_i + b_i c_i)$. This implies that $\sum_{i=0}^{n}(a_i c_i + b_i d_i) = 0$. Thus, $\phi(x) \in \phi(C^\perp)$ for all $\phi(x)$, i.e. $(\phi(C))^\perp \subset \phi(C^\perp)$. Now, assume that $v = (c_1, d_1, \ldots, c_n, d_n) \in (\phi(C))^\perp$. Then, $\langle v, \phi(c) \rangle = 0$ for all $c \in C$. Let $c = (a_1 + u b_1, a_2 + u b_2, \ldots, a_n + u b_n) \in C$. Thus,

$$0 = v \phi(c) = \sum_{i=1}^{n} a_i c_i + \sum_{i=1}^{n} b_i d_i.$$ 

Further, $uc \in C$ since $C$ is a submodule of $R^n$. This implies that

$$0 = \langle v, \phi(u c) \rangle = \sum_{i=0}^{n}(a_i d_i + b_i c_i).$$

Then,

$$\langle \phi^{-1}(v), c \rangle = \langle (c_1 + u d_1, \ldots, c_n + u d_n), (a_1 + u b_1, a_2 + u b_2, \ldots, a_n + u b_n) \rangle$$

$$= \sum_{i=0}^{n}(a_i c_i + b_i d_i) + u \sum_{i=0}^{n}(a_i d_i + b_i c_i)$$

$$= 0 + u 0 = 0.$$ 

Thus, $v \in \phi(C^\perp)$, i.e. $\phi(C))^\perp \subset \phi(C^\perp)$. □
**Theorem 4.2.1** Let $C$ be a code of length $n$ over $R$. Then, the relation between the complete weight enumerator of $C$ and its dual is given by

$$W_{C^*}^u (z_{00}, z_{01}, z_{10}, z_{11}) = \frac{1}{|C|} W_{C_{\phi}}^u (z_{00} + z_{01} + z_{10} + z_{11}),$$

$$(4.16)$$

Proof: Applying Corollary 2.6.1 to $\phi(C)$ which is a code of length $2n$, we get

$$W_{\phi(C)}^b (z_{00}, z_{01}, z_{10}, z_{11}) = \frac{1}{|C|} W_{\phi(C)}^b (z_{00} + z_{01} + z_{10} + z_{11}),$$

$$z_{00} - z_{01} + z_{10} - z_{11}, z_{00} + z_{01} - z_{10} - z_{11}, z_{00} - z_{01} - z_{10} + z_{11}).$$

By Lemma 4.2.2, we have

$$W_{\phi(C)}^b (z_{00}, z_{01}, z_{10}, z_{11}) = \frac{1}{|C|} W_{\phi(C)}^b (z_{00} + z_{01} + z_{10} + z_{11}),$$

$$z_{00} - z_{01} + z_{10} - z_{11}, z_{00} + z_{01} - z_{10} - z_{11}, z_{00} - z_{01} - z_{10} + z_{11}).$$

By Lemma 4.14 we have $W_{\phi(C)}^b (z_{00}, z_{01}, z_{10}, z_{11}) = W_{\phi(C)}^b (z_{00}, z_{01}, z_{10}, z_{11})$ and since $\phi$ is injective $|\phi(C)| = |C|$. Now, if we combine the results above, then we get the identity $(4.16)$. □

**4.2.2 Ring of invariants of codes over $\mathbb{F}_2 + u\mathbb{F}_2$**

In this section we are going to determine the ring of invariants of the complete weight enumerators of self-dual codes over $R = \mathbb{F}_2 + u\mathbb{F}_2$ with $u^2 = 1$. This is equivalent to determination of the ring of invariants of 2-byte weight enumerators of binary self-dual codes by Lemma 4.2.1 and Lemma 4.2.2.
We are going to investigate the ring of invariants of linear codes over $R$ whose images are Type I and II codes over $\mathbb{F}_2$. Recently, the ring of invariants of certain classes of codes over $\mathbb{F}_2 + u\mathbb{F}_2$ with $u^2 = 0$ is determined in [12] and [14] by Dougherty et al.

