Binary Self-Complementary Codes

by

Ifeyinwa Asemota
B.Sc., Benson Idahosa University, Nigeria, 2004

A Report Submitted in Partial Fulfillment of the
Requirements for the Degree of

MASTER OF ENGINEERING

in the Department of Electrical and Computer Engineering

© Ifeyinwa Asemota, 2016
University of Victoria

All rights reserved. This report may not be reproduced in whole or in part, by
photocopying or other means, without the permission of the author.

Binary Self-Complementary Codes

by

Ifeyinwa Asemota

B.Sc., Benson Idahosa University, Nigeria, 2004

Supervisory Committee

Dr. T. Aaron Gulliver, Supervisor

(Department of Electrical and Computer Engineering)

Dr. Mihai Sima, Departmental Member

(Department of Electrical and Computer Engineering)

Supervisory Committee

Dr. T. Aaron Gulliver, Supervisor

(Department of Electrical and Computer Engineering)

Dr. Mihai Sima, Departmental Member

(Department of Electrical and Computer Engineering)

ABSTRACT

A binary linear code C is said to be self-complementary if the all-ones codeword belongs to C. This report focuses on binary self-complementary codes and their weight distributions which in some cases are generated using the MAGMA Computational Algebra System. Several classes of binary codes, namely Hamming, simplex, single parity check, repetition, extended Hamming and Reed-Muller, are examined. This report considers binary codes up to length 12, and the best self-complementary codes are compared to the best linear codes in terms of their minimum Hamming distance. Future research can consider binary codes of longer lengths and nonbinary codes.

Contents

List of Tables

ACKNOWLEDGEMENTS

Many thanks go to my supervisor, Dr. T. Aaron Gulliver for providing his support academically and personally to the successful completion of this project. I would also like to acknowledge Dr. Mihai Sima for serving as a supervisory committee member.

I acknowledge my family who supported me all the way with their love and words of encouragement. My husband and our lovely son, Osamudiamen and Hilary Asemota, you both were my backbone and rock throughout my academic journey. My wonderful parents, Dr. and Dr. (Mrs) Osemene for all your prayers and encouragement.

My wonderful uncle and his wife, Azu and Maria Osemene, I cannot thank you enough for your emotional support and sacrifice. I also acknowledge my sisters, aunt Rhoda Osemene, Sam and Dr. Nora Osemene, thank you so much. My sister in-laws, Nosa Idemudia and Isoken Omoirauwa, God bless you both for your love and encouragement and a big thank you to Mummy Ayo and father in-law Mr. O.N.E. Asemota.

I would like to acknowledge two wonderful personalities, Lori Nolt and Anne Cirillo, you both made my stay at the University of Victoria a memorable one. You made my academic journey smooth and without worries. My family and I appreciate all you did and we say a Big THANK YOU.

DEDICATION

This work is dedicated to my family, the Leston family, the Jumbo family, Alanna and Lois Lewis, the Farinu family, and friends.

Chapter 1

Introduction

Error control coding is an important part of modern data communication and storage systems since the accuracy of a sequence of received data symbols is not guaranteed. An error-correcting code can be used to encode a message so that the original message can be retrieved even if a number of errors occur in transmission. These codes add redundancy to the original message before it is transmitted through a noisy channel. At the receiver, error correction is performed to recover the original message using the redundant information.

Block coding of information organizes the messages into groups of k bits so there are 2^k possible messages. The encoder takes each block of message bits and converts it into a block of $n \geq k$ coded bits called the codeword. The rate of the code, $R = k/n$, is the ratio of message bits to coded bits [6].

The binary field F_2 can be represented by two elements $\{0, 1\}$ with binary operations addition denoted by $\oplus$ and multiplication denoted by $\times$. Tables for binary (modulo-2) addition and multiplication are given below [6].

Modulo-2 addition

$\oplus$	0	1
0	0	1
1	1	0

Modulo-2 multiplication

$\times$	0	1
0	0	0
1	0	1

1.1 Vector Spaces

The vector space F_2^n consists of the set of vectors of length n over F_2 on which two operations vector addition $\oplus$ and scaler multiplication $\times$ are defined. The vector addition operation $\oplus$ is defined as follows. If $u = (u_1, u_2, \ldots, u_{n-1})$ and $v = (v_1, v_2, \ldots, v_{n-1})$, then

$$u \oplus v = (u_1 \oplus v_1, u_2 \oplus v_2, \ldots, u_n \oplus v_n) \tag{1.1}$$

A vector space is closed under vector addition so the addition of any two vectors in a vector space is another vector in the vector space [6]. For convenience let $F_2^n = V_n$.

Example: Consider the vector space V_3 which consists of the following $2^3 = 8$ vectors

$$V_3 = \{(000), (001), (010), (011), (100), (101), (110), (111)\}$$

The addition of any two of these vectors produces another vector in the vector space, for example

$$(011) \oplus (101) = (110)$$

For a vector (011), scalar multiplication by 0 and 1 is

$$\begin{aligned} 0 \times (011) &= (000) \\ 1 \times (011) &= (011) \end{aligned}$$

1.1.1 Basis

A basis of a vector space V_n is a subset $v_1, \ldots, v_n$ of vectors in V_n that are linearly independent and span V_n. Let $v_1, v_2, \ldots, v_n$ be vectors in V_n. These vectors form a basis if and only if every $v \in V_n$ can be uniquely written as

$$v = a_1 v_1 + a_1 v_2 + \ldots + a_n v_n \tag{1.2}$$

where $a_1, a_2, \ldots, a_n$ are elements of F_2.