**Definition 4.2.1** A linear code $C$ over $R$ is said to be a Type I-R or Type II-R code if and only if $C$ is a self-dual code and its image over $\mathbb{F}_2$ is a Type I or Type II code respectively.

**Ring of invariants of Type I-R codes**

Let $C$ be a Type I-R code of length $n$. By definition, its image $\phi(C)$, $\phi$ is given by (4.11), is a Type I code of length $2n$.

For brevity we will use

$$Z = (z_{00}, z_{01}, z_{10}, z_{11}).$$

Let

$$f_1(Z) := 12z_{00}^2 + 12z_{11}^2 + 8z_{00}z_{11} + 4z_{01}^2 + 8z_{01}z_{10} + 4z_{10}^2,$$

$$f_2(Z) := 16(z_{00}^2 + z_{01}^2 + z_{10}^2 + z_{11}^2),$$

$$f_3(Z) := 16(z_{00}^2 - 2z_{00}z_{11} + z_{11}^2 + z_{01}^2 + 2z_{01}z_{10} + z_{10}^2),$$

and

$$f_4(Z) := -8z_{00}z_{01}z_{10}z_{11} + 6z_{00}^2z_{00}^2 + 6z_{10}^2z_{00}^2 + 6z_{01}^2z_{11}^2 +$$

$$+ 6z_{10}^2z_{11}^2 + z_{00}^4 - 2z_{00}^2z_{01}^2 - z_{01}^2z_{10}^2 + z_{01}^4 + z_{10}^4 + z_{11}^4.$$
Theorem 4.2.2 Let $C$ be a Type $I$-$R$ code of length $n$. Then, the 2-byte weight enumerator of $C$

$$W_{C}^{2b}(Z) \in \mathbb{C}[f_{1}(Z), f_{2}(Z), f_{3}(Z), f_{4}(Z)].$$

Proof: First we identify the group, say $G$ which leaves the 2-byte weight enumerator of a Type $I$-$R$ code invariant. By Theorem 4.2.1 we see that 2-byte weight is invariant under:

$$M_{1} := \frac{1}{2} \begin{bmatrix}
1 & 1 & 1 & 1 \\
1 & -1 & 1 & -1 \\
1 & 1 & -1 & -1 \\
1 & -1 & -1 & 1
\end{bmatrix}. \quad (4.17)$$

Since $C = C^{\perp}$, by Lemma 4.2.2, this implies that $\phi(C)$ is a binary self-dual code. Thus, the all one codeword, say $1$, is in $\phi(C)$. By definition of $\phi$, $1 + u := (1 + u, 1 + u, \ldots, 1 + u) \in C$. Since $C$ is linear $1 + u + C = C$. So the following holds:

$$W_{\phi(C)}^{2b}(z_{00}, z_{01}, z_{10}, z_{11}) = \sum_{v \in \phi(C)} z_{00}^{\mu_{00}(v)} z_{01}^{\mu_{01}(v)} z_{10}^{\mu_{10}(v)} z_{11}^{\mu_{11}(v)}$$

$$= \sum_{1 + u + v \in \phi(C)} z_{00}^{\mu_{00}(1+u+v)} z_{01}^{\mu_{01}(1+u+v)} z_{10}^{\mu_{10}(1+u+v)} z_{11}^{\mu_{11}(1+u+v)}$$

$$= \sum_{v \in \phi(C)} z_{00}^{\mu_{11}(v)} z_{01}^{\mu_{10}(v)} z_{10}^{\mu_{01}(v)} z_{11}^{\mu_{00}(v)}$$

$$= W_{\phi(C)}^{2b}(z_{11}, z_{10}, z_{01}, z_{00}).$$

149
In matrix form, 2-byte weight enumerator of \( C \) is left invariant under
\[
M_2 := \begin{bmatrix}
0 & 0 & 0 & 1 \\
0 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 \\
1 & 0 & 0 & 0
\end{bmatrix}
\] (4.18)

Further, we consider the map \( \phi_u(v) = uv \) where \( u \in R \) and \( v \in C \). This map is injective since \( \phi^2(v) = 1 \) for all \( v \in C \) and \( u^2 = 1 \). This implies that \( uC = C \). Now, clearly we have
\[
W_{2b}^C(z_{00}, z_{01}, z_{10}, z_{11}) = W_{2b}^C(z_{00}, z_{10}, z_{01}, z_{11})
\]

Thus, the matrix
\[
M_3 := \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\] (4.19)

leaves the 2-byte weight enumerator invariant.