1.1.2 Vector Subspaces

For a vector space V_n there exist subsets S of vectors which obey all the conditions for a vector space. Such a subset is called a subspace of the vector space V_n. A subset S is a subspace if the following conditions hold.

1. For any two vectors $u, v \in S$, $(u + v) \in S$.

2. For any element $a \in F_2$ and any vector $u \in S$, $a \times u \in S$.

Example: The following subset is a subspace of the vector space V_3

$$S = \{(000), (101), (110), (011)\}$$

Consider the vectors (011) and (010) of the vector space V_3. They form a basis of the subspace S given above

$$\begin{aligned} 0 \times (011) \oplus 0 \times (110) &= (000) \\ 0 \times (011) \oplus 1 \times (110) &= (110) \\ 1 \times (011) \oplus 0 \times (110) &= (011) \\ 1 \times (011) \oplus 1 \times (110) &= (101) \end{aligned}$$

The dimension of a subspace is the number of basis vectors.

1.1.3 Dual Spaces

If S is a k-dimensional subspace of the n-dimensional vector space V_n, then the set of vectors orthogonal to S is called the dual space $S^{\perp}$ of S [6].

Example: Consider the vector space V_4 and the following two dimensional subspace

$$S = \begin{matrix} 0000 \\ 0111 \\ 0011 \\ 0100 \end{matrix}$$

The dual space of S is

$$S^{\perp} = \begin{matrix} 0000 \\ 0011 \\ 1000 \\ 1011 \end{matrix}$$

The sum of the dimension of a subspace S, $dim(S)$, and its dual space $S^{\perp}$, $dim(S^{\perp})$, is equal to the dimension of the vector space $dim(V)$, i.e.

$$dim(S) + dim(S^{\perp}) = dim(V).$$

Example: Consider the subspace S generated by the basis

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

The dimension of the vector space is $dim(V_4) = 4$ and the dimension of S is $k = 2$. Therefore the dimension of the dual space $S^{\perp}$ is $4 - 2 = 2$.

1.2 Binary Linear Block Codes

Messages to be encoded are grouped into blocks of k bits $m = (m_0, m_1, \ldots, m_{k-1})$, so there are 2^k possible messages. The encoder takes a message and generates the codeword $c = (c_0, c_1, \ldots, c_{n-1})$, where $n \geq k$. A code of length n and 2^k codewords is said to be a linear block code C if the codewords form a vector subspace of dimension k over F_2. Since a linear code is a vector subspace of the vector space V_n, there will be k linearly independent vectors that are codewords $(g_0, g_1, \ldots, g_{k-1})$, such that each codeword is a linear combination of them. A linear block code can then be defined by a generator matrix G which is a basis of the corresponding vector space [1].

A codeword c is then given by

$$c = mG. \tag{1.3}$$

The generator matrix G of a (n, k) linear block code is a $k \times n$ matrix with linearly independent rows which span the vector space and is given by

$$G = \begin{bmatrix} g_{0,0} & g_{0,1} & \cdots & g_{0,k-1} \\ g_{1,0} & g_{1,1} & \cdots & g_{1,k-1} \\ \vdots & \vdots & & \vdots \\ g_{k-1,0} & g_{k-1,1} & \cdots & g_{k-1,n-1} \end{bmatrix} \tag{1.4}$$

The parity check matrix H of a (n, k) linear block code C is an $(n, n-k)$ matrix whose rows are linearly independent and form a basis of the dual code $C^{\perp}$ of C. The parity check matrix is thus a generator matrix of the dual code $C^{\perp}$. Therefore the parity check matrix H of a code C has dimensions $(n-k) \times n$ which can be expressed as

$$H = \begin{bmatrix} h_{0,0} & h_{0,1} & \cdots & h_{0,n-1} \\ h_{1,0} & h_{1,1} & \cdots & h_{1,n-1} \\ \vdots & \vdots & & \vdots \\ h_{n-k-1,0} & h_{n-k-1,1} & \cdots & h_{n-k-1,n-1} \end{bmatrix} \tag{1.5}$$

The Hamming weight of a codeword is defined as the number of nonzero elements in the codeword. For binary codes, the Hamming weight is the number of 1s in the codeword. For instance, the Hamming weight of the codeword $c_1 = (0011011)$ is 4. The Hamming distance is the number of positions in which two codewords x and y differ

$$d(x, y) = \{|\ i : x_i \neq y_i\ |\} \tag{1.6}$$

The Hamming distance between the codewords $c_1 = (0011011)$ and $c_2 = (1010101)$ is $d(c_1, c_2) =$

4. The weight of the sum of two vectors is the Hamming distance between them. The minimum Hamming distance of a code C is defined as the smallest Hamming distance between all pairs of codewords in the code

$$d_{min} = \min\{d(c_1, c_2)\}, c_1, c_2 \in C : c_1 \neq c_2 \tag{1.7}$$

A linear code of length n, dimension k and minimum distance d_{min} is called an (n, k, d) code. A code with minimum distance d_{min} can detect $d_{min} - 1$ errors and correct $\lfloor (d_{min} - 1)/2 \rfloor$ [6].

Example: Consider the $(7, 4)$ binary code C with generator matrix

$$G = \begin{bmatrix} 1001011 \\ 0101110 \\ 0010111 \end{bmatrix}$$

The codewords of C are

$$\begin{aligned} C &= \{0000000, 1110000, 1001100, 1101001, 0111100, 1100110, 1000011, \\ &\quad 0011001, 1011010, 0100101, 0010110, 1010101, 0110011, 0001111, 1111111\}. \end{aligned}$$

The minimum Hamming distance of this code is $d_{min} = 3$ as this is the minimum Hamming weight code of the codewords.

1.3 Equivalent Codes

Two binary codes are said to be equivalent if one can be obtained from the other by a permutation of the rows and columns of the generator matrices and linear combinations of the rows. Equivalent linear codes have the same parameters (n, k, d).

Example: Let C_1 and C_2 be binary codes with generator matrices

$$G_1 = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$G_2 = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}$$

respectively. interchanging the first and second rows and permutating the columns transforms G_1 to G_2, and so shows that C_1 and C_2 are equivalent.

The weight distribution of an (n, k) block code C is a series of coefficients $A_0, A_1, \ldots, A_n$, where A_i is the number of codewords in C of weight i. The weight distribution of C_1 and C_2 is $A_0 = A_3 = A_4 = A_7 = 1$.

1.4 MacWilliams Identity

Let $A_0, A_1, \ldots, A_n$ be the weight distribution of an (n, k) linear code C and $B_0, B_1, \ldots, B_n$ be the weight distribution of the $(n, n-k)$ dual code $C^{\perp}$. These weight distributions can be represented in polynomial form as

$$A(x) = A_0 + A_1 x + \ldots + A_n x^n \tag{1.8}$$

and

$$B(x) = B_0 + B_1 x + \ldots + B_n x^n \tag{1.9}$$

The polynomials $A(x)$ and $B(x)$ are called the weight enumerators for C and $C^{\perp}$, respectively. The polynomials $A(x)$ and $B(x)$ are related by the following identity

$$A(x) = 2^{-k}(1+x)^n B\left[\frac{1-x}{1+x}\right] \tag{1.10}$$

which is known as the MacWilliams Identity. From MacWilliams Identity, if the weight distribution of $C^{\perp}$ is known, then the weight distribution of C can be determined [1].

Example: Determine the weight distribution for the $(7,5,2)$ dual code of the $(7,2,3)$ code with weight distribution

$$B(x) = 1 + x^3 + x^4 + x^7$$

From (1.10)

$$\begin{aligned}
A(x) &= 2^{-k}(1+x)^n B[\tfrac{1-x}{1+x}] \\
&= 2^{-2}(1+x)^7 B[\tfrac{1-x}{1+x}] \\
&= \tfrac{1}{4}(1+x)^7 B[\tfrac{1-x}{1+x}] \\
&= \left[\tfrac{1}{4}(1+x)^7[1 + x\tfrac{(1-x)^3}{(1+x)^3} + x\tfrac{(1-x)^4}{(1+x)^4} + x\tfrac{(1-x)^7}{(1+x)^7}\right] \\
&= 1 + 9x^2 + 19x^4 + 3x^6
\end{aligned}$$

1.5 Self-Complementary Codes

Let C be a binary code of length n. If for any codeword $a = (\beta_1, \beta_, \ldots, \beta_n) \in C$ the complement $\overline{a} = (1+\beta_1, 1+\beta_2, \ldots, 1+\beta_n)$ also belongs to C the code is called self-complementary. For a binary linear code C, this is equivalent to the fact that the all-ones codeword $(1,1,\ldots,1)$ belongs to C [2].

The number of codewords $\mid C \mid$ of a self-complementary code C with minimum Hamming distance d satisfies the Grey-Rankin bound [2]

$$| C | \leq \frac{8d(n-d)}{n-(n-2d)^2} \tag{1.11}$$

provided that the right hand-side is positive. Note that $| C |= 2^k$ for a binary linear code. The parameters of a binary self-complementary code meeting the Grey-Rankin bound are [10]

$$\left[2^{2m-1} - 2^{m-1}, 2_m + 1, 2^{2m-2} - 2^{m-1}\right] \tag{1.12}$$

or

$$\left[2^{2m-1} + 2^{m-1}, 2_m + 1, 2^{2m-2}\right] \tag{1.13}$$

Example: A binary code meeting the Grey-Rankin bound is the $(8, 4, 4)$ code given by

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

From (1.11), the bound on the number of codewords is

$$\begin{aligned} | C | &\leq \tfrac{8\times4(8-4)}{8-(8-2\times4)^2} \\ &\leq \tfrac{32(4)}{8-(8-8)^2} \\ &\leq \tfrac{128}{8} \\ &\leq \tfrac{128}{8} = 16 \end{aligned}$$

Since $k = 4$, the number of codewords is $2^4 = 16$, so the $(8, 4, 4)$ code meets the Grey-Rankin bound.