Moreover, \( \langle v, u \rangle = 0 \) for all \( v \in C \) since \( C = C^\perp \). Since \( \langle v, u \rangle = \mu_{01}(v) + \mu_{10}(v) \) for all \( v \in C \), we see that \( \mu_{01}(v) + \mu_{10}(v) \equiv 0 \mod 2 \). Since \( C \) is a self-dual code the length of \( C \) say \( n \) is divisible by 2. We have \( n = \mu_{01}(v) + \mu_{10}(v) + \mu_{00}(v) + \mu_{11}(v) \) and this implies that \( \mu_{00}(v) + \mu_{11}(v) \equiv 0 \mod 2 \). Hence, the 2-byte weight enumerator is also invariant under the following matrices:

150
Thus, the 2-byte weight enumerator of a Type I-R code is invariant under a group $G$ which is generated by the matrices: $M_1, M_2, M_3, M_4$ and $M_5$. Computation of this group has been carried out in GAP. $G$ has order 32.

Symbolic computations are done in MAPLE. The Molien series of $G$ is

$$
\Phi_G(\lambda) = \frac{1}{(1 - \lambda^2)^4}. \tag{4.21}
$$

The Molien series of $G$ suggests to look for 4 free invariants of degrees 2.

We have averaged the following polynomials:

$$
f(Z) = z_{00}^2, \quad f(Z) = (z_{01} + z_{11})^2, \quad f(Z) = (z_{00} - z_{11})^2, \quad \text{and} \quad f(Z) = z_{01}^2 - z_{00}^2
$$

over $G$.

Respectively, the averaged polynomials of the above polynomials are $f_1(Z), f_2(Z), f_3(Z)$, and $f_4(Z)$.

The Jacobian of the polynomials $f_1(Z), f_2(Z), f_3(Z)$, and $f_4(Z)$ is nonzero. Hence, by Theorem 1.3.5 $f_1(Z), f_2(Z), f_3(Z)$, and $f_4(Z)$ are algebraically independent. Therefore, we have the result. $\square$
Let

\[ g_1(Z) := 12z_{00}^2z_{01} + 12z_{00}z_{01}^2z_{11} + 12z_{00}z_{10}^2z_{11} + 9z_{00}^3 + 9z_{11}^3 + 12z_{01}z_{10}z_{11}^2 + z_{01}^4 + 4z_{00}z_{11}^2 + 4z_{00}^2z_{11} + 6z_{00}z_{01}^2 + 6z_{00}z_{10}^2 + 6z_{00}z_{11}^2 + 4z_{01}z_{10}^3 + 4z_{01}z_{10}^3 + 4z_{01}z_{10}^3 \]

\[ + 4z_{01}z_{10}^2 + 6z_{01}^2z_{11} + 6z_{10}^2z_{11} + 24z_{00}z_{01}z_{10}z_{11}^2, \]

\[ g_2(Z) := z_{00}^4 + 4z_{00}z_{11} + 6z_{00}z_{11}^2 + 4z_{00}z_{11}^3 + z_{11}^4, \]

\[ g_3(Z) := z_{01}^4 - 4z_{01}z_{10} + 6z_{01}z_{10}^3 - 4z_{01}z_{10}^3 + z_{10}^4, \]

\[ g_4(Z) := z_{00}^4 + z_{11}^4 + z_{01}^4 + 3z_{00}z_{01}^3 + 3z_{00}z_{10}^3 + 3z_{01}z_{10}^3 + 3z_{10}z_{11}^3, \]