Chapter 2

Classes of Codes

Trivially the vector space V_n is an $(n, n, 1)$ code.

Example: The $[4, 4, 1]$ code has generator matrix

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

The weight distribution of this code is $A_0 = A_4 = 1, A_1 = A_3 = 4, A_2 = 6$. Since this code contains all vectors of length n, it is a self-complementary code.

2.1 Repetition Codes

Repetition codes are a simple class of linear block codes. A single message bit is encoded into a block of n identical bits, producing an $(n, 1, n)$ block code. The only two codewords in this code are the all-zero codeword and the all-ones codeword, so the minimum distance of a length n repetition code is n.

Example: The $(8, 1, 8)$ repetition code has generator matrix

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

The weight distribution of this code is $A_0 = A_8 = 1$. All repetitions codes are self-complementary because they contain the all-ones codeword $(1, 1, 1, \ldots, 1)$.

2.2 Single Parity Check (SPC) Codes

SPC codes are $(n, n-1, 2)$ codes. SPC codes are the duals of the repetition codes. The weight distribution of the $(8, 1, 8)$ code which is the dual to the $(8, 7, 2)$ code is

$$B(x) = 1 + x^8$$

The MacWilliams Identity can be used to determine the weight distribution $A(x)$ of the dual code

$$\begin{aligned} A(x) &= 2^{-k}(1+x)^n B[\tfrac{1-x}{1+x}] \\ &= 2^{-2}(1+x)^8 B[\tfrac{1-x}{1+x}] \\ &= \tfrac{1}{4}(1+x)^8 B[\tfrac{1-x}{1+x}] \\ &= 1 + 28x^2 + 70x^4 + 28x^6 + x^8 \end{aligned}$$

The coefficient of x^n is 1, hence the $(8, 7, 2)$ code is self-complementary. SPC codes contain all even weight vectors of length n, therefore they are self-complementary if n is even. For example, the $(7, 6, 2)$ SPC code cannot be self-complementary because the weight of the all-ones vector is odd.

2.3 Hamming Codes

Hamming codes are the first class of binary linear codes designed for error correction [9]. These codes and their variations have been widely used for error control in digital communication systems. For any integer m there is a Hamming code of length $n = 2^{m-1}$ with m parity bits and $k = 2^m - 1 - m$ message bits, where $m = n - k$. The parameters of some binary Hamming codes are given below.

m	Hamming Codes
3	$(7, 4, 3)$
4	$(15, 11, 3)$
5	$(31, 26, 3)$

The weight enumerator of a Hamming code of length $n = 2^m - 1$ is [9]

$$A_1(x) = \frac{1}{(n+1)} \left((1+x)^n + n(1-x)(1-x^2)^{(n-1)/2}\right) \tag{2.1}$$

Example: For $m = 3$, the weight enumerator for the $(7, 4, 3)$ binary Hamming code is

$$\begin{aligned} A_1(x) &= \tfrac{1}{(7+1)}(1+x)^7 + 7(1-x)(1-x^2)^{(7-1)/2} \\ &= \tfrac{1}{8}(1+x)^7 + 7(1-x)(1-x^2)^3 \\ &= 1 + 7x^3 + 7x^4 + x^7 \end{aligned}$$

The coefficient of $x^n = 1$, hence the $(7, 4, 3)$ Hamming code is self-complementary.

Theorem: $A(n) = 1$ for all Hamming codes.

Proof

Consider the weight enumerator of the Hamming codes

$$A_1(x) = \frac{1}{(n+1)}\left((1+x)^n + n(1-x)(1-x^2)^{(n-1)/2}\right).$$

The coefficient x^n is given by

$$\begin{aligned}
&\tfrac{1}{(n+1)}(x^n + n(-x)(-x^2)^{(n-1)/2}) \\
&= \tfrac{1}{(n+1)}x^n + nx \times (x^2)^{(n-1)/2}) \\
&= \tfrac{1}{(n+1)}(x^n + nx \times x^{n-1}) \\
&= \tfrac{1}{(n+1)}(x^n + n(x^n)) \\
&= x^n
\end{aligned}$$

2.4 Simplex Codes

Simplex codes are the duals of the Hamming codes. The dual code of a $(2^m - 1, 2^m - m - 1)$ Hamming code is a $(2^m - 1, m)$ linear code. This code has a very simple weight distribution which consists of the all-zero codeword and $2^m - 1$ codewords of weight 2^{m-1}, so the weight enumerator is [9]

$$B(x) = 1 + (2^m - 1)x^{2^{m-1}} \tag{2.2}$$

Example: For $m = 3$, the weight enumerator for the $(7, 3, 4)$ binary simplex code is

$$\begin{aligned}
B(x) &= 1 + (2^3 - 1)x^{2^{3-1}} \\
&= 1 + (2^3 - 1)x^{2^2} \\
&= 1 + 7x^4
\end{aligned}$$

Clearly the coefficient of x^n is 0, so the $(7, 3, 4)$ simplex code is not self-complementary. From (2.2), the coefficient of x^n is always 0, so the simplex codes are not self-complementary.

2.5 Extended Hamming Codes

An extension of a binary code results from adding to each codeword a new digit that is an even parity check of the codeword. Therefore, every codeword in an extended Hamming code has an even number of ones so the minimum distance of the code is increased from 3 to 4. Note that by extending a code, the dimension of the code remains the same and the length is increased by 1. Therefore, the extended binary Hamming codes have parameters $(2^m, 2^m - m - 1, 4)$.

Example: By adding an even parity check to the $(7, 4, 3)$ Hamming code, the code is extended to an $(8, 4, 4)$ code

$$G(7,4) = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}$$

$$G(8,4) = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

The weight enumerator of an extended Hamming code $A_2(x)$ can be determined from the weight enumerator of the Hamming code $A_1(x)$. $A_2(x)$ consists of only even weight codewords [9], and

$$A_2(x) = \frac{1}{2}\left[A_1(x) + A_1(-x)\right] \quad (2.3)$$

Since the weight enumerator of the Hamming code $A_1(x)$ is known, the weight enumerator of the extended Hamming code can be determined from (2.1) and (2.3) as

$$A_2(x) = \frac{1}{2(n+1)}\left((1+x)^n + (1+x)^n + 2n(1-x^2)^{(n-1)/2}\right) \quad (2.4)$$

Example: For $m = 3$, the weight enumerator for the $(8, 4, 4)$ extended Hamming code is

$$\begin{aligned} A_2(x) &= \tfrac{1}{2(n+1)}\left((1+x)^n + (1+x)^n + 2 \times n(1-x^2)^{(n-1)/2}\right) \\ &= \tfrac{1}{2(8+1)}\left((1+x)^8 + (1+x)^8 + 2 \times 8(1-x^2)^{(8-1)/2}\right) \\ &= \tfrac{1}{18}\left((1+x)^8 + (1+x)^8 + 16(1-x^2)^{7/2}\right) \\ &= 1 + 14x^4 + x^8. \end{aligned}$$

The coefficient of x^n is 1, hence the $(8, 4, 4)$ extended Hamming code is self-complementary. Hamming codes have odd length $2^m - 1$ and are self-complementary. Therefore they contain the all-ones codeword which has odd weight. Adding an even parity check results in an all-ones codeword of even weight. Therefore, all extended Hamming codes are self-complementary.

2.6 Reed-Muller Codes

Reed-Muller codes are amongst the oldest and most well known classes of codes. They were discovered independently by D. E. Muller and I. S. Reed in 1954 [1]. Reed-Muller codes are often used as building blocks to obtain other codes. One of the major advantages of Reed-Muller codes is their relative simple encoding and decoding [1].

A Reed-Muller code is denoted as $RM(r, m)$, where the length $n = 2^m$ and r is the order of the code $0 \leq r \leq m$. The minimum distance is 2^{m-r} and the dimension is

$$k = \binom{m}{0} + \binom{m}{1} + \cdots + \binom{m}{r}.$$

$RM(0, m)$ is the repetition code of length $n = 2^m$ and $RM(m-1, m)$ is a single parity check code of length $n = 2^m$.

2.6.1 Plotkin Construction

The Plotkin $(u|u+v)$ construction is a simple method of combining two linear codes to obtain a new code. Given an (n, k_1, d_1) code C_1 and an (n, k_2, d_2) code C_2, we define a new code

$$C = \{(u, u+v) : u \in C_1, v \in C_2\} \tag{2.5}$$

The generator matrix for C is

$$G = \begin{bmatrix} G_u & G_u \\ 0 & G_v \end{bmatrix} \tag{2.6}$$

2.6.2 Recursive Construction of RM Codes

Reed-Muller codes can be obtained recursively using the Plotkin construction. Let C_1 be $RM(r, m-1)$ and C_2 be $RM(r-1, m-1)$. The recursive construction given by (3.1) is

$$RM(r, m) = \{(u|u+v) : u \in RM(r, m-1), v \in RM(r-1, m-1)\} \tag{2.7}$$

From the recursive construction, the generator matrix of a Reed-Muller code is given by

$$G(r, m) = \begin{bmatrix} G(r, m-1) & G(r, m-1) \\ 0 & G(r-1, m-1) \end{bmatrix} \tag{2.8}$$

Example: For $r = 0$, $G(0,0) = [1]$, $G(0,1) = [11]$, and $G(0,2) = [1111]$ are the repetition codes which contain the all-ones codewords, and consider the generator matrix $G(1, m)$. Using the recursive construction, we have

$$G(1,1) = \begin{bmatrix} G(r,m-1) & G(r,m-1) \\ 0 & G(r-1,m-1) \end{bmatrix} = \begin{bmatrix} G(1,0) & G(1,0) \\ 0 & G(0,0) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Then $G(1,2)$ using the recursive construction is

$$G(1,2) = \begin{bmatrix} G(1,1) & G(1,1) \\ 0 & G(0,1) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

$G(1,2)$ and $G(0,2)$ can be used to obtain

$$G(1,3) = \begin{bmatrix} G(1,2) & G(1,2) \\ 0 & G(0,2) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Using the recursive construction, $G(1,4)$ code is given by

$$G(1,4) = \begin{bmatrix} G(1,3) & G(1,3) \\ 0 & G(0,3) \end{bmatrix} = \begin{bmatrix} 1010101010101010 \\ 0101010101010101 \\ 0011001100110011 \\ 0000111100001111 \\ 0000000011111111 \end{bmatrix}$$

Considering the recursive construction of RM codes, every generator matrix includes a repetition code. As defined earlier, the repetition codes are $(n,1,n)$ codes which contain the all-zero and all-ones codewords. Therefore all RM codes are self-complementary.