\[ g_6(Z) := 108(z_{00}^2z_{01}z_{10} + z_{00}z_{01}z_{11} + z_{00}z_{10}z_{11} + z_{01}z_{10}z_{11}) + z_{01}^4 + z_{10}^4 + 6z_{01}z_{10}^3 + 217(z_{00}^4 + z_{11}^4) + 644(z_{00}z_{11} + z_{00}z_{11}^2) + 54(z_{00}^2z_{01}^2 + z_{00}z_{10}^2 + z_{01}z_{10}^2 + z_{10}z_{11}^2)
+ 870z_{00}z_{11}^2 + 4(z_{01}z_{10} + z_{10}z_{11}) + 216z_{00}z_{01}z_{10}z_{11}, \]

\[ g_7(Z) := z_{00}^4 + 4z_{00}z_{11}^3 - 4z_{00}z_{11}^3 + 4z_{00}z_{11}^3 + 6z_{00}z_{11}^3 + 4z_{01}z_{10}^3 + 4z_{01}z_{10}^3 + 6z_{01}z_{10}^3. \]

**Theorem 4.2.3** Let \( C \) be a Type II-R code of length \( n \). Then, the 2-byte weight enumerator of \( C \)

\[ W_{C}^{2b}(Z) \in \mathbb{C}[g_1(Z), g_2(Z), g_3(Z), g_4(Z)] \oplus g_5(Z)\mathbb{C}[g_1(Z), g_2(Z), g_3(Z), g_4(Z)] \]

\[ \oplus g_6(Z)\mathbb{C}[g_1(Z), g_2(Z), g_3(Z), g_4(Z)] \oplus g_7(Z)\mathbb{C}[g_1(Z), g_2(Z), g_3(Z), g_4(Z)]. \]

**Proof:** Since Type II-R codes are Type I-R it is clear that the 2-byte weight enumerator of \( C \) is invariant under the group \( G \) which leaves the 2-byte weight enumerators of Type I-R codes invariant. Further, by definition \( \phi(C) \) is a Type II code.
Hence, by Corollary 1.3.4, the length of $\phi(C)$ which is $2n$ is divisible by 8. This implies that $n$ is divisible by 4. In other words,

$$\mu_{00}(v) + \mu_{01}(v) + \mu_{10}(v) + \mu_{11}(v) = n \equiv 0 \mod 4 \text{ for all } v \in C.$$ 

Thus, the 2-byte weight enumerator of $C$ is invariant under the following matrices:

$$N := \begin{bmatrix} i & 0 & 0 & 0 \\ 0 & i & 0 & 0 \\ 0 & 0 & i & 0 \\ 0 & 0 & 0 & i \end{bmatrix}, \quad N^{-} := \begin{bmatrix} -i & 0 & 0 & 0 \\ 0 & -i & 0 & 0 \\ 0 & 0 & -i & 0 \\ 0 & 0 & 0 & -i \end{bmatrix}.$$

Thus, the 2-byte weight enumerator of $C$ is invariant under $G$ and matrices $N, N^{-}$ which generate a group, say $H$, of order 64. The Molien series of $H$ is

$$\Phi_{G}(\lambda) = \frac{3\lambda^{4} + 1}{(1 - \lambda^{4})^{4}}. \quad (4.22)$$

The Molien series suggests to search for 4 free invariants of degrees 4 and for 3 transient invariants of degrees 4. Averaging the following polynomials

$$z_{00}, (z_{00} + z11)^4, (z_{00} - z11)^4, (z_{01} - z10)^4, (z_{01} + z01)^4, \text{ and } (z_{00} + z01 - z10 - 2z11)^4.$$ 

we obtain $g_1(Z), g_2(Z), g_3(Z), g_4(Z), g_5(Z), g_6(Z)$ and $g_7(Z)$ respectively.

The polynomials $g_1(Z), g_2(Z), g_3(Z)$ and $g_4(Z)$ are algebraically independent. Hence, the result. \[\Box\]
BIBLIOGRAPHY


154


155

[24] T. Aaron Gulliver, Masaaki Harada, Codes over \(\mathbb{F}_3 + \alpha \mathbb{F}_3\) and Improvements to the Bounds on Ternary Linear Codes. Des. Codes and Crypt., submitted in May 1998.