Chapter 3

Other Codes

This chapter presents the best binary linear self-complementary codes which are not in the classes of codes presented previously. The generator matrices and weight enumerators of these codes are as follows.

$(4, 2, 2)$

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

Weight Enumerator $1 + 2x^2 + x^4$

$(5, 2, 2)$

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

Weight Enumerator $1 + x^2 + x^3 + x^5$

$(5, 3, 2)$

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

Weight Enumerator $1 + 3x^2 + 3x^3 + x^5$

$(6, 2, 3)$

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

Weight Enumerator $1 + 2x^3 + x^6$

$(6, 3, 2)$

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

Weight Enumerator $1 + x^2 + 4x^3 + x^4 + x^6$

$(6, 4, 2)$

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

Weight Enumerator $1 + 7x^2 + 7x^4 + x^6$

$(7,2,3)$

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

Weight Enumerator $1 + x^3 + x^4 + x^7$

$(7,3,3)$

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

Weight Enumerator $1 + 3x^3 + 3x^4 + x^7$

$(7,5,2)$

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

Weight Enumerator $1 + 10x^2 + 5x^3 + 5x^4 + 10x^5 + x^7$

$(8,2,4)$

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

Weight Enumerator $1 + 2x^4 + x^8$

$(8, 3, 4)$

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

Weight Enumerator $1 + 6x^4 + x^8$

$(8, 5, 2)$

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

Weight Enumerator $1 + 8x^2 + 14x^4 + 8x^6 + x^8$

$(8, 6, 2)$

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

Weight Enumerator $1 + 16x^2 + 30x^4 + 16x^6 + x^8$

$(9, 2, 4)$

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

Weight Enumerator $1 + x^4 + x^5 + x^9$

$(9, 3, 4)$

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

Weight Enumerator $1 + 3x^4 + 3x^5 + x^9$

$(9, 4, 4)$

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

Weight Enumerator $1 + 7x^4 + 7x^5 + x^9$

$(9, 5, 3)$

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

Weight Enumerator $1 + 6x^3 + 9x^4 + 9x^5 + 6x^6 + x^9$

$(9, 6, 2)$

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

Weight Enumerator $1 + 11x^2 + 5x^3 + 15x^4 + 15x^5 + 5x^6 + 11x^7 + x^9$

$(9,7,2)$

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

Weight Enumerator $1 + 21x^2 + 7x^3 + 35x^4 + 35x^5 + 7x^6 + 21x^7 + x^9$

$(10,2,5)$

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

Weight Enumerator $1 + 2x^5 + x^{10}$

$(10,3,4)$

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

Weight Enumerator $1 + x^4 + 4x^5 + x^6 + x^{10}$

$(10, 4, 4)$

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

Weight Enumerator $1 + 7x^4 + 7x^6 + x^{10}$

$(10, 5, 3)$

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

Weight Enumerator $1 + 2x^3 + 7x^4 + 12x^5 + 7x^6 + 2x^7 + x^{10}$

$(10, 6, 3)$

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

Weight Enumerator $1 + 10x^3 + 15x^4 + 12x^5 + 15x^6 + 10x^7 + x^{10}$

$(10, 7, 2)$

$$G = \begin{bmatrix} 1000000001 \\ 0100000001 \\ 0010000001 \\ 0001000001 \\ 0000100001 \\ 0000010001 \\ 0000001111 \end{bmatrix}$$

Weight Enumerator $1 + 21x^2 + 42x^4 + 42x^6 + 21x^8 + x^{10}$

$(10, 8, 2)$

$$G = \begin{bmatrix} 1000000001 \\ 0100000001 \\ 0010000001 \\ 0001000001 \\ 0000100001 \\ 0000010001 \\ 0000001001 \\ 0000000110 \end{bmatrix}$$

Weight Enumerator $1 + 29x^2 + 98x^4 + 98x^6 + 29x^8 + x^{10}$

$(11, 2, 5)$

$$G = \begin{bmatrix} 11111000000 \\ 00000111111 \end{bmatrix}$$

Weight Enumerator $1 + x^5 + x^6 + x^{11}$

$(11, 3, 5)$

$$G = \begin{bmatrix} 10010101101 \\ 01011001011 \\ 00110011001 \end{bmatrix}$$

Weight Enumerator $1 + 3x^5 + 3x^6 + x^{11}$

$(11, 4, 4)$

$$G = \begin{bmatrix} 10010101101 \\ 01010110101 \\ 00110011001 \\ 00001111110 \end{bmatrix}$$

Weight Enumerator $1 + x^4 + 6x^5 + 6x^6 + x^7 + x^{11}$

$(11, 5, 3)$

$$G = \begin{bmatrix} 10000000011 \\ 01000000101 \\ 00100001001 \\ 00010010001 \\ 00001100001 \end{bmatrix}$$

Weight Enumerator $1 + 5x^3 + 10x^4 + 10x^7 + 5x^8 + x^{11}$

$(11, 6, 3)$

$$G = \begin{bmatrix} 10010000011 \\ 01010000101 \\ 00110000110 \\ 00001001011 \\ 00000101101 \\ 00000011001 \end{bmatrix}$$

Weight Enumerator $1 + 3x^3 + 16x^4 + 12x^5 + 12x^6 + 16x^7 + 3x^8 + x^{11}$

$(11, 7, 3)$

$$G = \begin{bmatrix} 10000000011 \\ 01000000101 \\ 00100000110 \\ 00010010110 \\ 00001010101 \\ 00000110011 \\ 00000001111 \end{bmatrix}$$

Weight Enumerator $1 + 13x^3 + 26x^4 + 24x^5 + 24x^6 + 26x^7 + 13x^8 + x^{11}$

$(11, 8, 2)$

$$G = \begin{bmatrix} 10000000001 \\ 01000000001 \\ 00100000001 \\ 00010000001 \\ 00001000001 \\ 00000100001 \\ 00000010001 \\ 00000001110 \end{bmatrix}$$

Weight Enumerator $1 + 28x^2 + x^3 + 70x^4 + 28x^5 + 28x^6 + 70x^7 + x^8 + 28x^9 + x^{11}$

$(11, 9, 2)$

$$G = \begin{bmatrix} 10000000001 \\ 01000000001 \\ 00100000001 \\ 00010000001 \\ 00001000001 \\ 00000100001 \\ 00000010001 \\ 00000001001 \\ 00000000111 \end{bmatrix}$$

Weight Enumerator $1 + 36x^2 + 9x^3 + 126x^4 + 84x^5 + 84x^6 + 126x^7 + x^8 + 9x^8 + 36x^9 + x^{11}$

$(12, 2, 6)$

$$G = \begin{bmatrix} 111111000000 \\ 000000111111 \end{bmatrix}$$

Weight enumerator $1 + 2x^6 + x^{12}$

$(12, 3, 6)$

$$G = \begin{bmatrix} 100101011010 \\ 010100101011 \\ 001110001110 \end{bmatrix}$$

Weight Enumerator $1 + 6x^6 + x^{12}$

$(12, 4, 5)$

$$G = \begin{bmatrix} 100100011010 \\ 010110010100 \\ 001101001110 \\ 000000111111 \end{bmatrix}$$

Weight Enumerator $1 + 4x^5 + 6x^6 + 4x^7 + x^{12}$

$(12, 5, 4)$

$$G = \begin{bmatrix} 100000000001 \\ 010000000010 \\ 001000000100 \\ 000100001000 \\ 000010010000 \end{bmatrix}$$

Weight Enumerator $1 + 9x^4 + 12x^6 + 9x^8 + x^{12}$

$(12, 6, 4)$

$$G = \begin{bmatrix} 100100000110 \\ 010100000101 \\ 001100000011 \\ 000010010110 \\ 000001010101 \\ 000000111100 \end{bmatrix}$$

Weight Enumerator $1 + 19x^4 + 24x^6 + 19x^8 + x^{12}$

$(12, 7, 4)$

$$G = \begin{bmatrix} 100100000110 \\ 010100000101 \\ 001100000011 \\ 000010010110 \\ 000001010101 \\ 000000110011 \\ 000000001111 \end{bmatrix}$$

Weight Distribution $1 + 39x^4 + 48x^6 + 39x^8 + x^{12}$

$(12, 8, 3)$

$$G = \begin{bmatrix} 100000000111 \\ 010000001010 \\ 001000001100 \\ 000100001111 \\ 000010001001 \\ 000001000101 \\ 000000100011 \\ 000000011110 \end{bmatrix}$$

Weight Enumerator $1 + 16x^3 + 39x^4 + 48x^5 + 48x^6 + 48x^7 + 39x^8 + 16x^9 + x^{12}$

$(12, 9, 2)$

$$G = \begin{bmatrix} 100000000001 \\ 010000000001 \\ 001000000100 \\ 000100001000 \\ 000010001000 \\ 000001001000 \\ 000000101000 \\ 000000011000 \\ 000000000011 \end{bmatrix}$$

Weight Enumerator $1 + 22x^2 + 127x^4 + 212x^6 + 127x^8 + 22x^{10} + x^{12}$

$(12, 10, 2)$

$$G = \begin{bmatrix} 100000000001 \\ 010000000001 \\ 001000000100 \\ 000100000100 \\ 000010000100 \\ 000001000100 \\ 000000100100 \\ 000000010100 \\ 000000001100 \\ 000000000011 \end{bmatrix}$$

Weight Enumerator $1 + 34x^2 + 239x^4 + 476x^6 + 239x^8 + 34x^{10} + x^{12}$

Chapter 4

Minimum Distances of the Best Binary Self-Complementary Codes

An (n, k) binary linear code C is said to be a best known code if C has the highest minimum distance among all known (n, k) binary linear codes. Table 4.1 gives the minimum distances of the best binary linear codes up to length $n = 12$ from the bounds on the minimum distance of linear block codes over GF(q) [7]. An (n, k) self-complementary binary linear code C is said to be a best known self-complementary code if C has the highest minimum distance among all known (n, k) self-complementary binary linear codes. Table 4.2 gives the minimum distances of the best self-complementary binary linear codes up to length 12. The far right diagonal shows the trivial codes, and the next diagonal the SPC codes with $d_{min} = 2$ for even length. The far left column are the repetition codes. Comparing Tables 4.1 and 4.2, most of the best known self-complementary binary linear codes up to length 12 have the same minimum distance as the best binary linear codes. For $k = 1$, the best self-complementary codes are the repetition codes, so they have the same minimum distances as the best binary linear codes. For $k = 2$, only the $(4, 2, 2)$ self-complementary code has a minimum distance equal to that of the best binary linear code. Other than the SPC codes, only 9 other self-complementary codes have a lower minimum distance, and in these cases the difference

Table 4.1: Bounds on the minimum distance of the best binary linear codes up to length 12 [7].

n/k	1	2	3	4	5	6	7	8	9	10	11	12
1	1	-	-	-	-	-	-	-	-	-	-	-
2	2	1	-	-	-	-	-	-	-	-	-	-
3	3	2	1	-	-	-	-	-	-	-	-	-
4	4	2	2	1	-	-	-	-	-	-	-	-
5	5	3	2	2	1	-	-	-	-	-	-	-
6	6	4	3	2	2	1	-	-	-	-	-	-
7	7	4	4	3	2	2	1	-	-	-	-	-
8	8	5	4	4	2	2	2	1	-	-	-	-
9	9	6	4	4	3	2	2	2	1	-	-	-
10	10	6	5	4	4	3	2	2	2	1	-	-
11	11	7	6	5	4	4	3	2	2	2	1	-
12	12	8	6	6	4	4	4	3	2	2	2	1

is only 1.

Table 4.2: Bounds on the minimum distance of the best self-complementary binary linear codes up to length 12.

n/k	1	2	3	4	5	6	7	8	9	10	11	12
1	1	-	-	-	-	-	-	-	-	-	-	-
2	2	1	-	-	-	-	-	-	-	-	-	-
3	3	1	1	-	-	-	-	-	-	-	-	-
4	4	2	2	1	-	-	-	-	-	-	-	-
5	5	2	2	1	1	-	-	-	-	-	-	-
6	6	3	2	2	2	1	-	-	-	-	-	-
7	7	3	3	3	2	1	1	-	-	-	-	-
8	8	4	4	4	2	2	2	1	-	-	-	-
9	9	4	4	4	3	2	2	1	1	-	-	-
10	10	5	4	4	3	3	2	2	2	1	-	-
11	11	5	5	4	3	3	3	2	2	1	1	-
12	12	6	6	5	4	4	4	3	2	2	2	1

Chapter 5

Conclusion and Future Work

This project considered binary self-complementary codes which are codes that contain the all-ones codeword. Various well-known classes of binary codes were considered. The weight distributions of these codes were examined to determine if they are self-complementary. This was achieved by considering the coefficient of x^n in the weight enumerator. Among the classes of codes examined, the simplex codes and the SPC codes of odd length are not self-complementary. Conversely, the trivial vector space, even length SPC, Hamming, extended Hamming, and Reed-Muller codes are self-complementary, Chapter 3 presented the best self-complementary codes found for parameters not among the classes of codes. Magma was used to determine the weight distributions. A comparison was given between the best binary linear codes and the best self-complementary binary linear codes. It was observed that the minimum distance of many of the best self-complementary codes is equal to that of the best binary linear codes. This project considered binary codes up to length 12. Future work can investigate longer code lengths and nonbinary codes.

Bibliography

[1] S.B. Wicker. 1994. Error Control Systems for Digital Communication and Storage. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

[2] A. Betten, M. Braun, H. Fripertinger, A. Kerber, A. Kohnert, and A. Wassermann, 2006. Error-correcting linear codes: Classification by isometry and applications, vol.18.

[3] S.M. Dodunekov, S.B. Encheva, and S.N. Kapralov. On the [28, 7, 12] binary self-complementary codes and their residuals. Designs, Codes and Cryptography,4, 57-67 (1994).

[4] S.Lin and DJ. Costello. 2004. Error Control Coding, Second Edition. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

[5] K. Olga. Reed-Muller Codes: Spherically-punctured codes and decoding algorithms, August 2013.

[6] J.C.E. Moreira and P.G.Farrell,2006.Essentials of error-control coding. Wiley, New York.

[7] http://codetables.markus-grassl.de/

[8] W.C. Huffman, and V. Pless, 2010. Fundamentals of Error-Correcting Codes. Cambridge university press.

[9] S. Lin and D.J. Costello, 1983. Error Control Coding, Fundamentals and Applications. Prentice-Hall, Inc., Englewood Cliffs, NJ, USA.

[10] I. Bouyukliev, S. Bouyuklieva, and S. Dodunekov. On binary self-complementary [120, 9, 56] codes having an automorphism of order 3 and associated SDP designs. Problems of Information Transmission 2007, 43(2), pp.89-96.